# Drift Adaptation via Joint Distribution Alignment

Bin Zhang
*Centre for Artificial Intelligence*
*University of Technology Sydney*
Sydney, Australia
Bin.Zhang-4@student.uts.edu.au

Jie Lu
*Centre for Artificial Intelligence*
*University of Technology Sydney*
Sydney, Australia
Jie.Lu@uts.edu.au

Guangquan Zhang
*Centre for Artificial Intelligence*
*University of Technology Sydney*
Sydney, Australia
Guangquan.Zhang@uts.edu.au

***Abstract*—Machine learning in evolving environment faces challenges due to concept drift. Most concept drift adaptation methods focus on modifying the model. In this paper, a method, Drift Adaptation via Joint Distribution Alignment (DAJDA), is proposed. DAJDA performs a linear transformation to the drift instances instead of modifying model. Instances are transformed into a common feature space, reducing the discrepancy of distributions before and after drift. Experimental studies show that DAJDA has abilities to improve the performance of learning model under concept drift.**

***Index Terms*—concept drift adaptation, Maximum Mean Discrepancy, machine learning**

## I. Introduction

Large-scale streaming data are generated as the development of the Internet. Statistical learning methods have shown advantages in recognizing the pattern hidden behind the data, and been applied in variety of fields, including email filtering, recommend systems, etc. Traditional statistical learning methods are under the stationary distribution assumption. which is not established in data stream cases. The distribution of the data changes by time, known as concept drift. [1]

In the last decades, lots of concept drift adaptation methods have been proposed. A clear taxonomy of existing concept drift adaptation methods is given in [2]. Most methods reconstruct the classifier when concept drift is detected. However, drift only occurs in some regions rather than the whole feature space. Global adaptation strategies are waste of computation. Some methods leverage the property of tree-based algorithms that the feature space is separated as several hyper-rectangle and each one is represented by a leaf node. These methods identify the region where concept drift occurs and replace the classifier only in the drift region. Though local adaptation methods have flexibility to adjust rather than retrain the classifier, they are restricted to a specific base learner. You cannot have your cake and eat it too.

To break the learner limits of current local drift adaptation method, we divert the focus from adjusting classifiers to transforming instances, and proposed a concept drift adaptation method based on Joint Distribution Adaptation (JDA) [3], named Drift Adaptation via Joint Distribution Alignment (DAJDA). JDA is well-known as an effective transfer learning method. Transfer learning hope to improve the learning model in a specific domain, which is usually noted as target domain, using the knowledge in a related source domain [4]. Concept drift adaptation can be considered as to improve the learning model for newly arrived data using knowledge learnt from historical data.

JDA trains a classifier based on instances of source domain firstly and generates pseudo label on the instances of target domain. The discrepancy of different distributions is measured by Maximum Mean Discrepancy (MMD) [5]. By jointly minimizing the MMD of both marginal distribution and conditional distribution, JDA give the representation of instances from both two domain in a new latent feature space. Different from transfer learning, concept drift adaptation usually assume that the real label of the instance can be obtained a few moment after the prediction. As a consequence, DAJDA estimates the conditional distribution using real label. DAJDA reacts to the drift by transforming the data rather than modifying the model.

Our main contribution is to propose a novel concept drift adaptation method which can overcome the insufficient training problem caused by scarce newly arrived data. We train the classifier on a latent feature space using knowledge learnt from historical data to help predict on the newly arrived data. The rest of the paper is organized as follows. In Section II we give a briefly reviews of concept drift adaptation and transfer learning. In Section III, the detail of our method is given. In Section IV, we conducted several experiments to evaluate our approach. In Section V, we discuss the drawback of DAJDA and the future direction.

## II. Related Work

### A. Concept Drift Adaptation

There are several algorithms which can adjust parts of model to address concept drift problems. Most of these are restricted to a specific type of learning model: some are based on tree model, the others are based on kNN model. CVFDT [6] monitors the accuracy of each node of the tree. A new tree is developed on the node where an accuracy decreasing. The newly developed tree will replace the old one if its performance shows advantages over the old one. Hoeffding Tree [7] is another widely used base learning model due to its ability to limit the error by Hoeffding bound. Hoeffding Adaptive Tree (HAT) [8] and FIMT-DD [9] are Hoeffding tree-based examples. KNN-PAW [10] using a window strategy to maintains a set of samples. Each instance is given a weight. New instances have higher weights, while old ones have lower. Then a kNN model is established from the weighted

instances. Same implementations have been adopted in SAM-KNN [11]. NN-DVI [12] using a kNN-like strategy, named regional density estimation, to detect the region where concept drift occurred.

