Converting Output Scores from Outlier Detection Algorithms into Probability Estimates

Jing Gao

Dept. of Computer Science and Engineering
Michigan State University
East Lansing,MI 48824
gaojing2@msu.edu

Pang-Ning Tan

Dept. of Computer Science and Engineering
Michigan State University
East Lansing,MI 48824
ptan@cse.msu.edu

Abstract

Current outlier detection schemes typically output a numeric score representing the degree to which a given observation is an outlier. We argue that converting the scores into well-calibrated probability estimates is more favorable for several reasons. First, the probability estimates allow us to select the appropriate threshold for declaring outliers using a Bayesian risk model. Second, the probability estimates obtained from individual models can be aggregated to build an ensemble outlier detection framework. In this paper, we present two methods for transforming outlier scores into probabilities. The first approach assumes that the posterior probabilities follow a logistic sigmoid function and learns the parameters of the function from the distribution of outlier scores. The second approach models the score distributions as a mixture of exponential and Gaussian probability functions and calculates the posterior probabilites via the Bayes' rule. We evaluated the efficacy of both methods in the context of threshold selection and ensemble outlier detection. We also show that the calibration accuracy improves with the aid of some labeled examples.

1 Introduction

Outlier detection has been extensively studied for many years, resulting in the development of numerous algorithms [7, 3, 6]. These algorithms often produce a numeric-valued output score to represent the degree to which a given observation is unusual. In this paper, we argue that it is insufficient to obtain only the magnitude or rank of outlier score for any given observation. There are many advantages to transforming the output scores into well-calibrated probability estimates. First, the probability estimates provide a more systematic approach for selecting the appropriate threshold for declaring outliers. Instead of requiring the user to choose the threshold in an ad-hoc manner, the Bayesian risk model can be employed, which takes into account the relative cost of misclassifying normal examples as outliers, and vice-versa. Second, the probability estimates also provide a more robust approach for developing an ensemble outlier detection framework than methods based on aggregating the relative rankings of outlier scores [9]. Finally, the probability estimates are useful to determine the uncertainties in outlier prediction.

Obtaining calibrated probability estimates from supervised classifiers such as support vector machine (SVM), Naïve Bayes, and decision trees has been the subject of extensive research in recent years [12, 11, 1, 13]. The calibration methods generally fall under two categories: parametric and non-parametric. Parametric methods assume that the probabilities follow certain well-known distributions, whose parameters are to be estimated from the training data. The typical methods used for calibrating classifier's outputs include logistic regression, asymmetric Laplace distribution, and piecewise logistic regression. Non-parametric methods, on the other hand, employ smoothing, binning, and bagging methods to infer probability estimates from the classifier's output scores.

Each of the preceding methods require labeled examples to learn the appropriate calibration function. They are inapplicable to calibrating outlier scores because outlier detection is an unsupervised learning task. Therefore a key challenge in this work is handling the *missing label* problem. Our solution is to treat the missing labels as hidden variables and apply the Expectation-Maximization (EM) algorithm [4] to maximize the expected likelihood of the data. We consider two approaches for modeling the data. The first approach models the posterior probability for outlier scores using a sigmoid function while the second approach models the likelihoods for the normal and outlier classes separately.

Our previous work on semi-supervised outlier detection [5] suggests that adding a small number of labeled examples helps to improve the detection rate and false alarm rate of an outlier detection algorithm. In this paper, we further investigate the benefits of semi-supervised outlier detection in the context of improving the probability estimation of outlier scores by modifying our proposed methods to maximize the joint likelihoods of the labeled and unlabeled data.

In short, our main contributions are summarized below:

1. We develop two calibration methods for transforming outlier scores into probabilities. Unlike existing approaches which are developed for supervised classifiers, our proposed methods do not require labeled examples. Instead, the labels are treated as hidden variables to be learnt together with the model parameters.

2. We devise a semi-supervised method to further improve the calibration of probability estimates.

3. We illustrate the benefits of converting outlier scores into probabilities in the context of threshold selection and ensemble outlier detection. Our results show that a better performance is achieved using the probability estimates instead of the raw outlier scores.

The rest of the paper is organized as follows. Section 2 describes our calibration method using sigmoid function while Section 3 presents an alternative method using a mixture of exponential and Gaussian distributions. Section 4 shows how to extend these methods to incorporate labeled examples. In Section 5, we describe the advantages of using calibrated probabilities for threshold selection and ensemble outlier detection. Experimental results are given in Section 6 while the conclusions are presented in Section 7.

2 Calibration Using Sigmoid Function

Logistic regression is a widely used method for transforming classification outputs into probability estimates. Converting outlier scores, on the other hand, is more challenging because there are no labeled examples available. This section describes our proposed method for learning the labels and model parameters simultaneously using an EM-based algorithm.

2.1 Modeling Sigmoid Posterior Probability

Let $X = \{x_1, x_2, \cdots, x_N\}$ denote a set of N observations drawn from a d-dimensional space, R^d. Suppose the data is generated from two classes: the outlier class O and the normal class M. Let $F = \{f_1, f_2, \cdots, f_N\}$ be the corresponding outlier scores assigned to each observation in X. Without loss of generality, we assume that the higher f_i is, the more likely x_i is an outlier.

