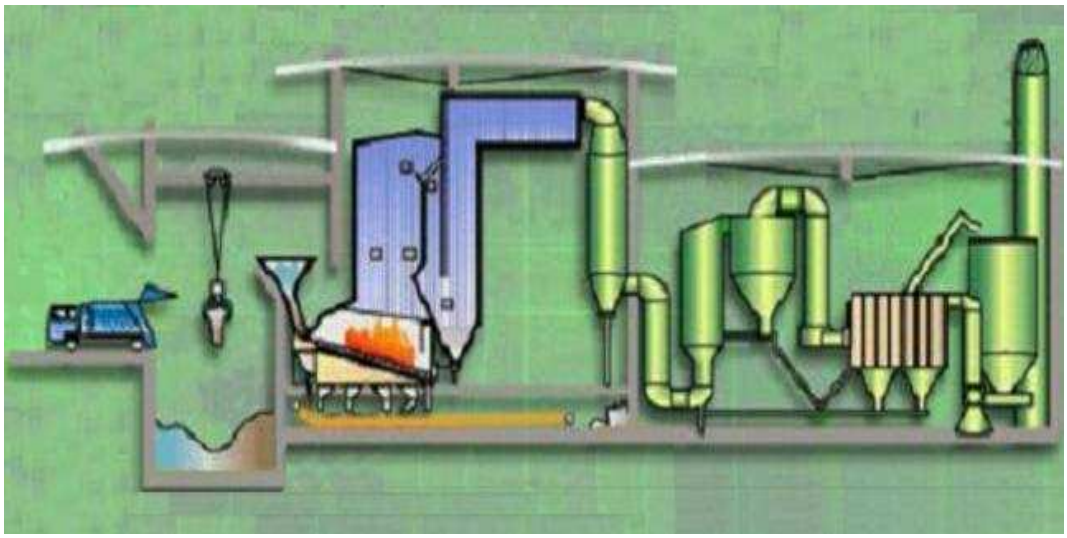
utc
Université de Technologie
Compiègne

par Astride Arégui

Détection de nouveauté dans le cadre de la théorie des fonctions de croyance. Application à la surveillance d'un système d'incinération de déchets.

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC.



Soutenue le : 28 novembre 2007
Spécialité : Technologie de l'information et des systèmes

Détection de nouveauté dans le cadre de la théorie des fonctions de croyance. Application à la surveillance d'un système d'incinération de déchets.

Thèse soutenue le 28 novembre 2007 devant le jury composé de :

Mr.	Stéphane Canu	Professor, Insa, Rouen (France)	(Rapporteur)
Mr.	Eyke Hüllermeier	Professor, Philipps-Universität, Marburg (Germany)	(Rapporteur)
M.	Walter Schön	Professor, UTC, Compiègne (France)	
Mme.	Xia Ding	Expert Senior, CIRCEE, Le Pecq (France)	
Mr.	Thierry Denœux	Professor, UTC, Compiègne (France)	(Directeur de thèse)

A mes parents et grands-parents,
A Manu et Amélie,
et à Philippe Smets.

Thanks

My greatest thanks go to Pr. Thierry Denœux, who supervised this work with care and rigour, and always took the time to assess the current work and new plans for the future on a regular basis. He helped me face theoretical, technical and practical problems but also human difficulties, and working with him taught me a lot in all these respects.

Xia Ding did an excellent job in interfacing industrial and academic points of view, and the project would never have been so well defined without her experience, the fineness of her analysis of practical situations, her technical knowledge, and her great kindness. It was a pleasure to work with her during the last three years.

I am obliged to Farrock Fotoohi, who entrusted me with the task of solving the problem that was presented to him by Novergie's Technical Direction, and to the latter, in the persons of José de Freitas and Laure de Saint Phalle, who initiated the project. I am also grateful to these last two people for introducing me to the waste incineration domain in a most interesting way, supporting me in all the important steps of the project.

The help I received from the staff of our pilot plant was excellent. They welcomed me warmly, provided me with all the information I needed, gave me a fantastic amount of details and hints on how to monitor the process they are in charge of... They included me and my project in their everyday routine with great enthusiasm, in spite of the additional work it represented for them, and this was extremely helpful and encouraging. The names of Mr. Guérin, Managing Director, Mr. Laverre, Technical Director, Frédéric, Technical Manager, and Jean-Luc, who was my most enthusiastic teacher, deserve a special mention.

I also wish to thank Séverine Pruvot for the amount of work she managed to fit in six months, and for the motivation she showed in her work on the prototype we set in the pilot plant. Experimental results would not have been obtained without her help.

I am extremely grateful to Eyke Hüllermeier for his very detailed evaluation of my work, for the very interesting perspectives that arise from his report, and for a very stimulating discussion during the viva. I want to show my gratitude to Stéphane Canu as well, for accepting to evaluate my work, and for not bearing me a grudge for the fact that we missed only by inches the opportunity to work together during these three years. He, too, made very interesting remarks, which created interesting perspectives for future work.

I am obligated to all the members of my jury for accepting to evaluate my work during the viva. Thanks go again to Eyke, Stéphane, Thierry and Xia for this particular task, but I really want to make a special mention to Pr. Walter Schön, who presided over this jury with attention and kindness, and who showed great concern on and interest in my work and during my presentation.

The atmosphere in the local team, that is, the whole *département de Génie Informatique (GI)* and the *Ecole Doctorale*, –including secretaries, technicians, engineers, and researchers of all kinds–, made it nice to work with them on an everyday basis. A particular mention must be made here for those who helped me prepare my viva. Very special thanks go to all the PhD students for the solidarity that exists in the team, and for the smiles, kind words, coffee breaks, and for all their help that smoothed difficult times down a lot. Discussions with the team of Thierry's PhD students, my seniors, Benjamin and David, but also Frédéric, Amel and all the others, were of course particularly helpful.

There are no words to tell friends, in and out of Compiègne, University of Technology, what their support represented and still represents for me. Many thanks to:
Mélanie, Olivier, and Manon, for fantastically helpful discussions on the value of this

PhD thesis and what to make of it in the future.
All my friends, for their everyday support in all respects, and here, particularly in connection with this PhD project.
The PhD students of the *département GI* of Compiègne University for their technical support in the lab, but also for making life in Compiègne much nicer than it would have been without their contributions, together with those of other friends from Compiègne and Picardie.

Last but not least, I am extremely grateful to my family for their support, and for all the little things that made me be who and what I am, and allowed me to achieve all this.

Finally, to all these people, a single word, for anything you can say around it seems to hide all the other things that weren't and cannot be mentioned without making the list endless: Thank you very much, MERCI BEAUCOUP à vous tous !

Contents

Table of contents xi

Introduction 1

Belief functions 3

1 The TBM 5
1.1 Introduction . . . 9
1.2 Discrete Case . . . 9
1.2.1 Belief Representation . . . 9
1.2.2 Handling and Revision of Beliefs . . . 14
1.2.3 Decision Making . . . 23
1.3 Continuous Case . . . 24
1.3.1 Continuous Domain . . . 24
1.3.2 BF on $\mathbb{R}$ . . . 24
1.3.3 Belief Updating . . . 27
1.3.4 Ordering . . . 27
1.3.5 Decision Making . . . 27
1.4 Predictive BF . . . 30

2 From raw data to BF 33
2.1 Introduction . . . 37
2.2 Type I PBF . . . 37
2.2.1 Confidence Bands . . . 38
2.2.2 Discrete predictive belief function on a discrete domain . . . 41
2.2.3 Discrete predictive belief functions on $\mathbb{R}$ . . . 42
2.2.4 Continuous predictive belief functions on $\mathbb{R}$ . . . 46
2.2.5 Conclusion . . . 53
2.3 Type II PBF . . . 54
2.3.1 Consonant Belief Function Induced by a Set of Pignistic Probabilities 55
2.3.2 Application to a Sample of a Discrete Random Variable . . . 57
2.3.3 Construction of $\mathcal{P}$ . . . 58
2.3.4 Determination of the q-MCD Possibility Distribution . . . 59
2.3.5 Application to Continuous Parametric Models . . . 62
2.3.6 Exponential Distribution . . . 62
2.3.7 Normal Distribution . . . 64
2.3.8 Conclusion . . . 65
2.4 classification example . . . 66
2.5 Conclusion . . . 70

One-class classification 73

3 One-class classification 75
3.1 Introduction . . . 79
3.2 Desired properties . . . 79

3.2.1 Generalization ability . . . 79
3.2.2 Robustness . . . 80
3.2.3 Computational qualities . . . 80
3.3 Taxonomy . . . 80
3.3.1 Density based techniques . . . 81
3.3.2 Boundary based approaches . . . 81
3.3.3 Reconstruction approaches . . . 81
3.3.4 Clustering-based approaches . . . 82
3.3.5 Summary . . . 82
3.4 1-class classifiers . . . 82
3.4.1 Density based approaches . . . 82
3.4.2 Density based approaches . . . 84
3.4.3 Clustering-based approaches . . . 86
3.4.4 Reconstruction-based approaches . . . 87
3.4.5 Boundary-based approaches . . . 90
3.5 Conclusion . . . 97

4 From classifier to BF **101**
4.1 Introduction . . . 105
4.2 General approach . . . 105
4.3 Step 1 . . . 106
4.4 GBT solution . . . 110
4.5 The cognitive inequality . . . 110
4.5.1 Definition 1 . . . 110
4.5.2 Determining the LCBF satisfying a cognitive inequality of type I . 114
4.5.3 Definition 2 . . . 116
4.5.4 Determining the LCBF satisfying a cognitive inequality of type II . 117
4.6 CIneq-based solutions . . . 117
4.6.1 Model 1 . . . 117
4.6.2 Model 2 . . . 121
4.7 Discussion . . . 124
4.8 Examples . . . 126
4.8.1 Simple novelty detection: example 1 . . . 126
4.8.2 Simple novelty detection: example 2 . . . 128
4.8.3 Classifier Fusion Example . . . 128
4.9 Conclusion . . . 131

Monitoring of a waste incineration process **133**

5 Application **135**
5.1 Introduction . . . 139
5.2 The waste incineration process . . . 141
5.2.1 Waste combustion . . . 141
5.2.2 Energetic promotion . . . 142
5.2.3 Flue gas purification . . . 142
5.2.4 Pilot plant . . . 145
5.3 PMAT . . . 145

6 Implementation and results **147**
6.1 Implementation . . . 151
6.1.1 General structure of the PMAT . . . 151
6.1.2 Implementation of the PMAT . . . 153

6.1.3 Classifiers implemented in the PMAT . . . 156
6.2 Parameter tuning . . . 156
6.2.1 Case study examples . . . 156
6.2.2 SVM-based classification . . . 158
6.2.3 KPCA-based classification . . . 162
6.3 Results . . . 164
6.4 Conclusion . . . 178

Conclusion and Perspectives **179**

Appendices **191**

A Additional PMAT units **193**

B Intuitive justification of expressions (2.48), (2.57) and (2.58) **195**

C Proof of Proposition 7 **197**

Acronyms

bba	basic belief assignment
bbd	basic belief density
BF	Belief Function
cdf	cumulated density function
CI	Cheng and Iles' method
CIneq	Cognitive Inequality
EVT	Extreme Value Theory
FGCR	Flue Gas Leaning Residue
GBT	Generalized Bayes' Theorem
GMM	Gaussian Mixture Models
iff	if and only if
KDE	Kernel Density Estimation
KH	Kriegler and Held method
KNN or kNN	k Nearest Neighbours
(K)PCA	(Kernel) Principal Component Analysis
KRE	Kernel Reconstruction Error
LC	Least Committed
LCBF	Least Committed Belief Function
LCP	Least Commitment Principle
MCD	Most Committed Dominating
MVE	Minimum Volume Ellipsoid
NN	Neural Network
PBF	Predictive Belief Function
p-box	probability-box
PMAT	Process Monitoring Assistance Tool
r.v.	random variable
RSC	Residual Sodium Chemicals
SPE	Squared Prediction Error
SVM	Support Vector Machine
T^2	Hotelling's statistic
TBM	Transfereable Belief Model

Notations

A	arbitrary subset of Ω or $\mathcal{X}$
A^w	simple bba such that $m(A) = 1 - w$ and $m(\Omega) = w$
B	arbitrary subset of Ω or $\mathcal{X}$
b	implicability function
bel	belief function
$BetF$	cumulated pignistic density function
$Betf$	pignistic density function
$BetP$	pignistic probability
d	dimension of space
D_n	Kolmogorov's statistic
$d_{n,\alpha}$	fractil of D_n of order α
F	cumulated density function
$\overline{F}$	upper bound on F
$\underline{F}$	lower bound on F
f	density function
$\overline{f}$	upper bound on f
$\underline{f}$	lower bound on f
$\mathbf{G}$	generalization matrix
H	arbitrary subset of Ω or $\mathcal{X}$
H	separating hyperplane (SVM)
$\mathcal{H}$	transfer function (neural network)
$I_{[\alpha,\beta]}$	set of closed intervals included in $[\alpha, \beta]$
$\mathcal{K}$	kernel function
m	mass of belief
m^*	normalized mass of belief
$m_\emptyset$	mass function allocating a mass of 1 to $\emptyset$
$m_{1\textcircled{\cap}2}$	conjunctive combination of m_1 and m_2
$m_{1\textcircled{\cup}2}$	disjunctive combination of m_1 and m_2
${}^\alpha m$	discounted mass of belief
O	discrete variable taking values in Ω
o	value of O
p	number of retained principal components
P	probability distribution
$\mathbb{P}$	probability measure
Π	possibility distribution or measure
pl	plausibility function
q	commonality function
S	arbitrary subset of Ω or $\mathcal{X}$
$\mathscr{S}$	source of information
$\mathbf{S}$	specialization matrix
$\mathbf{S_m}$	Dempsterian specialization matrix based on m
T	(discrete or continuous) arbitrary variable taking value in $\mathcal{T}$
$\mathcal{T}$	domain of variable T
$\mathbb{T}_{[\alpha,\beta]}$	Triangular representation of $I_{[\alpha,\beta]}$
X	continuous variable taking values in $\mathcal{X}$
x	value of X

$\mathcal{X}$	continuous frame of discernement
α	discounting rate, or false negative rate
β	false positive rate
δ	Dirac delta function
Φ	mapping into a space in which $\mathcal{K}$ acts as a dot product
Ω	discrete frame of discernment
ω_i	element of Ω
$\downarrow$	projection (or conditioning)
$\Downarrow$	marginalization
$\uparrow$	ballooning extension
$\Uparrow$	vacuous extension
$\varnothing$	empty set
$\bigcirc\!\!\!\!\cap$	conjunctive combination
$\bigcirc\!\!\!\!\cup$	disjunctive combination
$\sqsubseteq_{pl}$	pl-more committed
$\sqsubseteq_{q}$	q-more committed
$\sqsubseteq_{s}$	s-more committed

List of tables

1.1 Binary representation of the subsets of a frame of discernement Ω 12
1.2 The conjunctive combination rule : the three suspects' saga. 17
1.3 The disjunctive and conjunctive combination rules : the three suspects' saga. 18
1.4 Pignistic probability associated with the disjunctive and conjunctive combination rules. 24
2.1 Pignistic probabilities and corresponding q-LC isopignistic possibility distributions of Example 10. 56
2.2 Calculation of q_{max} for the data of Example 10. 57
2.3 Goodman simultaneous confidence intervals for the data of Example 11, at confidence level $1 - \alpha = 0.90$. 58
2.4 Possibility distributions computed for the failure mode data of Example 12: 60
3.1 Pros and cons of the novelty detection techniques 98
3.2 Pros and cons of the novelty detection techniques 99
6.1 Variables to be monitored and associated process points 152
6.2 Values of α and β obtained on the test data set for $t_{pig} = 0.95$ and varying values of C and h. 160
6.3 Values of α and β obtained on the test data set for $t_{pig} = 0.9$ and varying values of C and h. 161
A.1 Variables to be monitored other than those mentioned in 6.1, associated process points, and action to be taken. 194

List of figures

1.1 Different types of bba . . . 11
1.2 Consonant and bayesian bba . . . 11
1.3 Relationship between confidence measures . . . 12
1.4 Marginalization and vacuous extension operations . . . 19
1.5 Conditionning and ballooning extension . . . 20
1.6 The three steps of the GBT . . . 22
1.7 Triangular representation of $\mathcal{I}_{[\alpha,\beta]}$. . . . 25
1.8 Domains of integration for bel, pl and q. . . . 26
1.9 q-LC possibility distribution induced by an exponential probability density $\mathcal{E}(\mu)$, for three different values of μ. . . . 29
1.10 q-LC possibility distribution induced by a normal probability density $\mathcal{N}(\mu, \sigma^2)$ for $\mu = 0$ and three different values of σ. . . . 30
2.1 Sample cdf S_n and Kolmogorov confidence band at level $1 - \alpha = 0.95$ for the bearings data. . . . 40
2.2 Continuous confidence band and cumulative density function estimated through Cheng and Iles' algorithm. . . . 41
2.3 Principle of the construction of a basic belief assignment from a p-box. . . . 44
2.4 Focals intervals of the PBF constructed from the Kolmogorov confidence band at level $1 - \alpha = 0.95$ (bearings data). . . . 45
2.5 Plausibility profile function (left) and pignistic probability density function (right) of the discrete PBF constructed from the Kolmogorov confidence band . . . 46
2.6 Contour plots of functions $bel(\cdot;\mathbf{X})([x,y])$, $pl(\cdot;\mathbf{X})([x,y])$ and $q(\cdot;\mathbf{X})([x,y])$ constructed from Kolomogorov's confidence band . . . 47
2.7 Plausibility profile function obtained from the continuous confidence band of Figure 2.2. . . . 52
2.8 Contour plots of functions $bel([x,y];\mathbf{X})$, $pl([x,y];\mathbf{X})$ and $q([x,y];\mathbf{X})$ constructed from Cheng and Iles' confidence band. . . . 52
2.9 Definition of the q-most committed dominating (q-MCD) bba m^* associated with a set $\mathcal{P}$ of probability distribution. . . . 56
2.10 Illustration of the approach introduced in [42]: . . . 57
2.11 Plot of $\pi^*(x)$ for the exponential distribution with $\overline{x} = 1$, $\alpha = 0.1$, and $n = 10, 30, 100$ and ∞. . . . 63
2.12 Shape of Mood's exact region: . . . 64
2.13 Plot of $\pi^*(x)$ for the normal distribution with $\overline{x} = 0$, $s^2 = 1$, $\alpha = 0.1$, $\alpha_1 = \alpha_2$, and $n = 10, 30, 100$ and ∞. . . . 66
2.14 (a): Plot of $pl(x|\omega_1)$ (solid lines) and $pl(x|\omega_2)$ (dashed lines) . . . 68
2.15 (a): Plot of $pl(y|\omega_1)$ (solid lines) and $pl(y|\omega_2)$ (dashed lines) . . . 69
2.16 Box plots of error rates for the LC, MCD and CI methods as well as sensor S_x alone . . . 71
3.1 Influence of the kernel on KDE . . . 83
3.2 A formal neuron . . . 88
3.3 Outlier detection through balloon plot . . . 93
3.4 Outlier detection through nested convex hull volume change . . . 94
3.5 SVM-based one-class classification . . . 95
3.6 Separating hyperplanes H_{+1}, H_{-1} and H for a toy example . . . 96
4.1 The ring data set. . . . 106

4.2 Contour lines of the SVM novelty measure $T = -f(x)$ for the ring data set. 107
4.3 Plausibility function obtained via Kriegler and Held's algorithm (ring data). 108
4.4 Contour-lines of the plausibility function obtained by the KH method, with respect to the position of the data (ring data). 108
4.5 Plausibility function as obtained via the CI method (ring data). 109
4.6 Contour-lines of the plausibility function obtained by the CI method, with respect to the position of the data (ring data). 109
4.7 Bba on Ω knowing $T = t_*$, GBT solution, discrete case (ring data), for pl_0 obtained via Kriegler and Held's algorithm. 111
4.8 Contour-lines of $m^{\Omega}(\omega_1)$ knowing $T = t_*$, GBT solution, discrete case (ring data), for pl_0 obtained via the KH method. 112
4.9 Bba on Ω knowing $T = t_*$, GBT solution, continuous case (ring data) for pl_0 obtained via the CI method. 112
4.10 Contour-lines of $m^{\Omega}(\omega_1)$ knowing $T = t_*$, GBT solution, (ring data), for pl_0 obtained via the CI method. 113
4.11 Representation of the integration area for the plausibility function 115
4.12 Bba on Ω knowing $T = t_*$, Model 1, discrete case (ring data), for pl_0 obtained via Kriegler and Held's algorithm. 119
4.13 Contour-lines of $m^{\Omega}(\omega_1)$ knowing $T = t_*$, Model 1, discrete case (ring data), for pl_0 obtained via Kriegler and Held's algorithm. 120
4.14 Bba on Ω knowing $T = t_*$, Model 1, continuous case (ring data), for pl_0 obtained via Cheng and Iles' confidence-band. 121
4.15 Contour-lines of $m^{\Omega}(\omega_1)$, knowing $T = t_*$, Model 1, discrete case (ring data), for pl_0 obtained via the CI method. 122
4.16 Bba on Ω knowing $T = t_*$, Model 2, continuous case, for pl_0 obtained via Cheng and Iles' confidence-band (ring data). 125
4.17 Contour-lines of $m^{\Omega}(\omega_1)$, knowing $T = t_*$, Model 2, discrete case (ring data), for pl_0 obtained via the CI method. 125
4.18 Kolmogorov confidence band around the distribution of the value of T for a KPCA-based classifier (Breast-cancer data). 126
4.19 Predictive belief function on the value of T for a KPCA-based classifier (Breast-cancer data). 127
4.20 Pignistic probability function on ω_1 for a KPCA-based classifier (Breast-cancer data). 127
4.21 Kolmogorov confidence band around the distribution of the value of T for a SVM-based classifier (Combustion data). 128
4.22 Predictive belief function on the value of T for a SVM-based classifier (Combustion data). 129
4.23 Pignistic probability function on ω_1 for a SVM-based classifier (Combustion data). 129
4.24 Test estimates of the ROC curves for the three classifiers 130
4.25 Zoom on the top left hand corner of Figure 4.24. 131
5.1 Screen shot of one of the monitoring screen 140
5.2 Global balance sheet of a waste incineration process. 142
5.3 The four oven combustion zones. 143
5.4 Fumes treatment route 144
6.1 Physical structure of the PMAT (general organization) 154
6.2 A single unit of the PMAT 155
6.3 Matlab default simulation 158
6.4 $BetP(\omega_1) = g(-f)$ 162
6.5 Parameter selections 163
6.6 Sensor breakdown detection, combustion data (Example 22, Section 6.2.1) 166

6.7 Sensor breakdown detection, fumes data (Example 23, Section 6.2.1) . . . 167
6.8 Sensor decalibration detection, combustion data (Example 22, Section 6.2.1). 168
6.9 Sensor decalibration detection, fumes data (Example 23, Section 6.2.1). . . 169
6.10 Sensor decalibration detection, steam data. 170
6.11 Slow drift detection, combustion data (Example 22, Section 6.2.1). 171
6.12 Slow drift detection, fumes data (Example 23, Section 6.2.1). 172
6.13 Slow drift detection, steam data. 173
6.14 Detection of a drift of the upper chamber sensor No.2. 174
6.15 Detection of a 10 degree drop of the flue gas temperature. 174
6.16 Attempts in smoothing the pignistic probability of fault 176
B.1 Calculation of the least committed bel associated with a continuous confidence band 196
C.1 Representation of $pl_0([t-dt;+\infty))$ and $pl_0([t;+\infty))$ 197

Introduction

Suez Environnement is one of the leading companies of the Water Distribution, Waste Water Treatment and Waste Promotion Industry in France. Via some internal audit, the group's Waste Promotion Technical Direction noticed that the waste incineration process monitoring was most of the time not complying with the simplest rules. For example, it quite often happened that the percentage of oxygen in the oven was maintained around 5%, whereas good combustion requires a minimum of 6% O_2. Further analysis showed that the value of some key measures was actually slowly drifting, so slowly that no-one noticed the change, and, after some time, habit made the wrongest things sound normal to the operators. The result of the assessment was alarming.

The control and command desk most of the time displayed many data (up to five thousands!), but no analysis or synthesis of these measurements was shown on screen. No reference data were shown either, apart from the fact that an alarm was set on some of the variables in case they went beyond some upper or lower threshold. A posteriori analysis was performed on the basis of weekly reports, allowing the detection of some malfunctioning. However, most of these problems were detected too late, and sometimes lead to further problems that may have been avoided, had their origin been detected earlier.

Hence, the Technical Direction decided to take some action on the issue. An attempt in building a thermodynamic model of the installations failed in predicting the outcomes, due to the uncertainty and imprecision of the involved data. It was thus decided to submit the case to the group's research department, the *Centre International de Recherche sur l'Eau et l'Environnement* (International Centre of Research on Water and Environnement). They, in turn, submitted the problem to Thierry Denœux, professor at the *Université de Technologie de Compiègne* (Compiègne University of Technology). It was agreed that the problem would be the subject of an MSc thesis, which would lead to a PhD thesis if necessary. It is in this context that the present work was undertaken.

The project aims at the production of a tool establishing a permanent reference on the process key points, and continuously evaluating the process performances, in terms of safety, productivity, and standards compliance. The problem may thus be seen as a classification task, in which the normal, –or desired– state of the system needs to be differentiated from the different types of faults. Nevertheless, the latter are too numerous and varied to allow the building of a training set of data, while the normal state of the process is very well represented in a waste incineration plant database. Therefore, the problem turns out to be a one-class classification task. Moreover, as already mentioned, the problem involves a lot of uncertainty.

The objectives of the project were thus specified as follows. The theoretical problem would be solved in such a way that the uncertainty attached to classification decisions would be available to the end user, and a functional prototype would be delivered at the end. It was however decided that the PhD project would only be concerned with the regulation of the oven-and-boiler subunit, to allow the study to fit in the allocated time. This does not constitute an important restriction. The roles of the oven and boiler are indeed determinant for the rest of the installation, and keeping this part of the process under control ensures the stability of the whole system.

Because of its ability to handle imprecision and uncertainties, the belief function theory, and more particularly Smet's Transferable Belief Model [126, 120], seemed to be an ideal framework in which to solve the problem. A feasibility study showed that the data have particularly complex correlations. This study concluded that models underlying

the hypothesis of linear correlation, such as standard Principal Component Analysis, widely used as a basis for process monitoring in the chemical industry, would not work efficiently. It was thus decided that kernel-based models, which act as though the data had undergone a prior transformation that makes them linearly correlated, would be better adapted.

From a theoretical point of view, this constitutes an interesting problem. In effect, the multi-class classification problem, for which data of several classes are available for training, have already been tackled in different ways in the Transferable Belief Model [38, 37, 36, 56, 138]. On the other hand, the one-class or novelty detection task had not yet been studied in this framework. Let us consider a system that can only be in two possible situations: the reference state ω_0, or a situation ω_1 including all other possible states. The problem under consideration is the assessment of the hypothesis that the system is in state ω_0 when the only available information about the system is a sample of observations $\mathbf{x_1}, \ldots, \mathbf{x_n}$ of some variables, representative of the system state, conditioned on ω_0. It underlies two different subproblems, which will be studied in this PhD thesis.

The first subproblem is to express the available knowledge in terms of belief functions. First, a novelty measure T, whose value will be small in the region of space containing the data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and larger as the distance to this region increases, needs to be built as a function of $\mathbf{x}_1, \ldots, \mathbf{x}_n$. This may be done using a one-class classifier. Then, a methodology allowing us to express our belief about the realization of a new sample drawn from the same distribution as T needs to be established, i.e., we need to be able to predict what the next observation of T will be. We suggest two possible approaches to this problem. The first approach is based on Hacking's frequency principle [55, 117], which equates the degree of belief of an event to its probability, when the latter is known. The second method is based on a weak form of Hacking's principle, which states that the pignistic probability of an event should be equal to its long run frequency, when the latter is known.

As we only dispose of data collected when the system was in a given state, the obtained belief function expresses our belief in future values of T, knowing the system is in this particular state. From this, we need to infer our belief in the present system state, knowing the value of the studied statistic. This constitutes the second subproblem. We propose three different solutions: the first is based on the GBT, and the other two are based on the notion of cognitive inequality, introduced in this report, and which may be seen as the belief function theory equivalent of the stochastic inequality.

This report is structured in three main parts, divided in six chapters. The first part of the work consisted in solving the problem of constructing a belief function from raw data, and will be presented in Part I. The basics of the transferable belief model will first be introduced in Chapter 1, and different methodologies for the building of belief functions from data will be detailed in Chapter 2. Then, the issue of novelty detection with belief functions had to be tackled, and this is what Part II is about. Chapter 3 is a review of existing one-class classification techniques, and the new solution developed in the TBM framework is presented in Chapter 4. Finally, a prototype had to be developed. It is depicted in Part III. The adopted solution is described in Chapter 5, and results are shown in Chapter 6. General conclusion and perspectives conclude the report.

Part I

Belief functions

The Transferable Belief Model (TBM)

Contents

1.1 Introduction **9**
1.2 Discrete Case **9**
1.2.1 Belief Representation 9
1.2.2 Handling and Revision of Beliefs 14
1.2.3 Decision Making 23
1.3 Continuous Case **24**
1.3.1 Continuous Domain 24
1.3.2 BF on $\mathbb{R}$ 24
1.3.3 Belief Updating 27
1.3.4 Ordering 27
1.3.5 Decision Making 27
1.4 Predictive BF **30**

Summary

In this chapter, the main notions pertaining to the belief function theory, –and more particularly to Smet's Transferable Belief Model– will be introduced. We will start with the case of discrete belief functions. Basic definitions will first be recalled. Mechanisms for the handling and updating of beliefs will then be described. Lastly, the decision making process will be explained.
After that, the belief functions on $\mathbb{R}$ will be presented. The distinction between discrete and continuous belief functions on the real line will be established. Then, the mechanisms for the handling and revision of belief functions on $\mathbb{R}$ will be clarified. A partial order on belief functions will be introduced, and the decision making process will be described for belief functions on $\mathbb{R}$.

Résumé

Dans ce chapitre, les notions principales de la théorie des fonctions de croyance, –et, en particulier, du modèle des croyances transférables de Smets–, sont abordées. Tout d'abord, le cas des fonctions de croyance discrètes est présenté. Les définitions de bases sont rappelées, et les mécanismes de modification et mise à jour des fonctions de croyances sont décrits en détails. Finalement, le procédé de prise de décision est introduit.
Dans un second temps, les fonctions de croyance sur $\mathbb{R}$ sont présentées. La distinction entre fonctions de croyance discrètes et continues dans $\mathbb{R}$ est établie. Les mécanismes de manipulation et de mise à jour des fonctions de croyance sur $\mathbb{R}$ sont clarifiés. Un ordre partiel sur les fonctions de croyance est introduit, et le procédé de prise de décision est décrit pour les fonctions de croyance sur $\mathbb{R}$.

1.1 Introduction

The necessity of handling imprecisions and uncertainties of data lead to the development of several mathematical theories, such as the theory of possibilities, the imprecise probability theory, and the belief function theory. The latter, sometimes also called evidence theory or Dempster-Shafer theory, was introduced by Dempster in the late 1960's [30, 31, 32, 33, 34, 35], and further developed by Shafer in [113] (1976). Several interpretations of this framework have been introduced since then, amongst which the Transferable Belief Model (TBM) and Kohlas' hints theory [65].

The TBM, a subjectivist interpretation of the evidence theory introduced by Smets [126, 120], establishes an interesting framework for the resolution of problems of diagnostic [119], pattern recognition [38, 37, 36, 56, 138] and information fusion [48, 85]. It is a two level model. Information is handled at the credal level, where beliefs can be represented, updated and combined. Decisions are made at the pignistic level, from Latin *"pignus"*, to bet.

In this chapter, the main notions pertaining to the TBM will be introduced. The concepts that will be used throughout this thesis will be especially emphasized.

1.2 Discrete Case

1.2.1 Belief Representation

Basic belief assignment and equivalent functions

Considering a given question (variable) O and a set of possible answers (values) Ω, termed *frame of discernment*, we would like to model the belief of an agent Ag in the fact that the answer to question O is in Ω. However, the agent may not be able to distinguish amongst single answers, and may need to allocate part of his belief to arbitrary subsets of Ω. A formal definition of this is given in the TBM.

Definition 1. (Basic belief assignment) *Let $\Omega = \{\omega_1, \ldots, \omega_K\}$ be a finite set, and let O be a variable taking values in Ω. Given some evidential corpus EC, the knowledge held by a given agent Ag at a given time over the actual value of variable O can be modeled by a so-called* basic belief assignment *(bba) m^Ω defined as a mapping from 2^Ω into $[0,1]$ such that:*

$$\sum_{A \subseteq \Omega} m^\Omega(A) = 1. \tag{1.1}$$

Each mass $m^\Omega(A)$ is interpreted as the part of the agent's belief allocated to the hypothesis that O takes some value in A [113, 126]. The subsets $A \in \Omega$ such that $m(A) > 0$ are called *focal sets* of m.

Where there is no ambiguity, m^Ω will be shortened m.

There exists a number of equivalent representation of m, including the belief, plausibility, commonality, and implicability functions defined, respectively, as:

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B), \tag{1.2}$$

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \tag{1.3}$$

$$q(A) = \sum_{B \supseteq A} m(B), \tag{1.4}$$

and

$$b(A) = \sum_{B \subseteq A} m(B), \tag{1.5}$$

for all $A \subseteq \Omega$.

Interpretation: The quantity $bel(A)$ (*belief* in A) can be interpreted as the degree of belief that can be allocated to a subset A, and is strictly ascertained by the available evidence.

The *plausibility* of A, on the other hand, measures the quantity of information that is not in contradiction with A. In other words, it is the maximum degree of belief that could be allocated to A upon the collection of additional information.

The *commonality* of A is the sum of masses allocated to supersets of A, and may be interpreted as the a measure of ignorance when the answer to question O is known to belong to A [118].

Finally, the *implicability* of A is the sum of masses allocated to subsets of A.

Each of these functions is in one to one correspondence with the others. The belief and plausibility functions are the most commonly used in the formalization of problems, as their interpretation is easy. The implicability and commonality functions play an important role in calculations, and often make the mathematics of the TBM simpler. By misuse of language, any of them may sometimes be designated by the term "belief function".

Special cases

A belief function is said to be *normal* if the empty set is not a focal set, *subnormal* if it is. Only normal belief functions are considered in Dempster's and Shafer's work, but the TBM allows subnormal belief functions as well. The mass assigned to the empty set may actually play an important part, and can be seen as the mass of belief granted to the hypothesis that the truth does not lie in Ω, hence carrying the idea that the chosen model might not fit reality with enough precision.

However, as normality is imposed in many interpretations of the evidence theory – and especially in Dempster's work, from which the TBM directly follows–, it is important that the normalization operation be defined here:

Definition 2. (Normalization) *Let m be a subnormal bba. Normalization transforms it into a normal bba m^* defined as:*

$$\begin{aligned} m^*(A) &= \frac{m(A)}{1-m(\emptyset)}, \ \forall A \neq \emptyset, \\ m^*(\emptyset) &= 0. \end{aligned} \tag{1.6}$$

Through this transformation, the value of $m(\emptyset)$ is spread onto the other subsets of Ω, leading to a -sometimes important- loss of information.

The mass allocated to Ω, i.e., the part of belief shared between all possible answers, represents ignorance. Consequently, if the entire mass of belief is assigned to Ω, the corresponding bba is said to be *vacuous*. Conversely, when Ω is not a focal set, the mass function is termed *categorical*, as it leaves no place for ignorance.

Finally, a bba is called *simple* if it has at most two focal sets, including Ω, and *categorical* if it is simple and dogmatic, i.e., the entire mass is set onto one focal set only and that focal set is not Ω, reflecting the fact that the agent has no doubt where the truth lies.

The different types of bba are represented in Figure 1.1.

Link to other theories

When the focal sets are nested, m is said to be *consonant*, and the associated plausibility function is a possibility measure: it verifies $pl(A \cup B) = \max(pl(A), pl(B))$ for all $A, B \subseteq \Omega$. The corresponding belief function is the dual necessity measure.

When the focal sets are all singletons, m is termed *Bayesian* and $pl = bel = P$ is a probability distribution. A Bayesian belief function is maximally precise.

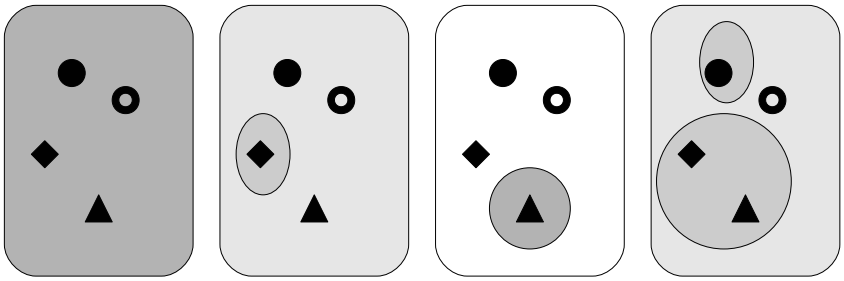


Figure 1.1: Different types of bba
Different types of bba: from (a) to (d): vacuous, simple, categorical, arbitrary bba

Representations of a consonant and a Bayesian bba are given in Figure 1.2. Shaded areas indicate a non empty mass on the covered elements while white indicates a mass $m = 0$.

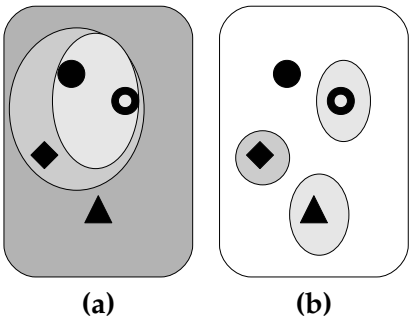


Figure 1.2: Consonant and bayesian bba
Different types of bba: (a) consonant bba, (f): bayesian bba

From the above, it is easy to see that belief function theory includes both possibility and probability theories. The relationship between the different confidence measures is better represented on a graph (see Figure 1.3).

Matrix representation

A mass function may be represented by a vector $\mathbf{m}$ of length $2^{|\Omega|}$ containing, in a predefined order, the value of the mass assigned to each subset of Ω. Any order could be used, but one particular order proved extremely efficient in compacting mathematical expressions, and that is the so-called *binary order*.

Write the elements of Ω on a unique line L in no particular order, and then represent each subset A of Ω by a binary number composed of ones on the positions corresponding to the positions on L of the elements of Ω that are included in A, and zeros elsewhere. Sort the obtained binary numbers increasingly, and you will get a binary ordering of the subsets of Ω. This means that, if Ω is e.g. $\{a, b, c\}$, then the empty set ($\emptyset$) is coded by '000', $\{a, b\}$ will be denoted by '011' and $\{a, c\}$ by '101'.

The i^{th} element of vector $\mathbf{m}$ will indifferently be denoted $\mathbf{m}_i$ or $\mathbf{m}(A)$, where A equals, e.g. $\{a, c\}$ if $i = 6$. Note that the binary code of A is $i - 1$. Functions *bel*, *pl*, *q* and *b* may similarly be represented by vectors, respectively denoted **bel**, **pl**, **q**, and **b**. (see Table 1.1).

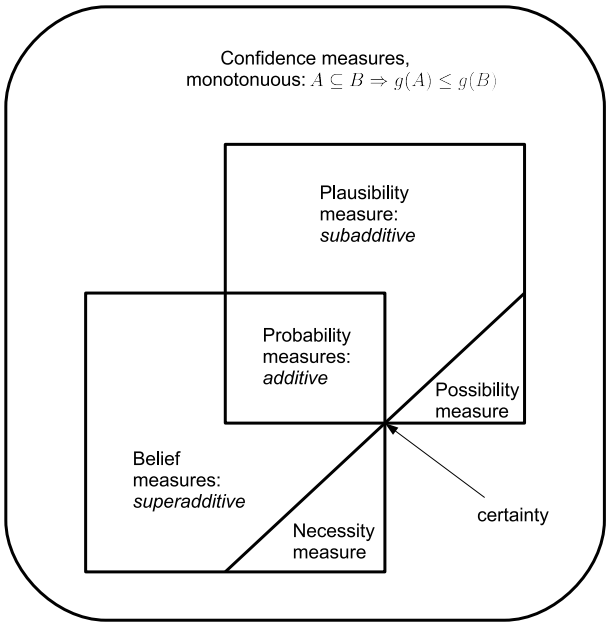


Figure 1.3: Relationship between confidence measures

Vector line no. (decimal)	Binary code of set A	set A	$\mathbf{m}(A)$	$\mathbf{f}(A)$ *
1	000	$\emptyset$	$\mathbf{m}(\emptyset)$	$\mathbf{f}(\emptyset)$
2	001	$\{a\}$	$\mathbf{m}(\{a\})$	$\mathbf{f}(\{a\})$
3	010	$\{b\}$	$\mathbf{m}(\{b\})$	$\mathbf{f}(\{b\})$
4	011	$\{a,b\}$	$\mathbf{m}(\{a,b\})$	$\mathbf{f}(\{a,b\})$
5	100	$\{c\}$	$\mathbf{m}(\{c\})$	$\mathbf{f}(\{c\})$
6	101	$\{a,c\}$	$\mathbf{m}(\{a,c\})$	$\mathbf{f}(\{a,c\})$
7	110	$\{b,c\}$	$\mathbf{m}(\{b,c\})$	$\mathbf{f}(\{b,c\})$
8	111	$\Omega = \{a,b,c\}$	$\mathbf{m}(\Omega)$	$\mathbf{f}(\Omega)$

Table 1.1: Binary representation of the subsets of a frame of discernement Ω and associated matrix representation of a belief function on Ω.*(* **f** may be one of* **bel**, **pl**, **q**, **b**.*)*

The Least Commitment Principle (LCP)

The LCP plays a role similar to the principle of maximum entropy in Bayesian Probability Theory.

Definition 3. (The Least Commitment Principle (LCP)) *It dictates that, in a set of belief function compatible with the available information, the least informative should always be chosen.*

This principle reflects a cautious attitude. It conveys the idea that no more credit should ever be given to an hypothesis than is strictly accounted for by available evidence, nor should any hypothesis be ruled out without sufficient information.

In order to allow the possibility to pick up the least informative BF, a partial order should be defined.

Ordering

It is sometimes important to be able to compare the precision of two different belief functions. Several measures have been introduced in order to quantify the precision of an arbitrary belief function, the quantity of information it carries, or its degree of uncertainty. Both ordinal and quantitative methods can be used; however, we will restrict ourselves to ordinal approaches. The reader is referred to [115, 64, 92] for a description of quantitative approaches.

Several partial orderings, generalizing set inclusion, have been proposed for the comparison of belief functions [134, 44]. The pl-, q- and s- orderings will be defined in the sequel.

It is obvious to see that the least precise of all belief functions is the vacuous bba. On the contrary, the most precise belief function would be a categorical bba focusing on a singleton. The intuitive deductions are that,

- the more precise a belief function, the smaller the subsets of Ω its focal sets are;
- there are several, (possibly maximally) equally precise belief functions for a given frame of discernment Ω.

Let A and B be two non empty subsets of Ω such that $A \subset B$. Let m_A and m_B be to categorical bba such that $m_A(A) = 1$ and $m_B(B) = 1$, and let pl_A and pl_B be the corresponding plausibility functions. They are defined as

$$pl_A(C) = \begin{cases} 1 & \text{if } C \cap A \neq \emptyset, \\ 0 & \textit{otherwise}; \end{cases} \tag{1.7}$$

and

$$pl_B(C) = \begin{cases} 1 & \text{if } C \cap B \neq \emptyset, \\ 0 & \textit{otherwise}. \end{cases} \tag{1.8}$$

Now, as $A \subset B$, $C \cap A \neq \emptyset \Rightarrow C \cap B \neq \emptyset$. Consequently, $pl_A(C) \leq pl_B(C)$, $\forall C \subseteq \Omega$, and m_A may be said to be *pl-more informative* than m_B. This will be denoted

$$m_A \sqsubseteq_{pl} m_B. \tag{1.9}$$

This leads to the following general definition of the pl-ordering for ordinary belief functions. Let pl_1 and pl_2 be two plausibility functions on Ω.

Definition 4. (pl-ordering) *pl_2^Ω is pl-less committed than pl_1^Ω iff:*

$$pl_1^\Omega(A) \leq pl_2^\Omega(A), \quad \forall A \subseteq \Omega. \tag{1.10}$$

Moreover, the same two bba m_A and m_B also yield:

$$q_A(C) = \begin{cases} 1 & \text{if } A \supseteq C, \\ 0 & \textit{otherwise}; \end{cases} \tag{1.11}$$

and

$$q_B(C) = \begin{cases} 1 & \text{if } B \supseteq C, \\ 0 & \textit{otherwise}. \end{cases} \tag{1.12}$$

As $A \supseteq C$ implies $B \supseteq C$, $\forall C \subseteq \Omega$, then $q_A(C) \leq q_B(C)$, $\forall C \subseteq \Omega$, and m_A may be said to be *q-more informative* than m_B, denoted $m_A \sqsubseteq_q m_B$.

The general definition of the q-ordering is:

Definition 5. (q-ordering) *q_2^Ω is q-less committed than q_1^Ω iff:*

$$q_1^\Omega(A) \leq q_2^\Omega(A), \quad \forall A \subseteq \Omega \tag{1.13}$$

However, the pl- and q- partial orderings are not equivalent in general, and cannot be compared (neither of them implies the other). Nevertheless, these two orderings are equivalent in the special case of consonant belief functions: if m_1 and m_2 are consonant, then

$$m_1 \sqsubseteq_q m_2 \Leftrightarrow m_1 \sqsubseteq_{pl} m_2 \Leftrightarrow pl_1(\{\omega\}) \leq pl_2(\{\omega\}), \quad \forall \omega \in \Omega. \tag{1.14}$$

A stronger partial order may be defined through the notion of specialization that will be defined in the next section (Section 1.2.2). If m_1 is a specialization of m_2, then it is more informative: in effect, we will see that it means that m_1 may be obtained from m_2 by transferring the masses $m_2(C)$ onto subsets of C, for all $C \subseteq \Omega$. This order is termed s-ordering, and implies both the q- and pl-orderings:

$$m_1 \sqsubseteq_s m_2 \Rightarrow \begin{cases} m_1 \sqsubseteq_{pl} m_2 \\ m_1 \sqsubseteq_q m_2 \end{cases}. \tag{1.15}$$

Properties

$$\begin{array}{l} \forall m, m' \\ \qquad \begin{array}{lll} m \textcircled{\cap} m' & \sqsubseteq_{pl} & m \textcircled{\cup} m', \\ m \textcircled{\cap} m' & \sqsubseteq_q & m \textcircled{\cup} m', \\ m \textcircled{\cap} m' & \sqsubseteq_s & m \textcircled{\cup} m', \end{array} \end{array} \tag{1.16}$$

and

$$\begin{array}{lll} m_\emptyset & \sqsubseteq_{pl} & m_\Omega, \\ m_\emptyset & \sqsubseteq_q & m_\Omega, \\ m_\emptyset & \sqsubseteq_s & m_\Omega, \end{array} \tag{1.17}$$

where $m_\emptyset$ denotes a bba of maximal conflict ($m_\emptyset(\emptyset) = 1$) and m_Ω is the vacuous belief function.

The interpretation of these and other ordering relations is discussed in [44] from a set-theoretical perspective, and in [46] from the point of view of the TBM.

1.2.2 Handling and Revision of Beliefs

Specialization and Generalization

We will see that most operations for the handling and revision of beliefs may be simply expressed in terms of mass transfer from one subset of Ω to another, hence the name *Transferable Belief Model*. In fact, these operations all derive from two types of operations, namely specialization and generalization.

The operation of specialization consists in transferring the mass allocated to each focal element A onto a series of subsets B of A. It constitutes a refinement of the available information (the belief allocated to a subset $A \subset \Omega$ is transferred onto subsets of A) or an addition of new information (part of the belief assigned to Ω may be transferred onto subsets of Ω, thus reducing the part of ignorance).

Definition 6. (Specialization) *A mass function* $\mathbf{m}_2$ *is a specialization of* $\mathbf{m}_1$ *if there exists a stochastic matrix* S *of dimensions* $2^{|\Omega|} \times 2^{|\Omega|}$ *such that* $\mathbf{S}(A, B) = 0$ *for all* $A \not\subseteq B$*, and*

$$\mathbf{m}_2 = \mathbf{S} \cdot \mathbf{m}_1, \tag{1.18}$$

or equivalently,

$$\mathbf{m}_2(A) = \sum_{B \subseteq \Omega} \mathbf{S}(A, B)\mathbf{m}_1(B). \tag{1.19}$$

$\mathbf{S}$ is an upper-triangular matrix whose element $\mathbf{S}(A, B)$ represents the part of the mass $\mathbf{m}_1(B)$ that will be transferred onto $A \subseteq B$.

The opposite of a specialization is a generalization. The operation of generalization consists in transferring the mass assigned to each subset B of Ω onto a series of subsets A of Ω that include B.

Definition 7. (Generalization) $\mathbf{m}_2$ *is a generalization of* $\mathbf{m}_1$ *if there exist a stochastic matrix* $\mathbf{G}$ *of dimensions* $2^{|\Omega|} \times 2^{|\Omega|}$ *such that* $\mathbf{G}(A, B) = 0$ *for all* $B \not\subseteq A$*, and*

$$\mathbf{m}_2 = \mathbf{G} \cdot \mathbf{m}_1, \tag{1.20}$$

$\mathbf{G}$ is a lower-triangular matrix whose element $\mathbf{G}(A, B)$ represents the part of the mass $\mathbf{m}_1(B)$ which is transferred onto $A \supseteq B$.

Conditioning and ballooning extension When a given hypothesis $H \subseteq \Omega$ is ascertained, the beliefs are altered to reflect the new state of knowledge. Masses associated to subsets B, $B \subseteq \Omega$, are transferred onto subsets $H \cap B$.

Definition 8. (Conditioning) *Consequently, the mass of belief on* Ω *conditioned to* H *is:*

$$m^{\Omega}[H](A) = \sum_{B \cap H = A} m(B), \quad \forall A \subseteq \Omega \tag{1.21}$$

The dual operation is termed *ballooning extention*.

Definition 9. (Ballooning extention) *Let* $m^{\Omega}[H]$ *be the bba on* Ω *conditioned with respect to* $H \subseteq \Omega$*. Assume we now learn* H *finally does not necessarily hold and all previous states of knowledge have been lost. Masses associated with any non-empty set* A *of* Ω *are then transferred onto* $B = A \cup \overline{H}$*:*

$$m[H]^{\Uparrow\Omega}(B) = \begin{cases} m[H](A) \text{ if } B = A \cup \overline{H} \\ 0 \text{ otherwise.} \end{cases} \tag{1.22}$$

The ballooning extension operation only allows finding the least committed bba whose conditioning on H will lead back to $m^{\Omega}[H]$. Hence, some information might be lost in the process of successive conditioning and "deconditioning" operations, and the original bba m cannot always be recovered.

Remark 1. *The conditioning operation is a particular form of specialization, and the ballooning extension is a form of generalization.*

Remark 2. *Let* m *be a bba on* Ω*. A specialization matrix* $\mathbf{S}_m$ *whose elements are defined as follows:*

$$\mathbf{S}_m(A, B) = m[B](A), \quad \forall A, B \subseteq \Omega \tag{1.23}$$

is termed Dempsterian specialization matrix associated with m*.*

Combination

In order to decide how to combine two pieces of information, it is important to know whether they were induced by common evidence or not. In effect, if two sources of information come to the same conclusion from different pieces of evidence, then the two results should reinforce one another. However, if they use the same evidence, then this two conclusions should not reinforce each other. Two such pieces of information are said not to be *distinct*.

Conjunctive and Disjunctive combination The most common combination rules are the conjunctive and disjunctive combination rules. Both rules require the two sources of information to be distinct.

Given two distinct pieces of evidence m_1 and m_2, given by two different sources, the conjunctive combination $m_{1\bigcirc\!\!\!\cap 2}$ of m_1 and m_2 can be defined as follows:

Definition 10.

$$m_{1\bigcirc\!\!\!\cap 2}(A) = \sum_{B\cap C=A} m_1(B)m_2(C), \quad \forall A \subseteq \Omega. \tag{1.24}$$

This operation corresponds to a very intuitive rule: if two different, equally reliable, witnesses provide two different testimonies that do not entirely contradict each other, then the natural way to built a conclusion is to cross-check the two declarations and to keep only the hypotheses that comply with both testimonies, i.e., it corresponds to a logical "AND". It should be used when all sources of information are known to be reliable.

Remark 3. *The conjunctive combination of a mass m_2 with a mass m_1 is a Dempsterian specialization. It generalizes the conditioning operation.*

Remark 4. *The mass allocated to the empty set by the conjunctive combination of two (normal) bbas may be seen as the degree of conflict (or contradiction) between the two sources of informations the bbas originated from.*

Definition 11. Dempster's rule of combination *is defined as the conjunctive combination of two normal bba followed by normalization.*

Example 1. (The murderer) *Consider a variant of the Peter, Paul and Mary saga introduced by Smets [120]. The outline of the saga reads as follows.* "Big Boss has decided that Mr. Jones must be murdered by one of the three people present in his waiting room and whose names are Peter, Paul and Mary." *Suppose now that you know nothing about the way the killer was selected, that Mary and Peter smoke but Paul does not, and that two persons witnessed the murder through the window, but did not had time to intervene. Witness-1 says the murderer was smoking. Witness-2 says the killer was a woman. If you rely equally on both of them, you will conclude the murderer is Peter.*

What happens if none of the witnesses is 100% certain of what he/she said ? *Suppose now that Witness-1 (denoted W1) is 80% sure the murderer was smoking, and Witness-2 (W2) is only 50% sure the killer was a woman. The belief functions associated with each witness are represented in table 1.2, together with their conjunctive combination.*

Property 1. *The conjunctive combination rule is associative and commutative; its neutral element is the vacuous belief function.*

Property 2.

$$q_{1\bigcirc\!\!\!\cap 2}(A) = q_1(A)q_2(A), \quad \forall A \subseteq \Omega \tag{1.25}$$

Now, recall that the application of the conjunctive combination rule requires the two sources of information to be distinct. In [121], Smets attempted to clarify this notion.

Suspect	m_1	m_2	$m_{1\textcircled{\cap}2}$
$\emptyset^*$	0	0	0
Peter	0	0	0
Paul	0	0	0
Peter or Paul	0	0	0
Mary	0	0.5	0.5
Peter or Mary	0.8	0	0.4
Paul or Mary	0	0	0
Peter, Paul or Mary	0.2	0.5	0.1

Table 1.2: The conjunctive combination rule : the three suspects' saga.
(* Not Peter, nor Paul nor Mary)

Distinctness can be formally defined as follows.

Suppose you first obtain some information I_1 about a particular variable O taking values in Ω and build a belief function m_1 on the actual value of O. You then obtain some additional information I_2 and consequently update your belief m_1 into m_{12} via a specialization matrix $\mathbf{S}_{12}$.

Now suppose you obtain the same pieces of information in the reverse order, i.e., you first get information I_2, build a belief function m_2, then get information I_1 and update your knowledge into m_{21} via a specialization matrix $\mathbf{S}_{21}$.

If your two sources of information are equally reliable, you would want $m_{12} = m_{21}$, i.e., the order in which you get the two sources of information should not matter. It is said that if $m_{12} = m_{21}$ implies that $\mathbf{S}_{12}$ only depends upon I_2 and S_{21} only derives from I_1, –in other words, $\mathbf{S}_{12}$ can entirely be defined from m_2 and $\mathbf{S}_{21}$ from m_1–, then m_1 and m_2 are distinct. In this case, $\mathbf{S}_{12}$ and $\mathbf{S}_{21}$ are, respectively, the Dempsterian specialization matrices $\mathbf{S}_{m_2}$ and $\mathbf{S}_{m_1}$.

In practice, m_1 and m_2 are termed distinct if they come from completely different, independent (in the sense that they cannot influence each other), sources of information.

If (at least) one of the sources might be unreliable, it is better to carry out a disjunctive combination:

Definition 12. (Disjunctive rule)

$$m_{1\textcircled{\cup}2}(A) = \sum_{B \cup C = A} m_1(B) m_2(C), \forall A \subseteq \Omega. \tag{1.26}$$

This operation corresponds to a more cautious attitude than that of the conjunctive combination rule. Knowing that :

- two different but not totally conflicting pieces of evidence were provided by two sources of information;
- one of the sources of information might not be reliable but we do not know which;

none of the possibilities suggested by the two pieces of evidence may be ruled out. The natural deduction is that the truth is represented either by one of the pieces of evidence, either by the other, either by both (this is equivalent to a logical "OR"). It is an operation in which the mass of belief respectively allocated to two subsets $A, B \subseteq \Omega$ by m_1 and m_2 is transferred onto $A \cup B$. The disjunctive combination of a mass m_1 with a mass m_2 is a particular form of generalization.

Example 2. (The murderer (continued)) *Reconsider the murder case described in example 1 under the hypotheses that both the witnesses are sure of what they are saying, but one of them*

might be lying. The murderer might then be either Peter or Mary. Now, if the witnesses are not sure of what they are saying and you are not sure they are telling the truth anyway, then, supposing Witness-1 (denoted W1) says he/she is 80% sure the murderer was smoking, and Witness-2 (W2) says he/she is only 50% sure the killer was a woman, the belief function associated with each witness is represented in table 1.3, together with their conjunctive and disjunctive combinations.

Suspect	m_1	m_2	$m_{1\textcircled{\cap}2}$	$m_{1\textcircled{\cup}2}$
$\emptyset^*$	0	0	0	0
Peter	0	0	0	0
Paul	0	0	0	0
Peter or Paul	0	0	0	0
Mary	0	0.5	0.5	0
Peter or Mary	0.8	0	0.4	0.4
Paul or Mary	0	0	0	0
Peter, Paul or Mary	0.2	0.5	0.1	0.6

Table 1.3: The disjunctive and conjunctive combination rules : the three suspects' saga. (* Not Peter, nor Paul nor Mary)

Property 3. *The disjunctive combination rule is commutative and associative; it neutral element is $m_\emptyset$, such that $m_\emptyset(\emptyset) = 1$.*

Note that the use of the disjunctive combination also requires m_1 and m_2 to be distinct. A definition of the term *"distinct"*, similar to that given for the conjunctive combination rule, could also be given for the disjunctive rule of combination.

Operations on a product space

From now on, we will work on the product space $\mathcal{X} \times \Omega$. X is a random variable (r.v.) varying over $\mathcal{X}$, assumed to be representative of the state of a system at a given time. $\Omega = \{\omega_0, \ldots, \omega_K\}$ is a finite set describing all possible states of the system. The ω_i are termed *classes*, and they are mutually exclusive. Variable O takes values in Ω.

Definition 13. (Marginalization)
Let $m^{\mathcal{X}\times\Omega}$ denote a bba defined on the Cartesian product $\mathcal{X} \times \Omega$ of the two variables X and O. The marginal bba *$m^{\mathcal{X}\times\Omega\downarrow\Omega}$ on Ω is defined, for all $B \subseteq \Omega$ as:*

$$m^{\mathcal{X}\times\Omega\downarrow\Omega}(B) = \sum_{\left\{A\subseteq(\mathcal{X}\times\Omega)\,\middle|\,A^{\downarrow\Omega}=B\right\}} m^{\mathcal{X}\times\Omega}(A), \tag{1.27}$$

where $A^{\downarrow\Omega}$ denotes the projection of A onto Ω:

$$A^{\downarrow\Omega} = \{o \in \Omega \,|\, \exists x \in \mathcal{X},\ (x, o) \in A\}. \tag{1.28}$$

The marginalization on $\mathcal{X}$ may be defined symmetrically.

Definition 14. (Vacuous Extension) *The inverse operation is the* vacuous extension. *Let m^Ω be a bba on Ω. Its vacuous extension on $\mathcal{X} \times \Omega$ is defined as:*

$$m^{\Omega\uparrow\mathcal{X}\times\Omega}(A) = \begin{cases} m^\Omega(B) \text{ if } A = B \times \mathcal{X} \text{ for some } B \subseteq \Omega \\ 0 \text{ otherwise.} \end{cases} \tag{1.29}$$

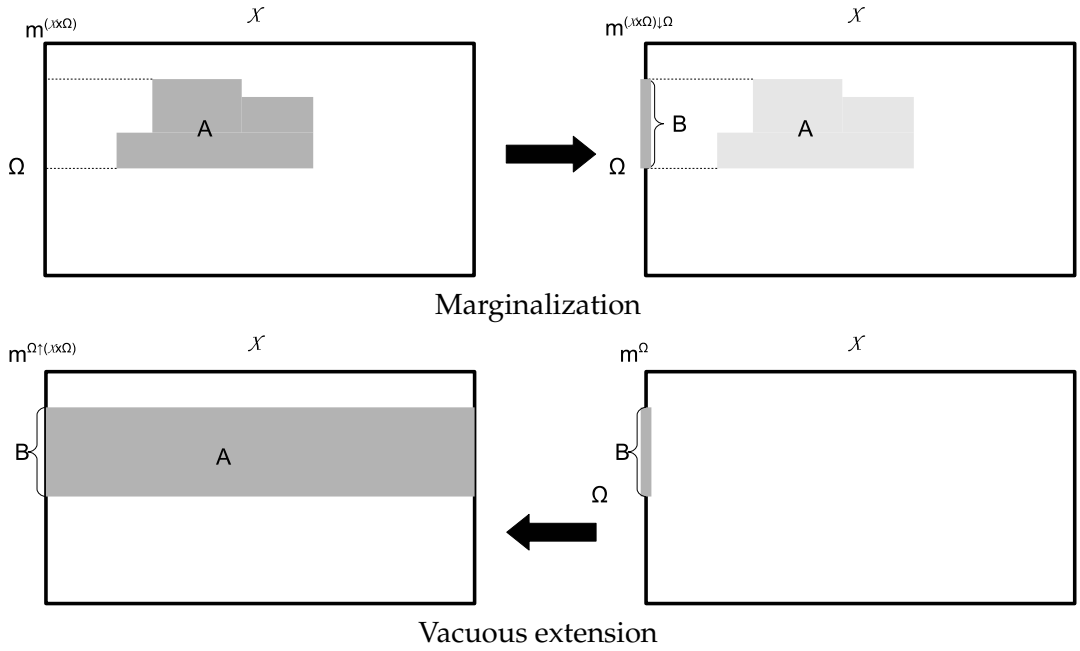


Figure 1.4: Marginalization and vacuous extension operations

We have already seen in the case of bba on a simple frame (as opposed to product space frame) that, when a given hypothesis $H \subseteq \Omega$ is ascertained, the beliefs are altered to reflect the new state of knowledge. In fact, the conditioning operation consists in combining masses conjunctively with a categorical bba m_H^Ω supporting hypothesis $H \subseteq \Omega$. In other words, knowing $m^{\Omega\times\mathcal{X}}$, and knowing that the hypothesis $H \subseteq \Omega$ holds, conditioning with respect to H consists in seeking the marginal bba $m^\mathcal{X}$ on $\mathcal{X}$ that takes all the available information into account. It may be found through a series of operations that have already been defined:

- first, m_H^Ω should be vacuously extended over $\mathcal{X} \times \Omega$ so as to get a belief function $m_H^{\Omega\uparrow\Omega\times\mathcal{X}}$ on $\Omega \times \mathcal{X}$ that may be combined with $m^{\Omega\times\mathcal{X}}$;
- second, $m^{\Omega\uparrow\Omega\times\mathcal{X}}$ and $m^{\Omega\times\mathcal{X}}$ should be combined conjunctively;
- the resulting belief function $m^{\mathcal{X}\times\Omega}\textcircled{\cap} m_H^{\Omega\uparrow\Omega\times\mathcal{X}}$ can then be marginalized on $\mathcal{X}$.

Definition 15. (Conditioning)
The mass of belief on $\mathcal{X}$ knowing that hypothesis $H \subseteq \Omega$ holds, i.e., $m_H^\Omega(H) = 1$, is:

$$m^\mathcal{X}[H] = \left(m^{\mathcal{X}\times\Omega}\textcircled{\cap} m_H^{\Omega\uparrow\mathcal{X}\times\Omega}\right)^{\downarrow\mathcal{X}}, \tag{1.30}$$

or, equivalently, the mass of belief allocated to $S \subseteq \mathcal{X}$ knowing that hypothesis $H \subseteq \Omega$ holds, is:

$$m^\mathcal{X}[H](S) = \sum_{B\subseteq\mathcal{X}\times\Omega \mid Proj(B\cap(H\times\mathcal{X})\downarrow\Omega)=S} m^{\mathcal{X}\times\Omega}(B). \tag{1.31}$$

where $Proj(C \downarrow \Omega)$ denotes the projection of a subset C of $\mathcal{X} \times \Omega$ on Ω.

Now let $m^\mathcal{X}[H]$ be the bba on $\mathcal{X}$ conditioned with respect to $H \subseteq \Omega$. Assume we now learn H finally does not necessarily hold and all previous states of knowledge have been lost. We have $m^\mathcal{X}[H]$, and would like to find the least committed bba $m^\mathcal{X}[H]^{\Uparrow(\mathcal{X}\times\Omega)}$ on $\mathcal{X} \times \Omega$) reflecting the available information. This procedure, opposite of the conditioning operation, is termed *ballooning extension* process [116], and yields to:

Definition 16. (Ballooning Extension)

$$m^{\mathcal{X}}[H]^{\Uparrow(\mathcal{X}\times\Omega)}(A) = \begin{cases} m^{\mathcal{X}}[H](S) \; if \; A = (S \times H) \cup (\mathcal{X} \times \overline{H}), \\ 0 \; otherwise. \end{cases} \tag{1.32}$$

Masses associated with any non-empty set S of $\mathcal{X}$ before the ballooning extension are then transferred onto $(S \times H) \cup (\mathcal{X} \times \overline{H})$ during this operation.

Note again that some information might be lost in the process of successive conditioning and "deconditioning" operations, and the original bba cannot always be recovered.

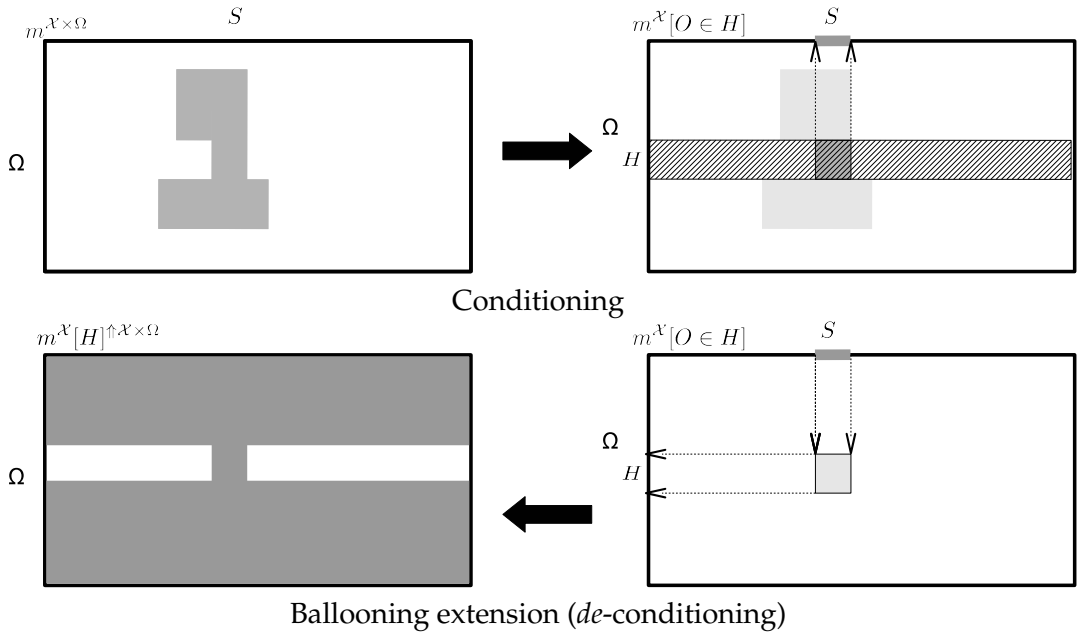


Figure 1.5: Conditionning and ballooning extension

The Generalized Bayes' Theorem

The Generalized Bayes' Theorem (GBT) was introduced by Smets [116]. It generalizes Bayes' theorem in that, whenever the belief functions are Bayesian, and we also have a Bayesian prior on the classes, the two theorems are exactly equivalent. However, the power of the GBT lies in the fact that it does not require any prior knowledge on Ω (for instance, no prior class probabilities).

Let us suppose we know all the conditional bbas $m^{\mathcal{X}}[\omega_k]$, $k = 0, \ldots, K$, we have no prior knowledge on Ω, and we observe $x_* \subseteq \mathcal{X}$. From that, we would like to derive our belief in the fact that the system is in a particular state ω_i, knowing the value of statistic X. In other words, we seek $m^{\Omega}[x_*]$. The GBT allows us to find the answer in three steps (See No. 1 to 3 of Figure 1.6 for an illustration).

