Assignment 2

Tutors: **Zhuozhuo Tu, Liu Liu**

Group members: **Chen Chen (cche5002) 480458339, Yutong Cao (ycao5602) 470347494, Yixiong Fang (yfan5798) 480133344**

Abstract

This report documents our modifications to the soft-margin support vector machine to improve its performance against class-dependent classification noise where the flip rates are $\rho_0 = 0.2$ and $\rho_1 = 0.4$.

Contents

1 Introduction

This report documents our modifications to the soft-margin support vector machine (SVM) to improve its performance against class-dependent classification noise (CCN).

The capability of learning with label noise is crucial for training machine learning models. According to the Law of Large Numbers [Härdle and Simar, 2007], the empirical risk converges to the expected risk asymptotically as the sample size approaches infinity. However, classification labels in large data set can be easily corrupted. For example, images are sometimes labelled manually by employing casual staff with minimal training. In this case, training a classification models using these images requires special treatment that accounts for noisy labels. Section 3 proposes three such treatments based on SVM to increase the accuracy with presence of CCN.

Section 3.3 proposed a learning method by deriving a new loss function using Expectation Maximisation [Bishop, 2006, p.423]. This method models the label noise using a Bernoulli distribution, and includes the label noise information into the loss function of SVM. This method is first studied by Biggio et al. [2011] for random classification noise (RCN), and we extended it to CCN.

Section 3.7 introduces a heuristic approach based on the work of Brodley et al. [1996]. This method follows the heuristic observation that the predicted probability of a sample point with wrong label is usually close to 0.5. In this case, our model is not capable of suggesting a prediction with confidence. We then exclude the potential noisy data points and train the model again with the remaining data which seems to be cleaner.

Section 3.4 implements importance reweighting proposed by Liu and Tao [2016]. This method uses a reweighting coefficient to reweight the loss function for each sample point. We fit SVMs with the given noisy data. We then implement cross-validation of SVM models to predict the probabilities of classifying each sample, and then use these probabilities to calculate the weightings.

To fairly compare these three approaches of attacking noisy data, all of the three methods implement the same base classification algorithm—SVM. We use SVM because it is proved to has a strong generalisation ability [Jin and Wang, 2012, Seeger, 2003, Cortes and Vapnik, 1995], and usually gives high classification accuracy when turned properly [Fernández-Delgado et al., 2014]. In addition, SVM is more susceptible to label noise than many other algorithms [Frénay and Verleysen, 2014], so we can better assess our modification to the original classification algorithm.

Section 4 applies the three methods proposed in Section 3 to two sets of noisy data with binary labels, both containing $10,000$ images. The first data set is a subset of the fashion-MNIST database and the second data set is from the CIFER database. For each data set, Section 4 also compares the simulation results, including accuracy and running times, from the three methods. In addition, we estimate the flip rate ρ_0, ρ_1 using the method proposed by Liu and Tao [2016]. Section 4 also compares our estimation with the true values.

2 Related work

Many recent literature discusses the topic of learning with label noise. One popular approach is to use classification algorithms that are proved to be robust to label noise. Frénay and Verleysen [2014] reported their findings that $0-1$ loss and least-squares loss are robust loss functions to uniform label noise. They also found the method of bagging is robust against label noise. Bagging detect the contaminated samples by estimating the variability of its base classifier when including and excluding them. Frénay and Verleysen [2014] also discussed filtering methods for learning with label noise. These methods remove mislabelled data before training a final model. Our second model follows this idea by excluding data points with vague predictions. Instead of removing the contaminated candidates directly, Yang et al. [2018] proposed a filter-like method that estimating the probability of mislabelling for each sample.

These approaches mentioned above do not require noise rates. Although this makes the approaches general, the methods may not be able to handle highly contaminated data set, especially when the noise rates are given. Biggio et al. [2011] applies Expectation Maximisation method to fit data with RCN. This method reformulates the loss function to include noise rate information. The label noise is not usually observed. They use the expected value as an estimation of unobserved labels contaminated by noise. Section 3.3 extends his method to solve problems with CCN. The idea behind this method is straightforward and intuitive, but simulation results in Section 4 finds it is less accurate than the method of importance reweighting proposed by Liu and Tao [2016]. Liu and Tao [2016] assigned a weight to each sample according to its probability of being contaminated. These probabilities are estimated from a pre-train model. They also provided an efficient method for estimating the noise rate. However, the need of fitting the model twice makes the method of reweighting slow.