Our objective is to estimate the probability that x_i is an outlier given its outlier score f_i, i.e., $p_i = P(O|f_i)$. The probability that x_i is normal can be computed accordingly by $P(M|f_i) = 1 - p_i$. According to Bayes' theorem:

$$\begin{aligned} P(O|f_i) &= \frac{p(f_i|O)P(O)}{p(f_i|O)P(O) + p(f_i|M)P(M)} \\ &= \frac{1}{1+\exp(-a_i)} \end{aligned} \quad (1)$$

where

$$a_i = \log \frac{p(f_i|O)P(O)}{p(f_i|M)P(M)} \quad (2)$$

As shown in [2], a_i can be considered as a discriminant function that classifies x_i into one of the two classes. For a Gaussian distribution with equal covariance matrices, a_i can be simplified to a linear function:

$$a_i = Af_i + B \quad (3)$$

Replacing Equation 3 into 1 yields:

$$p_i = P(O|f_i) = \frac{1}{1+\exp(-Af_i - B)} \quad (4)$$

Our task is to learn the parameters of the calibration function, A and B. Let t_i be a binary variable whose value is 1 if x_i belongs to the outlier class and 0 if it is normal. The probability of observing t_i is

$$p(t_i|f_i) = p_i^{t_i}(1-p_i)^{1-t_i} \quad (5)$$

which corresponds to a Bernoulli distribution. Let $T = [t_i]$ denote an N-dimensional vector, whose components represent the class labels assigned to the N observations. Assuming that the observations are drawn independently, the likelihood for observing T is then given by

$$P(T|F) = \prod_{i=1}^{N} p_i^{t_i}(1-p_i)^{1-t_i} \quad (6)$$

Maximizing the likelihood, however, is equivalent to minimizing the following negative log likelihood function:

$$LL(T|F) = -\sum_{i=1}^{N} \Big[t_i \log p_i + (1-t_i)\log(1-p_i) \Big] \quad (7)$$

Substituting Equation 4 into 7, we obtain:

$$LL(T|F) = \sum_{i=1}^{N} \big[(1-t_i)(Af_i+B) + \log(1+\exp(-Af_i-B)) \big] \quad (8)$$

In supervised classification, since labeled examples $\{x_i, t_i\}$ are available, we may create a training set (f_i, t_i) and learn the parameters of the sigmoid function directly by minimizing the objective function in Equation 8 (see [11] and [13]). Unfortunately, because outlier detection is an unsupervised learning task, we do not know the actual values for t_i. To overcome this problem, we propose to treat the t_i's as hidden variables and employ the EM algorithm to simultaneously estimate the missing labels and parameters of the calibration function. The learning algorithm is presented in the next section.

2.2 Learning Parameters of Sigmoid Function

The EM algorithm is a widely used method for finding maximum likelihood estimates in the presence of missing data. It utilizes an iterative procedure to produce a sequence of estimated parameter values: $\{\theta^s | s = 1, 2, ...\}$. The procedure is divided into two steps. First, the missing label t_i is replaced by its expected value under the current parameter estimate, θ^s. A new parameter estimate is then computed by minimizing the objective function given the current values of $T^s = [t_i^s]$. Table 1 shows a pseudocode of the algorithm.

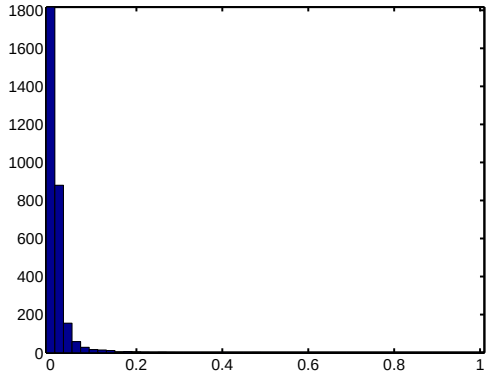
EM algorithm:
Input: The set of outlier scores, $F = \{f_1, \cdots, f_N\}$
Output: Model parameters, $\theta = (A, B)$
Method:
1. $s \leftarrow 0$.
2. Initialize the parameters to θ^0
3. Loop until algorithm converges
   3.1 E-step: Set $t_{ij}^{s+1} = \mathrm{E}(t_{ij}|F, \theta^s))$.
   3.2 M-step: Compute $\theta^{s+1} = \mathrm{argmin}_\theta LL(T|F)$.
   3.3 set $s \leftarrow s + 1$

