Learning with Augmented Class by Exploiting Unlabeled Data *

Qing Da Yang Yu Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{daq, yuy, zhouzh}@lamda.nju.edu.cn

Abstract

In many real-world applications of learning, the environment is open and changes gradually, which requires the learning system to have the ability of detecting and adapting to the changes. Class-incremental learning (C-IL) is an important and practical problem where data from unseen augmented classes are fed, but has not been studied well in the past. In C-IL, the system should beware of predicting instances from augmented classes as a seen class, and thus faces the challenge that no such instances were observed during training stage. In this paper, we tackle the challenge by using unlabeled data, which can be cheaply collected in many real-world applications. We propose the LACU framework as well as the LACU-SVM approach to learn the concept of seen classes while incorporating the structure presented in the unlabeled data, so that the misclassification risks among the seen classes as well as between the augmented and the seen classes are minimized simultaneously. Experiments on diverse datasets show the effectiveness of the proposed approach.

Introduction

Traditional machine learning approaches face many challenges raised in real-world applications, where the open and dynamic environments break the stationary settings implied in traditional approaches. A branch of methods dealing with the changing environments is the incremental learning, which mainly includes sub-branches of the example-incremental learning (E-IL) (Ruping 2001; Polikar et al. 2001; Fern and Givan 2003), the attribute-incremental learning (A-IL) (Vapnik, Vashist, and Pavlovitch 2009), the class-incremental learning (C-IL) (Fink et al. 2006; Muhlbaier, Topalis, and Polikar 2009; Kuzborskij, Orabona, and Caputo 2013) as concluded in (Zhou and Chen 2002). Among them, C-IL is an important problem which is often encountered in practice. For example, in building an image classification system for pictures in the Internet, the user may only label a few classes, say the *dog*, *fish* and *bird*. However, the system has to predict images from wide classes in the future. When an image of *tiger* comes, a traditional classification algorithm will predict it in seen classes, like *dog*, which could make the system unusable.

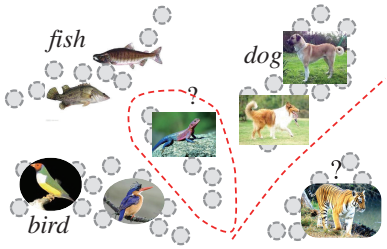


Figure 1: An illustration that unlabeled data helps the learning with augmented class problem.

This paper investigates one of the core problems in C-IL, i.e., how to recognize instances from unseen augmented classes. An augmented class is a class which is unknown during the training stage, but appears in the test stage. Once the system can tell the augmented classes from the seen ones, latter processing of the augmented classes can be handled. Therefore, we would like the system to report an extra option to denote that an instance is from the augmented class, with a high accuracy.

Specifically, the learning with augmented class (LAC) problem, is given a training dataset $D = \{(x_i, y_i)\}_{i=1}^{L}$, where $x_i \in R^d$ is an training instance and $y_i \in Y = \{1, 2, ..., K\}$ is the associated class label. Unlike the canonical classification, during test, we need to predict the class of the instances from an open dataset $D_o = \{x_i, y_i\}_{i=1}^{\infty}$, where $y_i \in Y_o = \{1, 2, ..., K, K+1, ..., M\}$ with $M > K$. As there are classes unobservable during the training time, the goal of learning with augmented class is to learn a model $f(x) : X \to Y' = \{1, 2, ..., K, novel\}$, where the option $novel$ indicates that x belongs to the augmented class, in order to minimize following expected risk

$$f^* = \text{argmin}_{f \in \mathcal{H}} \mathbb{E}_{(x,y)\sim D_o} err(y, f(x)), \qquad (1)$$

where $\mathcal{H}$ is a hypnosis space and err is LAC error

$$err(y, f(x)) = \begin{cases} I(f(x) \neq y), & y \in Y \\ I(f(x) \neq novel), & y \notin Y \end{cases} \qquad (2)$$

Here $I(\text{expression})$ is an indicator function which equals 1 when the expression is true and 0 otherwise.

*This research was supported by the 973 Program (2014CB340501), NSFC (61333014, 61375061) and JiangsuSF (BK2012303).

The main challenge of the task lies under that no instances from augmented classes are observed in the training set. Meanwhile, in many real-world applications a large amount of unlabeled data can be easily collected. Previous studies (Chapelle, Schölkopf, and Zien 2006; Zhu and Goldberg 2009; Zhou and Li 2010) have disclosed that unlabeled data implies useful information that can help improve the classification performance particularly when the number of training instances is limited. Inspired from these studies, we investigate using unlabeled data to help the LAC problem, where the number of seen classes is limited. Our intuition is that, as illustrated in Figure 1, when the unlabeled data is sufficient, large margin classifiers can be identified, which are usually separators between classes. Therefore the large margin separators surrounding the seen classes can help distinguish augmented classes.

