Learning Outlier Ensembles:
The Best of Both Worlds – Supervised and Unsupervised

Barbora Micenková[1,2]
barbora@cs.au.dk

Brian McWilliams[2]
brian.mcwilliams@inf.ethz.ch

Ira Assent[1]
ira@cs.au.dk

[1]Aarhus University
Aabogade 34
Aarhus, Denmark

[2]ETH Zürich
Universitätstrasse 6
Zürich, Switzerland

ABSTRACT

Years of research in unsupervised outlier detection have produced numerous algorithms to score data according to their exceptionality. However, the nature of outliers heavily depends on the application context and different algorithms are sensitive to outliers of different nature. This makes it very difficult to assess suitability of a particular algorithm without *a priori* knowledge. On the other hand, in many applications, some examples of outliers exist or can be obtained in addition to the vast amount of unlabeled data. Unfortunately, this extra knowledge cannot be simply incorporated into the existing unsupervised algorithms.

In this paper, we show how to use powerful machine learning approaches to combine labeled examples together with arbitrary unsupervised outlier scoring algorithms. We aim to get the best out of the two worlds—supervised and unsupervised. Our approach is also a viable solution to the recent problem of outlier ensemble selection.

Keywords

Outlier detection, outlier ensembles, semi-supervised outlier detection, feature construction

1. INTRODUCTION

Unsupervised outlier detection algorithms [2, 7] aim to reveal extraordinary data points in a data collection. Their original application is in pure data exploratory tasks (e.g., astrophysics, molecular biology) where almost no prior knowledge about the nature of outlierness is available and the goal is to find surprising patterns. Based on geometrical properties of the data (mostly distances and density), these algorithms assign a real-valued outlierness score to each data point thus enable a final outlier ranking.

In many applications, however, the semantics of outliers are known in advance, but not all the possible forms that they can take (e.g., in detection of fraud, intrusions, mislabeled data, measurements errors or faults). In these tasks, a small number of previously seen outliers is available in addition to a large number of unlabeled (mostly normal) data. In such a situation, unsupervised algorithms often perform poorly because there is no principled way how they can take advantage of these extra labels.

We present a new paradigm for outlier detection, *semi-supervised outlier detection*, that combines both the information from unlabeled data and supervision of some labeled data. Through a powerful learning technique, we aim at getting the best out of the two worlds—supervised and unsupervised. The strength of the proposed concept is in its universality because any existing unsupervised outlier scoring algorithm can be adapted, and, similarly, different machine learning approaches can be integrated. It also means that the concept opens a promising field of future research.

The key idea is to use the output scores of multiple unsupervised outlier detection algorithms as *transformed features* for learning with an extreme class imbalance. This transformation step is at the core of our solution. From the original attribute space, we move the problem to a transformed space where the dimensions are different types of exceptionality. In this exceptionality feature space, we employ an ensemble approach to learn feature weights and thus appropriately integrate the supervised and unsupervised information. In our initial setup, we use an ensemble of logistic regressors to learn the feature weights. Logistic regression is a convenient choice since (with an appropriate regularizer) it allows for a sparse solution where many weights can be drawn to zero. Furthermore, it outputs probabilities that can directly be used as outlier scores. This makes for an easily interpretable and verifiable result. Considering that the transformed features correspond to specific outlier detection algorithms with particular settings, the learnt feature weights can be interpreted as an outlier ensemble. We will further discuss the parallels to outlier ensembles and also the strengths and limitations of the approach (Sec. 4, 5).

The paper is organized as follows. We present our initial setup in Sec. 2 and preliminary results on two data sets in Sec. 3. After a brief description of the related work in Sec. 4, we devote an extended space to the discussion and future work in Sec. 5.

2. METHOD

Let $X \in \mathcal{X}$ be a set $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ of data points, and let $L = \{0, 1\}$ be a set of labels where 1 corresponds to an outlier and 0 to a normal data point. The number of outliers in X is much smaller than the number of normal data points. Let $l \ll n$ points in X be labeled outliers, the

ODD^2*'14,* August 24th, 2014, New York, NY, USA.

rest is unlabeled. For the sake of training, we assign label 0 (normal data class) to all unlabeled data points even if we know that some of them will be wrong. The goal is to predict labels of unlabeled and/or previously unseen data points. Since we are interested in ranking outliers according to the degree of their exceptionality, we require probabilities to be output together with the labels [9].

2.1 Logistic Regression with ℓ_1 Penalty

We instantiate our concept using logistic regression, which is a statistical classifier that models the outcome of a binary random variable, y. The probability of a data point belonging to class 1 is modeled as a linear function of variables (features) and a parameter vector, $\beta \in \mathbb{R}^d$ using the logistic function:

$$p(y = 1|x; \beta) = \frac{1}{1 + \exp(-\beta^\top x)} = \sigma(\beta^\top x). \quad (1)$$

The classifier predicts 1 if $\sigma(\beta^\top x) > 0.5$ and 0 otherwise.