Ensemble strategies have gained popularity recently in concept drift research community. Examples include Streaming Ensemble Algorithm (SEA) [13], Accuracy Updated Ensemble (AUE2) [14], and the Learn++ algorithm in Non-stationary Environments (Learn++.NSE) [15].

### B. Transfer Learning

Transfer learning has been an attractive research field recent years. A detailed review of transfer learning can be found in [4]. The review divided the existing transfer learning algorithms into four categories: transferring knowledge of instances, transferring knowledge of feature representations, transferring knowledge of parameters and transferring relational knowledge. The feature-representation methods constitute major competent of transfer learning. The idea is to transfer data into a common space and training a common model. Thus, this paper only focuses on the second type. Some methods embed distributions as points in a Grassmann manifold, and generate a geodesic flow. This type of methods include Sampling Geodesic Flow [16] and Geodesic Flow Kernel [17]. Some methods minimize the discrepancy of the distributions. Transfer Component Analysis (TCA) [18] minimizes the distance of marginal distribution. Joint Distribution Adaptation (JDA) [3] improved the TCA and minimize both the marginal distribution and the conditional distribution jointly. Similar methods include Transfer Subspace Learning (TSL) [19] which replaces the MMD by a Bregman divergences-based discrepancy.

## III. Basic Concepts and Notations

### A. Concept Drift

In a data stream, each instance is denoted as $(\mathbf{x}, y)$, where $\mathbf{x} \in \mathbb{R}^k$ and $y \in \{1, 2, \ldots, C\}$. Let instances within a period of time (from $t_1$, to $t_2$) be denoted as $\mathbf{S}_{(t_1,t_2)} = (\mathbf{x}_{(t_1,t_2)}, y_{(t_1,t_2)})$, where $\mathbf{x}_{(t_1,t_2)} = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_1+1}, \ldots, \mathbf{x}_{t_2}\}$, and $y_{(t_1,t_2)} = \{y_{t_1}, y_{t_1+1}, \ldots, y_{t_2}\}$. Formally, the concept drift means that $\exists t$, such that the distribution of instances changes,

$$P(\mathbf{S}_{(t_1,t)}) \neq P(\mathbf{S}_{(t+1,t_2)}).$$

The inequality shows that the learning model trained based on the historical data does not work very well on the newly arrived instances. Concept drift adaptation algorithms are needed.

### B. Maximum Mean Discrepancy

To measure the discrepancy of distributions of two groups of instances, Maximum Mean Discrepancy (MMD) is introduced. MMD embeds each distribution into a Reproducing Kernel Hilbert Space. Let a linear transformation $A$ be the kernel-induced map. Then we have the empirical estimate of MMD between $P(\mathbf{X})$ and $P(\mathbf{Y})$ is,

$$d_{MMD} = ||\frac{1}{n_1}\sum_{i=1}^{n_1} A\mathbf{X}_i - \frac{1}{n_2}\sum_{j=1}^{n_2} A\mathbf{Y}_j||^2_{\mathcal{H}}$$

where $||\cdot||_{\mathcal{H}}$ is the RKHS norm.

## IV. Proposed Method

We refer to the idea of the JDA to handle concept drift adaptation. We hope to find a linear transformation matrix $A \in \mathbb{R}^{k\times k}$, such that though the distributions have changed, i.e.

$$P(\mathbf{S}_{(t_1,t)}) \neq P(\mathbf{S}_{(t+1,t_2)}).$$

the transformed data stream $\bar{\mathbf{S}}_{(t_1,t_2)} = (\bar{\mathbf{x}}_{(t_1,t_2)}, y_{(t_1,t_2)})$ , where $\bar{\mathbf{x}}_{(t_1,t_2)} = \{A\mathbf{x}_{t_1}, A\mathbf{x}_{t_1+1}, \ldots, A\mathbf{x}_{t_2}\}$ has the property that,

$$P(\bar{\mathbf{S}}_{(t_1,t)}) \approx P(\bar{\mathbf{S}}_{(t+1,t_2)})$$

That is,

$$\min_A \quad d_{MMS}(P(\bar{\mathbf{S}}_{(t_1,t)}), P(\bar{\mathbf{S}}_{(t+1,t_2)})) \tag{1}$$

