Learning Latest Classifiers without Additional Labeled Data

Atsutoshi Kumagai
NTT Secure Platform Laboratories
kumagai.atsutoshi@lab.ntt.co.jp

Tomoharu Iwata
NTT Communication Science Laboratories
iwata.tomoharu@lab.ntt.co.jp

Abstract

In various applications such as spam mail classification, the performance of classifiers deteriorates over time. Although retraining classifiers using labeled data helps to maintain the performance, continuously preparing labeled data is quite expensive. In this paper, we propose a method to learn classifiers by using newly obtained unlabeled data, which are easy to prepare, as well as labeled data collected beforehand. A major reason for the performance deterioration is the emergence of new features that do not appear in the training phase. Another major reason is the change of the distribution between the training and test phases. The proposed method learns the latest classifiers that overcome both problems. With the proposed method, the conditional distribution of new features given existing features is learned using the unlabeled data. In addition, the proposed method estimates the density ratio between training and test distributions by using the labeled and unlabeled data. We approximate the classification error of a classifier, which exploits new features as well as existing features, at the test phase by incorporating both the conditional distribution of new features and the density ratio, simultaneously. By minimizing the approximated error while integrating out new feature values, we obtain a classifier that exploits new features and fits on the test phase. The effectiveness of the proposed method is demonstrated with experiments using synthetic and real-world data sets.

1 Introduction

The performance of classifiers deteriorates over time in various applications. For example, classifiers for identifying malicious web sites would become inaccurate since malicious web sites are uninterruptedly created to scam users [Ma *et al.*, 2009]. In activity recognition using sensor data, the classification error would increase over time since user activity patterns dynamically change [Abdallah *et al.*, 2012]. For tasks where the performance of classifiers deteriorates, retraining classifiers is required to maintain the performance. There have been many methods proposed that retrain classifiers, such as online learning [Crammer *et al.*, 2009], forgetting algorithms [Klinkenberg, 2004], and ensemble learning [Wang *et al.*, 2003]. These methods require labeled data to retrain classifiers. However, it is quite expensive to continuously prepare labeled data since labels need to be manually assigned by domain experts. In contrasts, unlabeled data can be easily collected. In this paper, we propose a method for learning classifiers by using newly obtained unlabeled data as well as labeled data collected beforehand.

There are two major reasons for the performance deterioration. The first is the emergence of new features that do not appear in the training phase. For example, in spam mail classification, spammers endlessly create new spam mails that include words (features) related to new products or services to cheat users. Therefore, new features to discriminate spam mails emerge over time. If we do not retrain the classifier, it would become impossible to classify these mails precisely [Fdez-Riverola *et al.*, 2007]. Figure 1 shows the time variation of the cumulative number of features in the real-world spam data sets (ECUE spam in [Gama *et al.*, 2014]). The number of features increases rapidly over time.

The other reason is the change of probability distribution by which data are governed. For example, in a brain computer-interface, the probability distribution of EEG data changes over time since EEG patterns are affected by user attention or fatigue [Li *et al.*, 2010]. If training and test data follow different distributions, the classification performance would become poor since the classifier is learned in order to accurately classify samples generated from the training data distribution rather than samples from the test data distribution [Gama *et al.*, 2014; Pan and Yang, 2010]. In our situation, the training data correspond to the labeled data that are collected until a certain time point, and the test data correspond to the unlabeled data that are collected after the time point, which we would like to accurately classify. Figure 2 shows distributions of a real-world spam data set for February of 2003 and October of 2003, where samples are visualized in a two-dimensional space by t-distributed stochastic neighbor embedding (t-SNE) [Van der Maaten and Hinton, 2008]. The distribution for October of 2003, where there are many samples around the upper left, is varied from that on February of 2003, which is concentrated around the lower left.

The proposed method learns latest classifiers without ad-

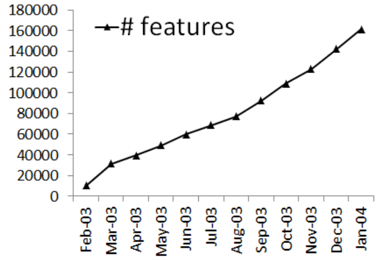


Figure 1: Time variation of cumulative number of features in a real-world spam data set

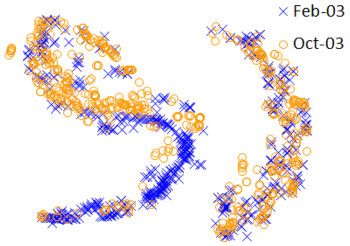


Figure 2: Distributions of a real-world spam data set for February of 2003 and October of 2003, where samples are visualized in two-dimensional space

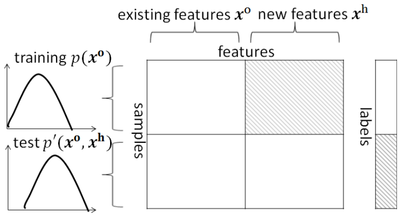

existing features $\boldsymbol{x}^{\mathrm{o}}$
new features $\boldsymbol{x}^{\mathrm{h}}$
features
training $p(\boldsymbol{x}^{\mathrm{o}})$
test $p'(\boldsymbol{x}^{\mathrm{o}}, \boldsymbol{x}^{\mathrm{h}})$
samples
labels

Figure 3: Observed and missing values of given data. Shaded and unshaded parts represent missing and existing values, respectively. Labeled samples are drawn from $p(\boldsymbol{x}^{\mathrm{o}})$, and unlabeled samples are drawn from $p'(\boldsymbol{x}^{\mathrm{o}}, \boldsymbol{x}^{\mathrm{h}})$

ditional labeled data that overcomes both problems, i.e., the emergence of new features and the change of distributions, simultaneously. Newly obtained unlabeled data contain information on the correlation between new features, which do not appear in the labeled data but appear in the unlabeled data, and existing features, which appear in the labeled data as well as in the unlabeled data. The proposed method models the conditional distribution of new features given existing features by a multivariate normal distribution, which is learned using the unlabeled data. The newly obtained unlabeled data also contain information on the distribution change. We consider a situation called "covariate shift", where training and test data follow different distributions but the class label distribution given the features is constant since it has been reported that this assumption matches various real-world problems, such as spam filtering [Bickel and Scheffer, 2007], HIV therapy screening [Bickel *et al.*, 2008], and human activity recognition [Hachiya *et al.*, 2012]. Using the labeled and unlabeled data, the proposed method estimates the density ratio between training and test distributions, which models the distribution change.

