École Polytechnique Fédérale de Lausanne

Report

Outlier-Novelty Detection

Châteaux Nock Lili - Ni Tingting - Nouira Mongi

January 22, 2021

1 Introduction

Anomaly or novelty detection refers to the techniques of identifying the "out of the ordinary" instances in a data set. This problem can be seen as a binary classification problem : the "normal" data belonging to a target class, and the rest of the data laying in an outlier class. However, it may be difficult to produce many examples belonging to the latter class. This can be the case if outliers are rare, or of unknown and unpredictable form. Because of this unbalanced class, Binary Classification with supervised learning cannot be implemented. Therefore unsupervised or semi-supervised methods will be used in a One-Class-Classification setting, that is learning without negatives examples.

There exist many anomaly detection algorithms [1] used in Machine learning. Two of them will be presented and compared in this report. First the One-Class Support Vector Machine (SVM) algorithm, which is a parametric method, then the Isolation Forest with isolation trees [2]. To conclude this report, those models will be applied on some data sets.

2 Models

2.1 Support Vector Machine

The idea behind this method is to estimate a simple subset S of input space such that the probability that a test point drawn from underlying probability distribution P lies outside of S is bounded by some a priori specified ν between 0 and 1. We approach this problem by trying to estimate a function f which is positive on S and negative on the complement [3].

First we introduce the definition of a multi-dimensional quantile function. Let $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$ be i.i.d. random variables in a set X with distribution P. Let C be a class of measurable subsets of x and let λ be a real-valued function defined on $\mathcal{C}$. The quantile function with respect to (P, λ, C) is

$$U(\alpha) = \inf\{\lambda(C) : P(C) \geq \alpha, C \in \mathcal{C}\} \quad 0 < \alpha \leq 1$$

The most common choice of λ is Lebesgue measure, in which case $C(\alpha)$ is the minimum volume that contains at least a fraction α of the probability mass. The paper [3] describes an algorithm which finds regions close to $C(\alpha)$. And the class C is defined implicitly via a kernel k as the set of half-spaces in a SV feature space. In terms of multi-dimensional quantiles, we take $\lambda(C_\omega) = \|\omega\|^2$ where $C_\omega = \{x : f_\omega \geq \rho\}$. Instead of minimizing the volume of C in input space, the algorithm minimizes a SV style regularizer by using a kernel, which controls the smoothness of the estimated function describing C. To separate the data set from the origin, we solve the following minimization equation:

$$\min_{\omega \in F, \boldsymbol{\xi} \in \mathbb{R}^\ell, \rho \in \mathbb{R}} \quad \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu\ell}\sum_i \xi_i - \rho \quad \text{subject to} \quad (\omega \cdot \Phi(\mathbf{x}_i)) \geq \rho - \xi_i,\ \xi_i \geq 0$$

Here, $v \in (0, 1]$ is upper bounded by the fraction of outliers and lower bounded by the fraction of support vectors. Let Φ be a feature map $X \to F$, i.e. a map into an inner product space F such that the inner product in the image of Φ can be computed by evaluating some simple kernels $K(x, y) = (\Phi(x), \Phi(y))$.

By solving the minimization equation above, we get ω and ρ which are called weight vectors controlling the hyperplane in the feature space associated with kernel, then the decision function will be positive for most data in the training set.

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

Introducing a Lagrangian method on minimization equation, the decision function becomes a kernel expansion:

$$f(x) = \text{sgn}\left(\sum_j \alpha_i k(x_i, x) - \rho\right)$$

ρ is computed below. First we get value of α_i by solving the dual problem of minimization equation:

$$\min_{\alpha} \frac{1}{2} \sum_{\boldsymbol{v}} \alpha_i \alpha_j k\left(\mathbf{x}_i, \mathbf{x}_j\right) \quad \text{subject to} \quad 0 \leq \alpha_i \leq \frac{1}{\nu \ell}, \ \sum_i \alpha_i = 1$$

Then we recover ρ below:

$$\rho = \left(w \cdot \Phi\left(\mathbf{x}_i\right)\right) = \sum_j \alpha_j k\left(x_j, x_i\right).$$

To get expansion coefficients α_i, the paper proposed quadratic program for computing regions that capture a certain fraction of the data. The strategy is to break up the constrained minimization equation into the smallest optimization steps possible.

