Fast Concept Drift Detection Using Singular Vector Decomposition

Dan Shang, Guangquan Zhang, Jie Lu
Centre for Artificial Intelligence
Faculty of Engineering and Information Technology
University of Technology Sydney
Email: dan.shang@student.uts.edu.au, guangquan.zhang@uts.edu.au, jie.lu@uts.edu.au

***Abstract*—Data stream mining is widely used in online applications such as sensor networks, financial transactions, etc. Such systems generate data at high velocity and their underlying distributions may change over time. This is referred to as concept drift problem and it is considered to be the root cause of performance degradation of online machine learning models. To tackle this problem, a reliable and fast drift detection method is required to achieve real time responsiveness to the drifts. This paper presents a fast and accurate drift detection method, namely *KS-SVD test — KSSVD*, to monitor the distribution changes of the data stream. Our method employs the SVD technique to first check the direction change of the data, followed by a KS test on each direction to detect the univariate distribution changes. Experiments show that our method is efficient and accurate, especially in high dimension situation.**

***Keywords*-concept drift; KS test; SVD; data stream**

I. Introduction

Data stream mining has become an important research field in machine learning. Data stream mining is widely used in online applications such as sensor networks, financial transactions, telecommunication, mobile applications, etc. Such systems generate data constantly and often at a high velocity, for example, tweets, transaction flows and network activities. Among others, one prominent characteristic of stream data is that their underlying distribution may change arbitrarily over time [1], [2]. This is considered to be an important root cause of performance degradation of online data mining systems as the machine learning prediction models, based on the distribution of previous training data, are no longer fit for the distribution of newly arrived data. This problem is known as the concept drift problem.

Formally, given an infinite sequence of stream data (x, y, t), with input data x, output class y, and time stamp t, concept drift is defined as such that the joint probability $p(x, y)$ of the stream data distribution, with density function ϕ, have changed at time T [3]. From a Bayesian point of view, the change of joint probability $p(x, y)$ may have three underlying sources: the change of prior density $p(x)$, the change of prior probability $p(y)$ and the change of posterior probability $p(y|x)$ [4].

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{1}$$

However, detecting changes in prior probability $p(y)$ and posterior probability $p(y|x)$ relies on immediate availability of data with true labels (manually classified by users), which is a demanding prerequisite and may not be feasible in most real world scenarios. So in this paper, we only consider the case of prior density $p(x)$ changes.

Further, concept drifts can be categorized into different types according to the extent of the drifts: sudden or abrupt drift; gradual or slow drift; recurrent drift [5]. Our method aims to handle the first two types of drifts by leveraging a specific window strategy [6].

Existing concept drift detection methods can be loosely grouped into three strategies. Early methods monitor distribution changes by calculating simple statistics from the data, such as cumulative sum in [7] or classifier error rate [8]. However, the detection accuracy of these methods are often limited because the chosen statistics are not always able to reflect all possible distribution changes. Other researches resolved to existing multivariate two sample tests or developed new ones to detect differences between the distributions of existing data and newly arrived data [9]. Although these tests are statistical guaranteed to be able to detect distribution difference, computing the statistics on high dimensional data can be costly. In data mining community, researches also use machine learning algorithms to develop methods as alternatives to two sample tests, such as competence model method in case base reasoning [10] and k-dimensional tree based methods [11]. These methods aim to achieve a balance between statistical rigor and computational cost, but are still not able to meet the requirements of real time online systems.

Motivated by these issues, this work aims to develop a fast concept detect method with statistically guaranteed accuracy as well as low computational complexity to achieve real time responsiveness, by combining the commonly used Kolmogorov-Smirnov(KS) test [12] and Singular Vector Decomposition(SVD) technique, which, as we will show in later chapters, will compensate each others weakness and result in an effect and distribution-free concept drift detection algorithm.