Following this idea, we present the LACU (Learning with Augmented Class with Unlabeled data) framework and the LACU-SVM approach to the learning with augmented class problem. The proposed method combines the large margin principle from the SVM learning algorithm, with the low density separator technique from semi-supervised learning algorithms (Chapelle and Zien 2005). By adopting the one-vs-rest approach, the LACU-SVM picks a classification boundary among all low density separators that minimizes the empirical risk and structure risk as well as the *augment risk* simultaneously. An efficient method based on the concave-convex procedure (CCCP) is applied for solving the optimization problem. Experiments on datasets from several diverse domains show that the proposed approach significantly outperforms comparison methods.

The rest of this paper starts with an introduction of the related work. Then the LACU framework is presented, which is followed by the empirical studies. The paper ends with a section of discussion and conclusion.

Related Work

Incremental learning requires the necessary adaption of machine learning methods to the changes of an open and dynamic environment, and class-incremental learning (C-IL) (Zhou and Chen 2002) is a particularly branch that focuses on the emerging classes. In (Fink et al. 2006; Kuzborskij, Orabona, and Caputo 2013), the binary classifiers of each new class is added incrementally and trained by sharing the hypothesis of the existing classes. In (Muhlbaier, Topalis, and Polikar 2009), ensemble method is applied by incrementally introducing base learner trained from data containing new classes. However, these C-IL methods address how to adapt to augmented classes only when a few of their instances are given, but can not be applied to the LAC problem as no such instances are available.

The *classification with a reject option* (Chow 1970) aims at making reliable prediction by introducing the *reject* option when the classifier is not confident on the prediction. Loss functions were proposed to incorporate the rejection cost (Bartlett and Wegkamp 2008; Yuan and Wegkamp 2010), where making a wrong prediction could result in an error with a cost 1 and making a rejection always has a cost smaller than 1. It is clear that the loss with rejection cost is different with the LAC error. Though one could still use the rejection methods for learning with augmented class, as noticed in (Scheirer et al. 2013) that high confidence predictions do not necessarily lead to a small LAC error.

The *open set recognition* problem is an alternative term for the LAC problem mainly used by the pattern recognition community. It has been applied in face recognition (Li and Wechsler 2005), speaker recognition (Reynolds 2002), etc. Most of these studies are based on simple heuristics. In (Phillips, Grother, and Micheals 2011), it focuses on the operating threshold so that an instance is classified to seen classes only if the confidence is above the threshold. In (Scheirer et al. 2013), the risk over open space is considered and new decision boundaries are added to minimize the regions for the seen classes.

The *outlier detection* problem (Hodge and Austin 2004) requires to identify abnormal instances from a given data set. It is possible that outlier detection methods can be used for the LAC problem by predicting abnormal instances as novel. However, outlier detection is fundamentally different with the LAC problem since it relies on the application-specific definition of outlier, but does not minimize the LAC error.

The *class discovery* problem tries to find the examples of the rare classes which are unknown as a prior, but known to be existent in the training data (Pelleg and Moore 2005; Hospedales, Gong, and Xiang 2013). The augmented classes differ from the rare classes mainly in two aspects: First, the augmented class is not necessarily a rare class, but can become a large class. Second, it is possible to query examples of a rare class since they are already in the training data, while it is not possible for the augmented classes since their examples only appear in the test data.

The LACU Framework

The Framework

In many applications, a large amount of unlabeled data can be easily collected. Thus we may use unlabeled data to help the LAC problem. Besides the training set, we can access an unlabeled dataset $D_u = \{x_i\}_{i=L+1}^{L+U}$ sampled from D_o during the training time.

Based on the successful large margin classifiers, our assumption is that classes, even without been labeled, should be divided by large margin separators. Therefore, when discriminating one seen class to other seen classes, the unlabeled data can help us identify many large margin separators that have similar performance to the seen classes. Then to minimize the augment risk, among these separators, we select the one that is closest to the labeled region.

Denote $f(x) \in \mathcal{H}$ be the classification function, $\ell_h(f, D)$ be the empirical loss on training examples, $\ell_u(f, D_u)$ be the loss of in-margin quantity on unlabeled data and $\ell_a(f, D)$ be the augment loss which supports a decision boundary closer to the labeled examples. In the LACU framework, we search for a classifier for the LAC problem by minimizing following objective function

$$\min_{f \in \mathcal{H}} \|f\|_{\mathcal{H}}^2 + C_1 \ell_h(f, D) + C_2 \ell_u(f, D_u) + C_3 \ell_a(f, D), \quad (3)$$

where the first term $\|f\|^2_{\mathcal{H}}$ measures the complexity of the classification model, and C_1, C_2, C_3 is the coefficients to balance these losses.

LACU-SVM

Following the one-vs-rest strategy to multi-class classification, we extends the support vector machine (Vapnik 2000) to the optimization objective in Eq. (3), to obtain the LACU-SVM approach. In LACU-SVM, the optimization in Eq. (3) is applied K times, each treating a seen class as positive ($y_i = +1$) and the rest seen classes as negative ones ($y_i = -1$) in turn.