3 Methods

We use SVM with Gaussian kernel as the base classification method for this task because of its generalisation ability.

Let vector x_i to represent the ith image in the dataset. The corresponding true classification labels y_i of images are corrupted by CCN and the we only observe the contaminated labels s_i.The clean and observed label y_i and s_i are both binary variables drawn from set $\{-1, 1\}$. The nature of CCN implies that class dependent noises $\epsilon(s_i)$ follows Bernoulli distributions with the mean parameters as either ρ_0 or ρ_1, depending on the value of label s_i. The observed label s_i is correct if and only if the noise $\epsilon(s_i) = 0$. For the ith image, it is straightforward to verify that

$$y_i = s_i(1 - 2\epsilon(s_i)). \tag{1}$$

SVM with Gaussian kernel was first published by Boser et al. [1992]. Cortes and Vapnik [1995] proposed an improvement on the original SVM with soft-margin to avoid over-fitting problem. The SVM is to minimise the Hinge loss function

$$\left[\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(w \cdot x_i - b)\right)\right] + \lambda\|w\|^2.$$

Here, w is the weighting factor, b is a constant. We classify the ith image into category 1 or -1 when $w \cdot x_i - b \geq 1$ or $w \cdot x_i - b \leq -1$, respectively. This loss function not only penalises points that misclassified, but also points that are closed to the dividing hyperplane, with a regularisation term $\lambda\|w\|^2$. Fernández-Delgado et al. [2014] suggested SVM is very likely to be one of the most powerful classifiers, by comparing 179 classification algorithms over 121 large data sets. The distances between data points and the dividing hyperplane gives an intuitive estimate of the generalisation ability of the trained SVM model [Hastie et al., 2001].

we use SVM as our classification method as of its strong generalisation ability. In this image classification assignment, we do not observe the true labels y_i. This section proposes three different approaches to modify ordinary SVM to attack label noise problem. However, we do not have access to a test data set with true labels (x_i, y_i) to verify the generalisation ability of our three different models, which are all trained by noisy data (x_i, S_i). In SVM, the gap between hyperplane and the training data set provides a natural and 'free' metric of its generalisation ability [Hastie et al., 2001], and hence use of test data set is not mandatory. Using this geometry factor as well as probably approximately correct learning framework, Jin and Wang [2012] proved that the SVM models have strong generalisation ability. Further, [Cortes and Vapnik, 1995, Seeger, 2003] also shows the strong generalisation ability from different perspectives.

Also comment on the poor robustness and why this is good for the purpose of assignment.

3.1 Preprocess

3.1.1 Photometric normalisation improves classification performance

We applied Photometric normalisation suggested by Jonsson et al. [2002] to re-scaled the image data sets to have mean of zero and standard deviation of one. This scaling visually removes the brightness difference among different images, and dramatically improves the performance of Gaussian kernel

SVM. This preprocess procedure is mandatory because Gaussian kernel is a radius based kernel, which only performs well when data are on a similar scale [Jonsson et al., 2002].

3.1.2 Principle component analysis reduces dimensionality

For the large CIFER data set, we applied principle component analysis (PCA) to decrease the dimensionality of the data set from 3072 to 100. Although the dimensionality is reduced by 97%, the remaining 3% features explains more than 85% of the variance in the data set.

No PCA is applied to the MNIST data set because the dimensionality (784) is already low comparing with the number of samples $(10,000)$ in the data set.

3.2 The original data set is balanced

Define random variables Y and S as the true and contaminated binary classification label of a random image vector X, respectively. The assignment instruction states the probabilities $\rho_0 = P(S = 1|Y = -1) = 0.2$ and $\rho_1 = P(S = -1|Y = 1) = 0.4$. The contaminated data has 40.0% labels S as one (i.e. $P(S = 1)$). Using this factor and the law of total probability