Our main contributions are:

- Analysis of the effectiveness of combining KS test and SVD as a multivariate two sample test.
- A fast and distribution-free concept drift detection method

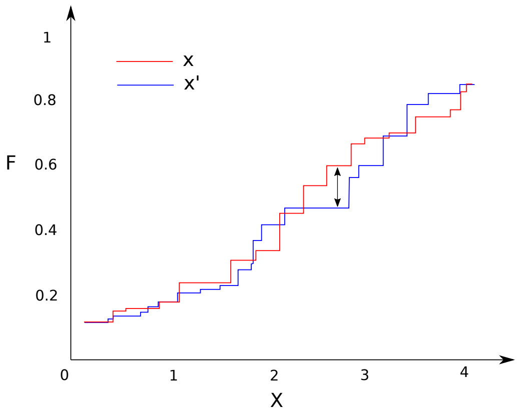

Fig. 1. KolmogoovSmirnov statistic D for two sample test.

with statistically guaranteed accuracy.

This rest of this paper is organised as follows. Section II first introduces KS test, SVD and their characteristics, then analyze how they can compensate each other as a two sample test method. Section III presents the concept drift detection method based on the combination of KS and SVD model, integrated with the a windowing scheme. Section IV presents our experiment results, with an analysis for both the effectiveness of our model and the performance of our concept drift detection method. Finally, Section V concludes our study and presents the future work.

II. Combining KS test and SVD

In this section, we will first review the commonly used KS test and SVD technique and their characteristics. Then we will demonstrate how combining the two can be used as a two sample test method for multidimensional data.

A. *Kolmogorov-Smirnov test*

Kolmogorov-Smirnov(KS) is a well known statistical test that can be used as one sample test or two sample test. Here we are only interested in the two sample test case which is to test whether two sets of data have same distribution [13]. Given two samples $X = x_0, x_1, \ldots, x_{n-1}$ and $X' = x'_0, x'_1, \ldots, x'_{m-1}$, $x, x' \in R$, with size n and m, and empirical distribution function $F(x)$ and $F'(x)$ respectively, the KolmogorovSmirnov statistic is

$$D_{n,m} = \sup |F_n(x) - F'_m(x)| \tag{2}$$

The null hypothesis, that the two samples have the same distribution, is rejected at level α (usually 0.05) when $D_{n,m} > c(\alpha)\sqrt{\frac{n+m}{nm}}$, where $c(\alpha) = \sqrt{-\frac{1}{2}\ln(\frac{\alpha}{2})}$, as shown in Figure 1.

Kolmogorov-Smirnov test is initially designed for one dimensional data. Since it relies on the ordering of the data points, it cannot be easily generalized to multidimensional data. Although several extensions of the Kolmogorov-Smirnov test to multivariate data have been introduced [14], [15], their computational complexity are often very high because they need to traverse the data in all possible orders of every dimension.

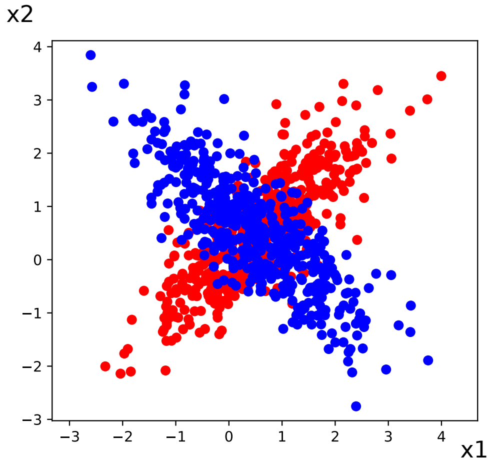

Fig. 2. Two data sets with same distribution on each dimension but different correlation.

A simple way to extend Kolmogorov-Smirnov test is to separately apply the test on each dimension. This strategy is computationally efficient. However, an obvious problem is that this approach cannot detect the correlation changes of the data when distribution of each dimension remains the same. To see this problem, we now take an example in two dimensional space for convenience. The following analysis still holds in the higher dimensional situations. As shown in Figure 2, the joint distribution of the red colored data set is a 2D normal distribution $N(0.5, 0.5, 1, 1, 0.8)$. The blue colored data set follows a similar 2D normal distribution $N(0.5, 0.5, 1, 1, -0.8)$. The two data sets are only different that the correlation coefficients between of x_1 and x_2 are opposite. However, the two data sets have same marginal distribution, which are both $x_1 \sim N(0.5, 1)$ and $x_2 \sim N(0.5, 1)$. In such case, if we apply KS test directly on each coordinate (x_1 or x_2), the null hypothesis will not be rejected. The coordinates x_1, x_2 can be seen as the projection on the standard basis vectors of $(1, 0)$, $(0, 1)$. But these standard basis vectors cannot present the direction information of the two data sets. So we use SVD of the covariance matrix to find the eigenvectors in order to discover this direction information. We will see, in the next subsection, how this issue can be handled by SVD technique.