Denote $f(x) = \boldsymbol{w}^T\phi(x) + b$ be the linear classification function, $\ell_h(f, D)$ be the hinge loss on labeled data

$$\ell_h(f, D) = \sum\nolimits_{i=1}^{L} \max(0, 1 - y_i f(x_i)).$$

To search for the large margin separator in the unlabeled data, we use the definition of $\ell_u(f, D_u)$ which can be evaluated by summing up the in-margin quantity as in (Chapelle and Zien 2005)

$$\ell_u(f, D_u) = \sum\nolimits_{i=L+1}^{L+U} \max(0, 1 - |f(x_i)|).$$

We define $\ell_a(f, D)$ to control the move of the separator by adjusting the minimum margin values to minimize the augment loss as

$$\ell_a(f, D) = \min_{i\in I^+} y_i f(x_i) - \min_{i\in I^-} y_i f(x_i),$$

where I^+ and I^- are the indices of positive examples and negative examples in the training data, respectively.

After K classifiers are trained from Eq.(3) with the above losses, each for an seen class, we use the following prediction rule for a test instance x

$$\hat{y} = \operatorname{argmax}_{k=1,..,K,novel} f_k(x),$$

where $f_{novel} \equiv 0$, ties at 0 are predict as $novel$, and other ties are broken randomly.

However, the objective function of LACU-SVM in Eq.(3) is complicated, we thus consider the alternative objective function $\|f\|^2_{\mathcal{H}} + C_1\ell_h(f, D) + C_2\ell_u(f, D_u)$ with an extra constraint defined as

$$\min_{i\in I^+} y_i f(x_i) - \min_{i\in I^-} y_i \le -\lambda.$$

Here $\lambda > 0$ is a parameter controlling the degree of how the decision boundary is close to the positive examples. This constraint is equivalent to a series of constraints by eliminating one of the $\min$ function as

$$\min_{i\in I^+} y_i f(x_i) + \lambda \le y_j f(x_j), \forall j \in I^-. \qquad (4)$$

Even though, the objective function is still overly complex because of the symmetric hinge loss used in $\ell_u(f, D_u)$. From the derivation in (Collobert et al. 2006), the symmetric hinge loss can be approximated by

$$\max(0, 1 - |z|) \approx R_s(z) + R_s(-z) + constant.$$

Here $R_s(z) = \min(1 - s, \max(0, 1 - z))$ is the ramp loss with a hyper-parameter $s \in (-1, 0]$. It can be further rewritten as the difference between two hinge losses, i.e., $R_s(z) = H_1(z) - H_s(z)$, where $H_s(z) = \max(0, s - z)$. We then turn to solve the following minimization problem

$$\min_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta}) = J_1(\boldsymbol{\theta}) + J_2(\boldsymbol{\theta}) \qquad (5)$$

$$s.t. \quad \min_{i\in I^+} y_i f(x_i) + \lambda \le y_j f(x_j), \forall j \in I^-$$

$$\frac{\eta}{L}\sum\nolimits_{i=1}^{L} y_i \le \frac{1}{U}\sum\nolimits_{i=L+1}^{L+U} f(x_i) \le \frac{1}{L}\sum\nolimits_{i=1}^{L} y_i \qquad (6)$$

where $\boldsymbol{\theta} = (\boldsymbol{w}, b)$ is the model parameter, $J_1(\boldsymbol{\theta}) = \|f\|^2_{\mathcal{H}} + C_1\ell_h(f, D) + C_2 \sum_{i=L+1}^{L+2U} H_1(y_i f(x_i))$ is a convex function, and $J_2(\boldsymbol{\theta}) = -C_2 \sum_{i=L+1}^{L+2U} H_s(y_i f(x_i))$ is a concave function. Here $y_i = +1$ for $L + 1 \le i \le L + U$ and $y_i = -1$ for $L + U + 1 \le i \le L + 2U$, $x_{L+U+i} = x_{L+i}$ for $1 \le i \le U$. The constraint of Eq.(6) is introduced to avoid classifying all unlabeled data to one class with a very large margin, by limiting the fraction of positive data in unlabeled data within a certain range defined by a parameter $\eta > 0$, since the positive instances always take less fraction in the unlabeled data than in the labeled data following the one-vs-rest strategy.

After the decomposition of the objective function, the concave-convex procedure (CCCP) (Yuille and Rangarajan 2003) then can be applied to the optimization problem as in (Collobert et al. 2006). CCCP is an algorithm which solves a difference of convex functions programming as a sequence of convex programming, and is proved to converge with both convex and non-convex constraints (Lanckriet and Sriperumbudur 2009). In each iteration of CCCP to solve Eq.(5), we needs to solve following subproblem

$$\boldsymbol{\theta}_{t+1} = \operatorname{argmin}_{\boldsymbol{\theta}} \big(J_1(\boldsymbol{\theta}) + J_2'(\boldsymbol{\theta}_t) \cdot \boldsymbol{\theta}\big), \qquad (7)$$

Algorithm 1 LACU-SVM Training Algorithm

Input:

D: Training examples $\{x_i, y_i\}_{i=1}^{L}$
D_u: Unlabeled data set $\{x_i\}_{i=L+1}^{L+U}$
C_1, C_2, λ, η: Parameters of algorithm