1. We shall first calculate the ballooning extension of each of the functions $m^{\mathcal{X}}[\omega_k]$, that is to say, *"de-condition"* them in order to obtain a belief on $\mathcal{X} \times \Omega$.

2. The obtained bbas $m^{\mathcal{X}}[\omega_k]^{\Uparrow\mathcal{X}\times\Omega}$ are distinct, as the original $m^{\mathcal{X}}[\omega_k]$ were distinct. Hence, the $m^{\mathcal{X}}[\omega_k]^{\Uparrow\mathcal{X}\times\Omega}$ can be combined by applying the conjunctive combination rule: this will be the second step. We now have a global and un-conditioned belief function on $\mathcal{X} \times \Omega$.

3. Conditioning with respect to x_* returns the belief function we need, namely $m^{\Omega}[x_*]$.

The GBT may then be defined as follows:

Definition 17. (GBT)

$$m^{\Omega}[x_*] = \left(\textcircled{\cap}_{k=0}^{K} m^{\mathcal{X}}[\omega_k]^{\Uparrow \mathcal{X}\times\Omega}\right)[x_*]. \tag{1.33}$$

It may be shown that:

$$m^{\Omega}[x_*] = \textcircled{\cap}_{k=0}^{K} \overline{\{\omega_k\}}^{pl^{\mathcal{X}}[\omega_k](x_*)}. \tag{1.34}$$

where A^w denotes a simple bba such that $m(A) = 1 - w$ and $m(\Omega) = w$. In particular, $\overline{\{\omega_k\}}^{pl^{\mathcal{X}}[\omega_k](x_*)}$ is a simple bba for which $m(\overline{\{\omega_k\}}) = 1 - pl^{\mathcal{X}}[\omega_k](x_*)$ and $m(\Omega) = pl^{\mathcal{X}}[\omega_k](x_*)$.

Equation (1.34) [40] allows an easy interpretation of the GBT: the less x_* is plausible under ω_k, the more weight is assigned to $\overline{\{\omega_k\}}$ when x_* occurs.

The equivalent formulation in terms of plausibility functions sometimes makes calculations simpler:

$$pl^{\Omega}[x_*](A) = 1 - \prod_{\omega_k \in A} (1 - pl^{\mathcal{X}}[\omega_k](x_*)), \quad \forall A \subseteq \Omega. \tag{1.35}$$

Discounting

A mass of belief $m^{\Omega}[EC]$ has been defined as the belief held by an agent over the actual value of a variable O *given an evidential corpus* EC, i.e. conditionally to EC. The reliability of the source $\mathscr{S}$ providing the evidential corpus EC may sometimes be assessed, and $m^{\Omega}[EC]$ should be updated accordingly.

If the source $\mathscr{S}$ is perfectly reliable, then the beliefs need not be updated. If, on the contrary, the source $\mathscr{S}$ is not reliable at all, then the mass $m^{\Omega}[EC]$ should be reduced to the vacuous belief function, representing total ignorance.

Finally, if the source $\mathscr{S}$ is only partially reliable, then mass $m^{\Omega}[EC]$ should be altered in such a way that the updated values of the mass are proportional to the reliability of $\mathscr{S}$.

Let us consider a frame of discernment $\mathcal{R} = \{R, NR\}$ whose two elements respectively stand for *Reliable* and *Non-Reliable*. Let us consider a mass $m^{\mathcal{R}}$ on $\mathcal{R}$ of the form

$$\begin{array}{l} m^{\mathcal{R}}(\{R\}) = 1 - \alpha \\ m^{\mathcal{R}}(\mathcal{R}) = \alpha, \end{array} \tag{1.36}$$

representing the reliability of source $\mathscr{S}$.

m^{Ω} and $m^{\mathcal{R}}$ may be combined to reflect the new state of knowledge.

- They first need to be expressed in a common frame, that is to say, the product space.
- They can then be conjunctively combined.
- The obtained belief function should finally be marginalized so that the beliefs on the value of O into Ω can be identified.

The resulting belief function, denoted ${}^{\alpha}m^{\Omega}$ may be expressed as follows [113, 116]:

Definition 18. (Discounting)

$${}^{\alpha}m^{\Omega} = \left(m^{\Omega}[\mathcal{R}]^{\Uparrow \Omega\times\mathcal{R}} \textcircled{\cap} m^{\mathcal{R}\uparrow\Omega\times\mathcal{R}}\right)^{\downarrow\Omega}. \tag{1.37}$$

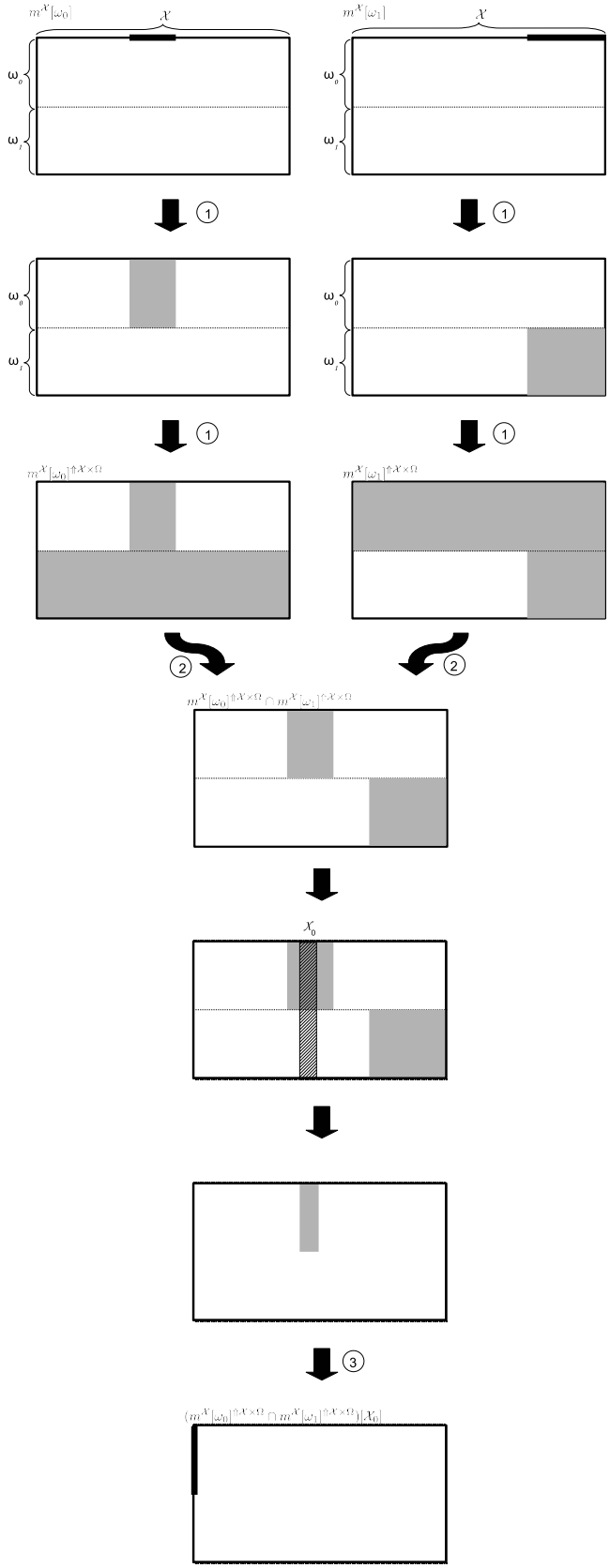


Figure 1.6: The three steps of the GBT

where α is called the *discounting rate*.

It may be shown that a simpler expression of ${}^{\alpha}m^{\Omega}$ is the following:

$$\begin{aligned} {}^{\alpha}m^{\Omega}(A) &= (1-\alpha)m^{\Omega}(A), \quad \forall A \in \Omega, \\ {}^{\alpha}m^{\Omega}(\Omega) &= (1-\alpha)m^{\Omega}(\Omega) + \alpha. \end{aligned} \tag{1.38}$$

This operation amounts to transferring part of the mass allocated to the focal elements onto Ω, thus accounting for partial ignorance.

Yet another way of writing this operation is:

$${}^{\alpha}m^{\Omega} = (1-\alpha)m^{\Omega} + \alpha m^{\Omega}_{\Omega}, \tag{1.39}$$

where m^{Ω}_{Ω} represents the vacuous belief function on Ω.

It is also possible that the source $\mathscr{S}$ is known to be reliable in a given context and less reliable in some other contexts. For example, a temperature sensor might be reliable in a given range of temperature values and less reliable (or totally unreliable) outside this range. In this case, it is possible to perform *contextual discounting*, as described in [86, 87].

1.2.3 Decision Making

Once all pieces of information have been collected and all beliefs have been modeled, updated and combined, time comes when a decision should be made. Making a decision consists in choosing a singleton according to a given rule.

Pignistic Probabilities

According to DeGroot (1970) [27], decisions will only be coherent if the underlying uncertainties can be described by a probability distribution defined on 2^{Ω}. Therefore, the belief function gathering the available knowledge on the possible value of O should first be transformed into a so-called *pignistic* probability ($BetP^*$), on the singletons of Ω. Then, the elected singleton is the one that maximizes $BetP$.

Definition 19. (Pignistic probability) *The pignistic transformation may be defined as follows (See [126, 124] for a justification):*

$$BetP^*(\omega) = \sum_{A \subseteq \Omega : \omega \in A} \frac{m^*(A)}{|A|}, \forall \omega \in \Omega. \tag{1.40}$$

Example 3. (The murderer (continued)) *Consider the Peter, Paul and Mary saga described in exemple 1 for the last time. The pignistic probability associated with the disjunctive and conjunctive combination of the two testimonies read:*

Isopignistic belief function

The pignistic transform is not bijective. In effect, it often happens that several belief functions lead to the same pignistic probability, while, in contrast, a given belief function can only lead to a specific pignistic probability. The set of bba sharing the same pignistic probability is called a set of *isopignistic* belief functions. Amongst isopignistic belief functions, and in the absence of additional information, the least committed one should always be chosen as a result of the LCP.

Dubois, Prade and Smets [45] demonstrated that the q-least committed mass function associated with a given pignistic probability distribution $BetP$ is unique and consonant. It may be recovered from $BetP$ as follows:

$$pl(\{\omega_i\}) = \sum_{j=1}^{n} \min(\mathbb{P}_i, \mathbb{P}_j), \quad \forall i \in \{1, \ldots, n\}, \tag{1.41}$$

Suspect	m_1	m_2	$m_{1\bigcap_2}$	$m_{1\bigcup_2}$	$BetP_{1\bigcap_2}$	$BetP_{1\bigcup_2}$
$\emptyset^*$	0	0	0	0	–	–
Peter	0	0	0	0	0.233	0.4
Paul	0	0	0	0	0.033	0.2
Peter or Paul	0	0	0	0	–	–
Mary	0	0.5	0.5	0	0.733	0.4
Peter or Mary	0.8	0	0.4	0.4	–	–
Paul or Mary	0	0	0	0	–	–
Peter, Paul or Mary	0.2	0.5	0.1	0.6	–	–

Table 1.4: Pignistic probability associated with the disjunctive and conjunctive combination rules.
The three suspect's saga. (* Not Peter, nor Paul nor Mary)

where

- $\mathbb{P}_i = BetP(\{\omega_i\})$,
- and $\mathbb{P}_1 \geq \mathbb{P}_2 \geq \ldots \geq \mathbb{P}_n$.

1.3 Continuous Case

1.3.1 Continuous Domain

The above described tools may be extended to the case where the frame of discernment $\mathcal{X}$ is continuous, typically $\mathcal{X} \subseteq \mathbb{R}$.

Let $\mathcal{I}_{[\alpha,\beta]}$ be a set of closed intervals such that:

$$\mathcal{I}_{[\alpha,\beta]} = \{[x,y] : \alpha \leq x \leq y \leq \beta\}. \tag{1.42}$$

$\mathcal{I}_{[-\infty,+\infty]}$ is denoted $\mathcal{I}$. Note that the infinities are included. The focal elements of a mass of belief $m^{\mathcal{X}}$ defined on a continuous domain $\mathcal{X}$ are elements of $\mathcal{I}_{[\alpha,\beta]}$, $\alpha, \beta \in \mathbb{R} \cup \{-\infty, +\infty\}$. They are assumed to be closed intervals, so that belief functions are additive i.e.

$$\begin{array}{l} \forall A_1, A_2, \ldots, A_n \in \mathcal{I}_{[\alpha,\beta]} : A_j \cap A_k \neq \emptyset, (j,k) \in \{1,\ldots,n\}^2, j \neq k \\ \text{then} \\ bel^{\mathcal{X}}(\bigcup_{i=1,\ldots,n} A_i) = \sum_{i=1}^{n} bel(A_i). \end{array} \tag{1.43}$$

An interesting representation (Figure 1.7) of the elements of $\mathcal{I}_{[\alpha,\beta]}$ is given by points in a two-dimensional frame represented by an isocel right-angled triangle $\mathbb{T}_{[\alpha,\beta]}$, oriented in such a way that it looks like the upper left hand corner of a square figure. The horizontal side of the triangle serves as abscissa, and represents the lower bound of an interval, and the vertical side serves as ordinate, representing the upper bound of an interval. Hence, any point in the triangle or on the triangle boundaries represents a non-empty interval of $\mathcal{I}_{[\alpha,\beta]}$. The triangle associated with $\mathcal{I}$ is denoted $\mathbb{T}$. An interval $[x,y]$ is shown as a point in Figure 1.7.

1.3.2 Types of Belief Functions Defined on $\mathbb{R}$

Discrete BF on $\mathbb{R}$

Under the constraint that the set of focal elements $\mathcal{F}(m)$ of $m^{\mathcal{X}}$ is finite, the tools described in Section 1.2 directly apply even in the case where $\mathcal{X}$ is continuous and $\mathcal{X} \subseteq \mathbb{R}$.

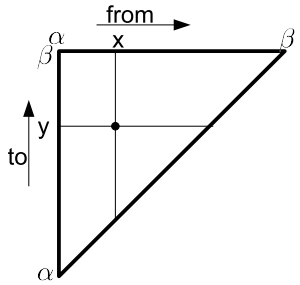


Figure 1.7: Triangular representation of $\mathcal{I}_{[\alpha,\beta]}$.

A bba $m^{\mathcal{X}} : \mathcal{X} \rightarrow [0,1]$ with the property $\sum_{i=1}^{n} m^{\mathcal{X}}(A_i) = 1$ is a discrete bba on $\mathcal{X}$ as described in section 1.2. Typically, focal elements are chosen among intervals or, more generally, Borel sets [136, 50, 135, 95]. Denoting $m_i = m(A_i)$, with $\sum_{i=1}^{n} m_i = 1$, and assuming $A_i \neq \emptyset$ for all i, Equations (1.2)-(1.4) may be rewritten as follows:

$$bel(A) = \sum_{A_i \subseteq A} m_i, \tag{1.44}$$

$$pl(A) = \sum_{A_i \cap A \neq \emptyset} m_i, \tag{1.45}$$

and

$$q(A) = \sum_{A_i \supseteq A} m_i, \tag{1.46}$$

for all $A \in \mathcal{B}(\mathcal{X})$, where $\mathcal{B}(\mathcal{X})$ denotes the Borel sigma-algebra on $\mathcal{X}$.

Continuous BF

A more complex generalization of the concepts of section 1.2 is given when the number of focal elements is not finite any more. In this case, m is no longer a bba but a basic belief *density* (bbd): instead of discrete masses defined on points of $\mathbb{T}_{[\alpha,\beta]}$, a continuous mass density is defined over the area of $\mathbb{T}_{[\alpha,\beta]}$ [2, 29, 123].

A normal basic belief density $m^{\mathcal{X}}$ is a density function such that:

$$\int_{x=\alpha}^{x=\beta} \int_{y=x}^{y=\beta} m^{\mathcal{X}}(x,y) dy dx = 1. \tag{1.47}$$

Normalization is of course not necessary. In order to define a *subnormal* bba, the integral of m over $\mathcal{I}_{[\alpha,\beta]}$ may be allowed to be less than 1, the complement being allocated to the empty set.

The belief, plausibility, commonality and implicability functions associated with m can be defined in the same way as in the finite case, replacing finite sums by integrals. The following definitions hold:

$$bel(A) = \iint_{[x,y] \subseteq A} m(x,y)\, dx\, dy, \tag{1.48}$$

$$pl(A) = \iint_{[x,y] \cap A \neq \emptyset} m(x,y)\, dx\, dy, \tag{1.49}$$

$$q(A) = \iint_{[x,y] \supseteq A} m(x,y)\, dx\, dy, \tag{1.50}$$

for all $A \in \mathcal{B}(\mathcal{X})$. In particular, when $A = [x,y]$,

$$bel([x,y]) = \int_x^y \int_u^y m(u,v) dv du, \tag{1.51}$$

$$pl([x,y]) = \int_{-\infty}^y \int_{\max(x,u)}^{+\infty} m(u,v) dv du, \tag{1.52}$$

$$q([x,y]) = \int_{-\infty}^x \int_y^{+\infty} m(u,v) dv du, \tag{1.53}$$

for all $x, y \in \mathcal{I}_{[\oslash,\odot]}$.

The domains of these integrals may be represented as the shaded areas in Figure 1.8.

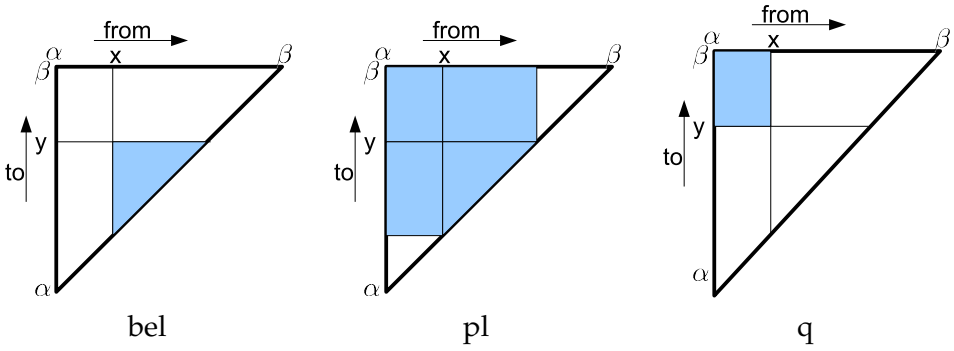


Figure 1.8: Domains of integration for bel, pl and q.

Moreover, m may be recovered from *bel* or q through:

$$m(x,y) = -\frac{\partial^2 bel([x,y])}{\partial x \partial y} = -\frac{\partial^2 q([x,y])}{\partial x \partial y}, \tag{1.54}$$

provided these derivatives exist. This recovery process does not lead to any loss of information.

All other concepts described in Section 1.2 remain valid except that masses become densities and sums are replaced with integrals. However, the associated mathematics sometimes become much more complex than in the discrete case.

Special cases

- A *vacuous* bbd on a continuous domain $\mathcal{X} = [\alpha, \beta]$ is a bbd for which $m(\mathcal{X}) = 1$;
- A *categorical* bbd is such that $m^{\mathcal{X}}(a,b) = \delta(x-a, y-b)$, for $[a,b] \in \mathcal{I}_{[\alpha,\beta]}$ and $[a,b] \neq \mathcal{X}$ (where δ denotes the Dirac delta function);
- A *consonant* bbd has nested focal elements;
- A *Bayesian* bbd is a bbd whose focal elements are singletons, i.e. its possible focal elements are on the hypotenuse of triangle $I_{[\alpha,\beta]}$. It is a probability density function.

Extension to $\mathbb{R}^n$

According to Smets [123], *"The real issue underlying the possibility of extending belief functions on $\mathbb{R}^n$ is the existence of a finite dimensional real space, the elements of which are in one-to-one correspondence with the focal elements."*. If all focal elements can be represented as a point in $\mathbb{R}^d$ for some $d > 0$, the theory extends directly. This can be done, e.g., when the focal elements are convex, closed geometrical figures. However, computations become highly complex in these cases.

Caron et al. carried out a generalization to $\mathbb{R}^n$ for the special case of basic belief densities induced by multivariate Gaussian probability density functions [20].

1.3.3 Belief Updating

The main operations for the updating of beliefs in the continuous case are the same as in the discrete case. It is therefore possible to marginalize, vacuously extend, condition or decondition a continuous belief function, to combine two continuous BF conjunctively or disjunctively, or to apply the GBT to continuous belief functions. The reader is referred to [123] for more details.

1.3.4 Ordering

The pl-, q- and s- ordering may be defined exactly as in the finite case, i.e.

- If $pl_1(X) < pl_2(X), \forall X \in \mathcal{X}$, then $m_1 \sqsubseteq_{pl} m_2$ (pl-ordering);
- if $q_1(X) < q_2(X), \forall X \in \mathcal{X}$, then $m_1 \sqsubseteq_q m_2$ (q-ordering);
- and if m_1 is a specialization of m_2 then $m_1 \sqsubseteq_s m_2$ (s-ordering).

The relative properties of the different orderings given in section 1.2.1 still hold.

Based on these orderings, the least commitment principle, a fundamental axiom of the TBM, still applies in the continuous case, and works as in the discrete case.

1.3.5 Decision Making

As already mentioned in Paragraph 1.2.3, and according to DeGroot [27], decision making requires the definition of a (pignistic) probability function that describes the odds in favour of each possible hypothesis. We will now describe how this can be done in the case of continuous belief functions.

From m to $BetP$

Consider a normalized bbd $m^{\mathcal{X}}$ describing the belief of agent Ag over the value of a variable X taking values in $\mathcal{X}$. Let us define a pignistic density function $Betf$ associated with $m^{\mathcal{X}}$ and the related pignistic distribution $BetF$. Let us additionaly define $BetP$ as the pignistic probability of a given event. Then

$$\begin{aligned} BetP([a,b]) &= \int_{x=-\infty}^{\infty} \int_{y=x}^{\infty} \frac{|[a,b] \cap [x,y]|}{|[x,y]|} m(x,y) dy dx \\ &= \int_{x=-\infty}^{b} \int_{y=a \vee x}^{\infty} \frac{y \wedge b - x \vee a}{y - x} m(x,y) dy dx, \end{aligned} \tag{1.55}$$

where, by continuity, $\frac{|\emptyset|}{|[x,y]|} = 0$, and $\frac{(y \wedge b - x \vee a)}{(y-x)} = 1$ when $a < x = y < b$.

Moreover,

$$Betf(a) = \lim_{\epsilon \to 0} \int_{x=-\infty}^{a} \int_{y=a+\epsilon}^{\infty} \frac{1}{y-x} m(x,y) dy dx, \tag{1.56}$$

hence

$$BetP([a,b]) = \int_{a}^{b} Betf(x) dx. \tag{1.57}$$

Remark 5. *Betting* always *requires normalization as no decision can be taken towards the empty set (a decision towards the empty set would not make sense).*

Remark 6. *There is no bijection between $m^{\mathcal{X}}$ and $Betf$, i.e. different bbd may lead to the same pignistic density function. Such bbds are said to be isopignistic.*

From $BetP$ to m

It is important to note that a probability function on a set of real numbers can be interpreted in two different ways.

On the one hand, it can be seen as directly representing an agent's belief about the values that may be taken by a variable X varying over $\mathbb{R}$ (or any subset of $\mathbb{R}$ including the observed values). In this case, a Bayesian belief function is directly observed.

On the other hand, the collected probability function represents the way the agent would bet about the value of variable X. In that case, the observed function is the pignistic probability associated with the agent's beliefs. Again, two sub-cases can be derived:

- either the number of values that can be taken by X is finite, and the observed values define a set of constraints over a discrete belief function on a finite domain,
- or X varies over a continuous domain and the observed values define a set of constraints over a continuous belief function.

In both cases, the least committed belief function satisfying those constraints should be selected as a working basis.

The solution for the discrete belief function with continuous domain has been described in Section 1.2.3, Equation (1.40). Let us consider the case where X varies over a continuous domain $\mathcal{X}$. The pignistic transform being a many-to-one operation, the least commitment principle again needs to be applied when $m^{\mathcal{X}}$ has to be deduced from $Betf$.

Generally speaking, Dubois, Prade and Smets [45], demonstrated that, both in the finite and continuous cases, the q-least committed element of a set of isopignistic belief functions is a consonant belief function. This means that the focal elements of the sought belief function are nested. The observed pignistic density hence defines a set of constraints on $m^{\mathcal{X}}$, and an optimization problem needs to be solved, so that $q^{\mathcal{X}}$ may be maximized under those constraints (Recall that the higher $q^{\mathcal{X}}$, the least committed the belief function). The solution cannot be described simply in the general case.

Nevertheless, Smets [123] demonstrated there exist a simple solution when $Betf$ is a unimodal, "bell-shaped" density. He showed that the focal sets of $m^{\mathcal{X}}$ are the level sets of the density function $Betf$. They are intervals $[a,b]$ such that $Betf(a) = Betf(b)$. Given the upper bound b of any such interval, the lower bound is uniquely defined by a function γ such that $a = \gamma(b)$ for all $b \geq \nu$. The bbd is defined by

$$m^{\mathcal{X}}(a,b) = \theta(b)\delta(a - \gamma(b)), \tag{1.58}$$

with

$$\theta(b) = (\gamma(b) - b)Betf'(b), \tag{1.59}$$

where $Betf'$ is the derivative of $Betf$ and δ is the Dirac delta function. As already mentioned, $m^{\mathcal{X}}$ is consonant. Consequently, the associated plausibility function is a possibility measure. The corresponding possibility distribution π is given by:

$$\pi(x) = pl(\{x\}) = \begin{cases} \int_x^{+\infty} (\gamma(t) - t) Betf'(t) dt & \text{if } x \geq \nu \\ \int_{\gamma^{-1}(x)}^{+\infty} (\gamma(t) - t) Betf'(t) dt & \text{otherwise.} \end{cases} \tag{1.60}$$

If $Betf$ is symmetrical, then $\gamma(x) = 2\nu - x$, and the above equation simplifies to

$$\pi(x) = \begin{cases} 2(x-\nu) Betf(x) + 2\int_x^{+\infty} Betf(t) dt & \text{if } x \geq \nu \\ 2(\nu - x) Betf(x) + 2\int_{-\infty}^{x} Betf(t) dt & \text{otherwise.} \end{cases} \tag{1.61}$$

Example 4. *Let f_0 be the density function of the exponential distribution $\mathcal{E}(\mu)$ with mean $\mu > 0$:*

$$f_0(x;\mu) = \begin{cases} \frac{1}{\mu} e^{-x/\mu} & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{1.62}$$

This is a unimodal density with mode $\nu = 0$, and $\gamma(b) = 0$ for all $b \geq 0$. The corresponding q-LC distribution may be computed from (1.60). It is equal to

$$\pi(x;\mu) = \int_x^{+\infty} \frac{1}{\mu^2} t e^{-t/\mu} dt \tag{1.63}$$

$$= e^{-x/\mu} \left(1 + \frac{x}{\mu}\right) \tag{1.64}$$

for $x \geq 0$ and $\pi(x) = 0$ for $x < 0$. This function is plotted in Figure 1.9 for different values of μ.

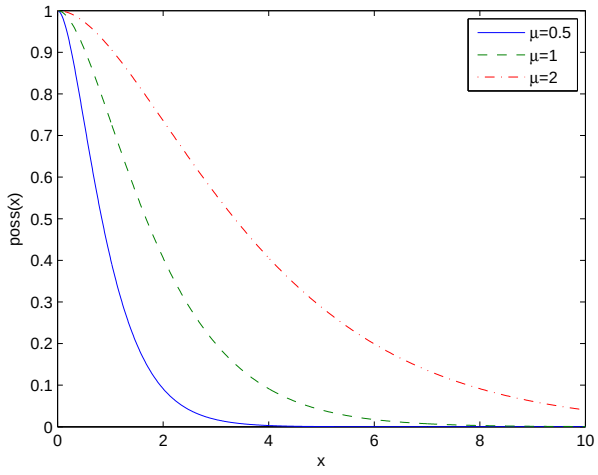


Figure 1.9: q-LC possibility distribution induced by an exponential probability density $\mathcal{E}(\mu)$, for three different values of μ.

Example 5. *Now, let f_0 be the density function of the normal distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2:*

$$f_0(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right).$$

This is a symmetrical unimodal density with mode μ. The corresponding q-LC distribution may be computed from (1.61). It is equal to

$$\pi(x;\mu,\sigma) = \begin{cases} \frac{2(x-\mu)}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) + 2\left(1-\Phi\left(\frac{x-\mu}{\sigma}\right)\right) & \text{if } x \geq \mu \\ \frac{2(\mu-x)}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) + 2\Phi\left(\frac{x-\mu}{\sigma}\right) & \text{otherwise,} \end{cases} \tag{1.65}$$

where Φ is the standard normal cumulative distribution function.

This function is plotted in Figure 1.10 for $\mu = 0$ and three different values of σ.

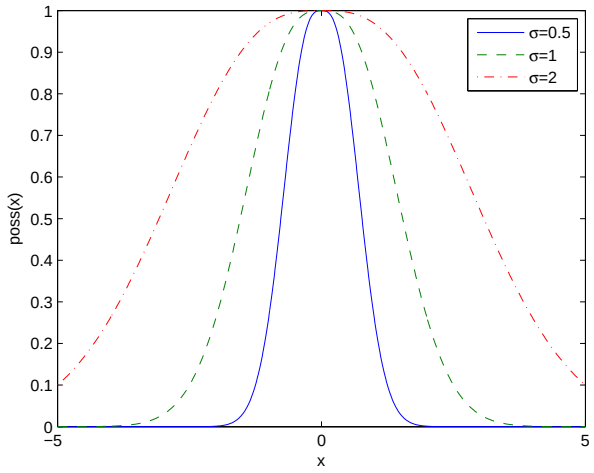


Figure 1.10: q-LC possibility distribution induced by a normal probability density $\mathcal{N}(\mu,\sigma^2)$ for $\mu = 0$ and three different values of σ.

1.4 Predictive Belief Functions (previous works)

In this section, we summarize the concept of Predictive Belief Function (PBF) introduced in [41]. The problem may be defined as follows. Let T be a random variable varying over a domain $\mathcal{T}$. It may be discrete or continuous. Having observed the realization of an independent and identically distributed iid random sample $\mathbf{T} = \{T_1, ..., T_n\}$ of unknown distribution P_T, we would like to be able to represent an agent's belief about the realization of a new sample drawn from the same distribution. In other words, we would like to be able to built a predictive belief function on the value of T.

As a toy example, consider the case where T denotes the color of a ball taken from an urn containing balls of different colors. Having observed the colors of n balls randomly taken from the urn with replacement, we would like to quantify our belief regarding the color of the next ball to be drawn from the urn.

Let $bel_{\mathbf{T}}$ denote a belief function on $\mathcal{T}$ constructed using $\mathbf{T}$. In [41], Denœux postulates that such a belief function should satisfy the following two requirements:

$$\mathbb{P}\left\{bel_{\mathbf{T}}(A) \leq P_T(A), \forall A \in \mathscr{T}\right\} \geq 1-\alpha, \tag{1.66}$$

where $\alpha \in (0,1)$, and

$$\forall A \in \mathscr{T}, \quad bel_{\mathbf{T}}(A) \xrightarrow{P} P_T(A), \text{ as } n \to \infty, \tag{1.67}$$

where $\xrightarrow{P}$ denotes convergence in probability and $\mathscr{T}$ is $2^{\mathcal{T}}$ if $\mathcal{T}$ is finite, and $\mathscr{T} = \mathcal{B}(\mathcal{T})$ (Borel sigma-algebra generated by $\mathcal{T}$) if $\mathcal{T} \subseteq \mathbb{R}$.

The introduction of these two requirements may be justified as follows. We do not know the true distribution P_T of T, therefore our knowledge should be represented by a belief function that is less committed (less informative) than the true distribution P_T of T. However, it is much too stringent to require that $bel_{\mathbf{T}}^{T} \leq P_T$ all the time, as this would systematically lead to the vacuous belief function. This is why it is only required that (1.66) be true *with some pre-defined probability*, i.e. asymptotically, for at least a fraction $1 - \alpha$ of the samples. This justifies requirement (1.66).

In addition, the precision of our knowledge on the distribution of statistic T depends on the number of observations we dispose of. Had we observed an infinity of values of T, we would know its true distribution. According to Hacking's principle, $bel_{\mathbf{T}}$ should thus tend toward the true distribution P_T of T as the number of observation we dispose of tends to infinity, hence requirement (1.67).

A belief function $bel_{\mathbf{T}}$ satisfying requirements (1.66) and (1.67) is called a *predictive belief function (PBF) at confidence level* $1 - \alpha$. Methods for constructing such belief functions in the case of a discrete random variable T with discrete domain were described by Denœux [41], based on multinomial confidence region. Combining the works of Ferson [50], and Kriegler and Held [68] permits to built PBF for discrete random variables with continuous domain. Finally, we introduced a way of building continuous PBF for continuous random variables with continuous domain [6]. All these techniques will be discussed in Chapter 2.

From raw data to belief functions

Contents

2.1 Introduction **37**
2.2 Type I PBF **37**
2.2.1 Confidence Bands 38
2.2.2 Discrete predictive belief function on a discrete domain 41
2.2.3 Discrete predictive belief functions on $\mathbb{R}$ 42
2.2.4 Continuous predictive belief functions on $\mathbb{R}$ 46
2.2.5 Conclusion 53
2.3 Type II PBF **54**
2.3.1 Consonant Belief Function Induced by a Set of Pignistic Probabilities 55
2.3.2 Application to a Sample of a Discrete Random Variable 57
2.3.3 Construction of $\mathcal{P}$ 58
2.3.4 Determination of the q-MCD Possibility Distribution 59
2.3.5 Application to Continuous Parametric Models 62
2.3.6 Exponential Distribution 62
2.3.7 Normal Distribution 64
2.3.8 Conclusion 65
2.4 classification example **66**
2.5 Conclusion **70**

Summary

In this chapter, we show how belief functions can be built from data, considering the special case where the variable X of interest is defined from the result of a random experiment. It is thus a random variable, with unknown probability distribution P_X. The available information is assumed to consist in past observations collected from n independent repetitions of the same experiment, forming an independent random sample from P_X. Based on this information, we would like to be able to represent an agent's belief about the realization of a new sample drawn from the same distribution. In other words, we would like to be able to predict what the next observation will be.
In [42], a formalization of this problem was suggested, using the concept of *predictive belief function* (PBF). Practical methods for building belief functions were presented for the case where the domain $\mathcal{X}$ of X is discrete, based on multinomial confidence regions.
In the first section, this approach is extended to the case where X is a *continuous random variable*. The extension is based on confidence bands, which play a role similar to that of multinomial confidence regions in the discrete case.
In the second section, another approach is proposed, which may be argued to be more in line with the the two-level (credal, pignistic) structure of the TBM. The starting point of this method is the assumption that, if the probability distribution P_X of a random variable X is known, then the pignistic probability distribution associated with the BF quantifying our belief regarding future values of X should be P_X. As many BFs may satisfy this property, and there is no unique least committed element in the general case, the suggested solution selects the most committed consonant belief function amongst the BFs less committed than those verifying this property.

Résumé

Dans ce chapitre, nous montrons comment une fonction de croyance peut être construite à partir de données. Nous nous attachons à résoudre le cas particulier dans lequel la variable d'intérêt, X, est définie comme le résultat d'une expérience aléatoire. Il s'agit donc d'une variable aléatoire, de distribution de probabilité inconnue P_X. On suppose que l'information disponible consiste en une série d'observations collectées au cours d'une série de n répétitions de la même expérience, formant un échantillon aléatoire issu de P_X. A partir de cette information, nous montrons comment représenter les croyances d'un agent vis à vis de futures réalisations de X issues de la même distribution. En d'autres termes, nous prédisons la valeur de la prochaine observation.
Dans [42], une formalisation de ce problème est suggérée, utilisant le concept de fonction de croyance prédictive. Des méthodes pratiques, basées sur l'utilisation de régions de confiance multinomiales, sont introduites pour construire de telles fonctions de croyance dans le cas où X est une variable aléatoire discrète.
Dans la première section de ce chapitre, cette approche est étendue au cas où X est une variable aléatoire continue. Cette extension se base sur l'utilisation de bandes de confiance, qui jouent un rôle similaire aux régions de confiance multinomiales dans le cas discret.
Dans la seconde section, une autre approche est proposée. Cette dernière peut être considérée étant plus en accord que la précédente avec la structure à deux niveaux (crédal, pignistique) du modèle des croyances transférables. Le point de départ de cette méthode est l'hypothèse suivante: si la distribution de probabilité P_X d'une variable aléatoire X est connue, alors la distribution de probabilité pignistique associée à la fonction de croyance quantifiant nos croyances vis à vis de valeurs futures de X doit être P_X. Comme de nombreuses fonctions de croyance satisfont cette propriété, et que l'élément le moins engagé n'est pas unique dans le cas général, la solution suggérée est de choisir la fonction

de croyance consonante la plus engagée parmi celles qui sont moins engagées que celles qui satisfont cette propriété.

2.1 Introduction

In the past few years, belief functions have been developed as a tool for data fusion, but also for the management of uncertainty and various aspects of data mining or decision making (see Chapter 1). They are a very flexible tool even when few data are available. However, it is not always clear how to obtain belief functions or build them from raw data.

In this chapter, we will consider the special case where the variable X of interest is defined from the result of a random experiment. It is thus a random variable, with unknown probability distribution P_X. The available information is assumed to consist in past observations collected from n independent repetitions of the same experiment, forming an independent random sample from P_X. Based on this information, we would like to be able to represent an agent's belief about the realisation of a new sample drawn from the same distribution. In other words, we would like to be able to predict what the next observation will be.

There are two possible ways of building such a belief function.

The first method is based on Hacking's frequency principle [55, 117], which equates the degree of belief of an event to its probability (long run frequency), when the latter is known. The second method is based on a weak form of Hacking's principle, which states that the pignistic probability of an event should be equal to its long run frequency, when the latter is known.

This chapter will be divided in two main sections. Each section derives a solution corresponding to one of the above mentioned two points of view. We will thus introduce two types of predictive belief functions. The first type will be introduced in the following section, and the second type will be introduced later.

2.2 Type I predictive belief functions

Let us consider the first interpretation. As the probability distribution of X is unknown, the available information is incomplete and the precision of the obtained belief function should depend on the number of observations. In [42], a formalization of this problem was suggested, using the concept of *predictive belief function* (PBF). A PBF was defined in Section 1.4 as a belief function less committed than P_X with some user-defined probability, and converging in probability towards P_X as the size of the sample tends to infinity. Practical methods for building belief functions were presented for the case where the domain $\mathcal{X}$ of X is discrete, based on multinomial confidence regions.

In this section, the above approach is extended to the case where X is a *continuous random variable*. The extension is based on confidence bands, which play a role similar to that of multinomial confidence regions in the discrete case. When a confidence band is defined by step upper and lower bounding functions, it is known to be equivalent to a belief function on the real line with a finite number of focal intervals. We first show that this belief function is a predictive belief function as defined in [42]. We then consider the generalization to continuous confidence bands. In that case, the corresponding belief function is continuous, and we derive the expression of its basic belief density.

The section is organized as follows. First, a definition of confidence bands is given, and the construction of confidence bands of particular interest are detailed. Then, the solution of the problem of building a discrete PBF on a discrete domain is recalled. Next, it is shown that the least committed belief function (LCBF) built from a step confidence band as described by Kriegler and Held [68] is a discrete PBF on a continuous domain. The approach is finally extended to the construction of a continuous PBF on a continuous domain.

The results presented here were first published in [6] and [7].

2.2.1 Confidence Bands

Definition

Let us assume that X is a random variable with cumulative distribution function (cdf) F_X. In some cases, F_X is not precisely known, but a lower bounding function $\underline{F} : \mathbb{R} \to [0,1]$ and an upper bounding function $\overline{F} : \mathbb{R} \to [0,1]$ can be specified such that $\underline{F}(x) \leq F_X(x) \leq \overline{F}(x)$ for all $x \in \mathbb{R}$. The convex set of probabilities compatible with these constraints,

$$\Gamma_X(\underline{F}, \overline{F}) = \{P | \forall x \in \mathbb{R}, \underline{F}(x) \leq P((-\infty, x]) \leq \overline{F}(x)\}, \tag{2.1}$$

is called a *distribution band* [68].

In the special case where $\underline{F}$ and $\overline{F}$ are step functions, then $\Gamma_X(\underline{F}, \overline{F})$ is called a *probability box*[1], or p-box for short [50]. A continuous distribution band can always be enclosed in a p-box. The smallest discrete approximation is always obtained by choosing the lower and upper bounding step functions to be right and left-continuous, respectively [50]. From now on, only p-boxes possessing this property will be considered.

Suppose now that the available information about F_X takes the form of an iid random sample $\mathbf{X} = (X_1, \ldots, X_n)$ with parent distribution F_X. Let $\underline{F}(\cdot; \mathbf{X})$ and $\overline{F}(\cdot; \mathbf{X})$ be two functions computed from $\mathbf{X}$ and such that $\underline{F}(\cdot; \mathbf{X}) \leq \overline{F}(\cdot; \mathbf{X})$. The distribution band $\Gamma_X(\underline{F}(\cdot; \mathbf{X}), \overline{F}(\cdot; \mathbf{X}))$ is called a *confidence band at level* $\alpha \in (0,1)$ [72, page 334] iff

$$\mathbb{P}\left\{\underline{F}(x; \mathbf{X}) \leq F_X(x) \leq \overline{F}(x; \mathbf{X}),\ \forall x \in \mathbb{R}\right\} = 1 - \alpha, \tag{2.2}$$

or, equivalently:

$$\mathbb{P}\left\{P_X \in \Gamma_X\left(\underline{F}(\cdot; \mathbf{X}), \overline{F}(\cdot; \mathbf{X})\right)\right\} = 1 - \alpha. \tag{2.3}$$

Note that, in the above equalities, F_X and P_X are fixed unknown functions, whereas $\underline{F}(\cdot; \mathbf{X})$ and $\overline{F}(\cdot; \mathbf{X})$ depend on random sample $\mathbf{X}$.

Kolmogorov's Confidence Band

An entirely non-parametric confidence band on a sample's cumulated distribution function of a statistic X can be built through Kolmogorov's statistic D_n. The value of D_n represents the supremum of the difference between the estimated and actual cdf at a confidence level α and is defined as

$$D_n = \sup_x |S_n(x; \mathbf{X}) - F_X(x)|, \tag{2.4}$$

where $S_n(\cdot; \mathbf{X})$ is the sample cumulated distribution function defined by

$$S_n(x; \mathbf{X}) = \begin{cases} 0, & x < X_{(1)} \\ k/n, & X_{(k)} \leq x < X_{(k+1)} \\ 1, & X_{(n)} \leq x, \end{cases} \tag{2.5}$$

for all $x \in \mathbb{R}$, where $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$ denote the observations sorted in increasing order.

The distribution of D_n is totally independent from the reference cdf. It only depends on the size of the set of data that was used to build the estimated cdf, and is actually inversely proportional to the sample size n. Thus, the bigger the set of data, the more precise the estimation of the cdf and the narrower the confidence band.

[1] Ferson *et al.* [50] actually used the term "p-box" as a synonym to "distribution band". However, following Kriegler and Held [68], we prefer to reserve the term "p-box" for the important case where the bounding functions are step functions.

The distribution of D_n was computed for fixed n by Kolmogorov [67], who also computed the asymptotic distribution of D_n. Let $d_{n,\alpha}$ denote the $(1-\alpha)$ percentile of D_n (defined as $\mathbb{P}(D_n > d_{n,\alpha}) = \alpha$). Thus,

$$\mathbb{P}\left\{ S_n(x;\mathbf{X}) - d_{n,\alpha} \leq F_X(x) \leq S_n(x;\mathbf{X}) + d_{n,\alpha}, \forall x \in \mathbb{R} \right\} = 1-\alpha, \tag{2.6}$$

which implies that $S_n \pm d_{n,\alpha}$ defines a confidence band at level $1-\alpha$ [63, page 481]. This band may be narrowed by using the inequalities $0 \leq F_X(x) \leq 1$ for all x. Hence,

$$\underline{F}(x;\mathbf{X}) = \max(0, S_n(x;\mathbf{X}) - d_{n,\alpha}), \tag{2.7}$$

$$\overline{F}(x;\mathbf{X}) = \min(1, S_n(x;\mathbf{X}) + d_{n,\alpha}). \tag{2.8}$$

If the support of X is bounded and known to be included in $[b, B]$, then the above bounds can be further narrowed.

Note that $S_n(\cdot;\mathbf{X})$ –as defined by (2.5)– and, consequently, both $\underline{F}(\cdot;\mathbf{X})$ and $\overline{F}(\cdot;\mathbf{X})$, are right-continuous step functions. However, $\overline{F}(\cdot;\mathbf{X})$ can be replaced by the left-continuous function $\overline{F}'(\cdot;\mathbf{X})$ taking the same values everywhere except at sample points, defined as $\overline{F}'(x;\mathbf{X}) = \lim_{h\to x^-} \overline{F}(h;\mathbf{X})$. The pair $(\underline{F}, \overline{F}')$ still defines a confidence band at level $1-\alpha$, that is to say,

$$\mathbb{P}\left\{ P_X \in \Gamma_X(\underline{F}, \overline{F}') \right\} = 1-\alpha. \tag{2.9}$$

Example 6. *The data reported in [109] consists in the operational lives (in hours) of 20 bearings. These are 2398, 2812, 3113, 3212, 3523, 5236, 6215, 6278, 7725, 8604, 9003, 9350, 9460, 11584, 11825, 12628, 12888, 13431, 14266, 17809. Here, the variable of interest, denoted X (the lifetime of a bearing), has a lower bound $b = 0$ and no upper bound ($B = \infty$). Figure 2.1 shows the sample cdf of this data, together with the lower and upper bounding functions defining the Kolmogorov confidence band at level $1-\alpha = 0.95$.*

Nevertheless, Kolmogorov's confidence bands lead to a pair of step functions, which are obviously not the best confidence bounds that can be set around a continuous cdf. Furthermore, Kolmogorov's statistic is well known to be very conservative, and thus leads to wider confidence bands than necessary.

Various authors [23, 108, 91, 73, 59, 62, 24] provide ways of building what we may term *"continuous confidence band"* (i.e. confidence bands without points of discontinuity) with good properties (for the goodness of a confidence band see [60]). The construction of non-parametric continuous confidence bands may be based, e.g., on bootstrap prediction [24, 4, 60].

However, when reasonable assumptions can be made about the distribution of the data, parametric confidence bands are generally narrower and thus less conservative than non parametric confidence bands.

Cheng and Iles' Parametric Confidence Bands

Methods for the construction of *parametric* continuous confidence bands were proposed by several authors, including Kanofsky and Srinivasan [61] and Cheng and Iles [23].

Of particular interest is the solution provided by Cheng and Iles [23] for the construction of confidence bands around functions of the location-scale family of mean μ and standard deviation σ. Their method, which will be used later to demonstrate the main findings of this section, will briefly be recalled in the sequel.

Let us assume that X is a continuous random variable with cdf $F_X(x,\theta)$, where θ is a vector of r unknown parameters. Cheng and Iles' approach consists in determining lower and upper bounds of the cdf when θ varies in a confidence region R. This confidence region is built from the statistics

$$Q(\theta) = (\widehat{\theta} - \theta)^T I(\theta)(\widehat{\theta} - \theta), \tag{2.10}$$

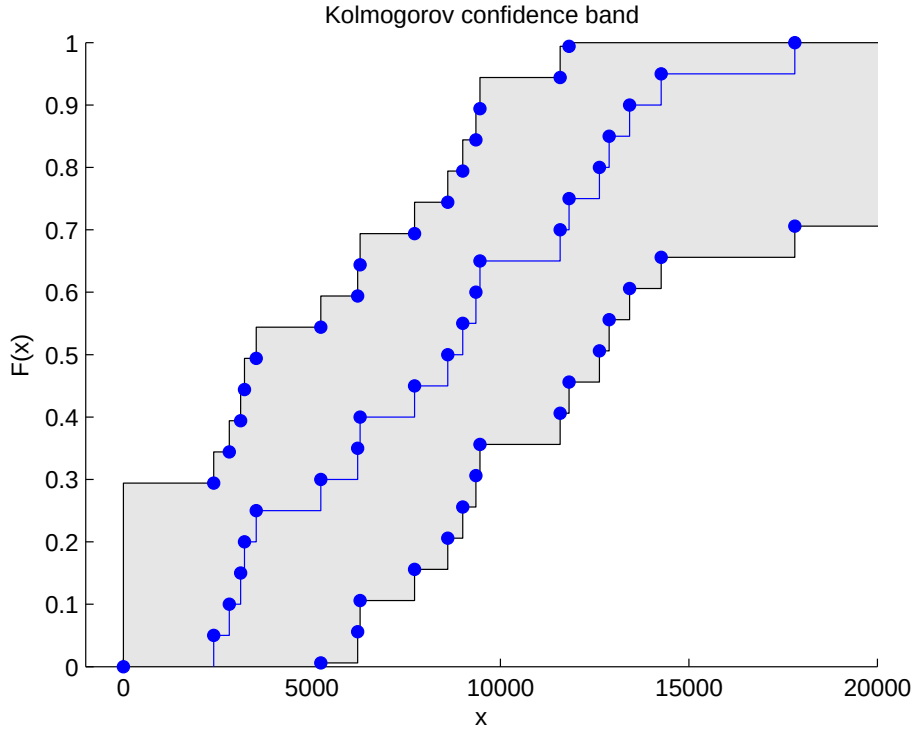


Figure 2.1: Sample cdf S_n and Kolmogorov confidence band at level $1-\alpha = 0.95$ for the bearings data.

where $\widehat{\theta}$ is the maximum likelihood estimate of θ, and $I(\theta)$ is Fisher's information matrix. It is known that $Q(\theta)$ is asymptotically a chi-squared variable with r degrees of freedom. In [23], Cheng and Iles apply their method in the case of a general location-scale parametric model of the form:

$$F_X(x) = G\left(\frac{x-\mu}{\sigma}\right), \tag{2.11}$$

where G is a fixed distribution function, and μ and σ are the unknown location and scale parameters. In that case Fisher's information matrix is of the form

$$I(\mu,\sigma) = \frac{n}{\sigma^2}\begin{pmatrix} k_0 & -k_1 \\ -k_1 & k_2 \end{pmatrix}, \tag{2.12}$$

where k_0, k_1 and k_2 are constants independent of μ and σ. The bounds of the confidence band may then be expressed as follows:

$$\overline{F}(x) = G(\xi + h), \tag{2.13}$$
$$\underline{F}(x) = G(\xi - h), \tag{2.14}$$

where $\xi = (x-\widehat{\mu})/\widehat{\sigma}$, $\widehat{\mu}$ and $\widehat{\sigma}$ are the maximum likelihood estimates of μ and σ, and

$$h = \sqrt{\frac{\gamma}{n\,k_0}\left(1 + \frac{(k_0\xi + k_1)^2}{k_0 k_2 - k_1^2}\right)}. \tag{2.15}$$

Coefficient γ is the value for which $\mathbb{P}(Q(\mu,\sigma) \leq \gamma) = 1-\alpha$. It can be approximated by the chi-squared quantile $\chi^2_2(\alpha)$. Cheng and Iles [23] demonstrated the application of these formula for the cases of the normal, lognormal, extreme-value (log-Weibull) and Weibull distributions. In the case of the normal distribution, $k_0 = 1$, $k_1 = 0$, and $k_2 = 2$.

Example 7. *This method was applied to the bearings data of examples 6 and 8 for computing a continuous predictive belief function. As in [23], we assumed these data have a lognormal distribution. Figure 2.2 shows the 95 % confidence band and the estimated cdf.*

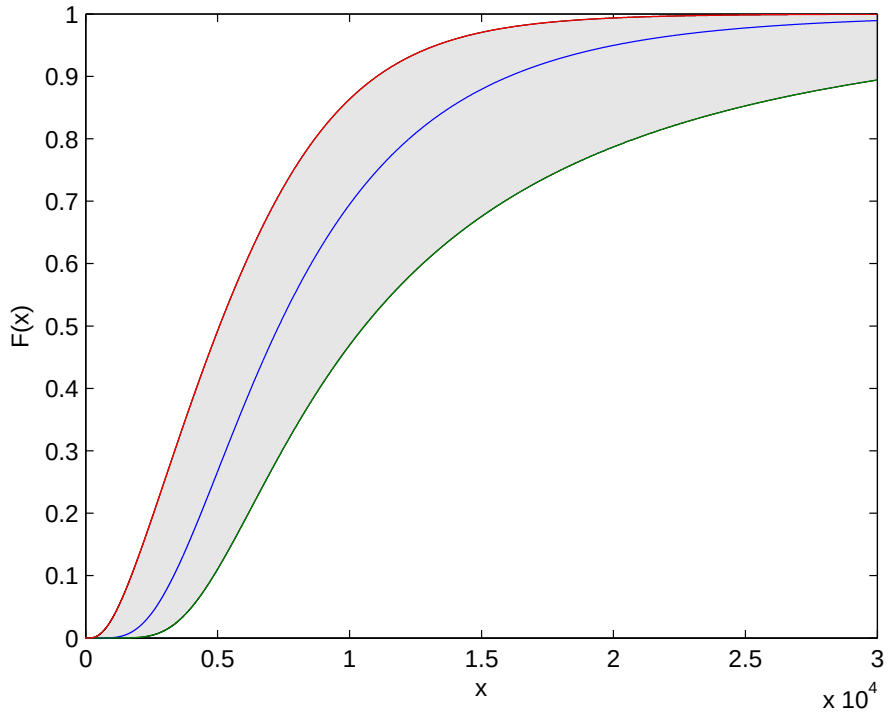


Figure 2.2: Continuous confidence band and cumulative density function estimated through Cheng and Iles' algorithm.

We will first describe how to build a discrete belief function from multinomial confidence intervals. We will then introduce a method for building discrete PBF on $\mathbb{R}$, based on Kolmogorov's confidence bands, and a method to built continuous PBF on $\mathbb{R}$, based on Chend and Iles' confidence bands.

2.2.2 Discrete predictive belief function on a discrete domain

In [41], Denœux provides a solution to the predictive belief function problem (defined in Section 1.4) in the case of discrete random variables, based on Goodman's simultaneous confidence intervals, which we simply recall in this section. He suggests multinomial confidence regions be used for the building of a belief function fulfilling requirements (1.66) and (1.67).

Given an iid sample $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ of a discrete random variable X taking values in $\mathcal{X} = \{\xi_1, \cdots, \xi_K\}$, let $N_k = \sum_{i=1}^{n} \ _{\xi_k}(\mathbf{x}_i)$ denote the number of observations in category ξ_k. The random vector $\mathbf{N} = (N_1, \cdots, N_K)$ has a multinomial distribution with parameters n and $\mathbf{p} = (p_1, \cdots, p_K)$, with $p_k = \mathbb{P}(\xi_k)$.

Additionnally, let $S(\mathbf{N})$ be a random subset of the parameter space

$$\theta = \left\{ \mathbf{p} = (p_1, \cdots, p_K) \in [0,1]^K \text{such that } \sum_{k=1}^{K} p_k = 1 \right\}. \tag{2.16}$$

$S(\mathbf{N})$ is said to be a confidence region for $\mathbf{p}$ at confidence level $(1-\alpha)$, if $\mathbb{P}(S(\mathbf{N}) \ni \mathbf{p}) \geq 1-\alpha$.

Such a confidence region $S(\mathbf{N})$ may be seen as defining either a set of plausible values for the vector parameter $\mathbf{p}$, either a family of probability measures, given that each value of $\mathbf{p}$ specifies a unique probability measure of $\mathcal{X}$.

Let $\overline{P}$ and $\underline{P}$ denote, respectively, the upper and lower envelopes of $S(\mathbf{N}) = [\underline{P}_1; \overline{P}_1] \times [\underline{P}_2; \overline{P}_2] \times \cdots \times [\underline{P}_K; \overline{P}_K]$. $\underline{P}$ and $\overline{P}$ can be computed using:

$$\underline{P}(A) = \max\left(\sum_{\xi_k \in A} \underline{P}_k, 1 - \sum_{\xi_k \notin A} \overline{P}_k\right);\ \overline{P}(A) = \min\left(\sum_{\xi_k \in A} \overline{P}_k, 1 - \sum_{\xi_k \notin A} \underline{P}_k\right). \tag{2.17}$$

Denœux proved that $\underline{P}$ satisfies requirements (1.66) and (1.67) and is a predictive belief function for X in the cases where $K = 2$ or $K = 3$. On the other hand, if $K > 3$, $\underline{P}$ may not be a belief function. It is therefore necessary to look for the most committed belief function amongst those less committed than $\underline{P}$. Any function *bel* solution of

$$\max_{m^{\mathcal{X}}}\left(\sum_{A \subseteq \mathcal{X}} bel^{\mathcal{X}}(A)\right) = \max_{m^{\mathcal{X}}}\left(2^k \sum 2^{-|B|} m^{\mathcal{X}}(B)\right) \tag{2.18}$$

under constraints:

$$\sum_{B \subseteq A} m^{\mathcal{X}}(B) \leq P^-(A),\ \forall A \subset \mathcal{X}, \tag{2.19}$$

$$\sum_{A \subseteq \mathcal{X}} m^{\mathcal{X}}(A) = 1, \tag{2.20}$$

$$\text{and } m^{\mathcal{X}}(A) \geq 0,\ \forall A \subseteq \mathcal{X}, \tag{2.21}$$

satisfies requirements (1.66) and (1.67), [41]. This solution is valid for any number of cases ($K \geq 2$) with the drawback that both the numbers of variables and constraints rapidly grow with K.

2.2.3 Discrete predictive belief functions on $\mathbb{R}$

We will now address the construction of a discrete predictive belief function from a step confidence band.

Predictive Belief Function Induced by a Kolmogorov Confidence Band The method described in Section 2.2.1 for constructing a confidence band yields a pair of lower and upper step functions, i.e., a p-box. The relationship between p-boxes and belief functions has been studied by several authors [136, 50, 130]. Recently, the exact correspondence between p-boxes with bounded support and discrete belief functions was proved by Kriegler and Held [68], who also proposed an algorithm for the rigorous construction of a discrete mass function m on $\mathbb{R}$ equivalent to a p-box.

Let the bounding step functions $\underline{F}$ and $\overline{F}$ be defined as follows :

$$\underline{F}(x) = \begin{cases} \underline{F}(x_{*i}) & x_{*i} \leq x < x_{*i+1} \\ 0 & x < x_{*1} \\ 1 & x_{*n} \leq x \end{cases} \quad \overline{F}(x) = \begin{cases} \overline{F}(x^*_{j+1}) & x^*_j < x \leq x^*_{j+1} \\ 0 & x \leq x^*_1 \\ 1 & x^*_m < x \end{cases}, \tag{2.22}$$

where the x_{*i} are the points of discontinuity of $\underline{F}$ sorted in increasing order so that $x_{*1} \leq x_{*2} \leq \ldots \leq x_{*n}$ and the x^*_j are the points of discontinuity of $\overline{F}$ sorted in increasing order so that $x^*_1 \leq x^*_2 \leq \ldots \leq x^*_m$.

The algorithm is the following.

Algorithm 1. *Let:*

- *index k run over the focal elements of the random set to be constructed;*
- *index i run over* x_{*i}*;*
- *index j run over* x_j^**;*
- p_k *denote the cumulative probability already accounted for at step k;*
- *the tuple* $\{\xi, m\} = \{(A_1, m_1), \cdots, (A_n, m_n)\}$ *denotes the set of focal elements* $A_k = (a, b]$ *of the predictive belief function with their associated basic belief assignments.*

Iterate:

1. *Initialize:* $k = 1$, $i = 1$, $j = 1$, *and assign* $p_0 = 0$*;*
2. *Construct focal element* $A_k = (x_j^*, x_{*i}]$*;*
3. (a) *If* $j = m$ *choose arbitrary* $x_{m+1}^* > x_m^*$*, thus* $\overline{F}(x_{m+1}^*) = 1$*;*
   (b) *else if* $\underline{F}(x_{*i}) < \overline{F}(x_{j+1}^*)$*:* $m_k = \underline{F}(x_{*i}) - p_{k-1}$*,* $p_k = \underline{F}(x_{*i})$*. Raise indices* $k \to k+1$*,* $i \to i+1$*. Return to step 2.*
   (c) *else if* $\underline{F}(x_{*i}) > \overline{F}(x_{j+1}^*)$*:* $m_k = \overline{F}(x_{j+1}^*) - p_{k-1}$*,* $p_k = \overline{F}(x_{j+1}^*)$*. Raise indices* $k \to k+1$*,* $j \to j+1$*. Return to step 2.*
   (d) *else if* $\underline{F}(x_{*i}) = \overline{F}(x_{j+1}^*)$*:* $m_k = \overline{F}(x_{j+1}^*) - p_{k-1}$*.*
      i. *If* $\underline{F}(x_{*i}) = \overline{F}(x_{j+1}^*) = 1$*,stop.*
      ii. *If* $\underline{F}(x_{*i}) = \overline{F}(x_{j+1}^*) < 1$*, set* $p_k = \overline{F}(x_{j+1}^*)$*. Raise indices* $k \to k+1$*,* $j \to j+1$*,* $i \to i+1$*. Return to step 2.*

For each step k, $x_j^* \leq x_{*i}$ since $\overline{F} \leq \underline{F}$, and $m_k > 0$ since $\overline{F}$ and $\underline{F}$ are monotonly increasing. The algorithm will always reach points x_{*n}, x_{m+1}^* with $\underline{F}(x_{*n}) = \overline{F}(x_{m+1}^*) = 1$ and stop, thus returning the least-committed belief function associated with the confidence region defined by $\underline{F}$ and $\overline{F}$ (this property is demonstrated in [68]).

The principle of this construction is illustrated in Figure 2.3. The lower and upper bounding functions are assumed to be right and left continuous, respectively. Each rectangle A_i in this figure corresponds to a focal interval $[a_i, b_i)$, with mass $m(a_i, b_i) = d_i - c_i$.

Let $\Gamma_X(bel)$ denote the set of probability measures compatible with bel, the belief function induced by m, i.e.,

$$\Gamma_X(bel) = \{P | bel(A) \leq \mathbb{P}(A), \forall A \in \mathcal{B}(\mathbb{R})\}. \tag{2.23}$$

Kriegler and Held [68] proved that $(\underline{F}, \overline{F})$ and bel are two equivalent representations of a unique family of probabilities, i.e.,

$$\Gamma_X(bel) = \Gamma_X(\underline{F}, \overline{F}). \tag{2.24}$$

If bel and pl denote the corresponding belief and plausibility functions, and if $\underline{P}$ and $\overline{P}$ denote the lower and upper envelopes of $\Gamma_X(\underline{F}, \overline{F})$, then $bel = \underline{P}$ and $pl = \overline{P}$. In particular, $bel((-\infty, x]) = \underline{F}(x)$ and $pl((-\infty, x]) = \overline{F}(x)$ for all $x \in \mathbb{R}$.

Note that, although Kriegler and Held only considered the case of p-boxes with bounded support, their algorithm and results may be applied directly to the case of p-boxes with unbounded support.

Let us now consider the case where $\underline{F}$ and $\overline{F}$ are the lower and upper bounding functions of Kolmogorov confidence band at level $1 - \alpha$, as defined by (2.7)-(2.8). Let $bel(\cdot; \mathbf{X})$ denote the belief function on $\mathbb{R}$ constructed from p-box $(\underline{F}, \overline{F}')$ using Kriegler and Held's algorithm. The following proposition holds.

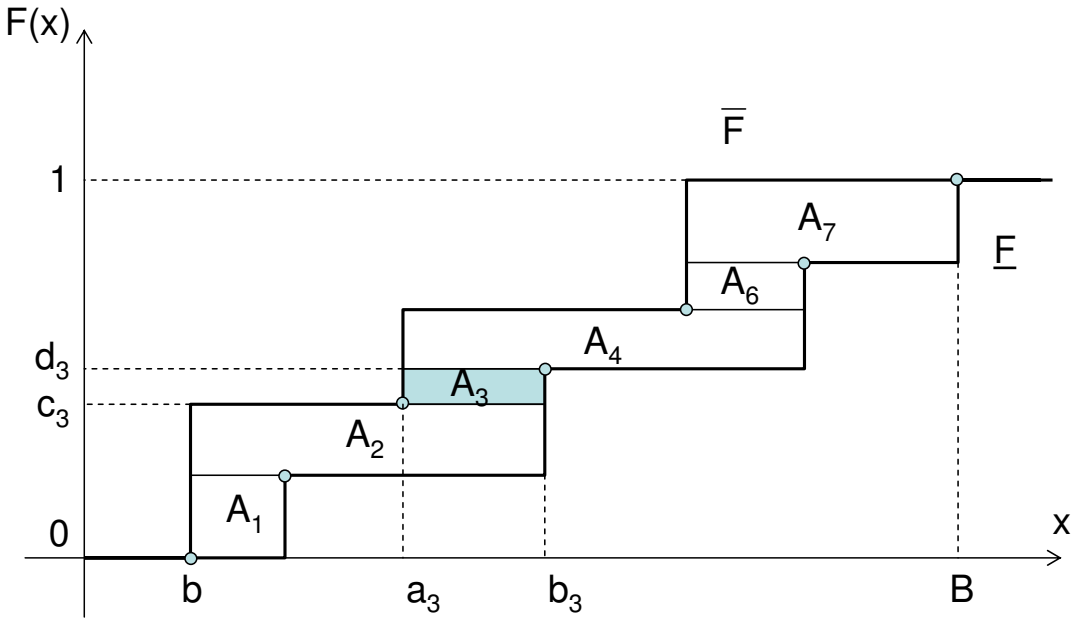


Figure 2.3: Principle of the construction of a basic belief assignment from a p-box.

Proposition 1. *$bel(\cdot;\mathbf{X})$ is a predictive belief function at level $1-\alpha$.*

Proof. We need to show that $bel(\cdot;\mathbf{X})$ satisfies requirements (1.66) and (1.67).

First, requirement (1.67) is obviously satisfied as a direct consequence of (2.9) and (2.24): since $\Gamma_X(bel(\cdot;\mathbf{X})) = \Gamma_X(\underline{F}, \overline{F}')$,

$$P\left\{bel(A;\mathbf{X}) \leq P_X(A), \forall A \in \mathcal{A}\right\} = P\left\{P_X \in \Gamma_X(bel(\cdot;\mathbf{X}))\right\} = 1-\alpha. \tag{2.25}$$

Let us now consider requirement (1.66).

We know that $S_n(x) \xrightarrow{P} F_X(x)$ for all $x \in \mathbb{R}$, as $n \to \infty$ and $\lim_{n\to\infty} d_{n,\alpha} = 0$. Consequently, $\underline{F}(x) \xrightarrow{P} F_X(x)$ and $\overline{F}'(x) \xrightarrow{P} F_X(x)$ for all $x \in \mathbb{R}$, as $n \to \infty$.

Now, as a consequence of Proposition 2.24,

$$bel(A;\mathbf{X}) = \inf_{P \in \Gamma_X(\underline{f}, \overline{f}')} P(A), \quad \forall A \in \mathcal{B}(\mathbb{R}). \tag{2.26}$$

Let us show that $bel(A;\mathbf{X}) \xrightarrow{P} P_X(A)$ for all interval A:

- $bel(\cdot;\mathbf{X})((-\infty, x]) = \underline{f}(x) \xrightarrow{P} F_X(x)$ for all $x \in \mathbb{R}$;
- $bel(\cdot;\mathbf{X})((x, +\infty)) = 1 - pl(\cdot;\mathbf{X})((-\infty, x]) = 1 - \overline{f}(x) \xrightarrow{P} 1 - F_X(x) = P_X((x, +\infty))$;
- $bel(\cdot;\mathbf{X})((x, y]) = \max(0, \underline{f}(y) - \overline{f}'(x)) \xrightarrow{P} F_X(y) - F_X(x) = P_X((x, y])$, for all $x, y \in \mathbb{R}$, $x < y$;
- $bel(\cdot;\mathbf{X})((x, y)) < bel(\cdot;\mathbf{X})((x, y])$ only if $y = x_i$ for some sample point x_i; as this event has probability zero, $bel(\cdot;\mathbf{X})((x, y)) \overset{a.s.}{=} bel(\cdot;\mathbf{X})((x, y])$ (where $\overset{a.s.}{=}$ denotes almost sure equality) and, consequently, $bel(\cdot;\mathbf{X})((x, y)) \xrightarrow{P} P_X((x, y))$;
- Similarly, $bel(\cdot;\mathbf{X})((-\infty, y)) \overset{a.s.}{=} bel(\cdot;\mathbf{X})((-\infty, y])$ and, consequently,

$$bel(\cdot;\mathbf{X})((-\infty, y)) \xrightarrow{P} P_X((-\infty, y)), \quad \forall y \in \mathbb{R}; \tag{2.27}$$

- By construction, no focal element of $bel(\cdot;\mathbf{X})$ can be reduced to a point; consequently:

$$\begin{aligned} bel(\cdot;\mathbf{X})([x,y]) &= bel(\cdot;\mathbf{X})((x,y]), \quad \forall x,y \in \mathbb{R}, x < y, \\ bel(\cdot;\mathbf{X})([x,+\infty))) &= bel(\cdot;\mathbf{X})((x,+\infty)), \quad \forall x \in \mathbb{R}, \\ bel(\cdot;\mathbf{X})([x,x]) &= 0 = P_X([x,x]), \quad \forall x \in \mathbb{R}. \end{aligned}$$

Now, any borel set $B \in \mathcal{B}$ can be written as $B = \bigcup_{i\in I} A_i$ for a countable family of intervals $(A_i)_{i\in I}$ with $I \subseteq \mathbb{N}$, such that $A_i \cup A_j$ is not an interval, for all $i, j \in I$. With such a decomposition,

$$bel(\cdot;\mathbf{X})(B) = \sum_{i\in I} bel(\cdot;\mathbf{X})(A_i) \xrightarrow{P} \sum_{i\in I} P_X(A_i) = P_X(B), \tag{2.28}$$

which completes the proof. □

Example 8. *In order to illustrate the construction of a predictive belief function from a Kolmogorov confidence band, let us consider again the data of Example 6. Based on this data, we would like to express our beliefs regarding the lifetime X of a new bearing taken randomly from the same population. For commodity of representation, let us adopt the reasonable assumption that X has an upper bound, which will arbitrarily be set to 30000, so that the support of X is assumed to be* $[0, 30000]$.

