2019 Global Conference for Advancement in Technology (GCAT)
Bangalore, India. Oct 18-20, 2019

Covariate Shift: A Review and Analysis on Classifiers

[1]Geeta Dharani.Y
Department of Computer Science and Information Systems
BITS-PILANI Hyderabad campus, Telangana, INDIA

[2]Nimisha G Nair
Department of Computer Science and Information Systems
BITS-PILANI Hyderabad campus, Telangana, INDIA

[3]Pallavi Satpathy
Department of Computer Science and Information Systems
BITS-PILANI Hyderabad campus, Telangana, INDIA

[4]Jabez Christopher
Department of Computer Science and Information Systems
BITS-PILANI Hyderabad campus, Telangana, INDIA
[4]jabezc@hyderabad.bits-pilani.ac.in

***Abstract* — Training and testing are the two phases of a supervised machine learning model. When these models are trained, validated and tested, it is usually assumed that the test and train data points follow the same distribution. However, practical scenarios are much different; in real world, the joint distribution of inputs to the model and outputs of the model differs between training and test data, which is called dataset shift. A simpler case of dataset shift, where only the input distribution changes and the conditional distribution of the output for a given input remains unchanged is known as covariate shift. This article primarily provides an overview of the existing methods of covariate shift detection and adaption and their applications in the real world. It also gives an experimental analysis of the effect of various covariate shift adaptation techniques on the performance of classification algorithms for four datasets which include, synthetic and real-world data. Performance of machine learning models show significant improvement after covariate shift was handled using Importance Reweighting method and feature-dropping method. The review, experimental analysis, and observations of this work may serve as guidelines for researchers and developers of machine learning systems, to handle covariate shift problems efficiently.**

Keywords—Covariate shift, Machine Learning, Classification algorithms, Dataset shift, Data Mining

I. Introduction

Machine learning deals with solutions of how to build computer-based systems and computing methods that automatically adapt and improve through experience. It is an emerging field of research and application; it can be considered to be at the intersection of computing science and statistics, and at the core of artificial intelligence. Machine Learning approaches and data mining techniques have provided successful results in many knowledge-engineering tasks such as classification, regression, and clustering. Though many forms of learning exist, the field of Machine learning has two well-known forms: supervised and unsupervised learning. Predictive modeling task and its corresponding solving approaches constitute one of the dimensions of machine learning namely, Supervised learning. It is used (or strategically implemented in the context of information technology) when variable values (independent variables) are used to predict another target variable (dependent variable) with known values. Unsupervised learning methods are more frequently deployed on data for which there is no target with known values. The steps involved in supervised learning includes collection of data from true sources, integration of the data, transformation of the data, training and validating the algorithm on the known data and finally applying it to the unknown testing data. However, for the better performance of these supervised learning algorithms, data quality plays an important role. The quality of data can be analyzed based on various perspectives like data complexity, missing values, noise, imbalanced data, outliers, scaled values etc. Another such data quality measure which plays a huge role in determining the performance of a machine learning model is dataset shift; it is a blanket term for three kinds of shifts mentioned below:

- Covariate Shift: Change in the independent variables;
- Prior probability shift: Change in the target variable;
- Concept Shift: Change in the relationship between the independent and the target variable;

Commonly used machine learning models work well under an assumption that the points, or instances, present in the test and train data, belong to the same feature space and the same distribution. But, when there is a change in distribution, the underlying statistical models need to be rebuilt from scratch using new training data. In practice it is very expensive and sometimes moreover difficult to integrate the training data again and rebuild the models. This usually occurs in scenarios where we consider assessing the risk of future events. For example, if we wish to predict the probability of a person becoming a victim of a chronic respiratory disease in the next five years (y) based on the current smoking trends: given the past trends (x). In such situations there is a high probability, or chance, that the data will change if some environmental conditions change. Consider that a government-imposed public smoking ban came into effect; such an incident will affect the distribution over habits of x. In this case it cannot be assumed always that the prediction of the risk for a new person, a test instance, with different habits x' would be correct; such problems can be modelled as covariate shift problems.