B. *Singular Vector Decomposition*

In scenarios with high dimensional data streams, it is still difficult to model the multidimensional distribution accurately, and the computational cost is usually high. As we have already discussed in the last section, correlation information will be lost when analyzing each dimension separately. The covariance matrix is commonly used to analyze the correlation of each feature of the data. In order to compensate the univariate KS test, we include the SVD of the covariance matrix in our

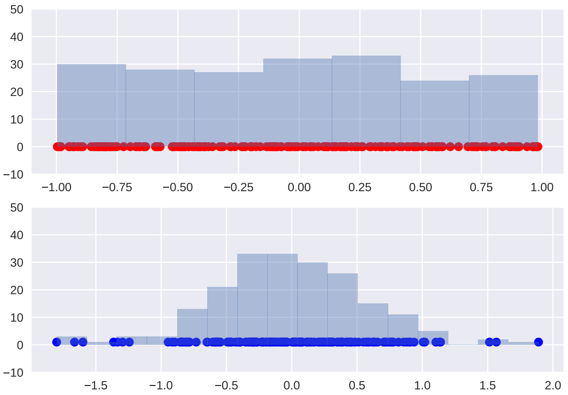

Fig. 3. Two data sets with same mean and variance but different distributions.

method. Suppose Σ_0 and Σ_1 are the covariance matrices of the data sets D_0 and D_1.The SVD of Σ is

$$\Sigma = V \Lambda V^T \tag{3}$$

The columns $v_0^{(i)}$ and $v_1^{(i)}$ are eigenvectors. The elements λ_0^i and λ_1^i of the principal diagonals of Λ_0 and Λ_1 are the eigenvalues. It can be seen as the variance of the data on the corresponding eigenvector direction. Compared with basic vectors, these eigenvectors are generated from the data sets. Also, because eigenvector of the largest eigenvalue is the direction that the data has the largest variance, eigenvectors can be used to represent direction information of the data. We could calculate the angle $\theta^{(i)}$ between the eigenvectors $v_0^{(i)}$ and $v_1^{(i)}$. If the directions are significantly different ($\theta^{(i)} \neq 0$ or π), then the data distributions must be different.

However, the opposite does not hold, which means that if the directions are same, we cannot come to the conclusion that two data sets have the same distribution. For example, in Figure 3 there are two data sets, both having the same size of 200. The red colored data set D_0 is generated from one dimensional uniform distribution within the range of $(-1, 1)$. The mean of D_0 is zero. We could calculate its variance as $\sigma_1^2 = (1 - (-1)^2/12 = 1/3$. The blue colored data set D_1 has the same mean and variance as D_0. But D_1 is generated from a normal distribution $N(0, 1/3)$. Clearly, their distributions are different. The SVD technique cannot detect this kind of difference. In such case, a mechanism to differentiate the distributions is required. We choose the commonly used univariate two sample test KS test. However, since KS test is only suitable for one dimensional data, we need to project the high dimensional data on each eigenvectors separately, and treat each projection as a one dimensional data set. Then we can perform the KS test on each projection respectively.

III. Change detection method

From the previous section, we see that KS test can be used to detect single dimension differences while SVD can be used to test changes of correlation between dimensions. In this section, we will explain in detail how to combine the two test as a general multivariate two sample test and use it as a novel drift detection method.