We approximate the classification error of a classifier, which exploits new features as well as existing features, at the test phase by incorporating both the conditional distribution of new features and the density ratio between training and test distribution. The approximation is calculated by integrating out new feature values by using newly obtained unlabeled data and labeled data collected beforehand without additional labeled data. By minimizing the upper bound of the approximated error, we obtain a classifier that exploits new features and fits on the test phase.

2 Related Work

The proposed method utilizes imputation techniques for handling the emergence of new features since it can be integrated with the framework for covariate shift in a principled way. The imputation techniques are typically used for obtaining estimates of missing values. For example, a single imputation method estimates a conditional distribution of missing features given the existing features and replaces missing values by the conditional mean [Donders *et al.*, 2006]. Since single imputation methods treat missing values as fixed data, the uncertainty of the estimated missing values is not considered. Multiple imputation methods take the uncertainty into account and have flourished in the statistics community [Rubin, 2004]. They generate multiple samples from the estimated conditional distribution, and the multiple imputed data are used for learning classifiers. Since many samples would be necessary to represent the uncertainty, the computational cost of multiple imputation methods would be high. In comparison, the proposed method takes the uncertainty into account but is efficient since it does not require multiple samples by analytically integrating out missing values using the conditional distribution. Semi-supervised learning methods that use both labeled and unlabeled data for learning classifiers have been developed [Nigam *et al.*, 2000; Grandvalet and Bengio, 2004; Kingma *et al.*, 2014]. Since semi-supervised methods learn classifiers by assigning pseudo labels to unlabeled data, new features that appear only in the unlabeled data can be exploited to the classifiers. However, the existing imputation and semi-supervised methods does not explicitly consider the emergence of new features over time and does not handle the change of the distribution.

Many methods have been developed for covariate shift. [Zhang *et al.*, 2013] proposed to match data distributions in the Hilbert space by aligning kernel matrices across domains. [Liu and Ziebart, 2014] proposed a minmax approach for learning classifiers. Recently, many works have converged to the direction of using importance weights for covariate shift [Shimodaira, 2000; Kanamori *et al.*, 2009; Sugiyama *et al.*, 2013]. This method learns classifiers by weighting a training sample with its importance, which is defined by a ratio between the data distribution at the training phase and the data distribution at the test phase. Many methods for estimating importance have been proposed, such as moment matching [Huang *et al.*, 2006], density matching under the Kullback-Leibler divergence [Sugiyama *et al.*, 2008], and least-squares importance fitting to the ratio [Kanamori *et al.*, 2009]. All of these methods learn classifiers by using the importance weighted training data. This means that features that appear only in the test data but not in the training data cannot be incorporated into the classifiers. The proposed method is the first attempt to incorporate new features for importance weighting methods in covariate shift adaptation.

Our task is related to transfer learning and concept drift.

Transfer learning utilizes data in a source domain to solve a related problem in a target domain [Pan and Yang, 2010]. In our task, we regard data generated until a certain time point as source domain, and data generated after the time point as the target domain since these data follow different distributions. Although various assumptions for the distribution change have been studied in transfer learning, in this paper, we focus on a situation of covariate shift since it has been reported that it matches various real-world applications [Bickel and Scheffer, 2007; Bickel *et al.*, 2008; Hachiya *et al.*, 2012]. Concept drift is an online supervised scenario, where a data distribution changes over time, and many methods for concept drift have been proposed [Gama *et al.*, 2014; Klinkenberg, 2004; Wang *et al.*, 2003; Haque *et al.*, 2016]. Although these methods for concept drift generally assume that labeled data are sequentially given to update classifiers, the proposed method utilizes unlabeled data to fit on the test distribution, and does not require labeled data to update. Methods for predicting future classifiers given only labeled data collected until the current time have been proposed [Kumagai and Iwata, 2016; 2017]. Although these methods are designed to maintain the classification performance, they do not cope with the problems of the emergence of new features and the distribution change since they do not use any additional training data, which contain rich information for the problems.

3 Proposed Method

We introduce notations and define the task studied in this paper. Let $\mathcal{D}{:=}\{(\boldsymbol{x}_n^{\rm o}, y_n)\}_{n=1}^N$ be a set of labeled samples collected until a certain time point, where $\boldsymbol{x}_n^{\rm o} \in \mathbb{R}^{D_{\rm o}}$ is the $D_{\rm o}$-dimensional feature vector of the n-th labeled sample, $y_n \in \{0,1\}$ is its class label, and N is the number of the labeled data. We suppose that the training feature vectors $\{\boldsymbol{x}_n^{\rm o}\}_{n=1}^N$ are drawn from a training distribution $p(\boldsymbol{x}^{\rm o})$. $\mathcal{D}'{:=}\{(\boldsymbol{x}_m^{\rm o}, \boldsymbol{x}_m^{\rm h})\}_{m=1}^M$ is a set of unlabeled samples collected after the time point, where $(\boldsymbol{x}_m^{\rm o}, \boldsymbol{x}_m^{\rm h})$ is the $(D_{\rm o}+D_{\rm h})$-dimensional feature vector of the m-th unlabeled sample, and M is the number of the unlabeled data. The unlabeled data contains new features $\boldsymbol{x}_m^{\rm h} \in \mathbb{R}^{D_{\rm h}}$, which do not appear in the labeled data $\mathcal{D}$ since the unlabeled data are collected after the labeled data are collected. The test feature vectors $\{(\boldsymbol{x}_m^{\rm o}, \boldsymbol{x}_m^{\rm h})\}_{m=1}^M$ are drawn from a test distribution $p'(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h})$. Figure 3 summarizes observed and missing values in a given data in our task. We consider a situation of covariate shift where training and test data follow different distributions but the conditional distribution of the class label given the feature vector is constant between training and test phases: $p(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h}) \neq p'(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h})$, $\quad p(y|\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h}) = p'(y|\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h})$. Our goal is to find classifier $h : \mathbb{R}^{D_{\rm o}+D_{\rm h}} \to \{0,1\}$, which can accurately classify samples drawn from $p'(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h})$, given the labeled and unlabeled data $\mathcal{D} \cup \mathcal{D}'$.