Output:

$\{\boldsymbol{\theta}^k\}_{k=1}^{K}$: Parameters of learned model

1: **for** each $k \in \{1, ..., K\}$ **do**
2: $t = 0$
3: Let $\boldsymbol{\theta}_t^k$ be the solution of a standard SVM trained on $D_k = \{(x_i, y_i^k = 2I(y_i = k) - 1) | (x_i, y_i) \in D\}$
4: Calculate the kernel matrix $\boldsymbol{G}$
5: Calculate linear coefficient $\boldsymbol{\zeta}$ by Eq.(9)
6: Calculate β_i and V as

$$\beta_i = \begin{cases} C_2, & y_i^k f(x_i|\boldsymbol{\theta}_t^k) < s \text{ and } 1 \le i \le L + 2U \\ 0, & \text{otherwise} \end{cases}$$

$$V = \min_{i\in I^+} y_i^k f(x_i|\boldsymbol{\theta}_t^k)$$

7: **repeat**
8: Solve Eq.(8) to obtain $\boldsymbol{\theta}_{t+1}^k$
9: $t = t + 1$
10: Update β_i and V as in step 6
11: **until** $\boldsymbol{\theta}_t^k = \boldsymbol{\theta}_{t-1}^k$
12: **end for**
13: **return** $\{\boldsymbol{\theta}^k\}_{k=1}^{K}$

where the objective function of the subproblem becomes a summation of a convex term and a linear term. Note that there is still a non-convex constraint which involves a $\min$ function as in Eq.(4). To deal with this issue, we combine the alternative optimization technique into the CCCP framework, i.e., at time $t+1$, we use a fixed value $V = \min_{i \in I^+} y_i f(x_i|\boldsymbol{\theta}_t)$ instead of $\min_{i \in I^+} y_i f(x_i|\boldsymbol{\theta}_{t+1})$. Then the above optimization of Eq.(7) can be reformulated as dual form of a quadratic programming problem using standard SVM techniques as

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\zeta}^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{G} \boldsymbol{\alpha} \qquad (8) \\ s.t. \quad & 0 \le y_i \alpha_i \le C_1, \quad 1 \le i \le L \\ & -\beta_i \le y_i \alpha_i \le C_2 - \beta_i, \quad L+1 \le i \le L+2U \\ & \alpha_{L+2U+1} \le 0, \quad \alpha_{L+2U+2} \ge 0 \\ & \alpha_i \le 0, \quad L+2U+3 \le i \le L+2U+2+|I^-| \\ & \boldsymbol{\alpha}^\top \mathbf{1} = 0 \end{aligned}$$

where $N = L + 2U + 2 + |I^-|$ is the total number of dual variables. β_i is C_2 for $y_i f(x_i|\boldsymbol{\theta}_t) < s$ $(L+1 \le i \le L+2U)$, 0 otherwise. $\boldsymbol{G} \in R^{N \times N}$ is the kernel matrix where $G_{ij} = \langle \phi(x_i), \phi(x_i) \rangle$. Besides the already defined $L+2U$ instances, here we introduce another $(2+|I^-|)$ instances, where $\phi(x_{L+2U+1}) = \phi(x_{L+2U+2}) = \frac{1}{U} \sum_{i=L+1}^{L+U} \phi(x_i)$ and $x_i = x_{[I^-]_{i-L-2U-2}}$ for $L+2U+3 \le i \le N$. The linear coefficient of dual variable is

$$\boldsymbol{\zeta}_i = \begin{cases} y_i, & 1 \le i \le L+2U \\ \frac{1}{L} \sum_{i=1}^{L} y_i, & i = L+2U+1 \\ \frac{\eta}{L} \sum_{i=1}^{L} y_i, & i = L+2U+2 \\ -V - \lambda, & L+2U+3 \le i \le L+2U+2+|I^-| \end{cases} \qquad (9)$$

This is a quadratic programming (QP) problem very close to the standard SVM dual problem and can be efficiently solved using the existing Sequential Minimal Optimization (SMO) solver (Bottou and Lin 2007). The final optimal solution is given by

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \phi(x_i),$$

and b can be obtained by the KKT condition, i.e., $y_i(\boldsymbol{w}^T x + b) = 1$, for $0 < y_i \alpha_i < C_1 (1 \le i \le L)$ or $-\beta_i < y_i \alpha_i < C_2 - \beta_i (L+1 \le i \le L+2U)$.

The overall description of the proposed method is presented in Algorithm 1. The method takes labeled training dataset D, unlabeled dataset D_u and the four parameters C_1, C_2, λ, η as input. It trains K binary classifiers for each seen class following the one-vs-rest strategy in the outer iteration. For each seen class $k \in \{1, ..., K\}$, LACU-SVM initializes the $\boldsymbol{\theta}_0^k$ as the solution to the standard SVM trained on labeled data D_k only as in line 3, and then calculates the coefficients of Eq.(8), i.e., $\boldsymbol{G}$, $\boldsymbol{\zeta}$, $\boldsymbol{\beta}$ and V as in line 4, 5 and 6. Then it starts the iteration of CCCP by solving the quadratic programming problem in Eq.(8) from line 7 to 11. In each inner iteration, it only requires solving a QP problem by SMO in line 8, and in our empirical study we find that it usually converges only after a few iterations, making the training procedure for each class roughly enjoy the computational complexity of SMO.