A. *Combine KS and SVD as two sample test*

Given two input data samples $X = x_1, x_2, \ldots, x_n$, $X' = x'_1, x'_2, \ldots, x'_n$, $X, X' \in R^d$, with equal size n, First, we compute the covariance matrices of X, X', denoted as Σ, Σ'. After applying SVD to the covariance matrices we get eigenvalues of each sample, denoted as $\lambda_1, \ldots, \lambda_d$, $\lambda'_1, \ldots, \lambda'_d$ and their eigenvectors, denoted as $v_1, \ldots, v_d$, $v'_1, \ldots, v'_d$, respectively. It should be noticed that the eigenvectors of each sample is sorted in descending order according to the eigenvalues, which represents the variances of the data on the eigenvector direction. Next, for each pair of v_i, v'_i we compute the angle of the two vectors, denoted as θ_i. If θ_i exceeds a predefined threshold β, a drift alert is triggered. If $\theta_i \leq \beta$, we then need to perform KS test on the two samples after being projected on the direction defined by the current eigenvector v_i. The p-value of the KS test is compared with predefined threshold α, if $p < \alpha$, a drift alert is triggered. If all the v_i are checked without drift alert, then the two samples are considered to have the same distribution. We denote this process as *KSSVD* test.

B. *Window strategy*

Our approach is to compare the data distribution of two windows. First, the reference window W_0 is set to be the first n samples $x_1, x_2 \ldots, x_n$. The test window W_1 is the next n samples $x_{n+1}, \ldots, x_{2n}$. Then, we perform the *KSSVD* test on the windows W_0 and W_1. If no drift is detected. W_0 stay at the same location while W_1 slide forward. If a drift is detected, the reference window W_0 will be updated to be the following n samples after the drift point x_t. Hence, the window W_0 can present the current distribution. This *fix-sliding windows* method [12] is suitable for abrupt and gradual drifts. The window size n can be adjusted according to the extent of drifts to be detected.

C. *Pseudo-code*

Algorithm 1 lists the pseudo-code of our method.

IV. Empirical Evaluation

In this section, we evaluate the effectiveness of the proposed KSSVD method. The data we are using is generated under the similar process described in [11]. We also compare our method with the method introduced in [11], which we refer to as KL for KL-divergence. We choose this method as comparison because it is a popular method and aims to achieve the same goals as our method — fast, distribution-free, high dimension robust.

A. *Accuracy*

In the following experiments, we show the accuracy of our method. The method uses *fix-sliding* window strategy explained in Section III-B, that is, keeping the reference window fixed while moving the sliding window as new data

Algorithm 1 KSSVD algorithm

```
$\alpha \leftarrow$ KS test threshold
$\beta \leftarrow$ SVD angle threshold
$n \leftarrow$ window size
$w_0, w_1 \leftarrow$ initial window
$v_0 \leftarrow \mathrm{SVD}(w_0)$
for all $v_0^{(i)}$ in $v_0$ do                ▷ project $w_0$ to eigenvectors
    $p_0^{(i)} \leftarrow$ PROJECTION($w_0, v_0^{(i)}$)
end for
while $x \leftarrow$ newly arrived data do
    SLIDE($w_1, x$)                              ▷ slide test window only
    $v_1 \leftarrow \mathrm{SVD}(w_1)$
    if ANGLE($v_0, v_1$) $> \beta$ then
        ALERT(True)                              ▷ drift alert
        UPDATE($w_0, w_1, n$)                    ▷ update fixed window
    end if
    for all $v_0^{(i)}$ in $v_0$ do             ▷ project $w_1$ to eigenvectors
        $p_1^{(i)} \leftarrow$ PROJECTION($w_1, v_0^{(i)}$)
        $s \leftarrow$ KS($p_0^{(i)}, p_1^{(i)}$)      ▷ KS test on each direction
        if $s < \alpha$ then
            ALERT(True)
            UPDATE($w_0, w_1, n$)
        end if
    end for
end while
```

TABLE I
PARAMETERS OF COMPARISON METHODS USED IN THE EXPERIMENTS.

Method	Parameter	Symbol	Value
KSSVD	SVD angle threshold	α	0.05π
	KS test threshold	β	0.05
KL	Minimum side length of a cell	δ	2^{-10}
	Maximum number of points in a cell	τ	100

arrives. The reference window is only updated after a drift is detected.

We measure the accuracy with the following four rules. If the window is moving across the drift point and the change is detected, then the result is called detected. If result is a change being detected and the window is not moving over a change point, we call this situation is false. If the window has passed a drift point and has not arrived the next change point, we call this situation is late. If the change is happening and the algorithm give a stationary result, we call this situation is missing.