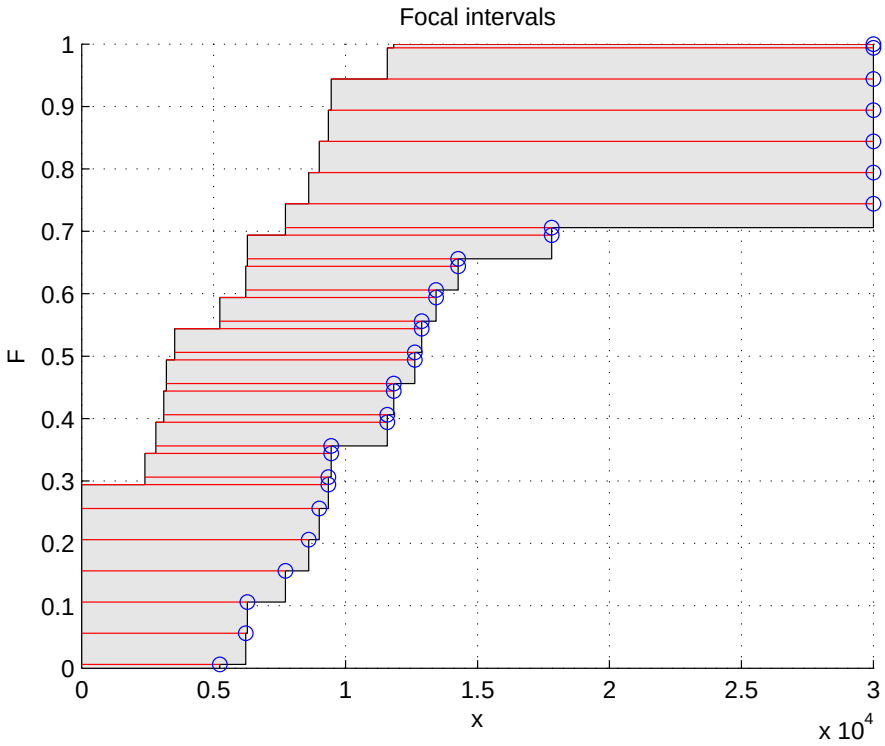


Figure 2.4: Focals intervals of the PBF constructed from the Kolmogorov confidence band at level $1 - \alpha = 0.95$ (bearings data).
The height of each segment representing a focal interval equals the cumulated mass allocated to intervals whose lower and upper bounds are, respectively, smaller than the lower and upper bounds of the considered interval.

The focal intervals of the corresponding PBF $bel(\cdot;\mathbf{X})$ *are displayed in Figure 2.4. Figures 2.5 and 2.6 are examples of graphical displays that reveal different aspects of the information contained*

in the belief function $bel(\cdot;\mathbf{X})$. *Figure 2.5 shows the plausibility profile function* $x \rightarrow pl(\{x\};\mathbf{X})$ *and the pignistic probability density function* $Betf$ *computed from (1.40), which are two left-continuous real-valued step functions with simple interpretation. Figure 2.6 shows grey level representations of* $bel([x,y];\mathbf{X})$, $pl([x,y];\mathbf{X})$ *and* $q([x,y];\mathbf{X})$ *as two-dimensional functions of* (x,y).

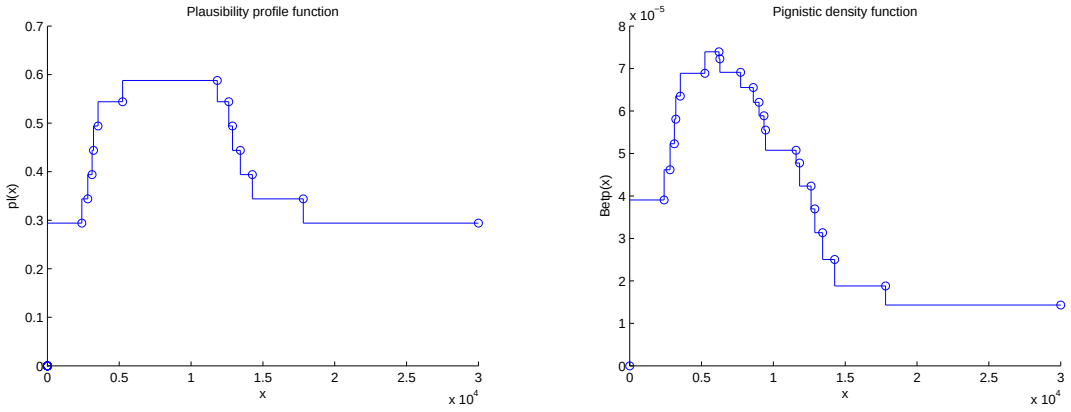


Figure 2.5: Plausibility profile function (left) and pignistic probability density function (right) of the discrete PBF constructed from the Kolmogorov confidence band (Bearings data).

Random Set Interpretation The bba m associated with a p-box $(\underline{F},\overline{F})$ may also be shown to formally correspond to a random set [3]. Let $\underline{F}^{-1}$ and $\overline{F}^{-1}$ be the pseudo-inverses of $\underline{F}$ and $\overline{F}$ defined, respectively, as:

$$\underline{F}^{-1}(\alpha) = \inf\{x \in \mathbb{R}, \underline{F}(x) \geq \alpha\}, \tag{2.29}$$

$$\overline{F}^{-1}(\alpha) = \inf\{x \in \mathbb{R}, \overline{F}(x) \geq \alpha\}, \tag{2.30}$$

for all $\alpha \in [0,1]$. Let us consider the mapping ρ from $[0,1]$ to the set of real intervals, such that $\rho(\alpha) = [\underline{F}^{-1}(\alpha), \overline{F}^{-1}(\alpha)]$, and let us consider the uniform probability distribution P_U on $[0,1]$. Then ρ is a random set, and it is formally equivalent to m. Let $\mathcal{F} = \{[\underline{F}^{-1}(\alpha), \overline{F}^{-1}(\alpha)], \alpha \in [0,1]\}$. For all $A \in \mathcal{F}$,

$$m(A) = P_U(\rho^{-1}(A)). \tag{2.31}$$

Note that the uniform probability distribution on $[0,1]$ and the mapping ρ are only considered here as mathematical constructs. In the TBM, only belief functions have an interpretation, and an underlying multi-valued mapping is not assumed. However, the random set point of view will guide us in the following section to propose a generalization of the above results in the case of continuous distribution bands.

2.2.4 Continuous predictive belief functions on $\mathbb{R}$

As already mentioned, Kolmogorov's confidence bands have the advantage of being exact and non-parametric. However, they have a constant width, which makes them unnecessarily broad in the tails. As a result, the equivalent belief functions may be excessively imprecise. Narrower confidence bands can be computed using parametric methods as shown in Section 2.2.1, but they are defined by continuous bounding functions. The usual approach to continuous distribution bands is to approximate them using

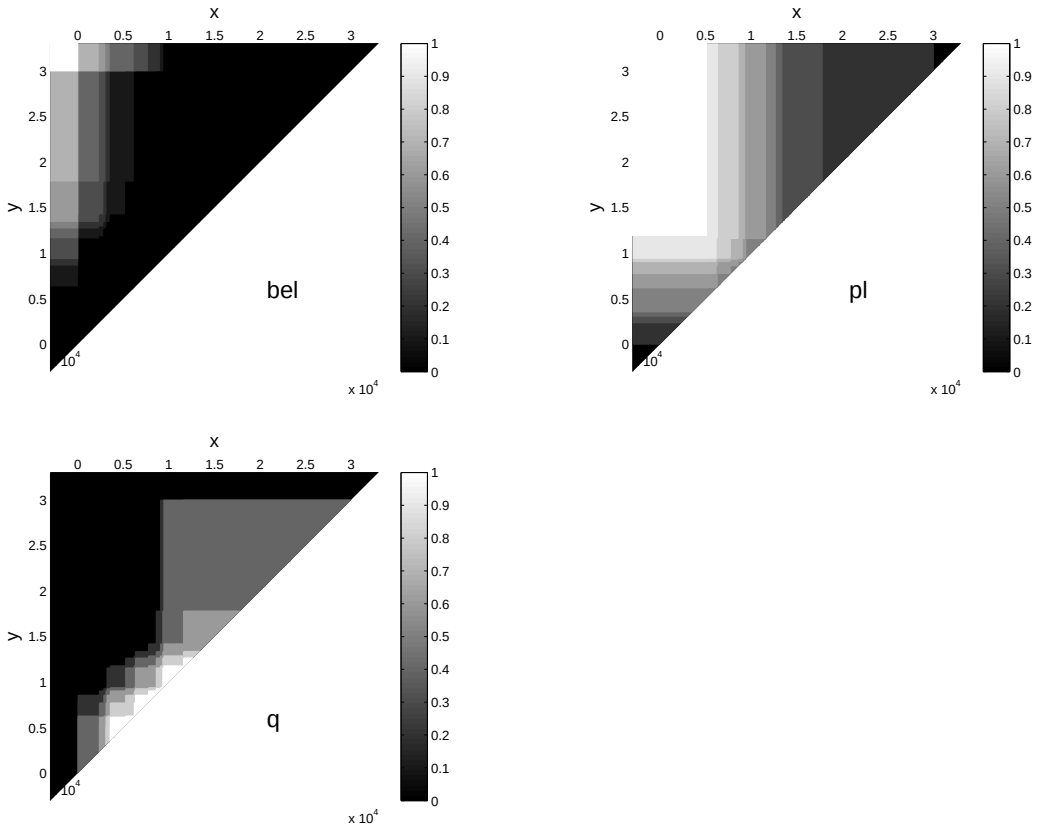


Figure 2.6: Contour plots of functions $bel(\cdot;\mathbf{X})([x,y])$, $pl(\cdot;\mathbf{X})([x,y])$ and $q(\cdot;\mathbf{X})([x,y])$ constructed from Kolomogorov's confidence band (Bearings data).

a p-box [50]. Here, we show that this approximation can be avoided, and a continuous predictive belief function on $\mathbb{R}$ can be constructed from a continuous confidence band, thus providing an extension to the results presented in the previous section.

PBF Induced by a Continuous Confidence Band

Let $\Gamma_X(\underline{F}, \overline{F})$ be a continuous distribution band for some continuous random variable X, and assume that the lower and upper bounding functions $\underline{F}$ and $\overline{F}$ are strictly increasing. Consider the mapping ρ from $[0,1]$ to the set of real intervals, such that $\rho(\alpha) = [\underline{F}^{-1}(\alpha), \overline{F}^{-1}(\alpha)]$, where $\underline{F}^{-1}$ and $\overline{F}^{-1}$ are the inverses of $\underline{F}$ and $\overline{F}$, respectively. If the $[0,1]$ interval is endowed with a uniform probability distribution, then mapping ρ defines a random set, which corresponds to a continuous belief function *bel* on $\mathbb{R}$ as described in Section 1.3.2. In other words, a continuous confidence band is formally equivalent to a continuous belief function on $\mathbb{R}$, which provides a continuous extension to the results presented in [68] and recalled in Section 2.2.3. As a consequence, a continuous confidence band constructed using, e.g., the parametric method of Cheng and Iles summarized in Section 2.2.1, is equivalent to a continuous predictive belief function.

As we are working within the TBM, the above random set is for us a purely mathematical construct, and we would like to express *bel* directly through its bbd $m(x,y)$, $x \leq y$. Hence, we will show that this belief function is such that $bel([x,y]) = \underline{P}([x,y])$ for all $x \leq y$, $\underline{P}$ being the lower envelope of the distribution band. This can be achieved using (1.54). However, it requires a little preparation work.

Let $\Gamma_X(\underline{F}, \overline{F})$ be a continuous distribution band for some continuous r.v. X, and let $\underline{P}$ denote its lower envelope defined as:

$$\underline{P}(A) = \inf_{P \in \Gamma_X(\underline{F},\overline{F})} P(A), \quad \forall A \in \mathcal{B}(\mathbb{R}). \tag{2.32}$$

We want to show that $\underline{P}$ is a belief function. For that purpose, let us start with the following lemma.

Lemma 1.

$$\begin{aligned} \frac{\partial^2 \underline{P}([x,y])}{\partial x \partial y} &= -\overline{f}(x)\underline{f}(y)\delta(\underline{F}(y) - \overline{F}(x)), & (2.33)\\ &= -\overline{f}(x)\delta(y - \underline{F}^{-1} \circ \overline{F}(x)), & (2.34)\\ &= -\underline{f}(y)\delta(x - \overline{F}^{-1} \circ \underline{F}(y)), & (2.35) \end{aligned}$$

where $\underline{f}$ and $\overline{f}$ are the first derivatives of $\underline{F}$ and $\overline{F}$, respectively, and δ is the Dirac delta function.

Proof. By definition,

$$\begin{aligned} \underline{P}([x,y]) &= \max(0, \underline{F}(y) - \overline{F}(x)) \\ &= (\underline{F}(y) - \overline{F}(x))H(\underline{F}(y) - \overline{F}(x)), \end{aligned}$$

where H is the Heavyside function. Consequently,

$$\frac{\partial \underline{P}([x,y])}{\partial x} = -\overline{f}(x)\left(H(\underline{F}(y) - \overline{F}(x)) + (\underline{F}(y) - \overline{F}(x))\delta(\underline{F}(y) - \overline{F}(x))\right), \tag{2.36}$$

and

$$\frac{\partial^2 \underline{P}([x,y])}{\partial x \partial y} = -\overline{f}(x)\Big(\delta(\underline{F}(y) - \overline{F}(x))\underline{f}(y) + \underline{f}(y)\delta(\underline{F}(y) - \overline{F}(x)) + (\underline{F}(y) - \overline{F}(x))\delta'(\underline{F}(y) - \overline{F}(x))\underline{f}(y)\Big). \tag{2.37}$$

Now, from the property of the delta function: $x\delta'(x) = -\delta(x),\ \forall x$, we get:

$$(\underline{F}(y) - \overline{F}(x))\delta'(\underline{F}(y) - \overline{F}(x)) = -\delta'(\underline{F}(y) - \overline{F}(x)). \tag{2.38}$$

Consequently, (2.37) is equivalent to (2.33).

In order to prove that (2.34) and (2.35) can be deduced from (2.33), we shall use the following property of the delta function: For all function g,

$$\delta\left(g(x)\right) = \sum_i \frac{\delta(x - x_i)}{|g'(x_i)|}, \tag{2.39}$$

where the x_i are the roots of g. For fixed x, $(\underline{F}(y) - \overline{F}(x))$ is a function of y with a unique root $(\underline{F}^{-1} \circ \overline{F}(x))$. Hence,

$$\begin{aligned}
\overline{f}(x)\underline{f}(y)\delta(\underline{F}(y) - \overline{F}(x)) &= \overline{f}(x)\underline{f}(y)\frac{\delta(y - \underline{F}^{-1} \circ \overline{F}(x))}{\underline{f}(\underline{F}^{-1} \circ \overline{F}(x))} \\
&= \begin{cases} 0, & \text{if } y \neq \underline{F}^{-1} \circ \overline{F}(x) \\ \overline{f}(x)\delta(y - \underline{F}^{-1} \circ \overline{F}(x)), & \text{if } y = \underline{F}^{-1} \circ \overline{F}(x), \end{cases} \\
&= \overline{f}(x)\delta(y - \underline{F}^{-1} \circ \overline{F}(x)).
\end{aligned}$$

Equation (2.35) can be deduced from (2.33) in a similar way, by fixing y and treating $(\underline{F}(y) - \overline{F}(x))$ as a function of x.

□

As a consequence of (1.54), Lemma 1 tells us that, if $\underline{P}$ is a belief function, the corresponding bbd should be

$$m(x, y) = \overline{f}(x)\underline{f}(y)\delta(\underline{F}(y) - \overline{F}(x)). \tag{2.40}$$

The following proposition states that this is actually the case. (Additionally, it can be checked that $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} m(x; y)dxdy = 1$ hence m is a bbd).

Proposition 2. *The lower envelope $\underline{P}$ of a continuous confidence band $\Gamma_X(\underline{F}, \overline{F})$ is a continuous belief function with basic belief density*

$$\begin{aligned}
m(x, y) &= \overline{f}(x)\underline{f}(y)\delta(\underline{F}(y) - \overline{F}(x)) && (2.41)\\
&= \overline{f}(x)\delta(y - \underline{F}^{-1} \circ \overline{F}(x)), && (2.42)\\
&= \underline{f}(y)\delta(x - \overline{F}^{-1} \circ \underline{F}(y)), && (2.43)
\end{aligned}$$

Proof. Let us first show that $\underline{P}(A) = bel(A)$ for all interval A. First, consider the case of a closed interval $A = [x, y]$.

By definition, the belief function associated with bba (2.42) is:

$$\begin{aligned}
bel([x, y]) &= \int_{u=x}^{u=y}\int_{v=u}^{v=y} \underline{f}(u)\delta(v - \underline{F}^{-1} \circ \overline{F}(u))dvdu && (2.44)\\
&= \int_{u=x}^{u=y} \underline{f}(u)\left(\int_{v=u}^{v=y} \delta(v - \underline{F}^{-1} \circ \overline{F}(u))dv\right)du. && (2.45)
\end{aligned}$$

The integral with respect to v equals

$$\int_{v=u}^{v=y} \delta(v - \underline{F}^{-1} \circ \overline{F}(u))dv = \begin{cases} 1, & \text{if } \underline{F}^{-1} \circ \overline{F}(u) \leq y \Leftrightarrow u \leq \overline{F}^{-1} \circ \underline{F}(y), \\ 0, & \text{otherwise.} \end{cases} \tag{2.46}$$

Consequently,

$$bel([x,y]) = \begin{cases} \int_{u=x}^{u=\overline{F}^{-1}\circ\underline{F}(y)} \underline{f}(u)du & \text{if } \overline{F}^{-1}\circ\underline{F}(y) \geq x \Leftrightarrow \underline{F}(y) \geq \overline{F}(x), \\ 0, & \text{otherwise.} \end{cases} \quad (2.47)$$

$$= \max(0, \underline{F}(y) - \overline{F}(x)) \quad (2.48)$$

$$= \underline{P}([x,y]). \quad (2.49)$$

By letting x tend to $-\infty$, we get $bel((-\infty,y]) = \underline{F}(y) = \underline{P}((-\infty,y])$. Similarly, by letting y tend to $+\infty$, we get $bel([x,+\infty]) = 1 - \overline{F}(x) = \underline{P}([x,+\infty))$. It can easily be checked that the equality $bel(A) = \underline{P}(A)$ holds for any half-closed or open interval A. For instance,

$$bel((x,y)) = \int_{u=x^+}^{u=y^-} \int_{v=u}^{v=y^-} \underline{f}(u)\delta(v - \underline{F}^{-1}\circ\overline{F}(u))dvdu \quad (2.50)$$

$$= \int_{u=x^+}^{u=y^-} \underline{f}(u) \left(\int_{v=u}^{v=y^-} \delta(v - \underline{F}^{-1}\circ\overline{F}(u))dv \right) du, \quad (2.51)$$

and

$$\int_{v=u}^{v=y^-} \delta(v - \underline{F}^{-1}\circ\overline{F}(u))dv = \begin{cases} 1, & \text{if } \underline{F}^{-1}\circ\overline{F}(u) < y \Leftrightarrow u < \overline{F}^{-1}\circ\underline{F}(y), \\ 0, & \text{otherwise.} \end{cases} \quad (2.52)$$

Consequently,

$$bel((x,y)) = \begin{cases} \int_{u=x^+}^{u=\overline{F}^{-1}\circ\underline{F}(y)^-} \underline{f}(u)du & \text{if } \overline{F}^{-1}\circ\underline{F}(y) \geq x \Leftrightarrow \underline{F}(y) \geq \overline{F}(x), \\ 0, & \text{otherwise.} \end{cases} \quad (2.53)$$

$$= \max(0, \underline{F}(y) - \overline{F}(x)) \quad (2.54)$$

$$= \underline{P}([x,y]). \quad (2.55)$$

□

It can be checked that (2.48) may be recovered from $m(x,y)$ using (1.51). Similarly, the expressions of $pl([x,y])$ and $q([x,y])$ can be obtained from $m(x,y)$ using (1.52) and (1.53). The following proposition holds.

Proposition 3. *Let m be the bbd associated with a continuous distribution band $(\underline{F},\overline{F})$. The belief, plausibility and commonality of any real interval $[x,y]$ are given by:*

$$bel([x,y]) = \max(0, \underline{F}(y) - \overline{F}(x)), \quad (2.56)$$

$$pl([x,y]) = \overline{F}(y) - \underline{F}(x), \quad (2.57)$$

$$q([x,y]) = \max(0, \overline{F}(x) - \underline{F}(y)). \quad (2.58)$$

Proof. The proof of (2.56) is given by (2.44) to (2.48). Let us prove (2.58).

$$q([x,y]) = \int_{-\infty}^{x} \int_{y}^{+\infty} m(u,v)dvdu$$

$$= \int_{-\infty}^{x} \overline{F}(u)I(u)du,$$

with

$$I(u) = \int_{y}^{+\infty} \delta\left(v - \underline{F}^{-1}\circ\overline{F}(u)\right) dv. \quad (2.59)$$

Now, $I(u) = 1$ if $\underline{F}^{-1} \circ \overline{F}(u) \geq y$, i.e., if $u \geq \overline{F}^{-1} \circ \underline{F}(y)$, and 0 otherwise. Hence $q([x,y]) = 0$ if $\overline{F}^{-1} \circ \underline{F}(y) \geq x$, i.e., if $\underline{F}(y) \geq \overline{F}(x)$; otherwise,

$$q([x,y]) = \int_{\overline{F}^{-1} \circ \underline{F}(y)}^{x} \overline{F}(u)du = \overline{F}(x) - \underline{F}(y). \tag{2.60}$$

The proof of (2.57) is similar. □

In addition to this formal proof, a fairly intuitive justification of expressions (2.48), (2.57) and (2.58) may be found in Appendix B.

Remark 7. *From (2.56) and (2.57), it can be checked that* $bel((-\infty, x]) = \underline{F}(x)$ *and* $pl((-\infty, x]) = \overline{F}(x)$*, for all* $x \in \mathbb{R}$.

Finally, the expression of the pignistic probability density associated with bbd m is given by the following proposition.

Proposition 4. *Let m be the bbd associated with a continuous distribution band $(\underline{F}, \overline{F})$. The associated pignistic probability density $Betf$ is given by*

$$Betf(x) = \int_{\overline{F}^{-1} \circ \underline{F}(x)}^{x} \frac{\overline{f}(u)}{\underline{F}^{-1} \circ \overline{F}(u) - u} du. \tag{2.61}$$

Proof. From (1.56), we get

$$Betf(x) = \lim_{\epsilon \to 0} \int_{-\infty}^{x} J(u)du, \tag{2.62}$$

with

$$\begin{aligned} J(u) &= \overline{f}(u) \int_{x+\epsilon}^{+\infty} \frac{\delta\left(v - \underline{F}^{-1} \circ \overline{F}(u)\right)}{v - u} dv \\ &= \begin{cases} \frac{\overline{f}(u)}{\underline{F}^{-1} \circ \overline{F}(u) - u} & \text{if } \underline{F}^{-1} \circ \overline{F}(u) \geq x + \epsilon \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The condition $\underline{F}^{-1} \circ \overline{F}(u) \geq x + \epsilon$ can be expressed as $u \geq \overline{F}^{-1} \circ \underline{F}(x + \epsilon)$, hence

$$\begin{aligned} Betf(x) &= \lim_{\epsilon \to 0} \int_{\overline{F}^{-1} \circ \underline{F}(x+\epsilon)}^{x} \frac{\overline{f}(u)}{\underline{F}^{-1} \circ \overline{F}(u) - u} du \\ &= \int_{\overline{F}^{-1} \circ \underline{F}(x)}^{x} \frac{\overline{f}(u)}{\underline{F}^{-1} \circ \overline{F}(u) - u} du. \end{aligned}$$

□

The above results are valid for any continuous distribution band $(\underline{F}, \overline{F})$. When $(\underline{F}, \overline{F})$ is a confidence band at level $1 - \alpha$, then it is easy to see, using the same line of reasoning as in Section 2.2.3, that the corresponding belief function is a predictive belief function at level $1 - \alpha$.

Example 9. *The plausibility profile function* $x \to pl(\{x\}; \mathbf{X})$ *obtained from the confidence band shown in example 7 is shown in Figure 2.7, and contour plots of* $bel([x,y]; \mathbf{X})$*,* $pl([x,y]; \mathbf{X})$ *and* $q([x,y]; \mathbf{X})$ *are shown in Figure 2.8. These figures should be compared to Figures 2.1, 2.5 and 2.6, respectively.*

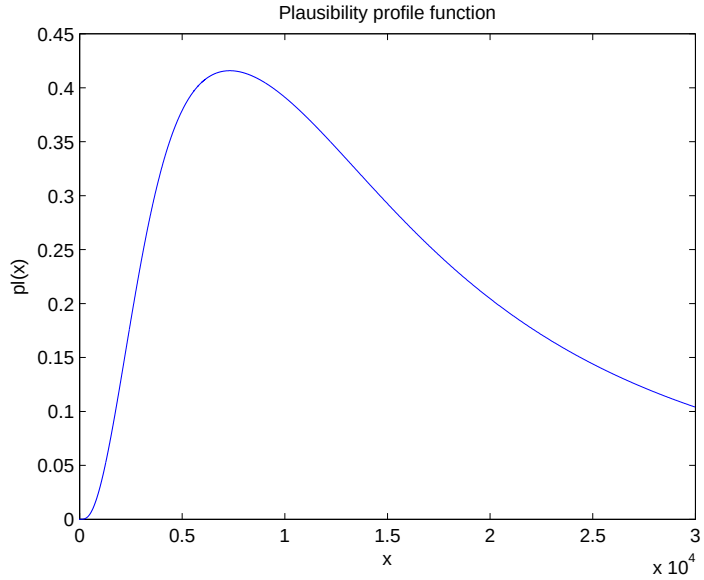


Figure 2.7: Plausibility profile function obtained from the continuous confidence band of Figure 2.2.

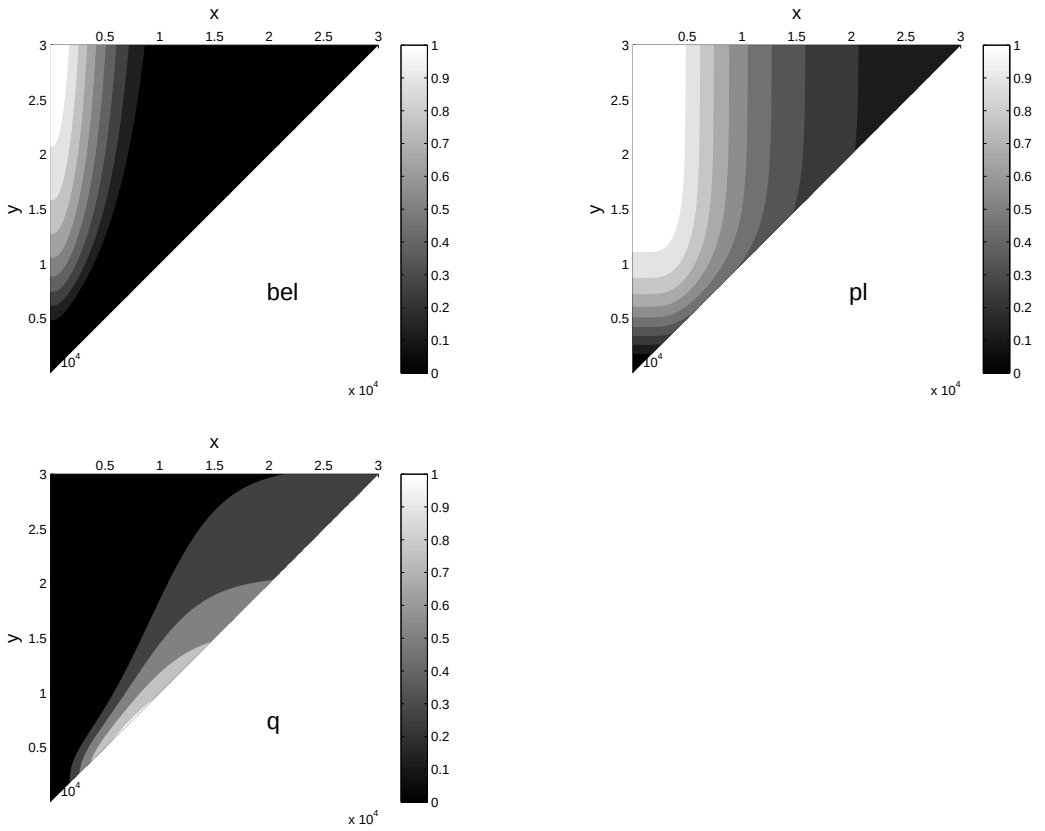


Figure 2.8: Contour plots of functions $bel([x,y];\mathbf{X})$, $pl([x,y];\mathbf{X})$ and $q([x,y];\mathbf{X})$ constructed from Cheng and Iles' confidence band.
(To be compared with Figure 2.6.)

2.2.5 Conclusion

In this section, we demonstrated how to built a predictive belief function from raw data. We first recalled Denœux's solution for the construction of a discrete predictive belief function with discrete domain [42], based on multinomial intervals. We then addressed the problem of constructing predictive belief functions as defined in [42], in the case where the random variable X is continuous. We showed that such belief functions can be constructed from confidence bands. We demonstrated that Krigler and Held's algorithm for constructing a discrete BF with a finite number of interval focal sets leads to a predictive belief function when applied to a Kolomogov confidence band. We then presented an original way of building a continuous predictive basic belief density from a continuous parametric confidence band. These belief functions are interpreted as quantifying our belief in future realizations of X, based on a realization of a random sample from the same distribution. An application of these results to classification is given at the end of this chapter, and an application to novelty detection is described in Chapter 4.

The above work is based on Hacking's frequency principle [55, 117], which equates the degree of belief of an event to its probability, when the latter is known. As mentioned in the introduction, we may require instead that a weaker form of Hacking's principle be satisfied, which states that the pignistic probability of an event should be equal to its long run frequency, when the latter is known. This other point of view will be presented in the next section.

2.3 Type II predictive belief functions

In this section, a new method for building a BF from raw data is introduced. A first approach to this problem was presented in [42] and Section 2.2 in the cases of discrete and continuous distributions, respectively, and a similar approach in the context of Possibility Theory was presented in [81]. However, the TBM [126, 122] is a two-level mental model in which the beliefs held by an agent are represented at the *credal level* by belief functions [113], whereas decision making is based on probability distributions and takes place at the *pignistic level* [124]. The new solution presented here is more in line with this two-level structure of the TBM.

More precisely, the problem considered in this section can be described as follows. Let X be a random variable with unknown probability distribution P_X. We would like to quantify the beliefs held by an agent about a future realization of X from past independent observations $X_1, \ldots, X_n$ drawn from the same distribution. In [42], it was argued that a belief function $bel(\cdot; X_1, \ldots, X_n)$ solution to this problem should meet requirements (1.66) and (1.67).

In the approach presented in [42] and in Section 2.2, the above-mentioned two requirements are derived from Hacking's frequency principle [55, 117], which equates the degree of belief of an event to its probability (long run frequency), when the latter is known. The relevance of Hacking's principle, however, can be questioned. For instance, consider the result X of a coin-tossing experiment, with $X \in \{H, T\}$, where H and T stand for "Head" and "Tail", respectively. If the coin is known to be perfectly balanced, then $P_X(\{H\}) = P_X(\{T\}) = 0.5$. If asked about our opinion regarding the result of the next toss, should we necessarily assign a degree of belief 0.5 to the event that this toss will bring a "Head"? This requirement seems difficult to justify. However, if we are forced to bet on the result of this random experiment, then it seems reasonable to assign equal odds to the two elementary events.

In the TBM, degrees of chance are not equated with degrees of belief: as emphasized above, decision making is assumed to be handled at the *pignistic level*, which is distinguished from the *credal level* at which beliefs are entertained [126, 124]. The pignistic transformation converts each belief function *bel* into a *pignistic* probability distribution *BetP* that is used for decision making. As a consequence, the use of Hacking's principle may be replaced by the weaker requirement that *the pignistic probability of an event be equal to its long run frequency, when the latter is known*. Coming back to the coin example, this requirement leads to the constraint $BetP(\{H\}) = BetP(\{T\}) = 0.5$, which defines a set of admissible belief functions. Within this set, the Least Commitment Principle (LCP) [116] dictates that the least committed one (i.e., the least informative) should be chosen, which leads here to the vacuous belief function.

In the above coin-tossing example, the probability distribution of X is assumed to be known. In this section, we consider a more realistic situation, where only partial information is available about this distribution, in the form of a random sample $X_1, \ldots, X_n$. In that case, it is possible to construct a set $\mathcal{P}$ of possible probability distributions for X defined, e.g., by a parametric confidence region. A natural extension of the line of reasoning suggested in [42] is then to require that *bel* be less committed than any belief function whose pignistic probability distribution is in $\mathcal{P}$. This leads to the definition of a set of admissible belief functions, among which the most committed can be chosen. This is the principle of the approach presented in this section.

The rest of this section is organized as follows. First, the proposed approach will be formalized. It will then be applied to the case of a discrete r.v., and to continuous parametric models. In particular, the exponential and normal distributions will be treated.

2.3.1 Consonant Belief Function Induced by a Set of Pignistic Probabilities

In the case where the pignistic distribution is know exactly, the solution was given by Dubois, Prade and Smets in the discrete case (see Equation (1.41) in Section 1.2.3), and by Smets in the continuous case (see Equation (1.58) in Section 1.3.5).

What if the pignistic probability is not known exactly? Let us suppose we only dispose of a set of realizations of a random variable drawn from the pignistic probability distribution or density, and we would like to calculate the least-committed belief function associated with the pignistic probability described by these observations. Our problem then decomposes in two subproblems :

1. The set $\mathcal{P}$ of admissible pignistic probability distribution underlying the variables first needs to be determined;

2. then, the associated least committed belief function should be deduced from $\mathcal{P}$. However, $\mathcal{P}$ does not necessarily have a unique LC element. Consequently, the most committed element of the set of BF less committed than those in $\mathcal{P}$ should be selected.

Let us assume that the pignistic probability distribution P_0 of an agent is only known to belong to a set $\mathcal{P}$ of probability distributions and we seek to approximate the agent's bba m_0. The problem is underdetermined, as we can only say that m_0 belongs to the set $\mathcal{M}(\mathcal{P}) = Bet^{-1}(\mathcal{P})$ defined by

$$\begin{aligned}\mathcal{M}(\mathcal{P}) &= \{m \mid Bet(m) \in \mathcal{P}\} \\ &= \bigcup_{P \in \mathcal{P}} \mathcal{M}(P),\end{aligned}$$

where $\mathcal{M}(P) = Bet^{-1}(P)$ denotes the set of bbas whose pignistic probability distribution equals P (see Figure 2.9).

According to the LCP, m_0 should be approximated by a bba m^* less committed than m_0, with respect to some ordering $\sqsubseteq$. In general, the set $\mathcal{M}(\mathcal{P})$ does not contain a LC element. However, we may define the *admissible* set $\mathcal{M}^*(\mathcal{P})$ as the set of bbas *dominating* (i.e., less committed than) all bbas in $\mathcal{M}(\mathcal{P})$:

$$\mathcal{M}^*(\mathcal{P}) = \{m' \mid m \sqsubseteq m', \forall m \in \mathcal{M}(\mathcal{P})\}. \tag{2.63}$$

It is then natural to choose m^* as the *most committed* element in $\mathcal{M}^*(\mathcal{P})$, if this element exists. The solution of this problem is not obvious in the general case. However, a simple solution can be found if we restrict the search to the subset $\mathcal{C}^*(\mathcal{P}) \subset \mathcal{M}^*(\mathcal{P})$ of *consonant* bbas less committed than all bbas in $\mathcal{M}(\mathcal{P})$, and we consider the q-ordering.

For all $P \in \mathcal{P}$, let $m_P = Bet_{LC}^{-1}(P)$ be the q-LC isopignistic bba induced by P. It is consonant. Let π_P denote the corresponding possibility distribution. Bba m_P is the q-least committed bba in the set $\mathcal{M}(P)$ of bbas whose pignistic probability distribution is P. Consequently, a consonant bba m belongs to $\mathcal{C}^*(\mathcal{P})$ if and only if it is q-less committed than m_P, for all $P \in \mathcal{P}$, ie, for all m_P in $\mathcal{M}(P)$. In other words, a consonant bba m belongs to $\mathcal{C}^*(\mathcal{P})$ if and only if

$$\pi_P \leq \pi, \quad \forall P \in \mathcal{P},$$

where π is the possibility distribution associated with m. It follows that the q-most committed element in $\mathcal{C}^*(\mathcal{P})$ is defined by the following possibility distribution

$$\pi^*(x) = \sup_{P \in \mathcal{P}} \pi_P(x), \quad \forall x \in \mathcal{X}. \tag{2.64}$$

Possibility distribution π^* will be termed the *q-most committed dominating* (q-MCD) possibility distribution associated with $\mathcal{P}$. The corresponding bba will be denoted m^*.

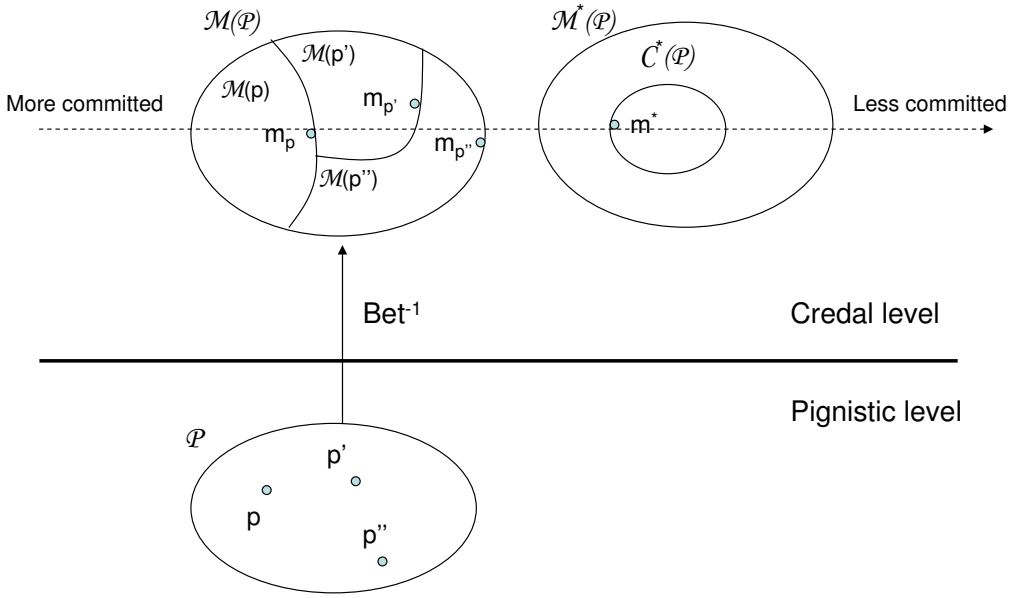


Figure 2.9: Definition of the q-most committed dominating (q-MCD) bba m^* associated with a set $\mathcal{P}$ of probability distribution.
The set $\mathcal{M}(\mathcal{P})$ contains all bbas with pignistic probability function in $\mathcal{P}$. The set $\mathcal{M}^*(\mathcal{P})$ contains all bbas dominating (i.e., less committed than) all bbas in $\mathcal{M}(\mathcal{P})$. The q-MCD bba m^* is the q-most committed consonant bba in $\mathcal{M}^*(\mathcal{P})$.

Example 10. *Let us consider a frame $\mathcal{X} = \{\xi_1, \xi_2, \xi_3\}$ with three elements, and a set $\mathcal{P} = \{P, P', P''\}$ of three probability distributions shown in Table 2.1. The corresponding q-LC possibility distributions π, π', π'' computed from (1.41) are displayed in Table 2.1. Note that there is no q-LC element among these three bbas, as π'' is not comparable to π and π' through the q-ordering. Possibility distribution π^* computed using (2.64) is shown in the last column of Table 2.1. The corresponding bba is*

$$m^*(\{\xi_1\}) = 0.35, \quad m^*(\{\xi_1, \xi_2\}) = 0.05, \quad m^*(\mathcal{X}) = 0.6. \tag{2.65}$$

This bba is q-less committed than all bbas whose pignistic distribution is in $\mathcal{P} = \{P, P', P''\}$, and it is the q-most committed among all consonant bbas in $\mathcal{M}^*(P)$.

x	$P(x)$	$P'(x)$	$P''(x)$	$\pi(x)$	$\pi'(x)$	$\pi''(x)$	$\pi^*(x)$
ξ_1	0.7	0.6	0.65	1	1	1	1
ξ_2	0.2	0.25	0.1	0.5	0.65	0.3	0.65
ξ_3	0.1	0.15	0.25	0.3	0.45	0.6	0.6

Table 2.1: Pignistic probabilities and corresponding q-LC isopignistic possibility distributions of Example 10.

Remark 8. *By definition, the q-MCD bba m^* is the q-most committed element among all* consonant *bbas that are q-less committed than all bbas in $\mathcal{M}(\mathcal{P})$. The restriction to consonant bbas is justified by the existence and unicity of a solution in $\mathcal{C}^*(\mathcal{P})$, whereas the existence of a q-most committed element in $\mathcal{M}^*(\mathcal{P})$ is not guaranteed in general. Additionally, finding the solution in $\mathcal{C}^*(\mathcal{P})$ is computationally tractable in several cases of practical interest, as will be shown below, and the result usually has a very simple expression. It may happen, however, that a q-most committed element in $\mathcal{M}^*(\mathcal{P})$ exists, and that it is strictly more committed than m^*. This*

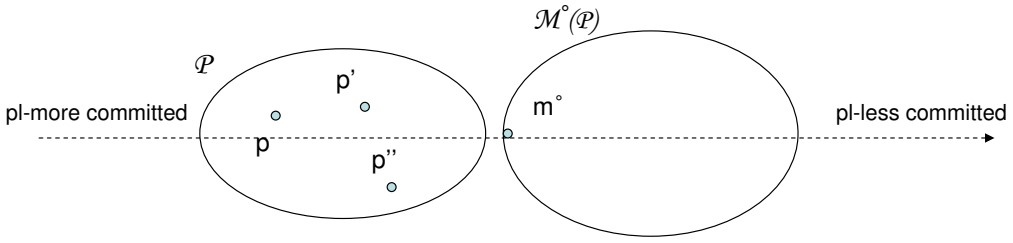


Figure 2.10: Illustration of the approach introduced in [42]:
m° is the *pl*-most committed bba in the set $\mathcal{M}^\circ(\mathcal{P})$ of bbas that are less committed than all probability measures in $\mathcal{P}$. This approach does not distinguish between the pignistic and credal levels (compare with Figure 2.9).

is the case, in particular, when function q_{max} defined by

$$q_{max}(A) = \max_{p \in \mathcal{P}} q_P(A), \quad \forall A \subseteq \mathcal{X} \tag{2.66}$$

is a commonality function, q_P being the commonality function associated with m_P. In that case, the corresponding bba m_{max} is obviously the q-most committed element in $\mathcal{M}^*(\mathcal{P})$. This is the case in Example 10: as shown in Table 2.2, $q_{max} = \max(q, q', q'')$ is a commonality function, and the corresponding bba m_{max} is strictly q-more committed than m^*.

A	$\{\xi_1\}$	$\{\xi_2\}$	$\{\xi_1, \xi_2\}$	$\{\xi_3\}$	$\{\xi_1, \xi_3\}$	$\{\xi_2, \xi_3\}$	$\mathcal{X}$
$q(A)$	1	0.5	0.5	0.3	0.3	0.3	0.3
$q'(A)$	1	0.65	0.65	0.45	0.45	0.45	0.45
$q''(A)$	1	0.3	0.3	0.6	0.6	0.3	0.3
$q^*(A)$	1	0.65	0.65	0.6	0.6	0.6	0.6
$q_{max}(A)$	1	0.65	0.65	0.6	0.6	0.45	0.45
$m_{max}(A)$	0.2	0	0.2	0	0.15	0	0.45

Table 2.2: Calculation of q_{max} for the data of Example 10.
In that case, q_{max} is a commonality function, and the corresponding bba m_{max} is strictly q-more committed than m^*, as $q_{max}(A) < q^*(A)$ for $A = \{\xi_2, \xi_3\}$ and for $A = \mathcal{X}$.

Remark 9. *The approach presented here is different from that introduced in [42] and 2.2, in which we searched for the pl-most committed bba m°, in the set $\mathcal{M}^\circ(\mathcal{P})$ of bbas that are less committed than* all probability measures *in $\mathcal{P}$ (see Figure 2.10). In that alternative approach, the solution is obtained as the lower envelope $\underline{P}$ of $\mathcal{P}$, when it is a belief function. This is the case, in particular, when $\mathcal{P}$ is a p-box (see Section 2.2), or when it is constructed from a multinomial confidence region with $K \leq 3$ [42]. Different heuristics were introduced in [42] for constructing a belief function less committed than $\underline{P}$ when $\underline{P}$ is not a belief function. As will be shown below, the approach adopted here usually yields a simpler result as it produces consonant belief functions. Additionally, it may be argued to be more in line with the two-level structure of the TBM, as it does not directly compare probabilities at the pignistic level with belief functions at the credal level.*

2.3.2 Application to a Sample of a Discrete Random Variable

In this section, we consider the application of the methodology described in Section 2.3.1 to the construction of a predictive belief function based on an independent and identically distributed (iid) sample $X_1, \ldots, X_n$ from a discrete variable X defined on a finite domain

$\mathcal{X}$. We first show that a set $\mathcal{P}$ of possible probability distributions of X can be constructed using multinomial simultaneous confidence intervals. An algorithm for finding the q-MCD possibility distribution π^* induced by $\mathcal{P}$ is then presented.

2.3.3 Construction of $\mathcal{P}$

As before (see Section 2.2.2), let X be a discrete r.v. on a finite domain $\mathcal{X} = \{\xi_1, \ldots, \xi_K\}$, with unknown probability distribution P_X. Given an iid random sample $X_1, \ldots, X_n$ from P_X, let $N_k = \sum_{i=1}^{n} 1_{\xi_k}(X_i)$ denote the number of observations in category ξ_k. The random vector $\mathbf{N} = (N_1, \ldots, N_K)$ has a multinomial distribution with parameters n and $\mathbf{p} = (p_1, \ldots, p_K)$, with $p_k = P_X(\{\xi_k\})$.

Of particular interest are simultaneous confidence intervals, i.e., regions defined as a Cartesian product of intervals:

$$S(\mathbf{N}) = [p_1^-, p_1^+] \times \ldots \times [p_K^-, p_K^+], \tag{2.67}$$

which have easy interpretation. Such asymptotic confidence regions were proposed by Quesenberry and Hurst [98], and Goodman [53]. Goodman's intervals are defined as:

$$p_k^- = \frac{a + 2n_k - \sqrt{\Delta_k}}{2(n+a)} \tag{2.68}$$

$$p_k^+ = \frac{a + 2n_k + \sqrt{\Delta_k}}{2(n+a)}, \tag{2.69}$$

where a is the quantile of order $1 - \alpha/K$ of the chi-square distribution with one degree of freedom (for $K > 2$), and

$$\Delta_k = a\left(a + \frac{4n_k(n - n_k)}{n}\right). \tag{2.70}$$

When $K = 2$, a should be defined as the quantile of order $1 - \alpha$ of the chi-square distribution with one degree of freedom. Note that p_k^- and p_k^+ both converge in probability towards p_k as $n \to +\infty$, for $k = 1, \ldots, K$.

As remarked in [42, 81], $S(\mathbf{N})$ can be seen as defining a family $\mathcal{P}$ of probability measures. Such a family, obtained by bounding the probability of each singleton, is called a *set of probability intervals* in [26]. Each vector $\mathbf{p}$ of probabilities corresponds to a possible probability measure $\mathbb{P}$ for X.

Example 11. *The data analyzed in [98] and [53] describe the frequency of ten modes of failure as recorded in a study of 870 machines that failed. These data are shown in Table 2.3, together with the corresponding Goodman confidence intervals at confidence level* $1 - \alpha = 0.90$.

Mode ξ_k	1	2	3	4	5	6	7	8	9	10
n_k	5	11	19	30	58	67	92	118	173	297
n_k/n	0.0057	0.013	0.022	0.035	0.067	0.077	0.106	0.136	0.199	0.341
p_k^-	0.002	0.006	0.012	0.022	0.048	0.057	0.082	0.109	0.166	0.301
p_k^+	0.017	0.027	0.039	0.054	0.092	0.104	0.136	0.168	0.236	0.384

Table 2.3: Goodman simultaneous confidence intervals for the data of Example 11, at confidence level $1 - \alpha = 0.90$.

2.3.4 Determination of the q-MCD Possibility Distribution

Following the approach outlined in the previous section, assume that $\mathcal{P}$ is interpreted as a set of pignistic probabilities. For each $Bet p$ in $\mathcal{P}$, the q-LC isopignistic belief function is defined by (1.41). Consequently, the q-MCD possibility distribution π^* defined by (1.41) can be obtained by solving the following maximization problems:

$$\pi_k^* = \max_P \sum_{\ell=1}^{K} \min(p_k, p_\ell) \tag{2.71}$$

under the constraints

$$p_\ell^- \leq p_\ell \leq p_\ell^+, \quad \ell = 1, \ldots, K \tag{2.72}$$

$$\sum_{\ell=1}^{K} p_\ell = 1. \tag{2.73}$$

Note that this problem is similar to the one addressed in [81] for a different probability-posssibility transformation. We may first notice that the solution has a simple upper bound $\widetilde{\pi}_k^*$ defined by

$$\widetilde{\pi}_k^* = \min\left(1, \sum_{\ell=1}^{K} \min(p_k^+, p_\ell^+)\right), \tag{2.74}$$

which can be used as an approximation.

The exact solution to optimization problem (2.71)-(2.73) may be found by reasoning as follows.

First observe that (2.71) can be written as

$$\pi_k^* = \max_{\mathcal{P}} \left(\sum_{\ell \in \overline{S}_k} p_\ell + |S_k| p_k \right), \tag{2.75}$$

where $S_k = \{\ell \in \{1, \ldots, K\} \mid p_\ell \geq p_k\}$ is the set of indices of probabilities p_ℓ at least equal to p_k, $|S_k|$ is its cardinality , and $\overline{S}_k$ is the complement of S_k. For fixed S_k, the objective function in (2.75) is linear and it may be maximized using a standard linear programming algorithm. An approach for solving problem (2.71)-(2.73) is thus to enumerate all possible sets S_k compatible with constraints (2.72), and for each S_k solve the following linear programming problem LP(S_k):

$$\max_{\mathcal{P}} \left(\sum_{\ell \in \overline{S}_k} p_\ell + |S_k| p_k \right) \tag{2.76}$$

under constraints (2.72), (2.73) and

$$p_\ell \geq p_k, \quad \forall \ell \in S_k. \tag{2.77}$$

$$p_\ell \leq p_k, \quad \forall \ell \in \overline{S}_k. \tag{2.78}$$

If problem LP(S_k) is feasible, let $\pi_k^*(S_k)$ denote its solution of the above problem. Then π_k^* is the maximum of $\pi_k^*(S_k)$ for all S_k such that the problem LP(S_k) is feasible.

To enumerate all possible sets S_k, we may observe that indices ℓ such that $p_\ell^- \geq p_k^+$ surely belong to S_k, whereas indices ℓ such that $p_\ell^+ < p_k^-$ cannot belong to S_k. All other indices may be included in S_k or not. Formally, let

$$S_k^* = \{k\} \cup \{\ell \in \{1, \ldots, K\} \mid p_\ell^- \geq p_k^+\}, \tag{2.79}$$

$$I_k^* = \{\ell \in \{1, \ldots, K\} \mid p_\ell^+ < p_k^-\}, \tag{2.80}$$

and

$$P_k^* = \{1, \ldots, K\} \setminus (S_k^* \cup I_k^*). \tag{2.81}$$

Then, all possible sets S_k are of the form $S_k = S_k^* \cup A$ for $A \subseteq P_k^*$.

The proposed algorithm may be summarized as follows:

1. Initialize $\pi_k^* = 0$.
2. Compute S_k^*, I_k^* and P_k^* using (2.79)-(2.81).
3. For all $A \subseteq P_k^*$:
   (a) Let $S_k = S_k^* \cup A$.
   (b) If constraints (2.72)-(2.73) and (2.77)-(2.78) are feasible, then
      i. Compute $\pi_k^*(S_k) = \max_{\mathcal{P}} \sum_{\ell \in \overline{S}_k} p_\ell + |S_k| p_k$ under constraints (2.72)-(2.73) and (2.77)-(2.78) using a linear programming procedure.
      ii. $\pi_k^* = \max(\pi_k^*, \pi_k^*(S_k))$.
   (c) End if.
4. End For.

Example 12. *Let us come back to the data of Example 11 reported in Table 2.3. The values of π_k^* for $k = 1, \ldots, 10$ are shown in Table 2.4, together with the approximations $\widetilde{\pi}_k^*$ computed using (2.74). The q-LC possibility distribution $\widehat{\pi}$ computed from the sample frequencies n_k/n is also shown in Table 2.4. This possibility distribution is more committed than π^* as it does not take into account sampling uncertainty. Detailed calculations for $k = 7$ are presented below.*

Mode ξ_k	1	2	3	4	5	6	7	8	9	10
n_k	5	11	19	30	58	67	92	118	173	297
n_k/n	0.0057	0.013	0.022	0.035	0.067	0.077	0.106	0.136	0.199	0.341
$\widehat{\pi}_k$	0.058	0.120	0.193	0.282	0.475	0.526	0.641	0.731	0.858	1
π_k^*	0.171	0.258	0.353	0.462	0.688	0.735	0.804	0.867	0.935	1
$\widetilde{\pi}_k^*$	0.171	0.258	0.353	0.462	0.688	0.747	0.875	0.973	1	1

Table 2.4: Possibility distributions computed for the failure mode data of Example 12: q-LC possibility distribution computed from the sample frequencies ($\widehat{\pi}$), q-MCD possibility distribution computed from the multinomial confidence intervals shown in Table 2.3 (π^*), and approximation computed using (2.74) ($\widetilde{\pi}$).

Detailed calculation for $k = 7$:

Let us consider the calculation of π_7^*. We know that $S_7^* = \{7, 9, 10\}$, $I_7^* = \{1, 2, 3, 4\}$ and $P_7^* = \{5, 6, 8\}$. Using the algorithm described in Section 2.3.4, we have to solve a distinct linear optimization problem for each of the $2^3 = 8$ subsets A of P_7^*. Let us consider these eight cases:

- *For $A = \emptyset$, $S_7 = \{7, 9, 10\}$. Constraints (2.72), (2.73) and*

$$p_\ell \geq p_k, \quad \forall \ell \in \{7, 9, 10\}, \tag{2.82}$$

$$p_\ell \leq p_k, \quad \forall \ell \in \{1, 2, 3, 4, 5, 6, 8\} \tag{2.83}$$

are consistent. The maximum of $\sum_{\ell=1}^{6} p_\ell + 3p_7 + p_8$ under these constraints is 0.804; it is achieved for

$$\mathbf{p} = (0.013, 0.021, 0.030, 0.043, 0.076, 0.086, 0.136, 0.128, 0.166, 0.301). \tag{2.84}$$

- *For* $A = \{5\}$, $S_7 = \{5,7,9,10\}$. *Constraints (2.72), (2.73) and*

$$p_\ell \geq p_k, \quad \forall \ell \in \{5,7,9,10\}, \tag{2.85}$$

$$p_\ell \leq p_k, \quad \forall \ell \in \{1,2,3,4,6,8\} \tag{2.86}$$

are not consistent, so the optimization problem is not feasible.

- *For* $A = \{6\}$, $S_7 = \{6,7,9,10\}$. *Constraints (2.72), (2.73) and*

$$p_\ell \geq p_k, \quad \forall \ell \in \{6,7,9,10\}, \tag{2.87}$$

$$p_\ell \leq p_k, \quad \forall \ell \in \{1,2,3,4,5,8\} \tag{2.88}$$

are not consistent, so the optimization problem is not feasible.

- *For* $A = \{8\}$, $S_7 = \{7,8,9,10\}$. *Constraints (2.72), (2.73) and*

$$p_\ell \geq p_k, \quad \forall \ell \in \{7,8,9,10\}, \tag{2.89}$$

$$p_\ell \leq p_k, \quad \forall \ell \in \{1,2,3,4,5,6\} \tag{2.90}$$

are consistent. The maximum of $\sum_{\ell=1}^{6} p_\ell + 4p_7$ *under these constraints is 0.804; it is achieved for*

$$\mathbf{p} = (0.013, 0.020, 0.029, 0.042, 0.074, 0.083, 0.136, 0.136, 0.166, 0.301). \tag{2.91}$$

- *For* $A = \{5,6\}$, $S_7 = \{5,6,7,9,10\}$. *Constraints (2.72), (2.73) and*

$$p_\ell \geq p_k, \quad \forall \ell \in \{5,6,7,9,10\}, \tag{2.92}$$

$$p_\ell \leq p_k, \quad \forall \ell \in \{1,2,3,4,8\} \tag{2.93}$$

are consistent. The maximum of $\sum_{\ell=1}^{4} p_\ell + 5p_7 + p_8$ *under these constraints is 0.659; it is achieved for*

$$\mathbf{p} = (0.010, 0.017, 0.026, 0.038, 0.092, 0.098, 0.092, 0.109, 0.192, 0.327). \tag{2.94}$$

- *For* $A = \{5,8\}$, $S_7 = \{5,7,8,9,10\}$. *Constraints (2.72), (2.73) and*

$$p_\ell \geq p_k, \quad \forall \ell \in \{5,7,8,9,10\}, \tag{2.95}$$

$$p_\ell \leq p_k, \quad \forall \ell \in \{1,2,3,4,6\} \tag{2.96}$$

are consistent. The maximum of $\sum_{\ell=1}^{4} p_\ell + p_6 + 5p_7$ *under these constraints is 0.688; it is achieved for*

$$\mathbf{p} = (0.017, 0.027, 0.039, 0.054, 0.092, 0.092, 0.092, 0.112, 0.170, 0.306). \tag{2.97}$$

- *For* $A = \{6,8\}$, $S_7 = \{6,7,8,9,10\}$. *Constraints (2.72), (2.73) and*

$$p_\ell \geq p_k, \quad \forall \ell \in \{6,7,8,9,10\}, \tag{2.98}$$

$$p_\ell \leq p_k, \quad \forall \ell \in \{1,2,3,4,5\} \tag{2.99}$$

are consistent. The maximum of $\sum_{\ell=1}^{5} p_\ell + 5p_7$ *under these constraints is 0.735; it is achieved for*

$$\mathbf{p} = (0.014, 0.024, 0.037, 0.054, 0.089, 0.104, 0.104, 0.109, 0.166, 0.301). \tag{2.100}$$

- *For $A = \{5,6,8\}$, $S_7 = \{5,6,7,8,9,10\}$. Constraints (2.72), (2.73) and*

$$p_\ell \geq p_k, \quad \forall \ell \in \{5,6,7,8,9,10\}, \tag{2.101}$$

$$p_\ell \leq p_k, \quad \forall \ell \in \{1,2,3,4\} \tag{2.102}$$

are consistent. The maximum of $\sum_{\ell=1}^{4} p_\ell + 6p_7$ under these constraints is 0.688; it is achieved for

$$\mathbf{p} = (0.017, 0.027, 0.039, 0.054, 0.092, 0.093, 0.092, 0.111, 0.170, 0.305). \tag{2.103}$$

The highest value obtained in these eight linear optimization problems is 0.804. Thus, $\pi_7^* =$ 0.804.

2.3.5 Application to Continuous Parametric Models

The general approach introduced in Section 2.3.1 can also be applied to the construction of a predictive belief function based on a sample from a continuous r.v. X with unimodal probability density function $f(x;\theta)$ depending on a parameter θ. For each value of θ, the q-LC possibility distribution $\pi(x;\theta)$ may be computed using (1.60) or (1.61). Given a confidence region $\mathcal{R}$ for θ, one may then compute the q-MCD possibility distribution π^* as

$$\pi^*(x) = \sup_{\theta \in \mathcal{R}} \pi(x;\theta), \tag{2.104}$$

for all $x \in \mathbb{R}$.

This approach is illustrated below in the cases of exponential and normal distributions.

2.3.6 Exponential Distribution

Let us assume that X has an exponential distribution $\mathcal{E}(\mu)$ with density function $f(x;\mu)$ defined by (1.62). As shown in Example 4, Section 1.3.5, the corresponding q-LC possibility distribution is defined for fixed μ by (1.64).

Here, we assume that μ is unknown but an iid sample $X_1, \ldots, X_n$ from $\mathcal{E}(\mu)$ has been observed. It is well known from standard textbooks (see, e.g. [47]) that the sample average $\overline{X}$ is an unbiased estimator for μ, and its variance is μ^2/n. From the Central Limit Theorem, the statistic

$$\frac{\sqrt{n}(\overline{X} - \mu)}{\mu} \tag{2.105}$$

converges in distribution to a r.v. that is normally distributed with mean 0 and variance 1. For large n and $\alpha \in (0,1)$, the following thus holds

$$\mathbb{P}\left(-u_{1-\alpha/2} \leq \frac{\sqrt{n}(\overline{X} - \mu)}{\mu} \leq u_{1-\alpha/2}\right) \approx 1 - \alpha, \tag{2.106}$$

where $u_{1-\alpha_1/2}$ is the upper $\alpha_1/2$ percentile of a standard normal distribution. Equivalently,

$$\mathbb{P}\left(\frac{\overline{X}}{1 + u_{1-\alpha/2}/\sqrt{n}} \leq \mu \leq \frac{\overline{X}}{1 + u_{1-\alpha/2}/\sqrt{n}}\right) \approx 1 - \alpha. \tag{2.107}$$

The interval

$$\mathcal{R}(X_1, \ldots, X_n) = \left\{\mu : \frac{\overline{X}}{1 + u_{1-\alpha/2}/\sqrt{n}} \leq \mu \leq \frac{\overline{X}}{1 - u_{1-\alpha/2}/\sqrt{n}}\right\} \tag{2.108}$$

is therefore an approximate confidence interval for μ at level $1-\alpha$.

In order to compute the supremum of $\pi(x;\mu)$ for $\mu \in \mathcal{R}(X_1,\ldots,X_n)$, we observe that

$$\frac{\partial \pi(x;\mu)}{\partial \mu} = \frac{x^2}{\mu^3} e^{-x/\mu} > 0. \tag{2.109}$$

Consequently,

$$\pi^*(x) = \pi\left(x;\widehat{\mu}^+\right) \tag{2.110}$$

with

$$\widehat{\mu}^+ = \frac{\overline{X}}{1-u_{1-\alpha/2}/\sqrt{n}}. \tag{2.111}$$

Figure 2.11 shows the possibility distribution $\pi^*(x)$ for $\overline{x}=1$ and various values of n. The case $n=\infty$ corresponds to the situation where parameter μ is known: in that case, π^* is simply the q-LC isopignistic possibility distribution induced by the exponential pignistic distribution with $\mu=1$.

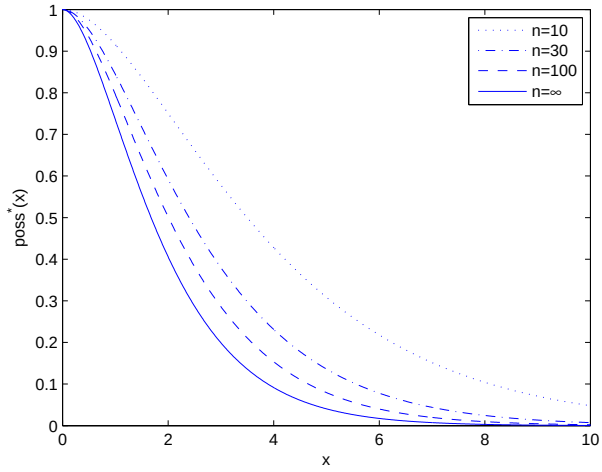


Figure 2.11: Plot of $\pi^*(x)$ for the exponential distribution with $\overline{x}=1$, $\alpha=0.1$, and $n=10$, 30, 100 and ∞.

Example 13. *Suppose that the life time X of light bulbs manufactured by a certain company follows an exponential distribution $\mathcal{E}(\mu)$ with unknown μ. For $n=20$ bulbs, the average observed life time was $\overline{X}=30.5$ thousands of hours. What are the belief and plausibility that the life time of a new bulb will exceed 50 thousands of hours ?*

For $\alpha=0.05$, $u_{1-\alpha/2}=1.96$ and $\widehat{\mu}^+ = 30.5/(1-1.96/\sqrt{20}) = 54.3$. Thus, the q-MCD possibility distribution is

$$\pi^*(x) = e^{-x/54.3}\left(1+\frac{x}{54.3}\right). \tag{2.112}$$

Now,

$$pl([50,+\infty)) = \sup_{x\geq 50} \pi^*(x) = \pi^*(50) = 0.76 \tag{2.113}$$

and

$$bel([50,+\infty)) = 1 - pl([0,50)) = 1 - \sup_{0\leq x<50} \pi^*(x) = 1-\pi^*(0) = 0. \tag{2.114}$$

2.3.7 Normal Distribution

Let us now assume that X has a normal distribution with mean μ and variance σ^2. If these two parameters are known, then the possibility distribution $\pi(\cdot;\mu,\sigma)$ is given by (1.65).

When μ and σ^2 are unknown but an iid sample $X_1,\ldots,X_n$ is available, then it is possible to define a joint confidence region for μ and σ^2 [9]. In particular, Mood's exact confidence region at level $1-\alpha=(1-\alpha_1)(1-\alpha_2)$ is defined by

$$\mathcal{R}(X_1,\ldots,X_n)=\left\{(\mu;\sigma^2):\overline{X}-u_{1-\alpha_1/2}\frac{\sigma}{\sqrt{n}}\leq\mu\leq\overline{X}+u_{1-\alpha_1/2}\frac{\sigma}{\sqrt{n}},\right.$$
$$\left.\frac{nS^2}{\chi^2_{n-1;1-\alpha_2/2}}\leq\sigma^2\leq\frac{nS^2}{\chi^2_{n-1;\alpha_2/2}}\right\},\quad(2.115)$$

where $\overline{X}$ is the sample mean, $S^2=(1/n)\sum_{i=1}^n(X_i-\overline{X})^2$ is the sample variance, $u_{1-\alpha_1/2}$ is the upper $\alpha_1/2$ percentile of a standard normal distribution, and $\chi^2_{n-1;\alpha_2/2}$ and $\chi^2_{n-1;1-\alpha_2/2}$ are the lower and upper $\alpha_2/2$ percentiles of a χ^2_{n-1} distribution. The shape of that region is illustrated in Figure 2.12. Values of α_1 and α_2 yielding a region of smallest possible size for a fixed confidence level are given in [9].

