Détection d'anomalies et de nouveautés

Noémie Haouzi
Satyanarayanan Vengathesa Sarma

19 mai 2017

Techniques Avancées d'Apprentissage

Prof. Stéphan Clémençon

1 Introduction et problématique

Une des tâches les plus communes en Machine Learning est la classification. Le problème prédictif correspondant consiste à déterminer la classe d'une nouvelle observation en utilisant un modèle entraîné au préalable sur une base d'entraînement composée de données issues de plusieurs classes. Cependant, il peut arriver qu'une seule classe soit disponible dans l'échantillon d'entraînement. C'est notamment le cas lorsque l'on s'intéresse au problème de la détection de nouveautés. On peut également se trouver dans une situation voisine, où une très petite fraction de la base d'entraînement est contaminée par des anomalies (*outliers*). Concrètement, cela peut arriver lorsque l'on s'intéresse au contrôle du système électronique d'un appareil, afin de prévoir un dysfonctionnement. Dans une telle situation, les pannes peuvent être très rares et il est donc difficile de réunir les données correspondantes. En d'autres termes, les observations négatives sont peu nombreuses voire inexistantes.

D'un point de vue statistique, le problème correspondant à la détection d'anomalies et de nouveautés est la classification à une classe (*one-class classification*). Dans ce type de problèmes, les classifieurs ne peuvent apprendre que sur la classe positive, celle qui est bien caractérisée par les données. A l'inverse, la classe négative est composée des anomalies ou des nouveautés, qui peuvent potentiellement être totalement absentes de la base de données. Cela marque une nette différence avec la classification multi-classe usuelle, pour laquelle la plupart des classifieurs supposent que les classes sont équilibrées. Dans un tel cadre, un classifieur one-class est un classifieur capable d'apprendre un partitionnement multi-dimensionnel des données en n'utilisant que les exemples positifs.
Khan et Madden (2013) proposent une taxinomie des différentes techniques existantes pour le problème de la classification à une classe. Selon le degré de disponibilité des exemples négatifs pour l'entraînement, ou encore le domaine d'applications, de nombreuses méthodes permettant de détecter des nouveautés ou des anomalies existent. En particulier, une grande majorité des méthodes adaptées à ces problèmes sont basées sur l'algorithme One-class Support Vector Machines (OSVM). Diverses variantes de cet algorithmes existent ; ainsi, nous allons commencer par présenter, dans une première partie, l'algorithme OSVM développé par Scholkopf et al. De façon générale, l'idée est de construire un hyperplan séparant la masse des données observées de l'origine. Par la suite, nous détaillerons quelques expérimentations que nous avons effectué avec l'algorithme OSVM, pour la détection d'anomalies sur des données simulées et réelles. Nous comparerons ces résultats avec ceux d'une méthode classique en détection d'outliers et d'anomalies, basée sur la distance de Mahalanobis [1].

2 Rappels sur le SVM

Considérons le cas d'un *Support Vector Machine* (SVM) classique avec un échantillon $((x_i, y_i))_{i=1,..,l}$ où $x_i \in \mathcal{X}$ est un *feature* et $y_i \in \{-1, +1\}$ est le label de l'exemple i. Le principe d'un SVM est de construire un hyperplan à marges maximales qui sépare l'espace des observations en deux parties : les features de label -1 sont d'un côté de l'hyperplan, les features de label $+1$ sont de l'autre. On rappelle que la marge est la distance entre l'hyperplan et les observations les plus proches, les vecteurs à support.