To conclude, we first establish function f that takes the value $+1$ in a 'small' region capturing most of the data points, and -1 elsewhere. The strategy is to map the data into the feature space by function Φ corresponding to the kernel, and to separate them from the origin with maximum margin. It is regularized by controlling the length of the weight vector in an associated feature space. The expansion coefficients are found by solving a quadratic programming problem. For a new point x, the value $f(x)$ is determined by evaluating which side of the hyperplane it falls on in feature space, and when the data set is separable, then the hyperplane is unique. Meanwhile, the theorem given in the paper gives confidence that ν and the width of the RBF kernel are suitable parameters to adjust.The larger the width of the kernel, the fewer support vectors are selected and the description becomes more spherical. In practice, grid search is often used to to get the optimal (ν, γ) pair for classification.

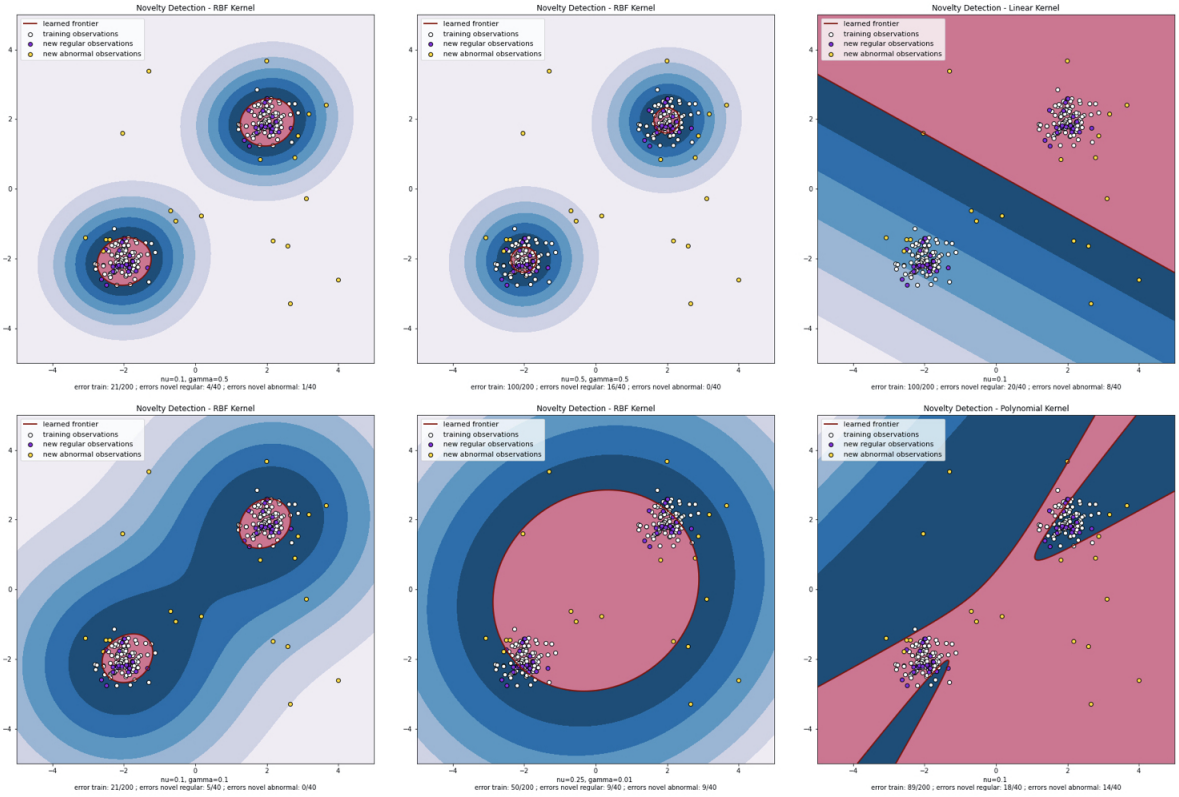

Figure 1: SVM with RBF kernel using different ν, γ and other kernels(poly and linear); (Generated Data)