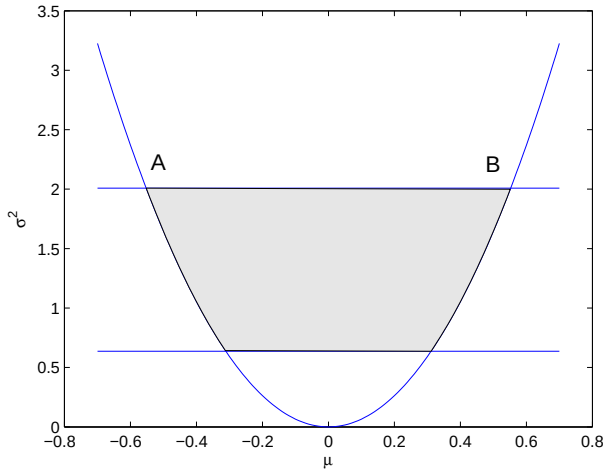


Figure 2.12: Shape of Mood's exact region:
Mood's Exact Region for $\alpha=0.1$, $\alpha_1=\alpha_2$ and $n=25$. Without loss of generality, $\overline{X}=0$ and $s^2=1$. The points with coordinates $(\hat{\mu}^-,(\hat{\sigma}^+)^2)$ and $(\hat{\mu}^+,(\hat{\sigma}^+)^2)$ are denoted A and B, respectively.

Let $\mathcal{P}$ denote the set of Gaussian distributions with parameters contained in confidence region $\mathcal{R}$. Applying the principle outlined in Section 2.3.1, the q-MCD possibility distribution π^* may be obtained for any x by maximizing $\pi(x;\mu,\sigma)$ given by (1.65) with respect to μ and σ, under the constraint $(\mu,\sigma^2)\in\mathcal{R}$. The result is given by the following proposition.

Proposition 5. *The q-MCD possibility distribution π^* associated with Mood's confidence confidence region $\mathcal{R}$ at level $(1-\alpha_1)(1-\alpha_2)$ is*

$$\pi^*(x)=\begin{cases}\pi(x;\hat{\mu}^-,\hat{\sigma}^+) & \text{if } x<\hat{\mu}^-\\ 1 & \text{if } \hat{\mu}^-\leq x\leq\hat{\mu}^+\\ \pi(x;\hat{\mu}^+,\hat{\sigma}^+) & \text{if } x>\hat{\mu}^+,\end{cases}\quad(2.116)$$

with

$$\hat{\sigma}^+ = \left(\frac{nS^2}{\chi^2_{n-1;\alpha_2/2}} \right)^{1/2}, \tag{2.117}$$

$$\hat{\mu}^- = \overline{X} - u_{1-\alpha_1/2}\frac{\hat{\sigma}^+}{\sqrt{n}}, \quad \hat{\mu}^+ = \overline{X} + u_{1-\alpha_1/2}\frac{\hat{\sigma}^+}{\sqrt{n}}. \tag{2.118}$$

Proof. By definition

$$\pi^*(x) = \sup_{(\mu,\sigma^2)\in\mathcal{R}} \pi(x;\mu,\sigma). \tag{2.119}$$

If $x \in [\hat{\mu}^-, \hat{\mu}^+]$, then we can get $\pi(x;\mu,\sigma) = 1$ by setting $\mu = x$.

If $x < \hat{\mu}^-$, then the value 1 cannot be reached. However, using standard calculus, we obtain, for $x < \mu$:

$$\frac{\partial\pi(x;\mu,\sigma)}{\partial\mu} = -\frac{(x-\mu)^2}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) < 0 \tag{2.120}$$

and

$$\frac{\partial\pi(x;\mu,\sigma)}{\partial\sigma} = \frac{(\mu-x)^3}{\sigma^4\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) > 0. \tag{2.121}$$

Consequently, $\pi(x;\mu,\sigma)$ is maximized by jointly minimizing μ and maximizing σ, and the maximum is reached for $(\mu,\sigma) = (\hat{\mu}^-, \hat{\sigma}^+)$. Similarly, for $x > \hat{\mu}^+$, we get:

$$\frac{\partial\pi(x;\mu,\sigma)}{\partial\mu} = \frac{(x-\mu)^2}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) > 0 \tag{2.122}$$

and

$$\frac{\partial\pi(x;\mu,\sigma)}{\partial\sigma} = \frac{(x-\mu)^3}{\sigma^4\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) > 0. \tag{2.123}$$

Consequently, the maximum of $\pi(x;\mu,\sigma)$ for $x > \hat{\mu}^+$ is reached for $(\mu,\sigma) = (\hat{\mu}^+, \hat{\sigma}^+)$. □

Figure 2.13 shows the possibility distribution $\pi^*(x)$ for $\overline{x} = 0$, $s^2 = 1$, $\alpha = 0.1$ and various values of n. The case $n = \infty$ corresponds to the situation where parameters μ and σ^2 are known: in that case, π^* is simply que q-LC isopignistic possibility distribution induced by the normal pignistic distribution with $\mu = 0$ and $\sigma^2 = 1$.

2.3.8 Conclusion

A new method for generating a belief function from statistical data in the TBM framework has been presented. The starting point of this method is the assumption that, if the probability distribution P_X of a random variable is known, then the belief function quantifying our belief regarding a future realization of X should be such that its pignistic probability distribution equals P_X. In the realistic situation where P_X is unknown but a random sample of X is available, it is possible to build a set $\mathcal{P}$ of probability distributions containing P_X with some confidence level. Following the LCP, it is then reasonable to impose that the sought belief function be q-less committed than all belief functions whose pignistic probability distribution is in $\mathcal{P}$. Our method selects the q-most committed consonant belief function verifying this property, referred to as the q-MCD possibility distribution induced by $\mathcal{P}$. This general principle has been illustrated in three special cases of general interest involving discrete, exponential and normal distributions, respectively.

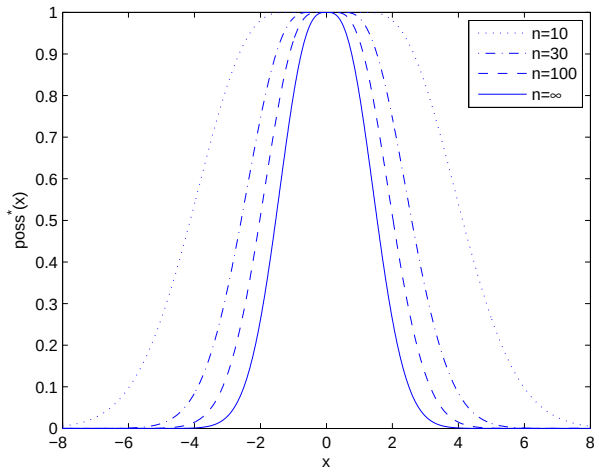


Figure 2.13: Plot of $\pi^*(x)$ for the normal distribution with $\overline{x} = 0$, $s^2 = 1$, $\alpha = 0.1$, $\alpha_1 = \alpha_2$, and $n = 10, 30, 100$ and ∞.

2.4 Multi-class classification example

In order to demonstrate the usefulness of the proposed approach for constructing a belief function from sample data, let us consider the following multi-sensor classification problem.

Problem Statement and Solution in the TBM

Let Σ denote a system that can be in two states (classes) ω_1 and ω_2 corresponding, e.g., to the normal state and a faulty state. Let $\Omega = \{\omega_1, \omega_2\}$. The system is equipped with two sensors S_x and S_y that deliver measurements X and Y, considered to be r.v.'s with distribution depending on the system state. Both r.v.'s are assumed to be normally distributed and independent conditionally on the system state.

Let us further assume that sensor S_x has been available for a long time, so that we have gathered a learning set $\mathcal{L}_x$ of $n_x = 1000$ observations of X from each class. In contrast, sensor S_y is recent and we have only a much small learning set $\mathcal{L}_y$ of $n_y \ll n_x$ observations of Y from each class.

Based on this information, we would like to construct a decision rule for predicting the system state from measurements x_0 and y_0 delivered by the two sensors.

In the TBM, the solution of this problem goes through the following steps [116, 28, 43]:

1. Compute the plausibilities $pl(x_0|\omega_k)$ and $pl(y_0|\omega_k)$ of observing x_0 and y_0, respectively, when the system is in state ω_k ($k = 1, 2$) using the learning data;

2. As X and Y are conditionally independent, let $pl(x_0, y_0|\omega_k) = pl(x_0|\omega_k)pl(y_0|\omega_k)$.

3. Using the General Bayesian Theorem (GBT) [116], compute the conditional bba $m^\Omega(\cdot|x_0, y_0)$ on Ω given $X = x_0$ and $Y = y_0$ using the following formula:

$$m^\Omega(\cdot|x_0, y_0) = \{\omega_1\}^{pl(x_0,y_0|\omega_2)} \textcircled{\cap} \{\omega_2\}^{pl(x_0,y_0|\omega_1)},$$

where the notation $\{\omega_k\}^w$ stands for the simple bba m such that $m(\{\omega_k\}) = 1 - w$

and $m(\Omega) = w$. Thus,

$$\begin{aligned} m^{\Omega}(\emptyset|x_0, y_0) &= (1 - pl(x_0, y_0|\omega_1))\,(1 - pl(x_0, y_0|\omega_2)) && (2.124)\\ m^{\Omega}(\{\omega_1\}|x_0, y_0) &= pl(x_0, y_0|\omega_1)\,(1 - pl(x_0, y_0|\omega_2)) && (2.125)\\ m^{\Omega}(\{\omega_2\}|x_0, y_0) &= (1 - pl(x_0, y_0|\omega_1))\,pl(x_0, y_0|\omega_2) && (2.126)\\ m^{\Omega}(\Omega|x_0, y_0) &= pl(x_0, y_0|\omega_1)pl(x_0, y_0|\omega_2). && (2.127) \end{aligned}$$

4. Compute the pignistic probability $BetP^{\Omega}(\cdot|x_0, y_0)$ induced by $m^{\Omega}(\cdot|x_0, y_0)$:

$$\begin{aligned} BetP^{\Omega}(\omega_1|x_0, y_0) &= \frac{m^{\Omega}(\{\omega_1\}|x_0, y_0) + m^{\Omega}(\Omega|x_0, y_0)/2}{1 - m^{\Omega}(\emptyset|x_0, y_0)},\\ BetP^{\Omega}(\omega_2|x_0, y_0) &= 1 - BetP^{\Omega}(\omega_1|x_0, y_0). \end{aligned}$$

5. Select the system state with the highest pignistic probability.

The approach exposed in this paper concerns step 1. The plausibilities $pl(x_0|\omega_k)$ and $pl(y_0|\omega_k)$ may be computed from (1.60) by substituting the mean and standard deviation by their sample estimates (this method will be referred to as LC), from (2.116) using Mood confidence regions (MCD method), or from (2.57), using Cheng and Ile's confidence band (CI method). In the latter two cases, function $pl(y_0|\omega_k)$ will reflect the additional sampling uncertainty.

Illustrative Example

Figures 2.14 and 2.15 show typical learning sets $\mathcal{L}_x$ and $\mathcal{L}_y$ with, respectively, $n_x = 1000$ and $n_y = 50$ observations for each class, as well as the corresponding possibility distributions computed using each of the three methods. For the MCD method, the confidence level of the Mood regions were fixed at $1 - \alpha = 0.8$. For the CI method, the confidence level of the confidence bands were fixed at $1 - alpha = 0.99$. The values $x_0 = 1.5$ and $y_0 = -1$ are indicated as vertical lines in the upper parts of Figures 2.14 and 2.15.

Let us first do the computations for the LC method: $pl(x_0|\omega_1) = 0.539$, $pl(x_0|\omega_2) = 0.966$, $pl(y_0|\omega_1) = 0.740$, $pl(y_0|\omega_2) = 0.522$. Hence (step 2),

$$\begin{aligned} pl(x_0, y_0|\omega_1) &= 0.539 \times 0.740 = 0.399\\ pl(x_0, y_0|\omega_2) &= 0.966 \times 0.522 = 0.504. \end{aligned}$$

Using (2.124)-(2.127) we get (step 3):

$$\begin{aligned} m^{\Omega}(\emptyset|x_0, y_0) &= (1 - 0.399)\,(1 - 0.504) = 0.298\\ m^{\Omega}(\{\omega_1\}|x_0, y_0) &= 0.399 \times (1 - 0.504) = 0.198\\ m^{\Omega}(\{\omega_2\}|x_0, y_0) &= (1 - 0.399) \times 0.504 = 0.303\\ m^{\Omega}(\Omega|x_0, y_0) &= 0.399 \times 0.504 = 0.201. \end{aligned}$$

The corresponding pignistic probability function is:

$$\begin{aligned} BetP^{\Omega}(\omega_1|x_0, y_0) &= 0.425,\\ BetP^{\Omega}(\omega_2|x_0, y_0) &= 0.575. \end{aligned}$$

Using the MCD method, $pl(x_0|\omega_1) = 0.600$, $pl(x_0|\omega_2) = 0.978$, $pl(y_0|\omega_1) = 0.922$, $pl(y_0|\omega_2) = 0.796$. Thus,

$$\begin{aligned} pl(x_0, y_0|\omega_1) &= 0.600 \times 0.922 = 0.553\\ pl(x_0, y_0|\omega_2) &= 0.978 \times 0.796 = 0.779, \end{aligned}$$

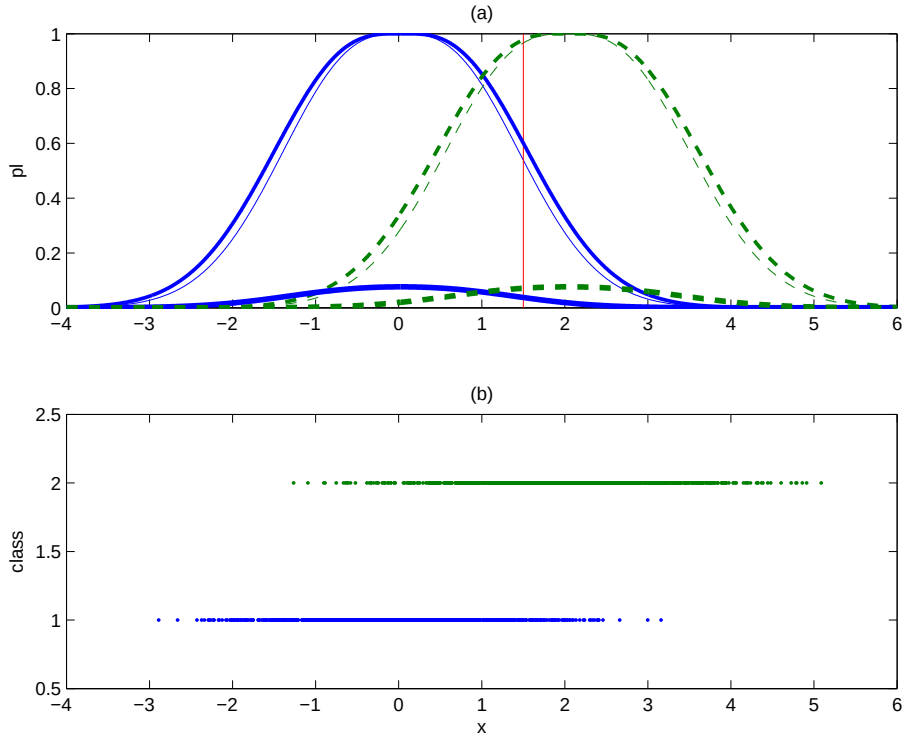


Figure 2.14: (a): Plot of $pl(x|\omega_1)$ (solid lines) and $pl(x|\omega_2)$ (dashed lines) computed using the LC, MCD and CI methods (thin, thick and very thick lines, respectively), as functions of x. (b) Dot plots of training data from each class in learning set $\mathcal{L}_x$.

and

$$\begin{aligned} m^{\Omega}(\emptyset|x_0,y_0) &= (1-0.553)\,(1-0.779) = 0.099 \\ m^{\Omega}(\{\omega_1\}|x_0,y_0) &= 0.553 \times (1-0.779) = 0.122 \\ m^{\Omega}(\{\omega_2\}|x_0,y_0) &= (1-0.553) \times 0.779 = 0.348 \\ m^{\Omega}(\Omega|x_0,y_0) &= 0.553 \times 0.779 = 0.431. \end{aligned}$$

The corresponding pignistic probability function is

$$\begin{aligned} BetP^{\Omega}(\omega_1|x_0,y_0) &= 0.375, \\ BetP^{\Omega}(\omega_2|x_0,y_0) &= 0.625. \end{aligned}$$

Finally, with the CI method, $pl(x_0|\omega_1) = 0.038$, $pl(x_0|\omega_2) = 0.071$, $pl(y_0|\omega_1) = 0.236$, $pl(y_0|\omega_2) = 0.174$. We get:

$$\begin{aligned} pl(x_0,y_0|\omega_1) &= 0.038 \times 0.236 = 0.009 \\ pl(x_0,y_0|\omega_2) &= 0.071 \times 0.174 = 0.012, \end{aligned}$$

and

$$\begin{aligned} m^{\Omega}(\emptyset|x_0,y_0) &= (1-0.009)\,(1-0.012) = 0.979 \\ m^{\Omega}(\{\omega_1\}|x_0,y_0) &= 0.009 \times (1-0.012) = 0.009 \\ m^{\Omega}(\{\omega_2\}|x_0,y_0) &= (1-0.009) \times 0.012 = 0.012 \\ m^{\Omega}(\Omega|x_0,y_0) &= 0.009 \times 0.012 = 1.08 \times 10^{-4}. \end{aligned}$$

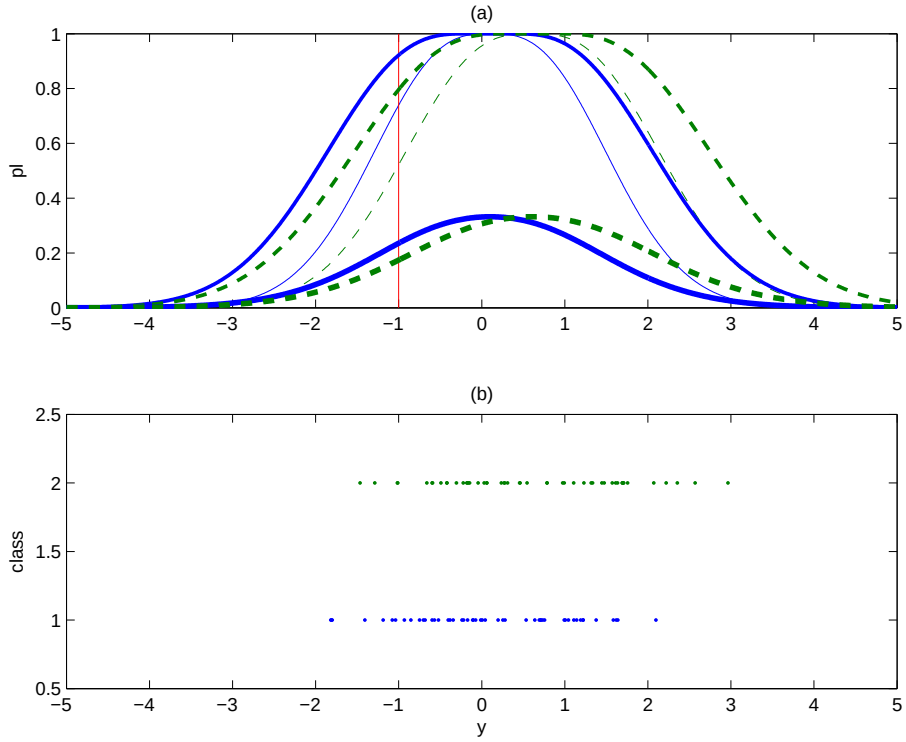


Figure 2.15: (a): Plot of $pl(y|\omega_1)$ (solid lines) and $pl(y|\omega_2)$ (dashed lines) computed using the LC, MCD and CI methods (thin, thick and very thick lines, respectively), as functions of y. (b) Dot plots of training data from each class in learning set $\mathcal{L}_y$.

Finally, the corresponding pignistic probability function is

$$\begin{aligned} BetP^{\Omega}(\omega_1|x_0, y_0) &= 0.416, \\ BetP^{\Omega}(\omega_2|x_0, y_0) &= 0.584. \end{aligned}$$

We observe that observation x_0 tends to point to class ω_2 (as $pl(x_0|\omega_2) > pl(y_0|\omega_1)$), whereas y_0 points to class ω_1 (as $pl(y_0|\omega_1) > pl(y_0|\omega_2)$). Using the LC method, the two observations counterbalance each other, and the resulting pignistic probabilities are close to 0.5. To a lesser extend, this is also true with the CI method. However, using the MCD and CI methods, the plausibilities $pl(y_0|\omega_1)$, $pl(y_0|\omega_2)$ are significantly closer to unity than the plausibilities $pl(x_0|\omega_1)$, $pl(x_0|\omega_2)$ calculated from the same method[2], reflecting weak knowledge of the distribution of Y in both classes, due to the small number of training examples in $\mathcal{L}_y$. As a consequence, the impact of observation y_0 is less important, resulting in a higher pignistic probability assigned to class ω_2.

In this simple example, the final decision does not change. However, it is clear that the three methods for computing the plausibilities of observations in each class may lead to different decisions. As the MCD and CI methods take into account the different sizes of $\mathcal{L}_x$ and $\mathcal{L}_y$ and, as a consequence, give less importance to sensor S_y in the decision, they may be expected to result in better performance. It also seems that the MCD method will perform even better than the CI method. All this will be verified in the following section.

[2]With these two methods (MCD and CI), the obtained bell-shape of the distribution is much fatter for $pl(y_0|\omega_1)$, $pl(y_0|\omega_2)$ than for $pl(x_0|\omega_1)$, $pl(x_0|\omega_2)$.

Numerical Experiment

To study the impact of the MCD and CI method for computing the class-conditional plausibilities in the above scheme, a numerical experiment was carried out as follows. The following conditional distributions of X and Y were assumed to be: $f(x|\omega_1) \sim \mathcal{N}(0,1)$, $f(x|\omega_2) \sim \mathcal{N}(2,1)$, $f(y|\omega_1) \sim \mathcal{N}(0,1)$, $f(y|\omega_2) \sim \mathcal{N}(0.5,1)$.

A test set of 1000 examples for each class was randomly generated. The size of $\mathcal{L}_x$ was fixed to $n_x = 1000$, while the size n_y of $\mathcal{L}_y$ was successively set to 10, 50 and 100. For each value of n_y, the following procedure was repeated 50 times:

- Generate randomly a learning set $\mathcal{L}_x$ of size $n_x = 1000$;
- Generate randomly a learning set $\mathcal{L}_y$ of size n_y;
- Classify each test example using the approach described in Section 2.4 and each of the following options
  - use only $pl(x_0|\omega_k)$ ($k = 1, 2$) computed using the MCD method (the decision is the same if the LC or CI method is used instead);
  - use $pl(x_0|\omega_k)$ and $pl(y_0|\omega_k)$ ($k = 1, 2$) computed using the LC method;
  - use $pl(x_0|\omega_k)$ and $pl(y_0|\omega_k)$ ($k = 1, 2$) computed using the MCD method;
  - use $pl(x_0|\omega_k)$ and $pl(y_0|\omega_k)$ ($k = 1, 2$) computed using the CI method;
- Compute the error rates err_x, err_{LC}, err_{MCD} and err_{CI} using the four methods.

The results are shown in Figure 2.16. We can see that both the MCD and CI methods significantly outperform the LC method, especially for small values of n_y. For $n_y = 50$ and $n_y = 100$, all three methods take advantage of information from sensor S_y, as they reach significantly lower error rates than that those obtained using sensor S_x alone. For $n_y = 10$, the LC method exhibits very poor performances and a very high variance. In contrast, the MCD and CI methods have uniformly good performances for all values of n_y, and a much lower variance for small sample size. For $n_y = 10$, the variance of the MCD method is lower than that of the CI method, but its mean error rate is higher. For larger values of n_y, both methods show very similar error rates, but the MCD method has a lower variance, which makes it more reliable.

2.5 Conclusion

In this chapter, two methods have been introduced to build belief functions from a sample of data, in the special case where the variable X of interest is a random variable, and the only available information is assumed to consist in an independent random sample from the unknown distribution P_X of X. Our methods differ from the solution suggested in [123] and [45], by the fact that they take two particular things into account. The first point is that, as the information we dispose of is incomplete, the obtained belief function should be less informative than P_X. The second is that the more observations we dispose of, the closer to P_X the obtained belief function should be *and vice versa*. Imposing these two requirements to be satisfied makes the proposed solutions more robust than those that do not take the sample size into account. This was illustrated through a comparative classification example. An application of these techniques to novelty detection will be shown in Chapter 4.

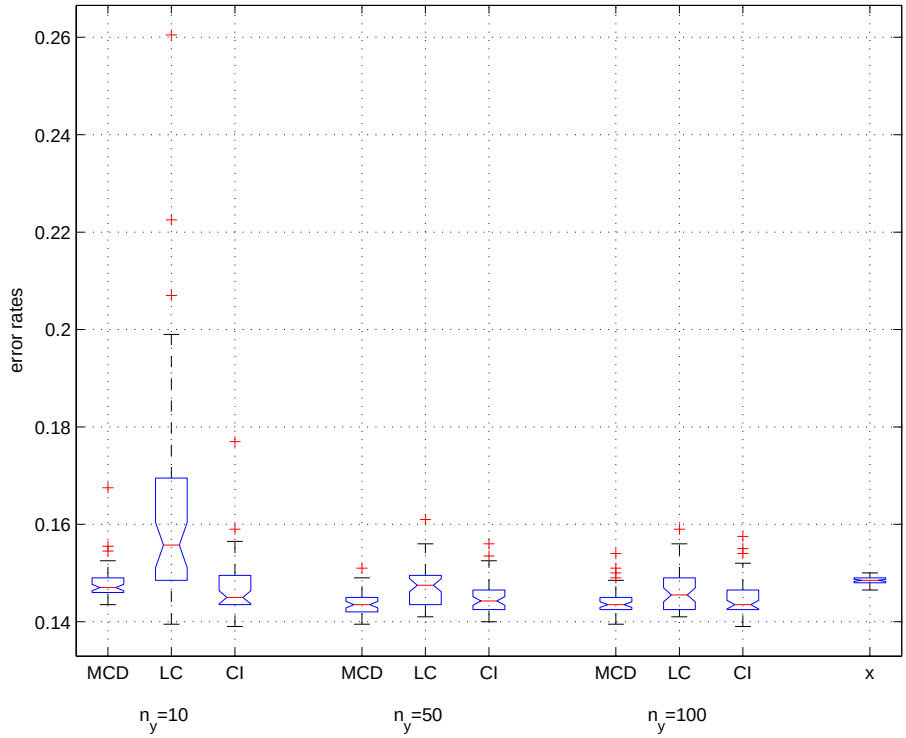


Figure 2.16: Box plots of error rates for the LC, MCD and CI methods as well as sensor S_x alone (rightmost box), for different sizes n_y of training set $\mathcal{L}_y$. Each box plot represents a distribution over 50 trials. Each box has lines at the lower quartile, median, and upper quartile values. The whiskers extending from each end of the box show the extent of the rest of the data (except outliers represented separately). Boxes whose notches do not overlap indicate that the medians of the two groups differ at the 5% significance level.

Part II

One-class classification

One-class classification

Contents

3.1 Introduction 79
3.2 Desired properties 79
3.2.1 Generalization ability 79
3.2.2 Robustness 80
3.2.3 Computational qualities 80
3.3 Taxonomy 80
3.3.1 Density based techniques 81
3.3.2 Boundary based approaches 81
3.3.3 Reconstruction approaches 81
3.3.4 Clustering-based approaches 82
3.3.5 Summary 82
3.4 1-class classifiers 82
3.4.1 Density based approaches 82
3.4.2 Density based approaches 84
3.4.3 Clustering-based approaches 86
3.4.4 Reconstruction-based approaches 87
3.4.5 Boundary-based approaches 90
3.5 Conclusion 97

Summary

In this chapter, a review of existing one-class classification techniques will be undertaken. We consider two classes: the positive or reference class, whose data are available for training, and the negative class, gathering all other classes. This review is limited to techniques which can be applied to any domain and any kind of data, and concentrates on the problem where data of only one class are available for training.
Desired properties of the classifiers are detailed first. They include generalization ability, robustness and computational qualities. The generalization ability (GA) of a classifier is its ability to differentiate normal, previously unseen data, from novel data. A classifier with good GA will show a good trade-off between the false positive and false negative rates (data classified as belonging to the positive class while they belong to the negative class and vice versa). Other important points to be considered include a robustness towards the number of considered features and reasonable performance in case of a low or very high number of samples. Additionally, classifiers should be robust against noise. Over-fitting should be avoided, e.g. by the choice of boundaries that are not to tight around the data points, and the model itself should have low variance. The required computational qualities are low complexity, easy on-line training and minimization of the number of parameters.
In a second section, a taxonomy of novelty detection techniques is given, through which the various ranges of application, advantages and drawbacks of the different approaches will be easy to identify. We mainly distinguish between four categories: density-based techniques, boundary based-approaches, reconstruction methods and clustering-based techniques.
This taxonomy is then used to describe the main algorithms that correspond to each category, from Parzen Windows to Support Vector Machines, including k-Nearest Neighbours, Extreme Value Theory, Convex Peeling, Principal Component Analysis, Neural Networks, etc.

Résumé

Dans ce chapitre, une revue des méthodes de classification à une classe existantes est entreprise. Elle se limite aux techniques qui peuvent être appliquées à n'importe quel domaine et n'importe quel type de données, et se concentre sur les problèmes pour lesquels des données en provenance d'une classe seulement sont disponibles pour l'apprentissage. On considère deux classes: la classe de référence dite positive, et la classe négative, rassemblant toutes les autres classes possibles.
Les propriétés désirées pour les classifieurs sont introduites en premier. Elles incluent la capacité de généralisation, la robustesse et les qualités algorithmiques. La capacité de généralisation d'un classifieur est sa capacité à différencier des données normales, mais non rencontrées précédemment, de données atypiques. Un classifieur avec une bonne capacité de généralisation permet d'obtenir un bon compromis entre les taux de faux positifs et de faux négatifs (données classifiées comme appartenant à la classe positive alors qu'elles proviennent de la classe négative et vice versa). Les autres points importants à prendre en compte sont la robustesse envers le nombre de dimensions utilisées et des performances raisonnables en cas d'échantillon d'apprentissage de très petite ou de très grande taille. De plus, un classifieur doit être robuste vis à vis du bruit de mesure. Le sur-apprentissage doit être évité, notamment par le choix de frontières qui ne sont pas trop "serrées" autour des points, et le modèle lui-même doit avoir une faible variance. Les qualités algorithmiques requises sont une faible complexité, une facilité d'apprentissage en ligne et la minimisation du nombre de paramètres.
Dans une seconde section, une taxonomie des techniques de détection de nouveauté est introduite, à travers laquelle les différentes applications, avantages et inconvénients

des différentes approches seront faciles à identifier. Nous distinguerons principalement quatre catégories: les techniques basées sur l'estimation de densité ou de frontière, les méthodes de reconstruction et les techniques de clustering.
Cette taxonomie est utilisée ensuite pour décrire les principaux algorithmes correspondants à chaque catégorie, des fenêtres de¨Parzen aux séparateurs à vaste marge en passant par les k plus proches voisins, la théorie des valeurs extrêmes, l'effeuillage convexe, l'analyse en composantes principales, les réseaux de neurones, etc.

3.1 Introduction

According to Barnett and Lewis [13], one-class classification consists in the detection of *observations (or subsets of observations) which appear to be inconsistent with the remainder of the (studied) set of data*. It is a very old, cross-disciplinary issue. It has therefore been widely studied in the literature, under such various names as *outlier detection, novelty detection, one-class classification, noise detection, deviation detection and exception mining*. We will generally refer to the problem as novelty detection or one-class classification. The existing techniques are grounded on the detection of the patterns that appear to deviate markedly from other members of the sample under consideration. How markedly the deviation needs to be before it matters and through which characteristics the data is assembled into a sample is application-dependent.

Applications of novelty detection are numerous. Amongst them, fraud or intrusion detection, image analysis, medical condition monitoring and fault detection may be mentioned. As we will concentrate on the latter in chapter 4 we will use the vocabulary of fault detection throughout. The data from the studied (or training) set will thus be termed normal or un-faulty, while deviating data will be called abnormal, faulty or novel. The studied set of data will correspond to a specific state, or a set of states of the system under consideration.

In this chapter, we will review the existing methods for novelty detection. Desired properties will first be described, and a taxonomy of the different approaches will be introduced. A synthetic description of each technique will then be given, with particular emphasis on those that will be referred to in the sequel.

This survey is mainly based on [77, 78, 57, 82]. However, a few of the techniques mentioned in these reviews are not recalled here because they perform too poorly and present a number of important drawbacks. Techniques that can only apply to a particular kind of data (such as character strings) or application (e.g. character recognition or network intrusion) have also been ignored as well as techniques designed for non stationary processes (e.g. sequence based). Finally, discrimination techniques have been ruled out as they do not fit in the constraints of our work, i.e., they require data of several classes for training. We concentrate here on the situation where data of only one class are available.

3.2 Desired properties

3.2.1 Generalization ability

The most important feature of a good novelty detection technique is its generalization ability [77, 82], that is to say, the system should be able to discriminate normal, previously unseen data, from novel data.

A one-class system may also be seen as a two-class system in which one of the classes (denoted ω_0) serves as reference, and corresponds to a well identified state of the system under consideration, while the other class $\omega_1 = \overline{\omega_0}$ gathers all other possible states of the system. Two kinds of errors may thus be defined.

An error of type-I occurs if a pattern of class ω_0 (positive) is deemed to come from ω_1 (negative). On the contrary, an error of type-II is encountered when an object of class ω_1 is classified as belonging to ω_0. Type-I errors are sometimes referred to as false negative, and type-II errors as false positive.

Intuitively, it is easy to realize that the minimizations of these two types of errors are conflicting. In effect, a system that classifies all patterns as objects of class ω_0 will minimize the false negative rate α but maximize the false positive rate β. On the other hand, a classifier that considers all patterns as novel (i.e., belonging to ω_1) will have

the opposite drawback. A novelty detection technique should thus show good trade-off between false positive and false negative rates.

In addition, many novelty detection techniques suffer the curse of dimensionality, that is to say, they are not robust against the number of considered features. Robustness towards this point should thus be considered in the choice of method, as well as reasonable performance in the case of a low number of samples, and computational tractability in case of a very high number of samples.

3.2.2 Robustness

Additionally, novelty detection techniques should be robust against noise. In other words, the classification results should not change drastically under a slight perturbation of the training data, nor should they be too training-sample dependent. This implies two different points.

First, the classifier's model (or the statistic it uses for classification) should have a low variance.

Moreover, many novelty detection techniques are based on the estimation of a boundary around the training data. Hence, the second point is that, in order to avoid over-fitting, the boundary should not be too tight around the data points, i.e. boundaries with a lesser degree of flexibility should be applied. Similarly, density estimates should be built in such a way that the transition from high densities too low densities is not too sharp.

3.2.3 Computational qualities

Depending on the application field, different computational considerations may be taken into account in the choice of the method:

- *Minimization of the nunber of parameters:* The higher the number of user-defined parameters, the more difficult it will be to tune the method. Moreover, it has also been noted that algorithms that imply a high number of parameters are more prone to over-fitting.

- *Low complexity:* For on-line use, novelty detection mechanisms should have the smallest possible computational complexity. However, low complexity should not come at the expense of too high a need for random access memory, which would slow the process down once the memory capacity of the computer is exceeded.

- *Easy on-line training:* For processes that might evolve slowly over time, the possibility of an on-line update of the model is appealing. Therefore, a system should be able to use the result of the classification of test samples for retraining. For rapidly evolving systems, different techniques should be considered altogether.

3.3 A taxonomy of novelty detection techniques

Outliers may arise in the distribution of a training set of data because of human error (mislabeling), sensor imprecision or malfunctioning, mechanical or other faults, change in a system behaviour, fraudulent behaviour, natural deviation of populations, etc. The building of the different techniques that may be used to detect novelties has been widely influenced by the type of outliers they were primarily designed for. However, most techniques may be used for the identification of a variety of outlier types. We will establish a taxonomy of methods, through which the various ranges of application, advantages

and drawbacks of the different approaches will be easy to identify. We will then use this taxonomy to describe the main algorithms that correspond to each category.

The oldest methods are mostly distance-based. In other words, they involve the calculation of some sort of distance between a point to be classified (a test point) and the training data. The test point is then deemed to be novel if its distance to the training data is greater than some threshold.

3.3.1 Density based techniques

Early techniques often also are density-based and parametric, i.e., they involve some assumption about the training data distribution (parametric density estimation, Gaussian mixture models, ...). This limits their domain of applicability but lessens their computational complexity and the amount of memory they require. For obvious reasons, non parametric techniques have been developed, with nearly opposite pros and cons. Non parametric techniques, typically Parzen Windows, solve the problem of the adequacy of the data to their supposed distribution, but drastically increase computational complexity and memory needs in comparison to parametric methods. Depending on the application field, the training data and the amount of prior knowledge about their distribution, parametric or non parametric techniques will yield better results.

3.3.2 Boundary based approaches

As the number of considered features increases, and for a training sample of fixed size, the proportion of available data associated with the relatively low density tails of the distribution increases as well. This makes it difficult to determine the importance that must be given to the tails of the distribution. Moreover, it has been shown that the number of data points required for an accurate estimate of a distribution increases as a power of the number of dimensions, leading to prohibitive computational costs. Boundary-based approaches, as opposed to density-based approaches, have thus been developed to tackle this problem. In such methods (e.g., support vector machines), a boundary is built around the training samples, and the classification is based on the calculation of a distance between the tested point(s) and the boundary. The boundary may obviously be defined with a limited number of training points, namely those situated at the edge of the distribution. After these points have been selected during the training phase, the remaining points are not needed any more for the test of an unknown data point, which solves the previously mentioned computational problem.

3.3.3 Reconstruction approaches

Another way to look at the novelty detection problem is to wonder what underlying structure actually relates the training points with one-another, rather than trying to define in what area of space they lie (as density- and boundary-based methods do). This introduces a third type of approach, namely the reconstruction approaches and, for example, principal component analysis. They imply the building of a model of the training data. The adequacy of tested observations to the model is then evaluated through the reconstruction error which reflects how well a given point may be represented by the model. The better a test point fits the model, the more likely it was generated by this model. These techniques also led to the introduction of novelty indexes that are not distances in the mathematical sense.

3.3.4 Clustering-based approaches

Finally, clustering-based approaches (e.g. k-means), are another way of solving the problem of reducing the computational costs of classification for large training data sets. They divide the training set into subsets of similar patterns termed clusters. Each cluster may then be represented by a unique pattern called prototype. It is then possible to work with prototypes only, hence considerably reducing computational costs. Novelty may be measured by calculating the distance of tested patterns to prototypes, measuring the reconstruction error made when a test pattern is represented by a prototype, or checking the labels of neighbouring prototypes.

3.3.5 Summary

In conclusion, four categories of one-class classification techniques may be defined: *density-based*, *boundary-based*, *reconstruction-based*, and *clustering-based* approaches. Amongst density-based techniques, *parametric* and *non parametric* methods should be differentiated. All four kinds of approaches may also be separated between *distance-based* algorithms, and mechanisms relying on *other types of novelty indexes*. We will see that, depending on the employed novelty measure, clustering approaches may sometimes be considered as reconstruction-based approaches.

In the next section, we will describe the most important one-class classification techniques according to the above described taxonomy.

3.4 Description of the main one-class classification techniques

3.4.1 Non-parametric density based approaches

Histograms

One of the simplest outlier detection techniques, as well as one of the oldest, is the *histogram*. It is one of the most widely used non parametric density estimates. However, the shape of the obtained density estimate may vary quite a lot depending on the considered number of bins, i.e., the necessary bin-width to cover the distribution. They are computationally inexpensive but very sensitive to the curse of dimensionality.

Histograms are sometimes called *naive estimators* [114], and constitute a good introduction to the idea of kernel density estimation. Moreover, the multivariate case being a generalization of the monovariate case, we will start with a description of this estimator in a one dimensional situation.

By definition of a probability density, if a real variable X has density f then

$$f(x) = \lim_{h \to 0} \left(\frac{1}{2h} \mathbb{P}(x - h < X < x + h) \right), \quad x \in \mathbb{R}. \tag{3.1}$$

For a given h, it is of course possible to estimate $\mathbb{P}(x - h < X < x + h)$ by the proportion of observations that lie in the interval $[x - h, x + h]$. A natural estimator $\hat{f}$ of f is thus given by the choice of a small number h and expression:

$$\hat{f}(x) = \frac{1}{2Nh} \sum_{i=1}^{N} \mathbb{1}_{[x-h,x+h]}(x_i), \tag{3.2}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h} \mathcal{K} \left(\frac{x - x_i}{h} \right), \tag{3.3}$$

where N is the size of the sample and $\mathcal{K}(x) = \frac{1}{2}$ if $|x| \leq 1$, 0 otherwise. In the sequel, Equation (3.3) will be termed "naive estimator".

It follows from (3.3) that the estimator is built by placing a "box" of width $2h$ and height $(2Nh)^{-1}$ around each observation and summing over the observations to obtain the estimate.

Parzen windows or kernel density estimation

The naive estimator is not entirely satisfactory from the point of view of density estimation for the representation and interpretation of data. In effect, it follows from definitions (3.2) and (3.3) that f is discontinuous at some points x and has a null derivative everywhere else. However, is easy to generalize the naive estimator in order to overcome some of these drawbacks. Let us replace the weight function $\mathcal{K}$ by a kernel function satisfying:

$$\int_{-\infty}^{+\infty} \mathcal{K}(x)dx = 1. \tag{3.4}$$

This is *the kernel density* or *Parzen window estimator* [114].

Most of the time, $\mathcal{K}$ is a symmetrical probability density function, such as the normal density. By analogy with the naive estimator, the Parzen estimator of kernel $\mathcal{K}$ is defined by:

$$\hat{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} \mathcal{K}\left(\frac{x - x_i}{h}\right), \tag{3.5}$$

where h is the width of the Parzen windows, also termed bandwidth, kernel-width or smoothing parameter. Provided kernel $\mathcal{K}$ is non negative and satisfies condition (3.4), (i.e. $\mathcal{K}$ is a probability density), $\hat{f}$ will be a probability density itself. Furthermore, $\hat{f}$ will inherit the continuity and differentiability properties of kernel $\mathcal{K}$. The technique is easily generalized to the multivariate case by replacing the monovariate representation window with a volume in dimension d. The estimator may then be written

$$\hat{f}(\mathbf{x}) = \frac{1}{Nh^d} \sum_{i=1}^{N} \mathcal{K}\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \tag{3.6}$$

The shape of the kernel and the value of the smoothing parameter widely influence the final result.

The bandwidth being fixed once and for all, an artificial perturbation may arise in the tails of the distribution when the estimator is applied to very sparse distributions. Numerous methods have been suggested to locally adjust the width of the kernel [93, 94, 133] so that data may be better represented, in particular in regions of low density. Figure 3.1 illustrates the influence of the kernel width on the shape of the final density estimate.

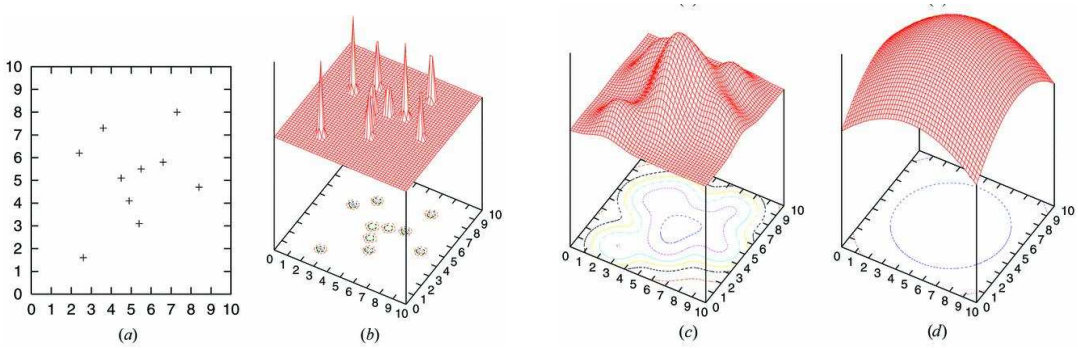


Figure 3.1: Influence of the kernel on KDE
Influence of the kernel width on the shape of Parzen density estimate

The Parzen window estimator requires all the observations vectors to be stored and used for each calculation, it is thus computationally expensive. On the other hand, it is quite robust against the presence of outliers in the training sample as they only influence the density estimation very locally. The main advantage of this method is its ability to estimate arbitrary distributions.

K-Nearest Neighbours (kNN)

The *k-nearest neighbour (kNN)* method may be seen as an attempt to improve the Parzen window estimator by locally adapting the smoothing parameter to the data [114]. In the case of the Parzen window estimator, the window width is fixed once and for all. In the kNN technique, on the other hand, it is the number of data that will be included in the window that is fixed. Let us define the distance between two real points x and y as the usual euclidean distance $\|x - y\|$, and for each x_i, the distances, in increasing order, of the observations to sample point x will be termed, respectively, $d_1(x) \leq d_2(x) \leq \ldots \leq d_n(x)$. The kNN estimator thus reads

$$\hat{f}(x) = \frac{k}{2Nd_k(x)}. \tag{3.7}$$

In order to explain this definition, let us suppose the density at point x is $f(x)$. Now, for a sample of size n, we expect about $2rNf(x)$ observations to fall in interval $[x-r; x+r], r > 0$. By definition, exactly k observations fall in interval $[x - d_k(x); x + d_k(x)]$. Consequently, an estimator of the density at point x may be obtained through $k = 2d_k(x)N\hat{f}(x)$, which can be re-arranged so that it looks exactly like the definition of the k^{th} nearest neighbour. In the multivariate case, the window of width $2d_k(x)$ is again replaced with a volume $\mathcal{V}$ of a domain $\mathcal{D}(x)$. The number of points in the volume $\mathcal{V}$ is fixed, and the estimator thus reads

$$\hat{f}_n(x) = \frac{k/N}{\mathcal{V}(\mathcal{D}(x))}. \tag{3.8}$$

While the Parzen estimator is based on the number of observations lying in a box of fixed width, and centered at the point of interest, the kNN estimator is inversely proportional to the size of the box needed to contain a given number of observations. In the tail of the distribution, the distance $d_k(x)$ will be greater than in the main part of the distribution, and the problem of lack of smoothing in the tails will be tackled.

Nevertheless, it is more sensible to outliers in the training data than the kernel density estimator. As for the latter, the sensitivity of the method to noise depends upon the choice of the smoothing parameter (k for the kNN, h for Parzen windows).

3.4.2 Parametric density-based approaches

Parametric density estimation

The expression *parametric density estimation* generally refers to any density estimation technique relying on an hypothesis on the general form of the data distribution.

The simplest technique is to assume the form of the distribution is known, for example the data follow a Gaussian distribution. The parameters of the distribution are then estimated from the training points, e.g., by the technique of maximum likelihood estimation. The general form of the distribution might be actually known or suggested by an expert as a reasonable assumption. A tested data point may then be deemed to be novel if the cdf at this point –under the hypothesis that it comes from the same distribution than the training samples– is under some threshold or if the likelihood function at this point –again under the hypothesis that it comes from the same distribution as the training samples– is under some threshold. Classical hypothesis tests may also be used to compare the distribution of a set of test points with that of the training points.

Gaussian Mixture Models (GMM)

In order to improve the adequacy of the data to the considered model, it may be considered that the data were not drawn from a given distribution but a mixture of several distributions. The best known example of this technique is the *Gaussian Mixture Model* [129, 84], in which data are deemed to be drawn from a mixture of Gaussian distributions of (possibly) different means and variances. The parameters of the Gaussian mixture can be found, e.g., by maximizing the likelihood over the training set using the EM algorithm.

Extreme value theory (EVT)

The above described methods aim at describing the distribution of the core of the training data. Another way to look at the problem is to model the distribution of the extremes. *Extreme value theory* thus aims at describing the limit distribution for maxima or minima. It was first developed with the idea of assessing risk for highly unusual events such as 100-year flow, stock market crashes, etc. The solution came from the observation that in real life situations, the tails of distributions, where by definition the extremes lie, are often fatter (heavier) than predicted by classical distributions such as the normal distribution and its cousins.

Suppose that X is a random variable. Let $\{X_1, X_2, \ldots, X_n\}$ be n independent realizations of X. Define the extreme observations as

$$Y_n = \max(X_1, X_2, \ldots, X_n)$$
$$Z_n = \min(X_1, X_2, \ldots, X_n)$$

The extreme value theory deals with the distributional properties of Y_n and Z_n as n becomes large.

There are two main theorems in EVT, which both deal with the convergence of extrema [100, 107]. The difference between the two theorems is due to the nature of data collection. For the first theorem, the data are generated in full range.

The *Fisher-Tippett* or *extremal-types theorem* (1928) [51] states that

Theorem 1. *If exist constants $a_n > 0$ and $b_n \in \mathbb{R}$ such that*

$$\frac{Y_n - a_n}{b_n} \xrightarrow{d} F \text{ as } n \to \infty \tag{3.9}$$

for some non-degenerate distribution F (i.e., $F(x)$ is continuous and has an inverse), then F must be one of the only three possible 'extreme value distributions', namely the Gumbel, Fréchet or Weibull distribution.

The so-called *generalized extreme value* (GEV) distribution was developed to embed the three above mentioned distributions, which come as special cases of the GEV distribution.

The second theorem –termed Pickands-Balkema-de Haan theorem [96, 12]– deals with data that were generated only when they surpass a given threshold (POT or Peak Over Threshold models), and corresponds to certain real life situations, for example in the insurance business, where only losses (pays out) above a certain threshold are accessible to the company. It may be expressed as follows.

Theorem 2. *As the threshold L becomes large, the distribution of the excesses over a threshold L tends to the* Generalized Pareto distribution, *provided the underlying distribution F belongs to the domain of attraction of the Generalized Extreme Value distribution.*

The role of these two theorems is similar to that of the central limit theorem for averages. The latter states that the limit distribution of the arithmetic mean of a sequence of iid random variable is the normal distribution no matter what the distribution of the variable may be. The extremal types theorems are similar in scope in that they tell us that the limiting distribution of the extremes always takes the same form, whatever the distribution of the parents from which the considered extremes were drawn.

Novelty may then be detected when the cdf associated with the considered extreme value distribution is above some threshold at the observed (test) point.

The technique is highly sensible to the presence of very few novel points in the training set. However, Roberts [102, 103], using the work of Fisher and Tipett [51], managed to apply the method to real data sets with very good performance.

Pros and cons of parametric methods

Parametric techniques are computationally inexpensive, as only the parameters of the applied method need to be stored. However, the assumption that the data follow a given distribution is not always reasonable, nor does it always permit to represent the whole distribution accurately, leading to a number of misclassifications. Moreover, parametric methods are often sensible to noise and outliers in the training data, as the variance of the distribution is often one of the parameters that needs to be estimated from the training set.

3.4.3 Clustering-based approaches

The k-nearest-neighbour estimator may be used for classification in many different ways. As mentioned in paragraph 3.4.1, it can be used as a classical density estimator, and a point will be termed novel if the estimated density of the training data is low at that point. It can also be seen as a simple clustering technique. In this case, the decision of classifying a point as novel can be made in many different ways [57].

First, let us define the *generalized kNN* of a point x [127]. The set NN_k of x and its generalized kNN may be built as follows:

1. Find the non generalized nearest neighbour n_1 of x and define $NN_1 = \{x, n_1\}$;
2. Calculate the distances between n_1 and all the other points in the training set except x, and the distances between x and all the other points in the training set except n_1; select the point that minimizes these distances, and call it n_2; define $NN_2 = NN_1 \cup n_2 = \{x, n_1, n_2\}$;
3. Calculate the distances between each of the points in NN_2 and all the other points in the training set, and select the point that minimizes these distances; it will be n_3 and $NN_3 = NN_2 \cup n_3$;
4. Iterate point 3 until you get NN_k.

A tested point may be classified as novelty

- if its k nearest neighbours lie within a distance d greater than some threshold;
- or if the chaining distance between the tested point x and its generalized kNN (i.e., the sum of distances between each of the points in NN_k and its own nearest neighbour in the set) is greater than some threshold.

Alternately, it is also possible to first divide the space in a series of cells [99] and use these to define the distance between points or groups of points. Then,

- if a given cell c and its adjacent[1] neighbours contain more than k points, then the

[1] *Adajacent cells* here means cells having at least one common edge or one common vertex with cell c.

points in cell c are assigned to the positive class;

- or if the number of points lying in cells less than a predefined distance apart from c is less than k, then all points in cell c are considered as novel.

The word "neighbour" may be understood either as "points in the training set" or "prototypes".

k-means and k-medoids

The *k-means algorithm* separates the data into k clusters by maximizing the inter-cluster distance while minimizing the intra-cluster distance. These combined two objectives constitute the objective function to be optimized. It is most of the time based on the Euclidean distance, but other distances may be used instead, such as the Mahalanobis distance. Each cluster is then best represented by its mean (or barycentre), which may be used as prototype for the cluster. A test point is deemed to be novel if it lies at a distance of the nearest prototype further than some threshold.

The k-medoids algorithm is a variant of the k-means in which a cluster is represented by its median rather than its mean. It is hence less sensible to outliers in the training data than the k-means algorithm.

Storage space required for training may be high, but classification through the k-means or k-medoids algorithm is computationally inexpensive as only prototypes need to be considered. The quality of the result is highly dependent on the chosen number k of clusters, which must be defined by the user.

Fuzzy clustering techniques

Fuzzy clustering techniques most of the time work in a similar fashion as their crisp equivalent except that class membership is defined by membership degrees rather than a crisp label. Each data point can thus be assigned a degree of membership to each of the clusters. All the novelty detection rules described above can be adapted to work with a fuzzy membership function. An additional rule is to detect novelty whenever the considered sample does not belong to any of the available clusters (i.e. its degree of membership to each of the clusters is less than some threshold). A fuzzy output intrinsically gives a representation of how precise and certain the obtained classification is.

Amongst other algorithms, the fuzzy k-means [16] may be mentioned. The algorithm depends upon a parameter $1 < m < \infty$ that controls how fuzzy the cluster are allowed to be. As m tends to 1, the algorithm converges towards the hard k-means. Conversely, as m tends to infinity all the prototype converge towards the centroid of all training data. The fuzzy k-means algorithm generally outperforms the hard k-means algorithm in that it lessens its tendency to getting stuck in local minima of the objective function during training.

3.4.4 Reconstruction-based approaches

Clustering techniques

Clustering algorithms may be seen as reconstruction-based approaches when they include prototyping. In effect, the reconstruction error associated to the representation of a given point by the associated prototype may be used as novelty index and compared to some threshold for classification purpose.

Supervised neural networks

Neural networks are mostly known as a discrimination technique. However, there exist neural networks that are trained to reproduce their input features as their output. They are termed *auto-encoders* or *auto-associators* [18, 10, 11, 106]. Figure 3.2 shows a formal neuron as introduced by Mc Culloch and Pitts [83]. It is a computing unit that carries out a weighted sum of the input signals to which it applies a transfer function $\mathcal{H}$ in order to obtain the answer of the cell to the input stimuli. With the notations of Figure 3.2, the exit of a neuron is defined as : $a = \mathcal{H}\left(\sum_{i=1}^{d} w_i x_i\right)$, where w_i is the weight associated to the input x_i of the neuron, and $\mathcal{H}$ is the transfer function.

A neural network is constituted of several layers of neurons connected to each other. The neurons of the entrance layer take the different features of the pattern to be studied as inputs. The neurons of each internal layer take the outputs of the neurons of the previous layer as input. Either the output layer is constituted of only one neuron, or the outputs need to be merged so that classification may be performed. In the sequel, the merging process associated with the last of these two possibilities will be considered as a particular, additional, one neuron layer.

The choice of the number of layers, the number of neurons in each layers, and the transfer function associated to each neuron widely influences the performance of the network. There exist a number of techniques and heuristics to specify the architecture of the network. The initial weights of the neurons are most of the time randomly initialized.

Training then consists in adapting the weights so that the output of the network reproduces the input as best possible. It is performed through calculating an objective function that compares the obtained and desired results and consequently updating the weights of each neuron until the objective function is optimized. Classification then consists in presenting a test pattern to the network and collecting the associated output.

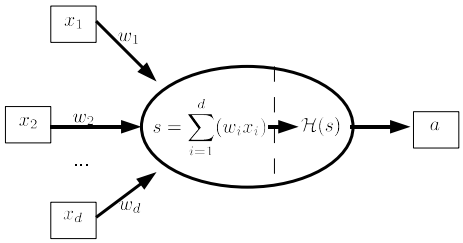


Figure 3.2: A formal neuron

Only observations whose structure is close to that of the training patterns are reproduced accurately. The reconstruction error, calculated as the Euclidean or Mahalanobis distance between the input and output of the network, hence becomes a novelty index. Numerous other variants of neural networks have been proposed in the literature.

Remark 10. *There is an interesting connection between principal component analysis and auto-associators [18]. Let us consider a three layer neural network with a hidden layer including p neurons. The training of this network as an auto-associator leads to an optimal map*[2] *between the inputs and outputs. It may be shown that this map is the combination of two operations: the projection onto the subspace spanned by the first p eigenvectors of the data's covariance matrix is first performed, and then the data are projected back into the original space. Hence, up to an arbitrary linear transformation, the activities in the hidden layer are identical to the principal component of the data as will be defined in the next section (Section 3.4.4).*

[2]In this case, the optimal map is the one that allows the best reconstruction of the inputs of the network

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [52, 111] allows the mapping of high dimensional data on a lower dimensional space through linear orthogonal projection, thus providing a more compact representation. The optimal subspace retains most of the variance of the original data and may have a significantly lower dimension than the original space. It is obtained by the identification of the dependences between the observations.

It can be shown that the optimal subspace of dimension $p \leq d$ (where d is the dimension of the original space) is defined by the first p eigenvectors of the covariance matrix of the data, the eigenvectors $\{\mathbf{v_1}, \ldots, \mathbf{v_d}\}$ being sorted by decreasing associated eigenvalue.

However, simple PCA relies on the hypothesis of a linear correlation between the data, which is obviously not always the case. The introduction of a kernel function $\mathcal{K}$ solves the problem: it acts as though the data had undergone a prior transformation k from the original space into a feature, Hilbert space, in which they are linearly correlated. The resulting technique is termed Kernel Principal Component Analysis (KPCA).

PCA based novelty detection is traditionally performed via the monitoring of different types of error, amongst which the squared prediction error (SPE), Hotelling's statistic (or T^2) and the reconstruction error. A pattern is considered novel if any of the monitored errors is above some threshold.

Lee et al. [71] derived formulae for the SPE and T^2 statistics in the KPCA framework, thus adapting the monitoring technique to highly non-linear data. Recently, Hoffmann [58] also provided a calculation of the reconstruction error (here termed kernel reconstruction error or KRE) adapted to KPCA. These statistics have the property of being small for data drawn from the same distribution as the data that were used to build the KPCA model, and greater for data drawn from a different distribution. They can thus be used as a novelty measure.

As KPCA includes the possibility of dimensionality reduction, it is interesting as a one-class classifier in the case where the dimensionality of the data to be studied is high. However, the method remains computationally complex as the determination of the novelty measure often implies the calculation of the distance of the test pattern to each of the training data. Noise and outliers influence performances as they influence the estimation of variances and covariances. A sequential version of the algorithm proposed by Whenming et al. [137] lessens the need for storage space but slows the training and classification processes down.

Mathematical formulation: For PCA, it can be shown that the optimal linear transformation is a projection on the subspace spanned by the eigenvectors of the sample's covariance matrix. Such a projection is optimal in the sense that, for a given dimension of the subspace, it has maximum variance. For KPCA, let us introduce a kernel function of the form: $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. The covariance matrix in feature space is:

$$C = \frac{1}{n} \sum_{i=1}^{n} \Phi(\mathbf{x}_j) \Phi(\mathbf{x}_i)^T \tag{3.10}$$

and the corresponding eigenvalue problem can be expressed as:

$$\lambda \mathbf{v} = C\mathbf{v}, \tag{3.11}$$

where the eigenvector $\mathbf{v}_1$ corresponding to the largest eigenvalue λ_1 becomes the first component of the feature space. It can be shown that this eigenvalue problem comes down to another eigenvalue problem:

$$n\lambda\alpha = \mathcal{K}\alpha, \tag{3.12}$$

with $\mathcal{K}_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{ff} = [\mathbf{ff}_1, \ldots, \mathbf{ff}_n]^T$, and $\mathbf{v} = \sum_{i=1}^{n} \mathbf{ff}_i \Phi(\mathbf{x}_i)$. The n eigenvectors solution of (3.12) are denoted $\mathbf{ff}_j$, j=1,...,n, and their i^{th} element is denoted $\alpha_{j,i}$. Let $\mathbf{x}$ be a datum vector in the original space. The projection of $\mathbf{x}$ in the KPCA-subspace will be denoted $\mathbf{t}$, and its elements t_j are termed principal components of vector $\mathbf{x}$. The following relation holds:

$$\begin{aligned} \mathbf{t}_j &= \left\langle \mathbf{v}_j, \Phi(\mathbf{x}) \right\rangle \\ &= \sum_{i=1}^{n} \alpha_{j,i} \mathcal{K}(\mathbf{x}_i, \mathbf{x}). \end{aligned} \tag{3.13}$$

Dimension reduction is then obtained by selecting the principal components retaining the greatest part of the variance, say 90% for example. If a number $p < n$ of principal components is selected, then $\mathbf{t}' = [\mathbf{t}_1, \ldots, \mathbf{t}_p]$, will be used instead of $\mathbf{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_n]$ in further calculations.

Hotelling's statistics (T^2) is the sum of the normalized squared scores, and is defined as

$$T^2 = [\mathbf{t}_1, \ldots, \mathbf{t}_p] \Lambda^{-1} [\mathbf{t}_1, \ldots, \mathbf{t}_p]^T, \tag{3.14}$$

where $\mathbf{t}_k$ is the projection of $\Phi(x)$ onto eigenvector $\mathbf{v}_k$ and can be calculated from function k, Λ^{-1} is the diagonal matrix of the inverse of the eigenvalues associated with the retained principal components, and p is the number of retained principal components. The SPE in feature space may be defined as:

$$SPE = \sum_{j=1}^{n} \mathbf{t}_j^2 - \sum_{j=1}^{p} \mathbf{t}_j^2. \tag{3.15}$$

The squared distance between a point in feature space and the centre of the KPCA subspace is termed (kernel) reconstruction error and denoted KRE. Hoffman demonstrated that:

$$KRE(x) = \mathcal{K}(x,x) - \frac{2}{n} \sum_{i=1}^{n} \mathcal{K}(x, x_i) + \frac{1}{n^2} \sum_{i,j=1}^{n} \mathcal{K}(x_i, x_j) - \sum_{\ell=1}^{p} f_\ell(x)^2 \tag{3.16}$$

$$\tag{3.17}$$

where:

$$f_\ell(x) = \sum_{i=1}^{n} \alpha_{\ell,i} \left[\mathcal{K}(x, x_i) - \frac{1}{n} \sum_{r=1}^{n} \mathcal{K}(x_i, x_r) - \frac{1}{n} \sum_{r=1}^{n} \mathcal{K}(x, x_r) + \frac{1}{n^2} \sum_{r,s=1}^{n} \mathcal{K}(x_r, x_s) \right], \tag{3.18}$$

where d is the dimension of the original space ($x_i \in \mathbb{R}^d, \quad i = 1, \ldots, n$), and ℓ is the index that denotes the ℓ^{th} eigenvector, with $\ell = 1$ for the eigenvector with the largest eigenvalue.

3.4.5 Boundary-based approaches

Self Organizing Maps (SOM)

Kohonen's *Self Organizing Maps (SOM)* [66, 1, 88, 90] are unsupervised one-layer neural networks in which the nodes are (initially) organized in a particular way, most of the time on a rectangular or hexagonal mesh. They are constituted of only one layer. During training, the weights of the nodes are first randomly initialized. Then, a pattern is presented

as input to the network: in other words, some distance d between this pattern and each neuron is calculated. The winning neuron is selected as the one the training pattern is closest to. Its weights are then updated so that it gets even closer to the training pattern it has just been checked against. To a lesser extent, the weights of its neighbours are modified as well, proportionally to their distance to the winning neuron. The process is repeated iteratively with all the training patterns until the weights of all neurons stabilize.

In case only one class was used for training, the distance of a test point to the activated node works as a novelty measure.

Alternately, a test pattern is repeatedly presented to the network and the weights are updated until the weights stabilize. The stabilization"time"(number of presentations of a given pattern before the weights of the network stabilize) may be used as a novelty index (with comparison to some threshold). This technique presents the drawback that classification is much more computationally demanding than with the previous method (it is in fact as demanding as the training phase).

Note that the network may be graphically represented in such a way that the distance D between two neighbouring neurons is used as the distance between their representations, then a node moves each time its weights are updated. When a node moves during training, the neighbouring nodes move as well. Hence, the mesh tends to reproduce the configuration of the input patterns.

Once trained, a SOM is computationally undemanding, but training involves distance calculations between each training pattern and all the neurons in the network. The number, type (transfer function) and topology of the neurons influence the performances. SOM are often appreciated for the very self explanatory representation of the data they provide.

Other neural networks

A number of other (sometimes fuzzy) types of neural network based algorithms have been described in the literature.

Very similar to the SOM are the so called *Habituation Network* [79, 80]. Habituation is the mechanism by which the brain learns to ignore repeated stimuli. Habituation network simulate this behaviour: they are trained in such a way that the nodes of the network are less and less activated by the training patterns. Then, the more a tested point activates the network, the more likely it is to be novel.

Amongst other types of neural network algorithms, Adaptive Resonance Theory (ART) and fuzzy ART [54, 21, 22], Learned Vector Quantization (LVQ) [66] and Cooper's Restricted Coulomb Energy network (RCE) [101], may be mentionned. All three of these algorithms use hyperspheres to surround the training classes and produce closed decisions boundaries. They differ in the way they determine the number and sizes of the hyperspheres. During training, an ART algorithm fixes the size of the hyperspheres, an RCE algorithm fixes the position and LVQ fixes the number. Depending of the application, some outperform the others, but they have similar pros and cons on a general point of view. Their more important drawback is that they all depend on the user-set parameter that fixes –or influences– the number of hyperspheres for the quality of the results.

Minimum volume ellipsoid (MVE)

The *Minimum Volume Ellipsoid (MVE)* method [105] fits the smallest possible ellipsoid around a given percentage p of the data distribution model, thus representing the most densely populated region of space. Subsets of p% of the data are examined. The smallest ellipsoid enclosing each subset is calculated in order to find the subset that minimizes the volume occupied by the data, that is to say, the subset with the smallest associated

ellipsoid. The best subset (smallest volume) is then used to calculate the covariance matrix of the data and the Mahalanobis distance to all the data points. An appropriate cut-off distance is then estimated, and the observations within distances that exceed that threshold are declared outliers.

As this technique fits boundaries around specific percentages of the data, it is insensible to outliers in the training data.

Convex peeling (or data depth) [13, 74, 69] is a similar, but somewhat more sophisticated approach. In this method, nested convex hulls, respectively enclosing all training data points, all data but the points on the first hull, all data but the points on the first and second hulls, all data but the points on the three outermost hulls, etc are used to peel away the records on the boundary of the data distribution. The convex hull of a set of points $\mathbf{X}$ in $\mathbb{R}^d$, denoted $CH(\mathbf{X})$, is the intersection of all convex sets in $\mathbb{R}^d$ containing $\mathbf{X}$.

Sequential algorithms permit to determine nested convex hulls for a set of points, thus limiting the computational complexity. The Convex peeling algorithm therefore works on massive data sets.

A way of determining which points may be deemed outliers is to consider that nested hulls define a depth based partial order on the data points and that the p% least deep points are outliers, for a user defined value of p. A slightly more sophisticated version of this is the so-called *Balloon plot* (see Figure 3.3). It is obtained by blowing the hull that includes 50% of the data (denoted $CH(\mathbf{X})_{.5}$) by a factor 1.5. Let $V_{.5}$ be a set of vertices for $CH(\mathbf{X})_{.5}$. The balloon $B_{.5}$ for outlier detection is:

$$B_{.5} = \{y_i \text{ such that } y_i = x_i + 1.5(x_i - CHPM), \quad x_i \in V_{.5}\}, \tag{3.19}$$

where CHPM is the *convex hull peeling median*, defined as follows:

Definition 20. (CHPM) *Recursive peeling leads to the inner most point or points. If there is more than one, then the average of the deepest points is the CHPM of the data set, otherwise the deepest point is the CHPM.*

The balloon plot may be seen as some sort of multidimensional boxplot without whiskers.

Big changes in the volume of two successive hulls may also be used to detect outliers (see Figure 3.4).

Both the MVE and Convex Peeling techniques suffer the curse of dimensionality as convex hulls become more and more difficult to determine when dimension increases.

One-class Support Vector Machines (SVMs)

One-class support vector machines (SVM), also termed $\nu-$SVM or Support Vector Data Description (SVDD), were introduced by Vapnik [132], Schölkopf [112] and Tax and Duin [128] as a way to estimate the support of a distribution. The underlying idea is that there is no need to estimate the exact density of a population in order to be able to determine whether a new measurement originates from the same distribution or not. The specification of the support of the distribution, i.e., the region of space containing a large fraction of points drawn from that distribution, is sufficient for most applications and much more computationally efficient than full density estimation.