$$P(S = 1) = P(S = 1|Y = 1)P(Y = 1) + P(S = 1|Y = -1)P(Y = -1),$$

An observe label can either be 1 or -1, $P(S = 1|Y = 1) = 1 - P(S = -1|Y = 1)$,

$$P(S = 1) = [1 - P(S = -1|Y = 1)]\, P(Y = 1) + P(S = 1|Y = -1)P(Y = -1)$$

Knowing the flip rates $P(S = -1|Y = 1) = 0.4$ and $P(S = 1|Y = -1) = 0.2$

$$P(S = 1) = 0.6P(Y = 1) + 0.2\,[1 - P(Y = 1)] = 0.4,$$

which implies $P(Y = 1) = P(Y = -1) = 0.5$. Thus, the original classification problem is balanced. In addition, define the Bernoulli random variable $\epsilon(S)$ with the means $E(\mu(\epsilon)(S = -1)) = P(Y = 1|S = -1) = 0.5 \times 0.4/0.6 = 1/3$ and $E(\epsilon(S = 1)) = P(Y = -1|S = 1) = 0.5 \times 0.2/0.4 = 0.25$. This random variable ϵ describe the unobserved random label noise. Hence the expectation

$$E\epsilon(S) = P(Y = -1|S = 1)P(S = 1) + P(Y = 1|S = -1)P(S = -1) = 0.3. \qquad (2)$$

3.3 Method 1: Expectation Maximisation

This section extends the Expectation Maximisation algorithm proposed by Biggio et al. [2011] to improve ordinary SVM against labels noises, with mathematical justifications.

3.3.1 Expectation Maximisation derives loss function

The original algorithm proposed by Biggio et al. [2011] was proposed to study classification problems with label noise where the flip rate $\rho_0 = \rho_1$ and we extend it to manage the case dependent label noise where the flip rates can be different.

Recall the dual problem of an SVM is to maximise

$$f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i k(x_i, x_j) y_j c_j, \qquad (3)$$

subject to $\sum_{i=1}^{n} c_i y_i = 0$, and $0 \le c_i \le \frac{1}{2n\lambda}$ for all i. Here y_i and x_i are the labels and features of the ith image, c_i is the ith Lagrangian multiplier, $k(x_i, x_j)$ is the Gaussian kernel product of x_i and x_j whose corresponding kernel is $\exp(-\gamma\|x_i - x_j\|)$. Substituting label noise (1) into loss function (3) gives

$$f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} S_i c_i k(x_i, x_j) S_j c_j (1 - 2\epsilon(S_i))(1 - 2\epsilon(S_j)). \qquad (4)$$

This loss function involves random variables. For a similar but simpler problem with RCN, Biggio et al. [2011] applied the technique of Expectation Maximisation, which uses the expected value of the loss function as the objective function for optimisation algorithms.

Here, when $i = j$, the expectation $E(1-2\epsilon(S_i))(1-2\epsilon(S_j)) = 1-4E\epsilon(S_j)+4E\epsilon(S_j^2) = 1$. When $i \neq j$, by substituting these expectations (2) from Section 3.2, the expectation $E(1-2\epsilon(S_i))(1-2\epsilon(S_j)) = (1-2E\epsilon(S_i))(1-2E\epsilon(S_j)) = 0.16$.

Define kernel correction matrix M with the (i,j)-th entry being m_{ij}.The diagonal entries $m_{ii} = 1$, and the off diagonal entries $m_{ij} = 0.16$ when indexes $i \neq j$. Using the technique of Expectation Maximisation, the loss function (4) simplifies to

$$Ef(c_1 \dots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} S_i c_i k(x_i, x_j) S_j c_j m_{ij}, \tag{5}$$

Comparing this loss function and the ordinary loss function of SVM(3), the only difference is to replace the kernel matrix K with our new proposed matrix $Q := K \circ M$, where $\circ$ denote the Hadamard product [Hastie et al., 2001].

This method gives us an accuracy of 94.6% on the testing data.

3.4 Method 2: Importance Reweighting

This section applies theorems proved by Liu and Tao [2016] to construct a weighted loss that includes noise information ρ_0 and ρ_1.

3.5 Sigmoid function estimates conditional probability

Define D to be the probability density function of the clean data (X, Y) and D_ρ to be that of contaminated data (X, S).

This section estimates the conditional probability $P_{D_\rho}(S = y|X)$.

