Chapter 7 - Supervised Outlier Detection

1 Introduction

Supervised outlier detection is harder than classification. The outlier class tends to be small compared to the rest of the data. The normal class is contaminated with some outliers (we usually only label points as outliers and assume the rest of the points are mostly normal points). We also may not have all possible anomalies in the outlier class (e.g. in intrusion detection, we might not know all possible intrusion methods).

You can first run unsupervised learning algorithms and feed them as features to the supervised learning problem. We also need to do some active learning, where we identify promising points to send to labelers for labeling. We also discuss how to use regression modeling (predict one attribute from the rest and combine the error scores to create outlier score).

2 Full Supervision: Rare Class Detection

False negatives are worse than false positives (i.e. it is better to mistakenly call a point an outlier than it is to miss an outlier). We cannot use the accuracy metric because classes are unbalanced. We must use cost-sensitive learning. Here, we weight the normal class (label 1) less than the outlier classes (labels $2, ..., K$) - the weighting $1/N_i$ for class i is popular. Alternatively we can use adaptive resampling where we oversample the rare case (or even better, undersample the normal class) to build our dataset.

In the MetaCost approach, we train our classifier and score each point with the probability that it belongs to class i ($p_i(\bar{X})$). The misclassification cost is then $\sum_{i \neq r} c_i p_i(\bar{X})$, where r is the true class. We relabel it with the class that minimizes the misclassification cost. Many classification models output scores, so you can use that for $p_i(\bar{X})$. For binary outputs, use bagging with unstable detectors and small samples to get a score. This bagging approach isn't perfect, because the components are correlated.

In weighting methods, we modify the training algorithm so that it can support a weight (misclassification cost) for each training example. We can do this for the Bayes Classifier, Proximity-Based Classifiers, Rule Based Classifiers, Decision Trees, and Support Vector Machines. Weighting works better than sampling because it retains more information. It is slower than sampling though.

Undersampling the normal class can achieve similar effects to weighting and also yields a smaller dataset that is easier to train on. If you ensemble (10 to 25 components), you can acheive accuracy similar to weighting.

Oversampling is problematic because you can duplicate points and thus overfit. To mitigate this, you can use the SMOTE approach. Basically, to oversample a point, you randomly pick a nearest neighbor and sample a point on the line segment to that nearest neighbor. You can boost SMOTE and ensemble it.

Adaboost classifies the dataset many times, updating the point weights each time based on whether they were misclassified in the previous round. In round t, the weight of point i is $D_t(i)$ (it starts at $1/N$). The weight of the next round is $D_{t+1}(i) = D_t(i)e^{\alpha_t}$ if point i is correctly classified and $D_{t+1}(i) = D_t(i)e^{-\alpha_t}$ otherwise. $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$ and ϵ_t is the fraction of incorrectly sampled points on a weighted basis. The Adacost algorithm extends this for outlier detection by replacing α_t with either

$\beta_-(c_i)\alpha_t$ or $\beta_+(c_i)\alpha_t$, dependeing on whether the point is correctly classified, to adjust point weights based on their misclassification cost c_i. If you combine boosting with SMOTE, you get SMOTEBoost. Use high bias, low variance classifiers with boosting because boosting reduces bias. Don't use boosting on very noisy datasets.

3 Semi-Supervision: Positive and Unlabled Data

Sometimes, negative (i.e. normal) examples are hard to define. How do you define a representative sample of non-spam emails, for example? Additionally, the normal class may have some outlier class points in it (contamination). One way to deal with this is to use heuristics to pick good normal examples that are not contaminants (we can also just give a normalness weight if we do not want to pick a subset of the unlabeled data as the normal one). We can also learn the normalness weights with a probabilistic model.

4 Semi-Supervision: Partially Observed Classes

Sometimes we don't know all the different kinds of anomalies. This is common in adversarial settings because attackers develop new strategies. One approach here is to train a model on the outliers only and consider ouliers to be the points most similar to them. In addition, we train a model on the normal points and consider outliers to be the points least similar to them. Let's look at each of these.

Sometimes, we do not have normal examples, we only have anomalies. Proximity based methods are really our only option here. It's difficult, but you might be able to get one-class SVMs to work here too.

Sometimes, we only have normal examples, no anomalies. You can use any outlier detection method here. For some reason this scenario is treated differently from regular outlier detection, and maybe it should not be.

Sometimes, we want to detect new kinds of outliers. In this case, we first check if it is similar to any of our known normal examples or outliers. If so, we do multiclass classification to identify what kind of outlier (or normal class) it is.

5 Unsupervised Feature Engineering in Supervised Methods

You can run unsupervised algorithms (LOF, k-nearest neighbors) and use the resulting outlier scores as features. Some features may be highly correlated, so using L_1 regularization can help.

6 Active Learning

First, identify a some interesting examples. Get those labeled. Then train a model on those, and find more interesting examples. And repeat.

How do you find interesting examples? Pick the most ambiguous examples (i.e. the ones closest to decision boundary) - these are high uncertainty. We also pick points that have low likelihood under our trained model (i.e. these are the outliers). Likelihood is easy to measure, because classifiers output it. How do you measure uncertainty? One approach is to pick points where the probability of being an outlier is greater than the fraction of outliers (you need to estimate this from domain knowledge). Another option is query by committee, where you train an ensemble and pick points that have the greatest disagreement by ensemble components.

7 Supervised Models for Unsupervised Outlier Detection

Make sure to standardize your data first. Go one feature at a time and make it the dependent variable. Learn a regression model and compute the error for each example (with cross validation). We then computed a weighted (by regression quality of fit) average of each error for each data point. That is, assuming the squared error of model k on point $\bar{X}_i$, we have $RMSE(M_k) = \sqrt{\frac{\sum_{i=1}^{N} \epsilon_k^2 \bar{X}_i}{N}}$ and $w_k = 1 - \min(1, RMSE(M_k))$. Thus, $Score(\bar{X}_i) = \sum_{k=1}^{d} w_k \epsilon_k^2(\bar{X}_i)$. Notice that this approach lets you identify the dimensions that contribute most to the error. The random forest is a good base detector. This is more flexible than nonlinear PCA because we use the random forest. Make sure not to do PCA before running this method because PCA will try to make each dimension independent of the others.

Another option is to consider r targets instead of just one. Pick r features, do PCA to get uncorrelated targets, and then train regressors to estimate each of them. Then we get: $Score(S_r, \bar{X}) = \frac{\sum_{k=1}^{r'} w_k \epsilon_k^2(\bar{X})}{r'}$.

You also may omit some features from being targets in this regression approach. If a feature is categorical, use a classifier instead of regressor.

You can also create synthetic outlier data (e.g. randomly sample some features from ranges of extreme values). This can cause high bias models though.

8 Conclusions and Summary

It's typical that your normal class is contaminated and you don't know what all the different kinds of outliers are, but we can handle both of these. Active learning is common here as well.