Solving the optimization problem in Equation (1) directly is not trivial. According to the definition of the conditional probability, $P(\mathbf{S}_{(t_1,t_2)}) = P(\mathbf{x}_{(t_1,t_2)})P(y|\mathbf{x}_{(t_1,t_2)})$. We separate the marginal distribution discrepancy and conditional distribution discrepancy. The optimization problem in Equation (1) is modified as follows,

$$\begin{aligned}\min_A \quad & d_{MMD}(P(\bar{\mathbf{x}}_{(t_1,t)}), P(\bar{\mathbf{x}}_{(t+1,t_2)})) \\ & + d_{MMD}(P(y|\bar{\mathbf{x}}_{(t_1,t)}), P(y|\bar{\mathbf{x}}_{(t+1,t_2)})).\end{aligned} \tag{2}$$

### A. Marginal Distribution Discrepancy

Denote $\mathbf{X} = [\mathbf{x_{t_1}}, \mathbf{x_{t_1+1}}, \ldots, \mathbf{x_{t_2}}]$, We calculate the marginal distribution discrepancy firstly,

$$\begin{aligned} & d_{MMD}(P(\bar{\mathbf{x}}_{(t_1,t)}), P(\bar{\mathbf{x}}_{(t+1,t_2)})) \\ &= ||\frac{1}{n_1}\sum_{i=t_1}^{t} A\mathbf{x}_i - \frac{1}{n_2}\sum_{j=t+1}^{t_2} A\mathbf{x}_j||^2 \\ &= tr\left(AXM_0X^TA^T\right)\end{aligned} \tag{3}$$

where $n_1 = t - t_1 + 1$, $n_2 = t_2 - t$, $M_0$ can be calculated as follows

$$(M_0)_{ij} = \begin{cases} \frac{1}{n_1^2} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{x}_{(t_1,t)}, \\ \frac{1}{n_2^2} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{x}_{(t+1,t_2)}, \\ -\frac{1}{n_1 n_2} & \text{otherwise.} \end{cases} \tag{4}$$

### B. Conditional Distribution Discrepancy

Note that in classification problems, all the labels is discrete. The empirical estimate of MMD calculates the discrepancy of average kernel embedding of each instance. However, the averaging discrete label does not make sense. In JDA, $P(\bar{\mathbf{x}}_{(t_1,t_2)}|y)$ is used to replace $P(y|\bar{\mathbf{x}}_{(t_1,t_2)})$ in Equation (2). We calculate conditional distribution $P(\bar{\mathbf{x}}_{(t_1,t_2)}|y=c)$ for each label in $\{1,2,\ldots,C\}$ and then add all the class-conditional distribution discrepancies up as the total conditional distribution discrepancy,

$$\begin{aligned}
&d_{MMD}(P(\bar{\mathbf{x}}_{(t_1,t)}|y), P(\bar{\mathbf{x}}_{(t,t_2)}|y)) \\
&= \sum_{c=1}^{C} d_{MMD}(P(\bar{\mathbf{x}}_{(t_1,t)}|y=c), P(\bar{\mathbf{x}}_{(t+1,t_2)}|y=c)) \\
&= \sum_{c=1}^{C} ||\frac{1}{n_{1,c}} \sum_{\mathbf{x}_i \in S_{1,c}} A\mathbf{x}_i - \frac{1}{n_{2,c}} \sum_{\mathbf{x}_j \in S_{2,c}} A\mathbf{x}_j||^2 \\
&= \sum_{c=1}^{C} tr\left(AXM_cX^TA^T\right) \\
&= tr\left(AX\sum_{c=1}^{C} M_cX^TA^T\right),
\end{aligned} \tag{5}$$

where $S_{1,c} = \{\mathbf{x}_i : y_i = c \text{ and } t_1 \le i \le t\}$, $S_{2,c} = \{\mathbf{x}_j : y_j = c \text{ and } t+1 \le j \le t_2\}$, and $n_{1,c} = |S_{1,c}|, n_{2,c} = |S_{2,c}|$ are the cardinals of $S_{1,c}, S_{2,c}$ respectively. The elements in MMD matrix $M_c$ for $c \in \{1,2,\ldots,C\}$ can be computed as follows

$$(M_c)_{ij} = \begin{cases} \frac{1}{n_{1,c}^2} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in S_{1,c}, \\ \frac{1}{n_{2,c}^2} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in S_{2,c}, \\ -\frac{1}{n_{1,c}n_{2,c}} & \begin{cases} if\ \mathbf{x}_i \in S_{1,c}, \mathbf{x}_j \in S_{2,c} \\ or\ \mathbf{x}_i \in S_{2,c}, \mathbf{x}_j \in S_{1,c}, \end{cases} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Incorporating Equations (3) and (5), we have the total distribution discrepancy

$$tr\left(AX\sum_{c=0}^{C} M_cX^TA^T\right)$$

and the matrix form of Equation (2) is,

$$\min_A \quad tr\left(AX\sum_{c=0}^{C} M_cX^TA^T\right) + \lambda||A||_{\mathcal{F}}^2 \tag{7}$$

Note that a regularization term $||A||_{\mathcal{F}}^2$ is added in Equation (7), where $\lambda$ controls the impact of the regularization term.