To fit the parameters β based on a set of observations X and true labels y, the following cost function needs to be minimized:

$$J(\beta) = -\frac{1}{n}[\sum_{i=1}^{n} y_i \log \sigma(\beta^\top x_i) + (1 - y_i) \log(1 - \sigma(\beta^\top x_i))] + \frac{\lambda}{2n} \sum_{j=1}^{m} |\beta^j|. \quad (2)$$

The first term is the usual logistic loss, $f(\beta)$. The second term in Eq. (2) performs ℓ_1 regularization often referred to as Lasso [14]. It penalizes models with extreme parameter values and shrinks many of them to 0 which has the advantage here of removing features that do not contribute to the outlier scoring task.

Eq. (2) can be minimized using stochastic coordinate descent (SCD) methods [13]. Briefly, SCD picks features uniformly at random and updates the corresponding coordinate of β until convergence as follows

$$\beta^j = s_{4\lambda}(\beta^j - 4\nabla_j f(\beta)),$$

where $s_\lambda(z) = \text{sign}(z) \max(|z| - \lambda, 0)$ is the soft thresholding operator and $\nabla_j f(\beta)$ is the derivative of the loss with respect to the j^{th} coordinate of β.

2.2 Feature Transformation

Instead of applying the logistic regression on the original data, we extract new features. We transform $X \in \mathcal{X}$ into $\Phi(X)$ in the following way. Let $\Phi = \{\Phi_1, \ldots, \Phi_m\}$ be a set of outlier scoring functions where $\Phi_i : \mathcal{X} \to \mathbb{R}^n$. Each $\Phi_i \in \Phi$ is applied to X. If the original data matrix is $X = [x_1, \ldots, x_n]^\top$, the transformed data matrix is

$$\begin{bmatrix} \Phi_1(X) & \cdots & \Phi_m(X) \end{bmatrix} = \Phi(X). \quad (3)$$

Instead of the original data set, we now work with the transformed data set $\Phi(X)$ in the outlier score feature space.

To form the set of functions Φ, we may use any existing unsupervised outlier detection algorithm. Besides using distinct algorithms, we may also use the same algorithm under a perturbation, e.g., change of parameter settings, a distance metric, different subspaces etc. The goal is a setup where different aspects of outlierness are captured by the different scoring functions to span the transformed feature space.

2.3 Re-sampling by Bagging

A challenge for logistic regression lies in the class imbalance. To combat this issue, we propose to use a standard re-sampling method: bootstrap aggregating, also known as *bagging* [4]. Bagging produces multiple versions of a model by training on different bootstrap samples of the training set. Then, an aggregated model is acquired by averaging the outcomes of all versions (for the case of logistic regression it is output probabilities).

A bootstrap sample is a subset of data that is sampled uniformly with replacement. However, in our case we select approximately the same number of outliers as of (contaminated) normal data points. Bagging can help us achieve class balance through downsampling the majority class without losing much information. Since there is a minority of outliers, the same points will get selected multiple times while the normal data class will substantially differ across samples. This also is a reasonable setting considering the semi-supervised setup of the problem where the normal class is contaminated and thus its labels are unreliable.

An appealing property of ℓ_1-regularized logistic regression is the ability of directly performing feature selection by shrinking parameters. In practice, however, the regularization parameter must be carefully chosen. *Stability selection* [11] combines ℓ_1 penalized methods with bootstrap sampling by tracking the proportion of times a particular feature is selected across all of the subsamples. For a large enough number of bootstrap samples, this can be considered as the probability that a given feature belongs in the model.

3. EXPERIMENTS

We compare our approach on two data sets with two different baselines, reporting standard outlier detection evaluation measures: the ROC curve and the area under the ROC curve (AUC).

3.1 Competitors

We present results of two versions of our algorithm, `proposed` and `proposed+`. `proposed` only uses the transformed features while `proposed+` combines both the original and transformed features. For the first two baselines, we use the same training setup as for the proposed algorithms but we train merely on the original set of features. The difference is that `base1:orig` uses the same partially labeled training set as our method while `base2:sup` is trained on fully labeled data (it gets more information than our method and likely more than there would be available in practice). Strictly speaking, `base2:sup` is not a baseline, but we include it to demonstrate the strength of our approach.

As another baseline, `base3:ens`, we adopt a recent outlier ensemble algorithm from Schubert *et al.* [12]. It is an unsupervised approach that builds a binary target vector based on the rankings of the ensemble members and then greedily selects a subset of them to maximize weighted Pearson correlation to the target vector. For an appropriate comparison, we adapt their method such that we build the target vector from the partially labeled training set instead of their proposed heuristic to make the supervision available to this approach as well.