3.1 Our Framework

We obtain a classifier by minimizing the following generalization error G at the test phase,

$$G := \int \mathrm{loss}(y, h(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h}))p'(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h}, y)d\boldsymbol{x}^{\rm o}d\boldsymbol{x}^{\rm h}dy, \tag{1}$$

where $\mathrm{loss}(y, y')$ denotes any loss function that outputs a point-wise error when y is predicted by y'. The generalization error G is the expectation of the loss function with the test distribution. This generalization error G can be rearranged as

$$G = \int \mathrm{loss}(y, h(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h}))\frac{p'(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h})}{p(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h})}p(\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h}, y)d\boldsymbol{x}^{\rm o}d\boldsymbol{x}^{\rm h}dy, \tag{2}$$

where the assumption, $p(y|\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h}) = p'(y|\boldsymbol{x}^{\rm o}, \boldsymbol{x}^{\rm h})$, is used. Since the new features do not appear in the labeled training data, it is impossible to obtain the distribution of the new features from the labeled training data. Thus, we assume that the distribution of the new features given the existing features in the training phase is the same with the distribution in the test phase, $p(\boldsymbol{x}^{\rm h}|\boldsymbol{x}^{\rm o}) = p'(\boldsymbol{x}^{\rm h}|\boldsymbol{x}^{\rm o})$ which enables us to learn classifiers that exploits new features. Then, the generalization error G is approximated by the following empirical risk,

$$G \approx \frac{1}{N}\sum_n \tau(\boldsymbol{x}_n^{\rm o}) \int \mathrm{loss}(y_n, h(\boldsymbol{x}_n^{\rm o}, \boldsymbol{x}_n^{\rm h}))p(\boldsymbol{x}_n^{\rm h}|\boldsymbol{x}_n^{\rm o})d\boldsymbol{x}_n^{\rm h}, \tag{3}$$

which is estimated using the labeled training data and the unlabeled test data. Here, $\tau(\boldsymbol{x}^{\rm o}) := \frac{p'(\boldsymbol{x}^{\rm o})}{p(\boldsymbol{x}^{\rm o})}$ is the importance weight at $\boldsymbol{x}^{\rm o}$. This weight depends only on the existing features but not on the newly emerged features. The empirical risk at the test phase (3) can be regarded as a importance-weighted average of the empirical risk at the training phase. To minimize the empirical risk (3), we must estimate the conditional distribution $p(\boldsymbol{x}_n^{\rm h}|\boldsymbol{x}_n^{\rm o})$ and the importance weight $\tau(\boldsymbol{x}^{\rm o})$ at first.

3.2 Conditional Distributions of New Features

We estimate the distribution of the new features given the existing features $p(\boldsymbol{x}_n^{\rm h}|\boldsymbol{x}_n^{\rm o})$ by assuming it is modeled as the following multivariate normal distribution,

$$p(\boldsymbol{x}_n^{\rm h}|\boldsymbol{x}_n^{\rm o}) = \mathcal{N}\left(\boldsymbol{x}_n^{\rm h} \mid \boldsymbol{A}\boldsymbol{x}_n^{\rm o} + \boldsymbol{a}, \boldsymbol{\Lambda}^{-1}\right), \tag{4}$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\boldsymbol{A}$ is a $D_{\rm h} \times D_{\rm o}$ matrix for defining transformation from existing features $\boldsymbol{x}^{\rm o}$ to new features $\boldsymbol{x}^{\rm h}$, $\boldsymbol{a} \in \mathbb{R}^{D_{\rm h}}$ is a bias term, and $\boldsymbol{\Lambda}$ is a $D_{\rm h} \times D_{\rm h}$ precision matrix. We assume that the matrix $\boldsymbol{A}$ is low-rank and can be decomposed into a product of two matrixes, $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{C}$, $\boldsymbol{B} \in \mathbb{R}^{D_{\rm h} \times K}$, $\boldsymbol{C} \in \mathbb{R}^{K \times D_{\rm o}}$. We can decrease the number of parameters by setting $K < \frac{D_{\rm h}D_{\rm o}}{D_{\rm h}+D_{\rm o}}$, which enables us to handle high-dimensional feature vectors. In addition, for simplicity, we restrict the precision matrix $\boldsymbol{\Lambda}$ to a diagonal matrix as $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})^2 + \epsilon\boldsymbol{I}_{D_{\rm h}}$, where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{D_{\rm h}}) \in \mathbb{R}^{D_{\rm h}}$ is a precision vector, $\mathrm{diag}(\boldsymbol{x})$ returns a diagonal matrix whose diagonal elements are $\boldsymbol{x}$, ϵ is a positive constant, and $\boldsymbol{I}_k$ is the $k \times k$ identity matrix. The term $\epsilon\boldsymbol{I}_{D_{\rm h}}$ is added in order to ensure the positive definiteness of $\boldsymbol{\Lambda}$. We estimate the unknown parameters $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{a}$, and $\boldsymbol{\lambda}$ on the basis of the maximum a posteriori (MAP) estimation by using unlabeled data $\mathcal{D}' = \{(\boldsymbol{x}_m^{\rm o}, \boldsymbol{x}_m^{\rm h})\}_{m=1}^M$. The conditional density $p(\boldsymbol{x}_n^{\rm h}|\boldsymbol{x}_n^{\rm o})$ are estimated by using samples $\mathcal{D}'$ drawn from test distribution $p'(\boldsymbol{x})$ but not using samples drawn from training distribution $p(\boldsymbol{x})$ since $p(\boldsymbol{x}^{\rm h}|\boldsymbol{x}^{\rm o}) = p'(\boldsymbol{x}^{\rm h}|\boldsymbol{x}^{\rm o})$ and the training data do not contain new features. For the priors of $\boldsymbol{B}$ and $\boldsymbol{C}$, we assume the normal distributions $\mathcal{N}(\mathbf{0}_{D_\mathrm{h}K}, r^{-1}\mathbf{I}_{D_\mathrm{h}K})$ and $\mathcal{N}(\mathbf{0}_{D_\mathrm{o}K}, r^{-1}\mathbf{I}_{D_\mathrm{o}K})$, where $\mathbf{0}_k$ denotes the k-dimensional zero vector and $r \in \mathbb{R}_+$ is a precision parameter. For the prior of the elements of the precision vector λ_k^2, we assume a Gamma distribution $\mathrm{Gam}(\lambda_k^2|a, b)$. Then, the log posterior $F := \log p(\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{a}, \boldsymbol{\lambda}|\mathcal{D}')$ is given by