### C. Preserve Data Variance

However, only minimizing the distribution discrepancy is not enough. SEA Moving Hyperplane Concepts (SEA) [13] is a widely used synthetic dataset in concept drift area. Instances in SEA have three features $x_1, x_2$ and $x_3$. The label is determined by a inequality

$$ax_1 + bx_2 \le \theta,$$

where $a, b, \theta$ are parameters. The third feature $x_3$ is a noisy feature and obeys uniform distribution. When concept drift occurs, i.e. some parameters changed, to reduce the discrepancy of distributions, the linear transformation matrix might give the third feature higher weight. An extreme case is that if

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

we have

$$P(\bar{\mathbf{D}}_{(t_1,t)}) = P(\bar{\mathbf{D}}_{(t,t_2)}).$$

Instances before and after drift have the same distribution after transformed, but we lose all the information.

Hence, while minimizing the distribution discrepancy, the properties of data that contain the classification information have to been preserved. Both TCA and JDA adopt the idea of Principal Component Analysis (PCA) to maximize the data variance. Let the centering matrix $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}$, where $\mathbf{I}$ is the identity matrix, $\mathbf{1}$ is matrix of ones, and $n$ is the quantity of samples. Now we have the covariance matrix $AXHX^TA^T$. The aim of PCA is to maximize the variance,

$$\max_A \quad tr\left(AXHX^TA^T\right) \tag{8}$$

We adopt the same optimization aim, to minimize the Equation (7) while preserve Equation (8) as much as possible. Generalized Rayleigh quotient theory shows that minimizing Equation (7) while maximizing Equation (8) is equivalent to the problem that minimizes Equation (7) with Equation (8) fixed. Now we have the final form of the optimization problem,

$$\begin{aligned} \min_A \quad & tr\left(AX\textstyle\sum_{c=0}^{C} M_cX^TA^T\right) + \lambda||A||_{\mathcal{F}}^2 \\ s.t. \quad & AXHX^TA^T = I \end{aligned} \tag{9}$$

### D. Learning Algorithm

In this section, we state the procedure of DAJDA. Denote $\Phi \in \mathbb{R}^{k\times k}$ as the Lagrange multiplier matrix, where

$$\Phi = \begin{bmatrix} \phi_1 & 0 & \cdots & 0 \\ 0 & \phi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \phi_k \end{bmatrix}.$$

Then we have the Lagrange function

$$\begin{aligned} \mathcal{L}(A,\Phi) = & tr\left(A\left(X\sum_{c=0}^{C} M_cX^T + \lambda I\right)A^T\right) \\ & + tr\left(\left(I - AXHX^TA^T\right)\Phi\right). \end{aligned}$$

The optimization problem with a equality constraint in Equation (9) is converted into an unconstrained optimization problem

$$\min_{A,\Phi} \quad tr\left(A\left(X\sum_{c=0}^{C}M_cX^T+\lambda I\right)A^T\right) + tr\left(\left(I-AXHX^TA^T\right)\Phi\right). \tag{10}$$

To find the local minimum of $\mathcal{L}$, let $\frac{\partial \mathcal{L}}{\partial A}=0$, and we have

$$A\left(X\sum_{c=0}^{C}M_cX^T+\lambda I\right)=\Phi AXHX^T. \tag{11}$$

Equation (11) shows that the proper linear transformation matrix $A$ consists of the left generalized eigenvectors of $(X\sum_{c=0}^{C}M_cX^T, XHX^T)$. Thus, we can find the matrix $A$ via applying generalized eigendecomposition to Equation (11).

To handle concept drift adaptation problem in data stream, we adopt a fixed-size window technique. Two fixed-size windows are maintained in our algorithm. A window, denoted as $W_1$ holds historical instances and another window, denoted as $W_2$ holds newly arrived instances. When new instance comes, the instance is added into $W_2$ until $W_2$ is full. Then a linear transformation matrix $A$ is learned by solving the generalized eigendecomposition problem in Equation (11). Both historical instance and new instances are transformed into a latent feature space. The labels of newly coming instances can be predicted via the knowledge learned from historical instance. The procedure of the DAJDA is summarized in Algorithm 1.