3.2 Data

For our experiments, we use two different data sets. The

synthetic `letter` data set is derived from the UCI letter recognition data set where letters of the alphabet are represented in 16 dimensions [3]. To get data suitable for outlier detection, we subsample data from 3 letters to form the normal class and randomly concatenate pairs of them so that their dimensionality doubles. To form the outlier class, we randomly select few instances of letters that are not in the normal class and concatenate them with instances from the normal class. The concatenation process is performed in order to make the detection much more challenging as each outlier will also show some normal attribute values. In total, we have 1500 normal data points and 100 outliers (6.25% outliers) in 32 dimensions.

The real-world `speech` data set consists of 3686 segments of English speech spoken with different accents.[1] The majority data corresponds to American accent and only 1.65% corresponds to one of seven other accents (these are referred to as outliers). The speech segments are represented by 400-dimensional so called *i-vectors* which are widely used state-of-the-art features for speaker and language recognition [8]. It is a subset of data described in [6].

We have made both data sets publicly available.[2]

3.3 Learning Setup

We split available data to a 60% training and 40% testing set. To simulate the semi-supervised scenario, we remove *half* of the outlier labels from the training set and consider them unlabeled data (which we treat as a contaminated normal class in our setup). For bagging, we construct 50 balanced samples from the training set, learn a logistic regressor on each of them and combine their outputs. At this point, no regularization has been applied for any method so that the results are more easily comparable.

For the sake of feature transformation we use a combination of established unsupervised outlier detection techniques: feature bagging [10] (selecting random subsets of features), k-NN outlier (compute sum of distances to k nearest neighbors) and LOF [5] (scores data based on local density). Precisely, it is feature bagging with LOF for the `letter` data set (50 random bags) and feature bagging with $k = 1$ and Canberra distance for the `speech` data (20 random bags). These settings are based on a coarse search for unsupervised algorithms that perform reasonably well on the training set. It is a mere starting point and alternatives should be investigated.

3.4 Results

In Fig. 1, we report the ROC curves for the `letter` data set. `proposed` and `proposed+` outperform the baselines (notice that they do especially well in the beginning of the ROC curve which is particularly important for real applications). We can see that they can even beat a fully supervised setup with original features (`base2:sup`) and that the most viable competitor is the ensemble of outlier detectors (`base3:ens`). Clearly, the outliers are better separable in the transformed than in the original domain. Fig. 2, on the other hand, shows the ROC curves for the `speech` data set. Here, the `proposed` approach performs comparably to the baselines (except for `base2:sup` which has access to all labels in the training set),

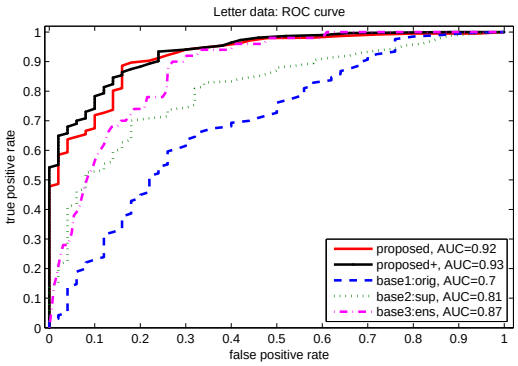

Figure 1: Letter data: ROC curves and the corresponding AUC values.

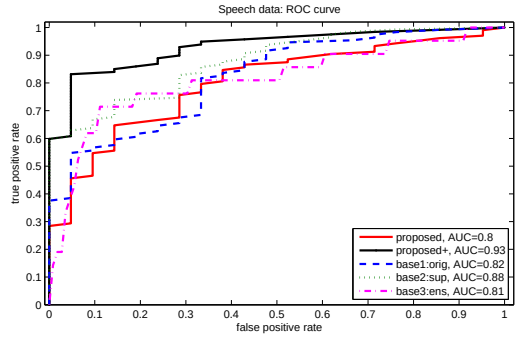

Figure 2: Speech data: ROC curves and the corresponding AUC values.

with only 20 features compared to the original 400. This suggests that both the original and the transformed domain are useful in detection of outliers and explains why `proposed+` that combines them has a superior performance to all other techniques with a great margin. The ROC curves are based on a single run; to show that there indeed is a trend, we report ROC AUC averaged over 20 runs with different train/test splits in Table 1. The proposed method with both original and transformed features outperforms all the other techniques, even the supervised setup. Undoubtedly, the performance improvement depends on the quality and diversity of the input pool of detectors that are used to construct the transformed features. Besides that, additional improvement is expected when applying regularization. For these preliminary experiments, we did not study these aspects in detail but the current results already show a great potential.

