UCLouvain

FACULTE DES SCIENCES

Faculté des sciences

Détection d'anomalies : méthodes basées sur l'analyse des valeurs extrêmes

Mémoire présenté en vue de l'obtention du grade académique de Master [120] en Science des données orientation statistique

Auteur : Wautier Lara
Promoteur : Segers Johan
Lecteur : El Ghouch Anouar
Institut de Statistique, Biostatistique et Sciences Actuarielles
Année académique 2022-2023

Abstract

La détection d'anomalies est le processus permettant d'identifier une observation considérée comme anormale. Cela permet notamment d'effectuer de la détection de fraude ou encore de diagnostiquer une maladie chez un patient. Pour ce faire, il existe une multitude de techniques tirées de modèles statistiques et de machine learning. Dans ce mémoire, nous allons nous intéresser aux méthodes liées à l'analyse des valeurs extrêmes, c'est-à-dire l'Extreme Value Machine, le Generalized Pareto Distribution Classifier et enfin le Generalized Extreme Value Classifier. Nous montrerons sur diverses bases de données que ces méthodes sont plus efficaces que celles du One-class Support Vector Machine et de l'Isolation Forest.

Novelty detection is the process of identifying an observation considered as abnormal. This allows to make fraud detection or diagnose a patient's disease. To do so, there exists a variety of techniques from statistical models to machine learning. In this thesis, we will investigate the methods related to the extreme value theory. That is the Extreme Value Machine, the Generalized Pareto Distribution Classifier and finally the Generalized Extreme Value Classifier. We will show on various databases that these methods are more efficient than One-class Support Vector Machine and Isolation Forest.

Remerciements

J'adresse mes sincères remerciements à mon promoteur M. Johan Segers qui a accepté d'encadrer ce mémoire et qui, par ses conseils et réponses à mes questions durant mes recherches, à permis de mener mes réflexions à terme. Je remercie aussi M. Anouar El Ghouch qui a accepté d'endosser le rôle de lecteur de ce mémoire.

J'adresse également mes remerciements à mon père Serge Wautier pour avoir accepté de relire et corriger mon orthographe.

Je tiens ensuite tout spécialement à remercier mes amis Corentin Lingier et Alexandre Fiset qui m'ont sauvé la mise lors de l'utilisation de certains packages et logiciels me donnant du fil à retordre. L'utilisation de toute la méthode d'Extreme Value Machine aurait été compromise sans l'intervention de Corentin.

Enfin, je remercie mon amie Mathilde Foulon pour son soutien non seulement durant la rédaction de ce mémoire mais également durant tout notre parcours de master.

Table des matières

Table des figures

Liste des tableaux

Nomenclature

$\mathbb{D}$	Domaine d'attraction
$\mathbb{P}$	Probabilité
$\mathcal{F}$	Feature space
$\mathcal{G}$	Formule généralisée des trois distributions des valeurs extrêmes
$\text{supp}(f)$	Support de f
ξ	Paramètre des distributions des valeurs extrêmes
D	Distance
F	Fonction de répartition
f	Fonction de densité
$F^n(x)$	Fonction de répartition du maximum de l'échantillon i.i.d. de taille n
G	Fonction de répartition non dégénérée
l	Fonction à variation lente
M_n	Maximum d'un échantillon
p	Dimension de l'espace prédicteur
X	Variable aléatoire
x	Observation de la variable aléatoire X
x_*	Borne supérieure
x_0	Point à classer comme normal ou anormal
$X_{(k)}$	$k^{\text{ième}}$ plus petite valeur d'un échantillon

Abréviations

AUC Area Under the ROC Curve.

EVM Extreme Value Machine.
EVW Extreme Value Weibull.

GEV Generalized Extreme Value.
GEVC Generalized Extreme Value Classifier.
GPD Generalized Pareto Distribution.
GPDC Generalized Pareto Distribution Classifier.

iForest Isolation Forest.

ROC Receiver Operating Characteristic.

SVM Support Vector Machine.

Chapitre 1

Introduction

De nos jours, la technologie occupe une place importante dans la vie de tout un chacun. Nous avons tous accès à des smartphones, des ordinateurs et bien d'autres appareils en tout genre. De ce fait, la technologie ne cesse de s'améliorer et, avec elle, les moyens de frauder s'adaptent et progressent. Heureusement, des méthodes de détection de fraude existent et sont efficaces mais doivent continuer de se perfectionner. Cette amélioration de la technologie de pointe permet également de faire émerger de nouveaux dispositifs médicaux ou de faire de nouvelles découvertes et cela dans le but d'améliorer la vie et la santé des gens. A la poursuite de cet objectif, une détection toujours plus efficace des maladies est nécessaire. Ceci ne sont que quelques exemples parmi tant d'autres qui montrent l'importance de la détection d'anomalies. Pour ce faire, il existe une multitude de techniques tirées de modèles statistiques et de machine learning. Dans ce mémoire, nous allons nous intéresser aux méthodes de détection d'anomalies liées à l'analyse des valeurs extrêmes.

La théorie des valeurs extrêmes intervient dans bien des aspects dont voici quelques exemples pratiques tirés de Castillo (2012). En météorologie, il est important de s'intéresser à l'apparition d'événements extrêmes tels que des températures extrêmes ou des tempêtes qui auront un impact considérable sur l'agriculture ou sur du matériel en tout genre comparé à la météo habituelle. Un autre exemple peut être donné avec la hauteur des vagues qui est importante dans le cadre de l'ingénierie océanique pour, par exemple, construire des digues ou des plateformes en mer dont la conception repose sur la connaissance d'apparition des plus hautes vagues. L'analyse des valeurs extrêmes implique donc de modéliser la distribution du maximum dans un échantillon de variables aléatoires. Sous certaines conditions décrites au chapitre 2, cette distribution ne peut appartenir qu'à l'une des trois lois : Weibull, Gumbel ou Fréchet. L'analyse des distances minimales entre les observations est au centre du domaine de la détection d'anomalies via la théorie des valeurs extrêmes. Parmi les trois lois citées, c'est celle de Weibull qui nous permettra

de déterminer si un nouveau point est normal ou non. Nous nous intéresserons dans le chapitre 3 à trois méthodes liées à cette théorie, à savoir l'Extreme Value Machine (EVM) de Rudd et al. (2018), le Generalized Pareto Distribution Classifier (GPDC) et le Generalized Extreme Value Classifier (GEVC), tous deux de Vignotto and Engelke (2020). Nous nous intéressons également à deux méthodes dont le domaine ne relève pas des valeurs extrêmes : le One-class Support Vector Machine (SVM) de Schölkopf et al. (1999) et l'Isolation Forest (iForest) de Liu et al. (2012).

L'EVM se base sur l'approximation de la distribution des distances marginales, c'est-à-dire, la distance entre les observations de différentes classes qui peut être vue comme une frontière entre ces dites classes. Ce classifieur à l'avantage de prendre en compte l'apparition de nouvelles classes dans le modèle mais nécessite la présence d'au moins deux classes normales durant l'entrainement du modèle. Déterminer la normalité d'une nouvelle observation repose sur la plus grande probabilité d'appartenir à une classe, estimée pour chacune d'elles à partir de la distribution de Weibull.

Les deux méthodes suivantes de Vignotto and Engelke (2020) ont l'avantage de se baser sur la distance entre les points et non pas entre les classes. Il n'est donc pas nécessaire de disposer de plusieurs classes pour créer un modèle. Le principe du GPDC est de déterminer si un nouveau point a été généré par la même fonction de densité que les points normaux. Pour ce qui est du GEVC, il nous faudra comparer la distance qui sépare le nouveau point avec son plus proche voisin afin de décider sur base d'un seuil s'ils sont suffisamment proches pour que ce point puisse être considéré comme normal.

Vient ensuite le SVM qui permet de faire de la classification dans de grandes dimensions et lorsque les classes ne sont pas linéairement séparables grâce à des fonction noyaux qui permettront d'agrandir l'espace de représentation des données d'entrées et ainsi d'établir une frontière linéaire entre les classes. Dans le cas de la détection d'anomalies, celles-ci seront représentées par l'origine qu'il faudra séparer du reste des données. Une fois la frontière établie, la nouvelle observation pourra être classée comme normale ou non en fonction de sa position par rapport à cette frontière.

Enfin, l'iForest consiste à isoler les données trop différentes des points normaux et à les considérer comme des anomalies, ce qui nécessite des calculs bien moins complexes que les méthodes précédentes impliquant la construction du profil des points normaux pour ensuite identifier les points ne s'y conformant pas. L'iForest va générer aléatoirement des arbres de décision dont le but sera d'isoler chaque point en partitionnant les données de manière récursive. Les anomalies correspondent aux points nécessitant peu de partitions pour être isolés.

Le chapitre 4 permettra de comparer ces différentes méthodes afin de vérifier si celles basées sur la théorie des valeurs extrêmes sont effectivement plus efficaces

que les autres, comme l'assurent Vignotto and Engelke (2020). Nous commencerons par évaluer les trois premières méthodes sur des données simulées à partir de distributions normales bivariées. Nous utiliserons le AUC comme méthode d'évaluation de la précision. Nous mettrons également en évidence un désavantage du SVM. Nous appliquerons ensuite, grâce aux données LETTER de Frey and Slate (1991), le protocole OLETTER de Bendale and Boult (2015) sur les cinq algorithmes. Ce protocole permet d'évaluer la performance d'un système de classification sur base de la mesure F en tenant compte du nombre de classes utilisées dans le modèle. Enfin, nous appliquerons ces méthodes à deux cas médicaux. Le premier portant sur le diagnostic d'une maladie de la thyroïde et le second sur le diagnostic du cancer du sein au Wisconsin, dont les données proviennent respectivement de Quinlan et al. (1987) et Mangasarian et al. (1995). Les anomalies étant considérées comme les personnes malades, il ne sera pas possible d'utiliser l'EVM, qui nécessite la présence de plusieurs classes normales. A nouveau, nous utiliserons le AUC pour évaluer les différentes précisions.

Enfin, dans le chapitre 5 nous essaierons de mettre les algorithmes de Vignotto and Engelke (2020) en difficulté afin d'en trouver les limites. Nous verrons notamment si ces méthodes sont sensibles à la proximité des classes et nous jouerons sur leur balancement.

Chapitre 2

Théorie des valeurs extrêmes

Dans le cadre de l'analyse des valeurs extrêmes, nous nous intéressons à l'analyse du maximum dans un échantillon et plus particulièrement à sa distribution. Nous disposons d'un échantillon comprenant n observations indépendantes et identiquement distribuées $X_1, \ldots, X_n \sim i.i.d.$ Notons le maximum de cet échantillon :

$$M_n = \max(X_1, \ldots, X_n)$$

et la borne supérieure :

$$x_* = \sup\{x \in \mathbb{R}, F(x) < 1\} \in \mathbb{R} \cup \{+\infty\}.$$

Un exemple de distribution dont la borne supérieure est finie est une loi uniforme entre 0 et 1 pour laquelle x_* vaut 1. En revanche, x_* n'est pas fini pour des lois telles que des lois normales ou exponentielles.

Pour analyser la distribution d'un maximum, la notion de convergence en distribution est nécessaire. Pour rappel, si F est une fonction de répartition, lorsque nous avons une variable aléatoire $Y_1, \ldots, Y_n$ avec $F_n(y) = \mathbb{P}(Y_n \leq y)$, si pour tout $y \in \mathbb{R}$ tel que F est continue en y on a :

$$\lim_{n \to \infty} F_n(y) = F(y),$$

alors Y_n converge en loi vers Y.

Pour appliquer cette notion au maximum dont nous voulons connaître la distribution, nous devons connaître sa fonction de répartition F qui se calcule de la façon suivante :

$$\begin{aligned}
\mathbb{P}(M_n \leq x) &= \mathbb{P}(\max(X_1, \ldots, X_n) \leq x) \\
&= \mathbb{P}(X_1 \leq x, \ldots, X_n \leq x) \\
&= \mathbb{P}(X_1 \leq x) \cdots \mathbb{P}(X_n \leq x) \qquad \text{car i.i.d.} \\
&= F(x) \cdots F(x) \\
&= F^n(x).
\end{aligned}$$

La fonction de répartition du maximum de l'échantillon i.i.d. de taille n est donc $F^n(x)$ dont les limites sont, lorsque $x < x_*$: $\lim_{n\to\infty} F^n(x) = 0$ car $F(x) < 1$ et quand $x \geq x_*$: $\lim_{n\to\infty} F^n(x) = 1$ car $F(x) = 1$.

Cette fonction est dégénérée car elle décrit une fonction de répartition d'une variable aléatoire qui n'est en réalité pas aléatoire. Sa variance est nulle ce qui n'est pas intéressant. Nous allons donc tenter de trouver une fonction de répartition pour le maximum standardisé qui sera non dégénérée. Pour ce faire, commençons par rappeler le principe du théorème central limite. Pour la somme d'une suite de variables aléatoires $S_n = X_1 + \cdots + X_n$ dont la moyenne empirique est $\bar{x} = \frac{1}{n}S_n$. Si la variance est finie, on peut définir $\mu = \mathbb{E}[X_1]$, $\sigma^2 = \mathrm{Var}(x)$ et si $\sigma > 0$, alors nous avons :

$$\sqrt{n}\frac{\bar{X}_n - \mu}{\sigma} = \frac{S_n - n\mu}{\sqrt{n}\sigma} \xrightarrow{\mathrm{d}} \mathcal{N}(0,1).$$

Le même principe que dans ce théorème est utilisé sur le maxima standardisé afin d'en trouver la distribution limite :

$$?\exists \quad a_n > 0, \quad b_n \in \mathbb{R} : \quad \frac{M_n - b_n}{a_n} \xrightarrow{\mathrm{d}} G?$$

avec G une fonction de répartition non dégénérée, a_n et b_n des constantes.

Si de telles constantes existent, alors la fonction de répartition de M_n appartient à l'une des trois lois suivantes : Gumbel, Weibull ou Fréchet, regroupées sous le nom de Generalized Extreme Value (GEV). La distribution dont provient le maximum appartient alors au domaine d'attraction de G noté $\mathbb{D}(G)$. Voici ci-dessous un exemple pour chacune de ces trois lois.

Commençons avec la distribution de Gumbel. Prenons une distribution exponentielle dont la fonction de répartition s'écrit :

$$F(x) = \begin{cases} 1 - e^{-x} & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

Avec $a_n = 1$ et $b_n = \ln(n)$, nous obtenons :

$$\begin{aligned} \mathbb{P}\left(\frac{M_n - b_n}{a_n} \leq x\right) &= \mathbb{P}\left(M_n \leq a_n x + b_n\right) \\ &= \left[F(a_n x + b_n)\right]^n \\ &= \left[F(x + \ln(n))\right]^n \\ &= \left[1 - \exp[-(x + \ln(n))]\right]^n \\ &= \left[1 - \frac{1}{n}e^{-x}\right]^n. \end{aligned}$$

La limite de ce résultat vaut $\lim_{n\to\infty}\left[1 - \frac{1}{n}e^{-x}\right]^n = e^{-e^{-x}}$, ce qui correspond à une distribution de Gumbel.

Le même procédé est reproduit pour illustrer la distribution d'Extreme Value Weibull (EVW) mais en partant d'une distribution uniforme dont la fonction de répartition est :

$$F(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } 0 \leq x \leq 1 \\ 1 & \text{si } x > 1 \end{cases}$$

et en prenant $a_n = 1/n$ et $b_n = 1 = x_*$. La fonction de répartition de M_n devient :

$$[F(a_n x + b_n)]^n = F\left(\frac{x}{n} + 1\right)^n = 1 \qquad \text{si } x \geq 0.$$

Si $x < 0$ alors $0 < \frac{x}{n} + 1 < 1$ pour n suffisamment grand, ce qui est plus intéressant que le cas $x > 0$. En effet :

$$F\left(\frac{x}{n} + 1\right)^n = \left(\frac{x}{n} + 1\right)^n \underset{n\to\infty}{\longrightarrow} e^x$$

qui correspond à une distribution d'Extreme Value Weibull.

Pour finir, illustrons la distribution de Fréchet grâce à une distribution de Pareto dont la fonction de répartition s'écrit :

$$F(x) = \begin{cases} 1 - x^{-\alpha} & \text{si } x \geq 1 \\ 0 & \text{si } x < 1. \end{cases}$$

En posant $a_n = n^{1/\alpha}$; $\alpha > 0$ et $b_n = 0$, on obtient :

$$\begin{aligned} [F(a_n x + b_n)]^n &= [F(n^{1/\alpha} x)]^n \\ &= [1 - (n^{1/\alpha} x)^{-\alpha}]^n \\ &= \left(1 - \frac{1}{n} x^{-\alpha}\right)^n \underset{n\to\infty}{\longrightarrow} e^{-x^{-\alpha}}. \end{aligned}$$

En sachant que si $x > 0$ alors $n^{1/\alpha} x > 1$ lorsque n est suffisamment grand, il s'agit d'une distribution de Fréchet alors que dans le cas où $x \leq 0$, la fonction de répartition vaut 0 ce qui n'a pas d'intérêt.