$$\begin{aligned} F =& \log p(\mathcal{D}'|\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{a}, \boldsymbol{\lambda}) + \log p(\boldsymbol{B}) + \log p(\boldsymbol{C}) + \log p(\boldsymbol{\lambda}^2) \\ =& -\frac{1}{2}\sum_m (\boldsymbol{x}_m^\mathrm{h} - \boldsymbol{B}\boldsymbol{C}\boldsymbol{x}_m^\mathrm{o} - \boldsymbol{a})^\top \boldsymbol{\Lambda}(\boldsymbol{x}_m^\mathrm{h} - \boldsymbol{B}\boldsymbol{C}\boldsymbol{x}_m^\mathrm{o} - \boldsymbol{a}) \\ & - \frac{r}{2}\mathrm{Tr}(\boldsymbol{B}^\top \boldsymbol{B}) - \frac{r}{2}\mathrm{Tr}(\boldsymbol{C}^\top \boldsymbol{C}) + \frac{1}{2}\log\det(\boldsymbol{\Lambda}) \\ & + \sum_k 2(a-1)\log\lambda_k - b\lambda_k^2 + const, \end{aligned} \tag{5}$$

where Tr is a trace, $\top$ denotes a transposition, and det is a determinant. The log posterior F is maximized using gradient-based methods over the parameters $\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{a}, \boldsymbol{\lambda}$. Although we use a multivariate normal distribution to model $p(\boldsymbol{x}_n^\mathrm{h}|\boldsymbol{x}_n^\mathrm{o})$ for simplicity, it is possible to use other distributions such as Gaussian mixtures and Bernoulli distributions. In addition, although we modeled the mean of the multivariate normal distribution of $\boldsymbol{x}^\mathrm{h}$ given $\boldsymbol{x}^\mathrm{o}$ by linear regression, it is possible to use more powerful non-linear functions such as neural nets.

3.3 Importance Weight

We utilize the unconstrained least-square importance fitting approach (uLSIF) [Kanamori *et al.*, 2009] for estimating the importance weight $\tau(\boldsymbol{x}^\mathrm{o})$. Since the uLSIF was shown to have excellent numerical stability and efficient run-time solution [Sugiyama *et al.*, 2013], we chose the uLSIF as the importance estimating method. The uLSIF models the importance $\tau(\boldsymbol{x}^\mathrm{o})$ by using the following linear model, $\hat{\tau}(\boldsymbol{x}^\mathrm{o}) = \boldsymbol{\alpha}^\top \boldsymbol{\psi}(\boldsymbol{x}^\mathrm{o})$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_L) \in \mathbb{R}^L$ is a parameter vector, L is the number of parameters, and $\boldsymbol{\psi}(\boldsymbol{x}^\mathrm{o}) \in \mathbb{R}^L$ are basis functions. Although the original uLSIF implicitly assumes that the importance weight depends on both existing and new features, the uLSIF utilized in the proposed method depends only on the existing features. We use the Gaussian kernel model centered at the existing features of unlabeled data $\boldsymbol{x}_m^o$ as basis functions: $\hat{\tau}(\boldsymbol{x}^\mathrm{o}) = \sum_{m=1}^M \alpha_m \exp\left(-\frac{\|\boldsymbol{x}^\mathrm{o} - \boldsymbol{x}_m^o\|^2}{2\gamma^2}\right)$, where $\|\cdot\|$ is ℓ_2-norm and γ^2 denotes the Gaussian width. The parameter vector $\boldsymbol{\alpha}$ is learned so that the following objective function $J(\boldsymbol{\alpha})$ is minimized,

$$J(\boldsymbol{\alpha}) = \int (\tau(\boldsymbol{x}^\mathrm{o}) - \hat{\tau}(\boldsymbol{x}^\mathrm{o}))^2 p(\boldsymbol{x}^\mathrm{o}) d\boldsymbol{x}^\mathrm{o} + \rho\|\boldsymbol{\alpha}\|^2, \tag{6}$$

which is the expected squared error with ℓ_2 regularization. With the empirical approximation, this objective function can be rearranged as

$$J(\boldsymbol{\alpha}) \approx \boldsymbol{\alpha}^\top \boldsymbol{H}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \boldsymbol{h} + \rho\|\boldsymbol{\alpha}\|^2, \tag{7}$$