The parameters used in the experiments are listed in Table I

1) Different data distributions: For different types of drift, We choose four data sets. The first three are generated from the 2 dimensional normal distribution. The first group M fix the standard deviation $\sigma_1 = \sigma_2 = 0.2$, and the correlation $\rho = 0.5$. The mean begin with $\mu_1 = \mu_2 = 0.5$, and vary within $[0.2, 0.8]$ randomly. The step size is 0.1 and 0.05. The second group D fix the mean as $\mu_1 = \mu_2 = 0.5$ and the correlation $\rho = 0.5$, but the standard deviation change within $[0, 0.4]$ randomly. The step size is 0.06 and 0.04. The third group C fix the mean $\mu_1 = \mu_2 = 0.5$, and $\sigma_1 = \sigma_2 = 0.2$. We change the correlation within $[-1, 1]$ and the step size is 0.6 and 0.4. The last data set P is generated using the method described in [16]. The data follows multivariate Poisson distribution $(X, Y) \sim Poisson(500(1-\rho), 500(1-\rho), 500\rho)$. Drift is introduced as such that ρ starts at 0.5 and then performs a random walk between 0 and 1 with step size $\Delta = 0.6, 0.4$.

TABLE II
DRIFT DETECTION RESULT ON DIFFERENT DATA TYPES WITH WINDOW SIZE 1000.

Data	Method	Detected	Late	False	Missed
M(0.1)	KSSVD	99	0	6	0
	KL	85	10	5	4
M(0.05)	KSSVD	86	11	3	2
	KL	24	24	2	51
D(0.06)	KSSVD	81	12	8	6
	KL	80	8	7	11
D(0.04)	KSSVD	76	10	9	13
	KL	72	12	1	15
C(0.6)	KSSVD	91	3	11	5
	KL	95	3	2	1
C(0.4)	KSSVD	79	14	10	6
	KL	69	20	6	10
P(0.6)	KSSVD	93	5	10	1
	KL	92	5	4	2
P(0.4)	KSSVD	82	5	3	12
	KL	78	13	3	8

The experiment result is shown in Table II. It shows that our method outperforms KL in most types of distributions. We notice that when drift is small, our method still maintains relatively high accuracy, which means that our method is more sensitive to small drifts.

2) Different window sizes: In this experiment, we reduce the window size and test if the performance of our method is stable under different window sizes. The test data sets and parameters used are all same as previous experiment, except the window size is reduce to 500. The result is shown in Table III. We can see that, as expected, as window size reduces, the methods' accuracy both decrease. However, our method still outperforms KL in most categories. This means that our method is robust to small window size and the performance of our method is stable.

3) Higher dimensions: To test the effectiveness of our method in higher dimension scenarios, we increase the dimension of one of the previously used data set and perform similar tests again. The data set $D(i)$ $(i = 4, 6, 10, 15, 20$ presents the dimension of the data) we will use is the normal data stream C(0.6). Additional dimensions are added to the data set, while the first two dimensions, where the drift happens, remain same. The result is shown in Table IV. We can see that our method still outperforms KL in most of categories. Moreover, the accuracy of KL drops greatly as dimension increases, but the accuracy of KSSVD remains relatively stable. This means that our method is more robust for high dimensional data.

B. Efficiency

In this last experiment, we measure the efficiency of our method. We compare our method to KL on data sets with

TABLE III

Drift detection result on different data types with window size 500.

Data	Method	Detected	Late	False	Missed
M(0.1)	KSSVD	97	2	6	0
	KL	61	24	3	14
M(0.05)	KSSVD	55	24	3	20
	KL	12	26	1	61
D(0.06)	KSSVD	51	13	8	35
	KL	42	15	2	42
D(0.04)	KSSVD	49	19	6	31
	KL	49	12	3	38
C(0.6)	KSSVD	81	10	7	8
	KL	78	15	0	6
C(0.4)	KSSVD	58	13	7	28
	KL	56	16	2	27
P(0.6)	KSSVD	75	15	9	9
	KL	83	14	4	2
P(0.4)	KSSVD	66	11	9	22
	KL	64	13	6	22

TABLE IV

Drift detection result on high-dimensional data with window size 1000.