Liu and Tao [2016] provided three methods to estimate the conditional probability $P_{D_\rho}(S = y|X)$. The methods in Sections 5.2 and 5.3 implement kernel smoothing. The two methods are similar—the formal method requires to estimate two density functions whereas the later method estimates the ratio of these two. Our data sets are high dimensional (784 and 3072), so methods two and three suffers from the curse of dimensionality. As a result, we implement a probabilistic classification method proposed in Section 5.1 by Liu and Tao [2016].

To estimate the probability $P_{D_\rho}(S = y|X)$, we first train an ordinary SVM. We then train the parameters of an additional sigmoid function to map the SVM outputs into probabilities Platt [1999]. We apply this approach rather than traditional probabilistic models because Remark 1 given by Liu and Tao [2016] indicates the traditional probabilistic method 'does not perform well'.

3.6 Reweighting coefficient improves robustness against label noise.

ATTENDTION! I used P in previous sections. Either use P_D in all probability notations in this report or none of them. Also define f which I have no clue what it is. Note that I have already occupied the letter f. Use something else if this is not the same f. Alternative change my previous letters f. Either way, Be consistent.

To improve the robustness against label noise, Liu and Tao [2016] constructed a reweighting coefficient vector

$$\beta(X, Y, S) = \frac{P_D(Y|X)}{P_{D_\rho}(S|X)}. \tag{6}$$

either use $\mathbb{E}$ for all expectation all none of them please fix

With some reasonable assumptions (what are the assumptions), Liu and Tao [2016] also derived the distribution of the unobserved true label using the observed label S, the flip rates ρ_0 and ρ_1

$$P_D(Y = y|X) = \frac{P_{D_\rho}(S = y|X) - \rho_{1-y}}{1 - \rho_y - \rho_{1-y}} \text{ where } y \in \{0, 1\}. \tag{7}$$

Substitute this estimation into the weighting coefficient (6):

$$\beta(X, Y, S) = \frac{P_{D_\rho}(S = y|X) - \rho_{1-y}}{(1 - \rho_y - \rho_{1-y})P_{D_\rho}(S = y|X)} \text{ where } y \in \{0, 1\}. \tag{8}$$

Further substituting the conditional probability $P_{D_\rho}(S = y|X)$ estimated in Section 3.5 gives reweighting coefficient $\beta(X, Y, S)$. This reweighting coefficient $\beta(X, Y, S)$ is then used to estimate the expected risk $R_{\ell,D}(f)$ in the with the empirical risk $R_{\ell,D_\rho}(f)$

$$R_{\ell,D}(f) = R_{\beta\ell,D_\rho}(f).$$

This equation allows us to estimate the true loss function $R_{\ell,D}(f)$ without knowing the true labels Y.

3.7 Method 3: heuristic approach

Method 3 still implements SVM. We chose SVM because classification algorithms with Hinge loss, including SVM, is robust against random classification label noise. Our label noise is class dependent. However, experiment shows that SVM is still robust against it.

3.7.1 Select samples

Ordinary SVM only gives a classification without revealing a probability that indicates the confidence of classification. Wu et al. [2003] proposed a five-fold cross-validation method to calculate the classification probabilities $P(Y = 1|X)$ for SVM. Using this method, we calculated the probability $P(Y = 1|X)$ with label noise by an SVM with Gaussian kernel. We have $10,000$ samples. We only use those with largest and smallest $P(Y = 1|X)$ (first $1/3$ and last $1/3$), because intuitively the contaminated samples are more likely to have a probability $P(Y = 1|X)$ close to 0.5 and the error rate $P(\epsilon = 1) = 0.3 \approx 1/3$.

3.7.2 Label correction

We relabel the $1/3$ of the samples with highest fitted probability $P(Y = 1|X)$ as 1 and the $1/3$ samples with the smallest fitted probability $P(Y = 1|X)$ as -1. This step is important because section 3.2 shows that the original data set is balanced ($P(Y = 1) = P(Y = -1) = 0.5$).

Training our SVM with Gaussian Kernel again with this subset of relabeled data gives our second model.

This method gives us an accuracy of 95.0% on the testing data.