**Input :** Training dataset $D_{(0,t)}$, Data stream $\mathbf{X}_{(t+1,\infty)}$, window size $m$, regularization parameter $\lambda$
**Output:** Labels $y_{(t+1,\infty)}$ of data stream $\mathbf{X}_{(t+1,\infty)}$

Train a base learner based on $D_{(0,t)}$ ;
Predict the label $\hat{y}_i$ of the new arrived instance $\mathbf{x}_i$ ;
Initial window $W_1 = D_{(0,t)}$, add the newly arrived instance $\mathbf{x}_i$ in window $W_2$ after the true label $y_i$ is obtained ;
**while** *the size of* $W_2$ *reach* $m$ **do**
    Initial $X$ as the the combine of instances in windows $W_1$ and $W_2$ ;
    Initial the MMD matrix $M_0$ and $\{M_c\}_{c=1}^{C}$ by Equations (4) and (6) ;
    Solve Equation (11) to construct the transformation matrix $A$ ;
    Train a new model based on the transformed data $Z = AX$ ;
    Let $W_1 = W_2$ and $W_2$ be empty ;
    Transform the newly coming instance $\mathbf{z}_j = A\mathbf{x}_j$ and predict the label $\hat{y}_j$by the new model ;
    add the newly arrived instance $\mathbf{x}_j$ in window $W_2$ after the true label $y_j$ is obtained ;
**end**

**Algorithm 1:** Drift Adaptation via Joint Distribution Alignment

## V. Experiments and Evaluation

We conducted several experimental studies to evaluate our method. We firstly verified the effectiveness of DAJDA. Then the performance is evaluated on some widely used real-world datasets.

### A. Verification of the Effectiveness

To verify the effectiveness of DAJDA, we generate some synthetic data stream, and two strategies are conducted as the baseline:

- **Baseline 1**: The model trained from training set is used all the time.
- **Baseline 2**: Retrain the model in every time window.

To investigate the effectiveness of DAJDA, three types of synthetic data streams are gendered. Several types of synthetic data streams have been used in the previous research. Data streams used in this paper is described as follows.

*1) SEA:* SEA was introduced firstly in [13]. Each instance has three features, $x_1, x_2,$ and $x_3$, all of which obey Uniform distribution from 0 to 10. The label is determined by

$$y=\begin{cases}1 & \text{if } ax_1+bx_2\le\theta,\\ 0 & \text{otherwise,}\end{cases}$$

where $a, b, \theta$ are parameters.

*2) ROT:* ROT is introduced in [14] to simulate the concept drift in which the decision boundary is rotated. Each instance has two features, $x_1$ and $x_2$. ROT rotates the instance to simulate the decision boundary rotation. Each instance is rotated a certain angle

$$x=(x-a)\cos\theta-(y-b)\sin\theta+a,$$
$$y=(x-a)\cos\theta+(y-b)\sin\theta+b,$$

where $a, b, \theta$ are parameters.

*3) CIR:* CIR is introduced in [20] to generate the concept drift in which the decision boundary is a sphere in the feature space. Considering the 2-dimensional case, each instance has two features, $x_1$ and $x_2$. The label is given by

$$y=\begin{cases}1 & \text{if } (x_1-a)^2+(x_2-b)^2\le\theta,\\ 0 & \text{otherwise,}\end{cases}$$

where $a, b, \theta$ are parameters.

In our simulation, some parameters are fixed and others are changed to simulate the concept drift. For SEA, we fixed $a = b = 1$ and changed $\theta$ every 1000 instances to simulate the movement of the decision boundary. For ROT, we fixed $a = b = 0$ and changed $\theta$ every 1000 instances to simulate the rotation of the decision boundary. For CIR, we fixed the radius of the ball $\theta$ and changed the center of the ball $(a, b)$ every 1000 instances. The statistics of generated synthetic data stream are shown in Table I. The value of parameters which

TABLE I
SYNTHETIC DATA STREAMS

| Data Stream | #Example | #Feature | #Label |
|---|---|---|---|
| SEA1 | 10000 | 3 | 2 |
| SEA2 | 10000 | 3 | 2 |
| ROT1 | 10000 | 2 | 2 |
| ROT2 | 10000 | 2 | 2 |
| CIR1 | 10000 | 2 | 2 |
| CIR2 | 10000 | 2 | 2 |