La forme des fonctions de densités de ces trois distributions est visible à la figure 2.1 çi-dessous et leurs formules peuvent être généralisées par :

$$\mathcal{G}_\xi(z) = \exp[-(1 + \xi z)^{-1/\xi}] \quad \text{avec } 1 + \xi z > 0. \tag{2.1}$$

Il s'agit alors d'une distribution de Fréchet lorsque $\xi > 0$, de Gumbel lorsque $\xi = 0$ et de Weibull lorsque $\xi < 0$. En principe, cette formule comprend trois paramètres : la forme de la courbe $\xi \in \mathbb{R}$, sa position $\mu \in \mathbb{R}$ et son échelle $\sigma > 0$. Pour faire apparaitre ces deux derniers paramètres, il suffit de remplacer z par $\frac{z-\mu}{\sigma}$ dans l'équation (2.1).

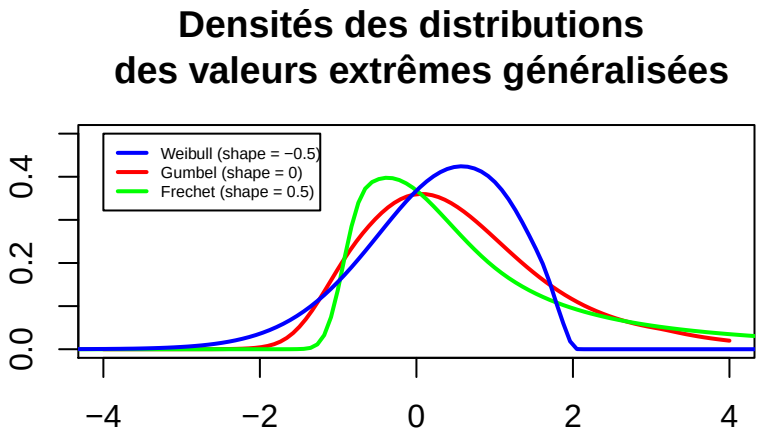

FIGURE 2.1 – Densités des distributions des valeurs extrêmes généralisées

Attention, il n'existe pas $a_n > 0,\ b_n$ tel que $(M_n - b_n)/a_n \xrightarrow{\text{d}} G$ pour toutes les distributions. C'est en effet le cas pour des fonctions suivant une loi de Poisson (λ) ou une loi Binomiale (n, p) par exemple.

Dans le cadre de ce mémoire, c'est la distribution d'Extreme Value Weibull qui nous intéresse. Son domaine d'attraction se décrit en passant par celui de Fréchet. Pour une meilleure compréhension, nous représentons la fonction de répartition d'une variable $V \sim F$ dont la borne supérieure est notée v_* à la figure 2.2.

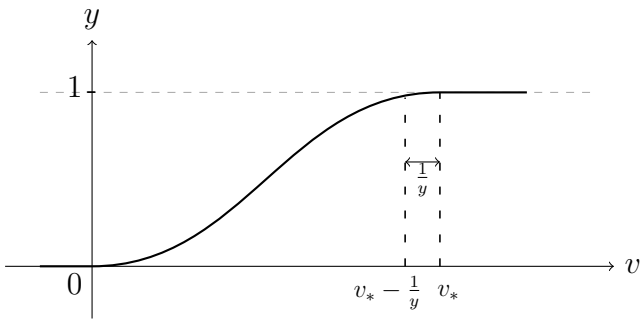

FIGURE 2.2 – Fonction de répartition de V

Mathématiquement, nous pouvons écrire :

$$\begin{aligned} & V \sim F \in \mathbb{D}(\text{Weibull}(\xi)) \\ \iff & \frac{1}{v_* - V} \in \mathbb{D}(\text{Fréchet}(-\xi)) \\ \iff & \mathbb{P}\left(\frac{1}{v_* - V} > y\right) = y^{1/\xi} l(y) \\ \iff & \mathbb{P}\left(V > v_* - \frac{1}{y}\right) = y^{1/\xi} l(y), \end{aligned} \tag{2.2}$$

avec $y > 0$, $\xi < 0$ et l étant une fonction à variation lente. C'est-à-dire une fonction qui varie de manière plus lente qu'une fonction de puissance comme une fonction constante ou une fonction logarithmique, par exemple. Plus précisément, une fonction l positive est à variation lente si $l(tx)/l(x) \to 1$ pour tout $t > 0$ lorsque $x \to \infty$.

Cette équation (2.2) est, comme le montre la figure 2.2, la probabilité que V soit dans le petit intervalle qui se situe juste avant la borne supérieure qu'il peut atteindre. C'est donc la queue de la distribution qui nous intéresse ici. Cela se résume par la proposition suivante.

Proposition 1 Embrechts et al. (2013) *Supposons que la distribution F de V a une borne supérieure notée v_*. Alors F se trouve dans le domaine d'attraction maximum d'une GEV avec $\xi < 0$ si et seulement si*

$$1 - F(v_* - 1/v) = v^{1/\xi} l(v),$$

avec l étant une fonction à variation lente à l'infini. Ceci est équivalent à $(v_* - V)^{-1}$ étant dans le domaine d'attraction d'une Fréchet avec paramètre $-\xi$.

Chapitre 3

Méthodes utilisées

Dans ce chapitre, nous décrivons les cinq algorithmes de détection d'anomalies que nous comparerons ensuite. Pour détecter une anomalie, ces algorithmes suivent la même méthodologie de base. Pourvu d'une base de données d'entraînement ne contenant aucune anomalie, le but est de classer un nouveau point x_0 comme normal ou anormal. Le test d'hypothèse réalisé est :

$$H_0 : x_0 \text{ est normal,}$$
$$H_1 : x_0 \text{ est anormal.}$$

3.1 Extreme Value Machine

La première méthode qui nous intéresse a été introduite par Rudd et al. (2018) et se base sur l'approximation de la distribution des distances marginales en utilisant la théorie des valeurs extrêmes. Cette notion de marge est liée à la distance qui se trouve entre les observations des différentes classes et peut être vue comme une frontière qui sépare ces classes.

Ce classifieur a l'avantage de prendre en compte l'apparition de nouvelles classes dans le modèle. En effet, il se pourrait qu'une classe existant bel et bien ne soit pas représentée dans la base de données d'entrainement. Dans ce cas, un nouveau point appartenant à cette classe devrait être considéré comme normal et classé comme appartenant à cette classe et non être considéré comme anormal.

Si nous disposons d'un point x_i appartenant à la classe c_i, la distance marginale maximale est définie par la moitié de la distance allant de ce point vers le point x_j le plus proche appartenant à une autre classe c_j. L'estimation de la marge m_{ij} pour les points (x_i, x_j) se calcule donc par :

$$m_{ij} = \frac{\|x_i - x_j\|}{2}.$$

Cet algorithme nécessite d'approximer la distribution des distances marginales maximales calculées pour chaque classe par une EVW. Cependant, comme souligné par Vignotto and Engelke (2020), la distribution choisie aurait du être une Generalized Pareto Distribution (GPD). En effet, cette distribution permet de modéliser les queues des distributions et c'est ce que l'on fait ici en modélisant la distribution des k plus petites observations des distances marginales. Cependant, pour retomber sur un problème lié à la théorie des valeurs extrêmes, nous allons modéliser les marges maximales $M_i = \min_{j:c_j \neq c_i}(m_{ij})$ correspondant à des distances minimums. La distribution d'EVW modélisant des maxima négatifs, nous pouvons réécrire ces distances : $M_i = \max_{j:c_j \neq c_i}(-m_{ij})$ avec $-m_{ij} < 0$.

L'équation (2.1) des distributions des valeurs extrêmes généralisées peut être réécrite pour le cas spécifique de Weibull de la façon suivante :

$$\mathcal{W}(z) = \exp\left(-\left(\frac{|z|}{\tau}\right)^{-\alpha}\right),$$

avec $\tau > 0$; $\alpha < 0$; $|z| = -z$; $z < 0$. Cela correspond à la probabilité qu'un maximum soit inférieur à la valeur z. Appliquer cette formule au négatif de la distance séparant x_i et x_0 revient donc à calculer la probabilité que x_0 soit inclus dans la frontière estimée par x_i, autrement dit que x_0 appartienne à la même classe que x_i :

$$\mathcal{W}(-\|x_0 - x_i\|) = \exp\left(-\left(\frac{\|x_0 - x_i\|}{\tau_i}\right)^{-\alpha_i}\right). \tag{3.1}$$

Cette formule montre que si la distance entre x_i et x_0 est grande, la probabilité que ces deux points appartiennent à la même classe sera petite, et inversement.

Afin de classer le nouveau point x_0, il nous faut calculer la probabilité

$$\mathbb{P}(C_l \mid x_0) = \underset{\{i:c_i=C_l\}}{\arg\max}\, \hat{\mathcal{W}}(-\|x_0 - x_i\|), \tag{3.2}$$

et ce pour toutes les classes C_l, afin d'obtenir pour chacune d'elles le point x_i qui maximise cette probabilité. On choisi ensuite C_l qui maximise cette probabilité et si celle-ci est suffisamment grande, c'est-à-dire si elle est supérieure ou égale à un certain seuil δ, on peut considérer que x_0 et x_i appartiennent à cette même classe C_l. Rudd et al. (2018) ont fixé ce seuil δ à

$$\delta = \frac{1}{2}\left(1 - \left(\frac{2N_T}{N_R + N_E}\right)^{1/2}\right), \tag{3.3}$$

avec N_T le nombre de classes utilisées dans la base de données d'entraînement, N_E le nombre de classes utilisées lors de l'évaluation du modèle et N_R le nombre de classes communes aux données d'entraînement et à celles de test.

Cette formule montre que si N_T augmente, δ diminue, la probabilité de rejet de l'hypothèse nulle diminue donc également et si N_E ou N_R augmente, δ augmente faisant cette fois augmenter la probabilité de rejet. Ce seuil se fixe en principe par validation croisée et c'est grâce à cette méthode que Rudd et al. (2018) ont pu établir cette formule (3.3).

Cependant cette classification requiert beaucoup de calculs car il faut estimer τ_i et α_i pour chaque point i. Rudd et al. (2018) proposent donc une réduction de ces calculs via deux approximations. La première consiste à permettre de parcourir uniquement certains points dans une classe C_l qu'on appellera des vecteurs extrêmes au lieu de parcourir tous les points de la classe. En effet, si plusieurs points de la classe sont suffisamment proches les uns des autres comparés aux points d'une autre classe, on peut s'attendre à de la redondance dans leur probabilité $\mathcal{W}$. On va donc fixer un seuil de probabilité de redondance minimum ζ afin de trouver un sous-ensemble le plus petit possible de points de cette classe. Il s'agit donc d'un problème de minimisation.

Les figures 3.1 et 3.2 montrent un exemple de choix de vecteurs extrêmes en rouge dans le cas d'une réduction de modèle correspondant à des seuils $\zeta = 0$ et $\zeta = 1$. Ces modèles ont été construits sur des données simulées comprenant trois classes normales, elles-mêmes construites à partir de trois distributions normales bivariées.

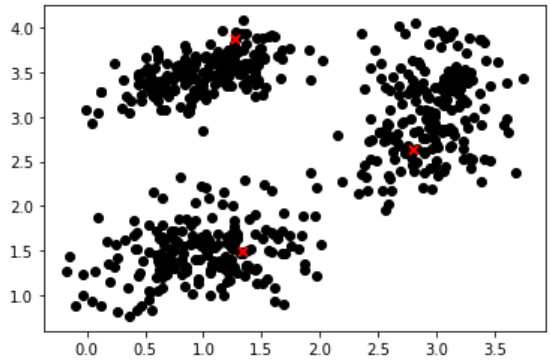

Figure 3.1 – $\zeta = 0$

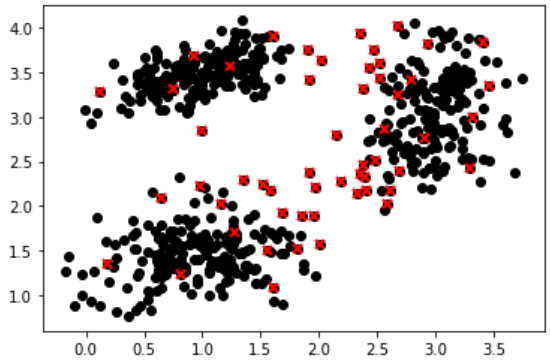

Figure 3.2 – $\zeta = 1$

Pour trouver ces vecteurs extrêmes, nous prenons x_i et x_j appartenant à la même classe c_i mais on fait comme si x_j était un nouveau point et on se demande à quelle classe il appartient. Si la probabilité (3.1) de l'accepter dans la même classe que x_i est supérieur ou égale à ζ (par exemple 0.5) alors x_j est redondant et dans ce cas on ne le garde pas dans nos points d'intérêt. En effet, si la probabilité que x_i et x_j appartiennent à la même classe est supérieure à ce seuil, alors la probabilité que x_j et x_0 appartiennent à la même classe est redondante par rapport à la probabilité que x_i et x_0 appartiennent à la même classe. Il n'est donc pas nécessaire

de garder x_j. Nous pouvons ainsi minimiser le nombre de vecteurs extrêmes retenus et appliquer la première méthode vue en prenant le maximum parmi les vecteurs extrêmes lors du calcul de l'équation (3.2) et non pas le maximum parmi tous les points de la classe considérée.

Cette méthode reste assez complexe car il faudrait en principe parcourir tous les sous-ensembles de vecteurs extrêmes possibles, ce qui correspond à un problème NP-hard, c'est-à-dire qui ne peut être résolu dans un temps polynomial. Rudd et al. (2018) proposent donc une deuxième approximation pour laquelle on se contente d'un bon choix de sous-ensemble de vecteurs extrêmes et non pas du choix optimal contenant les meilleures représentants, ce qui est quasi impossible à trouver. Ce choix acceptable se fait grâce à l'utilisation de l'algorithme proposé par Slavík (1996) que nous n'aborderons pas ici mais qui permet une résolution dans un temps polynomial.

Vignotto and Engelke (2020) ont relevé plusieurs limites à cet algorithme. Premièrement, l'estimation de la distribution devrait se faire sur base d'une distribution GPD et non GEV comme expliqué précédemment. Ensuite, le choix de δ ne permet pas de gérer l'erreur de type I. Enfin, cette méthode favorise de manière injustifiée les classes normales fortement éloignées des autres. En effet, un point anormal situé proche d'une classe normale éloignée, sera classé comme normal et appartenant à cette dite classe éloignée et ce simplement parce que les distances marginales des points de cette classe seront plus grandes. La distance entre les classes ne peut donc pas être utilisée dans le cadre de la détection d'anomalie car elle ne permet pas d'obtenir des informations sur les classes anormales. Les deux méthodes développées par Vignotto and Engelke (2020) pallient ces différents manquements notamment en se basant sur les distances entre les points et non entre les classes.

3.2 Generalized Pareto Distribution Classifier

Dans cette partie, nous nous intéressons au Generalized Pareto Distribution Classifier (GPDC) développé par Vignotto and Engelke (2020) et qui se base sur les distributions généralisées de Pareto. Ce classifieur se base plus précisément sur la distance entre les points normaux $x_1, \ldots, x_n$ provenant du vecteur aléatoire X dont la densité f est connue et le nouveau point x_0 que nous voulons classer comme normal ou anormal. Les points $x_1, \ldots, x_n$ sont ici regroupés en une seule classe contrairement à l'EVM vu à la section précédente. Le but est de déterminer si le point x_0 a été généré par la fonction de densité f ou par une autre densité f_0, auquel cas x_0 sera considéré comme anormal. Pour ce faire nous regardons la distribution des distances $D = \|X - x_0\|$ et supposons que x_0 est normal s'il se trouve à l'intérieur du support de f. C'est-à-dire, $f(x_0) > 0$ et f est continu en x_0. Autrement dit, il existe $z > 0$ tel que, pour tout y se trouvant à une distance de x_0

inférieure à z, $f(y)$ est positive aussi. La figure 3.3 montre deux points b et d ne se trouvant pas dans le support de f.

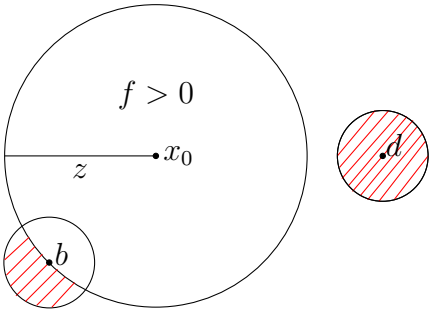

FIGURE 3.3 – Bord et extérieur du support de f

Nous voulons connaître le comportement de la queue basse de la distribution de D qui représente les plus petites distances séparant x_0 des autres points. C'est donc le paramètre de la forme de la distribution ξ qui nous intéresse. En prenant l'opposé des distances, le même raisonnement qu'utilisé dans la description du domaine d'attraction de Weibull vu à l'équation (2.2) peut y être appliqué. En prenant z petit et positif et f continue en x_0, le schéma de la figure 2.2 devient :

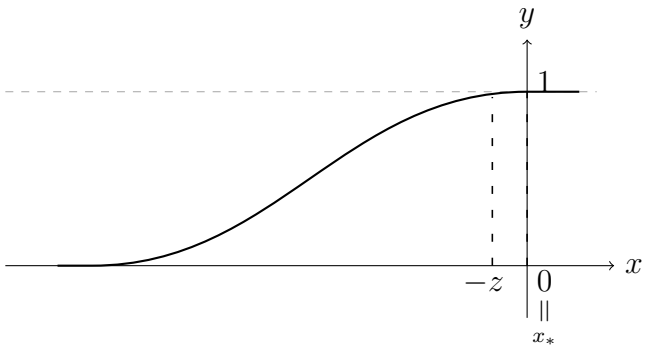

FIGURE 3.4 – Fonction de répartition de distances négatives

Avec p la dimension de l'espace prédicteur et c_p une certaine constante, la probabilité que les distances négatives soient plus grande que $-z$ est :

$$\begin{aligned}
\mathbb{P}\left(-D > -z\right) &= \mathbb{P}\left(D < z\right) \\
&= \mathbb{P}\left(X \in \text{boule}(x_0, z)\right) \\
&= \int_{\text{boule}} f(x)\,\mathrm{d}x \\
&\approx f(x_0)\,\text{Vol}(\text{boule}) = f(x_0)c_p z^p = z^p l(1/z), \qquad (3.4)
\end{aligned}$$

z correspondant à $1/y$ dans l'équation (2.2) ce qui implique que $p = -1/\xi$. Le volume de la boule est représenté à la figure 3.5 et est proportionnel à z^p. En effet, dans un espace de dimension 2, le volume d'un disque équivaut à πr^2. Dans une autre dimension p ce volume équivaut à une certaine constante multiplié par r^p. Nous obtenons donc bien dans notre cas, un volume proportionnel à z^p et on peut, grâce à l'équation (2.2), voir le lien entre la dimension p et le paramètre des valeurs extrêmes $-1/\xi$ qui est repris dans la proposition 2 ci-dessous.