The main contribution of this paper is that they propose an algorithm that has tractable computational complexity even in high-dimensional cases, is memory efficient because of the support vectors and is versatile in kernel choosing. Using the Gaussian kernel instead of the Polynomial kernel results in tighter descriptions, but it requires more data to support more flexible boundaries. The disadvantages are that if the number of features is much greater than the number of samples, avoiding over-fitting in choosing Kernel functions and regularization term is crucial and SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

2.2 Isolation Forest

Isolation Forest relies on identifying anomalies as the isolated points in the data set. The principle of this algorithm is the following : assuming that anomalies are rare and very different from the normal data, it should be easy to separate them from the rest of the sample through random partitioning.

The ease of separation of a data point is determined by the number of random splits necessary to isolate it, which is equivalent to its depth in a random isolation tree. Therefore, when an isolation forest produces short paths from the root of the tree for some points, these points are more likely to be outliers.

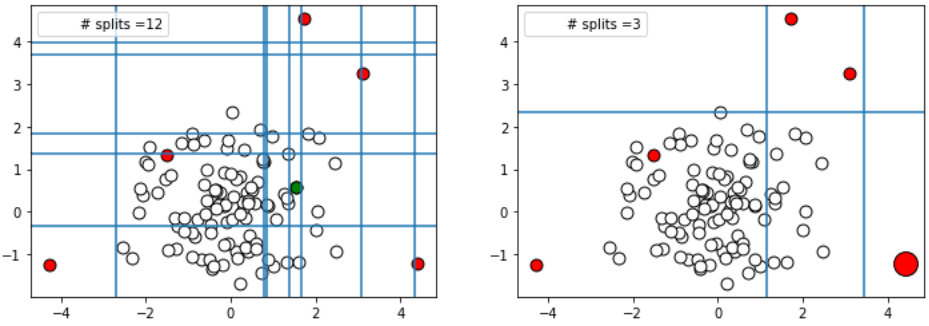

Figure 2: Number of splits needed for isolating a data point
green point (normal): Normal$((0,0), I_2)$
big red point (outlier): Uniform$([-5,5]^2)$

Isolation Forest constructs m isolation trees, each one with a sub-sample of the data of size n randomly selected. Then, the isolation tree is constructed by splitting these data points over a split value of a randomly selected attribute, the split value also selected at random.

The expected depth of a data point $\mathbb{E}[d(x)]$ in the random forest is compared to $c(n)$, the average depth in a binary tree with n leaves. This ratio will determine the following anomaly score for an instance x :

$$s(x,n) = 2^{-\frac{\mathbb{E}[d(x)]}{c(n)}}$$

A small $\mathbb{E}[d(x)]$ implies a large $s(x,n)$, so Isolation Forest will designate the fraction of the data with the highest anomaly score as outliers.

Sub-sampling the data allows this algorithm to work efficiently on small and big data set. Indeed, since the outliers should need few splits, we can abort the partitioning past some threshold. This allows Isolation Forest to be memory and time efficient. Using sub-samples also doesn't make us lose information, in fact it can even help prevent swamping (wrongly identifying normal instances as anomalies) and masking (a cluster of anomalies concealing their presence). Furthermore, since Isolation Forest doesn't use any distance or density measure, it can be implemented in high dimensions with good results.

3 Application on a Data Set

We tested the SVM model on a popular data set for anomaly detection named Mulcross. It generates a multi-variate normal distribution with a selectable number of anomaly clusters as following: contamination ratio = 10% (number of anomalies over the total number of points), distance factor = 2 (distance between the center of normal cluster and anomaly clusters), and number of anomaly clusters = 2. There are 262144 points and 4 features (Figure [3]). We split the data into 66% training set and 33% testing set.
For the model, we use the Radial Basis Function (RBF) kernel SVM with two hyper-parameters (Figure [1]):

- ν an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors. It trades off correct classification of training examples against maximization of the decision function's margin.
- γ defines how far the influence of a single training example reaches. It can be seen as the width of the kernel, the fewer support vectors are selected the more the description becomes spherical. If γ is too large, the radius of the area of influence of the support vectors only includes the support vectors themselves and no amount of regularization with ν will be able to prevent over-fitting.