Table 1. EM algorithm framework

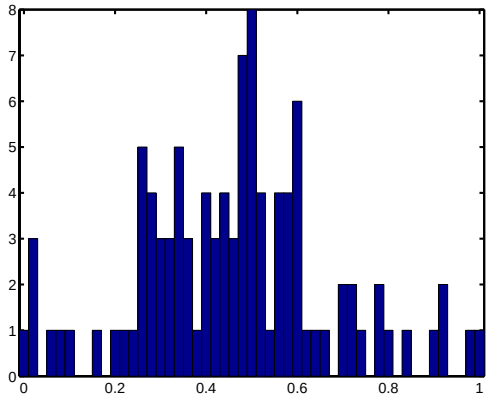
$LL(T|F)$ is the negative log likelihood function to be minimized. During the E-step, the model parameters are fixed while $LL(T|F)$ is minimized with respect to t_i. Since $LL(T|F)$ is a linear function of t_i, it is minimized by setting:

$$t_i = \begin{cases} 1 & \text{if } Af_i + B > 0 \\ 0 & \text{if } Af_i + B \leq 0 \end{cases} \quad (9)$$

During the M step, since $T^s = [t_i^s]$ is fixed, minimizing $LL(T|F)$ with respect to A and B is a two-parameter optimization problem, which can be solved using the model-trust algorithm described in [11].



(a) Outlier Score Distributions for Normal Examples



(b) Outlier Score Distributions for Outliers

Figure 1. Outlier Score Distributions

3 Calibration Using Mixture Modeling of Outlier Scores

Although fitting posterior probabilities into a sigmoid function is simple, the method makes a strong assumption that the outliers and normal examples have similar forms of outlier score distributions. In this section, we present an alternative method for modeling the outlier scores using a mixture of exponential and Gaussian distributions.

3.1 Using Mixture Models to Describe Outlier Score Distributions

Consider a data set containing outliers that are uniformly distributed and normal observations drawn from a Gaussian distribution. We are interested in modeling the distribution of their outlier scores. Suppose the outlier score

is computed based on the distance between a data point to its k-th nearest neighbor. Figure 1 shows the typical outlier score distributions for the outlier and normal classes[1]. Observe that the outlier scores for the normal class tend to have an exponential distribution whereas that of the outlier class seems to follow a Gaussian distribution. This result suggests that a mixture model consisting of an exponential and a Gaussian component may fit well to the outlier score distributions.

At first glance, it may seem quite surprising to observe that the scores for the outlier class follow a Gaussian distribution. In the following, we give a theoretical justification as to why the outliers may indeed have a Gaussian distributed outlier scores.

Theorem 1 *Suppose X and Y are 1-dimensional random variables. If $X \sim U(-\delta, \delta)$ and $Y \sim N(0,1)$, then the distances between examples drawn from X and Y follow a Gaussian distribution.*

Proof. Let Z be the random variable for the distance between X and Y, i.e., $Z = |X - Y|$. Then we need to prove that Z follows a Gaussian distribution. We begin with the cumulative distribution function (c.d.f.) for Z:

$$\begin{aligned} F_Z(z) = P(Z \leq z) &= P(|X - Y| \leq z) \\ &= \iint_R f(x,y)\,\mathrm{d}x\,\mathrm{d}y \end{aligned}$$

where R is the region defined by $|X - Y| \leq z$. If X and Y are independent, their joint probability density function in R is:

$$f(x,y) = f(x)f(y) = \frac{1}{2\delta\sqrt{2\pi}} \exp(-y^2)$$

and zero elsewhere. Therefore

$$\begin{aligned} F_Z(z) &= \iint_R \frac{1}{2\delta\sqrt{2\pi}} \exp(-y^2)\,\mathrm{d}x\,\mathrm{d}y \\ &= \int_{-\delta}^{\delta} \frac{1}{2\delta\sqrt{2\pi}} \int_{-\delta-z}^{\delta+z} \exp(-y^2)\,\mathrm{d}x\,\mathrm{d}y \end{aligned}$$

The c.d.f. for standard normal is often denoted as $\Phi(x)$, where

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-u^2)\,\mathrm{d}u$$

So

$$\begin{aligned} F_Z(z) &= \int_{-\delta}^{\delta} \frac{1}{2\delta} \Big[\Phi(\delta + z) - \Phi(-\delta - z)\Big]\,\mathrm{d}x \\ &= \Phi(\delta + z) - \Phi(-\delta - z) \end{aligned}$$

[1]The shapes of these histograms are consistently observed when the experiment is repeated several times.

The probability density function for Z is obtained by taking the derivative of its c.d.f with respect to z. Because the derivative of $\Phi(x)$ is a Gaussian distribution and a linear combination of two Gaussian distributions is also a Gaussian, therefore, Z must follow a Gaussian distribution. ■

If the outliers are uniformly distributed and the proportion of outliers is considerably smaller than the proportion of normal observations, it is reasonable to assume that the k-th nearest neighbor of an outlier corresponds to a normal observation. Theorem 1 suggests that the distance between a randomly chosen point from a uniform distribution to another randomly chosen point from a Gaussian distribution should follow a Gaussian distribution. While such analysis does not conclusively show that the outlier scores must follow a Gaussian distribution, it does suggest that modeling the scores of outliers using a Gaussian distribution may be quite a reasonable assumption. Furthermore, our empirical results also seem to support this assumption.