The principle of SVMs has two different geometrical interpretations. Schölkopf first introduced the method as the determination of the hyperplane that separates the training data from the origin with maximal margin. This is done through the definition of a function f that is positive in the support of the distribution and negative elsewhere. Given a learning set $x_1, \ldots, x_n$, it can be shown that an optimal function may be defined as:

$$f(x) = \sum_{i=1}^{n} \left(\alpha_i \mathcal{K}(x_i, x) - b\right), \tag{3.20}$$

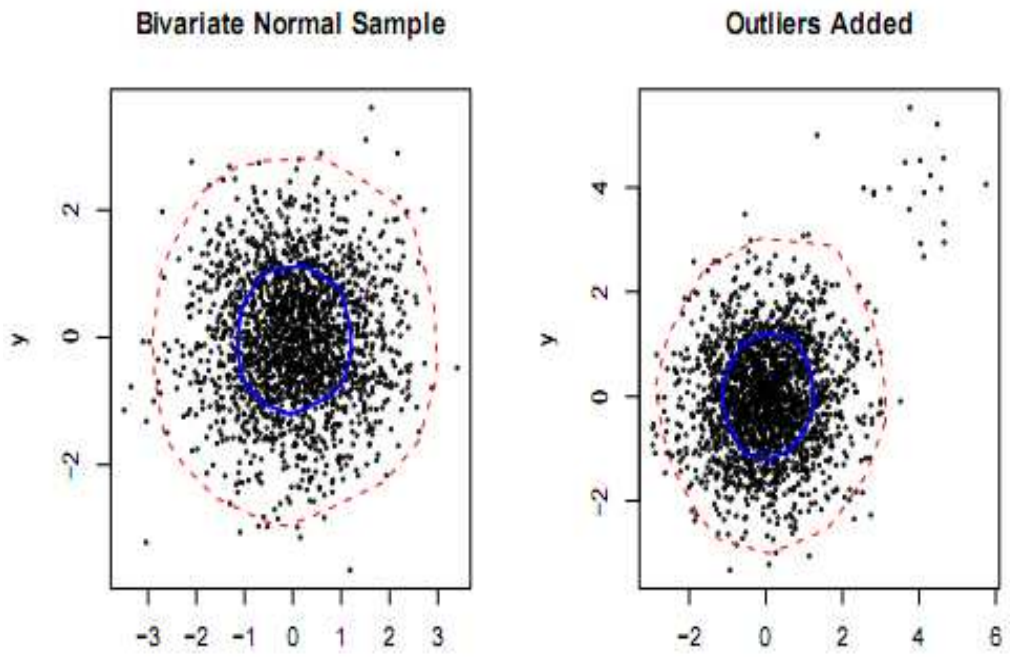


Figure 3.3: Outlier detection through balloon plot(reproduced from [70])

and satisfies the constraints $0 \leq \alpha_i \leq (\nu n)^{-1}$ and $\sum_i \alpha_i = 1$, where b is a scalar parameter called bias, ν is an hyperparameter, and $\mathcal{K}(\cdot, \cdot)$ is a kernel function. Function f can be determined by solving a quadratic programming problem. A pattern x is rejected if $f(x)$ is negative (or smaller than some threshold).

Tax and Duin then showed that comparable results may be obtained through defining the smallest hypersphere enclosing all training data. This hypersphere is defined by a centre a and a radius R.

In both cases, the use of kernel functions allows the definition of implicit mappings leading to more flexible descriptions. For instance, the data might not be separable from the origin with an hyperplane in the original space, and, –even in cases were it is possible–, it would not make sense to do this in the original space as it would very badly serve the purpose of novelty detection. Hence, the data need to be mapped to a (possibly high dimensional) feature space in which they are linearly separable from the origin. The resulting boundary will not be linear once mapped back to the original space.

Additionally, the boundary is made even more flexible by the introduction of slack variables. Let us consider the hypersphere interpretation: in order to account for the eventuality of outliers in the training set, the distance of each data to the centre of the sphere should not be strictly smaller than R, but larger distances should be penalized.

Furthermore, it can be shown that the parameter ν is both an upper bound on the fraction of outliers (i.e. errors) and a lower bound on the fraction of support vectors thus controlling the trade-off between precision and generalization capacity. Note that, when $\nu = 1$ and the kernel can be normalized as a density in input space, then (3.20) is exactly equivalent to a Parzen-window density estimate [110]. In Vapnik's original formulation [132], a parameter C was used instead of ν, and can be shown to be approximately equivalent to $(\nu n)^{-1}$.

Example 14. *Figure 3.5 shows a simple two-dimensional data set of $n = 100$ learning vectors, with a contour plot of function $-f(x)$ computed using (3.20), with a Gaussian kernel $\mathcal{K}(x, y) = \exp(\|x - y\|^2/(2\sigma^2))$. We can see that the support of the distribution is well approximated by contour lines of $f(x)$. A novelty detection rule may be implemented by rejected patterns for which*

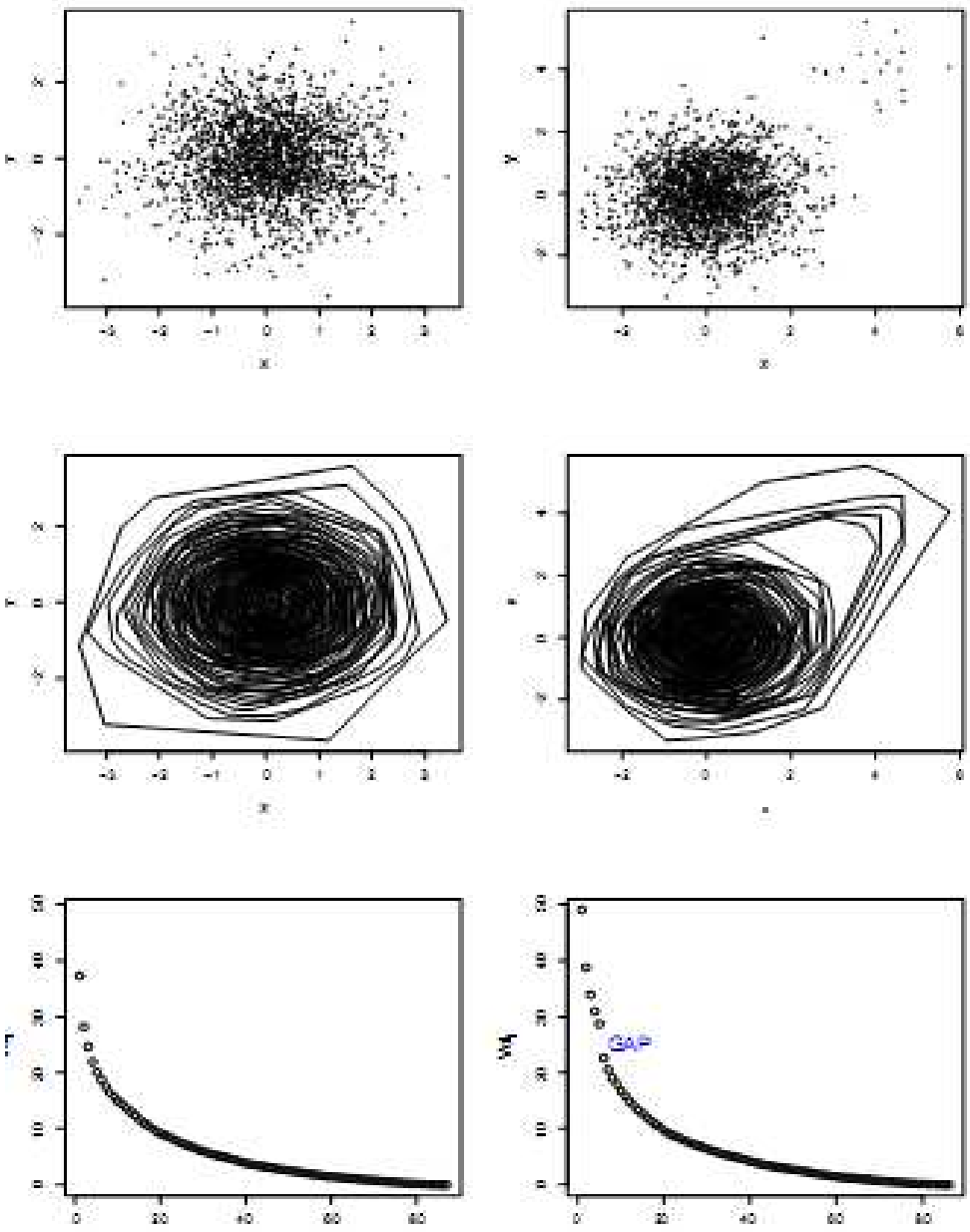


Figure 3.4: Outlier detection through nested convex hull volume change(reproduced from [70])

$-f(x)$ is higher than some threshold.

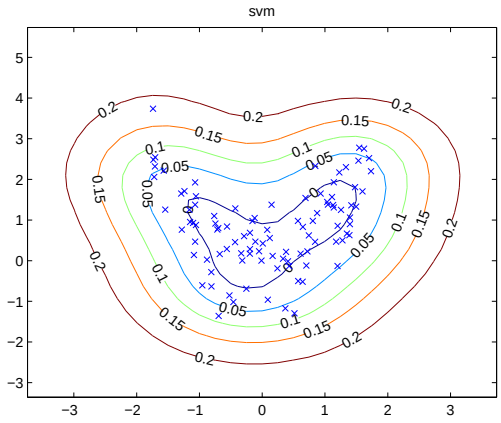


Figure 3.5: SVM-based one-class classification
Data set of Example 14, and contour plot of the SVM novelty measure $-f(x)$, with $\nu = 0.5$.

The training phase of SVMs is computationally demanding, as many iterations over all training patterns need to be processed for the selection of the support vectors and associated weights. Though recent works [17, 75, 76] show how to tackle this problem in the case of multi-class SVMs, to the author's knowledge, no solution is available in the literature for the one class problem yet.

Classification, on the other hand, is fairly undemanding.

The smoothing parameter h of Kernel $\mathcal{K}$ may be difficult to tune depending on the data. Cross-validation is often used for this purpose, thus increasing computational needs during training. Overall, SVMs have been found to show very good performances with respect to other techniques.

Mathematical formulation

Hyperplane technique We will first consider two classes of data respectively labelled $+1$ (positive class or class $+1$) and -1 (negative class) and determine the hyperplane H that separates the two classes with maximal margin.

Let us consider an hyperplane of equation $H : \langle w, x\rangle - b = 0$, or equivalently, $\langle w, x\rangle = b$. A plane supports a set of data (or class) if all points of the class are on one of its side. If the hyperplane H is to separate the two classes, then w and b should be such that $\langle w, x\rangle > b$ for all the points x whose label is $+1$ and $\langle w, x\rangle < b$ for the others.

Let us now suppose that the smallest value of $\langle w, x\rangle - b$ for the points of the positive class is κ, then $\langle w, x\rangle - b \geq \kappa$, and κ can be set to 1 as positive rescaling will leave the problem unchanged. Equation $\langle w, x\rangle - b = 1$ then defines an hyperplane H_{+1} that supports the data from class $+1$. Similarly, we may require $\{\langle w, x\rangle - b \leq -1\}$ for the data in class -1 and $\{\langle w, x\rangle - b = -1\}$ defines an hyperplane H_{-1} parallel to H_{+1}.

The maximization of the distance or *margin* between H_{+1} and H_{-1} will help us find the plane furthest from both sets of data, which can be defined as the plane H, parallel to H_{+1}, situated exactly at mid-distance between H_{+1} and H_{-1} (see Figure 3.6).

The distance between the supporting planes H_{+1} and H_{-1} is $\gamma = 2/\|w\|_2$. Hence, maximizing the margin is equivalent to minimizing $\|w\|_2 /2$ in the following minimization program [15]:

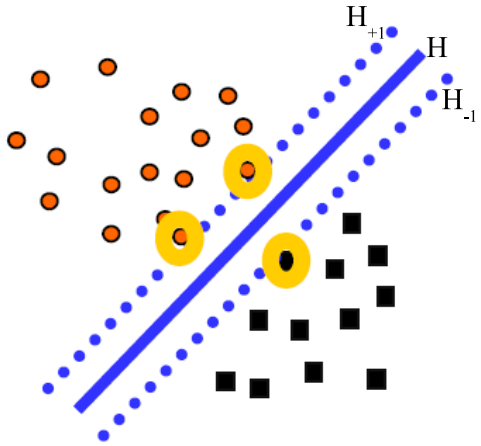


Figure 3.6: Separating hyperplanes H_{+1}, H_{-1} and H for a toy example(reproduced from [15])

$$\min_{w,b} \frac{1}{2} \|w\|^2, \tag{3.21}$$

$$\text{subject to: } y_i \left(\langle w, x \rangle_i - b \right) \geq 1, \tag{3.22}$$

where $y_i = 1$ when x_i belongs to the positive class and $y_i = -1$ otherwise.

In the one-class problem, the origin stands for the only point in the negative class. Thus, H is indeed the hyperplane that separates the training data from the origin with maximal margin. The constraint becomes $\langle w, x \rangle_i - 1 \geq 1$ as the dot product of any vector with the origin will be null.

The introduction of slack variables ξ_i, $i = 1, \ldots, n$, and of a kernel function $\mathcal{K}$ in place of the classic dot product leads to:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + \frac{1}{\nu n} \sum_i \xi_i - b, \tag{3.23}$$

$$\text{subject to: } \langle w, \Phi(x_i) \rangle \geq b - \xi_i, \tag{3.24}$$

$$\xi_i \geq 0, \forall i, \tag{3.25}$$

where ν is an hyperparameter that is used to weight the influence of each term in the objective function. As already mentioned, it may be shown to be an upper bound on the fraction of outliers (i.e. errors) and a lower bound on the fraction of support vectors at the same time.

The solution to this problem is the saddle point of the Lagrangian:

$$L(w, \xi, b, \alpha, \beta) = \frac{1}{2} \|w\|^2 + \frac{1}{\nu n} \sum_i \xi_i - b - \sum_i \alpha_i (\langle w, \Phi(x_i) \rangle - b + \xi_i) - \sum_i \beta_i \xi_i, \tag{3.26}$$

where α_i and β_i are Lagrange multipliers.

The Kuhn-Tucker conditions then lead to the following dual formulation:

$$\min_{\alpha} \frac{1}{2} \sum_{ij} \alpha_i \alpha_j \mathcal{K}(x_i, x_j), \tag{3.27}$$

$$\text{subject to: } 0 \leq \alpha_i \leq \frac{1}{\nu n}, \tag{3.28}$$

$$\sum_{i=1}^{n} \alpha_i = 1; \tag{3.29}$$

where $w = \sum_i \alpha_i \Phi(\mathbf{x}_i)$ [112] and $\mathcal{K}(x_i, x_j) = \langle \Phi(x_i).\Phi(x_j) \rangle$ [76]. Finally, novelty detection can be performed using

$$f(x) = \text{sgn}\left[\langle w, \Phi(x) \rangle - b\right] \tag{3.30}$$

$$f(x) = \text{sgn}\left[\sum_i (\alpha_i \mathcal{K}(x_i, x) - b)\right] \tag{3.31}$$

as a decision function. A pattern x is then deemed to be novel if $f(x)$ is negative.

Similar formulation can be derived for the hypersphere technique.

3.5 Conclusion

In this chapter, we have summarized the main novelty detection techniques. Tables 3.5 and 3.5 are an attempt to recapitulate the main features of the different methods, with respect to the desired properties described in Section 3.5. For a given algorithm, ticks and crosses respectively denote the properties about which positive and negative remarks have been made in the present chapter for that particular algorithm. This view is somewhat simplistic, but it gives an overview of the main advantages and drawbacks previously mentioned for each algorithm.

As outlined in [57], hybrid systems constitute the most recent development in outlier detection. They combine several classifiers and have been introduced as attempts to compensate for the drawbacks of a particular algorithm with another algorithm which may have complementary qualities. They may, of course, include more than two classifiers. Minimal redundancy should be observed as a rule, in order to avoid wasting resources, slowing processes and increasing complexity. Moreover, the combination of several techniques that may provide their result in different formats requires the representation of the outputs into a common framework that allows the combination and fusion of information, and possibly permits to handle information on the precision or certainty of the provided classifications.

Dempster-Shafer theory seems ideally fitted for such a task. However, until recently, the problem had not been studied in this framework. This task is undertaken in the next chapter. Building on previous work reported in [5, 6, 7, 8], we show how to convert the outputs of one-class classifiers such as one-class SVMs or KPCA into belief functions. Expressing one-class and multi-class classifiers in a common framework allows the combination of classifiers based on different numbers of classes, different features, different learning algorithms, or different datasets.

	Robustness				Generali--zation ability	Low No. of parameters
	No. of features	No. of training data	Noise in training data	Outliers in training data		
Histograms	✗	✓		✓	✓	✓
Parzen windows	✗	✗		✓	✓	✓
Parametric density estimation	✗	✓	✗	✗	✗	✓
GMM	✗	✓	✗	✗	✗	✓
EVT			✗	✗		
kNN		✓(if prototyping)	✗	✗		
kmeans				✗		
kmedoids				✓		
Fuzzy clustering techniques			✓	✓	✓	
Unsupervised NN					✓	
MVE and Convex Peeling		✗	✓		✗	
SVM			✓	✓	✓	
KPCA	✓	✗	✗	✗	✓	✗

Table 3.1: Pros and cons of the novelty detection techniques

	Low computational complexity and low storage requirement	
	Training	Classification
Histograms	✓	✓
Parzen windows	✗	✗
Parametric density estimation	✓	✓
GMM	✓	✓
EVT		
kNN	✗	✓(if prototyping)
kmeans		
kmedoids		
Fuzzy clustering techniques		
MVE and Convex Peeling		
SVM	✗	✓
Supervised NN		
KPCA	✗	✗

Table 3.2: Pros and cons of the novelty detection techniques

From the classifier output to belief functions

Contents

4.1 Introduction . . . 105
4.2 General approach . . . 105
4.3 Step 1 . . . 106
4.4 GBT solution . . . 110
4.5 The cognitive inequality . . . 110
4.5.1 Definition 1 . . . 110
4.5.2 Determining the LCBF satisfying a cognitive inequality of type I 114
4.5.3 Definition 2 . . . 116
4.5.4 Determining the LCBF satisfying a cognitive inequality of type II 117
4.6 CIneq-based solutions . . . 117
4.6.1 Model 1 . . . 117
4.6.2 Model 2 . . . 121
4.7 Discussion . . . 124
4.8 Examples . . . 126
4.8.1 Simple novelty detection: example 1 . . . 126
4.8.2 Simple novelty detection: example 2 . . . 128
4.8.3 Classifier Fusion Example . . . 128
4.9 Conclusion . . . 131

Summary

In this chapter, we will see how the work of Chapter 2 can be used together with one-class (and possibly other) classifiers in such a way that the outputs of the different classifiers may be expressed in a common framework, namely in the form of belief functions. The latter can then be compared or combined.
Let us consider a system that can only be in two possible situations: the reference state ω_0, or a condition ω_1 including all other possible states. The problem under consideration is the assessment of the hypothesis that the system is in state ω_0 when the only available information about the system is a sample of observations $\mathbf{x_1}, \ldots, \mathbf{x_n}$ of some variables, representative of the system state, conditioned on ω_0.
We suggest a three step scheme to solve this problem:

1. Build a novelty measure T from $\mathbf{x_1}, \ldots, \mathbf{x_n}$ and, given the observed sample $t_1, \ldots, t_n$ of T for the training data, build a belief function that quantifies our belief in future values of T drawn from the same distribution;

2. Build two belief functions that quantify our belief in T given that the system is in the normal state ω_0 or in any other state state ω_1, respectively;

3. Reverse the conditioning so as to build a belief function that quantifies our belief in the system state, given $T = t_*$.

The reader is referred to Chapter 2 for Step 1. Steps 2 and 3 are detailed in the present chapter, and different hypotheses are taken into consideration. A GBT-based solution is detailed first. Then, the concept of cognitive inequality is introduced and two models based on this notion are presented. Finally, several examples of applications are given.

Résumé

Dans ce chapitre, nous expliquons comment les travaux du chapitre 2 peuvent être utilisés avec les classifieurs à une ou plusieurs classes de manière à ce que les sorties des différents classifieurs puissent être exprimées dans un référentiel commun, sous la forme de fonctions de croyance. Ces dernières peuvent être ensuite combinées ou comparées.
Considérons un système qui peut se trouver dans deux situations seulement: un état de référence ω_0, ou une situation ω_1 qui comprend tous les autres états possibles. Le problème considéré est le test de l'hypothèse selon laquelle le système est dans l'état ω_0 lorsque la seule information disponible sur le système est un échantillon $\mathbf{x_1}, \ldots, \mathbf{x_n}$ de certaines variables représentatives de l'état du système, conditionné par rapport à ω_0.
Nous suggérons une procédure en trois étapes:

1. *Construire une mesure de nouveauté T à partir de $\mathbf{x}_1, \ldots, \mathbf{x}_n$ et, étant donnée les valeurs observées $t_1, \ldots, t_n$ de T pour les données d'apprentissage, construire une fonction de croyance qui quantifie notre croyance dans de futures valeurs de T issues de la même distribution;*

2. *Construire deux fonctions de croyance qui quantifient notre croyance dans la valeur de T sachant que le système est, respectivement, dans l'état de référence ω_0 ou dans n'importe quel autre état ω_1;*

3. *Renverser le conditionnement afin d'obtenir la fonction de croyance sur l'état du système connaissant la valeur $T = t_*$ de T.*

Le lecteur est renvoyé au chapitre 2 pour la première étape. Les étapes deux et trois sont détaillées dans le présent chapitre, et différentes hypothèses sont considérées. Une solution basée sur le théorème de Bayes généralisé est détaillée pour commencer. Ensuite, le concept d'inégalité cognitive est introduit, et deux modèles basés sur cette notion sont présentés. Finalement, plusieurs exemples sont donnés.

4.1 Introduction

As outlined in the previous chapter, there exist a wide variety of novelty detection algorithms, none of which is best in all situations and for all types of data. In industry, complex classification tasks such as, e.g., the sorting of letters in the post, chemical, mechanical or other system monitoring applications, etc, are carried out by existing classifiers and show good results. However, due to the ever increasing market competition, there is always a need to improve these results again and again. There seem to be no point in trying to propose yet another classifier, which will just be as good as any other, unless it is extremely application specific, and therefore not very attractive as it will not be very (easily) evolutive. Future seems to lie in hybrid systems, that combine several existing classifiers [57].

The combination of information is possible in many frameworks such as probability and possibility theory, fuzzy sets theory and belief function theory. In each of these frameworks, there exist a number of combination rules, to be chosen depending on whether the data to be combined are independent or not, equally reliable or not, etc. However, the application of those rules requires the information to be expressed in a common framework, and this is what Chapter 2 was mainly concerned with.

In this chapter, we will see how the work of Chapter 2 can be used together with one-class (and possibly other) classifiers so that the output of the different classifiers may be expressed in a common framework, namely in the form of belief functions. The latter can then be compared or combined. Hence, our purpose is not to propose a new novelty detection technique but to make the most of existing classifiers.

Building on previous work reported in [5, 8], we show how to convert the outputs of one-class classifiers such as one-class SVMs or KPCA into belief functions.

We will see that the use of belief functions allows a good exploitation of the available information. In effect, most classifiers provide crisp answers through thresholding a statistic representative of the state of the system under study. This leads to a loss of information, as no assessment of the decision (e.g., distance to the threshold) is carried out to the final output and thus to the user. The belief functions we obtain are both representative of the value of the statistic and of the uncertainty attached to it, thus allowing decision to be made in full knowledge.

Furthermore, expressing one-class and multi-class classifiers in a common framework allows to provide simple solutions to different fusion problems, such as the combination of several one-class classifiers based on different features or different learning algorithms, or the combination of one-class and multi-class classifiers built from different sets of data.

This chapter is organized as follows. A general approach to the problem, divided in three steps, is first presented. The reader is referred to Chapter 2 for Step 1. Steps 2 and 3 are then detailed for a first model, based on the GBT. Then, the concept of cognitive inequality is introduced, with two definitions, and the least committed belief functions satisfying each of this inequalities are presented. After that, two models based on the notion of cognitive inequality are presented. Steps 2 and 3 are detailed for each of these models. Finally, several examples of application are given.

4.2 General approach

One-class classification, or novelty detection, consists in assessing to what extent an observation may be deemed to correspond to a given model. In other words, given a set of observations $\mathbf{x_1}, \ldots, \mathbf{x_n}$ drawn from a given distribution, one-class classifiers are used to determine whether an unknown, new point, comes from the same distribution or not. Training the classifier to this task consists in building a novelty measure $T \in \mathcal{T} \subseteq \mathbb{R}$ as a function of $\mathbf{x_1}, \ldots, \mathbf{x_n}$ using, e.g., (3.20) or (3.16), whose value will be small in the

region of space containing the data $\mathbf{x_1}, \ldots, \mathbf{x_n}$, and larger as the distance to this region increases. Thus, T will be representative of the state of a system at a given time. A new observation is then rejected if the value of T exceeds some threshold.

The problem under consideration is the assessment of the hypothesis that a system is in class ω_0 when the only available information about the system concerns the distribution of statistic T conditioned on ω_0.

Let $\Omega = \{\omega_0, \omega_1\}$ be the set of possible states of the system, and our frame of discernment. Let $\mathbf{x_1}, \ldots, \mathbf{x_n}$ be a set of examples available for ω_0, the latter being the normal or reference state of the system under study. Additionally, let ω_1 be the set of all other states, for which no data is available. Having observed a value t_* of T, we want to define a bba $m^\Omega[t_*]$ on Ω, that quantifies our belief about the system state given t_*. Our approach is based on the following three-step procedure:

1. Build a novelty measure T from $\mathbf{x_1}, \ldots, \mathbf{x_n}$ and, given the observed sample $t_1, \ldots, t_n$ of T for the training data, build a *predictive belief function* m_*^T that quantifies our belief in future values of T drawn from the same distribution;

2. Build two belief functions $m^T[\omega_0]$ and $m^T[\omega_1]$ that quantify our belief in T given that the system is in the normal state ω_0 or in any other state state ω_1, respectively;

3. Reverse the conditioning in order to build a belief function $m^\Omega[t_*]$ that quantifies out belief in the system state, given $T = t_*$.

Different approaches to these three steps will be detailed in the sequel. They will all be illustrated with the following toy example.

Example 15. *(Ring data) Consider a two-dimensional data set that contains 1202 points. The first and main part of the data set (800 points) is built as follows. The first component of the data, x_1, is uniformly distributed over $[-5; 5]$, and the second component, x_2, is such that $x_1^2 + x_2^2 = 25$. Uniform noise is added to both components. This subset of the data is shaped as a ring. Another 402 points uniformly distributed on a circle of diameter 10 are added inside the ring, as shown in Figure 4.1.*

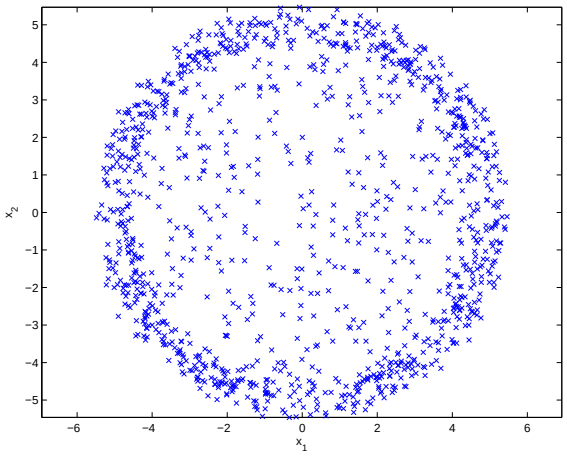


Figure 4.1: The ring data set.

4.3 Step 1: building of m_*^T

During step 1, given observations $\mathbf{x_1}, \ldots, \mathbf{x_n}$, a novelty measure T needs to be built. Any one-class classifier may be used for that purpose. As already mentioned, it is possible to

use (3.20) or (3.16), but simple density estimation or any other novelty measure may be used as well. From observations of the obtained values $t_1, \ldots, t_n$ of statistic T, a predictive bba $m_*^{\mathcal{T}}$ on future values of T drawn from the same distribution may be built. The reader is referred to Chapter 2 for a description of how to build such a belief function. In the sequel, and by construction, bba $m_*^{\mathcal{T}}$ will be either a discrete or a continuous belief function on $\mathcal{T} \subseteq \mathbb{R}$.

Example 16. *The result of the application of two of the techniques of Chapter 2 on the ring data is shown below.*

Figure 4.2 shows a contour plot of statistic $T = -f(\mathbf{x})$ computed using (3.20), with a Gaussian kernel $k(x,y) = \exp(\|x-y\|^2/(2\sigma^2))$. Parameter ν was set to 0.5, and the kernel bandwidth was defined as 0.7 times the mean Euclidean distance between two training vectors (It was suggested in [25] to use a half of this distance but adjusting to 0.7 times the distance gives better results in this particular case). The source code for the calculation of the SVM is Canu et al.'s and may be found at [19]. We can see that the support of the distribution is well approximated by contour lines of $f(x)$. A novelty detection rule may be implemented by rejecting patterns for which $-f(x)$ is higher than some threshold.

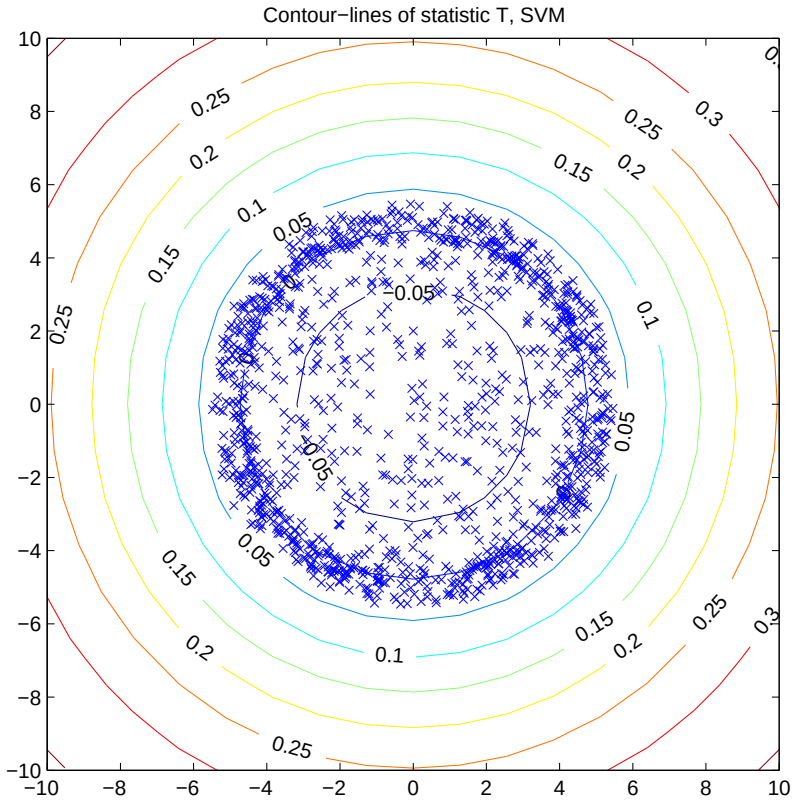


Figure 4.2: Contour lines of the SVM novelty measure $T = -f(x)$ for the ring data set.

Figure 4.3 shows the profile function and contour lines of pl_* obtained from T via Kriegler and Held's algorithm as described in Section 2.2.3. Note that Figure 4.3 shows the contour lines of pl_* with respect to the value of statistic T, and Figure 4.4 shows the contour lines of pl_* with respect to the position of the data. In the sequel, the method presented in Section 2.2.3 will be termed KH method.

Figure 4.5 shows the profile function and contour lines of pl_* as obtained via Cheng and Iles' continuous confidence band as described in Section 2.2.4 (the technique described there will be referred to as CI method). Figure 4.6 shows the position of the contour lines of pl_* with respect to that of the data.

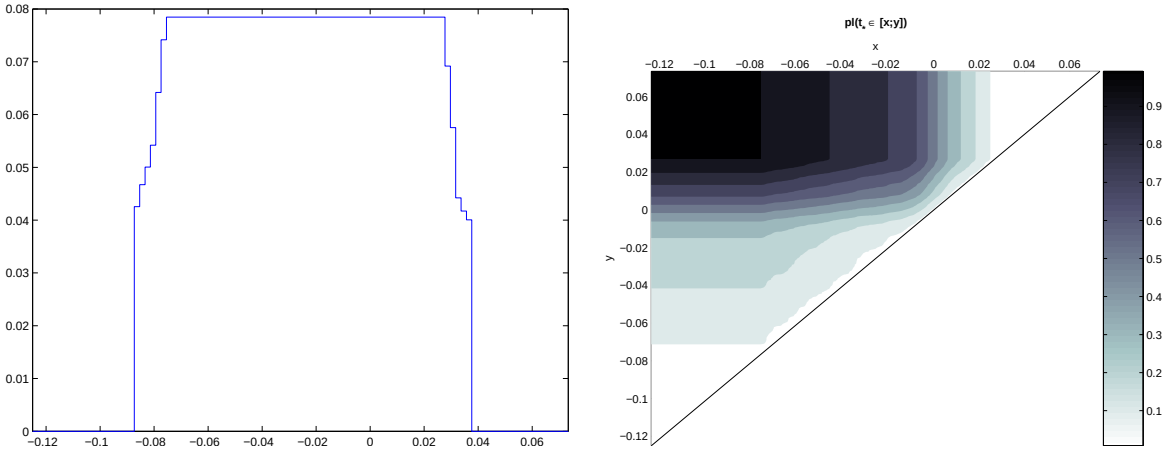


Figure 4.3: Plausibility function obtained via Kriegler and Held's algorithm (ring data).

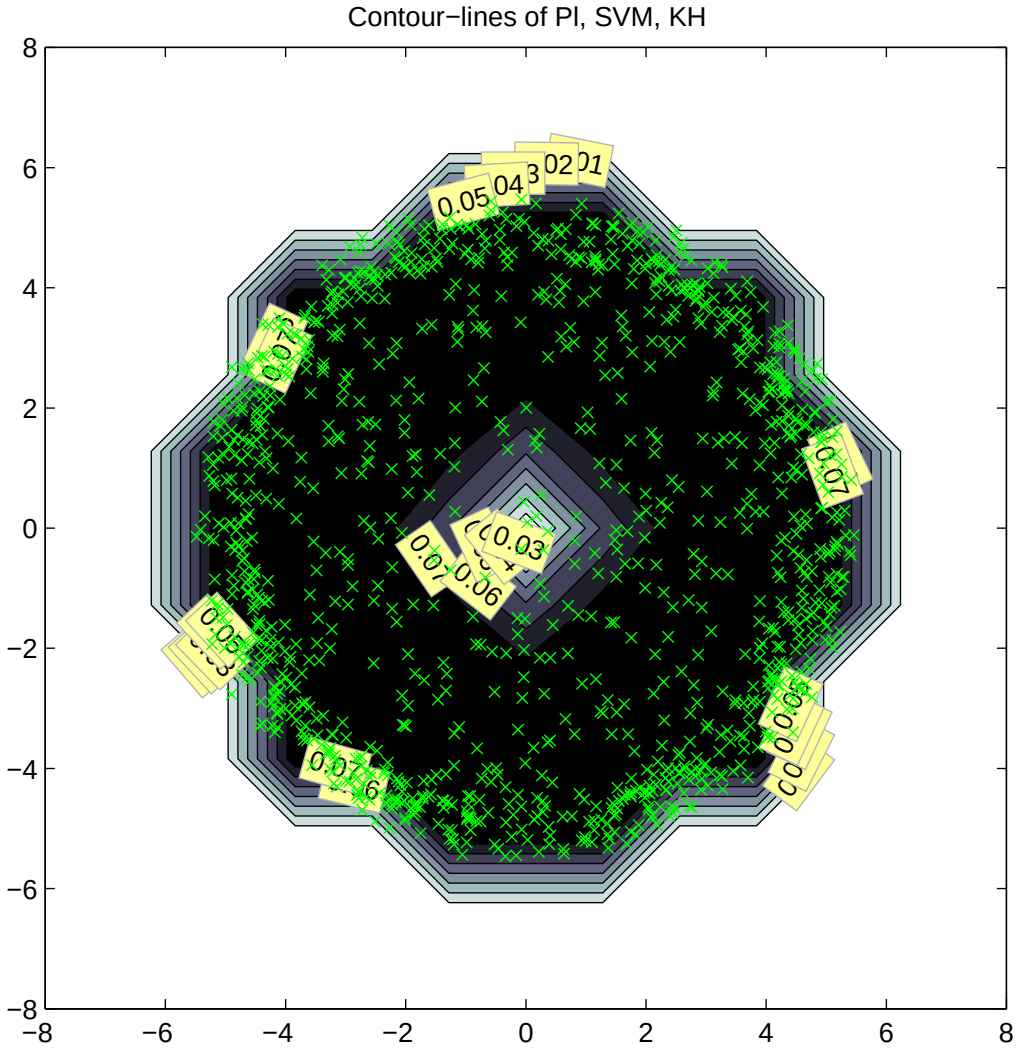


Figure 4.4: Contour-lines of the plausibility function obtained by the KH method, with respect to the position of the data (ring data).

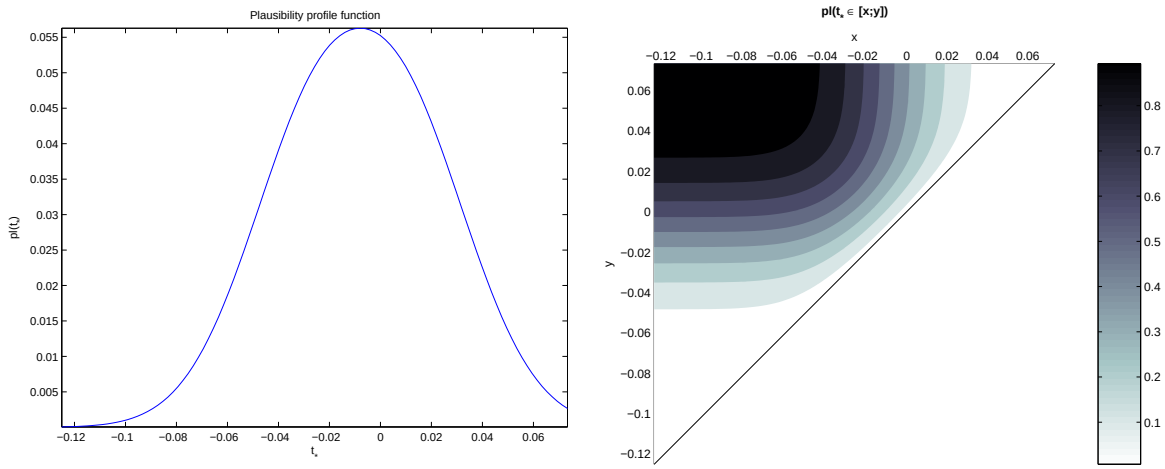


Figure 4.5: Plausibility function as obtained via the CI method (ring data).

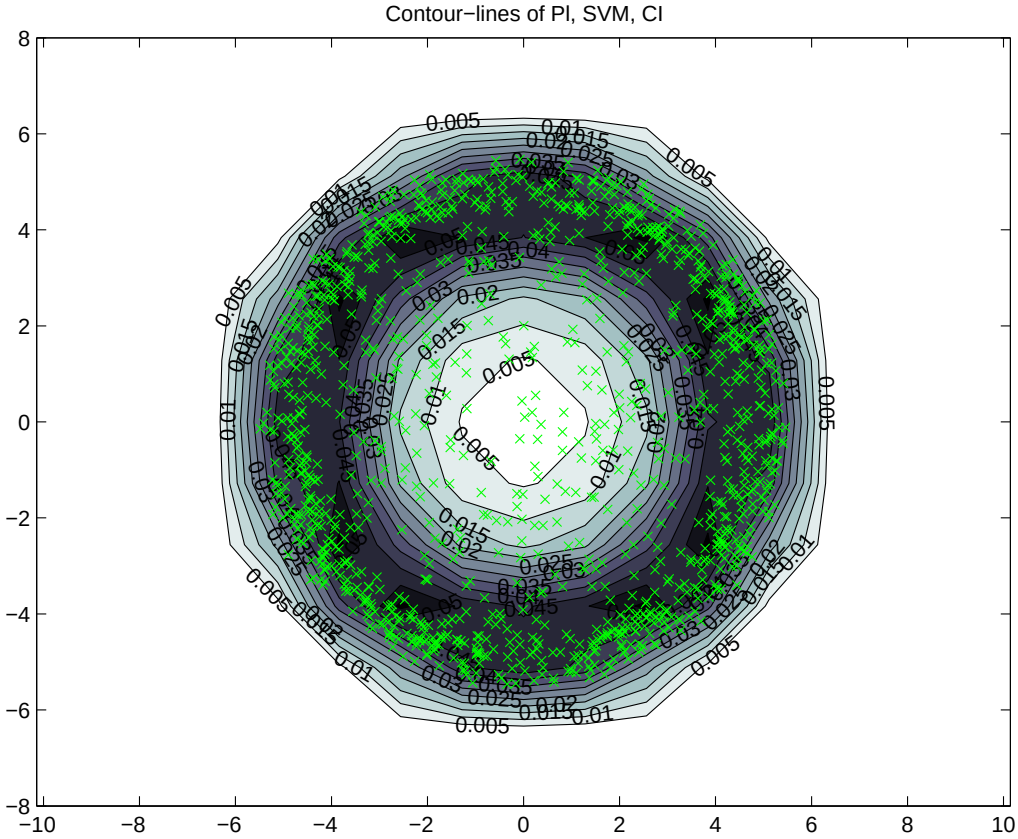


Figure 4.6: Contour-lines of the plausibility function obtained by the CI method, with respect to the position of the data (ring data).

4.4 Step 2 and 3: Solution using the GBT

In this section, we will introduce the straight-forward most (though maybe not the best) solution to perform Steps 2 and 3, which may be obtained through the GBT. In the sequel, $pl^{\mathcal{T}}[\omega_0]$ and $pl^{\mathcal{T}}[\omega_1]$ will be respectively denoted $pl_0^{\mathcal{T}}$ and $pl_1^{\mathcal{T}}$ or even pl_0 and pl_1 where there is no possible ambiguity. The associated bba will be respectively termed $m_0^{\mathcal{T}}$ and $m_1^{\mathcal{T}}$, or m_0 and m_1.

As we know nothing on the behaviour of T when ω_1 holds, the information we have at our disposal in this respect can be modeled with the vacuous belief function:

$$pl_1^{\mathcal{T}}(A) = 1, \quad \forall A \in \mathcal{T}. \tag{4.1}$$

On the other hand, $m_*^{\mathcal{T}}$ represents our belief on T drawn from the same distribution than those obtained when the system is in state ω_0. Consequently, we can take $m_0^{\mathcal{T}} = m_*^{\mathcal{T}}$. Equation (1.34) thus yields:

$$m^{\Omega}[t_*] = \{\omega_1\}^{pl_0(t_*)} \textcircled{\cap} \{\omega_0\}^{pl_1(t_*)}. \tag{4.2}$$

Hence, from (4.1) and (4.2),

$$\begin{aligned} m^{\Omega}[t_*](\{\omega_0\}) &= 0 && (4.3)\\ m^{\Omega}[t_*](\{\omega_1\}) &= 1 - pl_0^{\mathcal{T}}(t_*) && (4.4)\\ m^{\Omega}[t_*](\Omega) &= pl_0^{\mathcal{T}}(t_*). && (4.5) \end{aligned}$$

Interpretation: If the value of T is completely plausible assuming ω_0 to be the present state ($pl_0^{\mathcal{T}}(t_*) = 1$), it is not possible to say whether the system is in state ω_0 or in any other state that yields similar values of T. Thus, no value of T ever supports ω_0 only, leading to (4.3). Moreover, the nearer the values of T to those obtained under ω_0, the more plausible Ω is, hence (4.5). Finally, the more the value of T differs from those obtained when ω_0 holds, the greater the belief we have in ω_1: from that we get (4.4).

Example 17. *This solution is illustrated in Figures 4.7 and 4.8, for the ring data set, pl_0 being obtained via the KH algorithm (cf. Figure 4.3).*

A drawback of this method is that, in the specific case where $pl_0^{\mathcal{T}}$ is continuous and Bayesian, and t_* is a singleton, then $pl_0(t_*)$ equals zero. Equation (4.4) thus becomes:

$$m^{\Omega}[t_*](\{\omega_1\}) = 1 - pl_0^{\mathcal{T}}(t_*) = 1, \tag{4.6}$$

and the conclusion is that we always assign full belief to ω_1, without taking the value of t_* into account. There is a paradox there, but we argue that the problem is not in formula (4.3-4.5). In effect, when the belief about T is represented by a probability density function f_T, it does not really make sense to assume that $pl_0(t) = 0$ for all $t \in \mathcal{T}$. As an alternative, it seems more reasonable to use the plausibility function whose pignistic transform equals f_T (See Section 2.3).

Example 18. *This solution is illustrated in Figures 4.9 and 4.10, for the ring data set, pl_0 being obtained via the CI method (cf. Figure 4.5).*

4.5 The cognitive inequality

4.5.1 Definition 1

We considered up to now the case were the only available information is related to one of the classes. Nevertheless, some sort of a priori information is quite often available about

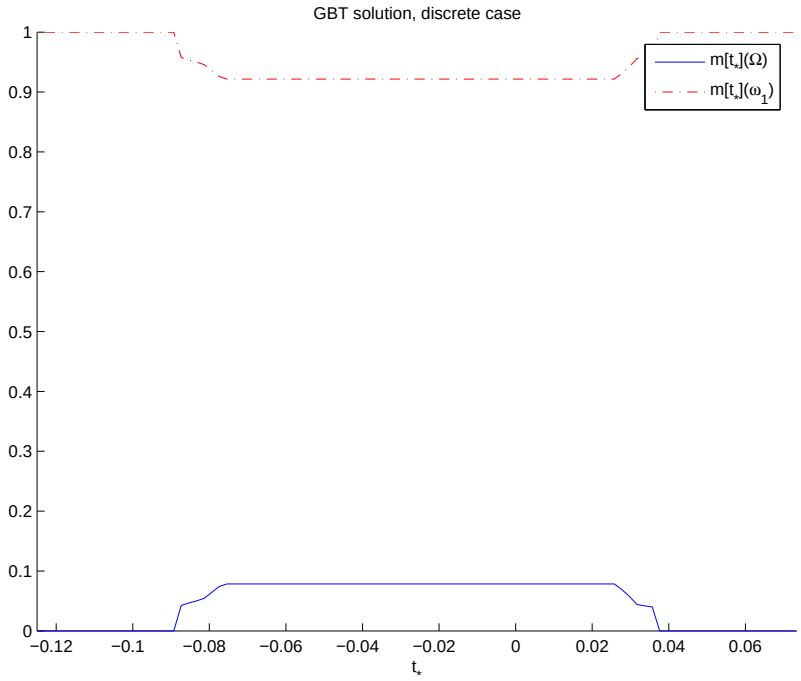


Figure 4.7: Bba on Ω knowing $T = t_*$, GBT solution, discrete case (ring data), for pl_0 obtained via Kriegler and Held's algorithm.

the other class, though it may be very weak. As any piece of knowledge can be turned into a belief function, no matter how incomplete or scarce it might be, there is no reason not to use it when it is available.

To keep things to a general level, let us use ω_a and ω_b instead of ω_0 and ω_1 as two different states of the considered system, and let variable T be a variable whose values depends on the state of the system. Let $pl_a = pl[\omega_a]$ be the belief function representing your belief on the value of variable T when the system is in condition ω_a and $pl_b = pl[\omega_b]$ be the belief function representing your belief on the value of variable T under the hypothesis that the system is in condition ω_b.

Suppose you know for a fact that variable T tends to be bigger when the system is in a well-defined condition ω_b than when the system is in some other condition ω_a (for example, the temperature in the oven tends to be warmer when the oven is *on* (ω_b) than when the oven is *off* (ω_a)).

In the probability theory, a r.v. X is said to be greater than another r.v. Y iff

$$F_X(x) \leq F_Y(x), \quad \forall x, \tag{4.7}$$
$$\Leftrightarrow \quad \mathbb{P}_X((-\infty; x]) \leq \mathbb{P}_Y((-\infty; x]), \quad \forall x, \tag{4.8}$$
$$\Leftrightarrow \quad \mathbb{P}_X((x; +\infty)) \leq \mathbb{P}_Y((x; +\infty)), \quad \forall x. \tag{4.9}$$

In the belief function theory, there exist two distinct ways of expressing this: the first is based on belief functions, the other on plausibility functions. We may write:

$$pl_b^{\mathcal{T}}((-\infty; t]) \leq pl_a^{\mathcal{T}}((-\infty; t]), \quad \forall t \in \mathbb{R}. \tag{4.10}$$

or

$$bel_b^{\mathcal{T}}((-\infty; t]) \leq bel_a^{\mathcal{T}}((-\infty; t]), \quad \forall t \in \mathbb{R}. \tag{4.11}$$

Note that both (4.10) and (4.11) boil down to stochastic inequality when $pl_a^{\mathcal{T}}$ and $pl_b^{\mathcal{T}}$ (and therefore $bel_a^{\mathcal{T}}$ and $bel_b^{\mathcal{T}}$) are probability measures. They will thus be termed *cognitive*

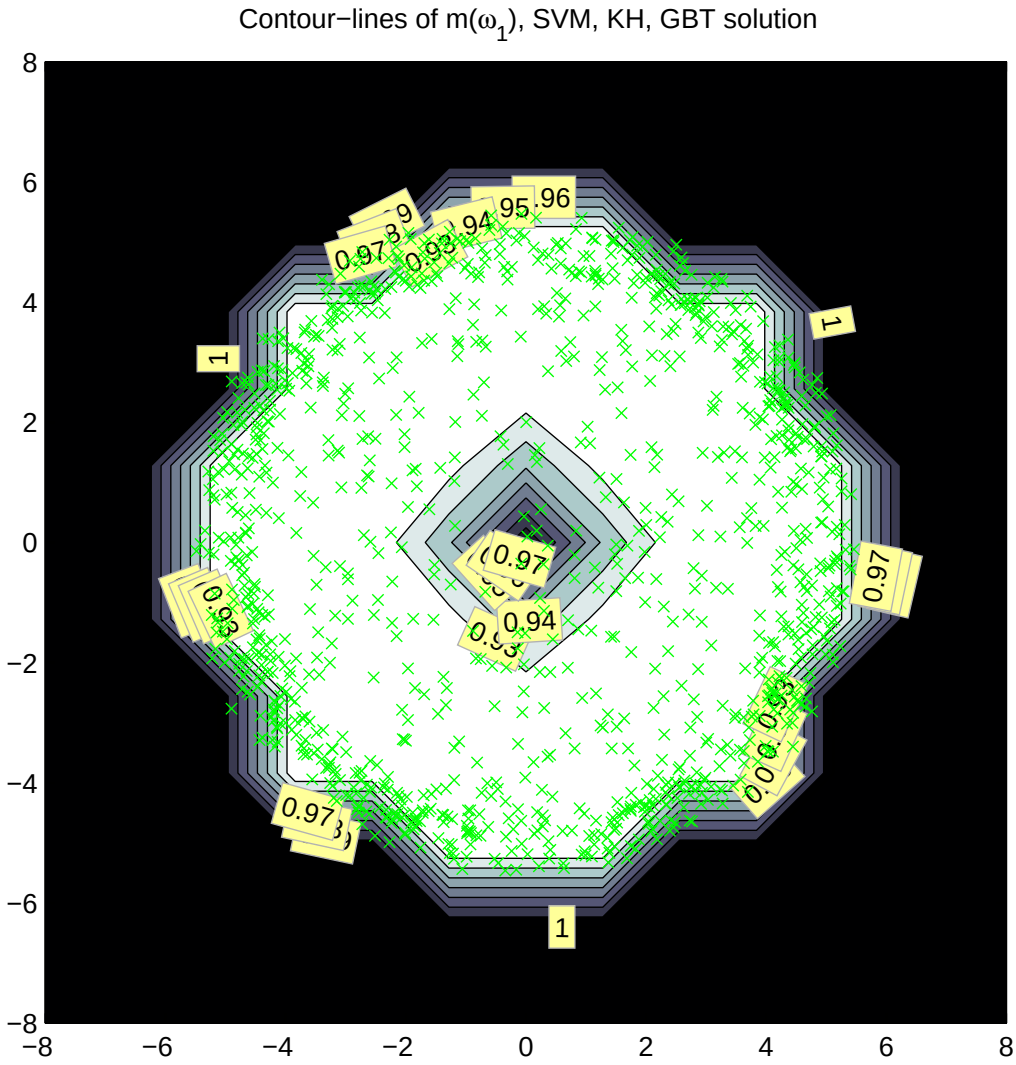


Figure 4.8: Contour-lines of $m^{\Omega}(\omega_1)$ knowing $T = t_*$, GBT solution, discrete case (ring data), for pl_0 obtained via the KH method.

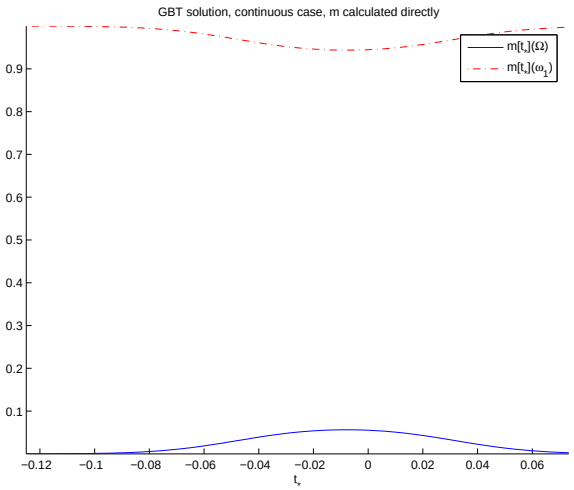


Figure 4.9: Bba on Ω knowing $T = t_*$, GBT solution, continuous case (ring data) for pl_0 obtained via the CI method.

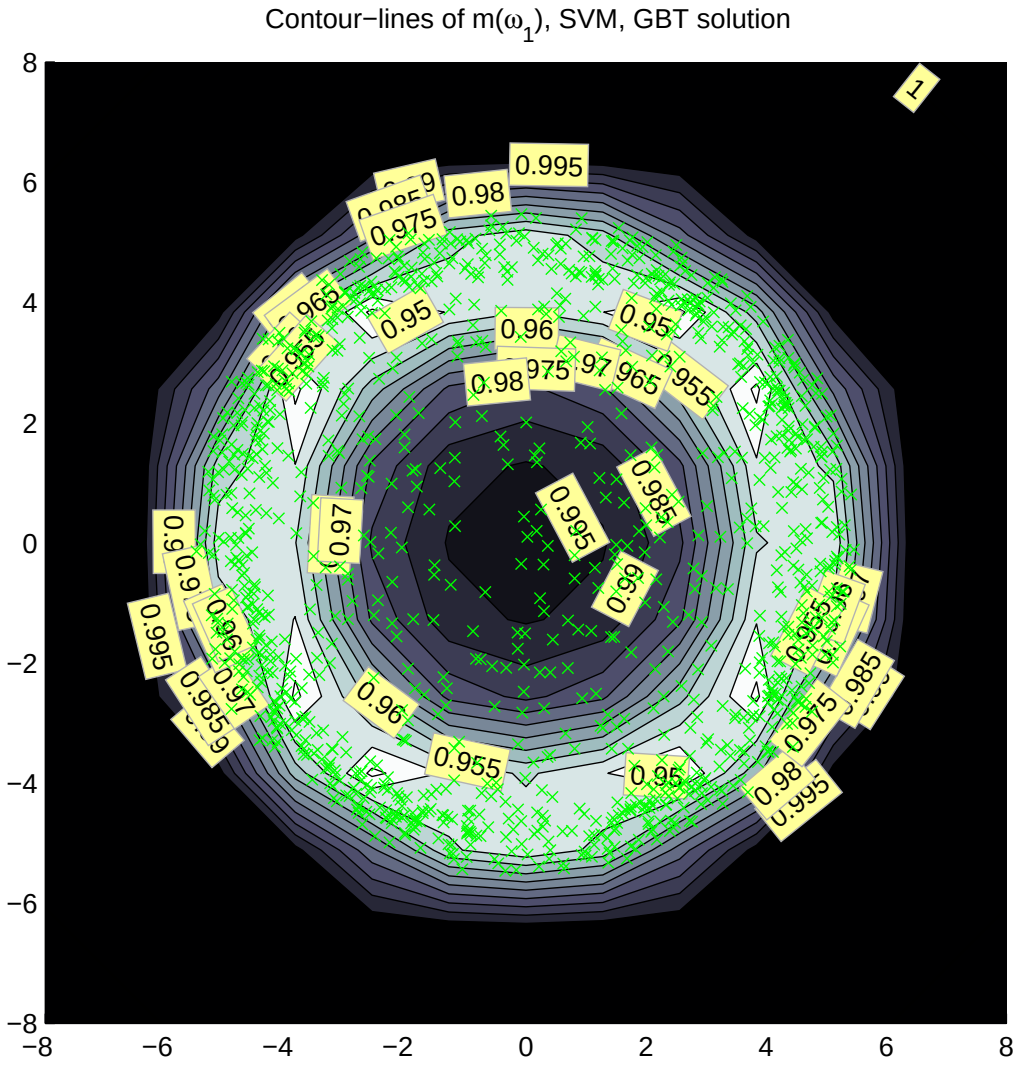


Figure 4.10: Contour-lines of $m^{\Omega}(\omega_1)$ knowing $T = t_*$, GBT solution, (ring data), for pl_0 obtained via the CI method.

inequalities. We will now use use (4.10), which will be called *type I cognitive inequality*. A second type will be defined later.

Suppose now that pl_a is known and that the only available information about pl_b is represented by (4.10). Let us build the least committed belief function pl_b satisfying the constraints enforced by this equation.

4.5.2 Determining the LCBF satisfying a cognitive inequality of type I

Case where pl_a is a discrete belief function on $\mathcal{T} \subseteq \mathbb{R}$

Let $m_a^{\mathcal{T}}$ be a bba on $\mathcal{T} \subseteq \mathbb{R}$ with a finite number of focal elements $I_1, \ldots, I_n$ of the form $I_i = (a_i, b_i]$. By construction, $pl_a^{\mathcal{T}}(-\infty; t)$ is a left continuous step function whose discontinuity point $a_1 \leq a_2 \ldots \leq a_n$ are the lower bounds of the focal elements of $m_a^{\mathcal{T}}$, sorted in increasing order[1].

The least committed a belief function, the greater the associated plausibility, hence the least committed BF verifying (4.10), is the ones that maximizes plausibilities, i.e. the one for which the equality is reached for all t. It may be expressed as follows:

$$m_b(-\infty; +\infty) = pl_a((-\infty; a_1]) \tag{4.12}$$

$$m_b(a_i; +\infty) = pl_a((-\infty; a_i + 1]) - pl_a((-\infty; a_i]), \quad \forall i = 1, \ldots, n-1 \tag{4.13}$$

$$m_b(a_n; +\infty) = 1 - pl_a((-\infty; a_n]) \tag{4.14}$$

Proof.

- In order to get $pl_b^{\mathcal{T}}((-\infty; t]) = pl_a^{\mathcal{T}}((-\infty; t]), \forall t \in (-\infty; a_1]$, the mass $pl_a((-\infty; t_1])$ must be allocated to the biggest possible interval that intersects $(-\infty; a_1]$, that is, $(-\infty; +\infty)$, hence we get Equation (4.12).

- So as to make sure that $pl_b^{\mathcal{T}}((-\infty; t]) \leq pl_a^{\mathcal{T}}((-\infty; t]), \forall t \in (a_1; a_2]$, a mass $pl_a((-\infty; a_2]) - pl_a((-\infty; a_1])$ must be allocated to the biggest possible interval that intersects $(-\infty; a_2]$ but not $(-\infty; a_1]$, i.e. $(a_1; +\infty)$, therefore:

$$m_b(a_1; +\infty) = pl_a(-\infty; a_2]) - pl_a(-\infty; a_1]). \tag{4.15}$$

- Again, a mass $pl_a((-\infty; a_3]) - pl_a((-\infty; a_2])$ must be allocated to the biggest possible interval that intersects $(-\infty; a_3]$ but not $(-\infty; a_i]$, $i < 3$, so as to ensure that $pl_b^{\mathcal{T}}((-\infty; t]) \leq pl_a^{\mathcal{T}}((-\infty; t]), \forall t \in (a_2; a_3]$. This interval is $(a_2; +\infty)$, thus:

$$m_b(a_2; +\infty) = pl_a(-\infty; a_3]) - pl_a(-\infty; a_2]). \tag{4.16}$$

- The same reasoning holds for any interval $(a_i; a_{i+1}]$, leading to (4.13).

- Finally, a mass $pl_a((-\infty; +\infty]) - pl_a((-\infty; a_n])$ must be allocated to the biggest possible interval that intersects $(-\infty; a_n]$ but not $(-\infty; a_i]$, $i < n$, so that $pl_b^{\mathcal{T}}((-\infty; t]) \leq pl_a^{\mathcal{T}}((-\infty; t]), \forall t \in (a_n; +\infty)$. This interval is $(a_n; +\infty)$, yielding Equation (4.14).

□

Remark 11. *The focal sets are nested. Subsequently, function pl_b is a possibility distribution. It increases over $\mathbb{R}$, and*

$$pl_b(t) = pl_b((-\infty; t]). \tag{4.17}$$

Remark 12. *By misuse of notations, pl(t)=pl({t}). Similarly, whenever pl is a possibility, the possibility measure and the possibility distribution are both denoted pl in the sequel.*

Remark 13. *Considering a bba m_a with focal elements I_i of the form $[a_i, b_i]$ would not change the result.*

[1] Without loss of generality, it is supposed here that the a_i are all distinct.

Link between the focal elements of m_a and those of m_b The function $t \mapsto pl_a((-\infty; t])$ is discontinuous at points a_i such that $(a_i; b_i]$ is a focal element of pl_a. The corresponding mass is $m_a((a_i; b_i]) = pl_a((-\infty; a_{i+1}]) - pl_a((-\infty; a_i])$. Consequently, m_b may be directly built from m_a by transferring the mass allocated to each focal element $(a_i; b_i]$ of m_a onto $(a_i; +\infty)$.

Case where pl_a is a continuous belief function on $\mathcal{T} \subseteq \mathbb{R}$

Let m_a be a bbd. By construction, $pl_a((-\infty; t])$ is a function increasing over $\mathbb{R}$, $\forall t$. In that case,

Proposition 6. *the least committed belief function $bel_a^{\mathcal{T}}$ compatible with constraints (4.29) is defined by the following bbd:*

$$m_b^{\mathcal{T}}(t; +\infty) = \int_t^{+\infty} m_a(t, v)dv \tag{4.18}$$

Proof. As a belief function $pl_2^{\mathcal{T}}$ is less committed than another belief function $pl_1^{\mathcal{T}}$ iff $pl_1^{\mathcal{T}}(A) \geq pl_2^{\mathcal{T}}(A)$, $\forall A \in \mathcal{T}$, the least committed belief function pl_b satisfying (4.10) is the one that maximizes pl_b under the constraint of Equation (4.10). Consequently, the LCBF satisfying (4.10) is the one for which the equality in (4.10) is reached, if such a BF exists. Hence we need:

$$pl_b((-\infty; t]) = pl_a((-\infty; t]), \quad , \forall t \in \mathcal{T}. \tag{4.19}$$

Now, $pl_a((-\infty; t])$ is the integral of bdd m_a on all intervals whose intersection with $(-\infty; t]$ is not empty. Let dt be an infinitesimal quantity. Then $pl_a((-\infty; t - dt])$ is the integral of bdd m_a on all intervals whose intersection with $(-\infty; t - dt]$ is not empty. Hence, the difference $pl_a((-\infty; t]) - pl_a((-\infty; t - dt])$ is the integral of m_a on all intervals intersecting with $(-\infty; t]$ but not with $(-\infty; t - dt]$ (cf. shaded area on Figure 4.11). The lower bound u of such intervals may vary between $t - dt$ and t, while their upper bound may vary between $u = \max(u, t - dt)$ and $+\infty$.

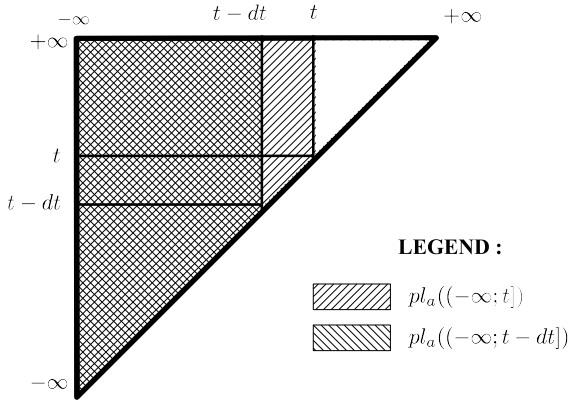


Figure 4.11: Representation of the integration area for the plausibility function

Hence,

$$\Delta(m) = pl_a((-\infty; t]) - pl_a((-\infty; t - dt]) \tag{4.20}$$

$$= \int_{u=t-dt}^{t} \int_{v=\max(u,t-dt)}^{+\infty} m_a(u, v)dvdu, \tag{4.21}$$

and, in this particular case, $\max(u, t - dt) = u$.

As we require

$$pl_b((-\infty;t]) = pl_a((-\infty;t])) \tag{4.22}$$

and

$$pl_b((-\infty;t-dt]) = pl_a((-\infty;t-dt]), \tag{4.23}$$

we also obviously require

$$pl_b((-\infty;t]) - pl_b((-\infty;t-dt]) = pl_a((-\infty;t]) - pl_a((-\infty;t-dt]). \tag{4.24}$$

Thus, the LCBF m_b that satisfies this equality is the one that allocates the amount of belief $\Delta(m)$ to the biggest possible interval that intersects with $(-\infty;t]$ but not with $(-\infty;t-dt]$, namely $(t-dt;+\infty)$. Therefore, m_b is the bbd such that

$$m_b(t-dt,+\infty) = \int_{u=t-dt}^{t} \int_{v=\max(u,t-dt)}^{+\infty} m_a(u,v)dvdu. \tag{4.25}$$

When $dt \to 0$, this becomes:

$$m_b(t,+\infty) = \int_{v=t}^{+\infty} m_a(t,v)dv. \tag{4.26}$$

Subsequently, masses m_a are allocated to intervals of the form $(t;+\infty)$, with $t \in \mathbb{R}$.
It may be checked that requirement (4.10) is met:

$$\begin{array}{rcl} pl_b((-\infty;t]) & = & \int_{u=-\infty}^{t} m_b(u;+\infty)du \\ & = & \int_{u=-\infty}^{t} \int_{v=u}^{+\infty} m_a(u,v)dvdu \\ & = & pl_a((-\infty;t]). \end{array} \tag{4.27}$$

□

4.5.3 Definition 2

We supposed up to now that variable T tends to be bigger when the system is in state ω_b than when the system is in state ω_a. Suppose now that we want to express the opposite hypothesis, i.e. variable T tends to be smaller when the system is in state ω_b than when the system is in state ω_a.

In terms of belief functions, this statement may be expressed as follows:

$$pl_a^{\mathcal{T}}((-\infty;t]) \leq pl_b^{\mathcal{T}}((-\infty;t]), \quad \forall t \in \mathbb{R}. \tag{4.28}$$

Alternately, we may try to express our hypothesis with belief functions instead of plausibility functions. It yields:

$$\begin{array}{rcl} & bel_a^{\mathcal{T}}((-\infty;t]) & \leq bel_b^{\mathcal{T}}((-\infty;t]), \quad \forall t \in \mathbb{R}, \\ \Leftrightarrow & 1 - bel_a^{\mathcal{T}}((-\infty;t]) & \geq 1 - bel_b^{\mathcal{T}}((-\infty;t]), \quad \forall t \in \mathbb{R}, \\ \Leftrightarrow & pl_a^{\mathcal{T}}((t;+\infty)) & \geq pl_b^{\mathcal{T}}((t;+\infty)), \quad \forall t \in \mathbb{R}. \end{array} \tag{4.29}$$

Remark 14. *In the belief function theory, there exist two distinct notions of cognitive inequality, namely (4.10) (called type I) and (4.29), which will be termed type II in the sequel.*

Now, it may be shown that trying to compute the LCBF satisfying requirement (4.28) systematically leads to the vacuous belief function. On the other end, we will show that trying to compute the LCBF satisfying requirement (4.29) does not lead to the vacuous belief function. Equation (4.10) and (4.29) define two forms of cognitive inequalities. The best possible use of the available knowledge is made when the one that does not lead to the vacuous belief function for pl_b is used.