1. Le langage utilisé dans le cadre de ce projet est `Python` avec sa librairie de machine learning `scikit-learn`

L'hyperplan séparateur est donné par l'équation $w^T x + w_0 = 0$ où $w, w_0 \in \mathcal{X}$. Ainsi, en utilisant la projection orthogonale de w sur F ainsi qu'une normalisation qui implique que pour tout $i \in \{1, \ldots, l\}$, $y_i(w^T x_i + w_0) \geq 1$, on cherche à déterminer w solution de :

$$\arg\max_w \ \frac{1}{2}\|w\|^2 \text{ s.c. } y_i(w^T x_i + w_0) \geq 1$$

Kernel Trick
L'espace des observations n'étant pas toujours linéairement séparable (ie qu'il existe $w \in \mathcal{X}$ tel que $y_i w^T x_i \geq 0$ pour tout $i = 1, \ldots, l$), on utilise une fonction non linéaire $\Phi : \mathcal{X} \to F$ (*feature map*) qui projette les features dans F, un espace de dimension plus grande. Ainsi, les observation x_i sont remplacées par leur projection dans F, $\Phi(x_i)$. Le problème avec ce changement de dimension est qu'il va nécessiter le calcul de produits scalaires dans un espace de grande dimension, ce qui peut être très coûteux en termes de calcul. L'astuce est d'utiliser une fonction noyau qui vérifie :

$$k(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$$

Le produit scalaire se fait alors dans l'espace d'origine, il est donc moins coûteux. De plus, la transformation Φ n'a pas besoin d'être connue : seule la fonction noyau intervient dans les calculs. Pour éviter le surapprentissage, des variables ressort (*slack variables*) ξ_i sont introduites. Elles autorisent certaines observations à se trouver du mauvais côté de l'hyperplan. On introduit également une constante $C > 0$ qui pénalise les variables ξ_i élevées dans la fonction objectif. C'est donc un hyperparamètre qui contrôle le compromis entre la compléxité du modèle et le nombre d'erreurs de classification.

Ainsi, le problème de minimisation du SVM avec variables ressort s'écrit :

$$\min_{w \in F, \xi \in \mathbb{R}^l} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i \tag{1}$$

$$\text{s.c. } y_i(w^T \Phi(x_i)) \geq 1 - \xi_i, \quad \xi_i \geq 0, \ i = 1, \ldots, l \tag{2}$$

En utilisant le *kernel trick* et après résolution du problème (1-2) grâce aux multiplicateurs de Lagrange $\alpha_i \geq 0$, on peut définir la fonction de décision suivante :

$$f(x) = \text{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i k(x_i, x) + w_0\right) \ ^2$$

3 One Class SVM (Schölkopf et al., 2001)

On se place maintenant dans un contexte très similaire à celui la section précédente, à ceci-près que l'on considère à présent un cadre non-supervisé. Nous cherchons à détecter des anomalies ou de la nouveauté.

Schölkopf et al. proposent d'estimer le support des observations. Pour cela, ils définissent un algorithme qui retourne une fonction binaire qui doit capturer le support de la densité associée

2. On utilisera la convention $\text{sgn}(z) = 1$ pour $z \geq 0$ et -1 sinon.

à P : elle est non nulle sur le support des inputs et nulle partout ailleurs. Ce problème est alors plus simple que d'estimer entièrement la distribution des features et de plus, cet algorithme est applicable même dans le cas où la densité des données n'est pas définie. L'algorithme proposé est en fait une forme particulière du One-Class SVM.

3.1 Un résultat fondamental

Nous reprenons le contexte de la section précédente où $\Phi : \mathcal{X} \to F$ est une *feature map* et $x_1, \ldots, x_l \in \mathcal{X}$ est notre échantillon d'entraînement issu d'une loi P.

Rappelons la définition d'un **hyperplan d'appui** (*supporting hyperplan*). Pour A une partie non vide d'un espace affine et x_0 un élément de A, on dit que H est un hyperplan d'appui de A en x_0 lorsque x_0 appartient à H et A est inclus dans un des demi-espaces limités par H.