where $\boldsymbol{H} = \frac{1}{N}\sum_n \psi(\boldsymbol{x}_n^\mathrm{o})\psi(\boldsymbol{x}_n^\mathrm{o})^\top$ and $\boldsymbol{h} = \frac{1}{M}\sum_m \psi(\boldsymbol{x}_m^\mathrm{o})$. This optimization problem is a convex and its minimizer $\hat{\boldsymbol{\alpha}}$ can be analytically calculated by $\hat{\boldsymbol{\alpha}} = \left(\frac{1}{N}\sum_n \psi(\boldsymbol{x}_n^\mathrm{o})\psi(\boldsymbol{x}_n^\mathrm{o})^\top + \rho\boldsymbol{I}\right)^{-1}\left(\frac{1}{M}\sum_m \psi(\boldsymbol{x}_m^\mathrm{o})\right)$. Since elements of the estimated parameter $\hat{\boldsymbol{\alpha}}$ can be negative, we modify them with $\tilde{\boldsymbol{\alpha}} = \max(\mathbf{0}_L, \hat{\boldsymbol{\alpha}})$, where max is applied in the element-wise manner [Kanamori *et al.*, 2009].

3.4 Learning Classifiers with Importance Weights While Integrating out New Features

We employ the negative log likelihood $-\log p(y_n|\boldsymbol{x}_n^\mathrm{o}, \boldsymbol{x}_n^\mathrm{h})$ as a loss function in (3) though it is possible to use other loss functions such as the 0/1-loss or hinge-loss. We assume that the conditional probability of class label y given feature vector $(\boldsymbol{x}^\mathrm{o}, \boldsymbol{x}^\mathrm{h})$ is modeled by logistic regression, $p(y = 1|\boldsymbol{x}^\mathrm{o}, \boldsymbol{x}^\mathrm{h}, \boldsymbol{w}) = \sigma\left(\boldsymbol{w}^\top(\boldsymbol{x}^\mathrm{o}, \boldsymbol{x}^\mathrm{h})\right)$, where $\boldsymbol{w} \in \mathbb{R}^{D_\mathrm{o}+D_\mathrm{h}}$ is a parameter vector of the classifier h, and $\sigma(\cdot)$ is the sigmoid function. We would like to minimize the empirical risk (3) with respect to the model parameter $\boldsymbol{w}$. However, the aforementioned integral is intractable because of the non-conjugacy of $p(y|\boldsymbol{x}^\mathrm{o}, \boldsymbol{x}^\mathrm{h}, \boldsymbol{w})$. To deal with this problem, we use the following inequality [Bishop, 2006],

$$\begin{aligned} & p(y_n|\boldsymbol{x}_n^\mathrm{o}, \boldsymbol{x}_n^\mathrm{h}, \boldsymbol{w}) \geq \\ & \qquad e^{y_n s_n}\sigma(\xi_n)\exp\left(-\frac{s_n + \xi_n}{2} - g(\xi_n)(s_n^2 - {\xi_n}^2)\right), \end{aligned} \tag{8}$$

where $s_n := \boldsymbol{w}^\top(\boldsymbol{x}_n^\mathrm{o}, \boldsymbol{x}_n^\mathrm{h})$, $\xi_n \in \mathbb{R}$ is a parameter that is associated with each labeled training sample and controls the accuracy of the approximation, and $g(\xi_n) = \frac{1}{2\xi_n}\left(\sigma(\xi_n) - \frac{1}{2}\right)$. By substituting the term on the right side of (8) with the negative log likelihood in the empirical risk (3), we obtain the new objective function $\tilde{G}$. Since $\tilde{G}$ is the upper bound of the original objective function G, decreasing the value of $\tilde{G}$ leads to an decreased value of G. Using the first and second moments of the normal distributions, $\int \boldsymbol{x}_n^\mathrm{h} p(\boldsymbol{x}_n^\mathrm{h}|\boldsymbol{x}_n^\mathrm{o}) d\boldsymbol{x}_n^\mathrm{h} = \boldsymbol{B}\boldsymbol{C}\boldsymbol{x}_n^\mathrm{o} + \boldsymbol{a}$ $(=: \mu(\boldsymbol{x}_n^\mathrm{o}))$, $\int \boldsymbol{x}_n^\mathrm{h}\boldsymbol{x}_n^{\mathrm{h}\top} p(\boldsymbol{x}_n^\mathrm{h}|\boldsymbol{x}_n^\mathrm{o}) d\boldsymbol{x}_n^\mathrm{h} = \mu(\boldsymbol{x}_n^\mathrm{o})\mu(\boldsymbol{x}_n^\mathrm{o})^\top + \boldsymbol{\Lambda}^{-1}$, we write the objective function $\tilde{G}$ with ℓ_2- regularizer to be minimized as follows,