3.8 Flip rates estimation methods

We apply techniques proposed by Liu and Tao [2016] to estimate the flip rates ρ_0 and ρ_1. The technique consists of two steps—firstly, we estimate probability $P_{D_\rho}(S = y|X)$; secondly, we estimate the flip rates ρ_0 and ρ_1 using the probability $P_{D_\rho}(S = y|X)$. We use the density ratio method for estimating $P_{D_\rho}(S = y|X)$. ***(refer to papers given on the github page of densratio package)

****

3.9 Tuning hyperparameters

The Kernel parameter for Gaussian Kernel γ is chosen to maximise the variance of Kernel matrix $K(x_i, x_j)$ over all i, j. The regularisation parameter is chosen by grid search. The model seems to be insensitive to the regularisation parameter.

3.10 Bootstrap constructs confidence intervals and hypothesis tests

The assignment instruction asks us to compare the accuracy, denoted as A, of the proposed label noise robust algorithms.

3.10.1 Bootstrapping percentile confidence intervals

To systematically compare the metrics, we construct an 95% confidence interval for A by bootstrapping percentile confidence interval. The idea of bootstrapping is straightforward—we resample a subset of 8 samples among the sample space of 16 simulation results and calculate the mean. We

Data	Null Hypothesis (H_0)	D	P-value	Reject H_0
MNIST	Relabelling algorithm is no more accurate than Expectation Maximisation.	0.625	0.0019	Reject
	Expectation Maximisation algorithm is as accurate as Reweighting.	0.3125	0.4154	Fail to reject
	Reweighting algorithm is as accurate as relabelling.	0.3125	0.4154	Fail to reject
CIFAR	Expectation Maximisation algorithm is no more accurate than relabelling.	0.5	0.0183	Reject
	Reweighting algorithm is no more accurate than Expectation Maximisation.	0.6875	0.0005	Reject
	Reweighting algorithm is no more accurate than relabelling.	0.6875	0.0005	Reject

Table 1: Hypothesis test

Table 2: Mean sd

Mean	Standard deviation	Confidence interval
0.938	0.003	(0.936,0.939)
0.942	0.003	(0.941,0.944)
0.94	0.003	(0.939,0.942)
0.835	0.004	(0.833.0.837)
0.832	0.008	(0.829,0.836)
0.844	0.005	(0.841,0.846)

repeat this process 1000 times. The 2.5% and 97.5% percentiles of the 1000 re-sampled means are then the bootstrapping percentile confidence interval

$$(A^*_{2.5}, A^*_{97.5}), \tag{9}$$

where A^*_α is the α percentile of the bootstrapped distribution from our sample space with 16 accuracy result from Monte-Carlo simulations.

Bootstrapping does not require the sample space follows specific distributions. We apply this non-parametric method to construct confidence interval here because we do not know about the exact distributions of accuracy A.

3.10.2 Kolmogorov-Smirnov test compares the accuracy of algorithms

We implement the Kolmogorov-Smirnov test to test the hypothesis that the algorithms proposed in Section 3 have different accuracy. Again, Kolmogorov-Smirnov test is distribution free, so we do not need to know the distributions of the evaluation metrics.

Let A_{ij} denote the accuracy generated from our ith algorithm from the jth Monte-Carlo simulation. Define the empirical distribution of accuracy results generated by algorithm i as

$$\hat{F}_i(x) = \frac{1}{n} \sum_{j=1}^{16} 1_{A_{ij} \le x}. \tag{10}$$

The test statistic is the supremum among the differences of the empirical distribution generated using definition (10) [Walck, 1996]

$$D = \sup_x |F_{i_1}(x) - F_{i_2}(x)|. \tag{11}$$

We compare the test statistics D with the critical value of 0.433, which corresponds to 16 samples and a 95% of confidence level. We reject the null hypotheses that the two algorithms produce similar Accuracy A with 95% confidence level if the test statistics $D > 0.433$. The technique is also applied to perform one tail test to test whether one algorithm is more accurate than an other.

4 Experiments and discussions

please plot the cifar with and without PCA, put them into one graph and compare. codes are already available. just run them and save and insert.

Please pay special attention to the instructions in section **??** regarding figures, tables, acknowledgments, and references.