Soit $\Phi(x_1), \ldots, \Phi(x_l)$ un échantillon séparable c'est-à-dire qu'il existe $w \in F$ tel que $w^T\Phi(x_i) > 0$ pour tout i. On peut alors montrer qu'il existe un unique hyperplan d'appui qui :

- sépare toutes les observations de l'origine
- a une distance à l'origine maximale parmi tous les autres hyperplans

Cet hyperplan est donné par la solution du problème, pour tout $\rho > 0$:

$$\min_{w \in F} \|w\|^2 \quad \text{s.c. } w^T\Phi(x_i) > \rho, \ i = 1, \ldots, l$$

3.2 L'algorithme

Le but du One Class SVM proposé par Schölkopf et al. est de déterminer un hyperplan qui sépare toutes les observations de l'origine et dont la distance à l'origine est maximale parmi tous les hyperplans aux mêmes propriétés. Grâce au résultat de la section précédente et en s'appuyant sur le problème de minimisation du SVM (1-2), on peut montrer que le problème à résoudre est le suivant :

$$\min_{w \in F, \xi \in \mathbb{R}^l, \rho \in \mathbb{R}} \frac{1}{2}\|w\|^2 + \frac{1}{\nu l}\sum_{i=1}^{l} \xi_i - \rho \tag{3}$$

$$\text{s.c. } w^T\Phi(x_i) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, l \tag{4}$$

avec w est le vecteur perpandiculaire à l'hyperplan séparateur et ρ est la distance de l'hyperplan séparateur à l'origine dans l'espace des features F.

Dans le programme de minimisation du SVM, C était le paramètre de lissage. Ici, ν est un paramètre qui contrôle le compromis entre ajouter des outliers en augmentant la taille de la région estimée et maximiser la marge entre l'hyperplan et l'origine. En effet, ν correspond à une borne supérieure de la fraction d'outliers. Il caractérise également une borne inférieure de la part de features utilisés comme vecteurs de support.

Comme dans le SVM, on définit la fonction de décision comme :

$$f(x) = \text{sgn}(w^T\Phi(x) - \rho)$$

L'idée est que pour une nouvelle observation x, $f(x)$ retourne $+1$ si $\Phi(x)$ se trouve du côté de la masse des observations : elle est dite "normale" ; elle retourne -1 si $\Phi(x)$ se trouve entre l'origine et l'hyperplan : x se comporte comme un outlier.

En utilisant les multiplicateurs de Lagrange $\alpha_i, \beta_i \geq 0$, on obtient les solutions suivantes :

$$w = \sum_{i=1}^{l} \alpha_i \Phi(x_i)$$
$$\alpha_i = \frac{1}{\nu l} - \beta_i \leq \frac{1}{\nu l}, \quad \sum_{i=1}^{l} \alpha_i = 1$$
$$\rho = w^T \Phi(x_i) = \sum_{j=1}^{l} \alpha_j k(x_j, x_i)$$

La fonction de décision devient :

$$f(x) = \text{sgn}\left(\sum_{i=1}^{l} \alpha_i k(x_i, x) - \rho\right) \tag{5}$$

3.3 Quelques remarques

La méthode proposée par Schölkopf et al. simplifie le problème de la détection d'outliers et d'anomalies en estimant le support des observations et non la distribution entière, comme on pourrait le faire traditionnellement. De plus, elle ne nécessite pas que la loi des observations soit absoluement continue et donc elle permet de résoudre une plus grande variété de problèmes. Par ailleurs, grâce à l'utilisation du kernel trick, il est possible de construire une fonction non linéaire qui sépare les observations normales de l'origine. La projection dans un espace de plus grande dimension et l'utilisation d'un noyau augmente les chances d'existence d'un hyperplan séparateur des données de l'origine. A forciori, si le noyau choisi est gaussien, le problème est automatiquement séparable et la détection d'outliers est possible. Notons tout de même que la région définie par l'hyperplan dans l'espace des features est déterminée par le noyau choisi. Or, ce choix est laissé au statisticien, ce qui peut donner lieu à des résultats changeants selon les noyaux. Enfin, l'article ne mentionne pas de méthode pour déterminer le paramètre ν, or celui-ci est central. L'algorithme nécessite donc d'avoir un a-priori sur la proportion d'outliers et de vecteurs de supports présents dans l'échantillon, ce qui n'est pas toujours évident.

La complexité algorithmique tout à fait acceptable de l'algorithme de Schölkopf et al. va nous permettre de faire des expériences de manière très rapide afin d'en mesurer son efficacité.