$$\begin{aligned} \tilde{G} =& \sum_n \hat{\tau}(\boldsymbol{x}_n^\mathrm{o}) \int \Bigg(\frac{1}{2}\xi_n - g(\xi_n)\xi_n^2 - \log\sigma(\xi_n) + (\frac{1}{2} - y_n) \\ & \times \boldsymbol{x}_n^{\mathrm{o}\top}\boldsymbol{w}_\mathrm{o} + g(\xi_n)(\boldsymbol{x}_n^{\mathrm{o}\top}\boldsymbol{w}_\mathrm{o})^2\Bigg) p(\boldsymbol{x}_n^\mathrm{h}|\boldsymbol{x}_n^\mathrm{o}) \\ & + \Bigg((\frac{1}{2} - y_n)\boldsymbol{x}_n^{\mathrm{h}\top}\boldsymbol{w}_\mathrm{h} + 2g(\xi_n)\boldsymbol{x}_n^{\mathrm{o}\top}\boldsymbol{w}_\mathrm{o}\boldsymbol{x}_n^{\mathrm{h}\top}\boldsymbol{w}_\mathrm{h} \\ & + g(\xi_n)(\boldsymbol{x}_n^{\mathrm{h}\top}\boldsymbol{w}_\mathrm{h})^2\Bigg) p(\boldsymbol{x}_n^\mathrm{h}|\boldsymbol{x}_n^\mathrm{o}) d\boldsymbol{x}_n^\mathrm{h} + \frac{1}{2}c\|\boldsymbol{w}\|^2 \\ =& \sum_n \hat{\tau}(\boldsymbol{x}_n^\mathrm{o}) \Bigg(\frac{1}{2}\xi_n - g(\xi_n)\xi_n^2 - \log\sigma(\xi_n) + (\frac{1}{2} - y_n) \\ & \times \boldsymbol{x}_n^{\mathrm{o}\top}\boldsymbol{w}_\mathrm{o} + g(\xi_n)(\boldsymbol{x}_n^{\mathrm{o}\top}\boldsymbol{w}_\mathrm{o})^2 + (\frac{1}{2} - y_n)\mu(\boldsymbol{x}_n^\mathrm{o})^\top\boldsymbol{w}_\mathrm{h} \\ & + 2g(\xi_n)\boldsymbol{x}_n^{\mathrm{o}\top}\boldsymbol{w}_\mathrm{o}\mu(\boldsymbol{x}_n^\mathrm{o})^\top\boldsymbol{w}_\mathrm{h} + g(\xi_n)[(\mu(\boldsymbol{x}_n^\mathrm{o})^\top\boldsymbol{w}_\mathrm{h})^2 \\ & + \boldsymbol{w}_\mathrm{h}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{w}_\mathrm{h}]\Bigg) + \frac{1}{2}c\|\boldsymbol{w}\|^2, \end{aligned} \tag{9}$$

where $\boldsymbol{w} = (\boldsymbol{w}_\mathrm{o}, \boldsymbol{w}_\mathrm{h}) \in \mathbb{R}^{D_\mathrm{o}+D_\mathrm{h}}$ and c is a positive constant. The objective function $\tilde{G}$ is minimized using gradient-based methods over the parameters $\boldsymbol{w} = (\boldsymbol{w}_\mathrm{o}, \boldsymbol{w}_\mathrm{h})$, $\{\xi_n\}_{n=1}^N$.

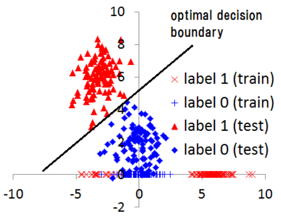


Figure 4: Synthetic Data in case of $(\pi_1, \pi_2) = (0.15, 0.75)$

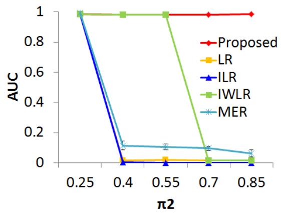


Figure 5: Average and standard error of AUC varying mixture ratio (π_1, π_2) in synthetic data

4 Experiments

We conducted experiments using one synthetic and three real-world data sets to confirm the effectiveness of the proposed method.

4.1 Synthetic Data

Each training sample $x_n^{\mathrm{o}} \in \mathbb{R}$ is drawn from $\pi_1 \mathcal{N}(-3, 1) + \pi_2 \mathcal{N}(6, 1)$ if $y_n = 1$ and $\mathcal{N}(0, 1)$ if $y_n = 0$, where $\pi_i \in \mathbb{R}_+$ are mixture ratios satisfying $\pi_1 + \pi_2 = 1$. Each test sample $\boldsymbol{x}_m = (x_m^{\mathrm{o}}, x_m^{\mathrm{h}}) \in \mathbb{R}^2$ is drawn from $\mathcal{N}((-3, 6), \mathbf{I}_2)$ if $y_m = 1$ and $\mathcal{N}((0, 2), \mathbf{I}_2)$ if $y_m = 0$, where new feature x_m^{h} is added in the test distribution. We illustrate this data in Figure 4. In the experiments, we generate 100 labeled training samples for each label y and 100 test samples for each label y.

4.2 Real-world Data

We used three real-world data sets: SPAM, URL, and LOG. SPAM is a collection of spam and legitimate email received by one from February 1st of 2003 to January 31th of 2004 [Gama *et al.*, 2014]. This data set is the same as that stated in Figure 1 and 2. The total number of emails is $11,905$, and the number of features is $166,047$. URL is a public data set of malicious and normal urls collected over 120 days [Ma *et al.*, 2009]. The total number of examples is $2,396,130$, and the number of features 3,231,961. These two data sets are well-used in concept drift literatures. LOG is our in-house data collection of malicious and normal WebProxy logs. Malicious WebProxy logs were created on the basis of malware communications. Since malware evolve over time, the malware communications also change over time. Each log includes information such as source IP address, destination IP address and url.

4.3 Setting

In our experiments using the synthetic data set, we evaluated AUC, which is a well-used evaluation measure for classification tasks, varying the values of the mixture ratio (π_1, π_2). We created ten different sets and evaluated the average AUC for each mixture ratio (π_1, π_2). In our experiments using the real-world data sets, we calculated AUC by using samples collected until a certain time point for labeled training data and samples after the certain time point for unlabeled training and test data for evaluating the classification performance on latest data. For SPAM, roughly, samples collected in the n-th month were used for labeled data, samples in the $n{+}1$-th month for unlabeled data, and samples in the $n + 2$-th month for test data, where n is $2, 3, \cdots, 11$. Therefore, we obtained ten different sets in total, and we evaluated the average AUC over the these sets. For URL, we first split 120 days, making a split every 10 days, and samples collected in the n-th unit were used for labeled data, samples in the $n + 1$-th unit for unlabeled data, and samples in the $n + 2$-th unit for test data, where n is $1, 2, \cdots, 10$. For LOG, samples collected in a day were used for labeled data, and samples collected over two days after the day, which labeled data were collected in, were used for unlabeled or test data.