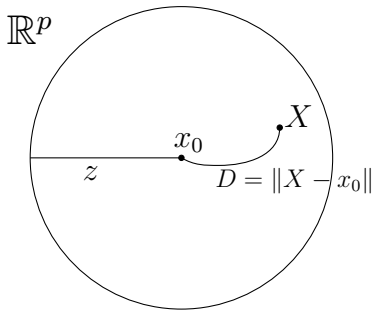

Figure 3.5 – Boule centrée en x_0 et de rayon z

Proposition 2 Vignotto and Engelke (2020) *Supposons que x_0 se trouve à l'intérieur du support* $\text{supp}(f) = \{x : f(x) > 0\}$ *des classes normales et que f est continue en x_0. Alors la distribution de $-D$ est dans le domaine d'attraction maximum de la distribution GEV avec un paramètre de la forme $\xi = -1/p$ où $p \in \mathbb{N}$ est la dimension de l'espace des prédicteurs.*

On sait maintenant qu'en théorie : $\xi = -1/p$. Autrement dit, si lorsque l'on estime ce paramètre $\hat{\xi}$, il se rapproche bien de cette valeur théorique alors on peut considérer x_0 comme normal. Sinon cela veut dire que l'une de nos hypothèses a été violée et notre point sera classé comme anormal.

De par cette proposition 2 et de par l'équation (2.2), nous pouvons écrire :

$$\mathbb{P}\left(1/D > x\right) = x^{-p} l(x)$$

et $1/D$ appartient au domaine d'attraction maximum de la distribution GEV avec un paramètre de la forme $\xi = 1/p$ qui peut être estimé par l'estimateur de Hill, ξ étant positif ici.

Cet estimateur peut être introduit de plusieurs manières mais nous nous focalisons sur celle faisant intervenir les probabilités et nous le noterons $\hat{\alpha}$ dans le cadre de cette explication.

En prenant un échantillon de statistiques d'ordre $B_{(1)} \leq \ldots \leq B_{(n)}$ suivant une distribution $F \in \mathbb{D}(\text{Fréchet}(\alpha))$ et B étant égal à $1/D_i$, alors pour un seuil u

élevé : $B/u \mid B > u$ peut être approchée par une Pareto(α) si u tend vers l'infini. En effet, pour tout $b \geq 1$,

$$\begin{aligned}\mathbb{P}\left(\frac{B}{u} > b \mid B > u\right) &= \frac{\mathbb{P}\left(B > ub, B > u\right)}{\mathbb{P}\left(B > u\right)} \\ &= \frac{1 - F(ub)}{1 - F(u)} \underset{u\to\infty}{\longrightarrow} b^{-\alpha},\end{aligned}$$

ce qui correspond à une distribution de Pareto. Ce sont donc les dépassements de ce seuil u qui nous intéressent. On pose $u = B_{(n-k)}$ qui correspond à la $(k+1)^{\text{ième}}$ plus grande statistique d'ordre. On a donc k dépassements de seuil : $B_{(n-i+1)}$ avec $i = 1, \ldots, k$ et on peut estimer α par maximum de vraisemblance en faisant comme si ce paramètre suivait une distribution de Pareto. A partir de la fonction de densité de $B_{(n-i+1)}/B_{(n-k)}$ correspondant précédemment à B/u,

$$f_\alpha(b) = \frac{\partial}{\partial b}(1 - b^{-\alpha}) = \alpha b^{-\alpha-1},$$

on obtient la fonction de log-vraisemblance pour une observation b :

$$\ln f_\alpha(b) = \ln \alpha - (\alpha + 1) \ln b.$$

De là, on maximise la fonction de log-vraisemblance afin d'obtenir l'estimateur de Hill :

$$\begin{aligned}\hat{\alpha} &= \arg\max_{\alpha>0} \sum_{i=1}^{k} \left(\ln \alpha - (\alpha+1) \ln \frac{B_{(n-i+1)}}{B_{(n-k)}}\right) \\ &= \frac{1}{\frac{1}{k}\sum_{i=1}^{k} \ln \frac{B_{(n-i+1)}}{B_{(n-k)}}}.\end{aligned}$$

Nous appliquons maintenant ce procédé aux distances $1/D_i$ correspondant précédemment aux B_i, où $D_i = \|X_i - x_0\|$, en rappelant que :

$$-D \in \mathbb{D}(\text{Weibull}(\xi < 0)) \iff 1/D \in \mathbb{D}(\text{Fréchet}(|\xi|)).$$

Nous en déduisons le paramètre de la distribution d'Extreme Value Weibull $\xi = -1/\alpha$. L'estimateur devient :

$$\hat{\xi} = \frac{-1}{\hat{\alpha}} = \frac{-1}{k} \sum_{i=1}^{k} \ln \frac{1/D_{(i)}}{1/D_{(k+1)}} = \frac{1}{k} \sum_{i=1}^{k} \ln \frac{D_{(i)}}{D_{(k+1)}}.$$

Sachant que c'est l'opposé des distances qui nous intéresse, nous obtenons l'estimateur de l'équation (3.5) qui converge vers $-1/p$.

Théorème 1 Vignotto and Engelke (2020) *Supposons que x_0 se trouve à l'intérieur du support* $\text{supp}(f)$ *des classes normales et que f est continue en x_0. Notons les distances entre x_0 et les points de la base de données d'entrainement par $D_1, \ldots, D_n$. Notons également $R_{(n)} \geq R_{(n-1)} \ldots \geq R_{(1)}$ les statistiques d'ordre de $R_i = -D_i$. Pour $k = k(n) \to \infty$ et $k(n)/n \to 0$ avec $n \to \infty$,*

$$\hat{\xi}_n = \frac{1}{k} \sum_{i=1}^{k} \ln \left(\frac{R_{(n+1-i)}}{R_{(n-k)}} \right). \tag{3.5}$$

Cet estimateur converge en probabilité

$$\hat{\xi}_n \xrightarrow{p} -1/p,$$

où $p \in \mathbb{N}$ est la dimension de l'espace prédicteur.

Afin d'illustrer ce théorème, nous avons simulé quatre échantillons contenant respectivement 75, 150, 900 et 1500 données d'entraînement et ce, une première fois à partir de distributions normales bivariées et ensuite à partir de distributions normales multivariées à 4 variables. Nous avons ensuite, à partir d'un échantillon de test comprenant uniquement des données normales, calculé les estimateurs de Hill pour chaque observation selon la formule (3.5) avec $k = 20$. Pour chacun des espaces prédicteurs, les figures 3.6 et 3.7 montrent les distributions de cet estimateur pour chaque échantillon. Dans les deux cas, au plus la taille n de l'échantillon augmente, au plus la distribution est centrée sur $-1/p$ comme l'indique le théorème 1.

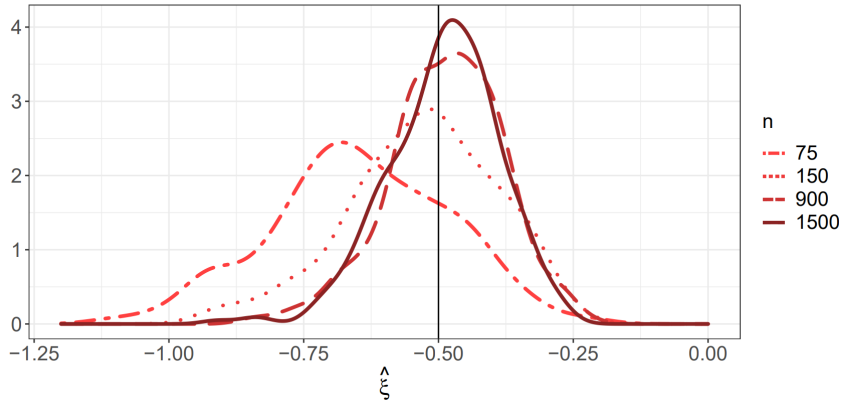

Figure 3.6 – Théorème 1 : $p = 2$

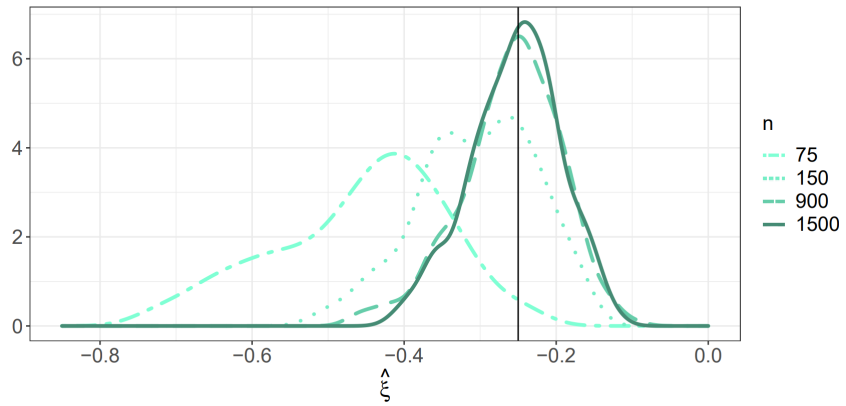

FIGURE 3.7 – Théorème 1 : $p = 4$

Dans le cas de l'hypothèse alternative d'anormalité, $\xi > -1/p$. En effet, la queue de la distribution est alors plus légère et de ce fait, les chances pour $-D$ de se retrouver proche de 0 sont moins grandes que dans le cas de l'hypothèse nulle.

Nous pouvons donc réaliser un test et classer x_0 comme anormal si $p\hat{\xi}$ est supérieur au seuil $s > -1$. Dans le cas contraire, nous ne pouvons pas encore être sûrs de la normalité du point et il faut réaliser des étapes supplémentaires. En effet, l'hypothèse nulle n'a pas été rejetée car x_0 se trouve dans le support de f mais cela ne prouve pas que le point est bel et bien normal.

Pour ce faire, il nous faut calculer le $(1-\gamma)$-quantile de $-D$ avec $0 < \gamma < k/n$ afin de vérifier si la région dans laquelle x_0 se trouve a la même densité que celle d'un point normal, γ étant la probabilité qu'un point se trouve à une certaine distance de x_0. Ce quantile s'obtient comme suit :

$$q_\gamma = R_{(n-k)}(n\gamma/k)^{-\hat{\xi}},$$

avec en général $\gamma = 1/n$. Si $-q_\gamma$ correspondant au rayon négatif de la boule autour de x_0, est significativement plus grand qu'un seuil $t > 0$, c'est-à-dire la quantité correspondante pour un point normal, x_0 est classé comme anormal car la densité autour de lui est trop faible. Sinon, x_0 est considéré comme normal.

Afin de déterminer les seuils s et t, nous allons exécuter l'algorithme sur chaque point de nos données d'entraînement en les considérant tour à tour comme étant l'inconnue x_0. Nous obtiendrons ainsi $\hat{\xi}_n^{(i)}$ et $-q_\gamma^{(i)}$ pour $i = 1, \ldots, n$ et nous fixons les seuils s et t aux $(1-\alpha/2)$-quantiles de respectivement $\hat{\xi}_n^{(1)}, \ldots, \hat{\xi}_n^{(n)}$ et $-q_\gamma^{(1)}, \ldots, -q_\gamma^{(n)}$. Ce choix de seuils permet de noter au plus une proportion α de données d'entraînement comme étant anormales.

Algorithm 1: GPDC

Inputs : A training set, x_0 the new point to classify, α a probability treshold and a constant k.

for *each training point* x_i **do**

  compute $-D_i = -\|x_i - x_0\|$;

end

Estimate $\hat{\xi}_n = \frac{1}{k}\sum_{i=1}^{k} \ln\left(\frac{R_{(n+1-i)}}{R_{(n-k)}}\right)$ using the biggest k negated distances $R_{(n)}, \ldots, R_{(n+1-k)}$;

Compute $q_\gamma = R_{(n-k)}(n\gamma/k)^{-\hat{\xi}}$;

for *each training point* x_i **do**

  Do the previous steps considering x_i as x_0 to obtain $\hat{\xi}_n^{(i)}$ and $-q_\gamma^{(i)}$;

end

Compute $s = (1-\alpha/2)$-quantile of $\hat{\xi}_n^{(1)}, \ldots, \hat{\xi}_n^{(n)}$;

Compute $t = (1-\alpha/2)$-quantile of $-q_\gamma^{(1)}, \ldots, -q_\gamma^{(n)}$;

if $p\hat{\xi}_n > s$ **then**

  x_0 is abnormal;

else if $-q > t$ **then**

  x_0 is abnormal;

else

  x_0 is normal;

end

3.3 Generalized Extreme Value Classifier

Le Generalized Extreme Value Classifier (GEVC) également introduit par Vignotto and Engelke (2020), se base sur les distributions des valeurs extrêmes généralisées (GEV) et plus particulièrement sur le calcul de la distance minimale entre deux points pour chaque point de l'échantillon :

$$D_i^{\min} = \min_{\substack{j=1\ldots n \\ j\neq i}} \|x_j - x_i\|.$$

Comme précédemment, nous prenons l'opposé de ces distances afin de pouvoir estimer leur distribution par une *EVW* qui modélise les maxima négatifs :

$$\begin{aligned} -D_i^{\min} &= -\min_{j\neq i} \|x_j - x_i\| \\ &= \max_{j:j\neq i}[-\|x_j - x_i\|]. \end{aligned}$$

Comme vu dans la section 3.1, l'équation (2.1) des distributions des valeurs extrêmes généralisées peut être réécrite pour le cas spécifique de Weibull de la façon suivante :

$$\mathcal{W}(z) = \exp\left(-\left(\frac{|z|}{\tau}\right)^{-\alpha}\right)$$

avec $\tau > 0$; $|z| = -z$; $z < 0$.

Afin de classer le nouveau point x_0 comme normal ou anormal, il faut en calculer la distance vers tous les autres points de l'échantillon d'entraînement et en retenir la distance minimale : $d_0^{\min} = \min_{j=1,\dots,n} \|x_j - x_0\|$. Le test d'hypothèse : H_0 : x_0 est normal ; H_1 : x_0 est anormal, peut ensuite être effectué. On rejettera l'hypothèse nulle si $d_0^{\min} > c$. Cela signifie en effet que le nouveau point et le point de l'échantillon qui en est le plus proche sont suffisamment éloignés, x_0 ne peut donc pas être considéré comme normal.

Pour définir cette valeur critique c, il faut résoudre $\mathbb{P}_0(d_0^{\min} > c) = \alpha_0$ en fixant le seuil d'erreur α_0 :

$$\begin{aligned}\hat{\mathbb{P}}_0(d_0^{\min} > c) &= \hat{\mathbb{P}}_0(-d_0^{\min} < -c) \\ &= \hat{\mathcal{W}}(-c) \\ &= \exp\left(-\left(\frac{c}{\hat{\tau}}\right)^{-\hat{\alpha}}\right) = \alpha_0.\end{aligned}$$

Le seuil c fournit par cette équation est : $c = \hat{\tau}(-\ln \alpha_0)^{-1/\hat{\alpha}}$.

Si x_0 est considéré comme normal car il est suffisamment proche des points de l'échantillon, on peut supposer que $-d_0^{\min}$ est approximativement un point de la distribution de $-D_i^{\min}$. Nous pouvons donc calculer c sur base des paramètres estimés $\hat{\tau}$ et $\hat{\alpha}$ de cette distribution et effectuer le test d'hypothèse.

Cet algorithme dont le pseudo-code se trouve ci-dessous, à l'avantage d'être rapide à mettre à jour avec l'apparition d'un nouveau point car il ne requiert que les calculs supplémentaires pour trouver la distance minimale entre ce nouveau point et les autres déjà présents.

Algorithm 2: GEVC

Inputs : A training set and x_0 the new point to classify.
for *each training point* x_i **do**
  `compute` $-D_i^{\min} = \max_{j:j\neq i}[-\|x_j - x_i\|]$;
end
Find $\hat{\tau}$ et $\hat{\alpha}$ with $\hat{\mathcal{W}}(-D_i^{\min})$;
Compute $c = \hat{\tau}(-\ln \alpha_0)^{-1/\hat{\alpha}}$;
Compute $d_0^{\min} = \min_{j=1,\dots,n} \|x_j - x_0\|$;
if $d_0^{\min} > c$ **then**
  x_0 `is abnormal`;
else
  x_0 `is normal`;
end

3.4 One-class Support Vector Machine

Dans cette section et la suivante, les algorithmes présentés ne reposent plus sur la théorie des valeurs extrêmes.