Therefore we will use a mixture of Gaussian and exponential distributions for modeling the outlier scores:

$$p_i = p(f_i|O) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(f_i - \mu)^2}{2\sigma^2}) \tag{10}$$

$$q_i = p(f_i|M) = \lambda \exp(-\lambda f_i) \tag{11}$$

where μ, σ, and λ are the parameters of the Gaussian and exponential distributions. The probability of observing t_i can be written as:

$$p(t_i, f_i) = [\alpha p_i]^{t_i} [(1-\alpha) q_i]^{1-t_i} \tag{12}$$

where α is the prior probability of the outlier component Using Bayes' rule:

$$p(t_i|f_i) = \frac{[\alpha p_i]^{t_i} [(1-\alpha) q_i]^{1-t_i}}{p(f_i)} \tag{13}$$

The model parameters $\theta = (\alpha, \mu, \sigma, \lambda)$ are estimated by minimizing the negative log likelihood function:

$$LL(T|F) = -\sum_{i=1}^{N} t_i \log(\alpha p_i) + (1 - t_i) \log((1-\alpha) q_i) \tag{14}$$

Given the estimate of the model parameters, the posterior probability of x_i being an outlier can be computed using Bayes' rule:

$$P(O|f_i, \hat{\theta}) = \frac{\alpha p(f_i|O, \hat{\theta})}{\alpha p(f_i|O, \hat{\theta}) + (1-\alpha) p(f_i|M, \hat{\theta})} \tag{15}$$

3.2 Learning Parameters of Mixture Model

Similar to the previous method, we use the EM algorithm to minimize the negative log likelihood function given in

Equation 14. During the E-step, the expected value for t_i is:

$$\begin{aligned} t_i^{s+1} &= \mathrm{E}(t_i|F,\theta^s)) \\ &= 1 \cdot P(t_i = 1|F,\theta^s) + 0 \cdot P(t_i = 0|F,\theta^s) \\ &= P(O|f_i,\theta^s) \end{aligned}$$

where the posterior is calculated from Equation 15.

During the M step, the model parameters are recomputed by solving the following partial derivatives:

$$\frac{\partial LL(T|F)}{\partial \theta} = 0 \tag{16}$$

After some manipulation, this leads to the following update equations for the model parameters:

$$\mu^{s+1} = \frac{\sum_{i=1}^{N} t_i^{s+1} f_i}{\sum_{i=1}^{N} t_i^{s+1}} \tag{17}$$

$$\sigma^{s+1} = \frac{\sum_{i=1}^{N} t_i^{s+1} (f_i - \mu)^2}{\sum_{i=1}^{N} t_i^{s+1}} \tag{18}$$

$$\lambda^{s+1} = \frac{\sum_{i=1}^{N} t_i^{s+1}}{\sum_{i=1}^{N} t_i^{s+1} f_i} \tag{19}$$

$$\alpha^{s+1} = \frac{\sum_{i=1}^{N} t_i^{s+1}}{N} \tag{20}$$

4 Incorporating Labeled Examples

This section presents a framework for incorporating labeled examples to improve the calibration accuracy of probability estimates. Let F^u and F^l be the corresponding sets of outlier scores assigned to the unlabeled data X^u and labeled data X^l, respectively. Furthermore, suppose $L = l_i$ denote the set of labels associated with the labeled data, where $l_i = 1$ if x_i is an outlier, and 0 otherwise.

In Sections 2 and 3, we have shown that the model parameters are calculated by minimizing the negative log likelihood. For semi-supervised learning[8], we may decompose the negative log likelihood into two parts, each corresponding to the labeled and unlabeled data:

$$\hat{\theta} = \min_{\theta} LL(F^u|\theta) + LL(F^l, L|\theta) \tag{21}$$

Although the EM algorithm is still applicable to learn the model parameters under this framework, some modifications are needed. During the E-step, we only need to estimate the values of t_i for the unlabeled data using Equation 9. The t_i values for the labeled data are fixed, i.e., $t_i = l_i(\forall x_i \in X^l)$.

During the M-step, it can be shown that the parameters are updated using a combination of the parameter estimates obtained from the unlabeled data and labeled data:

$$\mu^{s+1} = \frac{\sum_{f_i \in F_u} t_i^{s+1} f_i + \sum_{f_i \in F_l} \mathbf{1}\{l_i = 1\} f_i}{\sum_{f_i \in F_u} t_i^{s+1} + \sum_{f_i \in F_l} \mathbf{1}\{l_i = 1\}} \tag{22}$$

$$\sigma^{s+1} = \frac{\sum_{f_i \in F_u} t_i^{s+1} (f_i - \mu)^2 + \sum_{f_i \in F_l} \mathbf{1}\{l_i = 1\} (f_i - \mu)^2}{\sum_{f_i \in F_u} t_i^{s+1} + \sum_{f_i \in F_l} \mathbf{1}\{l_i = 1\}} \tag{23}$$

$$\lambda^{s+1} = \frac{\sum_{f_i \in F_u} t_i^{s+1} + \sum_{f_i \in F_l} \mathbf{1}\{l_i = 1\}}{\sum_{f_i \in F_u} t_i^{s+1} f_i + \sum_{f_i \in F_l} \mathbf{1}\{l_i = 1\} f_i} \tag{24}$$

$$\alpha^{s+1} = \frac{\sum_{f_i \in F_u} t_i^{s+1} + \sum_{f_i \in F_l} \mathbf{1}\{l_i = 1\}}{N} \tag{25}$$

where

$$\mathbf{1}\{l_i = 1\} = \begin{cases} 1 & \text{if } l_i = 1 \\ 0 & \text{otherwise} \end{cases}$$

5 Applications

This section presents two potential applications that may benefit from using probability estimates for outlier scores: threshold selection and ensemble outlier detection.