We compared the proposed method with four existing methods: logistic regression (LR), imputation logistic regression (ILR), importance weighted logistic regression (IWLR) and semi-supervised logistic regression by minimum entropy regularization (MER). LR learns classifiers by using only labeled data $\mathcal{D}{:=}\{(\boldsymbol{x}_n^{\mathrm{o}}, y_n)\}_{n=1}^N$. ILR learns classifiers by logistic regression after completing labeled data $\boldsymbol{x}_n^{\mathrm{o}}$ with the conditional mean in (4). IWLR learns classifiers by logistic regression with the importance weighted method (specifically, by uLSIF [Kanamori *et al.*, 2009]). MER is a semi-supervised extension of logistic regression [Grandvalet and Bengio, 2004]. MER learns classifiers so as to separate unlabeled data as much as possible on the basis of minimum entropy regularization. To evaluate the effectiveness of considering new features and importance weights, we used logistic regression for classifiers with all methods including the proposed method.

In our experiments, we chose the optimal hyper parameters for these methods from the following variations by using validation data: regularization parameter for classifiers $c \in \{10^{-1}, 1, 10^1\}$ in all methods, regularization parameter for importance $\rho \in \{10^{-1}, 1, 10^1\}$ in the proposed method and IWLR, regularization parameter for imputation $r, b \in \{10^{-1}, 1, 10^1\}$, $a \in \{1\}$, and imputation parameter $K \in \{1, 3, 6, 9\}$ in the proposed method and ILR. For the proposed method and IWLR, the Gaussian width of the Gaussian Kernel γ is determined by *median trick*, that is, γ is set by the median of squared distance between training points (labeled and unlabeled data). The weighting parameter of unlabeled data for MER is chosen in $\{\{0.1 \cdot 10^{-n}, 0.5 \cdot 10^{-n}\}_{n=0}^3, 1\}$, The correction term ϵ is set by 10^{-4} for the proposed method and ILR.

4.4 Results

We first show the classification performance in the synthetic data sets. Figure 5 shows the average and standard error of AUC, varying the values of the mixture ratio (π_1, π_2). When π_2 was 0.25, there was not much difference between each method. This is because the number of training samples with $y = 1$ located on the left side was much larger than training samples with $y = 1$ located on the right side, and therefore, the optimal decision boundary, whose normal vector points to the left direction, can be easily learned in every method. When π_2 was 0.4 and 0.55, the AUCs of the methods except the proposed method and IWLR became poor since they were affected by a lot of training samples with $y = 1$ located on the right side. However, the proposed method and IWLR could

Table 1: Average and standard error of AUC over $N/M = \{0.25, 0.5, 1.0\}$, where N labeled samples and M fixed unlabeled samples. Boldface indicates the best method, which is significantly better than the method(s) marked with $*$ (in paired t-test, p=0.05)

	Proposed	LR	ILR	IWLR	MER
SPAM	**0.8691** (0.0162)	0.8376 (0.0231)*	0.8371 (0.0227)*	0.8393 (0.0224)*	0.8311 (0.0231)*
URL	**0.9845** (0.0016)	0.9838 (0.0017)*	0.9839 (0.0017)*	0.9839 (0.0017)*	0.9841 (0.0017)
LOG	**0.6489** (0.0111)	0.6399 (0.0111)*	0.6482 (0.0109)	0.6403 (0.0109)*	0.6398 (0.0111)*

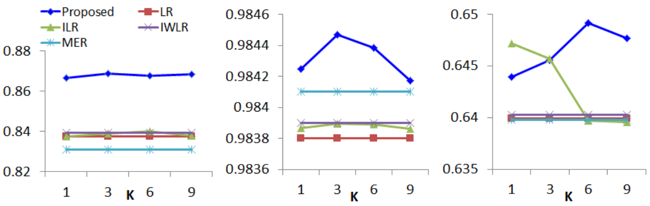
Table 2: Average number of existing features D_{o} and new features D_{h} in the case when #labeled samples equals to #unlabeled samples $(N/M = 1.0)$

Dataset	D_{o}	D_{h}
SPAM	14,915	10,983
URL	9,677	6,740
LOG	20,670	15,916

learn a good decision boundary since the importance weights of training samples located on the left side are bigger than those located on the right side. When π_2 was 0.7 and 0.85, only the proposed method achieved a high AUC, although IWLR became inaccurate. This is because the importance weights estimated by the proposed method only depend on existing features $\boldsymbol{x}^{\mathrm{o}}$, although the importance weights with IWLR depend on existing and new features $(\boldsymbol{x}^{\mathrm{o}}, \boldsymbol{x}^{\mathrm{h}})$, and therefore, IWLR tends to give large importance weights to training samples with $y = 1$ on the right side. Overall, the proposed method achieved a good classification performance compared with the others in the situation, where there were new features and training and test distributions differ.

Next, we investigate the classification performance when varying the number of labeled samples for the real-world data sets. Here we set the number of unlabeled and test samples uniformly so that it did not exceed the smallest number of samples in each time unit (here, the time unit is a month in the case of SPAM, ten days in the case of URL, and a day in the case of LOG). As a result, the number of unlabeled samples for SPAM, URL and LOG were 500, 1000 and 600, respectively. The number of test samples was also the same number of unlabeled samples. Table 2 shows the number of existing features D_{o} and new features D_{h} in the case of $N/M = 1.0$, where N is the number of labeled samples and M is the number of fixed unlabeled samples, for each real-world data set. Table 1 shows the average and standard error of AUC over $N/M = \{0.25, 0.5, 1.0\}$. For all data sets, the proposed method achieved the highest average of AUC over all N/M. In addition, the proposed method outperformed the other methods in many N/M (the proposed method showed the best results 6 out of 9 times). Although ILR and IWLR somewhat improved the average of AUC over all N/M compared with LR, the proposed method improved AUC further. This result suggests that it is better to learn classifiers taking into account two problems, that is, the emergence of new features and the distribution change, at the same time more than when they are considered separately.