We tune these parameters using 5-Fold Cross-Validation on a random 10% sample of the training set (because of running time even if it is computed in parallel on each CPU core) and got $\gamma = 0.0001$ and $\nu = 0.25$ (Appendix [4]). Note that the data imbalance (90%/10%) for the two classes is conserved in all sets.
After fitting the model on the training set with the last parameters, we predicted the classification on the testing set and got the following results:

- Global Miss-Classification error = 14.89% - Accuracy = 85.11%
- Normal Points in the testing set = 89.99% - Refused Normal = 16.55%
- Outliers in the testing set = 10.01% (8733 points) - Accepted Outliers = 0%

As a remark, we could increase the size of the sample for CV and the number of folds at the expense of computation time and forecasting speed but we get a smoother model.

Isolation Forest has also been applied to this data, although it is not adapted to it. Indeed, as seen on Appendix Figure [3], the anomaly clusters are quiet dense, and IF will most likely labelled them as normal instances. Actually, we have the following results : Global Miss-Classification error : 6.92%, Refused Normal : 3.82% and Accepted Outliers : 34.67%.
This type of error/miss-classification can be reduced by using a smaller sub-sampling size than the default parameter (128 instead of 256), so that the data points are less close to one another, we will then get the results : Global Miss-Classification error : 3.87%, Refused Normal : 2.14% and Accepted Outliers : 19.38%.

4 Conclusion

Anomaly Detection is a powerful tool. There are several techniques available that can be implemented, but choosing which one to use will depend on the data set and the end goal. The two algorithms presented in this paper have different approaches to solve this problem :

One-Class SVM tries to find the region concentrating most of the data, by projecting the data on a feature space with some kernel function. The hyper-parameters are critical and the computation is expensive.

On the other hand, Isolation Forest identifies the isolated points as outliers, with isolation determined as the depth in a isolation tree. The algorithm is really quick of execution and can be used on small and big data alike. Such a method doesn't give satisfactory results if the anomalies form a cluster or are too similar (close) with the normal data.

References

[1] Michael G.Madden and Shehroz S.Khan. "One-Class Classification: Taxonomy of Study and Review of Techniques". In: (2013).

[2] Fei Tony Liu, Kai Ming Ting, and Zhi-hua Zhou. "Isolation Forest". In: *In ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. IEEE Computer Society*, pp. 413–422.

[3] John Platt et al. "Estimating the Support of a High-Dimensional Distribution". In: (1999).