La méthode des Support Vector Machine (SVM) permet de faire de la classification dans de grandes dimensions. Le principe de base est de transformer un espace, appelé espace de représentation des données d'entrées, dans lequel il n'est pas possible d'établir une frontière linéaire entre les deux classes, en un espace dans lequel cela deviendra possible. Ce nouvel espace $\mathcal{F}$ s'appelle le "feature space". Cette transformation est réalisée à l'aide d'une fonction noyau, généralisation des produits scalaires des données qui vont nous renseigner sur l'emplacement des points les uns par rapport aux autres. La figure 3.8 ci-dessous montre la différence entre un problème de classification linéairement et non linéairement séparable dans deux dimensions.

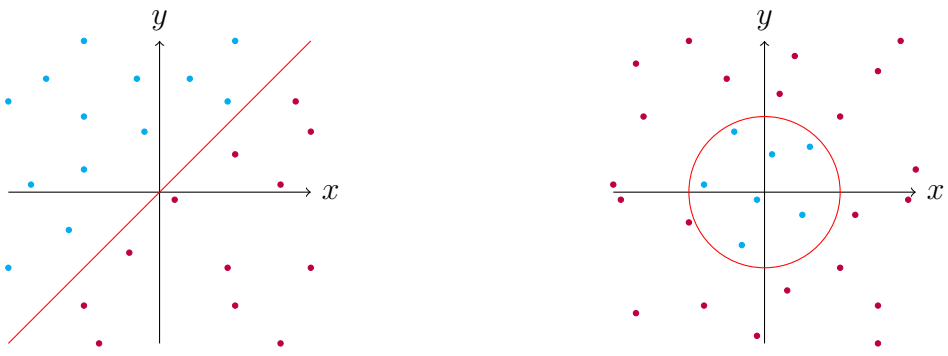

Figure 3.8 – Problèmes linéairement vs non linéairement séparables

La séparation linéaire entre les classes est réalisée à l'aide d'un hyperplan, c'est-à-dire, un sous-espace de dimension $p-1$. Son équation est :

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0. \tag{3.6}$$

Tout point $(x_1, \ldots, x_p)^\top$ dans un espace à p dimensions satisfaisant cette équation (3.6) se trouve sur l'hyperplan. Si ce n'est pas le cas, déterminer de quel côté de l'hyperplan se trouve le nouveau point x_0 repose simplement sur le fait de déterminer le signe du membre gauche de l'équation.

Si l'on dispose de n observations normales ou anormales, c'est-à-dire $y_i \in \{-1, 1\}$, afin de maximiser le nombre de points correctement classés, on cherche les coefficients β qui maximisent le nombre d'observations respectant cette inégalité :

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) > 0.$$

Dans le cas où les données sont parfaitement séparables, il existe une infinité d'hyperplans séparateurs. Il faut alors décider lequel choisir. Pour ce faire, nous choisissons celui qui est le plus éloigné des observations. La distance séparant l'observation la plus proche de l'hyperplan correspond à la marge qu'il faut maximiser. Ces points les plus proches s'appellent des vecteurs de support. L'hyperplan de marge maximale dépend donc uniquement de ces vecteurs et est résumé par le problème d'optimisation :

$$\max_{\beta_0, \beta_1, \ldots, \beta_p, M} M \tag{3.7}$$

$$\sum_{j=1}^{p} \beta_j^2 = 1, \tag{3.8}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \qquad \forall i = 1, \ldots, n, \tag{3.9}$$

où M correspond à la marge. Les contraintes (3.8) et (3.9) assurent que les points se trouvent du bon côté de l'hyperplan et ce à une distance minimale étant égale à la marge, pour peu que celle-ci soit positive.

Malheureusement, souvent il n'existe pas d'hyperplan permettant de séparer parfaitement les données et le fait que l'hyperplan de marge maximale repose uniquement sur les vecteurs de support pour le définir suppose que ce classifieur pourrait facilement faire du surapprentissage. Il est donc intéressant de permettre une marge d'erreur ϵ en laissant certaines observations se retrouver du mauvais côté de la marge ou de l'hyperplan. Dans notre problème d'optimisation, l'équation (3.9) devient $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i)$ avec $\epsilon_i \geq 0$ et $\sum_{i=1}^{n} \epsilon_i \leq C$, C contrôlant le compromis entre biais et variance et correspondant à la quantité d'erreurs tolérées. Si $C > 0$, il ne peut pas y avoir plus de C observations se trouvant du mauvais côté de l'hyperplan. Si $\epsilon_i = 0$, alors x_i est correctement classé.

Si $0 < \epsilon_i < 1$, bien que x_i soit bien classé, il se trouve du mauvais côté de la marge et si $\epsilon_i > 1$, le point se trouve du mauvais côté de l'hyperplan. Ceci est illustré à la figure 3.9.

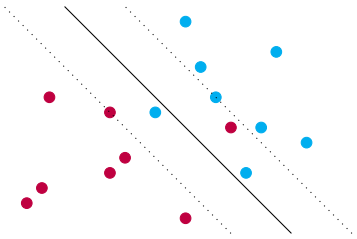

FIGURE 3.9 – SVM : Données du mauvais côté de la marge et de l'hyperplan

Les vecteurs de support correspondent maintenant aux observations se trouvant du mauvais côté de la marge en plus de celles se trouvant sur celle-ci.

Dans le cas où les frontières entre les classes ne sont pas linéaires, il nous faut agrandir l'espace de représentation des données d'entrées par l'utilisation de fonctions polynomiales de plus grand ordre des prédicteurs telles que des fonctions quadratiques ou cubiques. La frontière de décision dans cet espace de plus grande dimension devient alors linéaire. Cependant, il existe beaucoup de manières d'agrandir l'espace initial. Les SVM permettent l'utilisation de calculs efficaces via des fonctions noyaux que nous évoquions plus haut et il peut être montré que le problème d'optimisation (3.7)-(3.9) peut se résoudre grâce aux produits scalaires $\langle x_i, x_{i'} \rangle$ des observations.

La fonction de décision de x peut donc s'écrire :

$$f(x_0) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x_0, x_i \rangle.$$

Si une observation i n'est pas un vecteur de support, alors α_i vaut 0. Nous pouvons maintenant généraliser chaque produit scalaire grâce à une fonction noyaux $k(x_i, x_{i'})$. Il existe des noyaux de différentes formes : linéaire, polynomiale, etc. Les SVM utilisent des noyaux non-linéaires. La fonction de décision prend alors la forme :

$$f(x_0) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i k(x_0, x_i),$$

$\mathcal{S}$ correspondant à l'ensemble des indices des vecteurs de support. La figure 3.10 représente un SVM avec un noyau radial dont l'équation est :

$$k\left(x_i, x_{i'}\right) = \exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2\right).$$

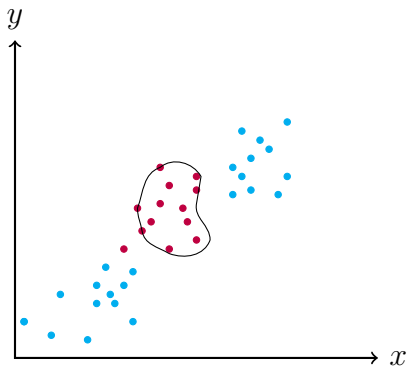

Figure 3.10 – SVM à noyau radial

Schölkopf et al. (1999) proposent d'appliquer les SVM sur une seule classe de données et de ce fait, rendent possible la détection d'anomalies avec cette méthode de classification. Le but est donc de séparer la région dans laquelle se trouve la majorité des données, de l'autre région. C'est-à-dire, séparer les points normaux prenant la valeur $+1$ des points anormaux prenant la valeur -1 et ce grâce à la fonction de décision f calculée par l'algorithme. Pour ce faire, les anomalies sont représentées par l'origine du feature space $\mathcal{F}$.

La fonction de décision $f : X \to \{-1, 1\}$ permet de déterminer la normalité d'un point $x \in X$. Cette fonction de décision s'écrit :

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i k\left(x, x_i\right) - \rho\right). \tag{3.10}$$

Le noyau peut s'écrire $k\left(x, y\right) = \langle \Phi(x), \Phi(y) \rangle$, ce qui correspond au produit scalaire de x et y dans l'espace imaginaire $\mathcal{F}$, Φ étant la transformation qui envoie le point x vers sa représentation $\Phi(x)$ dans cet espace. La somme présente dans l'équation (3.10) peut se réécrire $\langle \Phi(x), \sum_{i=1}^{n} \alpha_i \Phi(x_i) \rangle$ grâce à la linéarité du produit scalaire dans ses arguments. On pose $\sum_{i=1}^{n} \alpha_i \Phi(x_i) =: w$ qui correspond à un vecteur orthogonal à l'hyperplan. L'équation de l'hyperplan devient $\langle \Phi(x), w \rangle - \rho = 0$, où w en défini l'orientation et ρ défini la distance entre l'hyperplan et l'origine. Un example est visible à la figure 3.11.

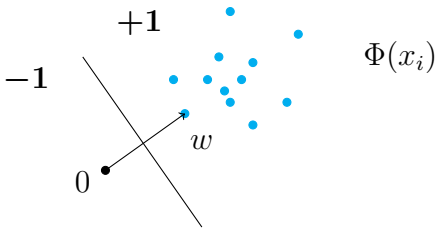

Figure 3.11 – One-class SVM

Afin de définir ces deux paramètres w et ρ, il faut trouver les points supports x_i tels que $\alpha_i > 0$ en résolvant le problème suivant :

$$\min \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k\left(x_i, x_j\right)$$

sous contraintes :

$$\sum_{i=1}^{n} \alpha_i = 1 \; ; \quad 0 \leq \alpha_i \leq \frac{1}{\nu n} \; ; \quad 0 < \nu < 1.$$

La solution à ce problème vaut $\frac{1}{2}\|w\|^2$, où w représente une combinaison convexe des points transformés de l'échantillon $\Phi(x_i)$. Notre problème devient un problème d'optimisation pour lequel il nous faut chercher la combinaison de points qui minimise $\|w\|$. Il s'agit du point le plus proche de l'origine qui se trouve dans le domaine, c'est-à-dire dans l'ensemble des points w possibles qui forment l'enveloppe convexe.

Dans le cas où 0 se trouverait dans le domaine des points de l'échantillon, nous aurions affaire à une situation peu intéressante. En effet, le point le plus proche de l'origine serait l'origine elle même et w vaudrait 0 ce qui implique que tous les points seraient catégorisés comme appartenant à la même classe. C'est dans le but d'éviter une telle situation qu'est nécessaire la condition permettant de borner les points $\alpha_i \leq 1/\nu n$. Jouer avec le paramètre ν va permettre de rétrécir plus ou moins la combinaison convexe w afin de sortir l'origine du domaine. Lorsque $\nu = 0$, aucune réduction de la combinaison n'est effectuée et lorsque $\nu = 1$, les vecteurs de support valent $1/n$. L'origine ne se trouvant plus dans le domaine, les données peuvent à nouveau être séparées de l'origine par l'hyperplan et la classification des points peut se faire correctement.

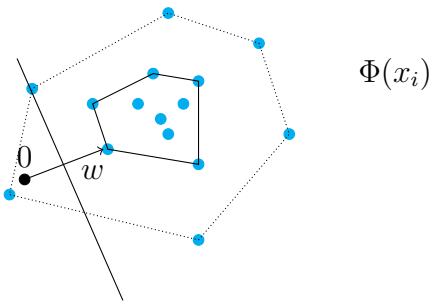

Figure 3.12 – One class SVM : 0 se trouve à l'intérieur du domaine

3.5 Isolation Forest

La dernière méthode que nous étudions est l'Isolation Forest (iForest) pour la détection d'anomalies, introduite par Liu et al. (2012). Cette méthode est assez différente des autres car au lieu de construire un profil des points normaux pour ensuite identifier les points qui ne s'y conforment pas, elle consiste à isoler les données trop différentes des points normaux et à les considérer comme des anomalies. En effet, l'un des désavantages des précédentes méthodes de détection d'anomalies est leur complexité de calculs nécessitant le calcul de beaucoup de distances.

Le principe de l'iForest est de générer aléatoirement plusieurs arbres de décision sur des sous-échantillons des données d'entraînement. En effet, détecter les anomalies via l'isolation est plus efficace sur des petits échantillons. Le fait de construire une forêt va donc permettre de garder cette efficacité tout en gardant la possibilité d'utiliser de grands échantillons. Afin d'illustrer ceçi, nous avons généré des données comprenant une classe de 2000 points normaux construits à partir d'une distribution normale bivariée et deux classes de 50 points chacune construits à partir de deux autres distributions normales bivariées. Ces points sont représentés à la figure 3.13. Ce graphe montre bien que certaines données anormales se superposent aux données normales, complexifiant de ce fait le processus de détection d'anomalies. La figure 3.14 montre un sous-échantillon de ces données avec seulement 200 points normaux pour 10 anomalies. On remarque immédiatement que ces derniers sont bien plus clairement identifiables. Les forêts d'isolation fonctionnent donc bien mieux avec des échantillons de petites tailles contrairement aux autres méthodes.

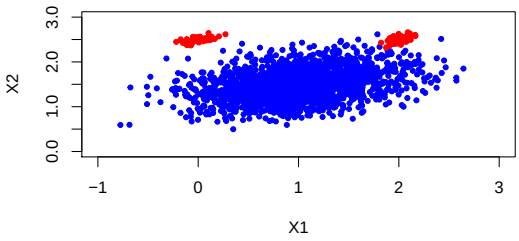

Figure 3.13 – Données originales

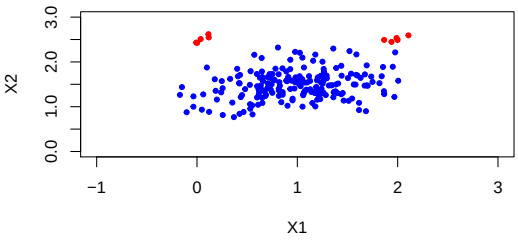

Figure 3.14 – Sous-échantillon

La construction d'arbres de décision va permettre d'isoler chaque point du sous-échantillon en partitionnant les données de manière récursive. Cette méthode à l'avantage de pouvoir fonctionner avec des modèles partiels car les anomalies correspondent aux points les plus proches de la racine de l'arbre. En effet, un point normal nécessitera beaucoup plus de partitions pour être isolé des autres qu'une

anomalie qui est moins proche de ces points. Ce nombre de partitions correspond à la longueur du chemin de la racine à la feuille correspondant au point en question. Le fractionnement se fait de manière aléatoire comme représenté ci-dessous. La figure 3.15 reflète bien le fait qu'un point normal nécessite plus de partitions qu'un point anormal visible à la figure 3.16.

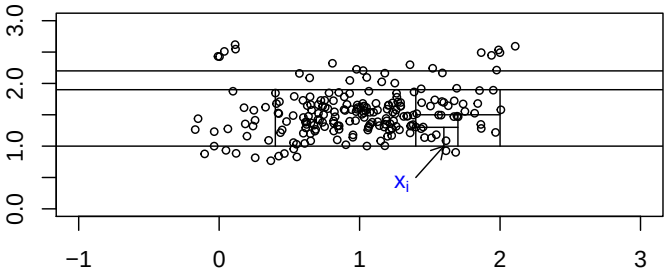

Figure 3.15 – Isolation d'un point normal

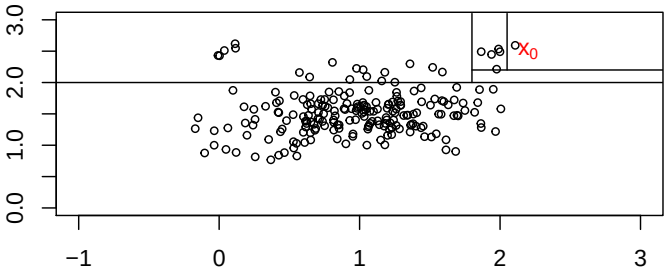

Figure 3.16 – Isolation d'une anomalie

L'arbre ne se basant aléatoirement que sur une partie de l'échantillon, il faut en réaliser plusieurs afin de prendre en compte toutes les données, d'où la forêt. Liu et al. (2012) ont trouvé de manière empirique, qu'il n'est en général pas nécessaire de disposer de sous-échantillons de plus de 256 données pour faire de la détection d'anomalie efficace.

Un score d'anomalie s est ensuite construit pour chaque point sur base de la moyenne de la longueur de son chemin nécessaire pour être isolé dans chaque arbre. Il peut être montré que cette moyenne converge avec l'augmentation du nombre d'arbres. A nouveau, Liu et al. (2012) ont montré empiriquement que cette convergence se fait en général avant d'atteindre 100 arbres. Ce score d'anomalie va ensuite être utilisé pour classer le point. Il se calcule de la manière suivante :

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}, \tag{3.11}$$

avec $h(x)$ la longueur du chemin de la racine à la feuille, $E(h(x))$ sa moyenne empirique et $c(n)$ sa moyenne définie par Preiss (1999) comme étant :

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n},$$

où $H(i) = \sum_{k=1}^{i} k^{-1}$ est le i-ième nombre harmonique qui peut être estimé par $H(i) = \ln(i) + e$ avec la constante d'Euler $e \approx 0.577$.