5.1 Threshold Selection

Most outlier detection algorithms require the user to specify a threshold so that any observation whose outlier score exceeds the threshold will be declared as outliers. A standard approach for determining the threshold is to plot the sorted values of outlier scores and then choose the knee point of the curve as the threshold. Such an ad-hoc method can be imprecise because the location of the knee point is subject to user interpretation.

This section illustrates a more principled approach for threshold selection using the Bayesian risk model, which minimizes the overall risk associated with some cost function. For a two-class problem, the Bayes decision rule for a given observation x is to decide w_1 if:

$$(\lambda_{21} - \lambda_{11}) P(w_1|x) > (\lambda_{12} - \lambda_{22}) P(w_2|x) \tag{26}$$

where w_1 and w_2 are the two classes while λ_{ij} is the cost of misclassifying w_j as w_i. Since $p(w_2|x) = 1 - p(w_1|x)$, the preceding inequality suggests that the appropriate outlier threshold is automatically determined once the cost functions are known. For example, in the case of a zero-one loss function, where:

$$\lambda_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases} \tag{27}$$

we declare any observation whose estimated posterior probability $P(O|f)$ exceeds 0.5 as an outlier.

5.2 Ensemble Outlier Detection

Recently, Lazarevic et al. [9] have proposed a feature bagging approach for combining outputs from multiple outlier detection algorithms. There were two approaches investigated in this study: *breadth first* and *cumulative sum*. In

both of these approaches, the outputs from each detector in the ensemble are sorted and ranked according to decreasing order of their magnitudes. For each observation, the ranks of its outlier scores in the ensemble are aggregated to obtain the final outlier score.

Instead of merging the ranks, we propose to combine their probability estimates. We employ techniques from reliability theory to perform the probability aggregation. Each outlier detector in the ensemble is considered as a component in a complex system. The components can be arranged in series, in parallel, or in any combination of series-parallel configurations. Detection of outliers by one of the detectors in the ensemble is analogous to having one of the components in the system fails. Using this framework, the overall probability that a given observation is an outlier is equivalent to the probability that the overall system fails.

Let $P(O|x)$ be the posterior probability that x is an outlier according to the ensemble and $P_i(O|x)$ be its posterior probability estimated by detector i. For the series configuration, at least one of the components must fail in order to make the entire system fails. Therefore:

$$P(O|x) = 1 - \prod_{i=1}^{R}(1 - P_i(O|x)) \tag{28}$$

For the parallel configuration, the system fails only if all the components break down. Analogously, the probability that x is an outlier is:

$$P(O|x) = \prod_{i=1}^{R} P(O|x_i) \tag{29}$$

6 Experimental Evaluation

We have conducted our experiments on several real and synthetic data sets to evaluate the performances of the calibration methods. We have also demonstrated the effectiveness of using the probability estimates in the context of threshold selection and ensemble outlier detection.

Table 2. Description of Data Sets

Data sets	Total number of instances	Number of features	Number of outliers
Letter	845	16	79
Opt	623	64	55
SVMguide1	3244	4	155
Cancer	489	10	45
Lpr	2103	183	101
Shuttle	4132	9	132

6.1 Experimental Setup

The data sets used for our experiments are summarized in Table 2. A description of each data set is given below:

Letter: This corresponds to the letter recognition data obtained from the UCI machine learning repository. We choose examples from two classes and designate one of the classes as normal and the other as outlier.

Optical: This is the optical handwritten data set from the UCI machine learning repository. Again, we randomly choose two classes and select data from one class as the normal examples and a small portion of the other class as outliers.

SVMguide1: This data set is used to test the libsvm software. The data originally comes from an astroparticle application. One of the two classes is chosen as the outlier class while the other is the normal class.

Cancer: This data set, which is obtained from UCI machine learning repository, records the measurements for breast cancer cases. There are two classes, benign, which is considered as normal and malignant, which is the outlier class.

Lpr: This is one of UNM's benchmark data sets. For each trace generated by a user, an ordered list of the frequency counts together with their class label showing "intrusive" or "normal" is recorded.

Shuttle: We use a subset of the shuttle data set from Statlog Project Database. We choose two of the large classes as normal and select examples from one of the remaining smaller classes as outliers.

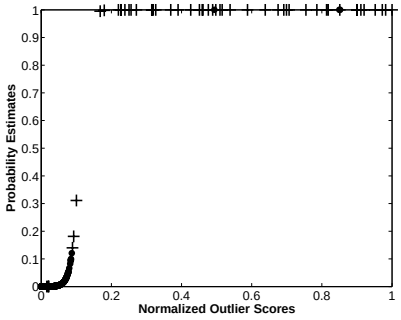
Our calibration methods utilize the outlier scores produced by a distance based outlier detection algorithm. The outlier score is computed based on the distance of each observation to its k-th nearest neighbor [6]. The value of k is set to be 3 to 5 times the number of true outliers depending on the data sets. We choose this method as our underlying algorithm because it is easy to implement and is widely used by many authors for comparison purposes.