Appendix

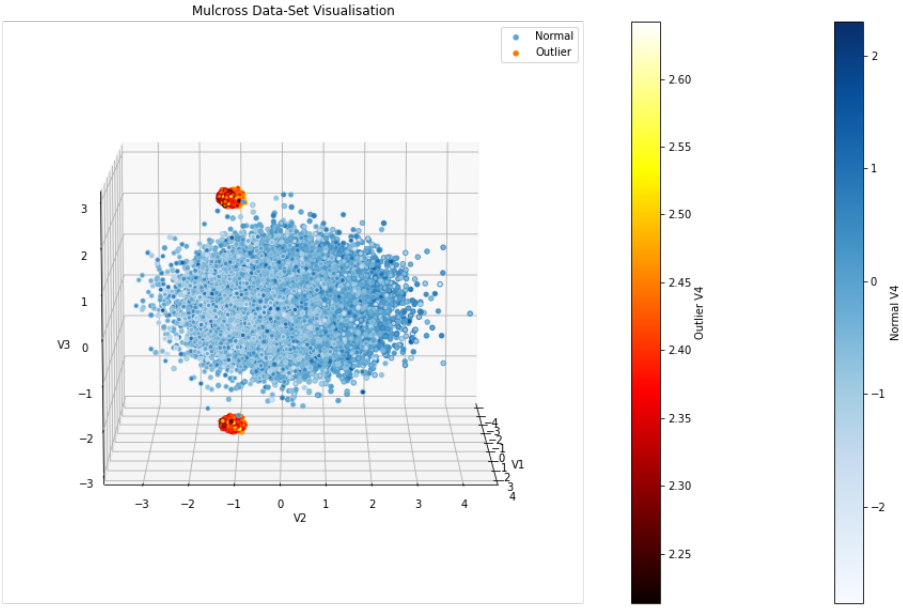

Figure 3: Mulcross Data-Set 4D Visualisation
V1,V2,V3,V4 : features

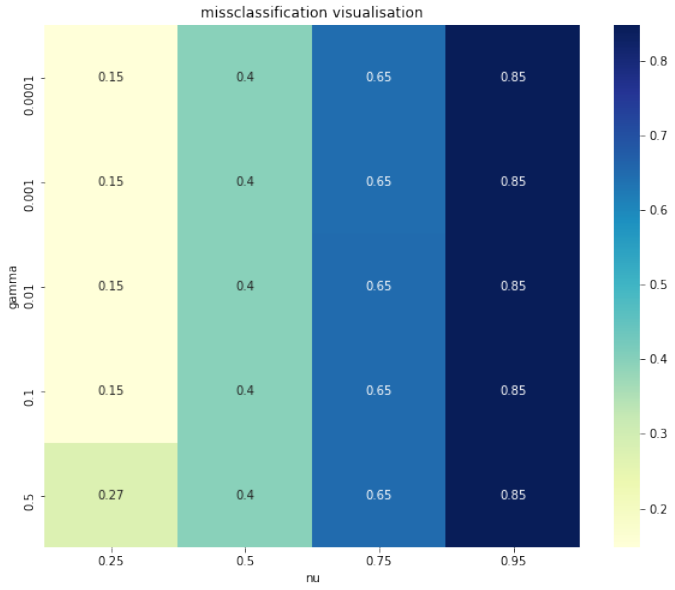

Figure 4: Mulcross 5-fold CV hyper-parameters/miss-classification heatmap

We also tested Isolation Forest and Support Vector Machine with poly, linear and RBF kernel on several data sets, of different dimensions and sizes. It confirms the versatility of Isolation Forest and time efficient compared to SVM. The accuracy of SVM varies among different kernels when RBF kernel performs best. Here, all the data sets are labelled and split with a 80-20 train to test ratio. The model is fit to the unlabelled training set and then tested on the testing set. We have the following results :

The Breast Cancer data is obtained by observing the cell of some breast mass. Each feature represents a characteristic (size, texture, ...). The instances are classified according to the diagnosis benign or malignant. In our example, we use anomaly detection to identify the cancerous cell, much rarer than the sane ones.

	isolation tree	SVM with linear	SVM with poly	SVM with RBF
Refused Normal	5.26%	39.08%	44.83%	11.49%
Accepted Outliers	4.44%	100%	100%	8%
Accuracy	95.00%	38.69%	35.04%	89.78%

Table 1: Breast Cancer dataset with 699 data(63.5% normal data and 36.5% outliers) at 9 dimension

The Forest cover dataset is originally a multiclass classification dataset. It predicts the type of forest cover from cartographic variables (soil type, ...). Here we consider the least represented class as an outlier class.

	isolation tree	SVM with linear*	SVM with poly*	SVM with RBF*
Refused Normal	14.39%	87.07%	87.91%	22.24%
Accepted Outliers	24.87%	0%	0%	21.20%
Accuracy	85.49%	13.85%	13.01%	77.77%

Table 2: Forestcover dataset with 286048 data(98.95% normal data and 1.05% outliers) at 10 dimension

* The SVM algorithm is stopped at the iteration 100 because we saw that the accuracy isn't improving enough compared to the increase in computation time even if the data is re-scaled using standard or min-max scaler. One possibility is that SVM struggles with high number of features but further analysis and reading should clarify this. The polynomial kernel has degree 3 for every model. The data is re-scaled to (0,1) using min-max scaler for SVM.