4.5.4 Determining the LCBF satisfying a cognitive inequality of type II

Case where pl_a is a discrete belief function on $\mathcal{T} \subseteq \mathbb{R}$

The function $t \mapsto pl_a((t;+\infty))$ is a right continuous step function whose discontinuity points are the upper bounds b_i of the focal intervals $(a_i; b_i]$, sorted in increasing order.

The least committed bba m_a satisfying (4.29) is

$$m_b^{\mathcal{T}}(-\infty, +\infty) = pl_a((b_n;+\infty)), \tag{4.30}$$
$$m_b^{\mathcal{T}}(-\infty, b_i) = pl_a^{\mathcal{T}}((b_{i-1}, \ldots, +\infty)) - pl_a^{\mathcal{T}}((b_i, \ldots, +\infty)), \tag{4.31}$$
$$m_b^{\mathcal{T}}(-\infty, b_1) = 1 - pl_a((b_1;+\infty)). \tag{4.32}$$

Proof. As for the type I case, the idea is to try and get an equality for relation (4.29), and to deduce bba m_b from this. Note that $pl_a((t;+\infty))$ is a right continuous step function decreasing over $\mathbb{R}$, and whose discontinuity points t_i are the upper bounds $b_{(i)}$ of the focal elements of m_a, sorted in increasing order. Deriving equation (4.29) for each t_i successively leads to the above result. □

Remark 15. *Bba m_b may directly be built from m_a by transferring the masses allocated to each focal elements $(a_i; b_i]$ of m_a onto $(-\infty; b_i]$.*

Remark 16. *Bba m_b is consonant and $pl_b(t) = pl_b((t;+\infty)) = pl_a((t;+\infty))$, $\forall t \in \mathbb{R}$.*

Case where pl_a is a continuous belief function on $\mathcal{T} \subseteq \mathbb{R}$

The line of reasoning of Section 4.5.4 directly extends to the case where $m_a^{\mathcal{T}}$ is a bbd. In that case,

Proposition 7. *the least committed belief function $bel_b^{\mathcal{T}}$ compatible with constraints (4.29) is defined by the following bbd:*

$$m_b^{\mathcal{T}}(-\infty, t) = \int_{-\infty}^{t} m_a(u, t)du \tag{4.33}$$

Proof. The proof is similar to that of the type I case. The reader is referred to appendix C for details. □

4.6 Steps 2 and 3: Solutions using the cognitive inequality

We will now introduce two solutions to the novelty detection problem using the cognitive inequality.

4.6.1 Model 1

Description of the model

Remind that the problem under consideration is the assessment of the hypothesis that a system is in class ω_0 when the only available information about the system concerns the distribution of statistic T conditioned on ω_0.

We defined ω_0 as the normal or reference state of the system under study, for which a set $\mathbf{x_1}, \ldots, \mathbf{x_n}$ of examples is available, and ω_1 is the set of all other states, for which no data is available. During step 1, a novelty measure T was built from observations $\mathbf{x_1}, \ldots, \mathbf{x_n}$. Having observed a value t_* of T, we want to define a bba $m^{\Omega}[t_*]$ on Ω, that quantifies our belief about the system state given t_*. We first need to built $m_0^{\mathcal{T}} = m^{\mathcal{T}}[\omega_0]$ and $m_1^{\mathcal{T}} = m^{\mathcal{T}}[\omega_1]$.

Step 2: Building of $m_0^{\mathcal{T}}$ and $m_1^{\mathcal{T}}$

As in the GBT solution (see Section 4.4), we will take $m_0^{\mathcal{T}} = m_*^{\mathcal{T}}$, as $m_*^{\mathcal{T}}$ represents our belief on T drawn from the same distribution as those obtained from training data collected when the system is in state ω_0.

Additionally, let us suppose that, by construction, our novelty measure T tends to be larger when ω_1 holds than when ω_0 is true[2]. This corresponds to (4.10), and pl_1 may thus be deduced from pl_0 by reasoning as in Section 4.5.2, with $pl_0 = pl_a$ and $pl_1 = pl_b$, and $m_0 = m_a$ and $m_1 = m_b$. It yields:

$$\begin{aligned} m_1(-\infty;+\infty) &= pl_0((-\infty;a_1]), && (4.34)\\ m_1(a_i;+\infty) &= pl_0((-\infty;a_{i+1}]) - pl_0((-\infty;a_i]), \quad \forall i = 1,\dots,n-1, && (4.35)\\ m_1(a_n;+\infty) &= 1 - pl_0((-\infty;a_n]). && (4.36)\end{aligned}$$

Remark 17. *pl_1 is the possibility distribution defined by:*

$$pl_1(t) = pl_0((-\infty,t]), \quad \forall t \in \mathbb{R}. \tag{4.37}$$

Step 3: Building of $m^{\Omega}[t]$

If we follow the reasoning of Section 4.2, we should now apply the GBT to $m_0^{\mathcal{T}}$ and $m_1^{\mathcal{T}}$ in order to obtain $m^{\Omega}[t_*]$. However, remember that a necessary condition for the application of the GBT is the independence of $m_0^{\mathcal{T}}$ and $m_1^{\mathcal{T}}$. It happens that, as we built $m_1^{\mathcal{T}}$ from $m_0^{\mathcal{T}}$, they are not independent. Consequently, the conjunctive combination rule cannot be applied here.

We need to build $m^{\mathcal{T}\times\Omega}$ such that:

$$m^{\mathcal{T}\times\Omega}[\{\omega_0\}\times\mathcal{T}]^{\downarrow\mathcal{T}} = m_0^{\mathcal{T}} \tag{4.38}$$

$$\text{and } m^{\mathcal{T}\times\Omega}[\{\omega_1\}\times\mathcal{T}]^{\downarrow\mathcal{T}} = m_1^{\mathcal{T}}, \tag{4.39}$$

and then condition it with respect to t_* so as to get $m^{\Omega}[t_*]$.

- **Combination of m_0 and m_1:**

Let I_1 to I_n be the focal elements of $m_0^{\mathcal{T}}$. To each $I_i = (a_i, b_i]$ is associated I_i', focal element of $m_1^{\mathcal{T}}$, such that: $I_i' = (a_i, = \infty)$. Thus,

$$m^{\mathcal{T}\times\Omega}\left(I_i\times\{\omega_0\}\cup I_i'\times\{\omega_1\}\right) = m_0^{\mathcal{T}}(I_i). \tag{4.40}$$

- **Conditioning with respect to $t_* \subseteq \mathcal{T}$:**

Note that $|t_*|$ may be greater than 1 and that t_* is not necessarily an interval. The following relations hold:

$$\begin{aligned} pl^{\Omega}[t_*](\{\omega_0\}) &= pl_0^{\mathcal{T}}(t_*)\\ pl^{\Omega}[t_*](\{\omega_1\}) &= pl_0^{\mathcal{T}}(-\infty,\sup(t_*))\\ &= pl_1^{\mathcal{T}}(t_*)\\ pl^{\Omega}[t_*](\emptyset) &= 1 - pl_0^{\mathcal{T}}(-\infty,\sup(t_*))\\ &= 1 - pl_1^{\mathcal{T}}(t_*).\end{aligned} \tag{4.41}$$

Hence,

$$\begin{aligned} m^{\Omega}[t_*](\omega_0) &= 0\\ m^{\Omega}[t_*](\omega_1) &= pl_0^{\mathcal{T}}(-\infty,\sup(t_*)) - pl_0^{\mathcal{T}}(t_*)\\ &= pl_1^{\mathcal{T}}(t_*) - pl_0^{\mathcal{T}}(t_*)\\ m^{\Omega}[t_*](\Omega) &= pl_0^{\mathcal{T}}(t_*)\\ m^{\Omega}[t_*](\emptyset) &= 1 - pl_0^{\mathcal{T}}(-\infty,\sup(t_*))\\ &= 1 - pl_1^{\mathcal{T}}(t_*).\end{aligned} \tag{4.42}$$

[2]This is very often true for novelty measures.

- **Interpretation:** This end result may be interpreted as follows.

– When the values of T are similar to those obtained when in state ω_0, nothing can be said about them being from one class or the other, and the belief is thus spread onto Ω.

– When the values of T are smaller than those we get when in condition ω_0, there is an inconsistency with our original information according to which, when there is a departure from state ω_0, T should tend to be bigger than in state ω_0. The corresponding amount of belief is thus allocated to the empty set, reflecting this conflict.

– When T gets bigger than its usual values when the system is in state ω_0, then our belief turns to ω_1, in agreement with (4.10).

- Finally, no value of T ever supports ω_0 only.

Note that normalizing yields:

$$\begin{aligned} m^{\Omega}[t_*](\omega_0) &= 0 \\ m^{\Omega}[t_*](\omega_1) &= 1 - \frac{pl_0^{\mathcal{T}}(t_*)}{pl_1^{\mathcal{T}}(t_*)} \\ m^{\Omega}[t_*](\Omega) &= \frac{pl_0^{\mathcal{T}}(t_*)}{pl_1^{\mathcal{T}}(t_*)} \end{aligned} \tag{4.43}$$

Example 19. *The result is illustrated in Figures 4.12 and 4.13, for the ring data set, pl_0 being obtained via Kriegler and Held's algorithm (cf. Figure 4.3). It may be observed that there is an improvement on the GBT solution as most of the data inside the ring induce a low mass of belief on ω_1.*

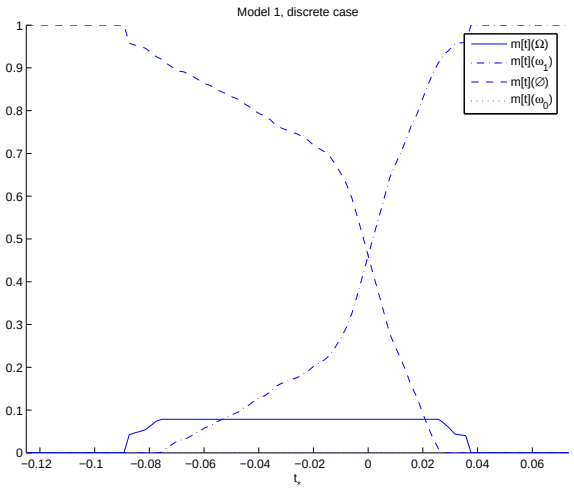


Figure 4.12: Bba on Ω knowing $T = t_*$, Model 1, discrete case (ring data), for pl_0 obtained via Kriegler and Held's algorithm.

Remark 18. *Note that, if pl_0 is Bayesian, we may end up always deciding in favour of ω_1, but the remark of 4.4 still holds: if our information with respect to ω_0 is a probability, then we should use the belief function whose pignistic transform is this probability, and not the belief function whose bba is this probability.*

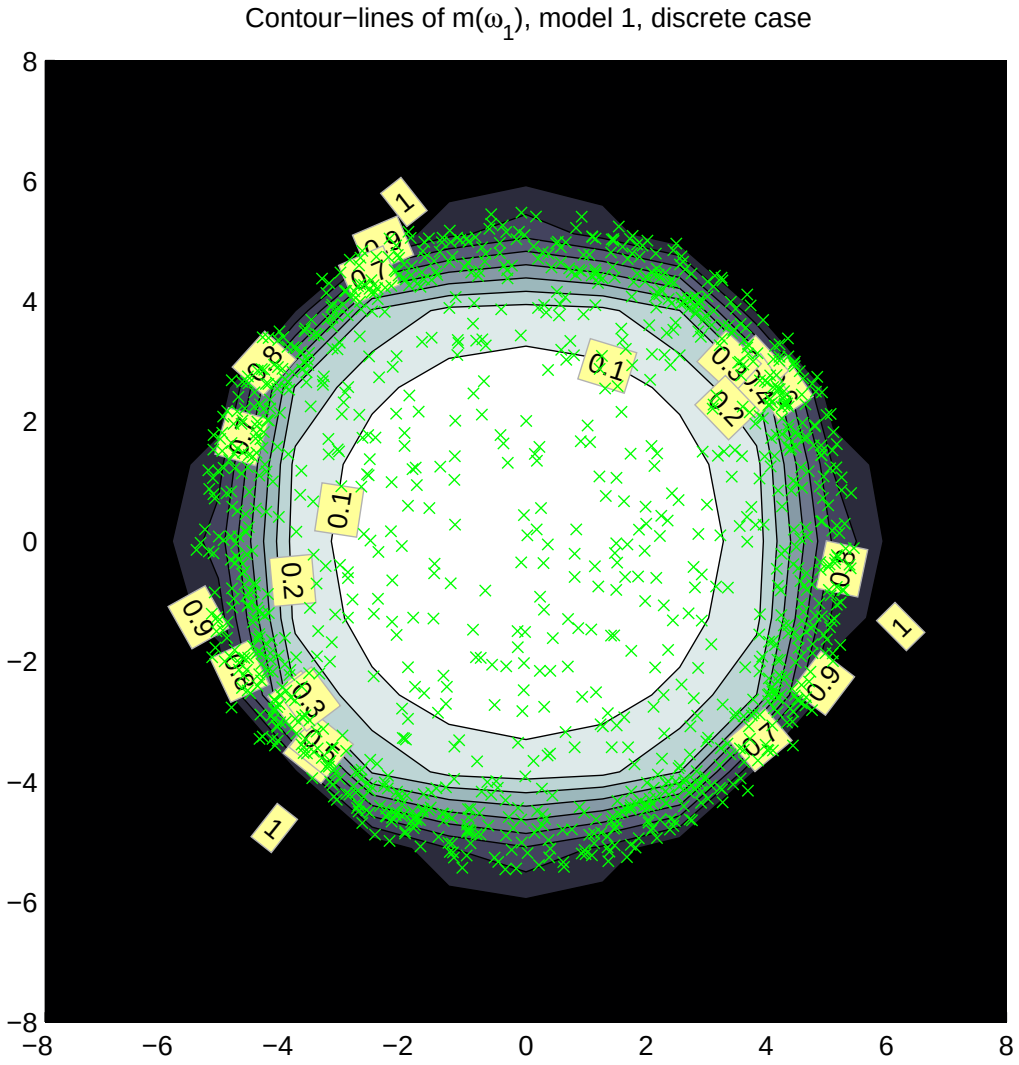


Figure 4.13: Contour-lines of $m^{\Omega}(\omega_1)$ knowing $T = t_*$, Model 1, discrete case (ring data), for pl_0 obtained via Kriegler and Held's algorithm.

Example 20. *The result is illustrated in Figures 4.14 and 4.15, for the ring data set, pl_0 being obtained via the CI method ((cf. Figure 4.5)). As required in this model, data for which the value of $T = -f(\mathbf{x})$ is lower than for the majority of the training data do not induce a high mass of belief on ω_1 whereas data for which the value of T is higher than for the majority of the training data do. The qualitative information that* "the value of T gets bigger for abnormal data" *has been successfully incorporated in the model, and leads to an improvement on the results obtained by the GBT solution.*

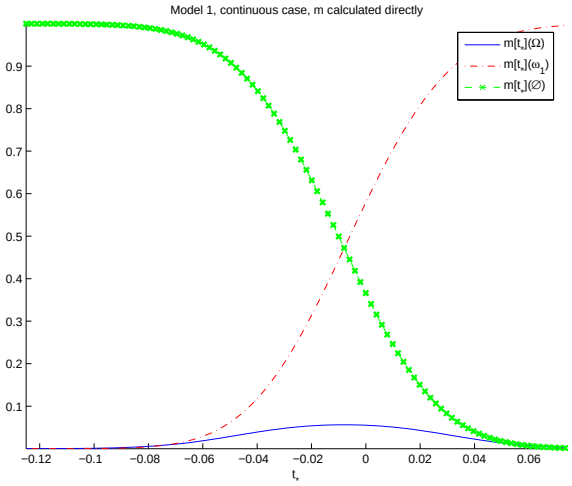


Figure 4.14: Bba on Ω knowing $T = t_*$, Model 1, continuous case (ring data), for pl_0 obtained via Cheng and Iles' confidence-band.

4.6.2 Model 2

Description of the model

In the previous model, we considered that pl_0 was built directly from the training data, and that, given the fact that T tends to be larger when ω_1 holds than when ω_0 is true, pl_1 could be deduced from pl_0 through (4.10).

Alternately, we may consider, as for the GBT solution, that we know nothing on the behaviour of T when ω_1 holds, hence the information we have at our disposal in this respect can be modeled with the vacuous belief function:

$$pl_1^{\mathcal{T}}(A) = pl^{\mathcal{T}}[\omega_1](A) = 1, \quad \forall A \subseteq \mathbb{R}.$$

However, we still need to take into account the fact that T was built in such a way that its value increases in case of departure from the normal state. Let us put it this way: as T tends to be larger when ω_1 holds than when ω_0 is true, we know that values of T smaller than those encountered in the training data do not indicate departure from the normal state.

It is this statement, denoted S_1, that we now would like to represent in terms of belief functions. We may thus consider a frame of discernment $\Omega = \{\omega_{0'}, \omega_{0''}, \omega_1\}$, where $\omega_{0'}$ is the observed, normal state, for which training data are collected, and $\omega_{0''}$ corresponds to non-observed states leading to values of T smaller than those observed for the normal state. We do not need to detect states corresponding to $\omega_{0''}$, as only an increase in the value of T would be of concern to us. Consequently, we can write $\Omega = \{\omega_0, \omega_1\}$, with $\omega_0 = \{\omega_{0'}, \omega_{0''}\}$, and build a belief function on Ω that does not distinguish between states $\omega_{0'}$ and $\omega_{0''}$.

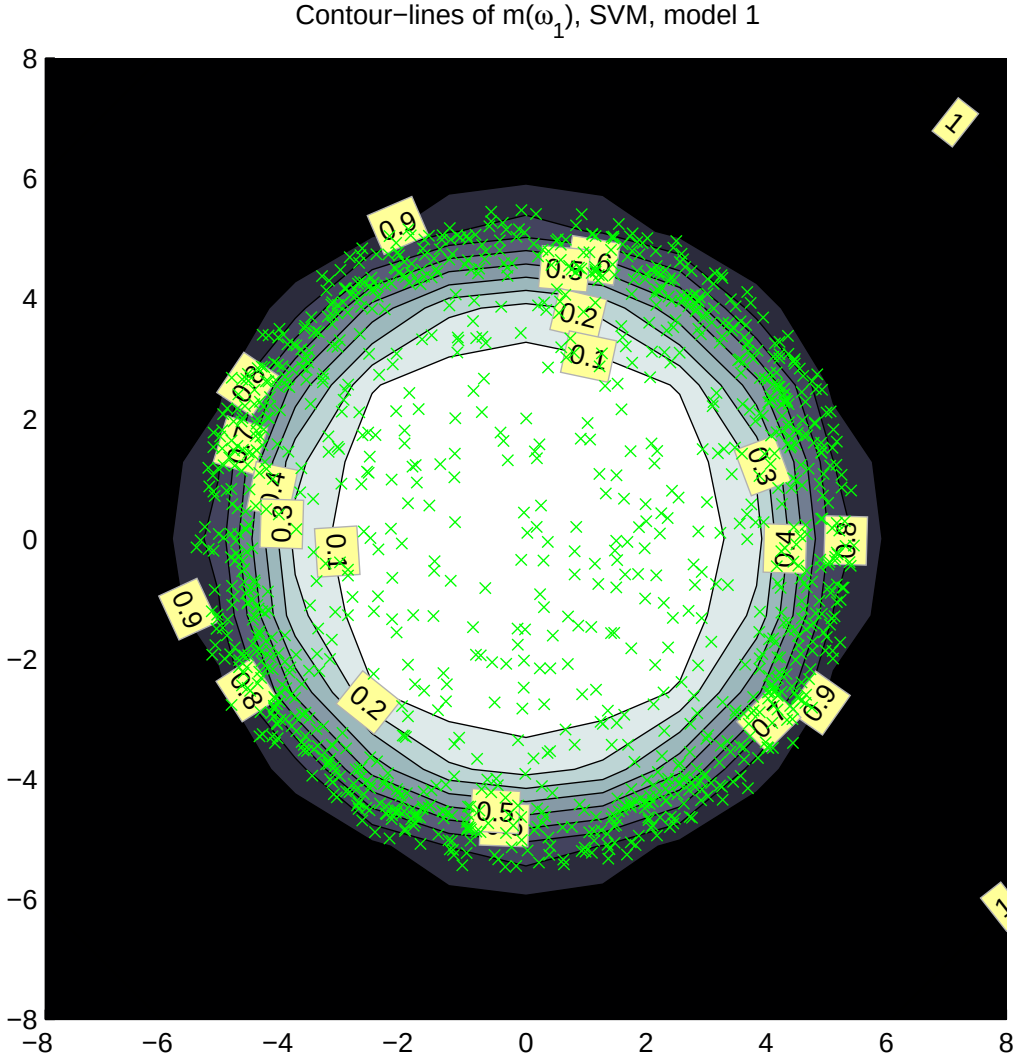


Figure 4.15: Contour-lines of $m^{\Omega}(\omega_1)$, knowing $T = t_*$, Model 1, discrete case (ring data), for pl_0 obtained via the CI method.

Additionally, bba $m_*^{\mathcal{T}}$ obtained at Step 1 represents our belief on T drawn from the same distribution than those obtained when the system is in state ω_0. Consequently, we can take $m^{\mathcal{T}}[\omega_{0'}] = m_{0'}^{\mathcal{T}} = m_*^{\mathcal{T}}$.

As explained in Section 4.5.3, statement S_1 may then be turned into the following constraint:

$$pl_0^{\mathcal{T}}((t,+\infty)) \leq pl_{0'}^{\mathcal{T}}((t,+\infty)), \quad \forall t \in \mathbb{R}. \tag{4.44}$$

where:

- $pl_{0'}^{\mathcal{T}}$ is the predictive plausibility function associated with the bba $m_{0'}^{\mathcal{T}}$ computed in the step 1 –that represents our belief in future values of T drawn from exactly the same distribution as the learning sample–, which corresponds to observations of T gathered while the system was in a normal state, under some well-defined experimental conditions EC,
- and $pl_0^{\mathcal{T}} = bel^{\mathcal{T}}[\omega_0]$ denotes the plausibility function associated with bba $m^{\mathcal{T}}[\omega_0]$ on the value of T knowing that the system is in the normal state ω_0.

We will now see how to solve our novelty detection problem when modeled in this way.

Step 2: Building of $m^{\mathcal{T}}[\omega_0]$ and $m^{\mathcal{T}}[\omega_1]$

Equation (4.44) defines a set of constraints that should be statisfied by $pl_0^{\mathcal{T}}$. In the TBM, the *least commitment principle* dictates to select the *least committed* belief function, among those compatible with a set of constraints [116].

The solution to this problem has been described in Section 4.5.4. With $m_{0'} = m_a$ and $m_0 = m_b$, we get, in the discrete case,

$$\begin{aligned} m_0^{\mathcal{T}}(-\infty,+\infty) &= pl_{0'}((a_n;+\infty)), & (4.45)\\ m_0^{\mathcal{T}}(-\infty,a_i) &= pl_{0'}^{\mathcal{T}}((a_{i-1},\ldots,+\infty)) - pl_{0'}^{\mathcal{T}}((a_i,\ldots,+\infty)), & (4.46)\\ m_0^{\mathcal{T}}(-\infty,a_1) &= 1 - pl_{0'}((a_1;+\infty)), & (4.47) \end{aligned}$$

and in the continuous case,

$$\begin{aligned} m_0^{\mathcal{T}}(-\infty,t) &= \int_{-\infty}^{t} m_{0'}(u;t)du; & (4.48)\\ & & (4.49) \end{aligned}$$

or, equivalently, in both cases, pl_0 is the possibility distribution defined by $pl_0(t) = pl_{0'}((t;+\infty)), \forall t \in \mathbb{R}$. As already mentionned, m_1 is the vacuous bba.

Special case: $m_{0'}^{\mathcal{T}}$ built from a confidence band Suppose now that $m_{0'}^{\mathcal{T}}$ was built using the confidence band based method described in Section 2.2.3 or 2.2.4. In order to build the least committed belief function compatible with (4.29), first observe that $pl_{0'}$ satisfies

$$pl_{0'}^{\mathcal{T}}((t,+\infty)) = 1 - bel_{0'}^{\mathcal{T}}((-\infty,t]) = 1 - \underline{F}(t), \tag{4.50}$$

where $\underline{F}$ is the step function defined by (2.22).

Hence pl_0 has a very simple expression

$$pl_0^{\mathcal{T}}(t) = 1 - \underline{F}(t), \quad \forall t \in \mathbb{R}, \tag{4.51}$$

and $pl_0^{\mathcal{T}}(A) = \sup_{t\in A} pl_0^{\mathcal{T}}(t)$ for all $A \subseteq \mathbb{R}$.

Step 3: Constructing $m^{\Omega}[t]$

The belief function $bel_0^{\mathcal{T}}[\omega_0]$ built in the previous step quantifies our beliefs on T, given that the system is in state ω_0. As already mentioned, since no data is available regarding state ω_1, our belief on T given ω_1 is vacuous, i.e.,

$$pl^{\mathcal{T}}[\omega_1](A) = pl_1^{\mathcal{T}}(A) = 1, \quad \forall A \subseteq \mathbb{R}. \tag{4.52}$$

This time, m_0 and m_1 are independent, and the GBT therefore allows us to compute our belief on Ω given that $T \in t_*$ for any $t_* \subseteq \mathbb{R}$. Using (1.34), we get:

$$\begin{aligned} m^{\Omega}[t_*](\{\omega_0\}) &= 0 && (4.53)\\ m^{\Omega}[t_*](\{\omega_1\}) &= 1 - pl_0^{\mathcal{T}}(t_*) && (4.54)\\ m^{\Omega}[t_*](\Omega) &= pl_0^{\mathcal{T}}(t_*). && (4.55) \end{aligned}$$

As pl_0 is a possibility distribution decreasing over $\mathbb{R}$, this can be rewritten as:

$$\begin{aligned} m^{\Omega}[t_*](\{\omega_0\}) &= 0 && (4.56)\\ m^{\Omega}[t_*](\{\omega_1\}) &= 1 - pl_0^{\mathcal{T}}(\inf(t_*)) && (4.57)\\ m^{\Omega}[t_*](\Omega) &= pl_0^{\mathcal{T}}(\inf(t_*)). && (4.58) \end{aligned}$$

In the special case where $pl_{0'}$ has been calculated either from Kolmogorov's or Cheng and Iles' confidence band, if $t_* = \{t\}$ (t_* is a singleton), we get:

$$\begin{aligned} m^{\Omega}[t_*](\{\omega_0\}) &= 0 && (4.59)\\ m^{\Omega}[t_*](\{\omega_1\}) &= \underline{F}(t) && (4.60)\\ m^{\Omega}[t_*](\Omega) &= 1 - \underline{F}(t). && (4.61) \end{aligned}$$

Note that this result has, again, a simple interpretation: a large value of T supports the hypothesis that the system is not in the normal state. The degree of support increases as a function of t.

On the contrary, a small value of T, similar to those obtained when the system is in normal conditions, may occur either when the system is in a normal state, or when the system is in an abnormal state that does not affect the values of T. Therefore, small values of T are highly plausible under both ω_0 and ω_1 and they do not support any specific hypothesis.

Example 21. *The result is illustrated in Figures 4.16 and 4.17, for the ring data set, pl_0 being obtained via Cheng and Iles' confidence-band as shown in Figure 4.5. The mass of belief on ω_1 increases with t. Again, the information according to which* "values of T smaller than those obtained for the training set do not indicate a departure from the normal state" *has been successfully incorporated in the model, and leads to an improvement on the results obtained by the GBT solution.*

4.7 Discussion

The use of the cognitive inequality is an obvious improvement to the simple GBT solution, as it allows the handling of additional, qualitative information. On the other hand, it is more difficult to compare the relative quality of Models 1 and 2. Model 1 is very simple and fairly straight-forward, but leads to a more complex solution than Model 2. On the contrary, the latter is a little far-fetched, but leads to a very simple solution, and is therefore very easy to use. This solution proved to show good performances in different novelty detection applications, thus validating the model experimentally.

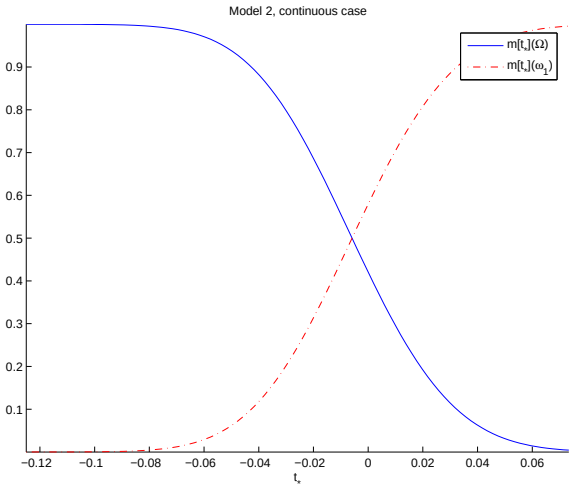


Figure 4.16: Bba on Ω knowing $T = t_*$, Model 2, continuous case, for pl_0 obtained via Cheng and Iles' confidence-band (ring data).

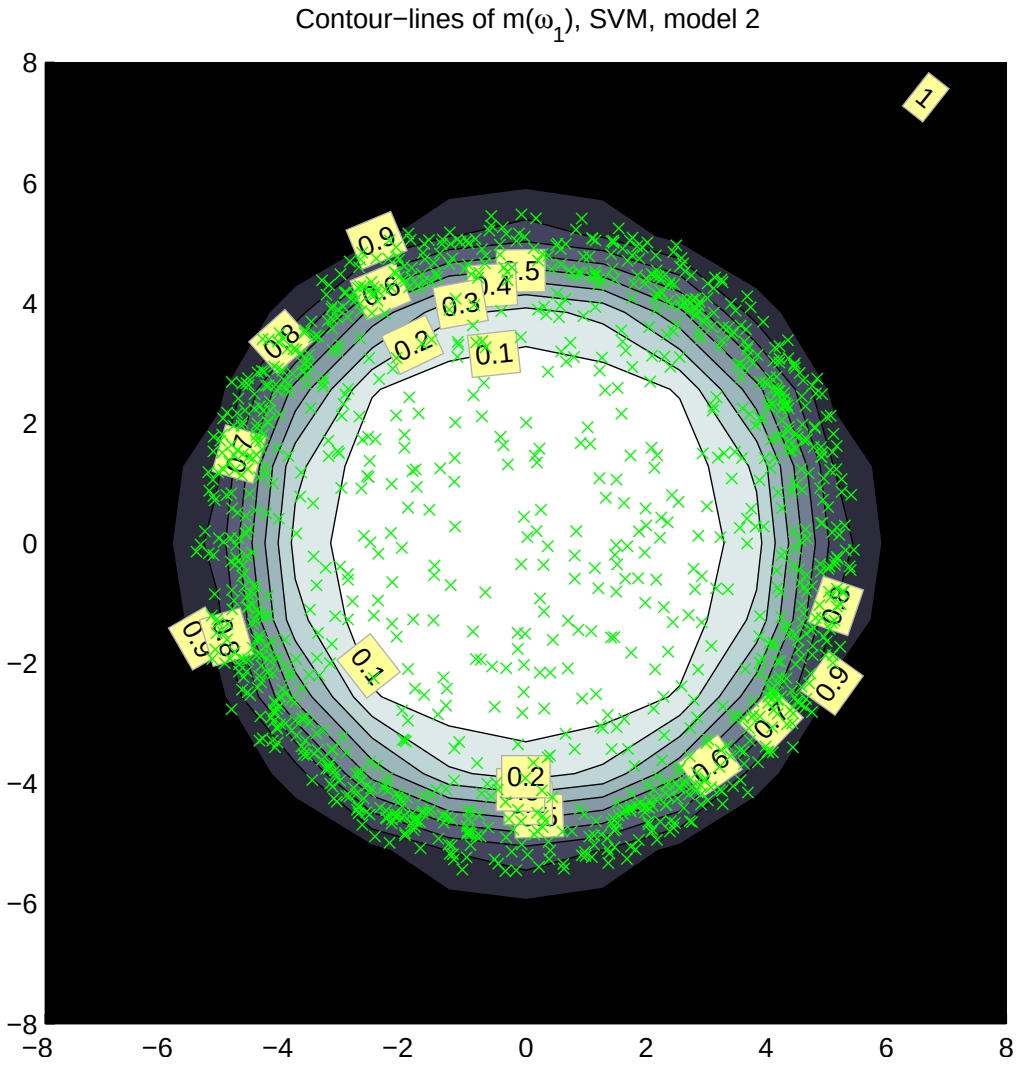


Figure 4.17: Contour-lines of $m^{\Omega}(\omega_1)$, knowing $T = t_*$, Model 2, discrete case (ring data), for pl_0 obtained via the CI method.

4.8 Examples

4.8.1 Simple novelty detection: example 1

The data set considered here is the breast-cancer data obtained from the UCI machine-learning repository [89]. The patterns in this data set belong to two classes: benign and malignant. Each pattern consists of nine cytological characteristics graded with an integer from 1 to 10. As in [58], a uniform noise in $[-0.05, 0.05]$ was added to each value to avoid numerical errors because of the discrete values. After removing patterns with missing characteristic values, the final data set consisted of 683 patterns. This data set was split into a training set of 300 patterns (200 benign, 100 malignant), and a test set of 383 patterns (244 benign, 139 malignant).

Assume that only data from the begnin class are available.We therefore would like to built a classifier that helps doctors making a diagnosis, given nine cytological characteristics of that new patient, and the distribution of a statistic T built from the same nine characteristics measured on benign tumors on other patients.

We need to build a statistic T, that will increase with the risk that the tumor might be malignant, i.e., when data cannot be deemed to come from the same distribution as in the case of a benign tumor. Let us build a KPCA-based one-class classifier from the 200 benign cases of the training data, and use the value of statistic $T = KRE(\mathbf{x})$ defined in Equation 3.16 as our novelty measure.

A KPCA model is built as described in 3.4.4. The kernel bandwidth is determined by the direct pluggin method [93][133, page 71]. Following the three points procedure described in paragraph 4.2, and using Kriegler and Held's algorithm and a Kolmogorov confidence band (see Figure 4.18) as explained in Section 4.6.2, we obtain the possibility distribution $pl_0(t) = 1 - F(t)$ that represents our belief in what the next value of T should be, if the next patient's tumor is benign. It is represented on Figure 4.19. From this, we deduce our belief that the patient's tumor is malignant or not, given the obtained value of T, as described in section 4.6.2. It is shown in Figure 4.20.

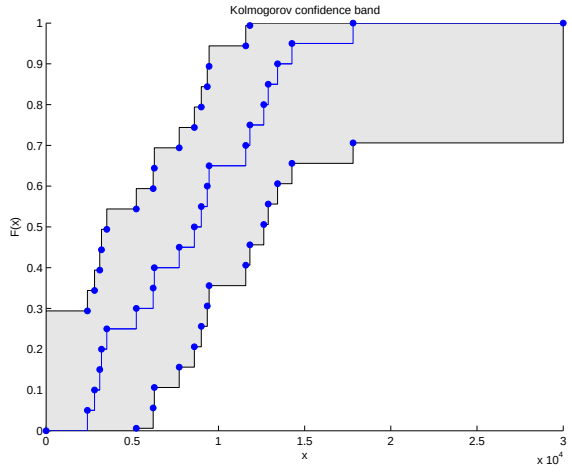


Figure 4.18: Kolmogorov confidence band around the distribution of the value of T for a KPCA-based classifier (Breast-cancer data).

Now suppose that, in order to take the intrinsic variability of biological measures into account, we measure the nine cytological characteristics several times on the same tumor, and obtain an interval $[x; y]$ of possible values for T. The plausibility of this interval under the hypothesis that the tumor is benign is $pl_0([x; y]) = pl_0(x)$. Consequently, there is no need to consider the triangle representation of $pl_0([x; y])$, as it is entirely defined by its profile function $pl_0(x)$.

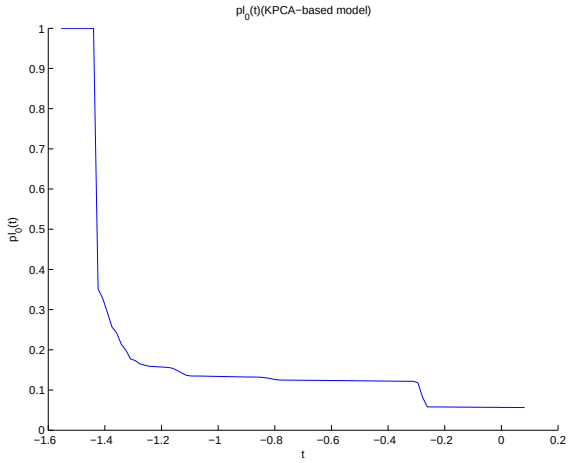


Figure 4.19: Predictive belief function on the value of T for a KPCA-based classifier (Breast-cancer data).

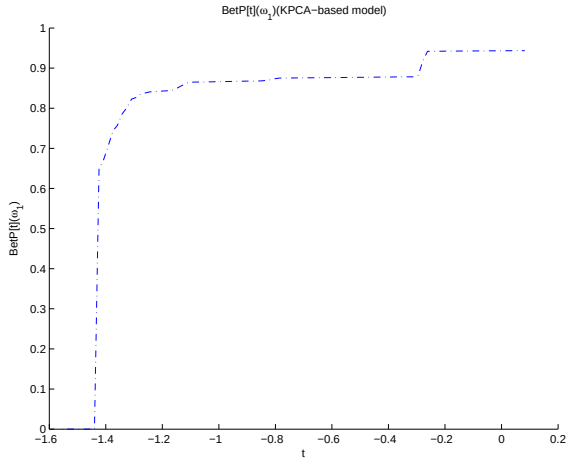


Figure 4.20: Pignistic probability function on ω_1 for a KPCA-based classifier (Breast-cancer data).

4.8.2 Simple novelty detection: example 2

Let us consider an industrial oven out of which fumes are collected and driven into a boiler so as to heat steam up. Suppose we want to evaluate the quality of the combustion. Experts suggested that the following variables be used:

- the percentage of O_2 at boiler exit;
- the steam flow in the boiler;
- the primary air flow in the oven;
- the secondary air flow in the oven;
- the percentage of CO in the chimney.

A one-class SVM model was built from reference data (i.e., measurements taken when the experts considered the quality of the combustion to be as good as possible). Following the three point procedure described in paragraph 4.2, and using Kriegler and Held's algorithm with a Kolmogorov confidence band (see Figure 4.21) as described in Section 4.6.2, we obtain the belief function $pl_0(t) = 1 - F(t)$ that represents our belief in what the next value of T should be, if the combustion quality is still good at the next measurement time. It is represented on Figure 4.22. From this, we deduce our belief that the combustion quality is good or not, given the obtained value of T, as described in section 4.6.2. It is shown in Figure 4.23.

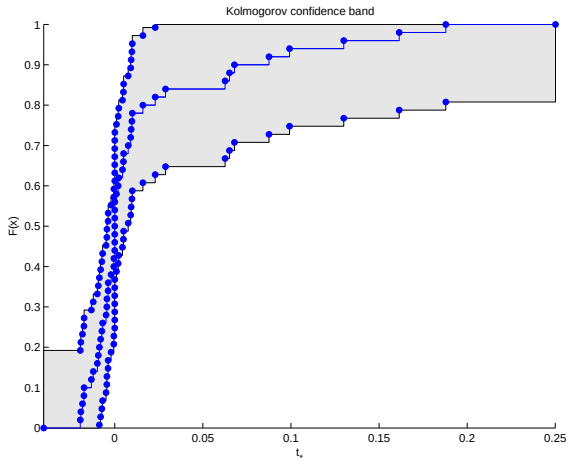


Figure 4.21: Kolmogorov confidence band around the distribution of the value of T for a SVM-based classifier (Combustion data).

4.8.3 Classifier Fusion Example

Problem Description

Consider the breast-cancer data of example 4.8.1. In order to illustrate the ability of our method to combine one-class classifier outputs with other information, we considered the following problem. We assumed that the first six characteristics were available for benign data only, whereas the other three characteristics were available for patterns from both classes. Note that this is a common situation: in many applications, more measurements are available for the class that occurs more frequently. Suppose doctors would like to know whether the modeling of the information on malignant cases could remove part of the uncertainty attached to each diagnosis. The problem is thus to merge:

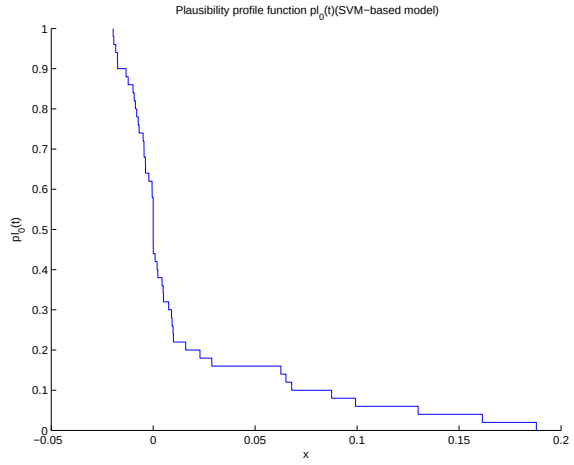


Figure 4.22: Predictive belief function on the value of T for a SVM-based classifier (Combustion data).

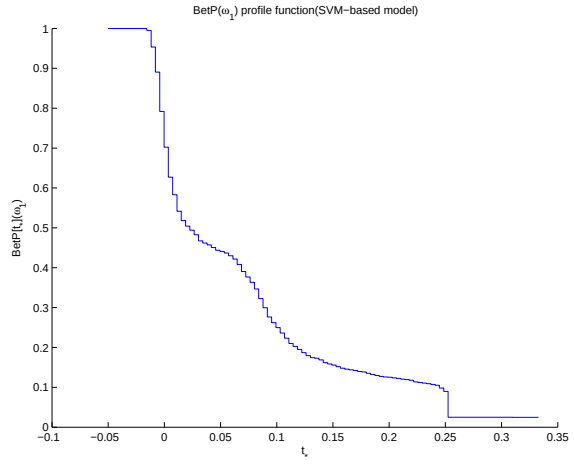


Figure 4.23: Pignistic probability function on ω_1 for a SVM-based classifier (Combustion data).

- the outputs of a one-class classifier trained on observations of the first six characteristics for benign cases only (we will use the classifier built in the previous example);

- with the the outputs from a two-class classifier trained on observations of the other three characteristics for benign and malignant cases.

Results

A two-class classifier based on characteristics 7, 8 and 9 was trained using the evidential neural network method introduced in [39]. This method is grounded on belief function theory, and produces for each input pattern a belief function on Ω.

For each pattern, the belief functions computed by the one-class classifier of section 4.8.1 and the above mentioned two-class classifier were combined using the TBM conjunctive rule (1.24), and the resulting bba was transformed into a pignistic probability function on Ω using (1.40).

Figures 4.24 and 4.25 show test estimates of the Receiver Operating Characteristic (ROC) curves for the one-class, two-class and combined classifiers.

ROC curves are a well-known, widely-used means of representing the performance of a classifier. In a two-class problem (faulty or normal system), they represent the proportion α of errors detected while the system is in a normal state (called false positive) against the proportion $1-\beta$ of faults detected when the system actually is faulty (termed true positive). β represents the percentage of faults that are not detected and should be. The ideal classifiers would minimize both α and β. However, it can be shown that, whatever the classifier, α always increases when β decreases and vice versa. Hence, the best classifier is the one that makes the best compromise, i.e., that minimizes β for a given value of α [49].

On our example, it can be observed that, although the one-class classifier has poor performances when considered alone, combining it with the two-class classifier does result in significantly improved performances. Such a combination has been made possible by expressing the outputs from both classifiers in a common framework.

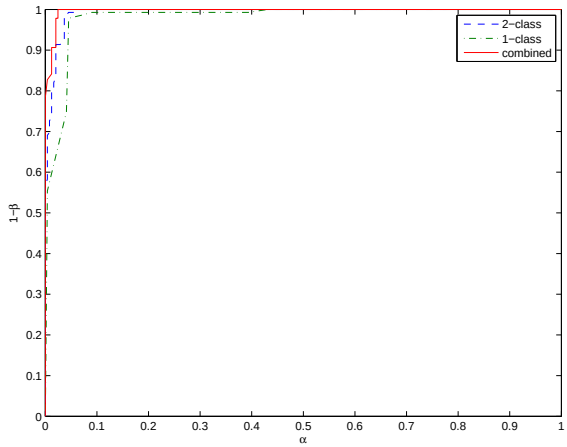


Figure 4.24: Test estimates of the ROC curves for the three classifiers (two-class classifier: dotted line; one-class classifier: dash-dotted line; combined: continuous line) on the breast cancer problem.

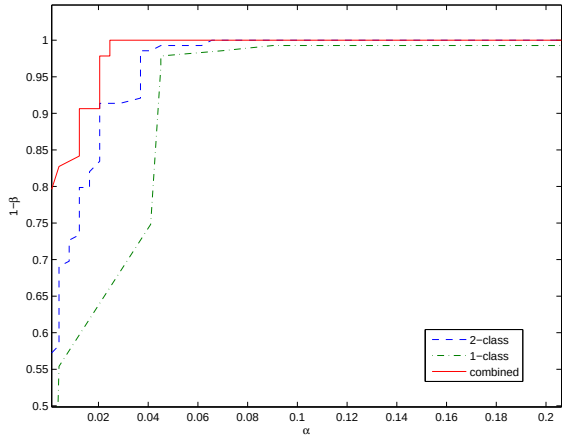


Figure 4.25: Zoom on the top left hand corner of Figure 4.24.

4.9 Conclusion

We built a solution to the problem of testing a null hypothesis under the belief function framework. Our solution takes advantage of the facilities offered by this theory to work with partial knowledge without addition of any assumption. It thus allows us to make a decision when very little information is available. Together with the method developped in Chapter 2, we thus provide the description of the whole process that leads from raw data to the decision stage through the TBM framework.

Moreover, expressing the outputs from one-class classifiers such as one-class SVMs or KPCA in the belief function framework makes it possible to combine them with other information expressed in the same framework, such as other one-class classifiers, evidential multi-class classifiers, or even expert opinions. This approach is expected to be particularly useful in system diagnosis and process monitoring applications, in which data corresponding to abnormal system states are not always available or are scarce. Results in this application area will be reported in upcoming chapters.

Part III

Monitoring of a waste incineration process

A waste incineration process monitoring application

Contents

5.1 Introduction 139
5.2 The waste incineration process 141
5.2.1 Waste combustion 141
5.2.2 Energetic promotion 142
5.2.3 Flue gas purification 142
5.2.4 Pilot plant 145
5.3 PMAT 145

Summary

In this chapter, the industrial project related to this PhD thesis is described in details, as well as the adopted technical solution.
This application concerns the monitoring of a waste incineration process. The principle of this process is very simple: ordinary wastes are burned, and the fumes are sent into a boiler where their heat is taken in by some steam. Cooled-down flue gas come out of the boiler and follow a de-polluting route before they are ejected into the atmosphere. Meanwhile, the overheated steam drives a gas turbine that produces electricity. Stabilization of the process is made complex by the important variability of the waste's calorific value and polluting component composition.
The incineration process is supervised from a monitoring room. Several monitoring screens show synoptics on which appear the instant measures of several sensors spread around the site. Measurements are renewed every 5 to 30 seconds. Operators have to monitor up to 5000 variables permanently while carrying out other tasks at the same time.
For these reasons, it was decided to built a Process Monitoring Assistance Tool (PMAT), that would show a synthesis of essential information on sensors validity, combustion quality, and process safety. It was also decided that it would be based on the principles of one-class classification. In effect, examples of the normal –or desired– state of the system are numerous, while possibles faults are too many, and too varied to allow the building of a training sample. Additionally, the involved data are extremely imprecise and uncertain, and have extremely complex correlations.
Both for financial and deadline-related reasons, the PMAT was built as an add-on to *IP21*, a software already used in most of the company plants as an analysis and reporting tool on the monitoring system. The prototype uses IP21's database and the monitoring statistics are stored into the same database. The user interface consists in a series of screens that show the probability of different faults in the system. The process has been divided into a series of subunit, each of which is monitored by a specific one-class classifier. The division of the original process monitoring problem in a series of subproblems was realized in collaboration with a group of company experts.

Résumé

Dans ce chapitre, le projet industriel auquel cette thèse se rattache est décrit en détails, ainsi que la solution adoptée.
L'application concerne la surveillance d'un système d'incinération de déchets. Le principe de ce procédé est très simple: des déchets ordinaires sont brûlés, et les fumées sont envoyées dans une chaudière où leur chaleur est absorbée par de la vapeur. Les fumées refroidies subissent un procédé de dépollution avant d'être rejetées dans l'atmosphère. Pendant ce temps, la vapeur est utilisée pour entraîner une turbine qui fabrique de l'électricité. La stabilisation du procédé est rendue complexe par l'importante variabilité du pouvoir calorifique des déchets, et de leur composition en polluants.
Le procédé d'incinération est supervisé à partir d'une salle de contrôle -commande. Une série d'écrans montrent des synoptiques sur lesquels figurent les mesures instantanées effectuées par différents capteurs répartis dans toute l'installation. Les valeurs sont renouvelées toutes les 5 à 30 secondes. Les opérateurs doivent surveiller jusqu'à 5000 variables en permanence tout en effectuant d'autres tâches simultanément.
Pour toutes ces raisons, il a été décidé de construire un Outil d'Aide à la Conduite (OAC), qui donnerait en permanence une synthèse des informations essentielles sur la validité des capteurs, la qualité de la combustion, et la sécurité du procédé. Il a également été décidé que cet outil se baserait sur le principe de la classification à une classe. En

effet, les exemples de situations normales ou désirées sont nombreux, alors que les défauts sont trop nombreux et trop divers pour permettre la construction d'un ensemble d'apprentissage. De plus, les données impliquées sont extrêmement imprécises et incertaines, et sont corrélées de manière extrêmement complexe.
A la fois pour des raisons financières et temporelles, le prototype de l'OAC a été construit comme un simple "add-on" à IP21*, un logiciel déjà utilisé dans la plupart des usines comme outil de reporting. Le prototype utilise la base de données d'*IP21 *et les statistiques de surveillance du système sont stockées dans cette même base de données. L'interface graphique consiste en une série d'écrans qui montrent la probabilité de différentes situations anormales et de différents défauts possibles du système. Le procédé a été divisé en une série de sous-unités, et chacune d'entre elles est surveillée à l'aide d'un classifieur à une classe spécifique. La division du problème originel en une série de sous-problèmes a été effectuée en collaboration avec un groupe d'experts de la compagnie.*

5.1 Introduction

Novergie, a subsidiary of *SITA* (waste management division of *Suez Environnement*), specializes in waste incineration and promotion. The company operates 43 waste processing units and 7 waste sorting centers. Over the last ten years, local authorities' requirements in terms of environmental services tightened, and new laws have been enforced. In response, *Novergie* closed 37 old units since 1997, and about 20 of its plants have been certified to comply with the ISO 4001 standards.

In addition, *Novergie* wants to develop a series of tools that will help optimize processes and diagnose -or even possibly forecast- defaults. In this framework, a collaboration had been established with the *Centre International de Recherche sur l'Eau et l'Environnement* (CIRSEE, international centre for research on water and environment, water division of Suez Environnement) and *l'Université de Technologie de Compiègne* (UTC, Compiègne, University of Technology), in order to develop the group's expert knowledge with respect to process monitoring.

The first problem to be explored may be described as follows. A waste incineration process is supervised from a monitoring room. A number of monitoring screens show up to 24 synoptics on which appear the instant measures of the flow, pressure and temperature sensors spread around the site. These represent up to 5000 variables shown as instantaneous values (see Figure 5.1). From these measures, operators must drive the process, in the aim of maintaining constant flow and temperature of the output steam. At the same time, they also have to carry out the feeding of the oven in garbage, and make current repairs and maintenance operations (drainings, calibrations, etc). A series of alarms (lights, tones, etc) aim at assisting them with the monitoring task. Nevertheless, it is extremely difficult to monitor such an important number of measurements while carrying out other tasks simultaneously. In addition, the slow drift of some parameters is difficult to detect with naked eyes when it happens, especially without a time scale. For these reasons, a *Process Monitoring Assistance Tool* (PMAT), that would show a synthesis of essential information on sensor validity, combustion quality, and process safety, would be of precious help to the operators.

Novergie thus decided such a tool had to be developed, and this is what our work is concerned with. The PMAT developed here is designed for the operators, and should be a complement or an assistance to the expert. This tool will be developed as a damage harnessing step, and should therefore have no direct interaction with the process. It should rather be a reference with respect to normal or optimum conditions.

Previous attempts showed that a thermodynamic model of the process is extremely difficult to establish and leads to very imprecise results. In effect, each subprocess is difficult to model in itself, and the interactions between them are no less complex. It was therefore decided that a statistical tool had to be developed instead: the supervised training of monitoring statistics should be a good alternative to modeling.

Consequently, statistics representative of the state of the process must be computed. These statistics will be based on sensor measurements and their values will be compared to those corresponding to the normal conditions. This will allow a clear positioning of the process with respect to normal, risky or dangerous states.

From on-line measurements, the tool will give, in real or slightly delayed time, a synthetic image of the state of the process that should be more elaborate and synthetic than the information currently displayed in the system monitoring room. It will allow the detection and identification of faulty variables.

A full-scale study has been started, that aims at establishing a functional prototype of the PMAT, and testing it on a pilot plant. This PhD work is concerned with the first phase only, namely fault detection, and will concentrate on the oven-and-boiler subsystem. These two units indeed constitute the core of the system, and their regulation should

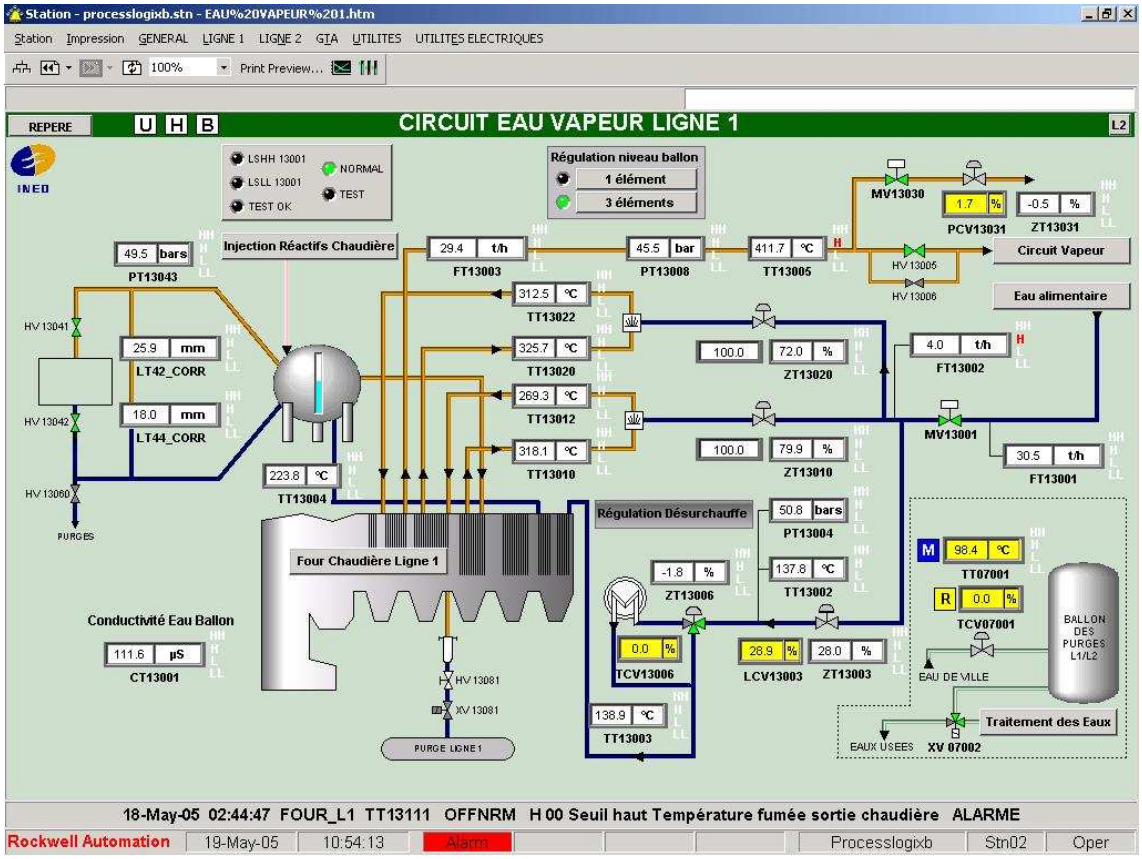


Figure 5.1: Screen shot of one of the monitoring screen

stabilize the whole process. Limiting the approach to the incineration unit will also allow to fit the study into the allocated time (three year PhD project).

The project involves the selection of the variables to be monitored, the elaboration of monitoring statistics, the determination of normal and abnormal ranges for these statistics, and finally, the building and test of the prototype.

5.2 The waste incineration process

Waste incineration is a thermic process that includes the combustion of the waste and the cleaning of flue gas. It produces three types of residue: clinker, ashes, and flue gas cleaning residue (FGCR). The combustion generated heat is promoted in the form of energy (electricity and heat production).

Combustion is a matter degradation, namely an oxidation, with 5 types of emissions:

- Water;
- Gas: CO, CO_2, NO_x, SO_2, HCl;
- Mineral dust (ashes);
- Heavy metals: lead, copper, mercury, cadmium, nickel, arsenic;
- Organic molecules: carbon, chlorinated organic compound.

The incineration process may be separated into 3 steps:

1. Waste combustion;
2. Energetic promotion;
3. Flue gas purification.

A global balance sheet is represented in Figure 5.2.

To each ton of household refuse correspond 3 to 500 kWh of electricity, but 25% of the original weight comes back out in the form of solid residue (clinker and FGCR).

5.2.1 Waste combustion

On their arrival on the site, wastes are weighted and dumped into a pit. An operator driven grab then picks up the garbage and drops it into a hopper that feeds the oven. A pushrod then moves them onto a metallic conveyor belt that allows them to move forward inside the oven. The conveyor belt is made of a series of bars that can turn around on themselves to move the waste so they will be regularly spread on the belt. The combustion is fanned by the blowing of previously warmed up air through the conveyor belt. This first air provision is termed *primary air*.

There are four zones along the belt (i.e., along the oven. See Figure 5.3). The first one is the drying zone, where the water contained in the waste evaporates. The second is called gasification zone. The light gases of the waste are burned there. The third zone is where the combustion actually happens. The last zone serves for the cooling down of the clinker. Additional air –or secondary air– is blown up on the third zone to help a complete gas combustion. A temperature of 850˚C must be maintained on the fumes for at least 2 seconds to allow perfect combustion. Whenever the temperature drops under 850˚C, a gas burner lights up in order to maintain a total combustion. Clinker are collected at the exit of the oven. They are then transported by conveyor belt into a specific storage area.

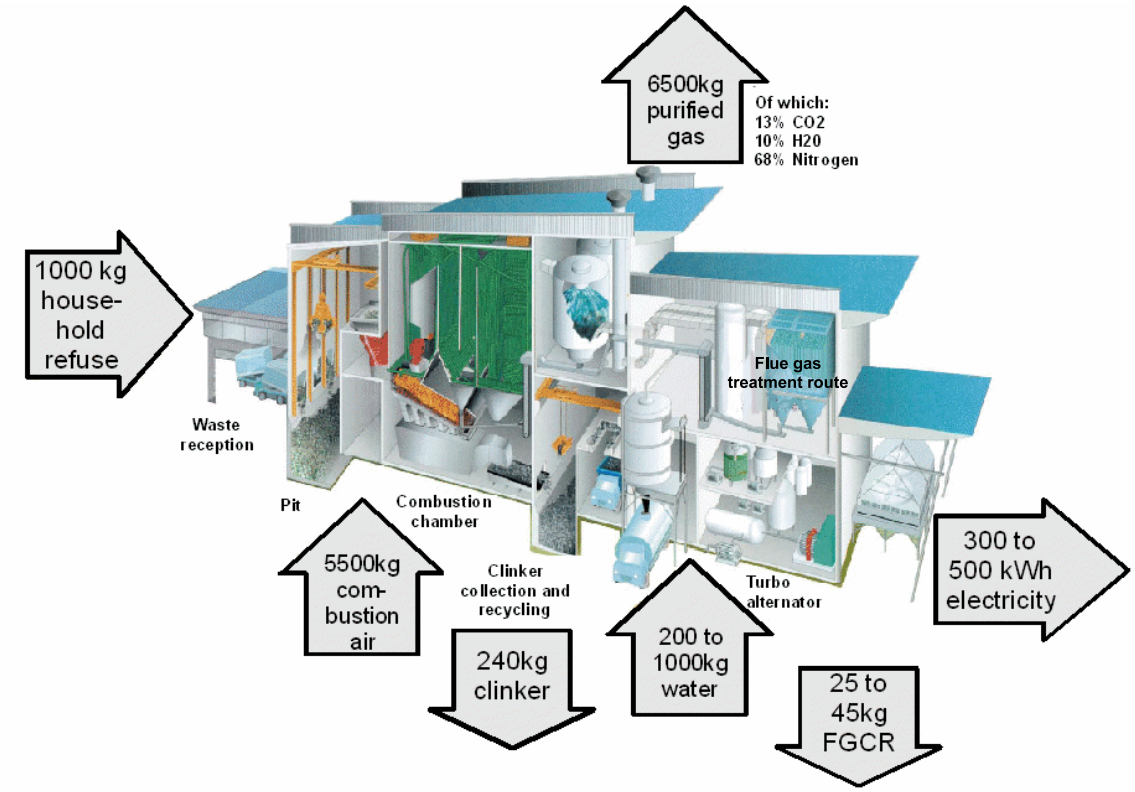


Figure 5.2: Global balance sheet of a waste incineration process.

5.2.2 Energetic promotion

The boilers capture the heat from the waste combustion. In the boiler, a thermic exchange occurs between the combustion fumes and some water. The fumes temperature drops from 1000 down to 200˚C while 400˚C, 45 bar steam is produced, and then used to drive a turbine and produce electricity. The steam collected out of the turbine is condensed in aero-condensers, and the obtained water is then sent back into the system.

5.2.3 Flue gas purification

There exist a number of flue gas purification technologies. The one that is used in the process considered in this study is termed *dry treatment route* (see Figure 5.4). The fumes first go through an electro-filter that eliminates flying ashes and dusts. Sodium bicarbonate and activated charcoal are then blown into the system, that will capture most of the pollutants, mercury, dioxins and furans in particular. Finally, heavy metals are trapped into a fabric-filter.

The standards for atmosphere fume rejection are very strict: fume purification is therefore a key element of the process. The main components of the fumes in the chimney may be listed as follows:

- $N_2 = 70\%$;
- $H_2O = 13$ to 16%;
- $O_2 = 7$ to 10%;
- $CO_2 = 8$ to 11%;

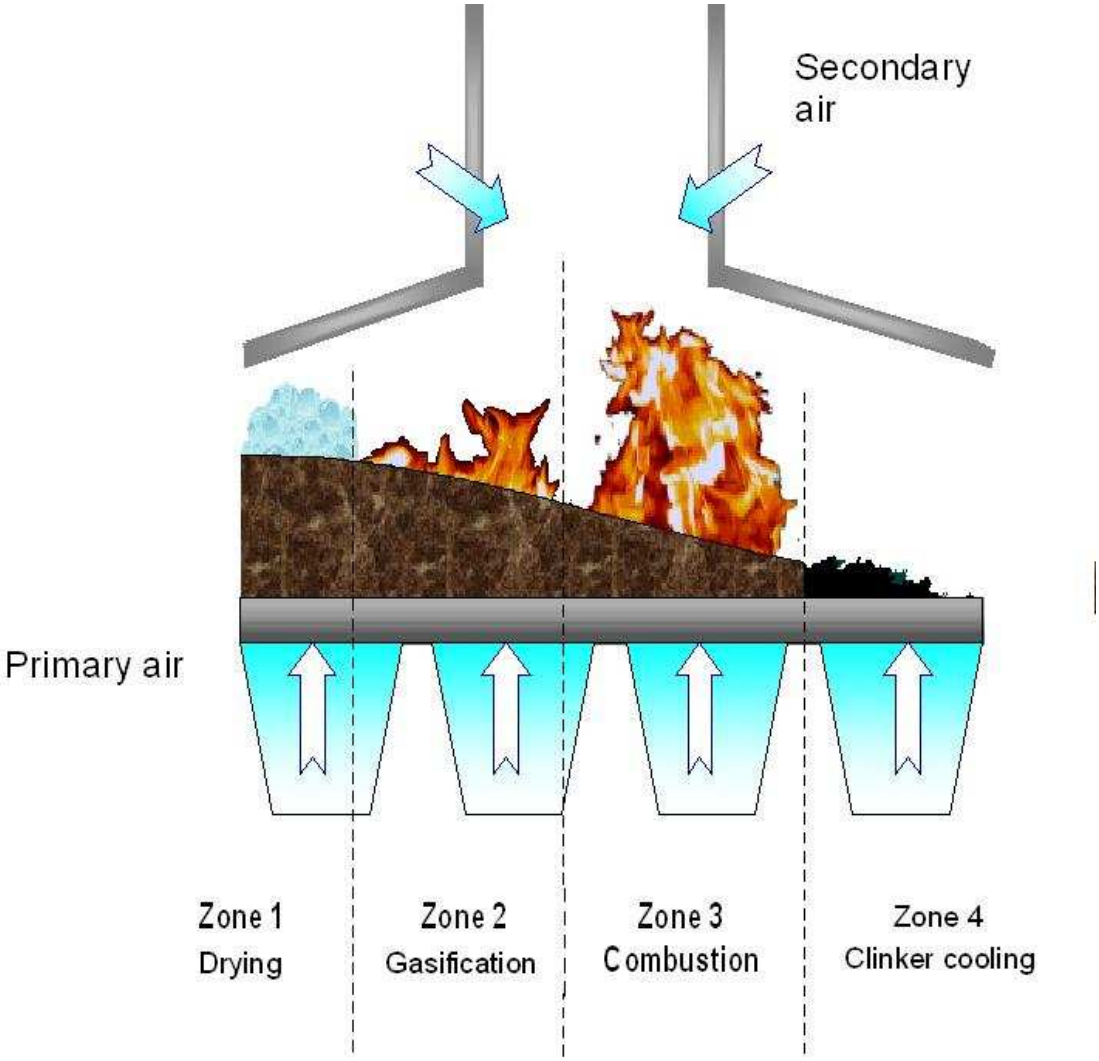


Figure 5.3: The four oven combustion zones.

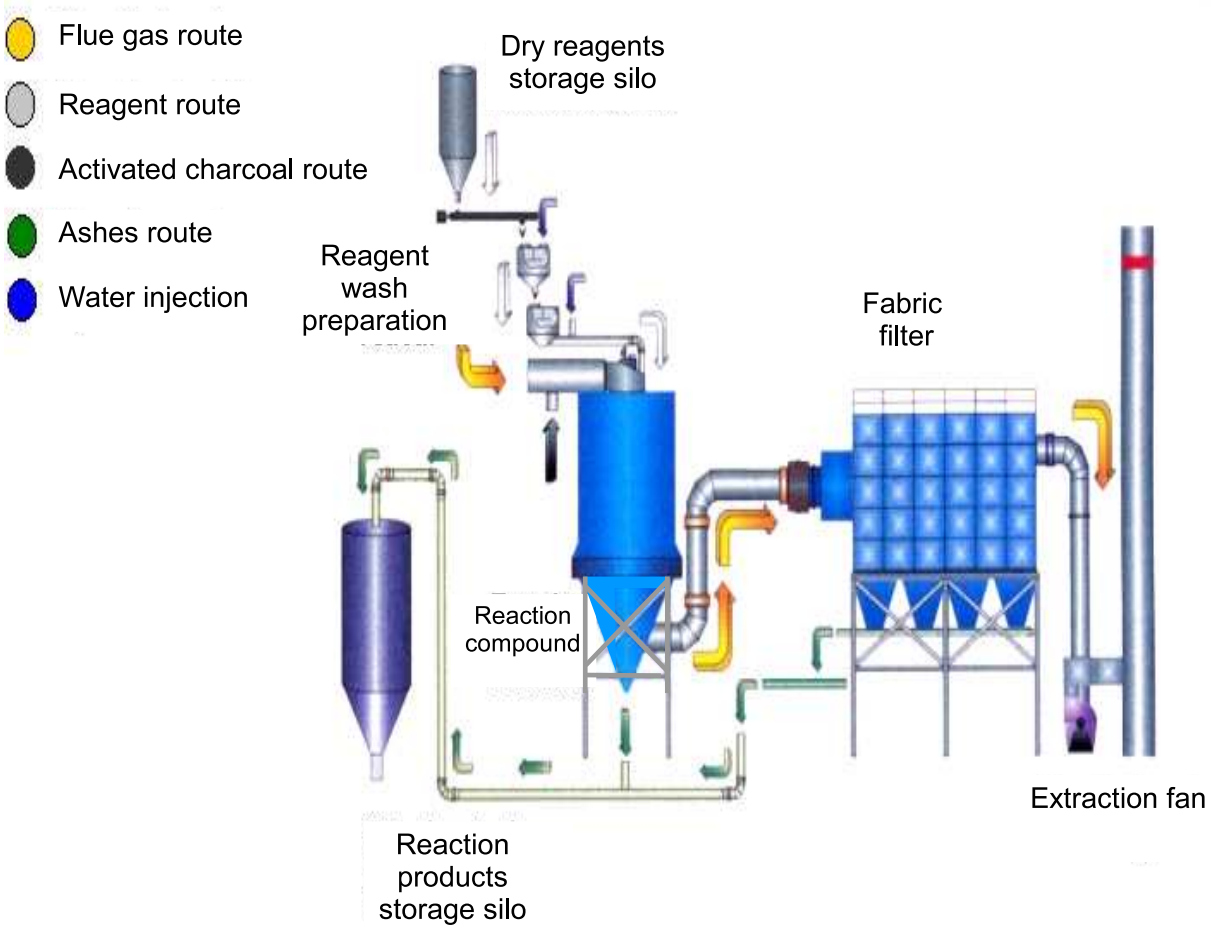


Figure 5.4: Fumes treatment route

- Dusts: $< 5 \quad mg/Nm^3$;
- $HCl : < 10 \quad mg/Nm^3$;
- $NO_x : < 200 \quad mg/Nm^3$;
- $SO_2 : < 25 \quad mg/Nm^3$.