Nous pouvons déduire de l'équation (3.11) que lorsque $E(h(x)) \to c(n)$ alors $s \to 0.5$, lorsque $E(h(x)) \to 0$ alors $s \to 1$ et enfin, lorsque $E(h(x)) \to n-1$ alors $s \to 0$.

De plus, Liu et al. (2012) établissent que si s est proche de 1, le point correspondant à ce score est une anomalie ; si s est beaucoup plus petit que 0.5, la donnée peut être considérée comme normale et si tous les points ont un score $s \approx 0.5$, l'échantillon ne semble pas contenir d'anomalie. Nous avons représenté ceci à la figure 3.17 grâce aux mêmes données qui ont servi aux illustrations des vecteurs extrêmes à la section 3.1 et du théorème 1 à la section 3.2.

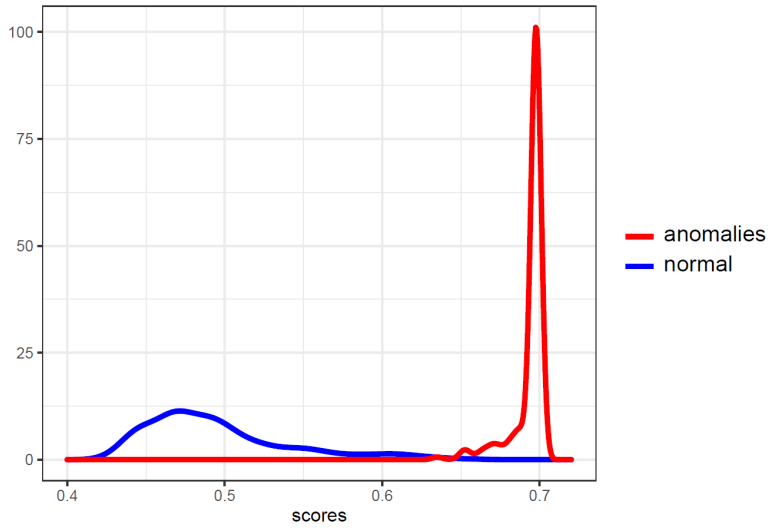

FIGURE 3.17 – iForest : classification en fonction du score

Si l'on désire trouver les m premières anomalies, il suffit d'ordonner les données selon leur score de manière décroissante et de sélectionner les m premières.

Chapitre 4

Analyses

Dans ce chapitre, nous comparons les différents algorithmes sur quatre bases de données distinctes. Nous commençons par simuler des données sur lesquelles nous comparons la précision des trois premières méthodes. A savoir l'EVM, le GPDC et le GEVC. Nous mettons également en évidence une faiblesse du SVM. Nous testons ensuite les cinq algorithmes sur la base de données réelles LETTER de Frey and Slate (1991) en suivant le protocole OLETTER introduit par Bendale and Boult (2015). Pour finir, ces algorithmes, excepté l'EVM, seront testés sur les bases de données médicales de Quinlan et al. (1987) et Mangasarian et al. (1995) portant sur la détection d'une maladie de la thyroïde et sur celle du cancer du sein. Pour rappel, le but de ces analyses est de montrer l'efficacité des méthodes de détection d'anomalies basées sur la théorie des valeurs extrêmes comparé au SVM et à l'iForest.

4.1 Données simulées

Nous évaluons ici la performance des trois méthodes basées sur la théorie des valeurs extrêmes, à savoir l'EVM, le GPDC et le GEVC. Pour ce faire, nous commençons par simuler un jeu de données d'entraînement comprenant 600 observations et 3 classes toutes construites à partir de distributions normales bivariées différentes. Ces données considérées comme normales sont visibles en noir à la figure 4.1. L'ensemble de test comprend 800 observations dont 600 données normales visibles en jaune et construites à partir des mêmes distributions que les données d'entraînement. Les 200 données restantes sont construites à partir d'une quatrième distribution bivariée et sont donc anormales. Ces dernières sont représentées en orange.

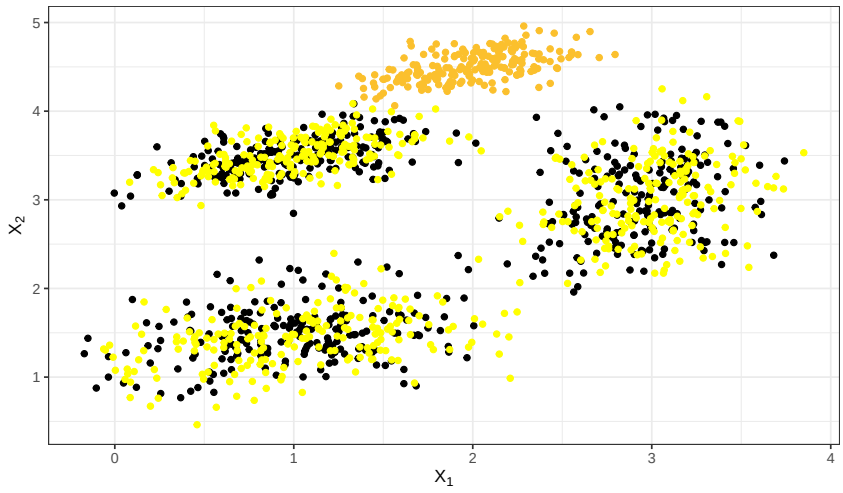

Figure 4.1 – Données simulées : training set (en noir) vs testing set (données normales en jaune, anormales en orange)

Afin de tester la performance des classifieurs, nous regardons l'aire se situant sous la courbe ROC signifiant "Receiver Operating Characteristic". Cette courbe permet de visualiser le compromis entre la sensitivité (taux de vrais positifs) et la spécificité (taux de vrais négatifs) d'un test pour chaque valeur comprise entre 0 et 1. Ces taux se calculent comme suit :

$$\text{sensitivité} = \frac{\text{nombre de vrais positifs}}{\text{nombre de vrais positifs} + \text{nombre de faux négatifs}}$$

et

$$\text{spécificité} = \frac{\text{nombre de vrais négatifs}}{\text{nombre de vrais négatifs} + \text{nombre de faux positifs}}.$$

Elle permet donc de visualiser les deux types d'erreurs et prend en compte tous les seuils de classification possibles. Si l'on souhaite obtenir une plus grande sensitivité, il faut sacrifier de la spécificité. Un exemple de courbe ROC est visible à la figure 4.2 pour la performance du GEVC sur les données simulées. The Area Under the ROC Curve (AUC) correspond au pouvoir discriminant du modèle testé, c'est-à-dire sa performance globale. Plus concrètement, le AUC résume toutes les combinaisons possibles de spécificité et sensitivité. Au plus cette aire est grande, au plus le classifieur est performant. Un AUC de 1 signifie donc que le modèle testé à une précision parfaite. Au contraire, un AUC de 0.5 signifie que le modèle a le même pouvoir prédicteur qu'un classifieur aléatoire.

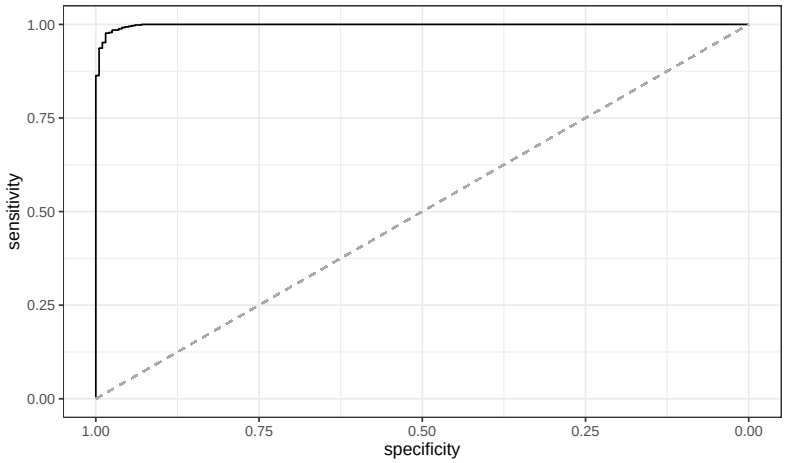

Figure 4.2 – Courbe ROC - GEVC

Le GPDC et le GEVC obtiennent tous deux de très bons résultats sur les données simulées avec respectivement des AUC de 0.999 pour $k = 20$ et 0.998, ce qui est proche de ce qu'obtiennent Vignotto and Engelke (2020) avec respectivement 0.997 et 0.999. Précisons que les résultats du GPDC restent stables pour différentes valeurs du paramètre k. Pour $k = 5$, nous avons en effet obtenu un AUC de 0.995 et pour $k = 50$, un AUC de 0.999. La précision de l'EVM dépend des paramètres choisis. Ainsi, en jouant avec ceux-ci, nous avons obtenus des précisions allant de 0.809 à 0.999. Les combinaisons utilisées pour ces résultats sont visibles à l'annexe A.1. Or, Vignotto and Engelke (2020) annoncent un score de seulement 0.853. Ils ne sont cependant pas clairs quant à leur façon de fixer certains paramètres tel que le seuil ζ. On remarque d'ailleurs que les meilleurs résultats sont obtenus lorsque ce seuil est fixé à 0.5 comme suggéré par Rudd et al. (2018). De plus, si l'on compare les données simulées de Vignotto and Engelke (2020) à la figure 4.3, on voit directement que leur classe d'anomalies se trouve très proche de leurs classes normales contrairement à nos anomalies qui sont beaucoup plus éloignées des points normaux. Ceci pourrait également expliquer cette différence de précision dans l'hypothèse où ils auraient tout de même utilisé un seuil ζ de 0.5. Rappelons en effet que l'EVM se base sur la distance entre les classes contrairement aux deux autres algorithmes.

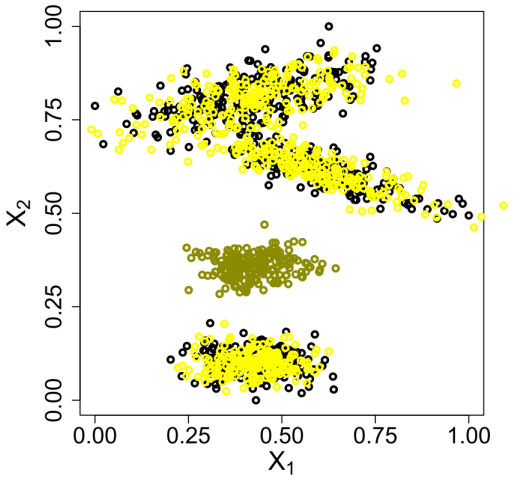

FIGURE 4.3 – Données simulées de Vignotto and Engelke (2020)

Ajoutons que la figure 4.4 confirme le théorème 1 selon lequel l'estimateur de Hill converge vers $-1/p$ soit $-1/2$ ici, lorsque le point x_0 est normal et se trouve à l'intérieur du support des classes normales. En effet, les estimateurs correspondant aux points normaux se retrouvent dans le pic de gauche aux alentours de -0.5 alors que ceux des points anormaux sont situés dans le pic de droite proche de 0 ce qui confirme la théorie affirmant que ξ est plus grand que $-1/p$ dans le cas de l'hypothèse alternative d'anormalité.

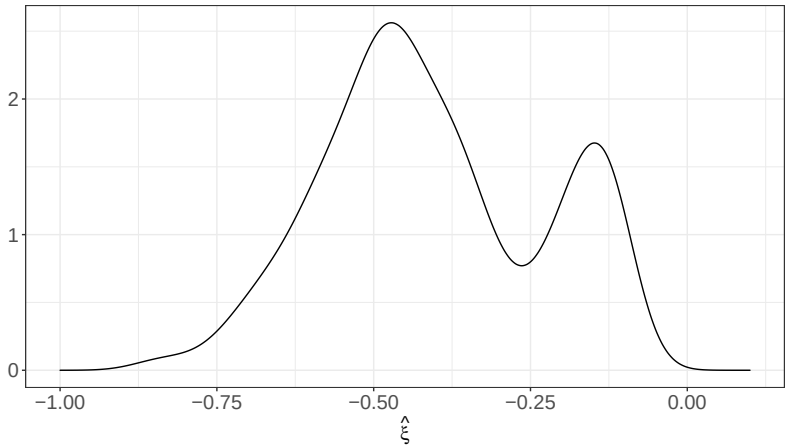

FIGURE 4.4 – Estimations des $\hat{\xi}$ pour les données simulées

Outre les méthodes basées sur la théorie des valeurs extrêmes, nous avons également testé le SVM afin de montrer à quel point cette méthode est sensible à

ses paramètres. Pour ce faire, nous avons considéré 90 combinaisons des hyperparamètres. Ces différents modèles ainsi que leur AUC respectif sont disponibles à l'annexe A.2. La figure 4.5 montre clairement que la précision du SVM varie en fonction du choix du modèle. Il est donc comme pour l'EVM, important de bien sélectionner ces paramètres.

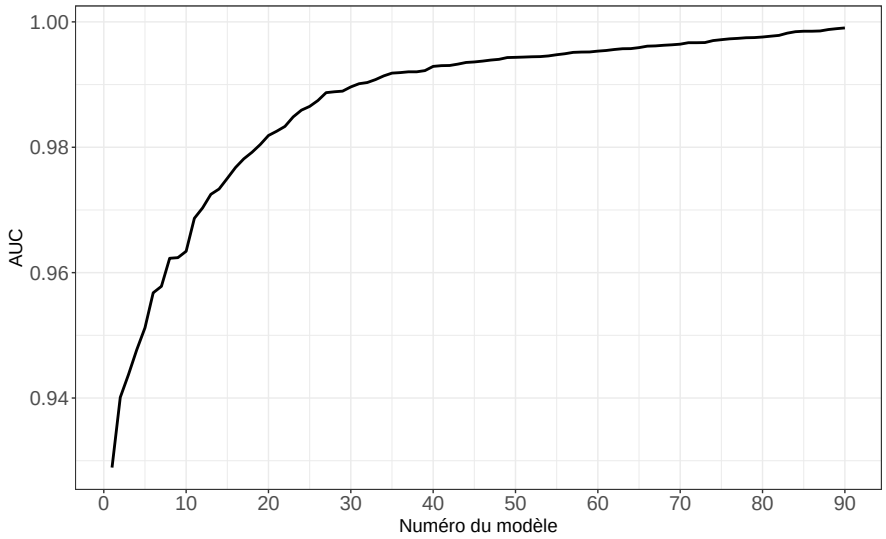

Figure 4.5 – AUC en fonction des combinaisons des paramètres

4.2 Protocole OLETTER

Le protocole OLETTER est une méthode d'évaluation de la performance pour les systèmes de classification qui fut introduit par Bendale and Boult (2015) sur base des données LETTER de Frey and Slate (1991). Cette base de données contient 20000 observations classées parmis 26 catégories représentant les 26 lettres de l'alphabet. Ces observations correspondent chacune à une lettre écrite à la main. Chaque observation dispose de 16 variables en plus de sa classe.

Afin de comparer la performance des cinq algorithmes vus précédemment, nous utilisons le protocole OLETTER qui consiste à continuellement ajouter des classes au modèle tandis que celui-ci est testé avec des classes inconnues.

La phase d'entraînement du protocole consiste à ajuster un modèle aux données d'apprentissage contenant uniquement un ensemble fixe de classes connues. Les classes inconnues restantes de l'ensemble d'apprentissage sont ensuite ajoutées une par une dans le modèle, tout en gardant les paramètres estimés fixes.

Lors de la phase de test, il nous faut séparer l'ensemble de test en deux, une partie contenant les classes connues et l'autre les classes inconnues. A chaque étape,

le modèle est évalué sur chacune de ces deux parties et nous répétons ce processus tant que l'on ajoute des classes au modèle. Le processus entier est ensuite répété plusieurs fois sur des pans de la base de données à la façon d'une validation croisée.

Dans notre cas, nous allons séparer la base de données en un ensemble d'apprentissage contenant 15000 observations et un ensemble de test en contenant 5000. Nous sélectionnons ensuite 15 classes de manière aléatoire que nous considérons comme normales et sur lesquelles nous entraînons le modèle. Nous commencerons ensuite par tester le modèle sur l'ensemble de test dans lequel nous aurons gardé uniquement les 15 classes normales ainsi qu'une classe anormale. Nous ajouterons ensuite les classes anormales restantes une par une en testant le modèle entre chaque ajout et en mettant le seuil de classification à jour avec le seuil δ (3.3) proposé par Rudd et al. (2018). Vignotto and Engelke (2020) ont ensuite répété tout le processus 20 fois pour faire une moyenne des précisions obtenues. En raison de la lenteur du processus, nous n'avons effectué le processus qu'une seule fois après nous être assuré sur des sous-échantillons que la variance des précisions d'une itération à l'autre soit très faible (annexe B). Ceci est d'ailleurs confirmé par le graphe 4.7 de Vignotto and Engelke (2020) sur lequel on aperçoit les intervalles de confiance des précisions en pointillés de part et d'autre de chaque ligne.

Comme Vignotto and Engelke (2020), nous utiliserons la mesure F comme critère de performance pour comparer les algorithmes. Ce score allie précision et rappel :

$$F_{\text{mesure}} = 2\,\frac{\text{précision} \times \text{rappel}}{\text{précision} + \text{rappel}},$$

avec

$$\text{précision} = \frac{\text{nombre de points anormaux correctement classés}}{\text{nombre de points classés comme anormaux}}$$

et

$$\text{rappel} = \frac{\text{nombre de points anormaux correctement classés}}{\text{nombre de points anormaux}}.$$

La figure 4.6 montre les résultats du processus que nous avons obtenus pour les cinq algorithmes.