4 Expériences numériques

Dans cette section nous présentons quelques expériences que nous avons réalisé afin de détecter des anomalies ou des nouveautés à l'aide de l'algorithme OSVM. Dans un premier temps, nous présentons les résultats de la détection d'anomalies sur un jeu de données simulées en nous intéressant à l'impact du paramètre ν. Puis dans un second temps, nous tenterons de détecter à la fois des anomalies et des nouveautés sur la base de données USPS (également utilisée par

Schölkopf et al. dans l'article présenté dans la section précédente). Pour cette seconde base, nous comparerons les résultats de la méthode OSVM à ceux d'une méthode simple basée sur la distance de Mahalanobis.

4.1 Détection d'anomalies sur données simulées

Nous commençons par générer un jeu de données avec 1000 observations et deux prédicteurs, afin de pouvoir visualiser les résultats sur un graphique. Les points normaux sont regroupés en un ou deux groupes compacts, selon les configurations considérées. Les points anormaux (outliers) peuvent quant à eux être placés à divers endroits : leurs coordonnées sont générées selon une loi $\mathcal{U}$([-10,10]). On décide de fixer à 5% la véritable proportion d'anomalies ; l'objectif du problème est alors de retrouver quelles observations sont anormales en utilisant l'algorithme OSVM. La Figure 1 présente les résultats obtenus, pour les trois configurations considérées et en faisant varier la valeur de ν dans $\{0.01, 0.05, 0.5\}$. Notons que nous ne nous intéressons pas ici à l'optimisation des paramètres classiques du SVM à noyau gaussien (comme le coefficient multiplicatif γ). La valeur par défaut dans `scikit-learn` est utilisé pour ceux-ci.

En premier lieu, on constate que quelque soit la configuration, les meilleurs résultats sont obtenus pour $\nu = 0.05$, qui est la proportion exacte d'outliers dans le jeu de données. A l'inverse, les pires résultats sont toujours obtenus pour $\nu = 0.5$. Nous pouvons penser que cela est dû au fait que ν est une borne inférieure pour la proportion de vecteurs supports. Ainsi, il s'agit d'un cas de sous-apprentissage : les vecteurs supports devant représenter au moins 50% de l'échantillon, il s'agit surtout de points normaux. Pour $\nu = 0.01$, on obtient le résultat inverse : puisque nu est une borne supérieure pour la proportion d'anomalies, on force l'algorithme à considérer que celle-ci ne peut dépasser 1%. C'est pourquoi on obtient de nombreux points anormaux qui sont détectés comme étant normaux. Il s'agit également d'une situation typique de sur-apprentissage. Il est donc est important d'avoir un à priori le plus précis possible sur la proportion d'outliers dans l'échantillon.

4.2 Données USPS

Nous considérons à présent le jeu de données US Postal Service (USPS) sur lequel Schölkopf et al. (2001) ont également réalisé des expériences. Il s'agit de données d'images représentant des chiffres allant de 0 à 9, écrits à la main. En plus de la détection d'anomalies, nous allons nous intéresser à la détection de nouveautés sur ce jeu de données. Dans les deux cas, nous comparerons les résultats de l'algorithme OSVM à ceux d'une approche plus classique basée la distance de Mahalanobis.

4.2.1 Détection d'anomalies

Comme de nombreuses bases classiques en apprentissage statistique, la base USPS est divisée en un ensemble d'entraînement et un ensemble de test par défaut. Il apparaît que dans la base de test USPS, il existe de nombreuses images dont il est impossible de déterminer le véritable label, car elles sont illisibles. L'objectif du problème de détection d'anomalies est ici de déterminer quelles sont ces images. Cependant, dans la mesure où il n'est pas possible de savoir quelles

images sont illisibles, la seule analyse possible réside dans la comparaison des performances de plusieurs méthodes.