Experiments

Comparison methods

To validate the the effectiveness of LACU-SVM, we conduct experiments on benchmark datasets from several diverse domains, compared with state-of-the-art methods including:

LOF: Local Outlier Factor (Breunig et al. 2000) is a powerful outlier detector, where the degree of being an outlier depends on how isolated the object is with respect to the surrounding neighborhood so that local outliers can also be detected. We use LOF for detecting augmented class, and use the predictions of the one-vs-rest SVM for the other none-outliers. This strategy is also employed for the following outlier detectors.

iForest: iForest (Liu, Ting, and Zhou 2008) is a state-of-the-art outlier detector which takes advantage of two outliers quantitative properties, i.e., few and different, by exploring the concept of isolation of samples.

OC-SVM: One-class SVM (Schölkopf et al. 2001) is another state-of-the-art outlier detector (Ma and Perkins 2003), which computes a binary function that is supposed to capture regions in input space where the probability density lives.

MOC-SVM: Since OC-SVM can hardly find local outliers since it essentially seeks for a hyperplane to separate the data and the origin. Thus, for this comparison method, we train multiple one-class svms for outlier detection, i.e., one OC-SVM for each seen class.

1-vs-Set: 1-vs-Set Machine (Scheirer et al. 2013) considers the risk over open space by introducing extra decision boundaries to minimize the regions for the seen classes.

OVR-SVM: One-vs-rest SVM is a powerful scheme for multi-class classification (Rifkin and Klautau 2004). In the original OVR-SVM, a test instance x is predicted as class y if $y = \text{argmax}_{k=1,2,..,K} f_k(x)$ where f_k is the binary SVM for class k. To adapt OVR-SVM for predicting the augmented class, we let the model return the class y only when $\max_k f_k(x) > 0$, otherwise return the augmented class.

In the experiments, we use the implementations of OVR-SVM, OC-SVM and MOC-SVM in the LIBSVM software (Chang and Lin 2011), and the implementations of 1-vs-Set Machine and iForest from the code released by the corresponding authors. The coefficient C in SVM is selected via cross validation on training data using the original OVR-SVM. The width for Gaussian kernel γ is set to a fixed value of $1/d$. For LOF, the minimum and maximum number of neighbors are 3 and 9, respectively. Note that the original LOF does not have a kernel version, so we replace the Euclid distance d_e with $1 - \exp(-\gamma d_e)$ for Gaussian kernel. Since such adaption can not be applied for iForest, we use the same version of iForest with the default parameters in its R-pacakge for both kernels. For LACU-SVM, s in the ramp loss is set to -0.3, C_1 is set to C, C_2 is set to $C_1 L/U$, and number of iterations is set to 10 for all experiments. Without further explanation, the parameters η and λ in LACU-SVM are set to 1.3 and 0.1 respectively by default. For evaluation, we focus on the macro-averaged F1 by treating the augmented class as the $(K+1)$-th class, to eliminate the influence of unbalanced data.

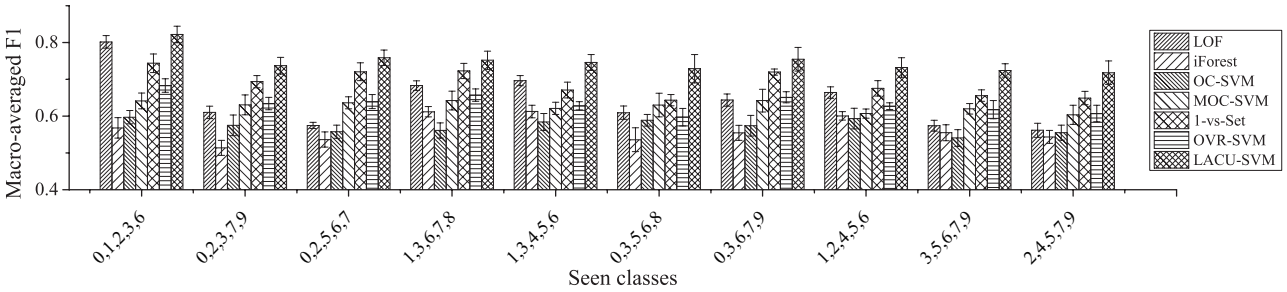


Figure 2: Comparisons of different methods on MNIST dataset (linear kernel)