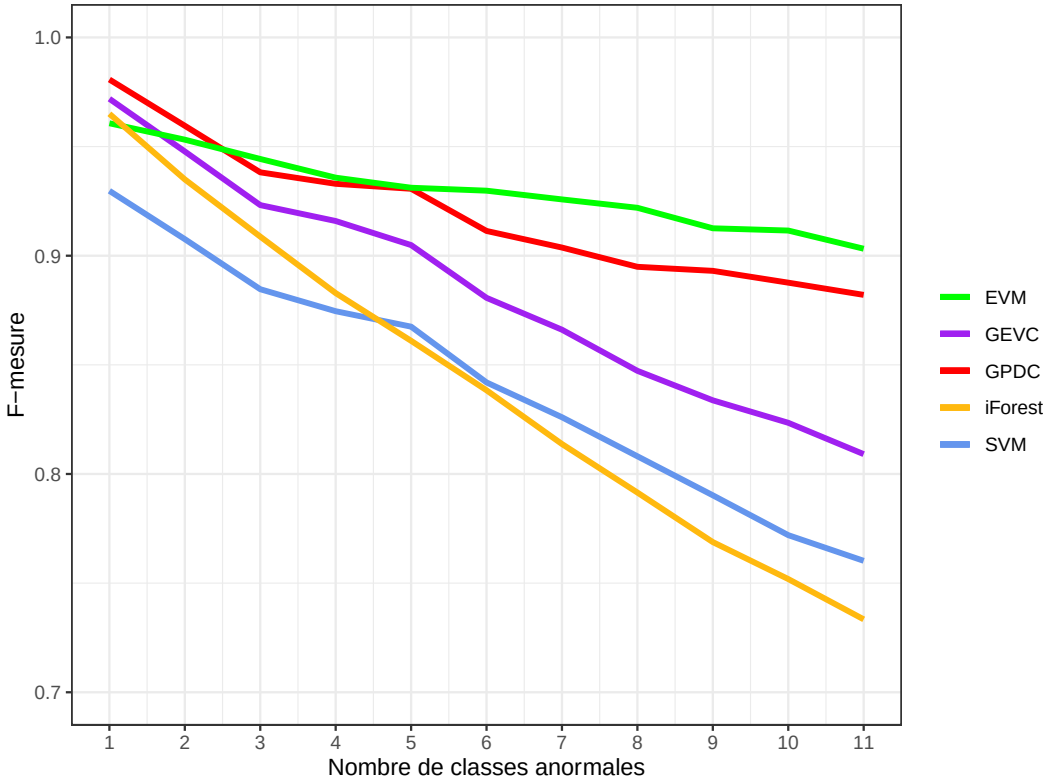

Figure 4.6 – Résultats du protocole OLETTER

Ce graphe peut être comparé à la figure 4.7 qui représente les résultats du protocole obtenus par Vignotto and Engelke (2020). Bien que les tendances des méthodes soient similaires, certaines différences apparaissent. La précision du GEVC chute légèrement plus vite dans notre cas. Elle reste cependant meilleure que l'iForest et le SVM. C'est également le cas du GPDC pour lequel nous avons fixé le paramètre k à 22, ce qui correspond à peu près à l'utilisation des 0.25% des distances les plus petites. Le fait que le GEVC et le GPDC aient des précisions légèrement moins bonnes qu'attendu provient probablement d'une utilisation différente du seuil δ dans le protocole. En effet, Vignotto and Engelke (2020) expliquent utiliser ce seuil sur quatre des algorithmes sans être clairs quant à son utilisation exacte. Leur code n'étant pas disponible, nous n'avons pas pu reproduire leurs expériences exactement de la même manière. Nous ne pouvons donc pas espérer obtenir les mêmes résultats au chiffre près. Malgré cela, ces deux algorithmes restent meilleures que l'iForest et le SVM, et le GPDC est en concurrence avec l'EVM dans ce cas-ci.

Pour ce qui est de la précision du One-class SVM, elle semble au départ moins bonne que celle obtenue par Vignotto and Engelke (2020) bien qu'elle chute légèrement moins vite et devient similaire sur la fin. Le SVM étant sensible à

ses paramètres comme montré à la section précédente, ceux-ci sont probablement ajustés de manière légèrement différente, ce qui expliquerait cette différence de précision.

Pour finir, l'EVM et l'iForest semblent agir de manière similaire que pour Vignotto and Engelke (2020). Cette dernière méthode ne nécessite que peu de paramètres pour lesquels nous avons, tout comme Vignotto and Engelke (2020) suivi les recommandations de l'auteur Liu et al. (2012). c'est-à-dire, 100 arbres et des sous-échantillons de taille 256. Bien que cette méthode à l'avantage d'être rapide car elle ne nécessite pas le calcul des distances entre les points, il s'agit du classifieur dont la précision chute le plus rapidement avec l'arrivée de nouvelles classes anormales. Quant à l'EVM il s'agit du classifieur le plus stable. Son paramètre k à été fixé à 75 comme spécifié dans le papier original de Rudd et al. (2018).

En conclusion, bien que les méthodes basées sur l'analyse des valeurs extrêmes nécessitent beaucoup de calculs, il est clair qu'elles sont plus efficaces que les autres et ce malgré l'utilisation de moins de paramètres, voir d'aucun dans le cas du GEVC.

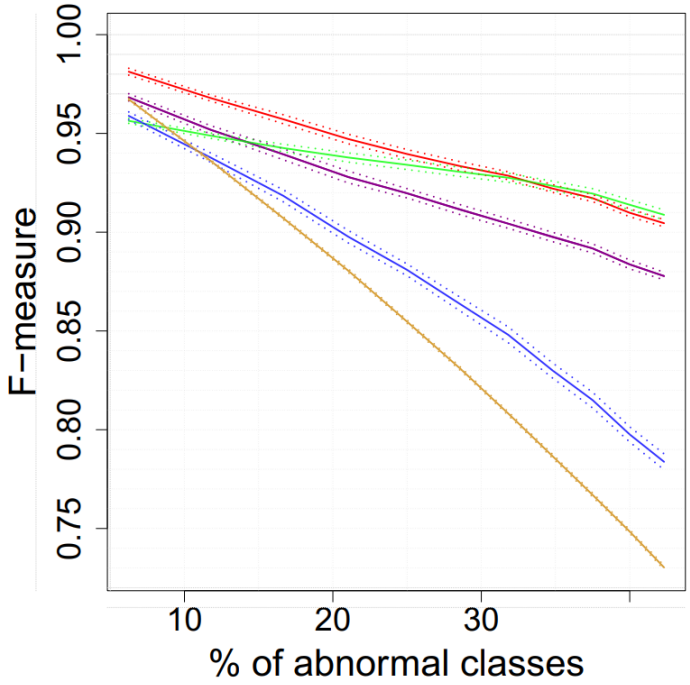

Figure 4.7 – Résultats du protocole OLETTER réalisé par Vignotto and Engelke (2020) : EVM (vert), GEVC (mauve), GPDC (rouge), iForest (jaune), SVM (bleu).

4.3 Diagnostics de maladies

Afin de tester une dernière fois nos algorithmes, nous les avons appliqués sur deux bases de données portant sur le diagnostic d'une maladie. La première porte sur l'hypothyroïdie, une maladie de la thyroïde, dont les données proviennent de Quinlan et al. (1987) et la deuxième provient de Mangasarian et al. (1995) et porte sur le diagnostic du cancer du sein qui représente 36% des cancers féminins Espié (2007). Ces bases de données sont toutes deux disponibles dans le répertoire *UCI Machine Learning Respository* Dua (2017).

Dans les deux cas, les données normales n'appartenant qu'à une seule classe correspondant aux patients sains, il ne sera pas possible d'y appliquer l'EVM sachant que cet algorithme repose sur le calcul de la distance entre les classes et nécessite de ce fait la présence de plusieurs classes normales. Il ne sera donc pas testé dans cette section. Nous comparerons uniquement les quatre méthodes restantes sur base du AUC comme à la section 4.1.

4.3.1 Maladie de la thyroïde

Cette base de données concernant le diagnostic de la maladie de l'hypothyroïdie contient initialement 3772 observations de 30 variables. Nous retirons 3 variables composées uniquement de données manquantes ou identiques. Certaines observations contenant trop de données manquantes sont également retirées et nous obtenons un total de 3622 observations dont 280 anomalies. Celles-ci ainsi que 280 données normales, choisies aléatoirement dans la base de données, composeront l'ensemble de test.

Afin d'obtenir le meilleur résultat possible pour le GPDC, nous l'avons testé avec cinq valeurs de k différentes correspondant à peu près à l'utilisation des $0.5\%, 1\%$, 2.5%, 5% et 10% des plus petites distances. C'est-à-dire, $k = 15, 30, 75, 150, 300$. Ces pourcentages correspondent à des choix communs dans le cadre de l'application de la théorie des valeurs extrêmes. La table 4.1 confirme ce que nous avions déjà observé dans le cas des données simulées à la section 4.1, c'est-à-dire que le GPDC est très peu sensible à son paramètre. Nous retenons $k = 75$ qui fournit la meilleure précision.

$k = 15$	$k = 30$	$k = 75$	$k = 150$	$k = 300$
0.9333	0.9356	0.936	0.9115	0.8836

Table 4.1 – Thyroïde : AUC pour k distances utilisées pour le GPDC

Pour le SVM, nous avons considéré différentes combinaisons de ses paramètres visibles à l'annexe C.1. Celle fournissant le meilleur résultat nous permet d'atteindre un AUC de 0.903.

Enfin, pour le choix des paramètres de l'iForest, ceux conseillés par Liu et al. (2012) ne fournissant pas le meilleur modèle dans ce cas-ci, nous avons considéré plusieurs autres combinaisons disponibles à l'annexe C.2. Le meilleur résultat obtenu est un AUC de 0.711.

La figure 4.8 permet de visualiser les courbes ROC pour chacun des modèles choisis. On voit immédiatement que les classifieurs GPDC, GEVC et SVM obtiennent les meilleurs résultats avec des AUC allant de 0.90 à 0.94. La performance de l'iForest est quant à elle bien moins bonne avec un AUC de seulement 0.71.

Les résultats sont donc favorables aux deux méthodes basées sur la théorie des valeurs extrêmes ainsi qu'au SVM dans ce cas-ci. Soulignons également que le GPDC à l'avantage de montrer une performance plus stable que le SVM pour différents paramètres.

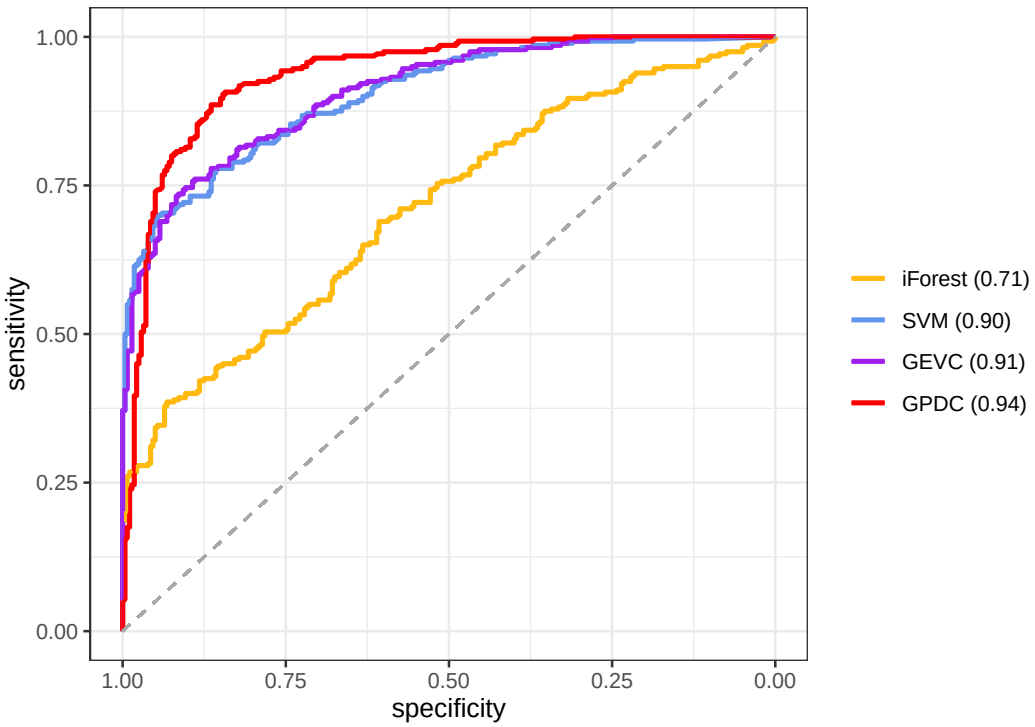

Figure 4.8 – Courbes ROC - thyroïde

4.3.2 Cancer du sein au Wisconsin

Cette base de données sur le diagnostic du cancer du sein au Wisconsin contient 569 observations de 31 variables dont la variable réponse. Parmi ces observations, 212 patientes sont atteintes de la maladie et feront partie des données de test accompagnés de 57 patientes saines choisies au hasard dans la base de données. De cette sélection, il reste 300 observations qui serviront à entraîner les différents modèles.

Afin de comparer nos quatre algorithmes, nous avons suivis le même procédé que dans la section précédente. Nous avons choisi les meilleurs modèles parmi plusieurs possibles et nous les avons ensuite comparé sur base du AUC.

Pour le GPDC, nous avons testé quatre valeurs de k différentes correspondant à peu près à l'utilisation des 1%, 2.5%, 5% et 10% des plus petites distances. C'est-à-dire, $k = 3, 7, 15, 30$. Les résultats obtenus sont visibles dans la table 4.2 et permettent de tirer la même conclusion que précédemment, c'est-à-dire que le GPDC est très peu sensible à son paramètre. Nous retenons $k = 30$ qui fournit la meilleure précision.

$k = 3$	$k = 7$	$k = 15$	$k = 30$
0.9963	0.9961	0.9961	0.9964

Table 4.2 – Cancer : AUC pour k distances utilisées pour le GPDC

Pour le SVM, nous avons considéré différentes combinaisons de ses paramètres visibles à l'annexe D.1. Celle fournissant le meilleur résultat nous permet d'atteindre un AUC de 0.9975.

Enfin, pour le choix des paramètres de l'iForest, nous avons considéré uniquement ceux recommandés par Liu et al. (2012).

La figure 4.9 permet de visualiser les courbes ROC pour chacun des modèles choisis. On voit immédiatement que les GPDC, GEVC et SVM obtiennent les meilleurs résultats avec des AUC très proches de 1. La performance de l'iForest est quant à elle légèrement moindre mais reste comparable à ces méthodes. Les résultats sont comme précédemment, favorables aux deux méthodes basées sur la théorie des valeurs extrêmes ainsi qu'au SVM, bien que l'iForest montre ici une bonne performance.

La conclusion reste donc la même pour les deux bases de données, cependant les résultats sont moins flagrant pour la deuxième et il est intéressant de voir comment de mêmes algorithmes peuvent obtenir des performances variant avec les données.

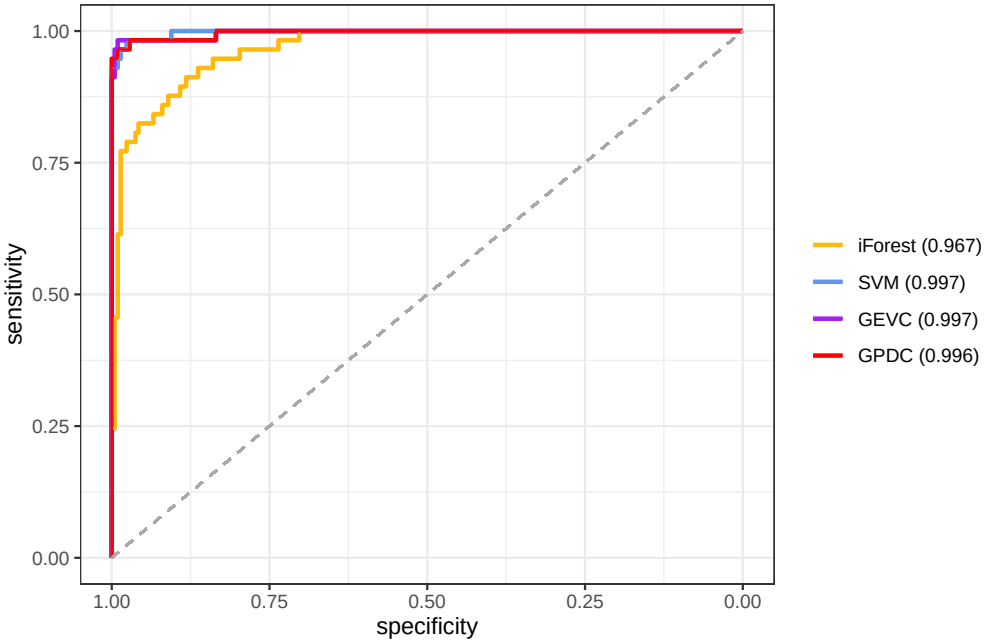

Figure 4.9 – Courbes ROC - cancer du sein

Chapitre 5

Limites

Dans ce chapitre nous allons essayer de mettre le GEVC et le GPDC en difficulté afin d'en trouver les limites. Nous commençons par étudier la réaction de ces algorithmes lorsque les classes de données ne sont pas parfaitement séparées. Nous verrons ensuite si le fait d'utiliser des données mal balancées influence le résultat. Ces tests seront effectués sur des données simulées.