Une façon classique de procéder dans un tel cadre est de supposer que les données sont distribuées selon une certaine loi de probabilité usuelle, puis de tenter de reconstituer la forme de la distribution. Ainsi, il est possible de détecter les anomalies en supposant qu'il s'agit de points placés en dehors de la distribution estimée. Dans notre contexte, en faisant l'hypothèse de distribution gaussienne, on peut calculer la distance de Mahalanobis pour identifier les points qui sont trop anormaux par rapport à cette distribution. Les 10 images les plus anormales par rapport à ce critère sont présentées dans la Figure 2.
Nous comparons les résultats de cette méthode couramment adoptée à ceux de l'algorithme OSVM. Nous choisissons la même valeur des hyper-paramètres que Schölkopf et al. (2001), en particuliers $\nu = 5\%$. Là aussi, les 10 images les plus anormales sont présentées dans la Figure 3. Il semblerait que dans ce cas, la détection d'anomalies par l'algorithme OSVM ne soit pas sensiblement meilleure que celle basée sur la distance de Mahalanobis. Nous pouvons penser que cela est dû au fait que les données sont effectivement distribuée selon une loi gaussienne.

4.2.2 Détection de nouveautés

On s'intéresse à présent à la tâche de détection de nouveautés ; cette fois, nous entrainons nos modèles sur la base d'entraînement puis nous évaluons leur performance sur la base de test. Cependant, nous n'utilisons que les observations de la base d'entraînement correspondant à des images de 0. L'objectif est alors de pouvoir reconnaître qu'une image de la base de test ne représentant pas un 0 est une nouveauté.

Nous commençons de nouveau par calculer la distance de Mahalanobis entre les données d'entraînement et les données de test. Par la suite, on sait que la proportion de 0 dans la base de test est d'environ 18%. On peut en déduire une règle de décision très simple pour détecter les nouveautés, en déclarant comme nouvelle toute observation dont la distance de Mahalanobis est supérieure au quantile à 18% des distances sur la base de test. Les résultats de cette méthode sont présentés sous la forme d'une matrice de confusion dans le Tableau 1. La méthode fonctionne relativement bien, cependant elle requiert de connaître la proportion de nouveautés dans la base analysée. Lorsque l'on se trompe sur cette proportion, le taux d'erreur peut s'avérer très élevé, comme le montre le Tableau 2.
Nous suivons la même démarche pour l'algorithme OSVM. Diverses valeurs de ν ont été testées, et les résultats sont présentés dans les Tableaux 3, 4 et 5. Tout d'abord, on remarque que pour les valeurs de ν élevées, aucune nouveauté n'est classifiée comme représentant un 0. En revanche de nombreuses images représentant des 0 sont détectées comme étant des nouveautés. Cela dit, la proportion d'erreurs ne dépasse jamais les 18% (proportion de 0 dans la base test), ce qui n'était pas le cas pour la méthode précédente. En outre, en prenant des valeurs plus faibles pour ν, les résultats sont sensiblement améliorés et rejoignent ceux de la méthode précédente. La différence essentielle est que, dans le cas où l'a-priori que l'utilisateur peut avoir sur ν est erroné, l'algorithme OSVM est garantit de ne pas dépasser un taux d'erreur correspondant à la proportion de 0. Ainsi, la méthode OSVM semble être une bonne alternative lorsqu'il s'agit de détecter des nouveautés.

5 Conclusions

Le One Class SVM proposé par Schölkopf et al. est un algorithme qui permet de déterminer des anomalies ou des nouveautés dans un cadre non supervisé. L'apport est double : d'une part les auteurs proposent la simplification du problème en estimant le support des observations et non la distribution entière ; d'autre part, l'utilisation du kernel trick est utile pour construire une région de décision non linéaire.
Nos expériences ont montré qu'en pratique, l'algorithme OSVM semble bien fonctionner pour la détection d'anomalies comme pour la détection de nouveautés. Toutefois, il semble qu'il soit plus performant pour cette seconde tâche. Par ailleurs, il est important d'avoir un bon à-priori sur la véritable proportion d'outliers dans l'échantillon afin de calibrer correctement le paramètre ν. Enfin, l'algorithme semble être rapide sur des jeux de données de taille modérée.
Comme nous l'avons vu en cours, d'autres méthodes existent pour détecter des anomalies comme par exemple l'algorithme d'*Isolation forest*. Cette méthode repose sur la construction d'un score d'anormalité des observations en fonction du nombre de conditions nécessaires pour séparer cette observation du reste de la masse des données. On a vu que contrairement à l'algorithme OSVM, il s'agit d'une méthode gloutonne. Ainsi, il pourrait être intéressant de comparer les performances de l'Isolation Forest avec le One-Class SVM, que ce soit en termes de performances ou en termes de temps d'exécution.

