Chapter 6 - Outlier Ensembles

1 Introduction

Outlier ensemble algorithms can combine many base detectors to create an improved detector. You can also select hyperparameters by trying all of them and creating an ensemble.

When using an ensemble method, you need to pick the base detector and methodology for combining the normalized scores together.

As in classification, we have a bias/variance tradeoff. Variance is how much the outlier score for a point changes depending on which dataset we train our model on (assume the points are sampled i.i.d from some unknown distribution). Bias measures how much our outlier scores deviate from the theoretically optimal scores (assuming they exist). Ensemble methods can reduce both bias and variance.

2 Categorization and Design of Ensemble Methods

A model-centric ensemble uses different models (or the same model with different hyperparameters) as base detectors. A data-centric ensemble uses the same model as base detector, but trains each on a different subset of the dataset.

Ensembles can be independent, where each base detector is run independently. They can be sequential, where one base detector feeds into the next.

Once the base detectors run, how do we combine their scores? We first need to normalize them so that we can compare them. First, we need to decide whether a larger score is more or less anomalous (you can flip the signs of scores based on your decision). Then we need to scale the scores. We can use range based scaling, which produces score $S_j(i)$ for detector j and point i from its original score $s_j(i)$ with

$$S_j(i) = \frac{s_j(i) - min_j}{max_j - min_j} \tag{1}$$

To make our scores less sensitive to the max and min, we can use standardization (even if the scores are not normally distributed, that's ok).

$$S_j(i) = \frac{s_j(i) - \mu_j}{\sigma_j} \tag{2}$$

Another option is to run the EM algorithm on the data points to turn them into probabilities. If you do this, you can get scores that different by orders of magnitude, so you should take a logarithm and use log probabilities as your scores.

To combine scores, you can average them (reduces variance) or take the maximum (can decrease bias, but increase variance).

3 Theoretical Foundations of Outlier Ensembles

The usual bias/variance analysis uses the mean squared error. However we don't have labeled data. We assume our algorithm output scores and oracle outlier scores have zero mean and unit variance. The bias is how much our algorithm output scores differ from the oracle scores. The variance is how much our algorithm output scores change as we randomly pick datasets (to this, you need to designate a set of test points). One way to randomly sample datasets is to take a large set of points and pick random subsets (or you can use a model that makes random choices, like isolation forest). You can then define a bias and variance expression as in classification.

Ensembling can easily reduce variance by averaging different base detector outputs, but it's harder to reduce bias.

4 Variance Reduction Methods

Ensembling by averaging scores always helps reduce variance. If your base algorithm is unstable, the ensemble will be much better than a single detector. That being said, a stable base algorithm might yield a higher overall accuracy (albeit a smaller improvement) when ensembled.

You can ensemble k-nearest neighbors by picking many values of k and averaging scores. More generally, you can avoid picking hyperparameters this way.

If you are using k-means (or histograms), which depends highly on initialization centroids, you can ensemble over many random initializations to reduce variance.

In feature bagging, we pick random subspaces, project points, and score the projection distances. If your subspaces dimensions are small (compared your data dimensionality). ensembling is more effective because each base algorithm is more unstable. Another way to look at this is that each ensemble component looks at the dataset from a different viewpoint and is therefore able to find outliers more easily than a single base detector. Since feature bagging usually uses between $d/2$ and $d-1$ dimensions, base detectors tend to be correlated, which makes the ensemble less useful. Rotated bagging fixes this by randomly rotating the random subspaces.

Isolation forests ensemble isolation trees. The score is the average path length from root to separate out the point. The trees are grown until they fully separate all points (we train each tree on a random subset of the data). We randomly pick a dimension and split point. Isolation forests do well at finding outliers at the edges of the space defined by the data points (i.e. multivariate extreme values). It struggles with outliers near the center of the data points.

For unstable detectors, bootstrap from your original dataset (i.e. randomly sample with replacement). When doing this, make sure you exclude a point (or its copies) when computing k nearest neighbor distances. Each base detector must be trained on a different bootstrapped sample.

If you sample without replacement, you are doing subsampling (not bootstrapping). Bootstrapping works better for isolation forest while subsampling works better for distance methods.

Pick small sample sizes for simple detectors or slow detectors (like k nearest neighbors). You can also pick the sample size randomly. Just randomly pick f between $\min(1, 50/N)$ and $\min(1, 1000/N)$ and sample fN points. Make sure to standardize outlier scores if you do this.

If you do the random sample size selecting, you could use rotated bagging as well. This is computationally efficient and works quite well.

Another variance reduction technique is randomized feature weighting, where we randomly scale the features. You can use wagging, which is bagging, but points are given a random (from Gaussian) weight. You can also use geometric subsampling, which is another way to randomly pick sample sizes.

5 Flying Blind with Bias Reduction

Bias reduction is hard. In classification, we can use boosting, but we can't do that here because we do not have labels. We can roughly do this with some hueristics.

One approach is remove outliers before fitting the detector - but how do we find the outliers? Well, we can fit a detector and remove points with high outlier scores. Then we fit a new outlier detector with the remaining points and keep repeating this.

Another approach is to remove bad ensemble components. First, we compute each points ensemble score and treat these as ground truth. Then we find the detector most correlated with this ground truth and put that in ensemble $\mathcal{L}$. We then iteratively add to $\mathcal{L}$ by adding the detector most correlated with $\mathcal{L}$. Use Pearson correlation here. You can also focus on outlier points only here, which requires you to binarize the scores.

When sampling random subspaces, you can use $HiCS$ or $OUTRES$ to pick them instead of just picking randomly.

When sampling random points, you can keep track of which points are marked as outliers and make them less likely to be chosen in the subsequent sample.

6 Model Combination for Outlier Ensembles

You can average (or take median) of outlier scores - this promotes stability (reduces variance). Alternatively, you can take a maximum (this works better with rank scores), but this does not always work. The maximum can sometimes reduce bias because it can ignore bad detectors and find pretty unique outliers, but it may increase variance (but you can mitigate this by using other variance reduction techniques).

Using rank scores can help avoid extreme scores, but can make it harder to distinguish between points (decrease variance, increase bias).

We can combine averaging and maximization. The Average of Maximums approach divides components into buckets, takes a max within each bucket, and averages over the results. The thresh method takes standardized scores and clips them to minimum value of t (usually $t = 0$) to avoid treating inliers differently from strong inliers. You then add these thresholded standardized scores. This can lead to points that have the same final score, so you break ties using the standardized (non-thresholded) scores.

7 Conclusions and Summary

Ensembles are becoming more popular recently, especially on challenging datasets. We have variance reduction and bias reduction techniques, where the latter are more difficult.