TABLE II
VALUES OF DRIFT PARAMETERS

| Stream | Value of Drift Parameters |
|---|---|
| SEA1 | $10 \to 5 \to 13 \to 5 \to 11 \to 12 \to 14 \to 14 \to 8 \to 7$ |
| SEA2 | $10 \to 12 \to 9 \to 14 \to 14 \to 14 \to 12 \to 11 \to 9 \to 9$ |
| ROT1 | $0.78 \to 1.06 \to 1.38 \to 1.49 \to 1.80 \to 1.93 \to 2.14$ $\to 2.13 \to 2.41 \to 2.32$ |
| ROT2 | $0.78 \to 1.04 \to 1.22 \to 1.26 \to 1.18 \to 1.41 \to 1.42$ $\to 1.71 \to 1.78 \to 1.87$ |
| CIR1 | $(0,0) \to (-1.03, 0.01) \to (1.18, 0.49) \to (1.81, -2.08)$ $\to (1.85, -4.21) \to (1.47, -3.78) \to (1.47, -2.20)$ $\to (0.64, -3.08) \to (3.38, -1.14) \to (3.92, -2.01)$ |
| CIR2 | $(0,0) \to (-0.02, 0.63) \to (-0.17, 1.75) \to (2.12, 2.93)$ $\to (3.36, 2.09) \to (3.26, 1.83) \to (3.64, 3.02)$ $\to (3.76, 2.80) \to (5.13, 1.90) \to (6.10, 1.51)$ |

have been changed to simulate concept drift are shown in Table II.

We set the window size $m = 20$ and the regularization parameter $\lambda = 1$. And naive Bayes classifier is chosen as the base model. The experiment results are listed in Table III. The results show that DAJDA performed steadily and obtained highest score for all metrics on all data stream except the precision score on data stream SEA2 and CIR2. It can be concluded that DAJDA has abilities to improve the performance of learning model under concept drift.

### *B. Evaluation on Real-world Dataset*

In this section, we evaluated our proposed method in some real-world datasets. Three real-world datasets are included. The statistics of three datasets are shown in Table IV.

Several state-of-the-art concept drift methods are compared with DAJDA in the experiment: DDM [20], ECDD [21], HDDM [22] and HAT [8]. All the algorithms are implemented in MOA framework [23]. Decision tree is adopted as the base learner. The window size $m$ in DAJDA is set as 1000 and the regularization parameter $\lambda$ is set as 1. The experiment results are listed in Table V and the average rank of performance on all datasets is listed in Table VI. Experiment results show that DAJDA has advantages over other methods on datasets

TABLE III
EXPERIMENT RESULTS OF VERIFICATION OF THE EFFECTIVENESS

| Metrics | Stream | baseline 1 | baseline 2 | DAJDA |
|---|---|---|---|---|
| Accuracy | SEA1 | 0.7356 | 0.8739 | **0.9079** |
| | SEA2 | 0.8133 | 0.8518 | **0.9074** |
| | ROT1 | 0.5643 | 0.8800 | **0.9232** |
| | ROT2 | 0.6674 | 0.8997 | **0.9222** |
| | CIR1 | 0.9078 | 0.9310 | **0.9431** |
| | CIR2 | 0.8990 | 0.9372 | **0.9466** |
| F1 score | SEA1 | 0.7313 | 0.8742 | **0.9082** |
| | SEA2 | 0.7852 | 0.7945 | **0.8746** |
| | ROT1 | 0.5680 | 0.8806 | **0.9236** |
| | ROT2 | 0.6713 | 0.9003 | **0.9225** |
| | CIR1 | 0.9512 | 0.9632 | **0.9699** |
| | CIR2 | 0.9462 | 0.9672 | **0.9719** |
| Precision | SEA1 | 0.7187 | 0.8689 | **0.9037** |
| | SEA2 | **0.8974** | 0.7539 | 0.8495 |
| | ROT1 | 0.5688 | 0.8791 | **0.9221** |
| | ROT2 | 0.6774 | 0.9038 | **0.9233** |
| | CIR1 | 0.9767 | 0.9966 | **0.9971** |
| | CIR2 | 0.9598 | **0.9983** | 0.9969 |
| Recall | SEA1 | 0.7443 | 0.8796 | **0.9128** |
| | SEA2 | 0.6979 | 0.8398 | **0.9014** |
| | ROT1 | 0.5672 | 0.8822 | **0.9252** |
| | ROT2 | 0.6652 | 0.8968 | **0.9217** |
| | CIR1 | 0.9269 | 0.9329 | **0.9442** |
| | CIR2 | 0.9331 | 0.9379 | **0.9481** |