Annexe

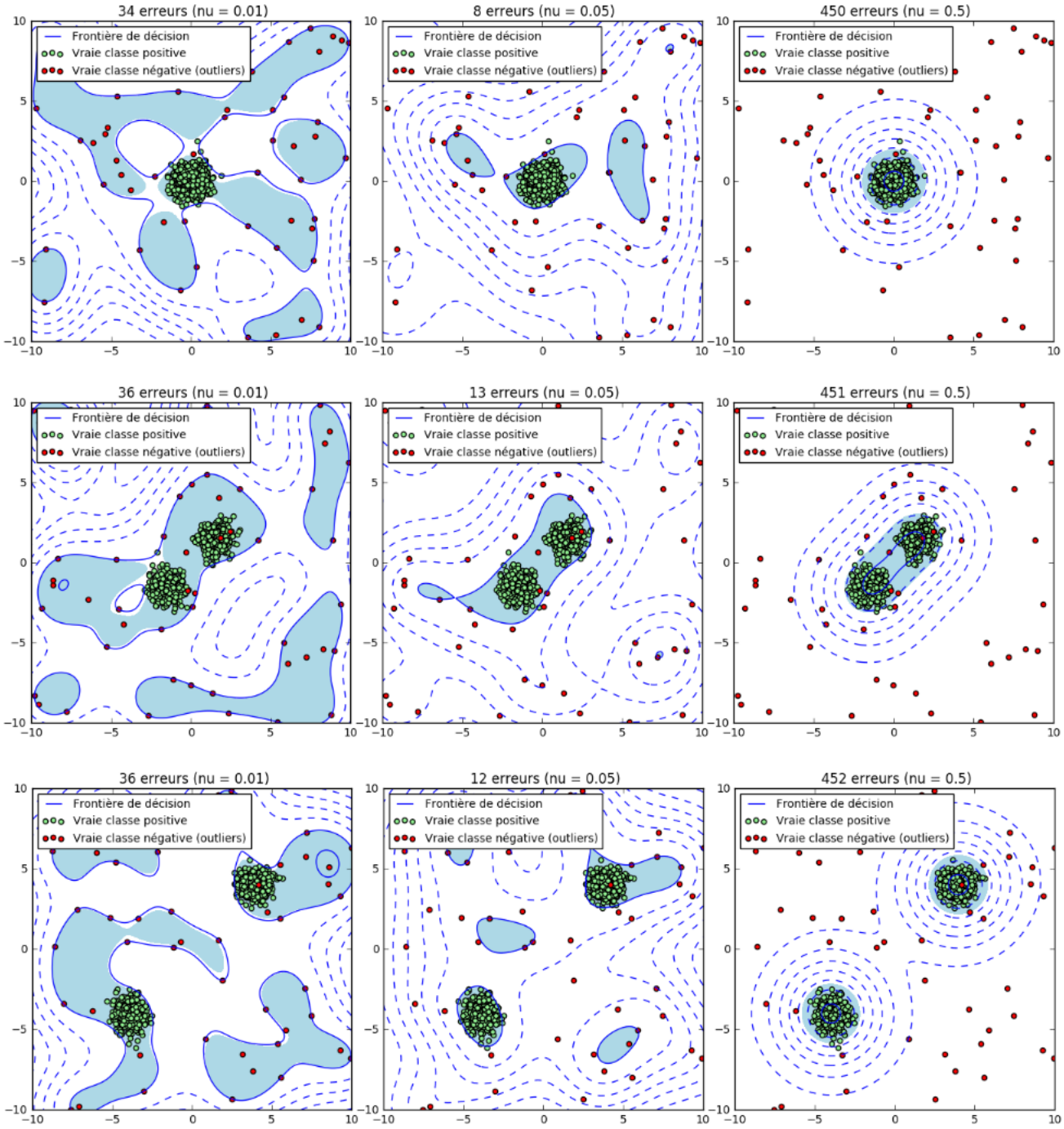

Figure 1. Résultats de la détection d'anomalies sur données simulées
Chaque ligne correspond à une configuration des données et