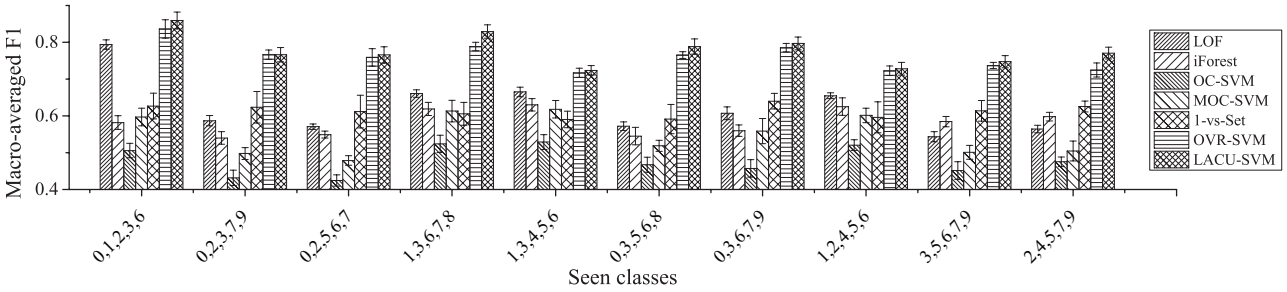


Figure 3: Comparisons of different methods on MNIST dataset (Gaussian kernel)

Results

Handwritten Digit Image Classification In the first experiment, we conducted experiments on the MNIST handwritten digit dataset, where 5 classes are randomly selected as seen classes from the all the 10 classes. The number of training data, unlabeled data and test data are 500, 500 and 1000. For each configuration (with different seen classes), the experiments are repeated for 10 times and both the mean and the standard variance of the performance are reported. The results of 10 randomly selected configurations are shown in Figure 2/3 and for linear/Gaussian kernel. For linear kernel, the 1-vs-set machine shows to outperform OVR-SVM in all cases and the LOF methods outperforms OVR-SVM in near 50% cases, while LACU-SVM always obtains the highest performance and is significantly better than the compared methods. For Gaussian kernel, LACU-SVM also achieves the best performance, while the OVR-SVM is slightly worse and significantly better than the other five methods. In both kernels, the outlier detection methods with iForest, OC-SVM and MOC-SVM produce the worst performance.

To further verify the effectiveness of LACU-SVM, we show some demonstrations of test images for the first configuration in Figure 4. The images are from unseen classes. LACU-SVM successfully identifies the novelty, while OVR-SVM predicts them to be seen classes. It is not surprising that OVR-SVM makes the wrong decisions since those samples are very similar to the seen classes and OVR-SVM never observes them during training. At the other hand, by exploiting the unlabeled data, the proposed LACU-SVM successfully recognizes these images as augmented classes.

Document Classification In the second experiment, we conduct experiments on the popular text dataset, i.e., 20 Newsgroups, to show some statistical results of all methods over various configurations. This dataset consists of documents from 20 different topics. Some of these topics are highly related like *comp.sys.ibm.pc.hardware* and

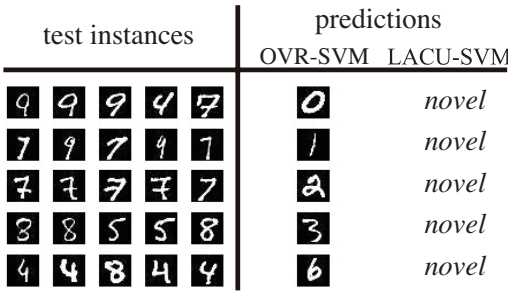


Figure 4: Examples of predictions made by OVR-SVM and LACU-SVM. The classes of 0,1,2,3,6 are observed in training set, and all classes are in the test set.

comp.sys.mac.hardware, which will be an issue, when one belongs to seen classes and the other one does not. We also limit the number of seen classes to 5. The number of training data, unlabeled data and test data are 500, 1000 and 1000 respectively. We randomly sample 100 configurations from all possible combinations, and for each configuration the experiment is repeated for 10 times. The results of *win/tie/loss* counts of all possible pairs of methods are shown in Table 1, for both linear and Gaussian kernel. The number in the ith row and the jth column denotes the times of win, tie and loss of the method in the corresponding row, versus that in the corresponding column, over all the 100 configurations. The detailed results of the macro-averaged F1 performances will be presented in the longer version of the paper.

It shows that, the proposed LACU-SVM wins all the configurations compared with the four outlier detection methods, and significantly outperforms the OVR-SVM and 1-vs-Set machine with the loss rate at most 7% for both linear and Gaussian kernels. Besides our method, OVR-SVM wins most comparisons, even has beaten 1-vs-Set machine in 74% case with linear kernel and 100% cases with Gaussian kernel, which implies that it maybe inappropriate to simply in-

Table 1: The results of *win/tie/loss* information with 100 randomly selected configurations on 20 Newsgroup dataset (paired two-tailed t-test at 95% significance level)