5.1 Classes mal séparées

Afin d'évaluer la précision du GEVC et du GPDC lorsque les classes anormales sont mal séparées des classes normales, nous avons simulé le même ensemble d'entraînement qu'utilisé dans la section 4.1 du chapitre 4 (figure 4.1). Nous avons également repris les mêmes 600 données normales pour la phase de test. En revanche, les 200 données anormales de test sont nouvelles et proviennent, comme le reste des observations, d'une distribution normale bivariée. La figure 5.1 montre en noir l'ensemble d'entraînement et en couleurs la classe d'anomalies se situant plus ou moins proche des classes normales.

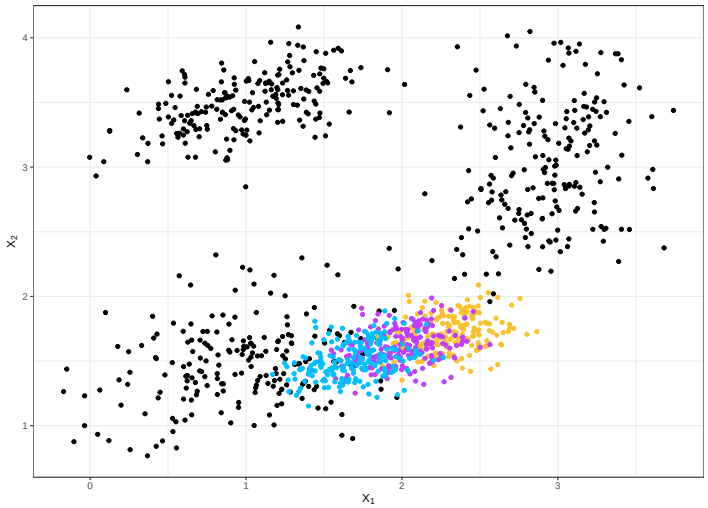

Figure 5.1 – Données simulées : classes mal séparées

Le paramètre k du GPDC a été fixé à 6 ce qui correspond à l'utilisation des 1% des distances les plus proches.

Les figures 5.2-5.4 montrent les courbes ROC pour les classes orange, mauve et bleue respectivement, c'est-à-dire pour la classe la plus éloignée à la plus proche des données. Ces graphes montrent distinctement la diminution de l'aire sous la courbe ROC pour le GEVC en mauve et le GPDC en rouge jusqu'à devenir pratiquement aussi inefficace qu'un classifieur aléatoire pour le GPDC.

Nous avons ensuite comparé ces méthodes à un modèle de SVM qui permet de faire de la détection d'anomalie notamment dans le cas de données mal séparées (voir figure 3.10 pour un rappel). Le modèle est un One-class SVM à noyau radiale dont les paramètres sont ceux donnant le meilleur résultat pour la classe mauve parmi plusieurs combinaisons possibles visibles à l'annexe E.1. Nous avons gardé les mêmes paramètres pour les classes orange et bleue, bien que ce ne soit pas forcément ceux qui maximisaient la précision de ces deux classes.

Alors que la précision du SVM semble compétitive avec les deux premiers algorithmes lorsque les classes sont bien séparées, on voit que malgré une légère diminution avec le rapprochement des classes, il reste bien plus efficace que le GPDC et le GEVC.

On voit donc une première limite aux algorithmes de Vignotto and Engelke (2020) qui sont assez efficaces lorsque les classes sont bien séparées mais dont la précision chute dès qu'elles se superposent un peu trop.

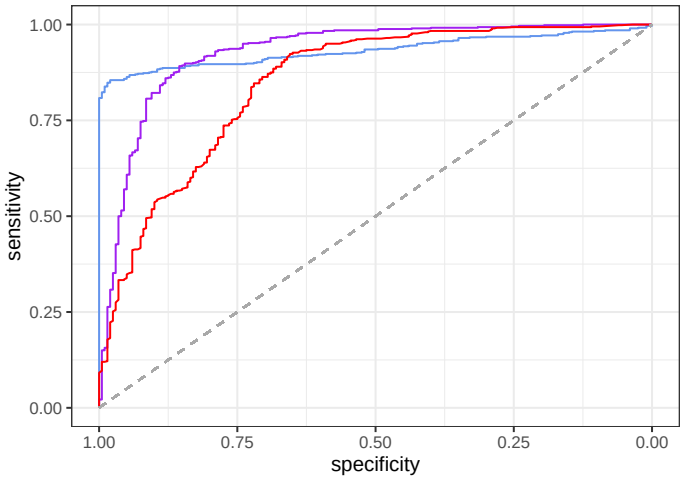

Figure 5.2 – ROC : classe orange

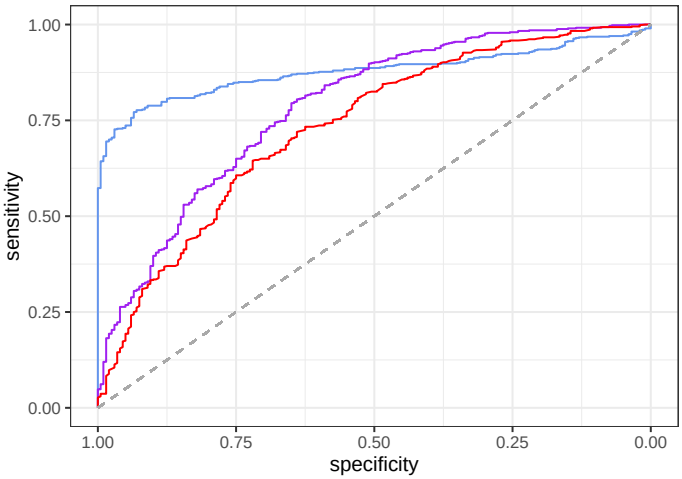

Figure 5.3 – ROC : classe mauve

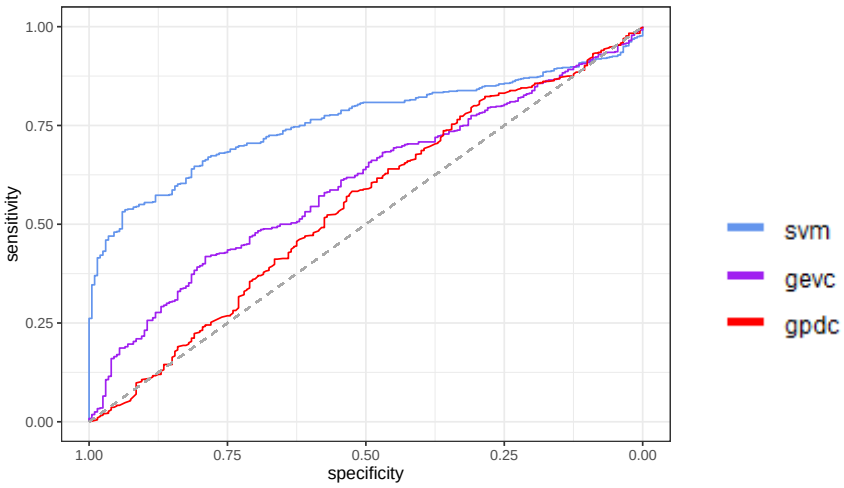

Figure 5.4 – ROC : classe bleue

5.2 Données mal balancées

Cette section étudie le potentiel impact de l'utilisation de données mal balancées. Pour ce faire, nous avons simulé quatre ensembles d'entraînement différents contenant chacun les deux mêmes classes de données provenant de distributions normales bivariées en proportions différentes. Le premier ensemble est parfaitement balancé avec 400 observations pour chaque classe. Le second contient 400 et 130 observations, ce qui correspond à peu près à 75/25% des données. Le suivant ne contient plus que 400 contre 20 observations, ce qui correspond à 95/5% des données et enfin le dernier contient 400 et 4 données de chaque classe ce qui correspond à 99/1% des données. Les quatre phases de test ont été réalisées avec le même ensemble de test, c'est-à-dire 300 données balancées provenant des deux classes normales et d'une classe anormale. Ces données sont visibles à l'annexe E.2.

Les figures 5.5 et 5.6 ci-dessous montrent les courbes ROC pour chaque modèle avec l'utilisation du GEVC à gauche et du GPDC à droite. Le paramètre k du GPDC a été modifié en fonction de la quantité de données dans le modèle afin de ne prendre en compte qu'environ un pourcent des distances les plus courtes. Ceci correspond à des valeurs de $8, 5, 4$ et 4 respectivement. On voit clairement que la précision des deux algorithmes n'est pas impactée par le changement de balancement des classes. Les méthodes de Vignotto and Engelke (2020) restent donc efficaces dans ce cas-ci. Des zooms des parties intéressantes de ces deux graphes sont disponibles à l'annexe E.2.

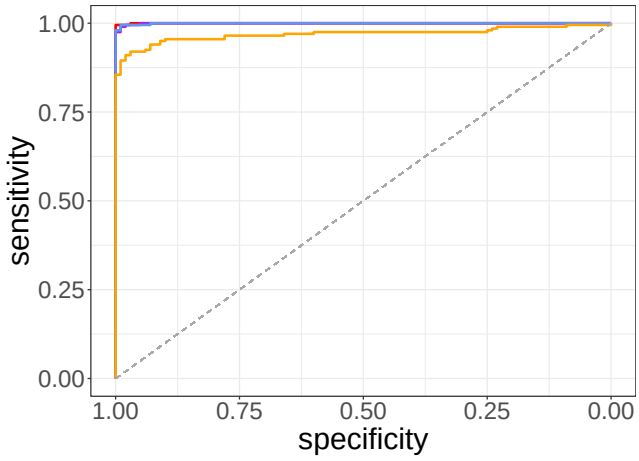

Figure 5.5 – ROC - GEVC

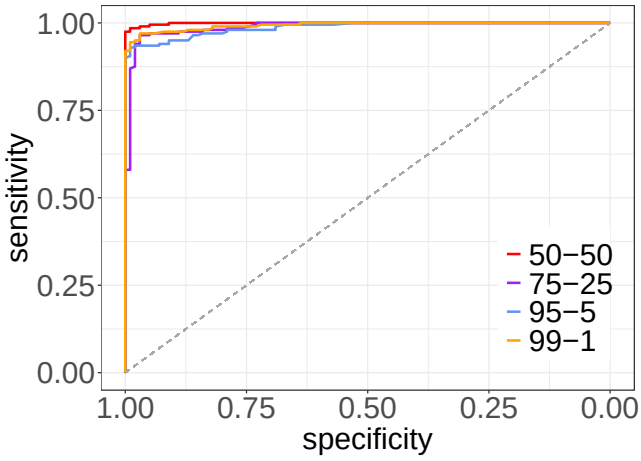

Figure 5.6 – ROC - GEVC

Chapitre 6

Conclusion

Dans ce travail basé sur l'analyse de Vignotto and Engelke (2020), nous avons comparé trois méthodes de détection d'anomalies basées sur l'analyse des valeurs extrêmes à deux méthodes qui ne le sont pas.

Après avoir posé les bases de la théorie des valeurs extrêmes, nous avons décrit chacun des cinq algorithmes d'intérêt : l'Extreme Value Machine de Rudd et al. (2018), le Generalized Pareto Distribution Classifier et le Generalized Extreme Value Classifier de Vignotto and Engelke (2020), le One class Support Vector Machine de Schölkopf et al. (1999) et enfin l'Isolation Forest de Liu et al. (2012).

Après avoir simulé des données à partir de distributions normales bivariées, nous avons testé nos algorithmes. Voici nos constatations : premièrement, les trois méthodes basées sur les valeurs extrêmes montrent toutes trois d'excellent résultats avec cependant une précision pouvant varier jusqu'à près de 20% pour l'EVM en fonction de ses paramètres, ce qui peut être retenu comme point négatif de cette méthode. Ensuite, sur base de ces mêmes données, nous avons pu montrer que la performance du SVM est également assez variable. On voit donc déjà une meilleure fiabilité de la part des méthodes de Vignotto and Engelke (2020).

Nous avons ensuite utilisé le protocole OLETTER de Bendale and Boult (2015) qui évalue la performance de systèmes de classification sur base des données LETTER de Frey and Slate (1991). Les méthodes basées sur la théorie des valeurs extrêmes montrent de meilleurs résultats mais leur précision chute légèrement plus vite chez nous que chez Vignotto and Engelke (2020). En effet, la mise en place du seuil δ de Rudd et al. (2018) permettant de tenir compte du nombre de classes anormales dans le processus n'est pas clair. De ce fait, il se pourrait que nos expériences diffèrent légèrement de celles pratiquées par Vignotto and Engelke (2020). Ceci expliquerait l'obtention de chiffres différents bien que les conclusions soient identiques.

Enfin, les analyses des bases de données de Quinlan et al. (1987) et Mangasarian et al. (1995) portant sur les diagnostics d'une maladie de la thyroïde et du cancer

du sein au Wisconsin, nous permettent de confirmer les conclusions précédentes avec de meilleures performances pour le GEVC et le GPDC comparés à l'iForest. Cependant, le SVM a dans ces deux cas-ci une performance aussi bonne que les deux premiers algortihmes.

Suite à ces résultats, nous avons cherché à aller plus loin en testant deux cas que l'on peut qualifier "d'extrêmes", afin de voir où s'arrête l'efficacité des algorithmes de Vignotto and Engelke (2020). Alors que leur performance est en chute libre lorsque les classes d'anomalies se rapprochent trop des classes normales, elle n'est pas impactée dans le cas de données mal balancées.

En conclusion, les méthodes de détection d'anomalies basées sur l'analyse des valeurs extrêmes fournissent en général de meilleurs résultats que l'iForest.

Elles fournissent également des résultats similaires voire meilleurs que le SVM. Ce n'est en revanche pas le cas lorsque les classes sont mal séparées.

Notons également que l'EVM et le SVM semblent plus sensibles à leurs paramètres que le GPDC.

De plus, sachant que le GEVC et le GPDC reposent sur la distance entre les points et non pas entre les classes comme l'EVM, ils peuvent sans problème être utilisés dans le cas où l'on ne dispose que d'une seule classe.

Dans un travail ultérieur, il pourrait être intéressant de tester d'autres limites de ces deux algorithmes. Nous pourrions par exemple évaluer leur performance dans le cadre de l'utilisation de données à très grandes dimensions. Il pourrait également être intéressant de pouvoir appliquer ces algorithmes à des images ce qui permettrait d'ouvrir largement le champ de leur utilisation.

Bibliographie

Beirlant, J., Goegebeur, Y., Segers, J., and Teugels, J. L. (2004). *Statistics of extremes : theory and applications*, volume 558. John Wiley & Sons.

Bendale, A. and Boult, T. (2015). Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902.

Bolton, R. J. and Hand, D. J. (2002). Statistical fraud detection : A review. *Statistical science*, 17(3) :235–255.

Castillo, E. (2012). *Extreme value theory in engineering*. Elsevier.

Dua, D. (2017). Graff. *C.,"UCI Machine Learning Repository [http ://archive. ics. uci. edu/ml].*

Embrechts, P., Klüppelberg, C., and Mikosch, T. (2013). *Modelling extremal events : for insurance and finance*, volume 33. Springer Science & Business Media.

Espié, M. (2007). Le cancer du sein en chiffres. *La lettre du Gynécologue*, 325.

Fan, J., Upadhye, S., and Worster, A. (2006). Understanding receiver operating characteristic (roc) curves. *Canadian Journal of Emergency Medicine*, 8(1) :19–20.

Frey, P. W. and Slate, D. J. (1991). Letter recognition using holland-style adaptive classifiers. *Machine learning*, 6(2) :161–182.

Gareth, J., Daniela, W., Trevor, H., and Robert, T. (2013). *An introduction to statistical learning : with applications in R*. Spinger.

Havil, J. (2003). Gamma : exploring euler's constant. *The Australian Mathematical Society*, page 250.

Hoo, Z. H., Candlish, J., and Teare, D. (2017). What is an roc curve ?

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1) :1–39.

Mangasarian, O. L., Street, W. N., and Wolberg, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4) :570–577.

Preiss, B. R. (1999). *Data Structure and Algorithms : With Object-oriented Design Patterns in Java*. John Wiley & Sons.

Quinlan, J. R., Compton, P. J., Horn, K., and Lazarus, L. (1987). Inductive knowledge acquisition : a case study. In *Proceedings of the Second Australian Conference on Applications of expert systems*, pages 137–156.

Rudd, E. M., Jain, L. P., Scheirer, W. J., and Boult, T. E. (2018). The extreme value machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3) :762–768.

Schölkopf, B., Williamson, R. C., Smola, A., Shawe-Taylor, J., and Platt, J. (1999). Support vector method for novelty detection. *Advances in neural information processing systems*, 12.

Slavík, P. (1996). A tight analysis of the greedy algorithm for set cover. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 435–441.

Vignotto, E. and Engelke, S. (2020). Extreme value theory for anomaly detection–the gpd classifier. *Extremes*, 23(4) :501–520.