TABLE IV
REAL-WORLD DATA STREAM

| Data Stream | #Example | #Feature | #Label |
|---|---|---|---|
| Covertype | 581012 | 54 | 7 |
| Electricity | 45312 | 8 | 2 |
| Weather | 18159 | 8 | 2 |

Covertype and Weather. HAT obtained the highest scores on dataset Electricity. However, Table VI shows that DAJDA has the best performance considering all three datasets and all metrics. We can conclude that DAJDA improve the precision significantly.

## VI. CONCLUSION

In this paper, a concept drift adaptation method, Drift Adaptation via Joint Distribution Alignment (DAJDA), is proposed. The method transforms the data into common feature space, reducing the discrepancy of distributions. Thus the proposed method is model-free. Experimental studies show that DAJDA

TABLE V
EXPERIMENT RESULTS OF EVALUATION ON REAL-WORLD DATASET

| Dataset | Method | Accuracy | F1 score | Precision | Recall |
|---|---|---|---|---|---|
| Covertype | DAJDA | **0.8457** | 0.6286 | **0.6424** | **0.6181** |
| | DDM | 0.5974 | 0.4473 | 0.5106 | 0.3983 |
| | ECDD | 0.5321 | 0.3811 | 0.4305 | 0.3427 |
| | HDDM-A | 0.6068 | 0.4574 | 0.5290 | 0.4030 |
| | HDDM-W | 0.6403 | 0.4580 | 0.5251 | 0.4062 |
| | HAT | 0.7887 | **0.6381** | 0.6341 | 0.6040 |
| Electricity | DAJDA | 0.6860 | 0.6790 | 0.6786 | 0.6793 |
| | DDM | 0.6727 | 0.6837 | 0.6822 | 0.6082 |
| | ECDD | 0.4784 | 0.5105 | 0.5069 | 0.5014 |
| | HDDM-A | 0.6228 | 0.6401 | 0.6407 | 0.6036 |
| | HDDM-W | 0.6600 | 0.6624 | 0.6684 | 0.6051 |
| | HAT | **0.7557** | **0.7502** | **0.7484** | **0.7051** |
| Weather | DAJDA | 0.7255 | **0.6826** | 0.6816 | **0.6836** |
| | DDM | **0.7435** | 0.6710 | **0.6900** | 0.6054 |
| | ECDD | 0.6434 | 0.5936 | 0.5908 | 0.5094 |
| | HDDM-A | 0.6846 | 0.6540 | 0.6448 | 0.6064 |
| | HDDM-W | 0.7033 | 0.6431 | 0.6445 | 0.6049 |
| | HAT | 0.7160 | 0.6665 | 0.6654 | 0.6068 |

TABLE VI
AVERAGE RANK ON ALL THREE DATASETS

| Method | Accuracy | F1 score | Precision | Recall |
|---|---|---|---|---|
| DAJDA | 1.67 | 2 | 2 | 1.33 |
| DDM | 2.67 | 3 | 2.67 | 4 |
| ECDD | 6 | 6 | 6 | 6 |
| HDDM-A | 4.67 | 4.33 | 3.33 | 5 |
| HDDM-W | 3.67 | 4 | 4 | 5 |
| HAT | 2 | 1.67 | 2 | 2.67 |

has abilities to improve the performance of learning model in evolving environment.

The main drawback of DAJDA is that a generalized eigen-decomposition problem need to be solved, so that the time complexity might be costly for an online learning case. In addition, DAJDA could only performs linear transformation on the instances. Some non-linearity is needed. This would be a problem remaining to be solved in the future.

## ACKNOWLEDGEMENTS

This work was supported by the Australian Research Council (ARC) under discovery grant DP190101733.