Data	Method	Detected	Late	False	Missed
D(4)	KSSVD	72	17	10	10
	KL	85	3	7	11
D(6)	KSSVD	65	10	8	24
	KL	75	13	7	11
D(10)	KSSVD	62	13	10	24
	KL	39	18	2	42
D(15)	KSSVD	52	16	7	31
	KL	47	22	2	30
D(20)	KSSVD	35	22	5	42
	KL	25	23	1	51

TABLE V

Running time with different dimensions and window sizes.

Dimension(d)	Window size(n)	Method	Construct(s)
2	1000	KSSVD	0.0194
		KL	0.597
2	2000	KSSVD	0.0758
		KL	1.134
4	2000	KSSVD	0.1137
		KL	1.1644
4	5000	KSSVD	0.489
		KL	3.3046

different window sizes and dimensions. The experiment environment is a PC with one 3.6GHz Intel i7 processor and 16GB memory. The test program is written in Python 2.7. Each test is conducted 100 times and the average time is measured. The result is shown in Table V. We can get a nearly ten times speed up compared with the KL method.

V. Conclusions and further studies

In this work, we proposed a fast concept drift detection method based on the combination of KS test and SVD technique, which compensate each other's weaknesses as a multivariate two sample test. Experiments have shown the efficiency of our method, especially in high dimensional situations. Our method does not require permutation or bootstrap procedure, thus is able to achieve up to ten times speed compared to other distribution tests that relies on them.

As for future studies, we aim to develop an machine learning model adaptation method which utilizes the output of our detection method. Another direction is developing a nonparametric framework, in which the threshold parameters can be automatically trained from the data, so that our method could have larger application scale and higher significance.

References

[1] J. C. Schlimmer and R. H. Granger Jr, "Incremental learning from noisy data," *Machine Learning*, vol. 1, no. 3, pp. 317–354, 1986.

[2] 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.

[3] J. Gama, R. Fernandes, and R. Rocha, "Decision trees for mining data streams," *Intelligent Data Analysis*, vol. 10, no. 1, pp. 23–45, 2006.

[4] C. Alippi, G. Boracchi, and M. Roveri, "Hierarchical change-detection tests," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2016.

[5] K. O. Stanley, "Learning concept drift with a committee of decision trees," *Informe técnico: UT-AI-TR-03-302, Department of Computer Sciences, University of Texas at Austin, USA*, 2003.

[6] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *Proceedings of the Thirtieth International Conference on Very Large Data Bases*. VLDB Endowment, 2004, pp. 180–191.

[7] E. Page, "Continuous inspection schemes," *Biometrika*, pp. 100–115, 1954.

[8] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence–SBIA 2004*. Springer, 2004, pp. 286–295.

[9] P. R. Rosenbaum, "An exact distribution-free test comparing two multivariate distributions based on adjacency," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 67, pp. 515–530, 2005.

[10] N. Lu, G. Zhang, and J. Lu, "Concept drift detection via competence models," *Artificial Intelligence*, vol. 209, pp. 11–28, 2014.

[11] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi, "An information-theoretic approach to detecting changes in multi-dimensional data streams," in *Proceedings of the Symposium on the Interface of Statistics, Computing Science, and Applications*. Citeseer, 2006.

[12] A. Kolmogoroff, "Confidence limits for an unknown distribution function," *The Annals of Mathematical Statistics*, vol. 12, no. 4, pp. 461–463, 1941.

[13] R. H. C. Lopes, *Kolmogorov-Smirnov Test*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 718–720.

[14] J. Peacock, "Two-dimensional goodness-of-fit testing in astronomy," *Monthly Notices of the Royal Astronomical Society*, vol. 202, no. 3, pp. 615–627, 1983.

[15] G. Fasano and A. Franceschini, "A multidimensional version of the kolmogorov–smirnov test," *Monthly Notices of the Royal Astronomical Society*, vol. 225, no. 1, pp. 155–170, 1987.

[16] K. V. Mardia, *Families of Bivariate Distributions*. London: Griffin, 1970.