Annexe A

Données simulées

AUC	k	zeta
0.82	5	0
0.809	20	0
0.812	50	0
0.998	5	0.5
0.999	20	0.5
0.98	50	0.5
0.991	5	1
0.996	20	1
0.999	50	1

Table A.1 – AUC en fonction des paramètres pour l'EVM

Modèle	gamma	nu	auc
1	0.1	0.9	0.929
2	0.1	0.8	0.940
3	0.2	0.9	0.944
4	0.1	0.7	0.948
5	0.1	0.6	0.951
6	0.2	0.8	0.957
7	0.1	0.5	0.958
8	0.3	0.9	0.962
9	0.2	0.7	0.962
10	0.1	0.4	0.963
11	0.2	0.6	0.969
12	0.1	0.3	0.970
13	0.3	0.8	0.972
14	0.1	0.2	0.973
15	0.2	0.5	0.975
16	0.4	0.9	0.977
17	0.3	0.7	0.978
18	0.2	0.4	0.979
19	0.1	0.1	0.980
20	0.3	0.6	0.982
21	0.4	0.8	0.983
22	0.2	0.3	0.983
23	0.5	0.9	0.985
24	0.4	0.7	0.986
25	0.3	0.5	0.987
26	0.5	0.8	0.987
27	0.4	0.6	0.989
28	0.2	0.2	0.989
29	0.6	0.9	0.989
30	0.3	0.4	0.990
31	0.5	0.7	0.990
32	0.6	0.8	0.990
33	0.7	0.9	0.991
34	0.4	0.5	0.991
35	0.5	0.6	0.992
36	0.8	0.9	0.992
37	0.3	0.3	0.992
38	0.6	0.7	0.992
39	0.7	0.8	0.992
40	0.9	0.9	0.993
41	0.4	0.4	0.993
42	0.5	0.5	0.993
43	0.8	0.8	0.993
44	0.7	0.7	0.994
45	0.6	0.6	0.994
46	1	0.9	0.994
47	0.9	0.8	0.994
48	0.3	0.2	0.994
49	0.6	0.5	0.994
50	0.8	0.7	0.994
51	0.5	0.4	0.994
52	0.7	0.6	0.994
53	1	0.8	0.994
54	0.2	0.1	0.995
55	0.9	0.7	0.995
56	0.4	0.3	0.995
57	0.6	0.4	0.995
58	0.8	0.6	0.995
59	0.7	0.5	0.995
60	0.4	0.2	0.995
61	1	0.7	0.995
62	0.8	0.5	0.996
63	0.5	0.3	0.996
64	0.9	0.6	0.996
65	0.7	0.4	0.996
66	1	0.6	0.996
67	0.9	0.5	0.996
68	0.8	0.4	0.996
69	0.6	0.3	0.996
70	1	0.5	0.996
71	0.7	0.3	0.997
72	0.8	0.3	0.997
73	0.9	0.4	0.997
74	1	0.4	0.997
75	0.5	0.2	0.997
76	0.9	0.3	0.997
77	0.7	0.2	0.997
78	0.6	0.2	0.997
79	0.3	0.1	0.997
80	0.8	0.2	0.998
81	1	0.3	0.998
82	0.9	0.2	0.998
83	1	0.2	0.998
84	0.4	0.1	0.998
85	0.6	0.1	0.999
86	0.7	0.1	0.999
87	0.5	0.1	0.999
88	0.8	0.1	0.999
89	0.9	0.1	0.999
90	1	0.1	0.999

Table A.2 – AUC en fonction du modèle de SVM

Annexe B

Protocole OLETTER

Comparaisons des mesures F pour chaque algorithme pour les trois premières classes lors de l'exécution du protocole OLETTER sur des sous-échantillons. Les classes normales sont modifiées à chaque itération.

Classe 1	Classe 2	Classe 3
0.947	0.919	0.897
0.937	0.901	0.881
0.967	0.934	0.889
0.96	0.904	0.871
0.954	0.915	0.88

Table B.1 – GEVC

Classe 1	Classe 2	Classe 3
0.972	0.942	0.916
0.955	0.923	0.895
0.966	0.942	0.912
0.967	0.932	0.901
0.97	0.938	0.905

Table B.2 – GPDC

Classe 1	Classe 2	Classe 3
0.928	0.912	0.909
0.944	0.908	0.884
0.906	0.881	0.853
0.934	0.911	0.884
0.946	0.92	0.889

Table B.3 – EVM

Classe 1	Classe 2	Classe 3
0.783	0.648	0.552
0.641	0.582	0.564
0.766	0.592	0.462
0.619	0.48	0.414
0.632	0.535	0.471

Table B.4 – SVM

Classe 1	Classe 2	Classe 3
0.966	0.936	0.91
0.961	0.929	0.902
0.956	0.929	0.896
0.96	0.925	0.898
0.971	0.941	0.913

Table B.5 – iForest

Annexe C

Thyroïde

AUC	Gamma	Nu
0.7314	/	/
0.8695	0.5	/
0.8202	0.2	/
0.8831	0.8	/
0.8855	0.9	/
0.7314	/	0.5
0.7379	/	0.2
0.7161	/	0.8
0.8695	0.5	0.5
0.8855	0.9	0.5
0.9033	0.9	0.2
0.9033	**0.9**	**0.1**
0.903	0.9	0.3

Table C.1 – AUC en fonction des paramètres pour le SVM

AUC	ntrees	sample_size
0.6573	100	256
0.6644	80	256
0.7114	**50**	**256**
0.7065	20	256
0.653	120	256
0.6771	50	200
0.6959	50	100
0.6516	50	300

Table C.2 – AUC en fonction des paramètres pour l'iForest

Annexe D

Cancer du sein au Wisconsin

AUC	**Gamma**	**Nu**
0.9617	/	/
0.9973	0.5	0.5
0.9973	0.5	0.2
0.9973	0.5	0.8
0.994	0.2	0.5
0.9975	**0.8**	**0.5**

Table D.1 – AUC en fonction des paramètres pour le SVM

Annexe E

Limites

E.1 Classes mal séparées : paramètres du SVM

Le choix du modèle a été fait grâce à l'étude de différentes combinaisons des paramètres afin de sélectionner celle fournissant le meilleur résultat pour la classe mauve.

AUC	**Gamma**	**Nu**
0.8242	0.5	0.5
0.8412	0.5	0.3
0.8775	**0.5**	**0.1**
0.8017	0.5	0.7
0.8212	0.5	0.4
0.8567	0.5	0.2
0.8407	0.3	0.1
0.8527	0.7	0.1
0.8396	0.9	0.1

Table E.1 – AUC en fonction des paramètres pour le SVM

E.2 Données mal balancées

L'ensemble d'entrainement est représenté en noir alors que l'ensemble de test est en couleur. En jaune les données normales et en orange les anomalies.

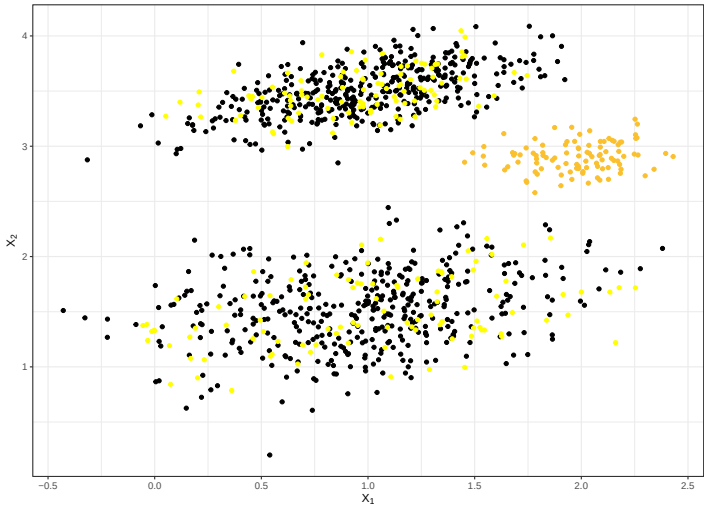

Figure E.1 – 50 – 50%

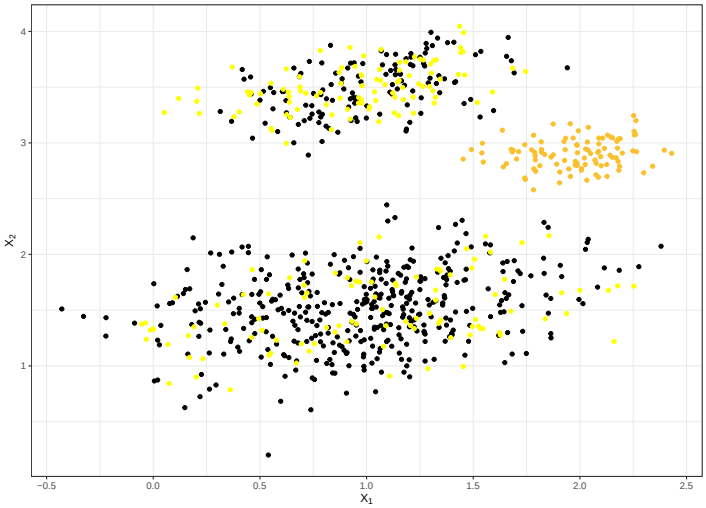

Figure E.2 – 75 – 25%

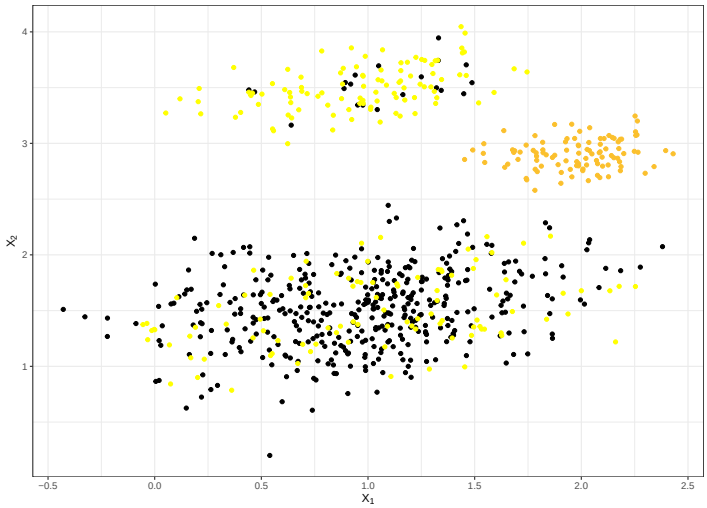

Figure E.3 – 95 – 5%

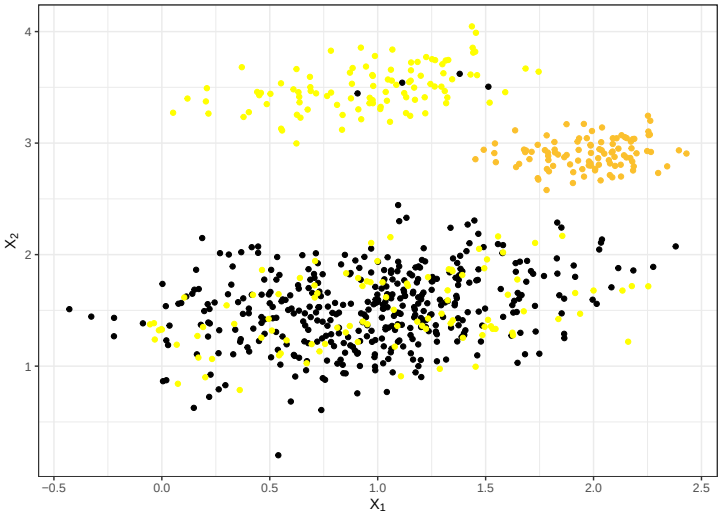

Figure E.4 – 99 – 1%

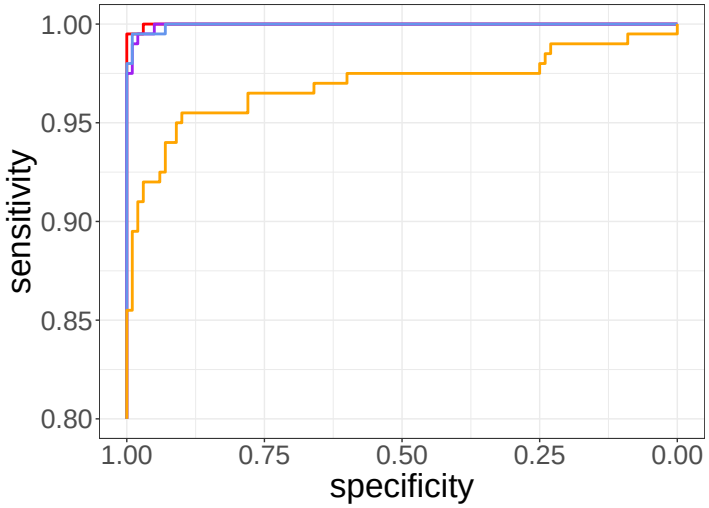

Figure E.5 – Zoom ROC - GEVC

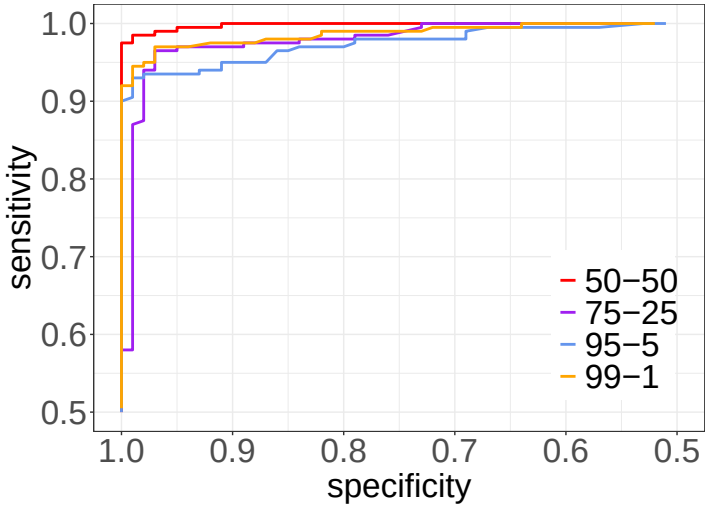

Figure E.6 – Zoom ROC - GPDC

Annexe F

Codes

F.1 Implémentation du GEVC

```r
library(extRemes)

# Load data
# Load testset

n <- nrow(data)
n_test <- nrow(testset)
seuil_alpha <- 0.1

# TRAINING
    ----------------------------------------------------------------
Distribution_dmin <- c() #Distances between x_i and the others
    points
for (i in 1:n){
  distances_neg <- c()
  for (j in 1:n){
    if (identical(data[i,], data[j,])) next
    d <- dist(rbind(data[j,], data[i,]), method = 'euclidean')
    distances_neg <- c(distances_neg, -d)
  }
  d_min <- max(distances_neg)
  Distribution_dmin <- c(Distribution_dmin, d_min)
}

# Fitting Weibull
fit <- fevd(Distribution_dmin,type="GEV")
tau_hat <- fit$results$par[["scale"]]
alpha_hat <- fit$results$par[["shape"]]

# Compute c threshold
```

```r
c <- tau_hat*(-log(seuil_alpha))^(-1/alpha_hat)

# TESTING
    ----------------------------------------------------------------
for(i in 1:n_test){
  distances_neg_d0 <- c()
  for (j in 1:n){
    d0 <- dist(rbind(testset[i,], data[j,]), method = 'euclidean')
    distances_neg_d0 <- c(distances_neg_d0, -d0)
  }
  d0_min_neg <- max(distances_neg_d0)

  # Hypothesis tests
  if (-d0_min_neg > c){
    testset[i,"prediction"] <- -1 # x_0 is abnormal
  } else {
    testset[i,"prediction"] <- 1  # x_0 is normal
  }
}
```

F.2 Implémentation du GPDC

```r
# Load data
# Load testset

n <- nrow(data)
n_test <- nrow(testset)
p <- ncol(data)
alpha <- 0.1
k <- 20

# TRAINING
    ----------------------------------------------------------------
gpdc_xi_q <- function(data, x_0, k, n){
  # Compute the negated distances between x0 and each train point
  Distribution_distances <- c() # Negated distances
  for (i in 1:n){
    if (identical(x_0, data[i,])) next
    d <- dist(rbind(data[i,], x_0), method = 'euclidean')
    Distribution_distances <- c(Distribution_distances, -d)
  }
  # Estimate xi_n using only the biggest k negated distances
  R <- sort(Distribution_distances)
  sum <- 0
  n_R <- length(Distribution_distances)
  for (i in 1:k){
    l <- log(R[n_R+1-i]/R[n_R-k])
```

```
    sum <- sum + l
  }
  xi <- sum/k
  q <- R[n-k]*(1/k)^(-xi)
  return(list("xi" = xi, "q_neg" = -q))
}

# Compute s and t thresholds
dist_xi <- c() # xi_n for each point in the training set
dist_q <- c() # -q for each point in the training set
for(i in 1:n){
  gpdc <- gpdc_xi_q(data, data[i,], k, n)
  dist_xi <- c(dist_xi, gpdc$xi)
  dist_q <- c(dist_q, gpdc$q_neg)
}
s <- quantile(dist_xi, 1-alpha/2)
t <- quantile(dist_q, 1-alpha/2)

# TESTING
  ----------------------------------------------------------------
# Hypothesis test for each new point x_0 in the testset
for(c in 1:n_test){
  x_0 <- testset[c,]
  gpdc_x0 <- gpdc_xi_q(data, x_0, k, n)
  xi_hat <- gpdc_x0$xi
  if (p*xi_hat >= s){
    testset[c,"prediction"] <- -1 # x_0 is abnormal
  } else {
    q_neg_x0 <- gpdc_x0$q_neg
    if (q_neg_x0 > t) {
      testset[c,"prediction"] <- -1 # x_0 is abnormal
    } else {
      testset[c,"predicion"] <- 1 # x_0 is normal
    }
  }

}
```

UNIVERSITE CATHOLIQUE DE LOUVAIN

Faculté des sciences

Place des sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/sc**