6.2 Empirical Results

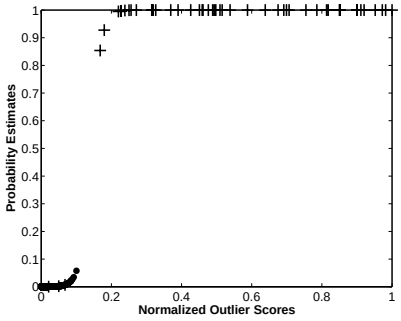
We have conducted several experiments to demonstrate that our proposed methods help to generate meaningful probability outputs as well as improving the effectiveness of threshold selection and ensemble outlier detection.

6.2.1 Probability Estimation

The posterior probabilities not only help us to determine the outlierness of an observation, they also provide estimates of confidence in outlier prediction. This experiment aims to demonstrate that the probability estimates are pushed closer towards 0.5, which indicates a low confidence in prediction,

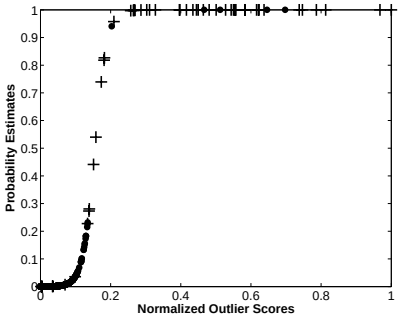


(a) Mixture model approach

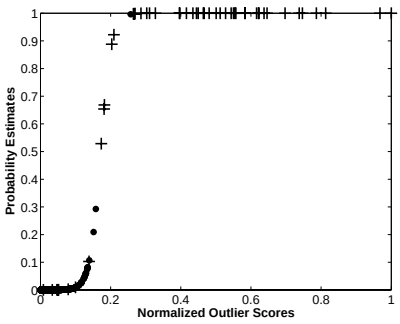


(b) Sigmoid approach

Figure 2. Plots of probability estimation for synthetic data set with variance = 40



(a) Mixture model approach



(b) Sigmoid approach

Figure 3. Plots of probability estimation for synthetic data set with variance = 150

when the outliers are difficult to be distinguished from normal observations.

For this experiment, we use two synthetic data sets generated from a mixture of two distributions. The normal observations are generated from a Gaussian distribution while the outliers are assumed to be uniformly distributed. The Gaussian distributions for both synthetic data sets have the same mean but different variance (40 versus 150). Because of its higher variance, it is harder to distinguish the outliers from normal observations in the second data set compared to the first data set.

Figures 2 and 3 show the calibrated probability estimates for the two data sets. For the first data set, because outliers are well-separated from the normal examples, most of the calibrated values are close to either 0 or 1. For the second data set, because it is harder to distinguish outliers from normal observations, there are more observations with probability estimates around 0.5. Without converting the outlier scores into probabilities, we may not be able to obtain an estimate of the confidence in outlier detection. The calibration plots for both sigmoid and mixture models look quite similar. However, note that the mixture model approach does not always yield a sigmoid-like curve. The shape of the curve depends on the parameters of the exponential and Gaussian distributions.

6.2.2 Threshold Selection

The purpose of this experiment is to compare the effectiveness of using probability estimates for threshold selection. The baseline method for threshold selection is obtained in the following way. For each data set, we first plot the outlier scores in increasing order of their magnitudes. Figure 4 shows an example of such plot for the SVMguide1 data set. The knee of the curve, which is located somewhere between 0.05 and 0.1, is then chosen as the threshold for declaring outliers. We refer to this method as "knee" in the remainder of this section.

Tables 3 to 8 show the results of applying different methods for threshold selection. "sigmoid" and "mix" corre-

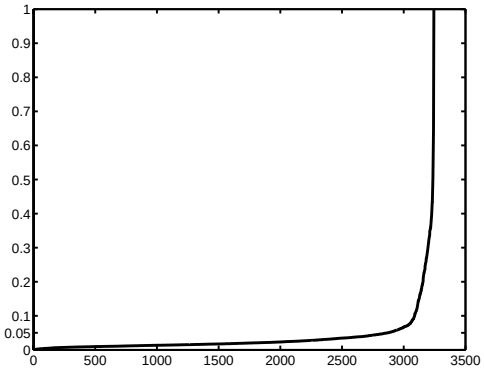


Figure 4. Sorted Outlier Scores

Table 3. Letter

	knee	mix	sigmoid	Smix	Ssigmoid
R	0.9937	0.6582	0.7759	0.7165	0.7848
P	0.4642	1.0000	0.9903	1.0000	0.9810
F	0.6328	0.7939	0.8701	0.8348	0.8720
FA	0.0007	0.0340	0.0226	0.0284	0.0217