Method	Linear kernel							Gaussian kernel						
	LOF	iForest	OC-SVM	MOC-SVM	1-vs-Set	OVR-SVM	LACU-SVM	LOF	iForest	OC-SVM	MOC-SVM	1-vs-Set	OVR-SVM	LACU-SVM
LOF	-	0/100/0	0/0/100	0/7/93	0/0/100	0/0/100	0/0/100	-	0/100/0	0/31/69	0/0/100	0/0/100	0/0/100	0/0/100
iForest	0/100/0	-	0/0/100	0/7/93	0/0/100	0/0/100	0/0/100	0/100/0	-	0/31/69	0/0/100	0/0/100	0/0/100	0/0/100
OC-SVM	100/0/0	100/0/0	-	72/28/0	0/0/100	0/0/100	0/0/100	69/31/0	69/31/0	-	0/42/58	0/0/100	0/0/100	0/0/100
MOC-SVM	93/7/0	93/7/0	0/28/72	-	0/0/100	0/0/100	0/0/100	100/0/0	100/0/0	58/42/0	-	0/1/99	0/0/100	0/0/100
1-vs-Set	100/0/0	100/0/0	100/0/0	100/0/0	-	0/26/74	1/15/84	100/0/0	100/0/0	100/0/0	99/1/0	-	0/0/100	0/4/96
OVR-SVM	100/0/0	100/0/0	100/0/0	100/0/0	74/26/0	-	7/15/78	100/0/0	100/0/0	100/0/0	100/0/0	100/0/0	-	7/18/75
LACU-SVM	100/0/0	100/0/0	100/0/0	100/0/0	84/15/1	78/15/7	-	100/0/0	100/0/0	100/0/0	100/0/0	96/4/0	75/18/7	-

troducing an extra parallel decision boundary to limit the positive region as in 1-vs-Set when the labeled examples of each seen class may not be distributed in a very concentrative region. The outlier detection methods nearly lose all the comparisons, especially for LOF and iForest which predict almost all test instances as none-outliers. It is possibly because outliers are usually decided by specific tasks, and outlier detection does not minimize the LAC error.

Object Recognition In the last experiment, we conduct the experiment on the object recognition task, i.e., the Caltech101 dataset (Fei-Fei, Fergus, and Perona 2007) (101 classes), to show the performance of different methods when LAC problem becomes more challenging (with more unseen classes). We limit the number of seen classes to 5 and let the number of test classes (M) vary form 30 to 100 . The number of training images for each seen classes is set to 20 as a popular setting. The number of unlabeled images and test images for each test classes are set to 5 and 10 respectively. We extract the spatial histograms of visual words with 3600 dimensions for each image. We only report the results with linear kernel since it is considered to be more appropriate for such representation. For different number of test classes, we randomly sample 200 configurations and for each configuration the experiment is repeated for 5 times. The overall mean of macro-average F1 is reported in Figure 5.

It shows that, with the increasing number of test classes, the performance of most methods degrade as expected. Among all the methods, LACU-SVM achieves the best performance. The 1-vs-set machine does not achieve a comparable performance. All the outlier detection methods fail in this data, which further indicates the instances from unseen classes should not be simply regarded as outliers when dealing with LAC problem.

We further investigate the influence of different parameters in LACU-SVM, i.e., η and λ, in Figure 6(a) and Figure 6(b) respectively. We let η vary from 1.1 to 1.5 with an interval of 0.5 and let λ vary from 0.05 to 0.45 with an interval of 0.05. The performance tendencies with different number of test classes are also studied, with $M = 40, 60, 80, 100$, for an unbiased analysis of the influence of parameters. The results show that, the performance of LACU-SVM is overall not very sensitive to the parameters, and only with a small M, the performance trends to increase with a larger η as well as a smaller λ. For $M > 40$, there is no significant difference between the performance produced by various parameters.

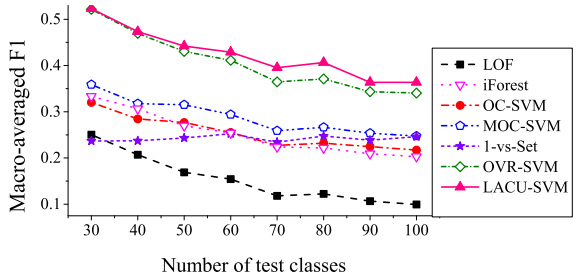
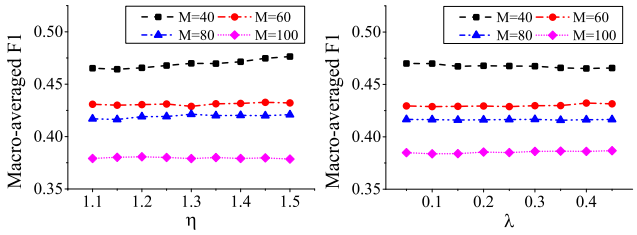


Figure 5: Performance with different number of test classes on Caltech101 dataset



(a) With different η (b) With different λ

Figure 6: The influence of parameters η and λ in LACU-SVM on Caltech101 dataset

Conclusion

Learning with augmented class is a very practical problem when a system needs to predict the data from an unlimited source. In this paper, we propose to tackle the problem with the help of unlabeled data. The experiments on several datasets show the effectiveness of the proposed method. In the future work, there are several important issues to be considered. One is that, besides the current SVM based method, we would like to apply more state-of-the-art multi-class algorithms to the LACU framework. Developing a theoretical grounded method for the LAC problem is also of interest.