Table 1: Average ROC AUC

Method	Letter	Speech
proposed	0.926	0.783
proposed+	**0.942**	**0.875**
base1:orig	0.766	0.814
base2:sup	0.806	0.857
base3:ens	0.881	0.774

[1]The authors would like to thank to the Speech Processing Group at Brno University of Technology, Czech Republic, who provided us with the data.

[2]Download the data sets at http://goo.gl/mGg8ti?gdriveurl.

4. RELATED WORK

Outlier Ensembles

It has been shown that an appropriate combination of multiple algorithms (later called detectors) can increase outlier detection performance [12] which has recently triggered a wide interest in outlier ensembles [1, 16]. The problem of selecting (building/weighting) an ensemble is complex and hard to do in completely unsupervised settings. Open questions concern, e.g., the tradeoff between accuracy of the single detectors and their diversity, correlation among them or the method to combine their outputs [16]. The problem is magnified by the fact that outputs of different scoring algorithms are on different scales and often cannot be interpreted as outlier probabilities [9]. The approach outlined in this paper could be viewed as an ensemble selection technique where the few provided labels guide the selection process, giving an elegant solution to the above stated problems.

Class-Imbalance Learning

In addition, the proposed method complements machine learning literature on classification of extremely imbalanced data sets. Common approaches to class-imbalance learning such as sampling, bagging and boosting, one-class classification and cost-sensitive learning (references to be found e.g. in [2, 15]) can readily be complemented by the proposed scheme. Outputs of unsupervised outlier detectors can be interpreted as additional non-linear features to enhance classification in a similar spirit as kernels do. We expect that the proposed concept will be useful in situations where the rare class is not well clustered but more investigation must be carried out on this topic to confirm this hypothesis.

5. DISCUSSION AND FUTURE WORK

We have presented a new concept of semi-supervised outlier detection that is useful in applications where some outlier examples are available on top of a vast amount of unlabeled data and where new types of outliers might occur in the future. We believe that this scenario is realistic for both research and commercial applications. The idea combines unsupervised outlier detection with established machine learning techniques for classification in the transformed outlier score space. Initial experiments indicate that if we add outputs of unsupervised outlier detection algorithms as new features to the original training data, we can get a superior performance to both unsupervised and supervised techniques. However, numerous challenges to both outlier detection and machine learning researchers remain.

Some of the favourable properties of the concept are:

- any outlier scoring function can be used as a feature,
- a well-interpretable result as each feature corresponds to a specific outlier detection algorithm with particular parameter settings,
- ℓ_1-regularization can eliminate most of the features,
- output scores can directly be interpreted as outlier probabilities,

Using appropriate outlier detection algorithms and thus getting good transformed features is crucial for the performance and it is the same challenge that outlier ensemble theory faces [12]. Our limitation compared to classical supervised methods is that computing the transformed features is relatively slow. Since the same process needs to be applied at prediction time, we want to discard as many features as possible already in the training phase via regularization. This requires further investigation.

Many other new directions are open for future work. E.g., it is worth trying to apply this learning setup to select the best parameter settings for a single algorithm. It also seems to be a good framework for developing active learning strategies and interactive detection algorithms. Further, it should be clarified in which applications and machine learning setups the proposed technique is viable.

6. ACKNOWLEDGEMENT

Part of this work has been supported by the Danish Council for Independent Research—Technology and Production Sciences (FTP), grant 10-081972.

7. REFERENCES

[1] C. C. Aggarwal. Outlier ensembles: position paper. *SIGKDD Explorations*, 14(2), 2012.

[2] C. C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[3] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[4] L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2), 1996.

[5] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *SIGMOD'00*, 2000.

[6] N. Brummer, S. Cumani, O. Glembek, M. Karafiát, P. Matějka, J. Pešán, O. Plchot, M. M. Soufifar, E. V. de, and J. Černocký. Description and analysis of the Brno276 system for LRE2011. In *Proc. of Odyssey 2012: The Speaker and Lang. Rec. Workshop*, 2012.

[7] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *CSUR*, 41(3), 2009.

[8] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Trans. on Audio, Speech & Lang. Proc.*, 19(4), 2011.

[9] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Interpreting and unifying outlier scores. In *SDM'11*, 2011.

[10] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *KDD'05*, 2005.

[11] N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Stat. Society. Series B*, 72(4), 2010.

[12] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel. On evaluation of outlier rankings and outlier scores. In *SDM'12*, 2012.

[13] S. Shalev-Shwartz and A. Tewari. Stochastic methods for l_1-regularized loss minimization. *The Journal of Machine Learning Research*, 12, 2011.

[14] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Stat. Society*, 1996.

[15] M. Wasikowski and X. wen Chen. Combating the small sample class imbalance problem using feature selection. *IEEE Trans. on Knowledge and Data Engineering*, 22(10), 2010.

[16] A. Zimek, R. J. G. B. Campello, and J. Sander. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *SIGKDD Explorations*, 15(1), 2013.