chaque colonne à une valeur du ν

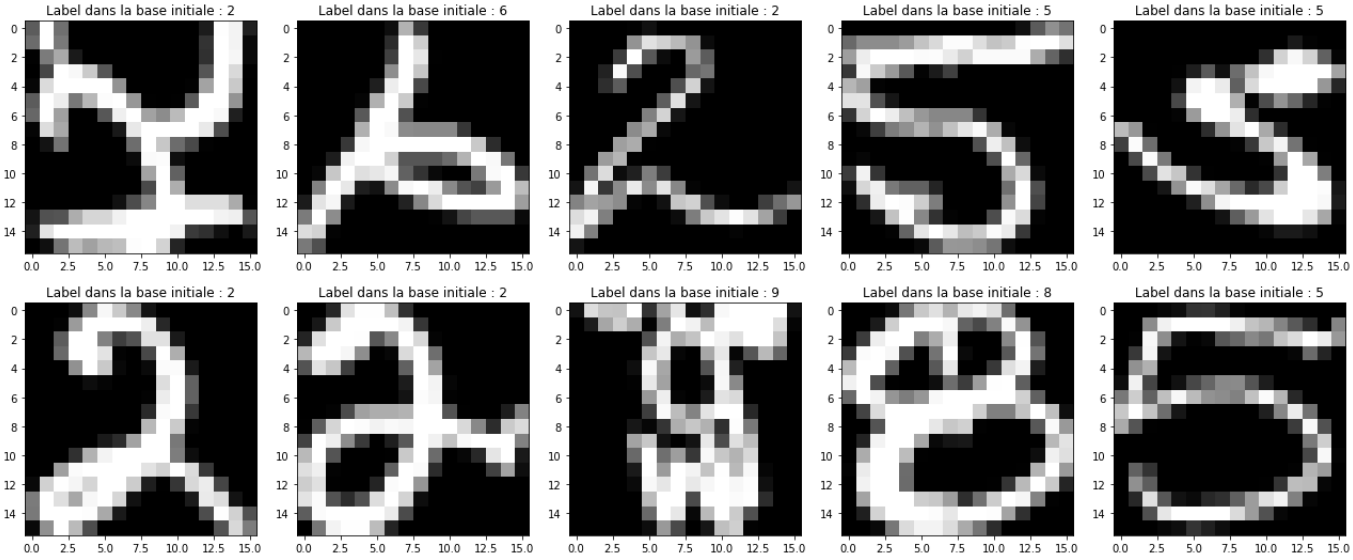

Figure 2. Résultats de la détection d'anomalies sur USPS-test (Mahalanobis, top 10 des anomalies détectées)