## REFERENCES

[1] G. Widmer and M. Kubat, "Learning in the Presence of Concept Drift and Hidden Contexts." *Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996. [Online]. Available: https://doi.org/10.1007/BF00116900

[2] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation." *ACM Comput. Surv.*, vol. 46, no. 4, pp. 44:1–44:37, 2014. [Online]. Available: https://doi.org/10.1145/2523813

[3] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer Feature Learning with Joint Distribution Adaptation." in *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, 2013, pp. 2200–2207. [Online]. Available: https://doi.org/10.1109/ICCV.2013.274

[4] S. J. Pan and Q. Yang, "A Survey on Transfer Learning." *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010. [Online]. Available: https://doi.org/10.1109/TKDE.2009.191

[5] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A Kernel Method for the Two-Sample-Problem." in *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, 2006, pp. 513–520. [Online]. Available: http://papers.nips.cc/paper/3110-a-kernel-method-for-the-two-sample-problem

[6] G. Hulten, L. Spencer, and P. M. Domingos, "Mining time-changing data streams." in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*, 2001, pp. 97–106. [Online]. Available: http://portal.acm.org/citation.cfm?id=502512.502529

[7] P. M. Domingos and G. Hulten, "Mining high-speed data streams." in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000*, 2000, pp. 71–80. [Online]. Available: https://doi.org/10.1145/347090.347107

[8] A. Bifet and R. Gavaldà, "Adaptive Learning from Evolving Data Streams." in *Advances in Intelligent Data Analysis VIII, 8th International Symposium on Intelligent Data Analysis, IDA 2009, Lyon, France, August 31 - September 2, 2009. Proceedings*, 2009, pp. 249–260. [Online]. Available: https://doi.org/10.1007/978-3-642-03915-7_22

[9] E. Ikonomovska, J. Gama, and S. Dzeroski, "Learning model trees from evolving data streams." *Data Min. Knowl. Discov.*, vol. 23, no. 1, pp. 128–168, 2011. [Online]. Available: https://doi.org/10.1007/s10618-010-0201-y

[10] A. Bifet, B. Pfahringer, J. Read, and G. Holmes, "Efficient data stream classification via probabilistic adaptive windows." in *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18-22, 2013*, 2013, pp. 801–806. [Online]. Available: https://doi.org/10.1145/2480362.2480516

[11] V. Losing, B. Hammer, and H. Wersing, "KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift." in *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, 2016, pp. 291–300. [Online]. Available: https://doi.org/10.1109/ICDM.2016.0040

[12] A. Liu, J. Lu, F. Liu, and G. Zhang, "Accumulating regional density dissimilarity for concept drift detection in data streams." *Pattern Recognition*, vol. 76, pp. 256–272, 2018. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.11.009

[13] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification." in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*, 2001, pp. 377–382. [Online]. Available: http://portal.acm.org/citation.cfm?id=502512.502568

[14] D. Brzezinski and J. Stefanowski, "Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm." *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 1, pp. 81–94, 2014. [Online]. Available: https://doi.org/10.1109/TNNLS.2013.2251352

[15] R. Elwell and R. Polikar, "Incremental Learning of Concept Drift in Nonstationary Environments." *IEEE Trans. Neural Networks*, vol. 22, no. 10, pp. 1517–1531, 2011. [Online]. Available: https://doi.org/10.1109/TNN.2011.2160459

[16] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach." in *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, 2011, pp. 999–1006. [Online]. Available: https://doi.org/10.1109/ICCV.2011.6126344

[17] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation." in *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI,*

*USA, June 16-21, 2012*, 2012, pp. 2066–2073. [Online]. Available: https://doi.org/10.1109/CVPR.2012.6247911

[18] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain Adaptation via Transfer Component Analysis." *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011. [Online]. Available: https://doi.org/10.1109/TNN.2010.2091281

[19] S. Si, D. Tao, and B. Geng, "Bregman Divergence-Based Regularization for Transfer Subspace Learning." *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, 2010. [Online]. Available: https://doi.org/10.1109/TKDE.2009.126

[20] J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues, "Learning with Drift Detection." in *Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence, São Luis, Maranhão, Brazil, September 29 - October 1, 2004, Proceedings*, 2004, pp. 286–295. [Online]. Available: https://doi.org/10.1007/978-3-540-28645-5_29

[21] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, "Exponentially weighted moving average charts for detecting concept drift." *Pattern Recognition Letters*, vol. 33, no. 2, pp. 191–198, 2012. [Online]. Available: https://doi.org/10.1016/j.patrec.2011.08.019

[22] I. I. F. Blanco, J. d. Campo-Ávila, G. Ramos-Jiménez, R. M. Bueno, A. A. O. Díaz, and Y. C. Mota, "Online and Non-Parametric Drift Detection Methods Based on Hoeffding's Bounds." *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 810–823, 2015. [Online]. Available: https://doi.org/10.1109/TKDE.2014.2345382

[23] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive Online Analysis." *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?id=1859903