Table 4. Opt

	knee	mix	sigmoid	Smix	Ssigmoid
R	0.4912	0.7018	0.9123	0.7193	0.9474
P	0.9655	0.9756	0.9123	0.9535	0.9000
F	0.6512	0.8163	0.9123	0.8200	0.9231
FA	0.0483	0.0289	0.0087	0.0273	0.0053

Table 5. SVMguide1

	knee	mix	sigmoid	Smix	Ssigmoid
R	0.4903	0.7419	0.7548	0.7419	0.6968
P	0.8636	0.7516	0.7500	0.7516	0.7941
F	0.6255	0.7468	0.7524	0.7468	0.7423
FA	0.0250	0.0129	0.0123	0.0129	0.0151

sponds to the sigmoid and mixture model approaches described in Section 2.1 and Section 3, respectively. "Ssigmoid" and "Smix" are the semi-supervised versions of these algorithms, which utilize some labeled examples to aid the calibration. The labeled examples count for 10% in the data set. We conduct our experiments using the 0-1 loss function, which means that the probability threshold for identifying outliers is 0.5.

The following evaluation metrics are used to compare the effectiveness of the threshold selection methods: Precision(P), Recall(R), F-measure(F) and False Alarm rate (FA). All of these metrics are computed from the confusion matrix shown in Table 9. The formula for calculating these metrics are listed in Equation 30.

Table 6. Cancer

	knee	mix	sigmoid	Smix	Ssigmoid
R	0.4667	0.9778	0.8222	0.9778	0.8222
P	0.9130	0.6667	0.8222	0.6769	0.8222
F	0.6176	0.7928	0.8222	0.8000	0.8222
FA	0.0515	0.0024	0.0180	0.0024	0.0180

Table 7. Lpr

	knee	mix	sigmoid	Smix	Ssigmoid
R	0.3564	1.0000	0.4554	1.0000	0.6139
P	1.0000	0.5372	1.0000	0.5372	1.0000
F	0.5255	0.6990	0.6259	0.6990	0.7607
FA	0.0314	0	0.0267	0	0.0191

Table 8. Shuttle

	knee	mix	sigmoid	Smix	Ssigmoid
R	0.4242	0.5152	0.6894	0.5455	0.7045
P	0.7568	0.7010	0.6894	0.7129	0.6889
F	0.5437	0.5939	0.6894	0.6180	0.6966
FA	0.0187	0.0159	0.0103	0.0149	0.0098

Table 9. Confusion Matrix

	True outlier	True normal
Predicted outlier	TP	FP
Predicted normal	FN	TN

$$\begin{aligned} P &= \frac{TP}{TP+FP} \\ R &= \frac{TP}{TP+FN} \\ F &= \frac{2TP}{2TP+FP+FN} \\ FA &= \frac{FN}{FN+TN} \end{aligned} \tag{30}$$

A good threshold must balance both precision and recall, therefore a higher F-measure value is favored. Our results clearly show that the F-measure for all the data sets improved after probability calibration. Thus, converting the outlier scores into probabilities improves threshold selection in terms of balancing its precision and recall. The results also show that the addition of labeled examples tends to produce better calibration. More specifically, both "Smix" and "Ssigmoid" approaches outperform their unsupervised counterparts in five out of six data sets in terms of their F-measure.

6.2.3 Outlier Detection Ensemble

This experiment compares the effectiveness of using probability estimates for combining outputs from multiple outlier

Table 10. AUC values for different approaches to combining outputs from multiple outlier detectors

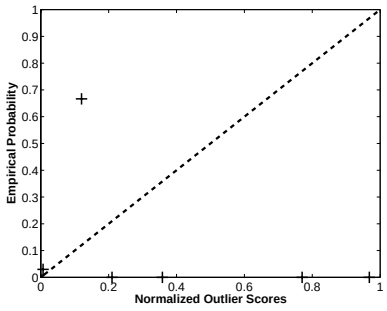
	knee	sum	breadth	mix	sigmoid	Smix	Ssigmoid
Cancer	0.9885	0.9903	0.9883	0.9941	0.9921	0.9940	0.9932
Svmguide1	0.9750	0.9798	0.9694	0.9846	0.9841	0.9846	0.9823
Letter	0.9555	0.9549	0.9412	0.9592	0.8826	0.9657	0.9580
Opt	0.9605	0.9583	0.9503	0.9695	0.9643	0.9719	0.9640
Lpr	0.9999	1.0000	0.9999	0.9965	1.0000	0.9965	1.0000
Shuttle	0.9881	0.9885	0.9885	0.9900	0.9890	0.9900	0.9908

detectors as opposed to the rank aggregation methods (denoted as breath and sum) proposed by Lazarevic and Kumar [9]. The number of ensembles used in experiments is 10. The receiver-operating characteristic (ROC) curve is often used to show the tradeoff between detection rate and false alarm rate. Alternatively, we may use the area under ROC curve (AUC) as our evaluation metric, where the better scheme will have an AUC value closer to 1. From Table 10, it can be observed that combining probability estimates from multiple outlier detectors yields higher AUC values than combining the ranks of their outlier scores. The semi-supervised approaches ("Ssigmoid" and "Smix") also tend to produce higher AUC values that unsupervised approaches in most of the data sets.