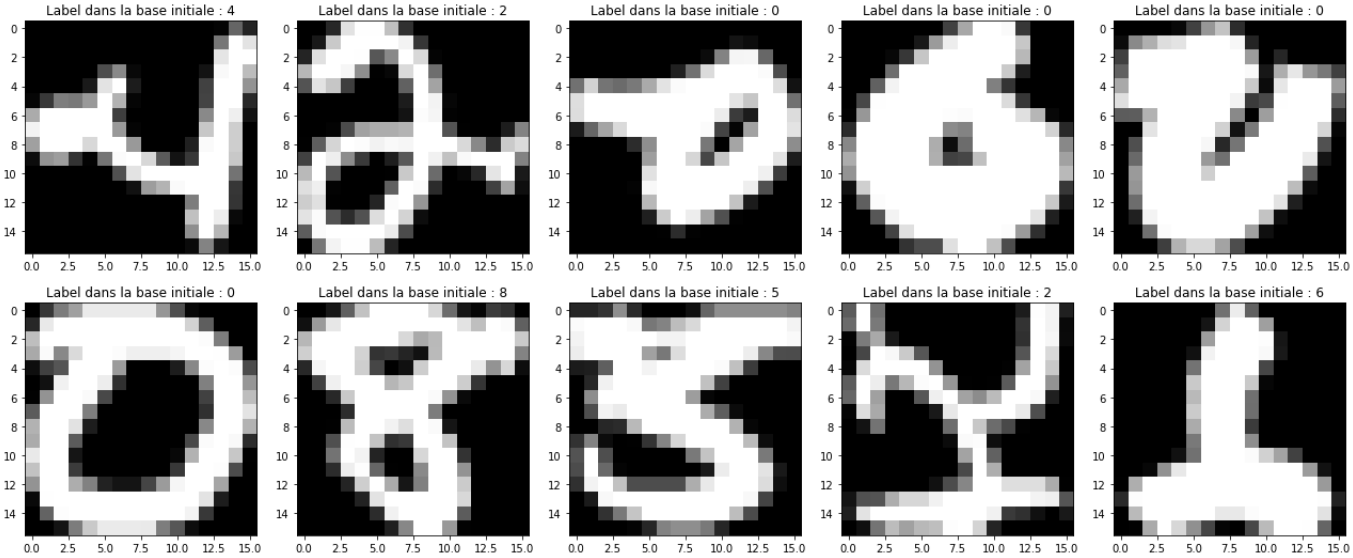

Figure 3. Résultats de la détection d'anomalies sur USPS-test (OSVM, top 10 des anomalies détectées)

	Nouveautés prédites	0 prédits
Vraies nouveautés	79.87%	2.24%
Vrais 0	2.24%	15.64%

Table 1. Résultats de la détection de nouveautés sur USPS-test (Mahalanobis, Quantile à 17.89%)

	Nouveautés prédites	0 prédits
Vraies nouveautés	49.03%	33.08%
Vrais 0	1%	16.89%

Table 2. Résultats de la détection de nouveautés sur USPS-test (Mahalanobis, Quantile à 50%)

	Nouveautés prédites	0 prédits
Vraies nouveautés	77.03%	5.08%
Vrais 0	1.49%	16.39%

Table 3. Résultats de la détection de nouveautés sur USPS-test (OSVM, $\nu = 0.01$)

	Nouveautés prédites	0 prédits
Vraies nouveautés	81.86%	0.24%
Vrais 0	4.78%	13.10%

Table 4. Résultats de la détection de nouveautés sur USPS-test (OSVM, $\nu = 0.2$)

	Nouveautés prédites	0 prédits
Vraies nouveautés	82.11%	0%
Vrais 0	17.89%	0%

Table 5. Résultats de la détection de nouveautés sur USPS-test (OSVM, $\nu = 1$)

Références

[1] Amer M., Goldstein M., Abdennadher S. "Enhancing One Class Support Vector Machines for Unsupervised Anomaly Detection" *ODD'13* (2013).

[2] Corinna C. and Vapnik V. "Support-vector networks" *Machine learning 20.3* (1995).

[3] Khan S. S. et Madden M. G. "One-class classification : taxonomy of study and review of techniques" *The Knowledge Engineering Review, 29(03), 345-374.* (2014).

[4] Schölkopf B., Williamson R. C., Smola A. J., Shawe-Taylor J., et Platt J. C. "Support vector method for novelty detection" *In Advances in neural information processing systems (pp. 582-588).* (2000).

[5] Schölkopf B., Platt J. C., Shawe-Taylor J., Smola A. J. et Williamson R. C. "Estimating the support of a high-dimensional distribution" *Neural computation, 13(7), 1443-1471* (2001).