5.2.4 Pilot plant

The chosen pilot plant is that of Villers-St-Paul (France). Household refuse and so-called *banal* industrial waste are incinerated on the site in two ovens of a capacity of 11 t/h each, leading to two boilers that can produce 32 t/h of steam each at a pressure of 45 bars. The energetic promotion is carried out by a 14 MW turboalternator. A dry treatment route allows the cleaning up of the fumes. A total of 35,000 tons of steam and 70,000 MWh are sold out every year. Another 40,000 tons of clinker and 4,400 tons of FGCR-RSC (RSC: Residual Sodium Chemicals) are promoted each year. At the same time, 2000 tons of FGCR a year are sent to a landfill for environmentally hazardous waste.

5.3 The process monitoring assistance tool (PMAT)

The PMAT project was born from the observation of the process. Indeed, waste having a very variable calorific value, a perfect control of the combustion is extremely difficult. However, it is important to obtain a regular overheated steam flow in order to be able to produce electricity. As already mentioned, for an optimal combustion and a good monitoring of the whole process, 5000 variables must be permanently supervised.

These variables are shown as instantaneous values on a series of synoptics. A number of alarms are set on key measurements to assist the operators. Nevertheless, quite a few malfunctions are detected very late, or even only during the plant's annual maintenance shutdown.

The aim of the PMAT is to show a synthesis of essential information on sensor validity, combustion quality, and process safety and stability. Its main objective is the early detection of process malfunctions such as leaks, sensor deviations, process deviations, clogging, and corrosion.

The tool will be developed as a statistic-based process monitoring system. It will allow the comparison of the current state of the system with past states through a model built from labelled observations. The building of monitoring statistics for this problem is a one-class classification problem in that it is not possible to establish an exhaustive and well labeled default set for training, while examples of the normal state are numerous. A previous feasibility study highlighted the highly non linear correlation of the variables to be monitored. Hence, classical linear process monitoring algorithms are not sufficiently efficient to fulfill the desired objectives. It was thus chosen to test kernel based one-class classifiers such as kernel density estimation, kernel principal component analysis and support vector machines. Kernel density estimation was very quickly ruled out for computational issues.

In the next chapter, we will see how KPCA and SVM based one-class classifiers can be used together with belief functions to obtain the desired system. First, we will explain the mathematical construction of the monitoring statistics. Then, the parameter tuning phase will be described. Finally, results will be shown and discussed.

Implementation and results

Contents

6.1 Implementation 151
6.1.1 General structure of the PMAT 151
6.1.2 Implementation of the PMAT 153
6.1.3 Classifiers implemented in the PMAT 156
6.2 Parameter tuning 156
6.2.1 Case study examples 156
6.2.2 SVM-based classification 158
6.2.3 KPCA-based classification 162
6.3 Results 164
6.4 Conclusion 178

Summary

The prototype of the Process Monitoring Assistance Tool (PMAT) was built based on SVM-founded one-class classification. Observations of the statistic $T = -f(x)$, opposite of the output function $f(x)$ of the SVM, was used to build a belief function, using a Kolmogorov confidence band and Kriegler and Held's algorithm, as described in Chapter 2. The obtained belief function was used for novelty detection following the three steps procedure proposed in Chapter 4, Section 4.6.2.
The prototype was tested both on real and simulated faults and showed very good performances. It was then implemented on-line, and proved to detect faults much earlier than human operators, allowing an important gain both in time and money. However, the high variance of the obtained pignistic probability of fault made its interpretation difficult for the operators, and several solutions were proposed to solve the issue. Computational problems were encountered and limited the size of the training samples.
An attempt to work with KPCA-based algorithms encountered even more important computational problems and had to be given up. However, it seemed to lead to less variability in the pignistic probability of fault, and was thus kept as a prospect for the future.
Nevertheless, the obtained results are extremely promising. It was decided that specifications should be written for a complete re-implementation of the algorithms. Meanwhile, the scope of the PMAT is to be extended to parts of the process other than the oven and boiler subunit. Finally, after a second series of full-scale tests and fine-tunings, including in particular a comparison of the results obtained with SVM and KPCA-based algorithms, the PMAT will be deployed in all waste incineration plants of the company.

Résumé

Le prototype de l'OAC se base sur un algorithme de classification à une classe alliant SVM et fonctions de croyance. Une série d'observations de la statistique $T = -f(x)$, opposée de la sortie $f(x)$ du SVM, a été utilisée pour construire une fonction de croyance, à l'aide d'une bande de confiance de Kolmogorov et de l'algorithme de Kriegler et Held, comme indiqué au chapitre 2. La fonction de croyance obtenue est utilisée pour la détection de nouveauté via la procédure en trois étapes décrite au chapitre 4.
Le prototype a été testé à la fois sur des défauts réels et simulés et montre de bonnes performances. Il a ensuite été implémenté en ligne, et détecte les défauts bien plus tôt que les opérateurs humains, ce qui a permis un gain important de temps et d'argent. Malheureusement, la grande variabilité de la probabilité pignistique de défaut obtenue au départ rendait son interprétation difficile pour les opérateurs, et différentes solutions ont dû être proposées et testées pour résoudre le problème. Une solution satisfaisante a finalement été obtenue. Des problèmes de mémoire ont été rencontrés et limitent la taille des ensembles d'apprentissage.
Une tentative de travail avec des algorithmes basés sur la KPCA a échoué à cause de problèmes de mémoire plus importants encore. Toutefois, il semble que les résultats obtenus aient une variabilité moins importante que ceux basés sur les SVM, et la solution a donc été gardée en mémoire pour de futures expériences complémentaires.
Malgré tout, les résultats obtenus sont prometteurs. Il a été décidé qu'un cahier des charges devait être mis en place pour une réimplémentation complète des algorithmes. Dans le même temps, le spectre de l'OAC doit être étendu aux parties du procédé autres que la sous-unité four-chaudière. Enfin, après une seconde série de tests à grande échelle et de réglages fin des paramètres, incluant notamment une comparaison et une tentative de combinaison des algorithmes basés que les SVM et la KPCA, l'OAC sera déployé dans toutes les usines de la compagnie.

6.1 Structure and implementation of the PMAT

6.1.1 General structure of the PMAT

The PMAT aims at monitoring the waste incineration process through a number of on-line measurements. The variables to be monitored have been selected by a group of experts through an extensive analysis of the possible problems and what variables they were more likely to affect. Table 6.1[1] summarizes the conclusions of the experts[2]. The variables were divided into a series of groups. Each group gathers the variables that may contain information about a specific point to be monitored. A point may be understood here as a subprocess, the quality of which should be permanently evaluated, or as a potential problem that may arise and whose likeliness should also be regularly evaluated. The PMAT will thus be divided into subunits, and the monitoring of each of the above points will be performed by a different *unit* of the PMAT. To each unit thus corresponds a specific group of variables to be monitored, that is to say, a specific classification problem.

Hence, for each group of variables, a classifier has to be built to monitor the associated subprocess or potential problem. Let us term *observation* a vector containing the simultaneous values of the variables in one group at instant i. Let us denote P the distribution of the observations when the process is in normal working conditions. The associated classifier must be able to separate observations coming from distribution P from other observations.

For each group of variables, it is thus necessary to label the observations as corresponding to *normal* or *faulty* working conditions. A one-class classifier must then be trained on the data labelled as *normal*. This training operation is carried out off-line. Then, the classifier is set on-line, and classifies each new observation as it is collected. The result of this classification is finally represented on screen together with that of observations associated with other groups of variables. These results must also be stored in a database.

The physical structure of the PMAT therefore needs to allow the execution of both off-line and on-line tasks.

The off-line tasks may be listed as follows:

- Allow a user to set labels on past data, and register these labels in the database;
- Extract from the database the data that are labelled as "normal" for a given group of variables;
- Build a model out of these data (training phase).

The on-line tasks are:

- Use the parameters of this model to monitor the process in real time, that is to say, classify an observation as normal or faulty (classification phase);
- Store the result of this classification in the database;
- Graphically display this result on the screen.

All these tasks must of course be performed for each unit of the PMAT. Labelling and training should be initiated by the user. Classification must then be automatically launched at a user defined frequency.

[1] In fact, this table only shows the groups of variables that will be monitored through one-class classification. Other, simpler tests where used to monitor other (smaller) groups of variables. These are listed in Appendix A.1.

[2] For a better understanding of the table, please note that the economizer, desuperheater and superheater No.1 and 2 are internal elements of the boiler. The upper chamber is a special point in the boiler where the temperature is measured to serve as a reference for process monitoring.

No.	Subprocess or potential problem to be monitored	Associated variables
1	Combustion quality or sensor drift amongst those positioned in the oven.	$\%O_2$ (boiler exit), Steam flow (boiler), Primary air flow(oven), Secondary air flow (oven), $\%CO$ (chimney)
2	Clogging of the boiler, leak in the boiler or sensor drift amongst those positioned in the boiler.	Steam temperature at different points in the boiler (5 measures)
3	Clogging of the boiler, leak in the boiler or sensor drift amongst those positioned in the boiler.	Flue gas temperature at different points in the boiler (6 measures)
4	Clogging of or leak in the boiler (localization of the clogging or leak)	Steam temperature at economizer entrance, Steam temperature at economizer exit, Flue gas temperature at economizer entrance, Flue gas temperature at economizer exit
5	Clogging of or leak in the boiler (localization of the clogging or leak) or bad steam cooling	Steam temperature at superheater 1 entrance, Steam temperature at superheater 1 exit, Flue gas temperature at superheater 1 entrance, Flue gas temperature at superheater 1 exit
6	Clogging of or leak in the boiler (localization of the clogging or leak) or bad steam cooling	Steam temperature at superheater 2 entrance, Steam temperature at superheater 2 exit, Flue gas temperature at superheater 2 entrance, Flue gas temperature at superheater 2 exit
7	Clogging of the boiler or upper chamber sensor(s) drift	Steam flow, Steam temperature at boiler exit, Upper chamber temperature, sensor No. 1
8	Clogging of the boiler or upper chamber sensor(s) drift	Steam flow, Steam temperature at boiler exit, Upper chamber temperature, sensor No. 2
9	Loss of pressure	Boiler depressurization, Flue gas flow, Extractor fan intensity, Total air flow
10	Parasite air entrance in the oven	Flue gas flow, Steam flow, Primary air flow, Secondary air flow, % O_2 (chimney), 1st extractor fan intensity
11	Steam cooling	Cooling water flow, Steam flow, Desuperheating temperature, Steam temperature at boiler exit, Upper chamber temperature No. 1, Upper chamber temperature No. 2

Table 6.1: Variables to be monitored and associated process points

6.1.2 Implementation of the PMAT

The prototype of the PMAT was developed as an add-on to *IP21*, a database management software package already used in most of *Novergie*'s plants. *IP21* allows the collection and storage of an important number of variables. The data are then available for a posteriori analysis of the process and reporting.

The core of the software is a real-time database connected to the control-command server. It is built out of records that have a number of functions and can carry out a number of tasks, just like objects in an object-oriented programming language.

To this database, data processing and graphical representation units may be added. It was thus decided that the prototype of the PMAT would be added to the core of the database in a similar way. It would therefore be able to work directly on the information contained in the database, and the information it would produce would be stored in the database as well.

Hence, the prototype of the PMAT is also made out of a number of *IP21*-records. Each of these records is a subcomponent of the prototype with its own functions. Some are hidden to the user who interacts with them through a graphical interface. On the other hand, the access to some other functions requires the user to work directly in *IP21* and manipulate the associated record. Figure 6.1 summarizes the general architecture of the prototype, and Figure 6.2 shows a single unit.

The tasks listed in Section 6.1.1 are represented on Figure 6.1. As already mentioned, they may be divided between off-line and on-line tasks. The off-line tasks should only be executed upon user request, while the on-line task should be executed automatically at some user-defined frequency. They have been implemented as follows:

- Labelling is carried out by the user through a specific graphical interface.
- Training is initiated by the user through that same interface. The user action in fact activates an *IP21* record that:
  - activates another *IP21* record that extracts the *normal* data from the database and writes them into a .csv file; this task is performed via a SQL+ request;
  - launches a C stand-alone application that reads the data from the .csv file and does the actual training of the classifier. It then writes the result into a .csv file that contains the necessary information for classification.
- Classification is then launched once and for all by the user. It activates an *IP21* record which:
  - Activates another *IP21* record, which gets each new observation on-the-fly and writes it into a .csv file; this task is performed via a SQL+ request;
  - Launches a C stand-alone application that reads the observation from the .csv file, gets the result of the training from the .csv file created during the training phase, and performs the actual classification task; it then writes the result into a .csv file;
  - At some user defined frequency, the record automatically reactivates itself and performs the above two points again.
- The result of classification is then shown on screen and refreshed at the same user defined frequency.
- Simultaneously, it is written into the database.

There are specific *IP21* records associated to specific training and classification C stand-alone applications for each unit of the PMAT. The application is robust towards missing data. The corresponding classifier simply produces a NaN (not a number) result.

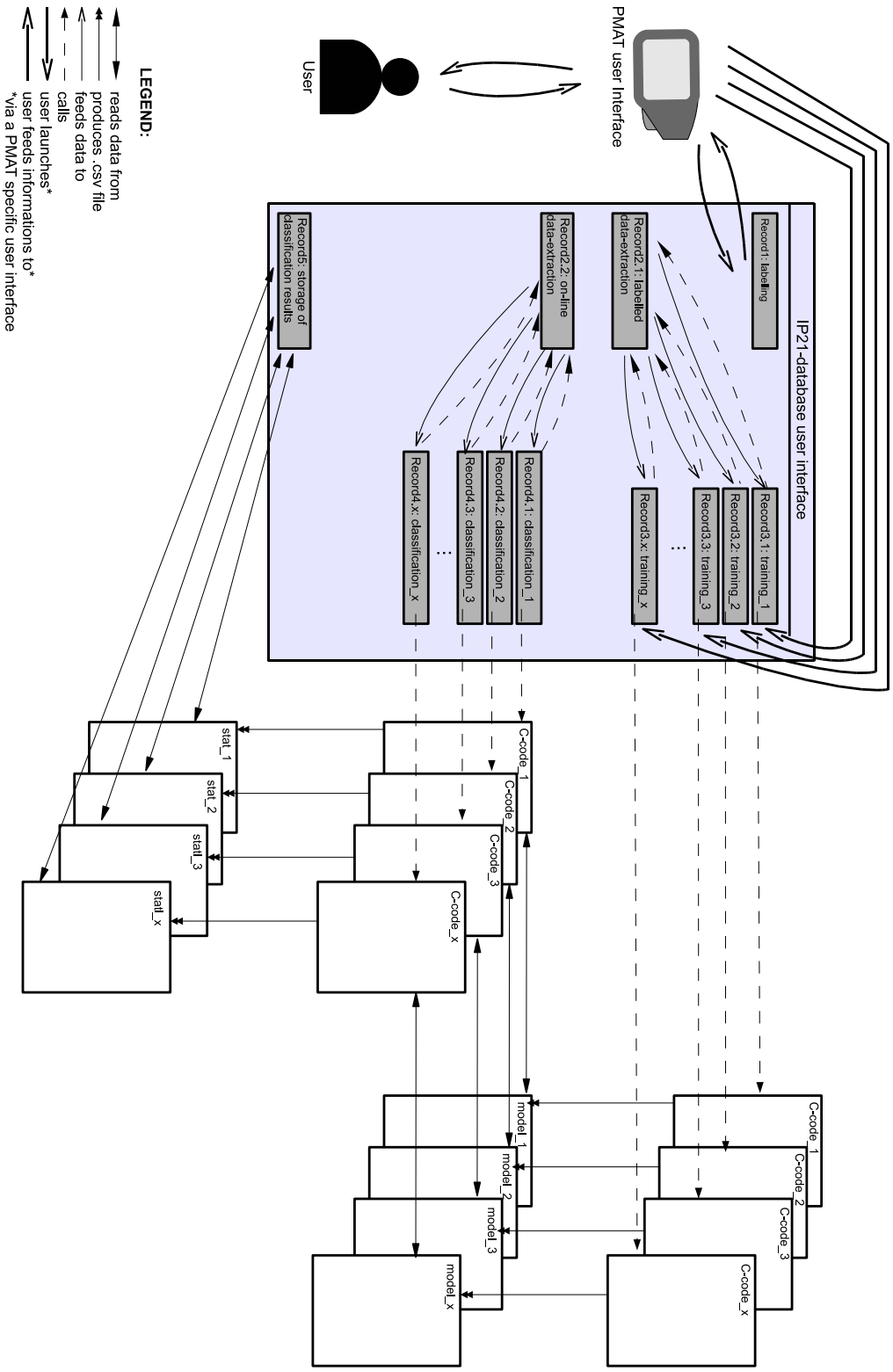


Figure 6.1: Physical structure of the PMAT (general organization)

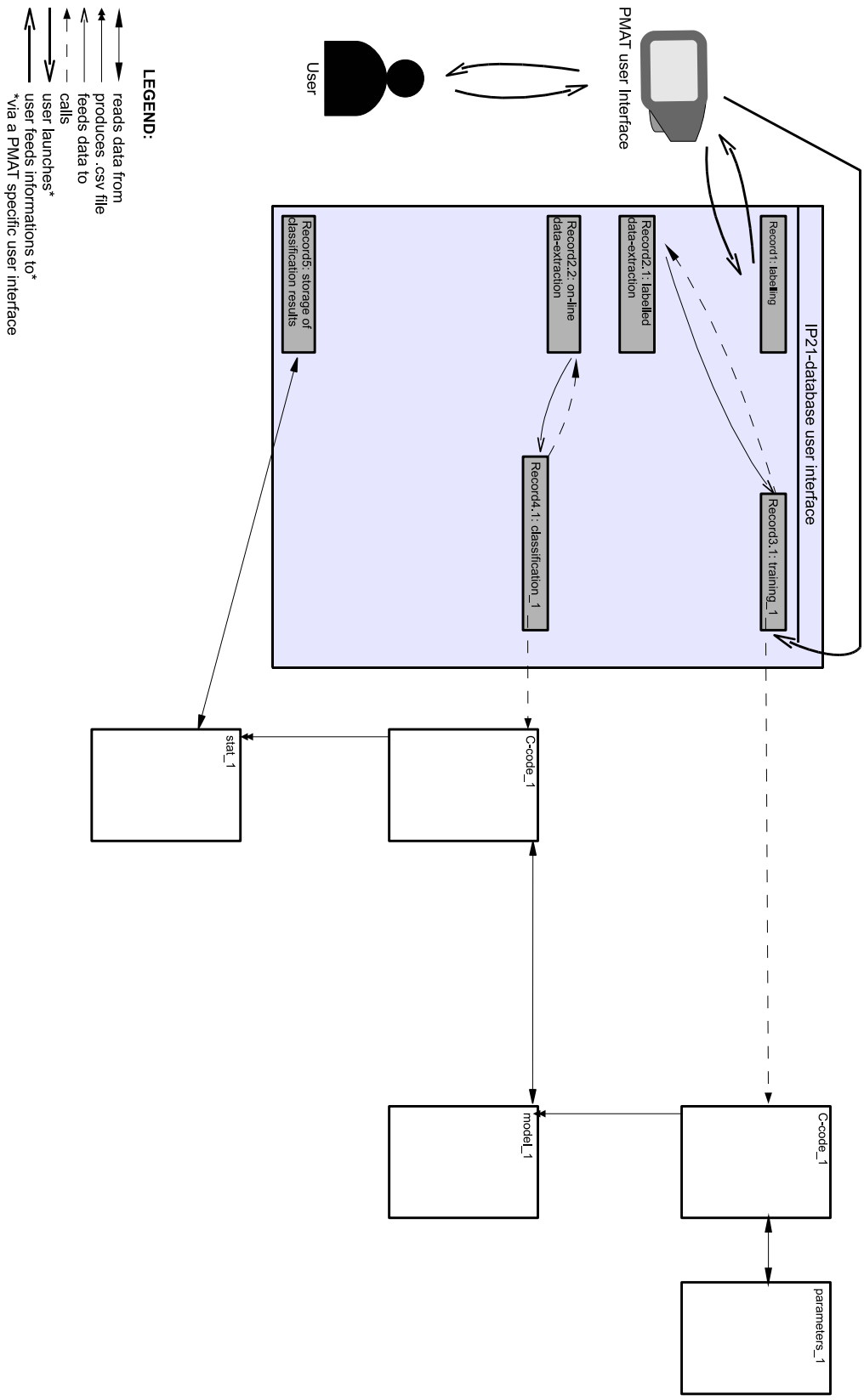


Figure 6.2: A single unit of the PMAT

6.1.3 Classifiers implemented in the PMAT

The classifiers implemented in the PMAT are one-class classifiers from whose result is then built a belief function on the state of the system (*normal* or *faulty*) via the methodology presented in Sections 4.2 and 4.6.2.

Hence, for each unit of the PMAT, a classifier is built from observations made when the process was in normal working conditions. This classifier uses a particular statistic for classification. The distribution of this statistic corresponding to observations labelled as *normal* is then used for the construction of a belief function on the distribution of the statistic when the system is in normal working conditions. This is done via the building of a Kolmogorov confidence band around the cdf of the statistic, and using Kriegler and Held's algorithm to determine the equivalent belief function m, as described in Section 2.2.3. Finally, a belief function m' on the state of the system is deduced from m via the methodology described in Section 4.6.2. The results stored into the database and presented on screen are the pignistic probabilities of fault associated with the belief functions obtained for each unit of the PMAT.

The one-class classifiers used for the prototype are Gaelle Loosli's Matlab implementation of SVM [75], and Roman Rosipal's implementation of KPCA [104].

Both these codes were embedded in some additional Matlab code to allow correct interfacing with *IP21* and automatic reading of the necessary data and parameters. As already mentioned, data are fed to the training algorithm via a .csv file. Another .csv file is used to provide the user defined parameters (e.g. parameters C and h for a SVM). For the sake of clarity, the .csv parameter file specific to each training algorithm does not appear in Figure 6.1; it is shown in Figure 6.2. Finally, the results are written in a last .csv file before it is copied into the database via some *IP21*-record.

In the sequel, we will describe how parameters were tuned for each unit of the PMAT and analyze the quality of the obtained classification results.

6.2 Parameter tuning

6.2.1 Case study examples

We will explain how parameters were set on two case study examples. The same procedure was repeated over each group of variables that needed monitoring, i.e., for each unit of the PMAT.

Example 22. *Our first case study example is a group of variables considered by the expert as being representative of the quality of the combustion. It encloses:*

- *the percentage of O_2 in the air at the exit of the boiler,*
- *the primary air flow,*
- *the secondary air flow,*
- *the steam flow,*
- *and the quantity of CO in the chimney.*

The corresponding PMAT unit will be termed combustion quality unit.

This example was chosen as a key element of the process. Indeed, the quality of the combustion conditions the quality of everything else in the process: a good combustion allows a steady output steam flow (hence good electricity production), low pollutant rates in the flue gas, etc. The monitoring of this group of variables is thus one of the most

important tasks the PMAT will have to carry out. For the same reasons, this example was also the one the experts used to define labelling rules, which are not described in this report, as they are process-specific, if not plant-specific.

Example 23. *The second case study example simply includes five measures of fumes temperature taken in different key-elements of the boiler.*

Test and training set selection

Two data sets where used to build the classifier:

- **A training set** made out of about 5000 observations selected over a 12 weeks period. These observations were selected as corresponding to an expert-defined acceptable working-order of the process for a desired output steam flow of 32 t/h. The exact number of observations varied depending on the PMAT unit to be tested as, at the same time step, some variables may be valid while other are not.

- **A test set** made out of 3000 to 5000 observations –depending on the PMAT unit to be tested– selected over a 12 week period as well. This test set did not overlap the training set. Observations were labelled as normal (corresponding to an acceptable working order of the process) or faulty (abnormal), to allow a later evaluation of the classifier's performances. Additionally, a series of artificial faults were inserted (and labelled as system faults) to cover for possible faults that needed be detected but for which no real example was available in the data base.

Simulation of faults

On site fault simulation For obvious safety and environmental reasons, it was not possible to make the process actually go wrong and collect the corresponding data. Based on expert knowledge and experience of past faults on other sites, it was thus decided to manipulate the sensors in such a way that the returned measurements would correspond to those of a faulty situation.

It would have been interesting to be able to decalibrate some of the sensors to simulate, e.g., slow drifts. However, most of them were too fragile to allow that sort of manipulation without undergoing the risk of a real damage. Moreover, recalibration would have required a considerable amount of time. Hence, it was decided to change some of the sensor scales, based on HART protocol[3].

Example 24. *For example, it was possible to simulate a leak in the boiler by modifying the scale of the water flow sensor at the entrance of the boiler. The water entrance flow therefore seemed lower than it actually was, while the steam output flow –whose associated sensor had not been modified– kept its normal value. This made things look as though more steam was produced than the boiler was fed with in water, hence suggesting a leak from the flue gas circuit of the boiler into the steam circuit (or a decalibration or drift of the sensors, or an increase in the permanent draining of the boiler).*

Matlab simulations: Not all sensors could be manipulated, hence a number of numerical simulations were carried out under Matlab. Three types of sensor faults were simulated in this way: complete breakdown, decalibration and slow drift. Figure 6.3 shows the original signal (a), a complete breakdown (b), a decalibration (c), and a slow

[3]HART (Highway Addressable Remote Transducer) protocol allows the communication of a sensor with compatible instrumentation. It allows dialog between transmitters and regulation gates. Communication through a modem gives access to all the sensor properties.

drift (d) for a temperature sensor. Faults (b) and (c) should obviously be detected instantly. The PMAT should also be able to detect fault (d) a long time before the human operator.

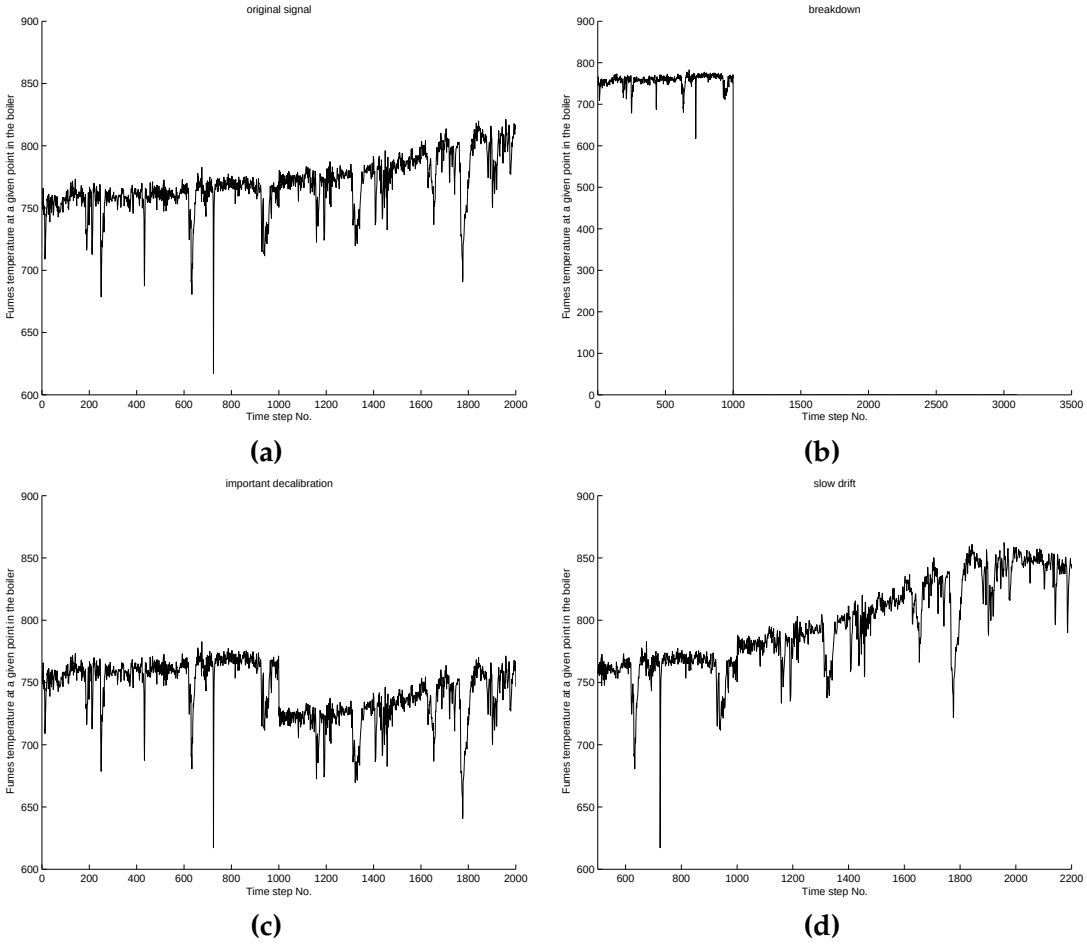


Figure 6.3: Matlab default simulation
original signal (a), complete breakdown (b), decalibration (c), and slow drift (d)

6.2.2 SVM-based classification

In the case of the SVM-based classifier, only two parameters needed tuning:

Parameter C**:** It is the equivalent of the smoothing parameter $(\nu n)^{-1}$ of the decision function f defined in Equation 3.20. It influences the way misclassified training points are penalized. Heuristic assessment shows that, for one-class SVMs, there is a threshold above which classification results are good provided the other parameters are properly set, and increasing values of C do not change the results. For values of C under that threshold, classification results are generally quite poor. Hence, it suffices to find this threshold experimentally and use a value of C greater than or equal to that threshold value.

Parameter h**:** It denotes the kernel bandwidth. If h is too large, the frontier will be too loose, and too many points will be accepted inside the frontier. On the other hand, if h is too small, the frontier will be too tight and the generalization capacity of the classifier will be poor. It was suggested in [25] that, for SVMs, the default value for h was chosen

as half the mean Euclidean distance between two training vectors. However, this being an empirical rule, we only used it as a starting point to try and adjust the value of h.

Parameter tuning

The evaluation of the system performance was mainly based on two statistics, namely the type I and type II errors described in Section 3.2. Recall that the type I error, or false negative rate, α, is the probability of detecting a fault when the system is in the normal state and the type II error, or false positive rate, β, is the probability of not detecting the fault when the system is in a faulty state.

The values of these two statistics were evaluated on the test data set for a number of pairs (C, h).

First, it was necessary to calculate the false positive and false negative rates (α and β) for the test data, on a series of classifiers set with different pairs (C, h). The results are shown in Tables 6.2 to 6.3. It was decided that, for the calculation of α and β, we would consider a fault to be detected if the pignistic probability of fault was above some threshold t_{pig}. This threshold value needed to be fixed before the experiment could start, which would allow calculating the desired values of α and β.

Figure 6.4 shows the value of the pignistic probability of fault with respect to the value of the SVM function $-f$ for $C = 100$, and $h = 1.4128$, both for training (continuous line) and test points (dotted line). The aspect of the curve was similar for other values of C and h. It shows that, for values of $-f$ that are not in the range of values obtained for the training set, the pignistic probability of fault increases drastically, in a very abrupt manner. Additionally, the greater values of $-f$ obtained for training data induce high values of the pignistic probability of fault, hence the fault detection threshold t_{pig} should be set above those values in order to avoid a number of type I errors. A slower slope would have allowed more choice for the value of t_{pig}, and more freedom of choice to the experts for the compromise between type I and type II error rates.

The obtained results lead us to try two values of t_{pig} only, that is to say, 0.9 and 0.95. Two tables were thus obtained, namely Tables 6.2 and 6.3.

It can be observed that the best possible false negative rate is between 2 and 10%, leading to a false alarm rate of about 30% on the test set.

Then, the system experts selected the pairs (C, h) they considered acceptable, and a complete ROC-curve was established for the performances of the fault detection system when used with each of these parameter pairs. Finally, the chosen pair was manually selected from these curves by the system experts.

Figure 6.5 shows the obtained ROC-curves. The selected pair was $(C, h) = (200, 1.1886)$, with a threshold $t_{pig} = 0.95$.

Discussion

The obtained performance on the test set might not seem outstanding. Nevertheless, they constitute a great improvement on human system monitoring (until then considered the best existing solution), as will be shown in Section 6.3. The system experts thus considered these results as satisfying and helpful. Moreover, for computational time issues, the training data set was reduced to a minimum size for the parameter tuning operations. Once the range of the possible pairs of parameters reduced, it would have been interesting to see whether the performance could be improved by a larger training set. However, experts were missing time to label a greater number of observations during the tests. Better performance can thus be expected from the final tool, after the training set will have been enlarged.

Another issue is that the desired output steam flow is in fact not constant, and may vary between 25 and 32 t/h. First tests included observations corresponding to that full

$t_{pig} = 0.95$		C				
h		**90**	**95**	**100**	**120**	**150**
0.3	$\alpha =$	0.1986		0.5174		
	$\beta =$	0.2751		0.2349		
0.5	$\alpha =$	0.0941		0.1045	0.0941	0.4669
	$\beta =$	0.247		0.2851	0.3414	0.241
0.7	$\alpha =$	0.1951	0.2108	0.1185	0.1429	0.0418
	$\beta =$	0.2992	0.3233	0.2972	0.3956	0.3454
0.8	$\alpha =$	0.0819	0.0453	0.0366	0.054	0.1812
	$\beta =$	0.3072	0.3635	0.3153	0.3153	0.3052
0.9	$\alpha =$	0.3589	0.3589	0.3589	0.3589	0.3589
	$\beta =$	0.2952	0.2952	0.2952	0.2952	0.2952
1	$\alpha =$	0.0139	0.1446	0.1115	0.1655	0.0767
	$\beta =$	0.3313	0.3855	0.3876	0.3454	0.3434
1.15	$\alpha =$	0.2091	0.0662	0.5	0.1899	0.1063
	$\beta =$	0.4578	0.3855	0.3494	0.3474	0.3253
1.1886 ($\sqrt{1.4128}$)	$\alpha =$	0.0592	0.0366	0.2247	0.1341	0.0139
	$\beta =$	0.3474	0.3373	0.3655	0.3534	0.4418
1.3	$\alpha =$	0.2578	0.2003	0.1028	0.2056	0.7021
	$\beta =$	0.2952	0.3695	0.3574	0.4699	0.3876
1.4128 (default)	$\alpha =$	0.1359	0.3031	0.2352	0.0401	0.0226
	$\beta =$	0.2691	0.3213	0.3012	0.4378	0.3253
1.5	$\alpha =$	0.0296	0.0436	0.0697	0.0192	0.0192
	$\beta =$	0.3554	0.3373	0.4416	0.3514	0.3514
1.6	$\alpha =$	0.0192		0.0192		0.0192
	$\beta =$	0.3574		0.3574		0.3574
1.7	$\alpha =$	0.0314		0.0314		0.0314
	$\beta =$	0.4378		0.4378		0.4378
1.8	$\alpha =$	0.0976	0.0052	0.0557	0.0174	0.0557
	$\beta =$	0.3012	0.4197	0.3534	0.4096	0.3032
1.9	$\alpha =$	0.1272	0.1272	0.1272	0.1272	0.1272
	$\beta =$	0.3032	0.3032	0.3032	0.3032	0.3032
1.996 ($= 1.4128^2$)	$\alpha =$	0.0366	0.0889	0.0889	0.0087	0.0122
	$\beta =$	0.3715	0.3916	0.3916	0.3695	0.4518
2	$\alpha =$	0.047		0.047		0.047
	$\beta =$	0.3916		0.3916		0.3916

Table 6.2: Values of α and β obtained on the test data set for $t_{pig} = 0.95$ and varying values of C and h.

$t_{pig} = 0.9$		C										
h		**50**	**80**	**90**	**95**	**100**	**120**	**150**	**170**	**200**	**300**	**1000**
0.3	$\alpha =$					0.7003						
	$\beta =$					0.1807						
0.5	$\alpha =$	0.0662	0.0889	0.1446	0.1063	0.1394	0.1359	0.5679	0.1603	0.7143	0.0209	0.0209
	$\beta =$	0.2129	0.2932	0.2209	0.2289	0.255	0.3153	0.1526	0.259	0.1225	0.3916	0.3916
0.7	$\alpha =$	0.1725	0.7735	0.3937	0.3415	0.2021	0.2787	0.0941	0.0488	0.1324	0.2962	
	$\beta =$	0.2831	0.1727	0.243	0.257	0.239	0.3373	0.2932	0.2952	0.2108	0.257	
0.8	$\alpha =$	0.2108	0.0854	0.122	0.1115	0.0889	0.108	0.3711	0.108	0.108	0.3206	0.3206
	$\beta =$	0.2791	0.3474	0.247	0.3333	0.247	0.2631	0.2309	0.2711	0.2912	0.2108	0.2108
0.9	$\alpha =$	0.6829	0.6829	0.6829		0.6829	0.6829				0.6829	
	$\beta =$	0.1747	0.1747	0.1747		0.1747	0.1747				0.1747	
1	$\alpha =$	0.4582	0.5958	0.0174	0.2753	0.2596	0.4111	0.2561	0.1794	0.0976	0.9686	0.9686
	$\beta =$	0.2912	0.1667	0.3032	0.3394	0.3434	0.243	0.3052	0.2651	0.3735	0.2329	0.2329
1.15	$\alpha =$	0.1272	0.1167	0.7474	0.1341	0.9425	0.3554	0.4721	0.223	0.4599	0.5105	
	$\beta =$	0.2851	0.2189	0.2932	0.3253	0.1827	0.2651	0.1968	0.2651	0.239	0.1908	
1.1886($= \sqrt{1.4128}$**)**	$\alpha =$	0.399	0.2753	0.1429	0.0592	0.7003	0.2578	0.0244	0.5035	0.0801	0.1289	0.1289
	$\beta =$	0.1285	0.2831	0.2731	0.2972	0.2289	0.3072	0.4217	0.1245	0.2209	0.2771	0.2771
1.3	$\alpha =$	0.2021	0.1307	0.6916	0.5192	0.1951	0.5174	0.9094	0.101	0.101	0.101	0.101
	$\beta =$	0.2229	0.2972	0.1727	0.2329	0.2932	0.3635	0.2711	0.259	0.259	0.259	0.259
1.4128(default)	$\alpha =$	0.0087	0.054	0.3484	0.8693	0.7125	0.0906	0.0453	0.1324	0.2021	0.1638	0.1638
	$\beta =$	0.4839	0.3173	0.1747	0.1446	0.1647	0.3855	0.2912	0.3373	0.2791	0.3454	0.3454
1.5	$\alpha =$	0.5453	0.1655	0.0418	0.0592	0.2509	0.1045	0.1045		0.1045		
	$\beta =$	0.1707	0.3755	0.3032	0.3173	0.3133	0.2791	0.2791		0.2791		
1.6	$\alpha =$		0.0244		0.0244	0.0244			0.0244			
	$\beta =$		0.3112		0.3112	0.3112			0.3112			
1.7	$\alpha =$		0.0941		0.0941	0.0941			0.0941			
	$\beta =$		0.3896		0.3896	0.3896			0.3896			
1.8	$\alpha =$	0.2247	0.0244	0.5749	0.0192	0.1341	0.0418	0.2404	0.1098	0.054	0.054	
	$\beta =$	0.2229	0.3253	0.1466	0.3876	0.2912	0.3574	0.2149	0.2671	0.3092	0.3092	
1.9	$\alpha =$	0.0941	0.6498	0.6498		0.6498					0.6498	
	$\beta =$	0.3795	0.1606	0.1606		0.1606					0.1606	
1.996($= 1.4128^2$**)**	$\alpha =$	0.4721	0.0296	0.0592	0.1829	0.1829	0.0226	0.0209	0.2021	0.0331	0.1498	0.1376
	$\beta =$	0.1245	0.3916	0.3273	0.2711	0.2711	0.3394	0.4357	0.3072	0.3735	0.2892	0.3233
2	$\alpha =$	0.0836				0.0836					0.0836	
	$\beta =$	0.3735				0.3735					0.3735	

Table 6.3: Values of α and β obtained on the test data set for $t_{pig} = 0.9$ and varying values of C and h.

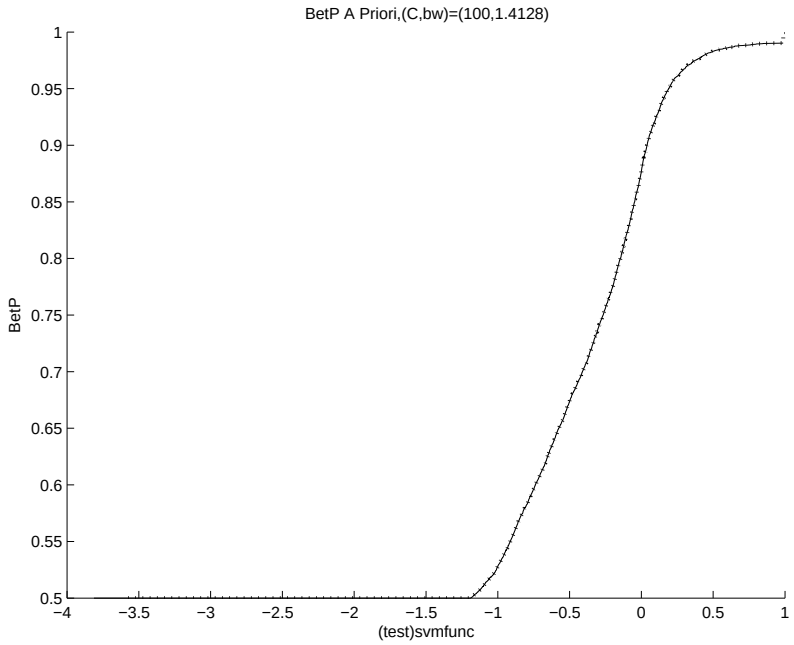


Figure 6.4: $BetP(\omega_1) = g(-f)$
Value of the pignistic probability of fault with respect to the value of the SVM's function $-f$ for $C = 100$, and $h = 1.4128$.

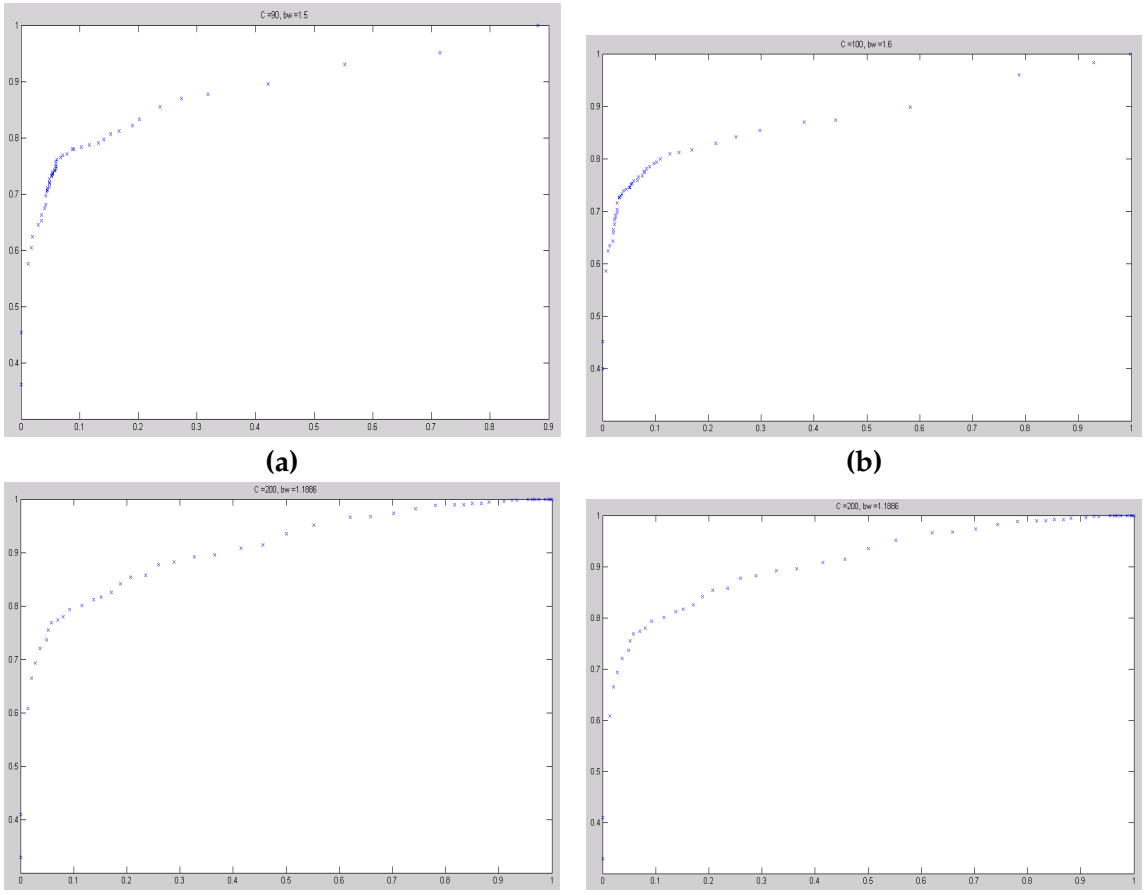
range of possible output steam flow, both in the training and test sets. However, the obtained performances in fault detection were very poor. We thus decided to work on a single value of the desired output steam flow, and chose the most common value, namely 32 t/h. Observations later showed that a classifier built in these conditions works well in an interval of 5 t/h centered on the targeted value of the output steam flow. It will therefore be necessary to train a classifier for a series of intervals of the desired output steam flow, and automatically select the parameter settings of the classifier according to the current desired output steam flow. The proposed intervals are [25,28], [28,30], and [30,32] t/h.

6.2.3 KPCA-based classification

In the case of the KPCA-based classifier, only two parameters needed tuning:

Parameter h: it denotes the kernel bandwidth. It allows smoothing the limit around the training points outside of which a test point will be deemed to be novel. If h is too large, the frontier will be too loose, and too many points will be accepted inside the frontier. On the other hand, if h is too small, the frontier will be too tight and the generalization capacity of the classifier will be poor. Many algorithms are described in the literature for an automated, data-dependent, choice of h. A comparison of the different techniques may be found in [93, 94, 133]. We used the direct plug-in method described, e.g., in [133], as it has small time and memory requirement. Then again, this was only used as a starting point to try and adjust the value of h.

Parameter p: it denotes the number of selected principal components. There exist a number of algorithms that allow an automatic selection of the principal components to



(a) **(b)**

(c) **(d)**

Figure 6.5: Parameter selections
ROC-curves for parameters (a) $(C, h) = (90, 1.5)$, (b) $(C, h) = (100, 1.6)$, (c) $(C, h) = (200, 1.8)$, (d) $(C, h) = (90, 1)$. Each curve represents, for the corresponding set of parameters, the portion α (abscissa) of errors detected while the system is in a normal state (called false positive) against the portion $1 - \beta$ (ordinate) of faults detected when the system actually is faulty (termed true positive). The best classifier is the one that maximizes $1 - \beta$ for a given value of $alpha$.

be retained. Amongst others, the cumulative percent variance [4], the average eigenvalue [5], the variance of the reconstruction error [6] [131] can be mentioned. However, most of these algorithms have been designed for simple PCA. We used the average eigenvalue algorithm as it is easy to adapt to the kernel PCA case, computationally inexpensive, and gave good results on a number of classical toy examples. This parameter was always set automatically.

Parameter tuning

The evaluation of the system performance was based once again on the type I and type II errors. The value of these two statistics where evaluated on the test data set for a number of values of h, p being set automatically, and for the same values (0.9 and 0.95) of the fault detection threshold t_{Pig} as in the SVM case.

Discussion

The results were obtained with a training set restricted to only 2,000 observations. For a greater number of data, the memory requirements exceeded the computer capacity and the program had to stop. The obtained results may thus be improved by increasing the number of training data and working with a better optimized version of the KPCA algorithm. However, whatever the novelty detection statistic used with the KPCA model (T^2, SPE or KRE, see Section 3.4.4), its computation involves all training points, thus making the computation time very long.

Whenming, Cairong and Li's sequential algorithm [137] for KPCA does not include the methodology for calculating T^2, SPE or KRE in a sequential manner as well, but the way to do it is fairly straight-forward from their work. However, it requires the storage of a number of intermediate results. Writing and reading them from a file or a database slows the calculation down and finally leads to a similar problem to the one encountered with the non sequential version.

Computational problems might thus be a provisional limitation to the use of PCA in real life process monitoring in the case where the non kernel version is not sufficient to model the process variables correlations. However, our work was not concerned with optimizing algorithms and it might be possible to get around the difficulty by giving it more thought.

6.3 Results

Once a satisfying parameter set had been chosen, the classifiers were installed on-line and a period of observation of their performance in real-life monitoring of the system started. Operator detected faults were checked against the classifiers' statistics. The classifiers showed good performance and proved to detect faults much earlier than human operators. This part was left in the hands of process experts and operators, who will be the final users. Implementation choices were thus made with an engineering and practical eye. The evaluation of the results was more qualitative than quantitative, and mainly based on a comparison between what could be done with the PMAT and what was usually done without it.

[4] The percent variance captured by the first p PCs is measured and the value of p that allows retaining say 80% of the variance is selected.

[5] This criterion accepts all PCs corresponding to eigenvalues greater than the average eigenvalue. The reason is that the PC contributing less than an average variable is insignificant.

[6] This approach consists in calculating the variance of the reconstruction error (VRE, defined by Qin and Dunia [97]) for each possible value of p, and then selecting the value of p that minimizes the VRE.

Results for the SVM-based classifier

The three types of sensor or process faults described in Section 6.2.1 and shown in Figure 6.3, namely sensor breakdown, sensor decalibration, and slow drift, were tested for each monitored group of variables. Faults were simulated for each case, and, whenever possible, past (extracted from the database) or current real faults were also used to test the classifier. A few examples of the results are given below.

Sensor breakdown The detection of the realistic simulation of a sensor breakdown was instantaneous. In such a case, the sensor indicates a constant value that equals its maximum or minimum range (see Figure 6.3(b)). Consider the data of example 22 (see Section 6.2.1). A SVM based novelty detector was built from these variables and the associated pignistic probability of fault was computed. A breakdown was simulated on the sensor that measures the percentage of O_2. Figure 6.6 shows the faulty signal (top) and the pignistic probability of fault (bottom) at the same time steps. It can be observed that the fault is correctly detected.

Now consider the data of example 23 (see Section 6.2.1). Again, a SVM based novelty detector was built from these variables and the associated pignistic probability of fault was computed. A sensor breakdown was simulated on the first temperature, and the simulated points were tested. Figure 6.7 shows the faulty signal (top) together with the pignistic probability of fault (bottom) at the same time steps. The pignistic probability of fault associated with faulty points is always 1, hence leaving no doubt with respect to the occurrence of a fault.

Sensor decalibration Sensor decalibration corresponds to a sudden drop or increase in the mean of the sensor measurements. Consider again the data of example 22, Section 6.2.1. Successive decalibrations (at time steps 1000 and 2000) were simulated on the sensor that measures the percentage of O_2. Their amplitude was, respectively, of 1% and 0.5% of O_2 (respectively 7 and 15% of the mean value of the percentage of of O_2 for training data). The SVM based novelty detector built in the previous paragraph was used to compute the associated pignistic probability of fault. Figure 6.8 shows the faulty signal (top) and the pignistic probability of fault (bottom) at the same time steps. It can be observed that a decalibration of 1% is correctly detected, but 0.5% is not enough and remains undetected. Figure 6.9 shows the same for the data of example 23, Section 6.2.1, the decalibration being simulated on the first temperature. Here again, a first decalibration of 25˚C (occuring at time step 1000) is not detected, but a decalibration of 50˚C at time step 2000 is detected, and the pignistic probability of fault reaches 0.99. Further decalibration of the sensor leads to a probability of fault of 1.

Another form of visualisation of the results will now be introduced. Suppose two successive 10 degree drops of the steam temperature (about 3 times its standard deviation) happen in a group of 4 steam temperatures. Each observation may be represented by a point in a 4 dimensional space. Figure 6.10 represents test data in the first three temperatures 3-D space. 4112 observations are shown, 1000 of which were recorded before any drop in temperature happened, 1000 after the first drop (and before the second), 1000 additional observations after the second drop, and another 1112 observations after the situation got back to normal. The pignistic probability of fault attached to each point is represented by a color scale, and the color of the point itself. A pignistic probability of fault in the [0.5;0.6] range is represented by a very light grey point, the [0.6;0.7] range yields a slightly darker grey point, etc. Black indicates a near certainty of fault. It can be observed that, for non faulty points (right and bottom-most points), the pignistic probability of fault is low at the core of the data set and increases towards it edges. This is similar to what happens with the training points, and shows that the classifier behaves

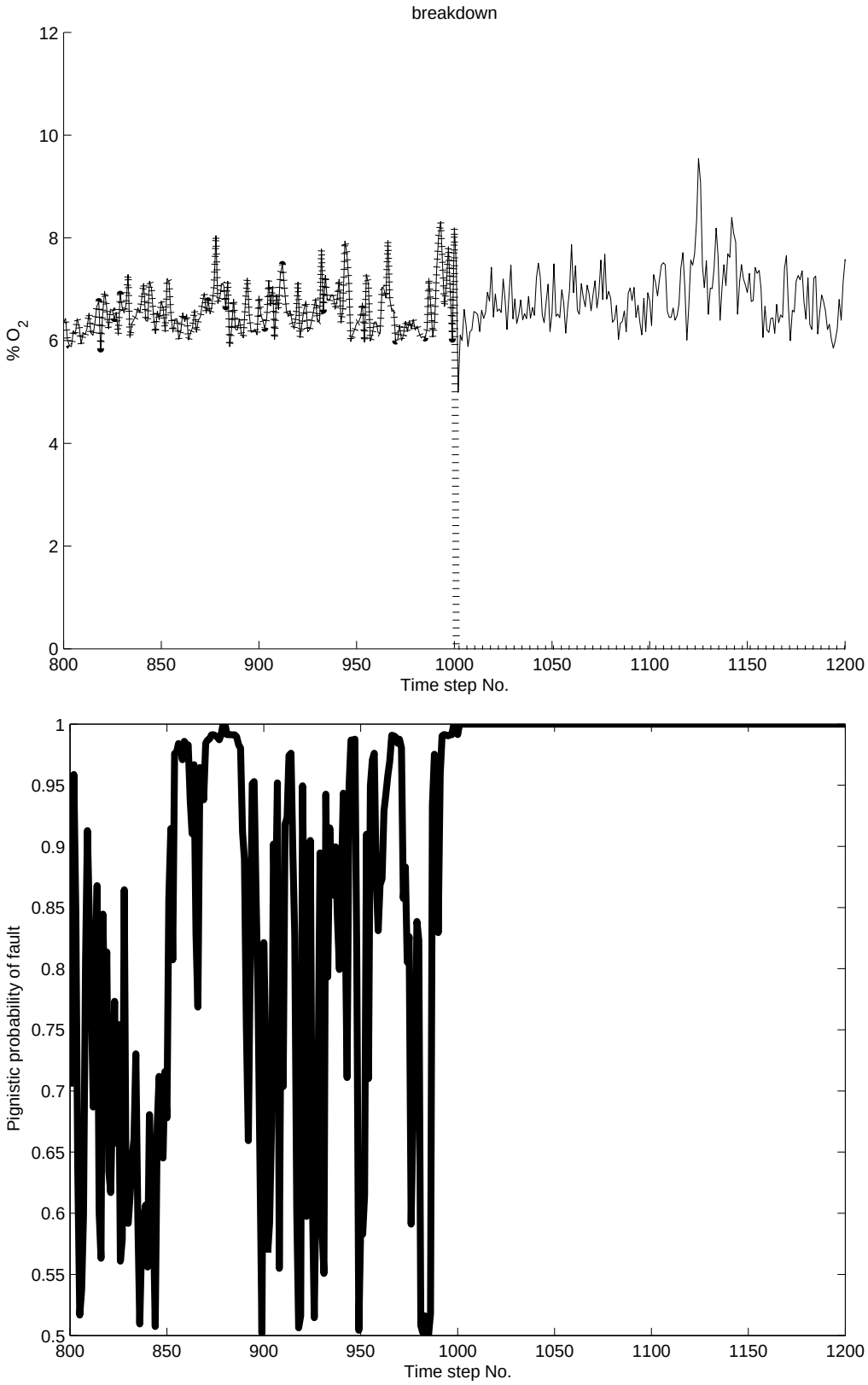


Figure 6.6: Sensor breakdown detection, combustion data (Example 22, Section 6.2.1)

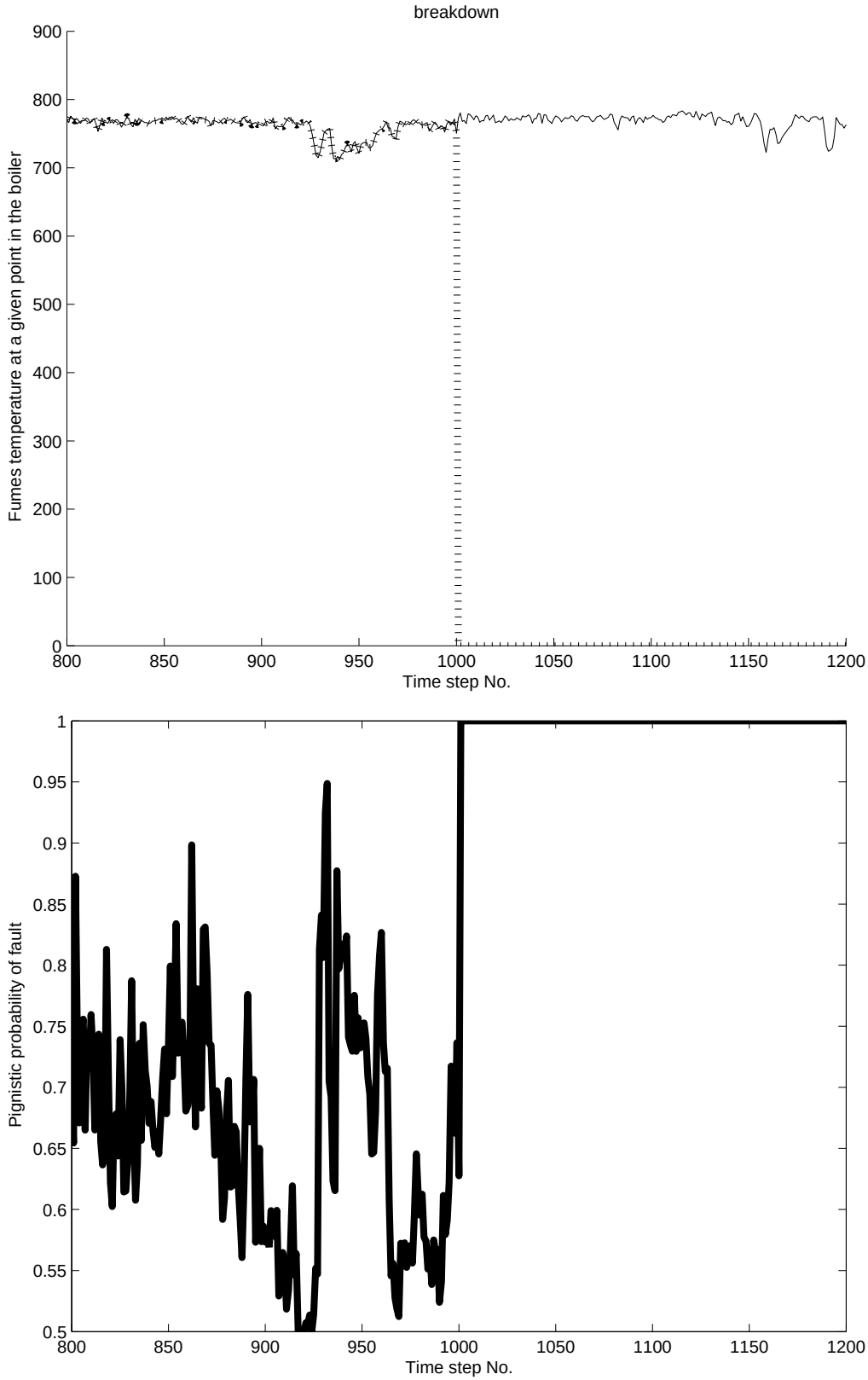


Figure 6.7: Sensor breakdown detection, fumes data (Example 23, Section 6.2.1)

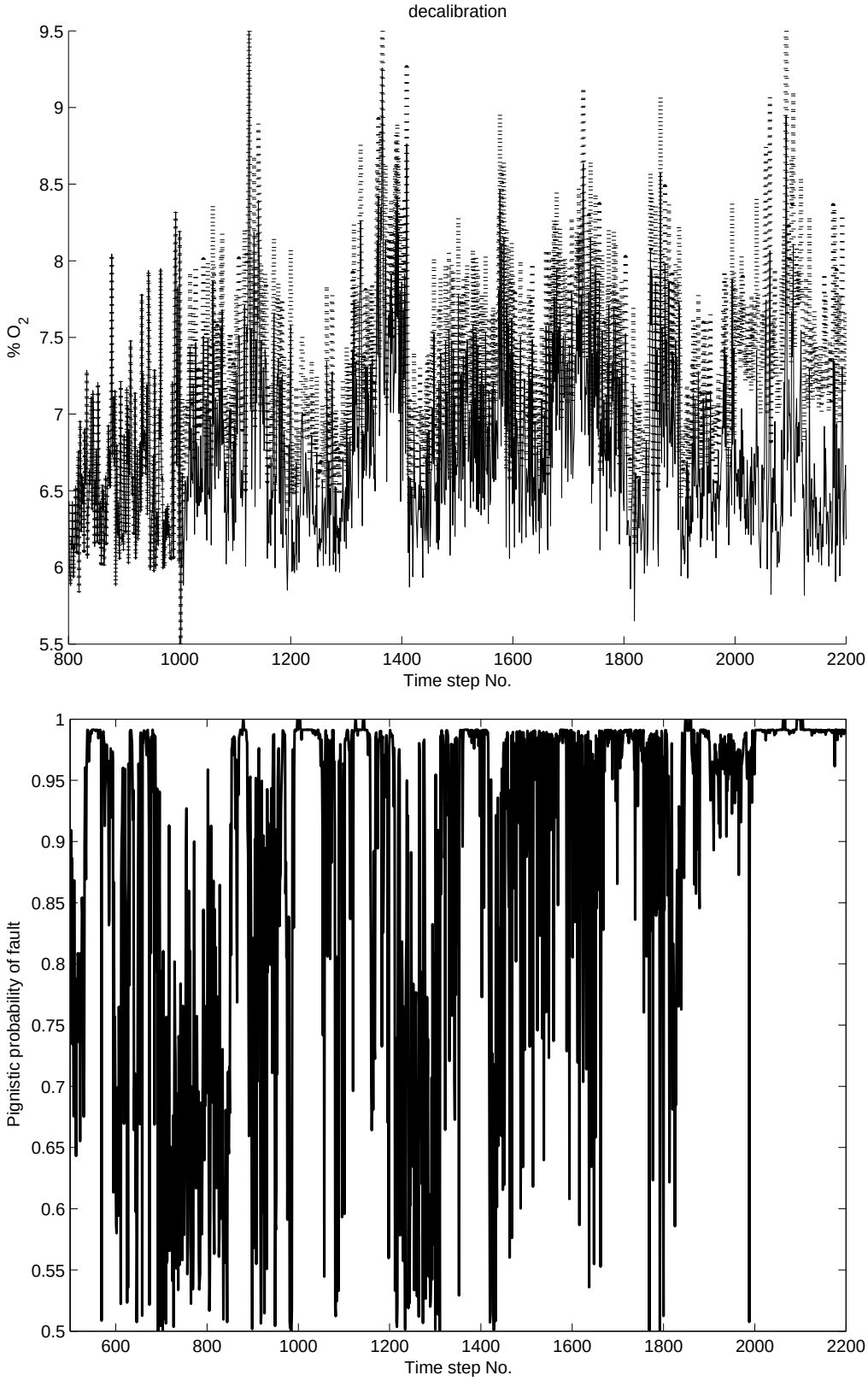


Figure 6.8: Sensor decalibration detection, combustion data (Example 22, Section 6.2.1).

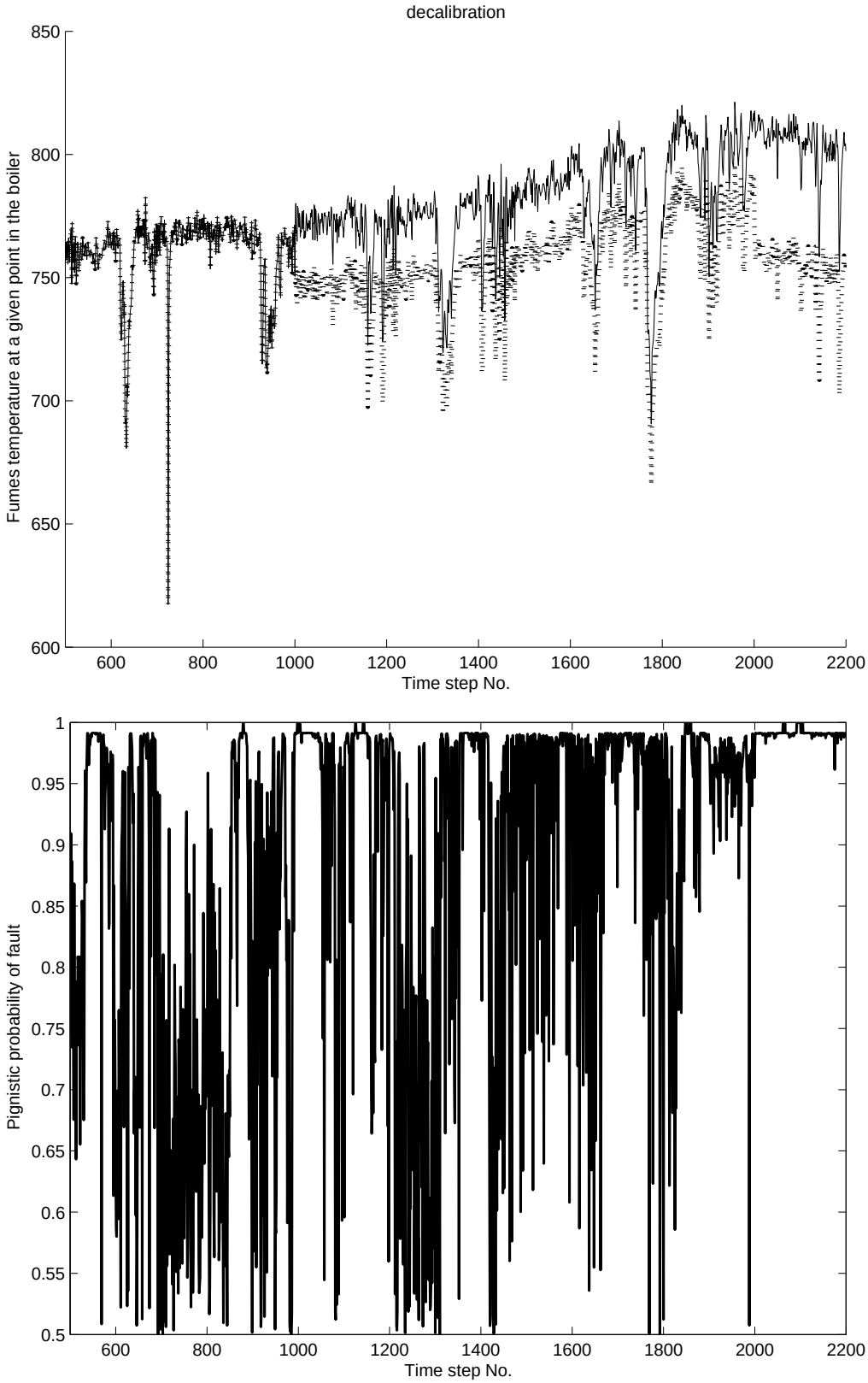


Figure 6.9: Sensor decalibration detection, fumes data (Example 23, Section 6.2.1).

as expected. As for the faulty points (left and front-most two clusters), the associated pignistic probability of fault is always 1. Both drops are instantly detected, and the pignistic probability of fault immediately reaches 1. The return to normal situation is immediately detected as well, and the pignistic probability of fault drops instantly.