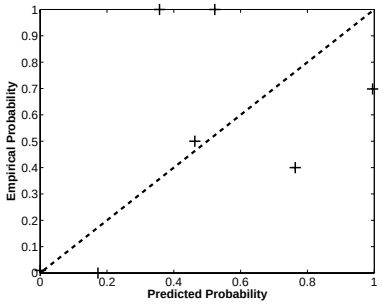
Another interesting observation can be made when comparing Tables 3 to 8 against Table 10. For threshold selection, the "sigmoid" approach tends to outperform the "mix" approach, whereas for ensemble outlier detection, the "mix" approach tends to produce higher AUC. One possible explanation is that both applications (threshold selection and ensemble outlier detection) have different requirements concerning the calibration accuracy. Threshold selection is more concerned with determining whether the posterior probability is greater than or less than 0.5 and places less emphasis on how far the estimated probabilities deviate from their true probabilities. In contrast, the difference between the estimated probability and true probability may affect the effectiveness of the ensemble outlier detection framework. Since the "mix" approach models the outlier score distributions separately for the normal and outlier classes, we expect its probability estimates to be more accurate. We plan to investigate this issue further as part of our future work.

6.2.4 Reliability Diagram

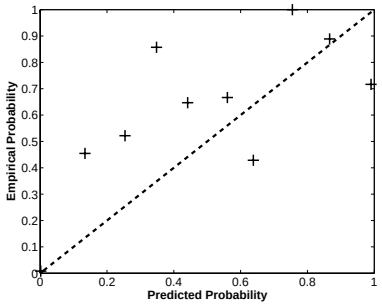
Reliability diagram [10] is a method for visualizing the calibration accuracy of a classifier. It can also be used in outlier detection to show the correspondence between outlier score values and their empirical probabilities. For the raw outlier scores, we normalize them to fall within the range of [0,1]. We then discretize the scores into equal with bins(0.1) and compute the fraction of examples in each bin that are true



(a) Before Calibration



(b) After Calibration



(c) After Semi-Supervised Calibration

Figure 5. Reliability Diagrams

outliers. The latter corresponds to the empirical probability estimate for the corresponding bin. Figure 5(a) shows the reliability diagram for the normalized raw outlier scores. If the outlier scores are in agreement with the estimated probabilities, the points should be close to diagonal (dashed) line. This plot shows that the normalized scores obtained directly from distance-based outlier detection algorithms have little to do with posterior probabilities.

After calibration, Figure 5(b) shows that the predicted probabilities are closer to the empirical ones. However the difference is still quite large since the calibration is done without using any labeled information. Finally, Figure 5(c) shows the reliability diagram for semi-supervised calibration. Note that most of the points have moved closer to the diagonal line, which means that the labeled examples help to improve the probability calibration.

7 Conclusions

In this paper, we study the problem of transforming outlier scores into probability estimates. Unlike existing calibration algorithms, our proposed methods do not require any labeled examples. In the first approach, we treat the labels as hidden variables and fit outlier scores into a sigmoid function. An efficient EM-based algorithm is developed to learn the function parameters. To obtain a more accurate calibration, we propose an alternative approach that models the score distributions using a mixture of Gaussian and exponential distributions. The posterior probability is then calculated using Bayes' rule. We show that the probability estimates from outlier scores have many potential applications. We discuss about the use of the probability estimates in selecting a more appropriate outlier threshold and in improving the performance of an outlier detection ensemble. Our experimental results suggest that our proposed methods can produce accurate calibration and can be used effectively for threshold selection and outlier detection ensemble. We further demonstrate that the calibration performance improves with the aid of some labeled examples.

References

[1] P. N. Bennett. Using asymmetric distributions to improve text classifier probability estimates. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 111–118, New York, NY, USA, 2003. ACM Press.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1995.

[3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104. ACM Press, 2000.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B*, 34:1–38, 1977.

[5] J. Gao, H. Cheng, and P.-N. Tan. A novel framework for incorporating labeled examples into anomaly detection. In *Proceedings of the Second SIAM International Conference on Data Mining*, 2006.

[6] W. Jin, A. K. H. Tung, and J. Han. Mining top-n local outliers in large databases. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 293–298. ACM Press, 2001.

[7] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *VLDB Journal:Very Large Data Bases*, 8(3-4):237–253, 2000.

[8] T. Lange, M. H. Law, A. K. Jain, and J. Buhmann. Learning with constrained and unlabelled data. In *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 730–777, 2005.

[9] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166, New York, NY, USA, 2005. ACM Press.

[10] M.H.Degroot and S.E.Fienberg. The comparison and evaluation of forecasters. *Statistician*, 32(1):12–22, 1982.

[11] J. C. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT press, 2000.

[12] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Machine Learning, Proceedings of the Eighteenth International Conference (ICML 2001)*, pages 609–616. Morgan Kaufmann, San Francisco, CA, 2001.

[13] J. Zhang and Y. Yang. Probabilistic score estimation with piecewise logistic regression. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 115, New York, NY, USA, 2004. ACM Press.