Acknowledgements: We want to thank anonymous reviewers for comments and suggestions, and Yu-Feng Li, De-Chuan Zhan for discussions and reading the draft.

References

Bartlett, P. L., and Wegkamp, M. H. 2008. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research* 9:1823–1840.

Bottou, L., and Lin, C.-J. 2007. Support vector machine solvers. In Bottou, L.; Chapelle, O.; DeCoste, D.; and Weston, J., eds., *Large Scale Kernel Machines*. Cambridge, MA: MIT Press. 301–320.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: Identifying density-based local outliers. *ACM SIGMOD Record* 29(2):93–104.

Chang, C., and Lin, C. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems and Technology* 2(3):27.

Chapelle, O., and Zien, A. 2005. Semi-supervised classification by low density separation. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 57–64.

Chapelle, O.; Schölkopf, B.; and Zien, A. 2006. *Semi-Supervised Learning*. Cambridge, MA: MIT press.

Chow, C. 1970. On optimum recognition error and reject tradeoff. *IEEE Trans. Information Theory* 16(1):41–46.

Collobert, R.; Sinz, F.; Weston, J.; and Bottou, L. 2006. Large scale transductive SVMs. *Journal of Machine Learning Research* 7:1687–1712.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding* 106(1):59–70.

Fern, A., and Givan, R. 2003. Online ensemble learning: An empirical study. *Machine Learning* 53(1-2):71–109.

Fink, M.; Shalev-Shwartz, S.; Singer, Y.; and Ullman, S. 2006. Online multiclass learning by interclass hypothesis sharing. In *Proceedings of the 23rd International Conference on Machine Learning*, 313–320.

Hodge, V. J., and Austin, J. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review* 22(2):85–126.

Hospedales, T. M.; Gong, S.; and Xiang, T. 2013. Finding rare classes: Active learning with generative and discriminative models. *IEEE Trans. Knowledge and Data Engineering* 25(2):374–386.

Kuzborskij, I.; Orabona, F.; and Caputo, B. 2013. From n to n+1: Multiclass transfer incremental learning. In *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition*, 3358–3365.

Lanckriet, G. R., and Sriperumbudur, B. K. 2009. On the convergence of the concave-convex procedure. In *Advances in Neural Information Processing Systems 22*. 1759–1767.

Li, F., and Wechsler, H. 2005. Open set face recognition using transduction. *IEEE Trans. Pattern Analysis and Machine Intelligence* 27(11):1686–1697.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining*, 413–422.

Ma, J., and Perkins, S. 2003. Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks*, 1741–1745.

Muhlbaier, M.; Topalis, A.; and Polikar, R. 2009. Learn++ nc: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes. *IEEE Trans. Neural Networks* 20(1):152–168.

Pelleg, D., and Moore, A. W. 2005. Active learning for anomaly and rare-category detection. In *Advances in Neural Information Processing Systems 17*. 1073–1080.

Phillips, P. J.; Grother, P.; and Micheals, R. 2011. Evaluation methods in face recognition. In Li, S. Z., and Jain, A. K., eds., *Handbook of Face Recognition*. New York: Springer. 329–348.

Polikar, R.; Upda, L.; Upda, S.; and Honavar, V. 2001. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. Systems, Man, and Cybernetics, Part C* 31(4):497–508.

Reynolds, D. A. 2002. An overview of automatic speaker recognition technology. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 4072–4075.

Rifkin, R., and Klautau, A. 2004. In defense of one-vs-all classification. *Journal of Machine Learning Research* 5:101–141.

Ruping, S. 2001. Incremental learning with support vector machines. In *Proceedings of the 1st IEEE International Conference on Data Mining*, 641–642.

Scheirer, W.; de Rezende Rocha, A.; Sapkota, A.; and Boult, T. 2013. Toward open set recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 35(7):1757–1772.

Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; and Williamson, R. C. 2001. Estimating the support of a high-dimensional distribution. *Neural Computation* 13(7):1443–1471.

Vapnik, V.; Vashist, A.; and Pavlovitch, N. 2009. Learning using hidden information (learning with teacher). In *Proceedings of International Joint Conference on Neural Networks*, 3188–3195.

Vapnik, V. 2000. *The Nature of Statistical Learning Theory*. New York: Springer Verlag.

Yuan, M., and Wegkamp, M. 2010. Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research* 11:111–130.

Yuille, A. L., and Rangarajan, A. 2003. The concave-convex procedure. *Neural Computation* 15(4):915–936.

Zhou, Z.-H., and Chen, Z.-Q. 2002. Hybrid decision tree. *Knowledge-Based Systems* 15(8):515–528.

Zhou, Z.-H., and Li, M. 2010. Semi-supervised learning by disagreement. *Knowledge and Information Systems* 24(3):415–439.

Zhu, X., and Goldberg, A. B. 2009. *Introduction to Semi-Supervised Learning*. San Rafael, Argentina: Morgan & Claypool.