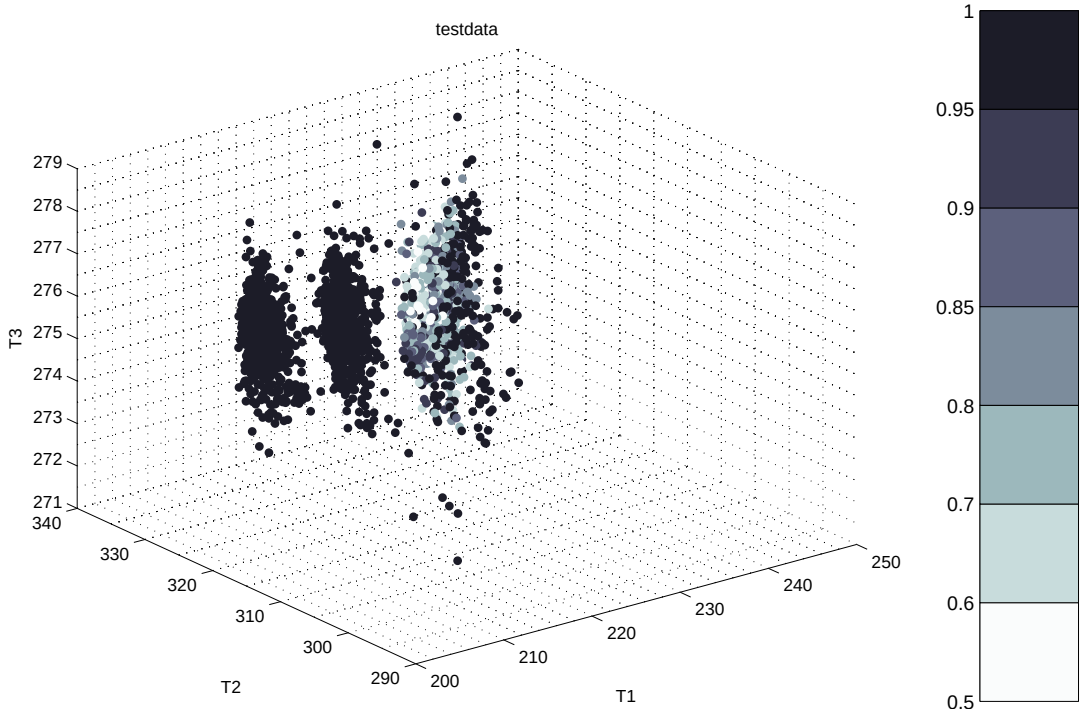


Figure 6.10: Sensor decalibration detection, steam data.
The pignistic probability of fault attached to each point is represented by a color scale, and the color of the point itself. A pignistic probability of fault in the [0.5;0.6] range is represented by a dark red point, the [0.6;0.7] range yields a light red point, etc. Dark blue indicates a near certainty of fault.

Slow drift detection Let us come back again to example 22, Section 6.2.1. A slow drift was simulated on the sensor that measures the percentage of O_2. The drift was about 0.06% O_2 during the first ten hours (starting at time step 1000), and then it was 0.03% O_2 an hour (from time step 2000 onwards). The previously built SVM-based novelty detector was used to compute the associated pignistic probability of fault, and the fault was detected after 90 hours, when it reached 0.3% O_2 (5% of the mean value for training data). Figure 6.11 shows the faulty signal (top) and the pignistic probability of fault (bottom) at the same time steps. It can be observed that the fault is correctly detected. Figure 6.12 shows the same for the data of example 23, Section 6.2.1, the fault being simulated on the first temperature. The simulated fault was a slow drift of the steam temperature at some specific point in the boiler. The drift occurred at a rate of about 0.76 degrees an hour (0.1% of the mean value for training data) for the first ten hours (starting at time step 1000) and then at a rate of 3.8 degrees an hour (from time step 2000 onwards). It was successfully detected when the difference in the mean temperature reached 19 degrees.

Similarly, a slow drift of the steam temperature at some specific point in the boiler was successfully detected by an SVM-based classifier built on four steam temperatures. Figure 6.13 shows the points represented in the first three dimensions.

The best performance was obtained for a drift simulated on a particular steam temperature at a rate of about 2.3 degrees an hour (1% of the mean value for training data), which

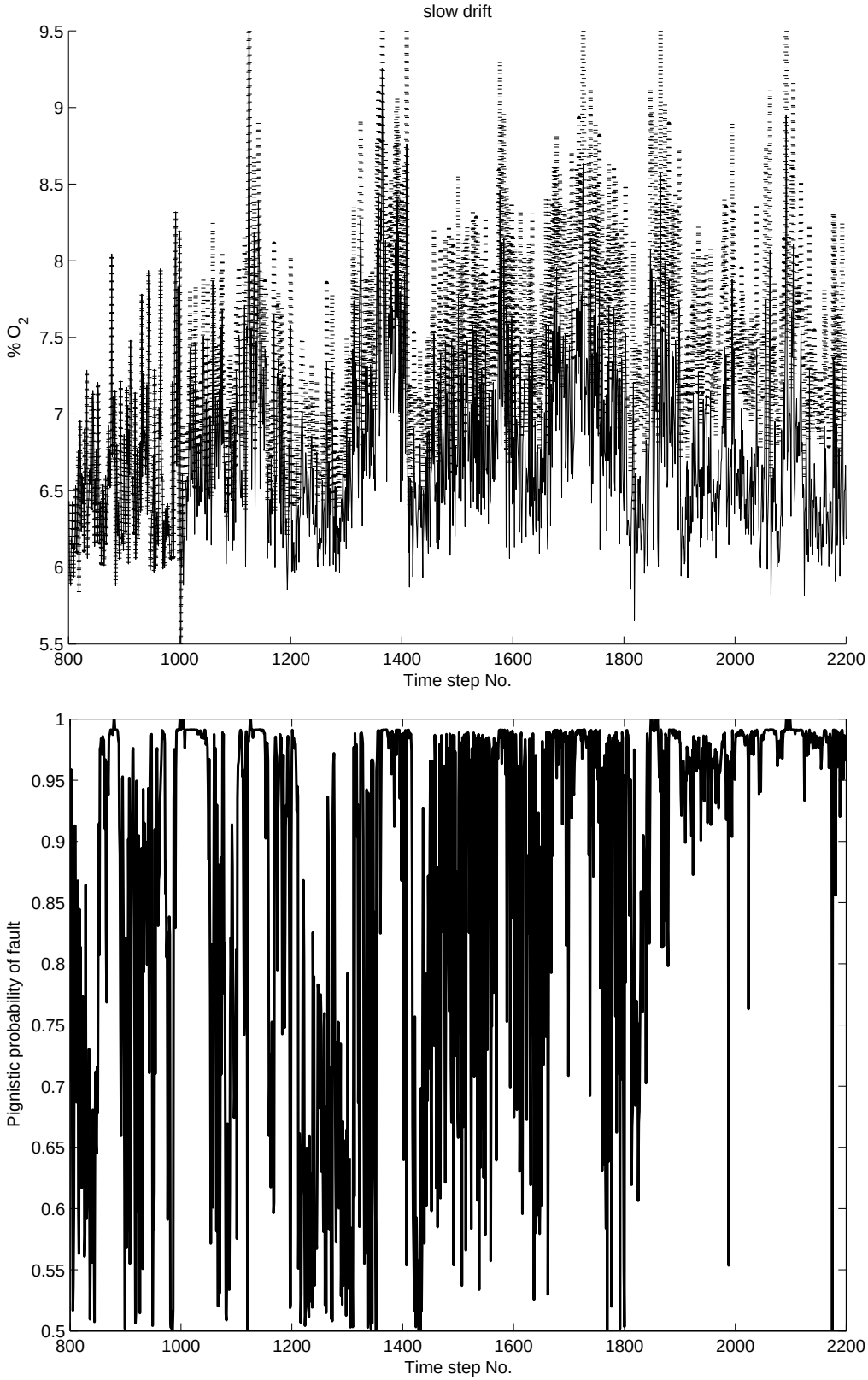


Figure 6.11: Slow drift detection, combustion data (Example 22, Section 6.2.1).

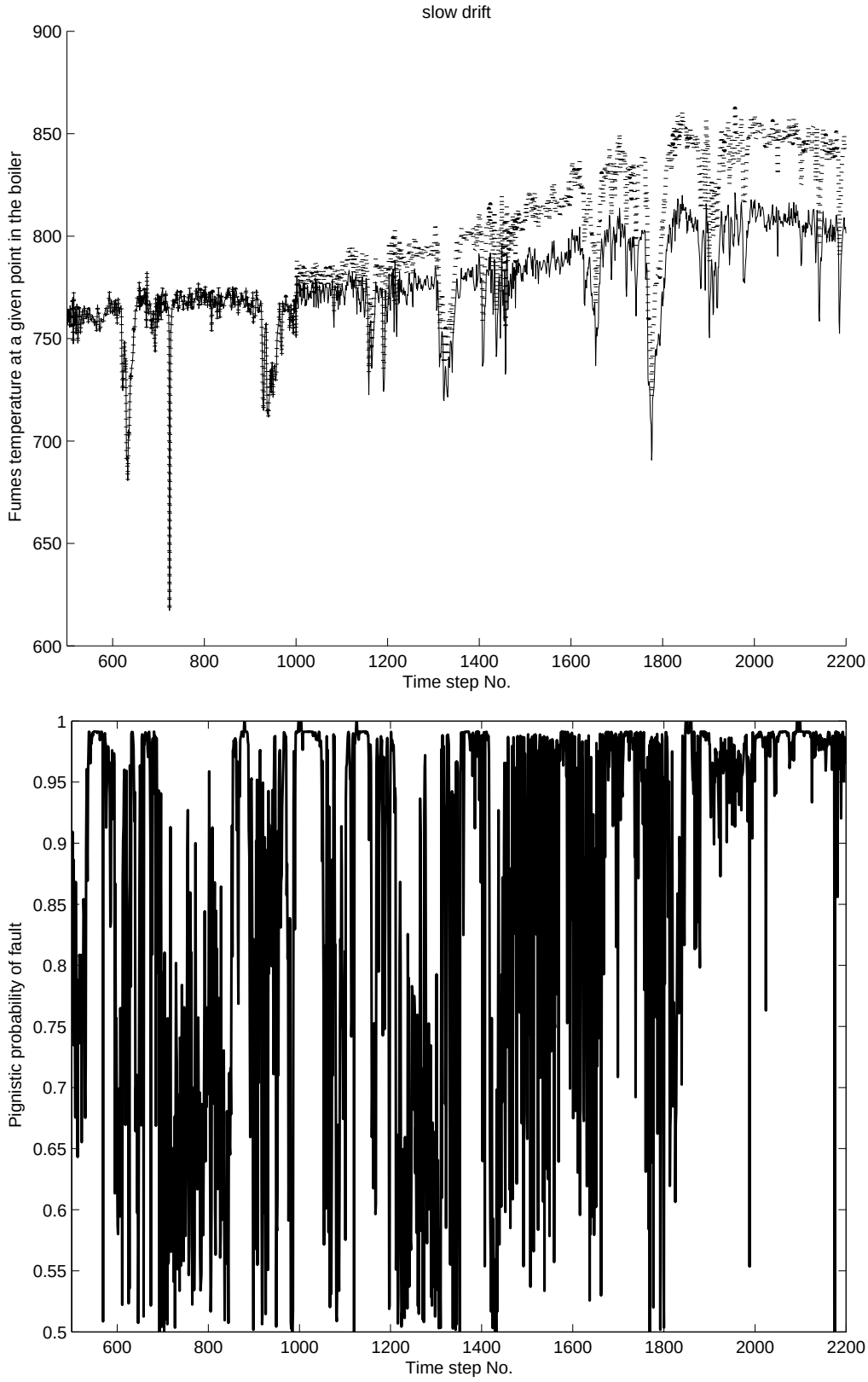


Figure 6.12: Slow drift detection, fumes data (Example 23, Section 6.2.1).

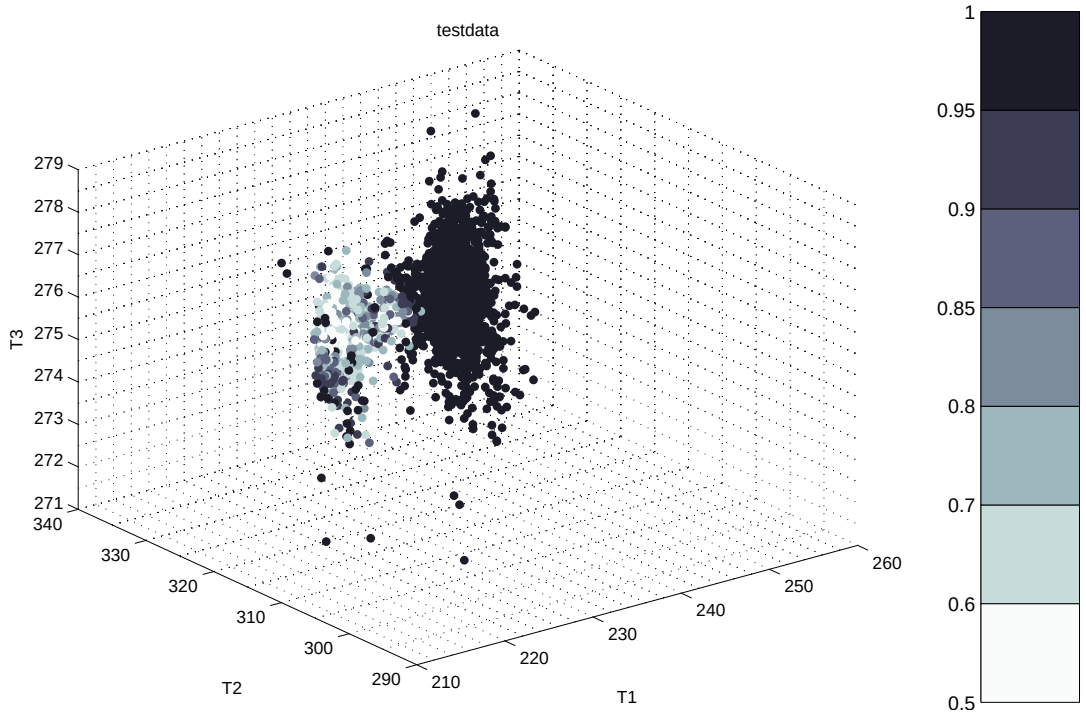


Figure 6.13: Slow drift detection, steam data.
The pignistic probability of fault attached to each point is represented by a color scale, and the color of the point itself. A pignistic probability of fault in the [0.5;0.6] range is represented by a dark red point, the [0.6;0.7] range yields a light red point, etc. Dark blue indicates a near certainty of fault.

was successfully detected by an SVM based classifier built on four steam temperatures. Detection occurred when the difference in the mean temperature reached 10 degrees, which took 5 hours. According to the experts, this result is highly satisfactory in a process monitoring point of view, and would never have been achieved manually.

Detection of real faults Later, real drifts appeared in the process and were successfully detected by the PMAT system.

A difference of 40 degrees of gas combustion temperature in the boiler due to a sensor deviation was successfully detected within an hour. Though this measurement is vital for the process and therefore regularly overseen, such a difference is never detected by the human eye before it reaches 100 degrees, which might take several days. In this particular case, the operators –who did not have access to the PMAT yet– noticed the fault only one week after the PMAT indicator showed it, and the sensor was changed thereafter. This temperature being a key element of the process regulation, and used in many regulation loops, this result of the PMAT was considered highly satisfactory. Figure 6.14 is a screen shot that shows the evolution of the temperature and that of the pignistic probability of fault calculated from a group of variables representative of the process quality (steam flow, steam temperature at boiler exit, upper chamber temperature (sensor No. 2)). Note that the calculation of the pignistic probability of fault from this group of variables only takes the upper chamber temperature No. 2 into account (cf Table 6.1, PMAT unit No. 8). Another classifier was built with the same variables but using sensor No. 1 instead of sensor No. 2. In this way, the fault was identified: it was not the upper chamber temperature sensor No.1 that had an increasing drift nor the process that was going wrong (in this case, both classifiers would have detected a fault) but the upper

temperature chamber No.2 that had a decreasing drift and should be either recalibrated or replaced.

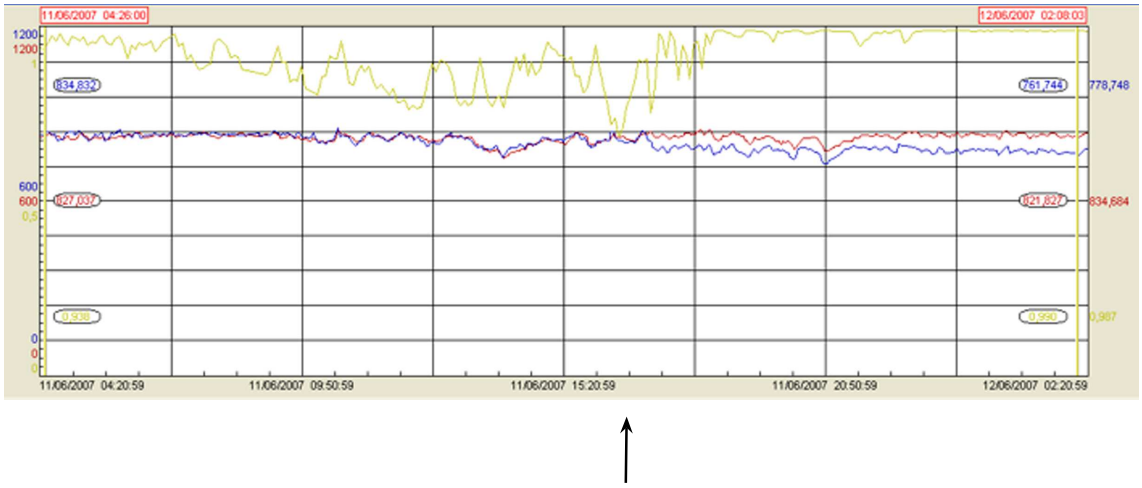


Figure 6.14: Detection of a drift of the upper chamber sensor No.2.
The top most line is the pignistic probability of fault, and the gas combustion temperature as measured by sensor 1 (working correctly) and sensor 2 (faulty). The fault starts at the time pointed by the arrow below the figure.

Another real drift, consisting in a 10 degrees drop in the flue gas temperature at the exit of the boiler (undetermined cause), was also successfully detected in less than 30 minutes. Figure 6.15 shows the evolution of the temperature and that of the pignistic probability of fault calculated from a group of variables representative of the heat exchange quality in the boiler (cooling water flow, steam flow, desuperheating temperature, steam temperature at boiler exit, upper chamber temperature No. 1, upper chamber temperature No. 2).

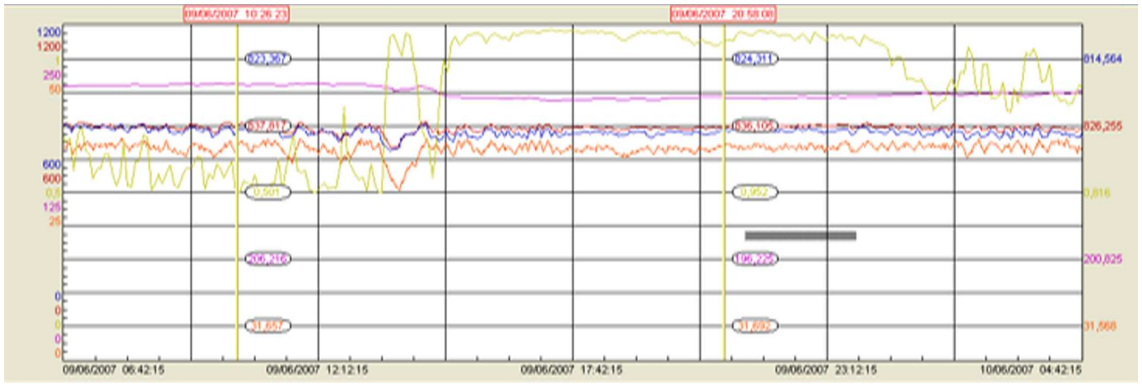


Figure 6.15: Detection of a 10 degree drop of the flue gas temperature.
The figure shows the pignistic probability of fault (beige, top most) line, the faulty temperature (pink, second line from top), the upper chamber temperature as measured by sensor 1 and sensor 2 (third and fourth lines from top). The fault starts at the time pointed by the arrow below the figure.

As a third example, it may be mentioned that a drop in the output steam flow due to

a bad distribution of the waste on the oven's conveyor belt (presence of heaps) was again successfully detected. Unfortunately, a screen shot is not available for this third example of real fault. The behaviour of the PMAT when this fault occurred may be described as follows. First peaks appeared in the pignistic probability of fault 45 minutes before the drop was detected by the operators, and the pignistic probability of fault reached its maximum value (of 0.99) 15 minutes before the drop was detected by the operators. The probability of fault then started decreasing nearly three hours later, 10 minutes before the situation went back to normal.

Interpretability of the indicators: The important variance of the pignistic probability of fault over time (when the process was in a non faulty state) made its interpretation difficult for the operators. This phenomenon may be observed in all the above figures. It was difficult for people outside the statistician community to understand the fact that this value varied as much as the measures it was built from, but only a threshold overrun would really imply that the process was in a faulty state. Moreover, this important variance lead to a number of false alarms that could be avoided. It was thus suggested to carry out a CUSUM test on the mean of the pignistic probability of fault to smoothen the results [14]. It is the CUSUM statistic that was finally used for graphical representation in the PMAT, to the satisfaction of the operators.

Two other solutions were suggested for this problem:

- Modification of the sampling selection period: as treating a point every 5 seconds would be too computationally demanding and would not be relevant (the process variations being much slower) the PMAT does not works with direct measurements but with mean values of the measurements, calculated over a 6 minutes period. This period was chosen in collaboration with process experts in such a way that the process variations would not be hidden. Sampling periods of 10, 20 and 30 minutes where tested as well, but the resulting pignistic probability of fault was not smoothened sufficiently until 20 and 30 minutes periods were used, which also hid the main process variations and therefore was not interesting in terms of process monitoring.

- A (possibly weighted) combination of the belief function obtained at each time step with that obtained at previous time step would smoothen the results and allow the direct use of the pignistic probability of fault as an indicator (instead of a CUSUM of this probability), making the interpretation easier. This idea is based on the fact that a fault rarely appears for a very short period of time. More precisely, a fault is less plausible if there was no fault at the previous time step. This solution has not be implemented in the PMAT yet, but has already been tested off-line. Figure 6.16 shows the result of different types of combinations or other means of smoothing the pignistic probability of fault over time. The upper subplot shows:

  1. In continuous line, the orignal signal, i.e., the pignistic probability of fault as obtained from the belief function on the state of the system calculated at each time step using the methodology presented in Section 4.2 and 4.6.2.;
  2. In dash-dotted line, the signal obtained by averaging the weights corresponding to the last ten obtained bba, and then computing $BetP(\omega_1)$;
  3. In dotted line, the signal obtained by performing a weighted conjunctive combination of the last ten obtained bba, i.e. by combining the last bba with the ten previous ones, each being discounted by a factor $1 - \exp(-\gamma\Delta(t_i))$, where $\Delta(t_i)$ is the time difference (in number of time steps) between the calculation of the last bba and the i^{th} before the last [125].

The two above solutions for smoothing the pignistic probability curve show promising results. However, further investigation would be necessary to compare them and to find the best time frame over which to perform the averaging or the weighted combination. The lower subplot of Figure 6.16 shows the result (alarm time) of a CUSUM test on the mean of the original signal. This solution is the one that is currently implemented in the PMAT.

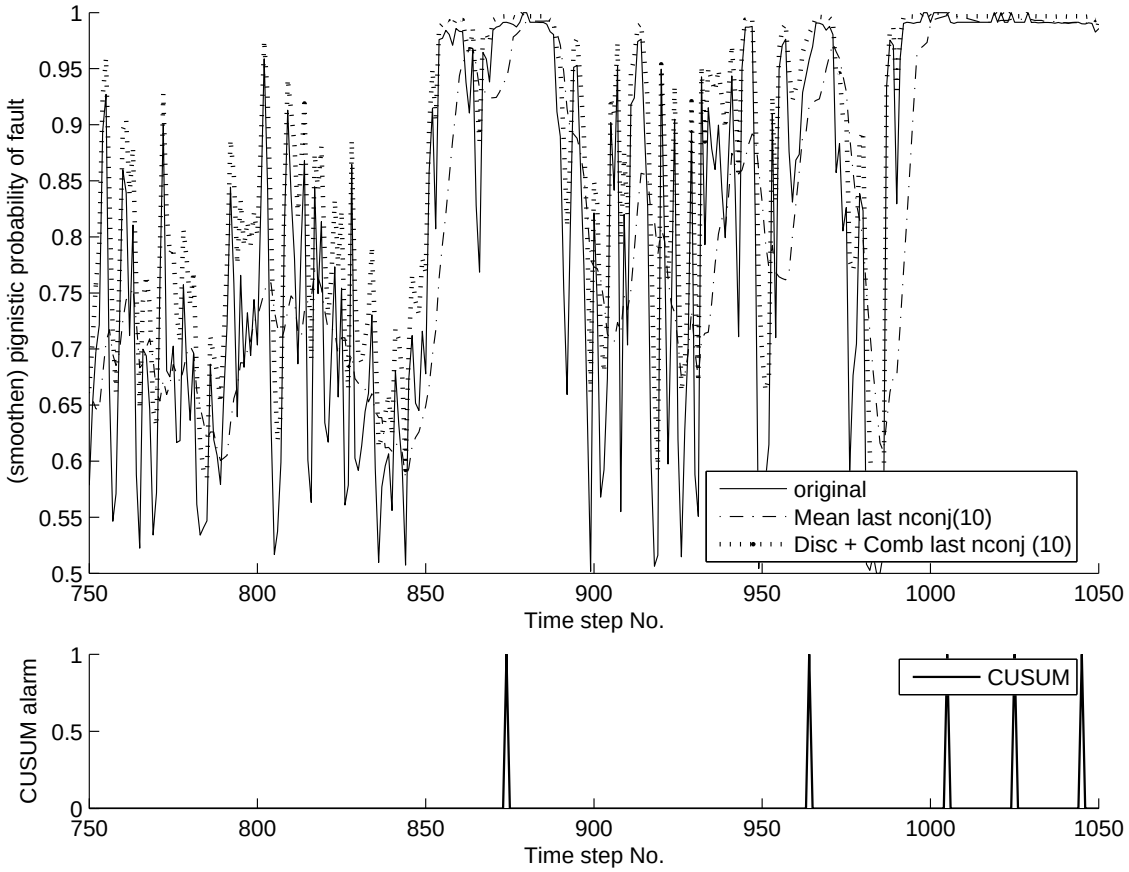


Figure 6.16: Attempts in smoothing the pignistic probability of fault

Discussion

PMAT versus human detection: A sensor breakdown is obvious, and is instantly detected by the human eye if the sensor measurement is represented over time. However, outside the PMAT, it is not the case for most of the variables to be monitored. Hence, a few minutes of observation is necessary to detect a sensor breakdown, going up to half and hour for measures with low variance and slow evolution. The operators very rarely have time to observe the same variable for several minutes, not to mention do a regular check of each variable. The PMAT is thus useful in this case to compensate for the lack of human time. A very simple CUSUM test is most of the time sufficient to monitor a single sensor, but a more complex (SVM or KPCA-based) novelty detector allows the monitoring of several sensors at the same time, and the checking of both process and sensor faults simultaneously. Fault location may be allowed by cross-checking several detectors or by human expertise once the occurrence of a fault has been outlined by the PMAT.

Sensor decalibration is more difficult to detect for the human eye. It is impossible to detect without a time scale, and extremely difficult to detect even with a time scale if it is quite small. Moreover, it again requires the operators to have a look at the screen before the decalibration disappears from the represented time scale. The PMAT will thus be of great use to improve the quality and earliness of maintenance interventions.

Finally, slow drifts are difficult and sometimes impossible to detect for the human eye. In any case, they can only be detected manually when they get quite important, which is often too late to prevent damaging the installation, decrease pollution rates or avoid loosing money. The main usefulness of the PMAT thus lies in fault detection time shortening, especially for slow drifts.

Labelling: Several relabellings of the data were necessary to obtain reasonable values of the type I and type II errors on the test set. At first, labels were often too strict, leading to over-training and bad generalization capacity of the classifier. This resulted in high values of the pignistic probability of fault. Labelling was improved until the pignistic probability of fault took values which varied over its full range, from 0.5 to 1[7].

Interpretability: It was greatly improved by the implementation of a CUSUM test on the mean of the pignistic probability of fault. However, it can, and will, be made even better by the implementation of a weighted combination. The operators will be consulted on the choice of this combination.

Quality of the results: All simulated faults were successfully detected. This result can perhaps be nuanced with the fact that engineers simulated faults that could not be detected by the human eye but which were maybe still quite important on a statistical scale, i.e., the deviation represented several times the standard deviation of the monitored variable. However, the engineers considered these deviations as realistic enough. The waste incineration process being quite an unstable one, –due to the non homogeneity of the wastes leading to an unsteady calorific value of the material to be burnt–, normal variations of the process are quite important and cannot allow the detection of much smaller deviations than those simulated without generating an important number of so-called false alarms. Additionally, such small deviations are not of real importance for the process and cannot be avoided.

As shown by the parameter tuning results, a rate of 30% undetected faults is to be expected. Some process deviation indeed did not lead to any variation in the pignistic probability of fault but were detected by the human eye when analyzing the process data a posteriori. Then again, these faults were not detected by the operators at the time of occurrence. In spite of these missed alarms, and according to the operators, the PMAT proves useful on an every day basis by detecting many faults early or detecting faults which would have otherwise remained unnoticed.

Results for the KPCA-based classifier

The KPCA-based algorithm was first tested on the case study examples of paragraph 6.2.1. As already mentioned, classification through the KPCA-based algorithm was slow because the calculation of the novelty detection statistic implies all training points. The calculation could only be set on-line every minute. However, the pignistic probability of fault calculated from this statistic had much smaller variance than that calculated from the SVM-based algorithm, thus making it much easier to interpret for the operators. The

[7] The pignistic probability of fault cannot drop below 0.5 as no prior class probability was introduced for classes ω_0 and ω_1, and there is always some evidence in favour of ω_1.

KPCA and SVM-based algorithms nevertheless allowed the detection of the same process faults over the observation window, and a graphical representation of their means over a short period of time would have shown fairly parallel curves.

The KPCA-based solution was thus momentarily abandoned with a strong recommendation that further research be done to quicken the computations. It was not tested on other cases.

6.4 Conclusion

The prototype has been validated, though there is a lot of additional work to be done to improve the implementation of the KPCA-based algorithm and the user interface. Once the KPCA-based algorithm will be fully functional, a combination of the SVM and KPCA-based algorithm should be tested to check if it improves the results.

Conclusion and Perspectives

In this report, we addressed the problem of monitoring a waste incineration process, which may be brought down to a one-class classification problem. The particularity of such an application is the fact that it involves numerous, but unreliable data, with particularly complex correlations. This makes it all the more complex, which explains why no solution had yet been implemented. The ever growing competition in the waste promotion industry, together with the drastic hardening of anti-pollution standards, lead Suez Environnement to make a new attempt to solve the issue via a collaboration with Compiègne University. Our approach to the problem is based on belief function theory, and, more precisely, on the Transferable Belief Model, a framework chosen for its ability to handle imprecision and uncertainty.

First, we introduced new methods for the construction of belief functions from raw data. We showed how to approximate the distribution of continuous variables with discrete belief functions, and, whenever possible, how to obtain more precise results with continuous belief functions. We established two different ways of tackling the problem. The first approach is based on Hacking's principle, and on previous work of Denœux reported in [41]. The second approach, which can be argued to be more in line with the two-level (credal, pignistic) structure of the TBM, is based on the notion of pignistic probability distribution.

These schemes proved to carry the imprecision of the data in an interesting way, and lead to much reduced variability in final classification decisions, especially in comparison with techniques that do not take the size of the training sample into account. However, it is not yet quite clear which of these methods is best appropriate. In effect, the first method presents the advantage of allowing an entirely non-parametric modeling of the problem, while the second technique does not. On the other hand, this last technique seems to find better justifications in the ground of the TBM.

Then, introducing the notion of cognitive inequality, we demonstrated how these techniques can be used together with one-class classifiers to detect novelty in a reliable way, while keeping track of the uncertainty attached to the decision. More generally, we explained how these methodologies could be used to combine the output of different classifiers so as to obtain better performances. Then again, we suggested three possible models and it is not clear which is best. As already mentioned in Chapter 4, the use of the cognitive inequality is an obvious improvement to the simple GBT solution, as it allows the handling of additional, qualitative information. However, the latter model remains interesting in cases where no hypothesis can be made on the monotonicity of statistic T. On the other hand, it is more difficult to compare the relative quality of Models 1 and 2. Model 1 is simpler, but leads to a more complex solution than Model 2. The latter may be more difficult to justify, but it leads to a very simple solution, and was experimentally validated on some examples.

Using the tools developed in this thesis, other aspects of novelty detection and information fusion could be explored. First, the technique has mainly been applied to SVM and KPCA-based algorithms. It would be interesting to try and compare the performances of as many one-class classifiers as possible, now that they can all be expressed in the same framework. Furthermore, the conditions in which the combination of classifiers will bring an improvement in the overall classification performances could be studied. In other words, we could try and characterize whether it is worth combining classifiers based on different features, different data sets, different classes, different numbers of classes, different algorithms, etc. The form of combination best appropriate to each case

would need to be specified.

Finally, our theoretical contributions were tested on a full-scale prototype in a waste incineration plant and proved to yield satisfying results. However, some improvements could be brought to the system, and additional work remains to be done.

To begin with, the computational performances of the prototype could be greatly improved by a complete recoding, paying attention to computational efficiency, memory requirement, etc, and trying to minimize all storage and calculation power costs. Algorithms based on KPCA could then be tested more exhaustively. Moreover, there is a lot to gain in trying to find clear graphical representation, which would make belief functions easily interpretable to the neophyte, and would help the expert make decisions quickly. Then, in view of the results, it would be interesting to further explore the problem of smoothing the obtained pignistic probability of fault. As we have seen, smoothing indeed allows avoiding a number of false alarms. It can be done, e.g., by combining or averaging the belief functions on the state of the system obtained from times $i - t$ to i. Tests could thus be carried out to find the best time frame, ensuring sufficient smoothing while retaining enough information so that small faults would not be hidden. Finally, the prototype of the process monitoring assistance tool should be extended to parts of the plant other than the oven and boiler units, before the final version of the tool can be developed. It will then be possible to test it on other types of thermodynamic systems, or other industrial processes altogether.

As a conclusion, there are a number of possible improvements and perspectives to the work accomplished during this three-year project, both on the theoretical and on the application sides. Nevertheless, what has already been achieved shows promising results. Suez Environnement made the decision to go on with the project, and an engineer has recently been recruited for the next phase. Meanwhile, the PMAT is used as it is on an everyday basis, and the staff makes good use of the information it provides.

Bibliography

[1] J. Ahola, E. Alhoniemi, and O. Simula. Monitoring industrial processes using the self-organizing map. In *SMCia/99 Proceedings of the 1999 IEEE Midnight-Sun Workshop on Soft Computing Methods in Industrial Applications*, pages 22–7, Piscataway, NJ, 1999. IEEE Service Center.

[2] R. G. Almond. *Graphical belief modeling*. Chapman and Hall, London, 1995.

[3] D. A. Alvarez. On the calculation of the bounds of probability of events using infinite random sets. *International Journal of Approximate Reasoning*, 43(3):241–267, 2006.

[4] M. Andersson and M. Lomakka. Evaluating implied rnds by some new confidence interval estimation techniques. *Journal of Banking and Finance*, 29:1535–1557, 2005.

[5] A. Aregui and T. Denoeux. Novelty detection in the belief functions framework. In *Proceedings of the 11th Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, volume 1, pages 412–419, Paris, July 2006.

[6] A. Aregui and T. Denoeux. Constructing predictive belief functions from continuous sample data using confidence bands. In *Proceedings of the International Symposium On Imprecise Probability: Theories And Applications*, pages 11–20, Prague, Tcheck Republic, July 2007.

[7] A. Aregui and T. Denoeux. From sample data to belief functions via pignistic probabilities. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (accepted)*, Hammamet, Tunisia, October 2007.

[8] A. Aregui and T. Denoeux. Fusion of one-class classifiers in the belief function framework. In *Proceedings of the International Conference on Information Fusion, Québec, Canada*, Québec, Canada, July 2007. Best student paper award.

[9] B.C. Arnold and R.M. Shavelle. Joint confidence sets for the mean and variance of a normal distribution. *The American Statistician*, 52:132–139, 1998.

[10] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.

[11] P. Baldi and K. Hornik. Learning in linear neural networks: A survey. *IEEE Transactions on Neural Networks*, 6(4):837–858, 1995.

[12] A. A. Balkema and L. de Haan. Residual lifetime at great age. *Annals of Probability*, 2:792–804, 1974.

[13] V. Barnett and T. Lewis. *Outliers in statistical data*. John Wiley and Sons, New York, 1994.

[14] Michèle BASSEVILLE and Igor NIKIFOROV. *Detection of abrupt changes : theory and application*. Prentice-Hall, 1993. *Previously published by Prentice-Hall, Inc. (April 1993 - Englewood Cliffs, N.J.), may now (November 1998) be downloaded, using pdf or (compressed postcript) ps.gz files, at the following url : http://www.irisa.fr/sisthem/michele/publis.html.*

[15] Kristin P. Benett and Colin Campbell. Support vector machines: hype or hallelujah? *SIGKDD exploration*, 2:1–13, December 2000.

[16] J.C. Bezdek, R. Erlich, and W. Full. Fcm: the fuzzy c-means clustering algorithm. *Computer and geosciences*, 10:191–203, 1984.

[17] A. Bordes and L. Bottou. The huller: a simple and efficient on-line svm. *Machine learning: ECML, Lecture notes in artificial intelligence, LNAI 3720*, pages 505–512, 2005.

[18] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59:291–294, 1988.

[19] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. Svm and kernel methods matlab toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2005.

[20] F. Caron, B. Ristic, E. Duflos, and P. Vanheeghe. Least committed basic belief density induced by a multivariate gaussian: Formulation with applications. *International Journal of Approximate Reasoning, In Press, Corrected Proof (doi: 10.1016/j.ijar.2006.10.003)*, 2007.

[21] G.A. Carpenter and S. Grossberg. Art 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26:4919–4930, 1987.

[22] G.A. Carpenter, S. Grossberg, and D.B. Rosen. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[23] R. C. H. Cheng and T. C. Iles. Confidence bands for cumulative distribution functions of continuous random variables. *Technometrics*, 25(1):77–86, 1983.

[24] A. Cuevas, M. Febrero, and R. Fraiman. On the use of bootstrap for estimating functions with functional data. *Computational statistics and data analysis*, 51:1063–1074, 2005.

[25] M. Davy, F. Desobry, A. Gretton, and C. Doncarli. An online support vector machine for abnormal events detection. *Signal Processing*, 86(8):2009–2025, 2006.

[26] L. M. de Campos, J. F. Huete, and S. Moral. Probability intervals: a tool for uncertain reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2(2):167–196, 1994.

[27] M.H. DeGroot. *Optimal statistical decisions*. McGraw-Hill, New York, 1970.

[28] F. Delmotte and Ph. Smets. Target identification based on the Transferable Belief Model interpretation of Dempster-Shafer model. *IEEE Transactions on Systems, Man and Cybernetics A*, 34(4):457–471, 2004.

[29] A. Dempster. *Construction and local computation aspects of network belief functions*. John Wiley and Sons, Chichester, 1990.

[30] A. P. Dempster. New methods of reasoning towards posterior distributions based on sample data. *Annals of Mathematical Statistics*, 37:355–374, 1966.

[31] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.

[32] A. P. Dempster. Upper and lower probability inferences based on a sample from a finite univariate population. *Biometrika*, 57:515–528, 1967.

[33] A. P. Dempster. A generalization of Bayesian inference (with discussion). *J. R. Statistical Society B*, 30:205–247, 1968.

[34] A. P. Dempster. Upper and lower probabilities generated by a random closed interval. *Annals of Mathematical Statistics*, 39(3):957–966, 1968.

[35] A. P. Dempster. A class of random convex polytopes. *Annals of Mathematical Statistics*, 43(1):260–272, 1972.

[36] T. Denoeux. Application of evidence theory to k-NN pattern classification. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice IV*, pages 13–24. Elsevier, Amsterdam, 1994.

[37] T. Denoeux. An evidence-theoretic neural network classifier. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 712–717, Vancouver, October 1995.

[38] T. Denoeux. Application du modèle des croyances transférables en reconnaissance de formes. *Traitement du Signal*, 14(5):443–451, 1998.

[39] T. Denœux. A neural network classifier based on Dempster-Shafer theory. *IEEE Trans. on Systems, Man and Cybernetics A*, 30(2):131–150, 2000.

[40] T. Denoeux. The cautious rule of combination for belief functions and some extensions. In *Proceedings of the 9th International Conference on Information Fusion*, Florence (Italy), July 2006. Paper #114.

[41] T. Denoeux. Constructing belief functions from sample data using multinomial confidence regions. *International Journal of Approximate Reasoning*, 42(3):228–252, 2006.

[42] T. Denoeux. Construction of predictive belief functions using a frequentist approach. In *Proceedings of the 11th Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, July 2006.

[43] T. Denoeux and P. Smets. Classification using belief functions: the relationship between the case-based and model-based approaches. *IEEE Transactions on Systems, Man and Cybernetics B*, 36(6):1395–1406, 2006.

[44] D. Dubois and H. Prade. A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. *International Journal of General Systems*, 12(3):193–226, 1986.

[45] D. Dubois, H. Prade, and P. Smets. A definition of subjective possibility. *International journal of approximate reasoning*, xx:xx, 2007. In press, corrected proof available online 30 March 2007 on `http://www.sciencedirect.com/`, doi:10.1016/j.ijar.2007.01.005.

[46] D. Dubois, H. Prade, and Ph. Smets. New semantics for quantitative possibility theory. In S. benferhat and Ph. Besnard, editors, *Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 2001)*, pages 410–421, Toulouse, France, 2001. Springer-Verlag.

[47] E. J. Dudewicz and S. N Mishra. *Modern Mathematical Statistics*. Wiley, New York, 1988.

[48] S. Fabre, A. Appriou, and X. Briottet. Presentation and description of two classification methods using data fusion based on sensor management. *Information Fusion*, 2:49–71, 2001.

[49] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. Technical report, HP Laboratories, Palo Alto, USA, 2004.

[50] S. Ferson, V. Kreinovitch, L. Ginzburg, D. S. Myers, and K. Sentz. Constructing probability boxes and Dempster-Shafer structures. Technical Report SAND2002-4015, Sandia National Laboratories, Albuquerque, NM, 2003.

[51] R.A. Fisher and Tipettn L.H.C. Limiting form of the frequency distribution of the largest and smallest memer of a sample. *Proceedings Cam. Philo. SOc.*, 24:180–190, 1928.

[52] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1990.

[53] L. A. Goodman. On simultaneous confidence intervals for multinomial proportions. *Technometrics*, 7(2):247–254, 1965.

[54] S. Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11:23–63, 1987.

[55] I. Hacking. *Logic of Statistical Inference*. Cambridge University Press, Cambridge, 1965.

[56] E. Hüllermeier. Similarity-based inference as evidential reasoning. *International Journal of Approximate Reasoning*, 26:67–100, 2001.

[57] V.J. Hodge and J. Austin. A survey of outliers detection methodologies. *Artificial Intelligence Review*, 22:85–126, 2004.

[58] Heiko Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40:863–874, March 2007.

[59] Shuen-Lin Jeng and William Q. Meeker. Parametric simultaneous confidence bands for cumulative distributions from censored data. *Technometrics*, 43:450–611, 2000.

[60] P. Kanofsky. Parametric confidence bands on cumulative distribution functions. *Sankhya, the indian journal of statistics*, 1967.

[61] P. Kanofsky and R. Srinivasan. An approach to the construction of parametric confidence bands on cumulative distribution functions. *Biometrica*, 59(3):623–631, 1972.

[62] P. Kanovsky and R. Srinivasan. An approach to the construction of parametric confidence bands on cumulative distribution functions. *biometrika*, 59:623–631, 1972.

[63] M. Kendall and A. Stuart. *The advanced theory of statistics*, volume 2. Charles Griffin and Co Ltd, London, fourth edition, 1979.

[64] G.J. Klir and M.J. Wierman. *Uncertainty-Based Information. Elements of Generalizes Information Theory*. Springer-Verlag, New-York, 1998.

[65] J. Kohlas and P. A. Monney. Representation of evidence by hints. In R. R. Yager, J. Kacprzyk, and M. Fedrizzi, editors, *Advances in the Dempster-Shafer Theory of Evidence*, pages 473–492. John Wiley, New York, 1994.

[66] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[67] A. N. Kolmogorov. Sulla determinazione empirica di una legge di distribuzione. *Istituto Italiano degli Attuari*, 4:83–91, 1933.

[68] E. Kriegler and H. Held. Utilizing belief functions for the estimation of future climate change. *International Journal of Approximate Reasoning*, 39:185–209, 2004.

[69] H. Lee. A convex hull peeling depth approach to nonparametric massive multivariate data analysis with applications. Department of statistics, the Pennsylvania state university.

[70] H. Lee. *Two Topics: A Jackknife Maximum Likelihood Approach to Statistical Model Selection and a Convex Hull Peeling Depth Approach to Nonparametric Massive Multivariate Data Analysis with Applications*. PhD thesis, The Pennsylvania State University, 2006.

[71] Jong-Min Lee, In-Beum Lee, et al. Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 59:223–234, 2004.

[72] E. L. Lehman. *Testing statistical hypotheses*. Springer-Verlag, New-York, 2nd edition, 1986.

[73] M.J. Lenhoff et al. Bootstrap prediction and confidence bands: a superior statistical method for analysis of gait data. *Gait and posture*, 9:10–17, 1999.

[74] R.Y Liu, J.M. Parelius, and K. Singh. multivariate analysis by data depth: descriptive statistics, graphics and inference. *The annals of statistics*, 27:783–858, 1999.

[75] G. Loosli, S. Canu, S.V.N. Vishwanathan, Alexander J. Smola, and Monojit Chattopadhyay. Une boîte à outils rapide et simple pour les svm. *CAp 2004 - Conférence d'Apprentissage*, pages 113–128, 2004.

[76] G. Loosli, S. Canu, S.V.N. Vishwanathan, Alexander J. Smola, and Monojit Chattopadhyay. Boîte à outils svm simple et rapide. *RIA - Revue d'intelligence artificielle*, 19, 2005.

[77] Markos Markou and Sameer Singh. Novelty detection: a review, part 1: statistical approaches. *Signal Processing*, 83:2481–2497, 2003.

[78] Markos Markou and Sameer Singh. Novelty detection: a review, part 2: neural network based approaches. *Signal Processing*, 83:2499–2521, 2003.

[79] S. Marsland, Nehmzow U., and J. Shapiro. A model of habituation applied to mobile robots. In *Proceedings of TIMR, Towards Intelligent Mobile Robots*, Bristol, 1999.

[80] S. Marsland, Nehmzow U., and J. Shapiro. A real-time novelty detector for a mobile robot. In *Proceedings of the European Advanced Robotics Systems Conference*, Salford, 2000.

[81] M. Masson and T. Denoeux. Inferring a possibility distribution from empirical data. *Fuzzy Sets and Systems*, 157(3):319–340, 2006.

[82] O. Mazhelis. One-class classifiers: a review and analysis of suitability in the context of mobile-masquerader detection. *ARIMA/SACJ, joint special issue : advances in end-user data-mining techniques*, 36, 2006.

[83] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 7:115 – 133, 1943.

[84] G. McLachlan and Peel D. *Finite Mixture Models*. Wiley, New York, 2000.

[85] D. Mercier, G. Cron, T. Denoeux, and Masson M. Fusion of multi-level decision systems using the transferable belief model. In *Proceedings of FUSION'2005*, Philadelphia, July 2005.

[86] D. Mercier, B. Quost, and T. Denoeux. Contextual discounting of belief functions. In *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU*, pages 552–662, 2005.

[87] D. Mercier, B. Quost, and T. Denoeux. Refined modeling of sensor reliability in the belief function framework using contextual discounting. *Information Fusion*, 2007, to appear.

[88] H. Morel, T. Krysinski, M. Ouladsine, and D. Brun-Picard. Diagnostic de défauts du rotor de l'hélicoptère par som supervisées : application aux mesures issues de vols de déréglages. In *Proceedings of MajecSTIC 2006, Manifestation des Jeunes Chercheurs francophones dans les domaines des STIC*, Lorient, France, 2006.

[89] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.

[90] Lab. of Computer, dpt of Computer Science Information Science, and Helsinki University of Technology Engineering. Bibliography of som papers. Available at `http://www.cis.hut.fi/research/som-bibl/`, 2007.

[91] R.A. Olshen et al. Gait analysis and the bootstrap. *The annals of statistics*, 17:1419–1440, 1989.

[92] N.R. Pal, J.C. Bezdek, and R. Hemasinha. Uncertainty measures for evidential reasoning i: a review. *International Journal of Approximate reasoning*, 7:165–183, 1992.

[93] B.U. Park and J.S. Marron. Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Association*, pages 66–72, 1990.

[94] B.U. Park and B.A. Turlach. practical performance of several data driven bandwidth. *computational statistics*, 7:251–270, 1992.

[95] S. Petit-Renaud and T. Denoeux. Nonparametric regression analysis of uncertain and imprecise data using belief functions. *International Journal of Approximate Reasoning*, 35(1):1–28, 2004.

[96] J. Pickands. Statistical inference using extreme order statistics. *Annals of Statistics*, 3:119–131, 1975.

[97] S.J. Qin and R. Dunia. Determining the number of principal components for best reconstruction. In *International federation of automatic control dynamics and control of Process Systems symposium (IFAC DYCOPS)*, Greece, 1998.

[98] C. P. Quesenberry and D. C. Hurst. Large sample simultaneous confidence intervals for multinomial proportions. *Technometrics*, 6(2):191–195, 1964.

[99] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers out of marge data sets. In *Proceedings of the ACM SIGMOD conference on management of data*, pages 427–438, Dallas, 2000.

[100] R. Ramazan Gencay, F Selcuk, and A. Ulugulyager. High volatility, thick tails and extreme value theory in value-at-risk estimation. *Insurance: Mathematics and Economics*, 33:337–356, 2003.

[101] D.L. Reilly, L.N. Cooper, and C. Elboum. A neural model for category learning. *Biological Cybernetics*, 45:35–41, 1982.

[102] S.J. Roberts. Novelty detection using extreme value statistics. *IEE Proceedings Vision, Signal, Image Processing*, 146:124–129, 1999.

[103] S.J. Roberts. Extreme value statistics for novelty detection in biomedical signal processing. In *Proceedings of the first international conference on advances in medical signal and information processing*, pages 166–172, 2002.

[104] R. Rosipal. Matlab-code. Available at `http://www.ofai.at/~roman.rosipal/soft_data.html`.

[105] P. Rouseeuw and A. Leroy. *Robust regression and outliers detection*. John Wiley and sons, New York, 1996.

[106] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing*, volume 1, pages 318–362. MIT Press, Cambridge, MA, 1986.

[107] M. Sarma. Extreme value theory and financial risk management. 2002.

[108] Glenn A. Satten. Upper and lower bound distributions that give simultaneous confidence intervals for quantiles. *The American statistical journal*, 90:747–752, 1995.

[109] R. E. Schafer and J. J. Angus. Estimation of Weibull quantiles with minimum error in the distribution function. *Technometrics*, 21:367–370, 1979.

[110] B. Schölkopf and A.J. Smola. *Learning with kernels*. MIT Press, Cambridge, 2002.

[111] B. Schölkopf, A.J. Smola, and K-R. Mûller. Nonlinear component analysis as a kernel eigenvalue problem. Technical report, Max-Planck institut fur biologishe kybernetik, 1996.

[112] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[113] G. Shafer. *A theory of statistical evidence*, volume 2, pages 365–436. Springer, Berlin, 1976.

[114] B.W. Silverman. *Density estimation for statistics and data analysis*, chapter xx, page xx. Chapman and Hall, London, 1986.

[115] Ph. Smets. Information content of an evidence. *International Journal of Man-Machine Studies*, 19:33–43, 1983.

[116] Ph. Smets. Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning*, 9:1–35, 1993.

[117] Ph. Smets. Belief induced by the partial knowledge of the probabilities. In D. Heckerman *et al.*, editor, *Uncertainty in AI'94*, pages 523–530. Morgan Kaufmann, San Mateo, 1994.

[118] Ph. Smets. The alpha-junctions: Combination operators applicable to belief functions. In *ECSQARU'97*, Bad Honnef, Germany, June 1997.

[119] Ph. Smets. The application of the Transferable Belief Model to diagnosis problems. *International Journal of Intelligent Systems*, 13:127–158, 1998.

[120] Ph. Smets. The transferable belief model for quantified belief representation. In D. M. Gabbay and Ph. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 1, pages 267–301. Kluwer, Doordrecht, The Netherlands, 1998.

[121] Ph. Smets. *The transferable belief model for quantified belief representation*, pages 267–301. Kluwer, Doordrecht, 1998.

[122] Ph. Smets. Practical uses of belief functions. In K. B. Laskey and H. Prade, editors, *Uncertainty in Artificial Intelligence 15 (UAI99)*, pages 612–621, Stockholm, Sweden, 1999.

[123] Ph. Smets. Belief functions on real numbers. *International Journal of Approximate Reasoning*, 40:181–223, 2005.

[124] Ph. Smets. Decision making in the tbm: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning*, 38:33–147, 2005.

[125] Ph. Smets. Analyzing the combination of conflicting belief functions. *Information Fusion (In press)*, 2007. doi:10.1016/j.inffus.2006.04.003.

[126] Ph. Smets and R. Kennes. The transferable belief model. *Artificial Intelligence*, 66:191–234, 1994.

[127] J. Tang, Z. Chen, A. Fu, and D. Cheung. A robust outlier detection scheme for large data sets. In *Proceedings of the conference on Advances in Knowledge Discovery and Data Mining*, volume 2336 of *Lecture Notes in Computer Science*. Springer, 2002.

[128] D.M.J. Tax and R.P.W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[129] D. Titterington, A. Smith, and Makov U. *Statistical Analysis of Finite Mixture Distributions*. John Wiley and Sons, New York, 1985.

[130] F. Tonon. Using random set theory to propagate epistemic uncertainty through a mechanical system. *Reliability Engineering & System Safety*, 85(1–3):169–181, 2004.

[131] S. Valle, W. Li, and S.J. Qin. Selection of the number of principal components: the variance of the reconstruction error with a comparison to other methods. *Industrial and Engineering Chemistry*, 38, 1999.

[132] Vapnik V.N. *the nature of statistical learning theory*. Springer, New-York, 2000.

[133] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman and Hall, London, 1995.

[134] R. R. Yager. The entailment principle for Dempster-Shafer granules. *International Journal of Intelligent Systems*, 1:247–262, 1986.

[135] R. R. Yager. Cumulative distribution functions from Dempster-Shafer belief structures. *IEEE Transactions on Systems, Man and Cybernetics B*, 34(5):2080–2087, 2004.

[136] Ronald R. Yager. Arithmetic and other operations on dempster-shafer structures. *International Journal of Man-Machine Studies archive*, 25:357 – 366, 1986.

[137] Whenming Zheng, Cairong Zou, and Li Zhao. An improved algorithm for kernel principal component analysis. *Neural processing Letters*, 22:49–56, 2006.

[138] L. M. Zouhal and T. Denoeux. An adaptive *k*-NN rule based on Dempster-Shafer theory. In *Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns (CAIP'95)*, pages 310–317, Prague, September 1995. Springer Verlag.

Part IV

Appendices

Additional PMAT units

No.	Subprocess or potential problem to be monitored	Associated variables	Action
12	Sensor drift or parasite air entrance	% O_2 at boiler exit, % O_2 in chimney	Calculate the difference between the two measures (using 30 minute sliding means). Possibly do a CUSUM test on the mean of the difference value.
13	Water or steam leak	Sum over an hour of the water flow, Sum over an hour of the steam flow	Calculate the difference between the two measures. Possibly do a CUSUM test on the mean of the difference value.
14	Water level in water tank (global steadiness of the boiler activity)	Water tank level	CUSUM test on the variance of the measure
15	Steadiness of O_2 flow, value of O_2 flow	O_2 flow, O_2 flow upper threshold, O_2 flow lower threshold	Calculate amplitude and frequency of thresholds overruns
16	Steadiness of CO flow, value of CO flow (sensor dirt or bad combustion)	CO flow, CO flow upper threshold	Calculate amplitude and frequency of thresholds overruns
17	Bad combustion or bad process monitoring	Burner gas start	Calculate frequency of burner start and average time length during which the burners are on.
18	Assessment of the reliability of the main sensors (not listed here)	CUSUM test on the mean or variance of the measured value. A change may indicate either a sensor fault or a system fault. Determination of the cause requires cross-checking with other classifiers.	

Table A.1: Variables to be monitored other than those mentioned in 6.1, associated process points, and action to be taken.

Intuitive justification of expressions (2.48), (2.57) and (2.58)

In addition to the formal proof given in Section 2.2.4, a fairly intuitive justification of expressions (2.48), (2.57) and (2.58) may be given: for instance, $bel([x,y])$ represents the sum of masses on all the intervals included in $[x;y]$.

$$bel([x,y]) = \int_{[a;b]\subseteq[x;y]} \underline{f}(b)\delta(a - \overline{F}^{-1}(\underline{F}(b)))dbda. \tag{B.1}$$

Let us first show that this expression may be expressed as a function of b only (or a only).

From (2.41-2.43), it is easy to see that the focal elements $[a,b]$ of m are intervals such that:

$$\begin{aligned} & b = && (\underline{F})^{-1} \circ \overline{F}(a), && \text{(B.2)}\\ \Leftrightarrow\quad & a = && \overline{F}^{-1} \circ \underline{F}(b), && \text{(B.3)}\\ \Leftrightarrow\quad & \overline{F}(a) = && \underline{F}(b). && \text{(B.4)} \end{aligned}$$

Let us denote:

$$x' = (\underline{F})^{-1}(\overline{F}(x)), \tag{B.5}$$

$$\text{and} \tag{B.6}$$

$$y' = \overline{F}^{-1}(\underline{F}(y)). \tag{B.7}$$

If $x' > y$ and $y' < x$, then there isn't any focal interval $[a;b]$ included in $[x;y]$ and $bel([x;y]) = 0$.

Now, note that, if $x' \leq y$ and $y' \geq x$, any (focal) interval $[a,b]$ (as defined in B.2) whose upper-bound b is smaller than $x' = (\underline{F})^{-1}(\overline{F}(x))$ is not included in $[x,y]$, as its lower bound is smaller than x. Moreover, any interval $[a,b]$ verifying (B.2) and whose lower-bound a is greater than $y' = \overline{F}^{-1}(\underline{F}(y))$ is not included in $[x,y]$ either, as its upper bound is greater than y (see Figure B).

Finally, note that $y' \leq y$, $x' \geq x$, and, if $x' \leq y$ and $y' \geq x$,

$$\left.\begin{array}{l} a \in (-\infty;y') \\ a \in [x;y] \end{array}\right\} \quad \Rightarrow \quad a \in [x;y'), \tag{B.8}$$

and that

$$\left.\begin{array}{l} b \in (x';\infty) \\ b \in [x;y] \end{array}\right\} \quad \Rightarrow \quad b \in (x';y]. \tag{B.9}$$

From (B.8) and (B.9), we may write

$$bel([x;y]) = \int_{b\in(x';y]} \underline{f}(b)db = \int_{a\in[x;y')} \overline{f}(a)da. \tag{B.10}$$

Now let us show that this may be expressed as a function of $\underline{F}$, $\overline{F}$, x and y.

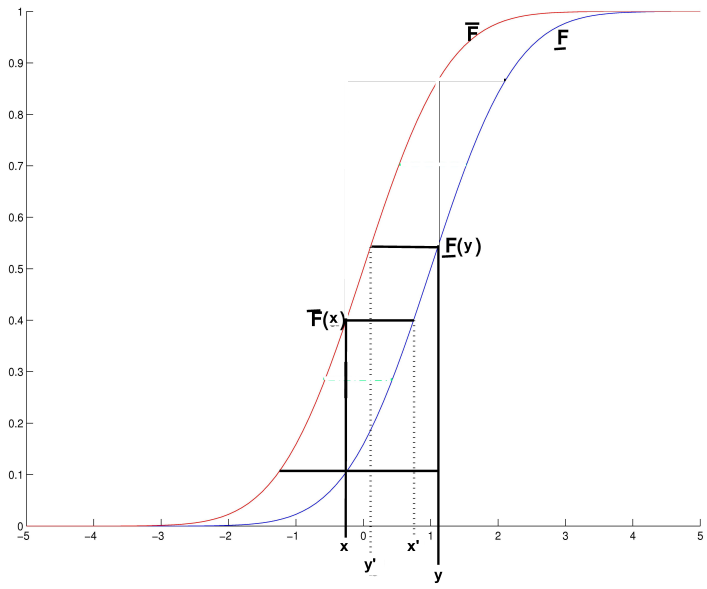


Figure B.1: Calculation of the least committed bel associated with a continuous confidence band

The following relation holds:

$$\underline{F}(y) = \int_{b \subseteq (-\infty;y]} \underline{f}(b)db = \int_{b \subseteq (-\infty;x') \cup [x',y]} \underline{f}(b)db \tag{B.11}$$

Similarly,

$$\overline{F}(x) = \int_{a \subseteq (-\infty;x]} \overline{f}(a)da = \int_{b \subseteq (-\infty;x']} \underline{f}(b)db \tag{B.12}$$

Hence, $\underline{F}(y) - \overline{F}(x) = \int_{b \subseteq [x',y]} \underline{f}(b)db = bel[x,y]$.

Similar arguments may be developed for the expressions of pl and q.

Proof of Proposition 7

Reminder: Proposition 7 reads: The least committed belief function $bel_1^{\mathcal{T}}$ compatible with constraints (4.29) is defined by the following bbd:

$$m_1^{\mathcal{T}}(-\infty, t) = \int_{-\infty}^{t} m_0(u; t)du$$

Proof. As a belief function $pl_2^{\mathcal{T}}$ is less committed than another belief function $pl_1^{\mathcal{T}}$ iff $pl_1^{\mathcal{T}}(A) \geq pl_2^{\mathcal{T}}(A)$, $\forall A \in^{\mathcal{T}}$, the least committed belief function pl_1 satisfying 4.29 is the one that maximizes pl_1 under the constraint of Equation (4.29). Consequently, the LCBF satisfying 4.29 is the one for which the equality in (4.29) is reached. Hence we need:

$$pl_1[t; +\infty) = pl_0[t; +\infty), \quad , \forall t \in \mathcal{T}. \tag{C.1}$$

Now, $pl_0([t; +\infty))$ is the integral of bdd m_0 on all intervals whose intersection with $[t; +\infty)$ is not empty. Let dt be an infinitesimal quantity. Then $pl_0([t - dt; +\infty))$ is the integral of bdd m_0 on all intervals whose intersection with $[t - dt; +\infty)$ is not empty. Hence, the difference $pl_0([t - dt; +\infty)) - pl_0([t; +\infty))$ is the integral of m_0 on all intervals intersecting with $[t - dt; +\infty))$ but not with $[t; +\infty))$ (cf. shaded area on Figure C). The lower bound u of such intervals may vary between $-\infty$ and t, while their upper bound may vary between $\max(u, t - dt)$ and t.

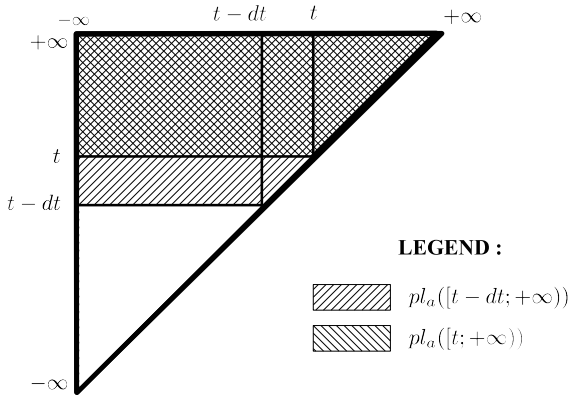


Figure C.1: Representation of $pl_0([t - dt; +\infty))$ and $pl_0([t; +\infty))$

Hence,

$$\Delta(m) = pl_0([t - dt; +\infty)) - pl_0([t; +\infty)) = \int_{u=-\infty}^{t} \int_{v=\max(u,t-dt)}^{t} m_0(u; v)dvdu. \tag{C.2}$$

As we require

$$pl_1([t - dt; +\infty)) = pl_0([t - dt; +\infty)) \tag{C.3}$$

and

$$pl_1([t; +\infty)) = pl_0([t; +\infty)), \tag{C.4}$$

we also obviously require

$$pl_1([t-dt;+\infty)) - pl_1([t;+\infty)) = pl_0([t-dt;+\infty)) - pl_0([t;+\infty)). \tag{C.5}$$

Thus, the LCBF m_1 that satisfies this equality is the one that allocates the amount of belief $\Delta(m)$ to the biggest possible interval that intersects with $[t-dt;+\infty)$ but not with $[t;+\infty)$, namely $(-\infty;t)$. Therefore, m_1 is the bbd such that

$$m_1(-\infty;t) = \int_{u=-\infty}^{t} \int_{v=\max(u,t-dt)}^{t} m_0(u;v)dvdu. \tag{C.6}$$

When $dt \to 0$, this becomes:

$$m_1(-\infty;t) = \int_{u=-\infty}^{t} m_0(u;t)du. \tag{C.7}$$

Subsequently, masses m_1 are allocated to intervals of the form $(-\infty;t)$, with $t \in \mathbb{R}$. It may be checked that requirement (4.29) is met:

$$\begin{aligned} pl_1([t;+\infty)) &= \int_{v=-\infty}^{\infty} m_1(-\infty,v)dv \\ &= \int_{v=-\infty}^{\infty} \int_{u=-\infty}^{t} m_0(u,v)dvdu \\ &= \int_{u=-\infty}^{t} \int_{v=-\infty}^{\infty} m_0(u,v)dudv, \text{ as } u \text{ and } v \text{ are independant} \\ &= \int_{u=-\infty}^{t} \int_{v=\max(u,-\infty)}^{\infty} m_0(u,v)dudv, \text{ as } m_0 \text{ is defined for } v \geq u \\ &= pl_0([t;\infty)) \end{aligned} \tag{C.8}$$

□

Titre Détection de nouveauté dans le cadre de la théorie des fonctions de croyance. Application à la surveillance d'un système d'incinération de déchets.

Résumé Cette thèse apporte deux contributions principales, l'une à la construction de fonctions de croyance et l'autre au problème de détection de nouveauté. La première partie de la thèse résume les principales notions de la théorie des fonctions de croyance (FC) avant d'introduire les contributions associées. Le problème considéré est celui dans lequel la variable d'intérêt est définie comme le résultat dune expérience aléatoire. Deux techniques basées sur des observations passées, et permettant de prédire quelle sera la prochaine observation, sont introduites. La seconde partie de la thèse établit un état de l'art de la classification à une classe avant de montrer quels peuvent être les apports de la théorie des FC dans ce domaine, notamment pour la comparaison ou la combinaison des sorties de différents classifieurs. Une application à la surveillance d'un procédé d'incinération de déchets est présentée en troisième partie de la thèse. Les résultats obtenus sont détaillés et critiqués.

Mots-clés Fonctions de croyance, Reconnaissance des Formes, Fusion d'Informations, Théories de l'Incertain, Diagnostic.

Title Novelty detection in the Belief Function Framework, with application to the monitoring of a waste incineration process.

Abstract The main two contributions of this PhD Thesis are related with the belief function theory and the problem of novelty detection. The thesis is divided into three parts. The first part introduces the main notions pertaining to the belief function theory before describing the associated contributions. The special case considered here is that where the variable of interest is defined from the result of a random experiment. Based on past observations, we introduce two different approaches to predict what the next observation will be. In the second part, the state of the art on the one-class classification problem is summarized before the benefits of the use of belief functions in this domain are shown. Indeed, this theory can be used together with novelty detectors so that the outputs of different classifiers are all expressed in the form of belief functions. The latter can then be compared or combined. Finally, an application to the monitoring of waste incineration plants is detailed.

Keyword Belief functions, Pattern Recognition, Information Fusion, Uncertainty Management, Diagnosis.