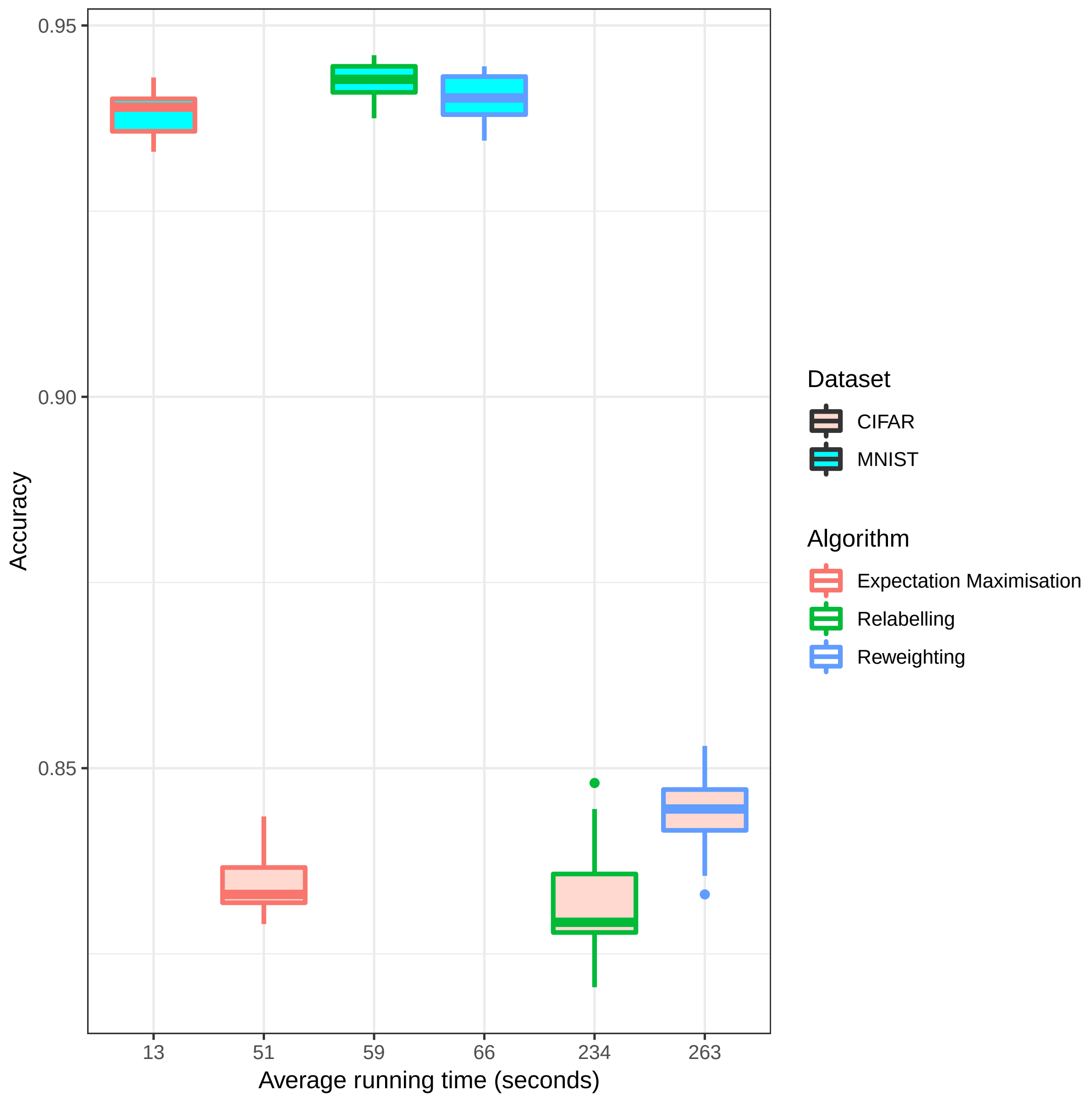

Figure 1: Boxplot

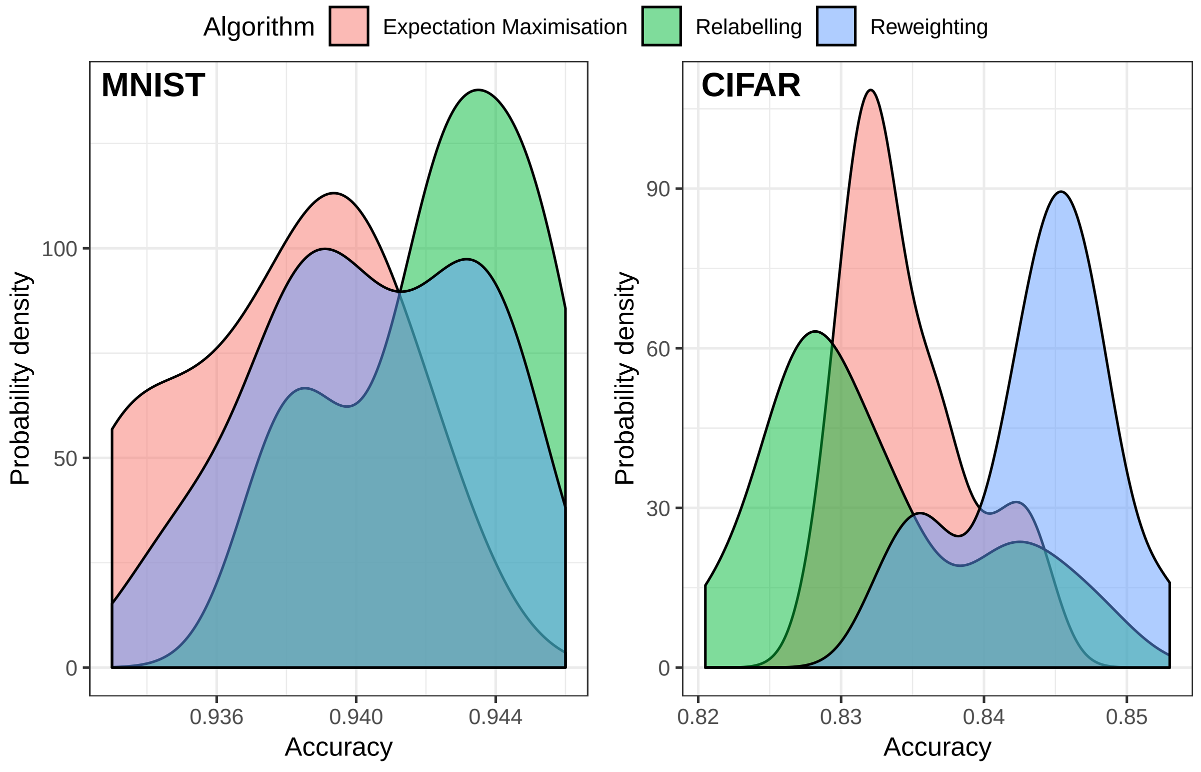

Figure 2: Density function from kernel smoothing

5 Conclusion

References

Wolfgang Härdle and Léopold Simar. *Applied multivariate statistical analysis*, volume 22007. Springer, 2007.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In *Proceedings of the Asian Conference on Machine Learning*, volume 20, pages 97–112. PMLR, 2011.

Carla E Brodley, Mark A Friedl, et al. Identifying and eliminating mislabeled training instances. In *Proceedings of the National Conference on Artificial Intelligence*, pages 799–805, 1996.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2016.

Chi Jin and Liwei Wang. Dimensionality dependent pac-bayes margin bound. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1034–1042. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4500-dimensionality-dependent-pac-bayes-margin-bound.pdf`.

Matthias Seeger. Pac-bayesian generalisation error bounds for gaussian process classification. *J. Mach. Learn. Res.*, 3:233–269, March 2003. ISSN 1532-4435. doi: 10.1162/153244303765208386. URL `https://doi.org/10.1162/153244303765208386`.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995. ISSN 1573-0565. doi: 10.1007/BF00994018.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.*, 15(1):3133–3181, January 2014. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2627435.2697065.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2014.

Pengyi Yang, John T Ormerod, Wei Liu, Chendong Ma, Albert Y Zomaya, and Jean YH Yang. Adasampling for positive-unlabeled and label noise learning with bioinformatics applications. *IEEE Transactions on Cybernetics*, (99):1–12, 2018.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X. doi: 10.1145/130385.130401. URL http://doi.acm.org/10.1145/130385.130401.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Kenneth Jonsson, Josef Kittler, YP Li, and Jiri Matas. Support vector machines for face authentication. *Image and Vision Computing*, 20(5-6):369–375, 2002.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

Tingfan Wu, ChihJen Lin, and Ruby C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2003.

Christian Walck. *Hand-book on statistical distributions for experimentalists*. 1996. URL http://www.fysik.su.se/~walck/suf9601.pdf.