Last, we investigated how the classification performance of the proposed method changed against imputation parameter K, which determines the number of parameters to be estimated for the conditional probability with new features given existing features. The number of labeled and unlabeled samples was same as in the earlier experiments. Figure 6 represents the averages of AUC over $N/M = \{0.25, 0.5, 1.0\}$ when changing the value of imputation parameter K within $\{1, 3, 6, 9\}$. Here, the AUCs of LR, IWLR, and MER were constant when varying the value of K since they do not depend on the value of K. Overall, the proposed method achieved a good classification performance for all the values of imputation parameter K. For SPAM and URL, the AUC of the proposed method was better than the other methods for all imputation parameters K. For LOG, the proposed method outperformed the others except in the case of $K = 1$. Nevertheless, even with $K = 1$, the proposed method achieved a relatively good classification performance. One of the reasons the proposed method with $K = 1$ achieved a good classification performance is that only a small part of the new features affected the classification performance. That is, there is a possibility that only a small K is enough to improve the classification performance even though it is not sufficient to represent a correlation of new and existing features perfectly. As a result, we assume that the proposed method requires only a small K to classify data correctly.



(a) SPAM (b) URL (c) LOG

Figure 6: Average of AUC over the N/M when varying value of imputation parameter K

5 Conclusion

We proposed a novel method for learning the latest classifiers by using labeled data collected beforehand and newly obtained unlabeled data. In experiments, we confirmed that the proposed method outperformed the various existing methods in the situation , where there were new features and training and test distributions differ. As future work, we will extend the proposed method to on-line learning in order to be able to apply it to large-scale data. In addition, applying deep learning methods for modeling conditional distribution of new features is also interesting.

References

[Abdallah *et al.*, 2012] Zahraa Said Abdallah, Mohamed Medhat Gaber, Bama Srinivasan, and Shonali Krishnaswamy. Streamar: incremental and active learning with evolving sensory data for activity recognition. In *ICTAI*, 2012.

[Bickel and Scheffer, 2007] Steffen Bickel and Tobias Scheffer. Dirichlet-enhanced spam filtering based on biased samples. *NIPS*, 2007.

[Bickel *et al.*, 2008] Steffen Bickel, Jasmina Bogojeska, Thomas Lengauer, and Tobias Scheffer. Multi-task learning for hiv therapy screening. In *ICML*, 2008.

[Bishop, 2006] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[Crammer *et al.*, 2009] Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. In *NIPS*, 2009.

[Donders *et al.*, 2006] A Rogier T Donders, Geert JMG van der Heijden, Theo Stijnen, and Karel GM Moons. Review: a gentle introduction to imputation of missing values. *JCE*, 2006.

[Fdez-Riverola *et al.*, 2007] Florentino Fdez-Riverola, Eva Lorenzo Iglesias, Fernando Díaz, José Ramon Méndez, and Juan M Corchado. Applying lazy learning algorithms to tackle concept drift in spam filtering. *ESWA*, 2007.

[Gama *et al.*, 2014] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *CSUR*, 2014.

[Grandvalet and Bengio, 2004] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2004.

[Hachiya *et al.*, 2012] Hirotaka Hachiya, Masashi Sugiyama, and Naonori Ueda. Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing*, 2012.

[Haque *et al.*, 2016] Ahsanul Haque, Latifur Khan, and Michael Baron. Sand: Semi-supervised adaptive novel class detection and classification over data stream. In *AAAI*, 2016.

[Huang *et al.*, 2006] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *NIPS*, 2006.

[Kanamori *et al.*, 2009] Takafumi Kanamori, Shohei Hido, and Masahi Sugiyama. Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection. In *NIPS*, 2009.

[Kingma *et al.*, 2014] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *NIPS*, 2014.

[Klinkenberg, 2004] Ralf Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 2004.

[Kumagai and Iwata, 2016] Atsutoshi Kumagai and Tomoharu Iwata. Learning future classifiers without additional data. In *AAAI*, 2016.

[Kumagai and Iwata, 2017] Atsutoshi Kumagai and Tomoharu Iwata. Learning non-linear dynamics of decision boundaries for maintaining classification performance. In *AAAI*, 2017.

[Li *et al.*, 2010] Yan Li, Hiroyuki Kambara, Yasuharu Koike, and Masashi Sugiyama. Application of covariate shift adaptation techniques in brain–computer interfaces. *Biomedical Engineering, IEEE Transactions on*, 2010.

[Liu and Ziebart, 2014] Anqi Liu and Brian Ziebart. Robust classification under sample selection bias. In *NIPS*. 2014.

[Ma *et al.*, 2009] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Identifying suspicious urls: an application of large-scale online learning. In *ICML*, 2009.

[Nigam *et al.*, 2000] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 2000.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 2010.

[Rubin, 2004] Donald B Rubin. *Multiple imputation for nonresponse in surveys*. John Wiley & Sons, 2004.

[Shimodaira, 2000] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *JSPI*, 2000.

[Sugiyama *et al.*, 2008] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, 2008.

[Sugiyama *et al.*, 2013] Masashi Sugiyama, Makoto Yamada, and Marthinus Christoffel du Plessis. Learning under nonstationarity: covariate shift and class-balance change. *Wiley Interdisciplinary Reviews: Computational Statics*, 2013.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008.

[Wang *et al.*, 2003] Haixun Wang, Wei Fan, Philip S Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *KDD*, 2003.

[Zhang *et al.*, 2013] Kai Zhang, Vincent Wenchen Zheng, Qiaojun Wang, James Tin-Yau Kwok, Qiang Yang, and Ivan Marsic. Covariate shift in hilbert space: a solution via sorrogate kernels. In *ICML*, 2013.