This article is organized in the following way. Section II lists the common definitions and notations used in the field of covariate shift. Sections III and IV explain the various algorithms used for covariate shift detection, adaption and their real world applications respectively. A detailed analysis of performance for various classification methods performed over synthetic and real world datasets after handling covariate shift is presented in sections V and VI.

II. Mathematical Background and Notations

The term covariate shift describes the as change in the distribution of input variable 'X' between the learning and

978-1-7281-3694-3/19/$31.00 ©2019 IEEE

generalization phases, commonly known as training and testing phases. Though covariate shift is the most studied type of shift, there exists some confusion in the exact definition. From the machine learning point-of-view, this predictive modeling setting is commonly known as transfer learning. There are also some similar names, but with minor conceptual differences, such as population drift, concept drift, dataset shift. Below is a list of definitions of covariate shift present in the literature:

- Let x be the explanatory variable or the covariate. Let $q_1(x)$ be the density of x for evaluation of the predictive performance, while $q_0(x)$ be the density of x in the observed data. The situation $q_0(x) \neq q_1(x)$ will be called covariate shift in distribution.
- The data distribution that generates the feature vector x and its related class label y changes as a result of a latent variable t. Thus, it may be stated that a covariate shift has occurred when P (y|x, t1) ≠ P(y|x, t2) [1].

III. Covariate Shift Detection and Adaption Algorithms

The error in predictions caused due to covariate shift can be removed by using importance weight given by

$$W(X) = \frac{p_{test}(X)}{p_{train}(X)} \tag{1}$$

where $p_{test}(X)$ and $p_{train}(X)$ are the probabilities of finding an input X in the test and train datasets respectively . Equation (1) comes from the intuition that if the probability of a particular training instance occurring in the test set is high, it must get a higher weight.

$W(X)$ gives the importance values at each of the input training points, which when multiplied with those points will lead to more accurate predictions. However, this value is not known apriori and thus there's a need to estimate its value from the data samples. Listed below are the most prominent importance estimation methods that have been introduced in this field.

A. Kernel Density Estimation (KDE)

KDE is a non-parametric method to get an approximation of the probability density function of a random variable. The Gaussian kernel given by eq.2 gives the KDE as shown in eq.3

$$K\,(x, x') = \exp\left(\frac{-\left|\left|x-x'\right|\right|^2}{2\sigma^2}\right) \tag{2}$$

$$\hat{p}(\mathrm{x}) = \frac{1}{n(2\pi\sigma^2)^{\frac{d}{2}}} \sum_{i=1}^{n} K_\sigma(x - x_i) \tag{3}$$

Where x and x' are two kernel samples and σ is the kernel width. The accuracy of the approximations given by KDE is solely determined by the chosen value of σ in the equation above. The best possible value of σ can be obtained by cross validation [2].Thus, the training and test data points can be used to obtain $\hat{p}_{train}$ and $\hat{p}_{test}$ respectively using eq.2 and the importance can be estimated as

$$W(X) = \frac{\hat{p}_{test}(X)}{\hat{p}_{train}(X)}.$$

However, the approach discussed above suffers from the curse of dimensionality [2, 3] and the amount of data required to support a reliable approximation often grows exponentially with dimensionality, which is a complication when the number of data samples is limited. Thus, KDE cannot be used for high dimensional data. A workaround for this is to directly find $W(X)$ without computing p_{train}(X) and $p_{test}(X)$.

B. Discriminative Learning

Probabilistic classifiers can also be used to directly estimate the importance [4-6]. Samples drawn from train set are labelled μ=0 and from test set are labelled μ=1.Thus, the densities can be given by

$$p_{tr}(X) = p(X|\mu = 0) \text{ and } p_{te}(X) = p(X|\mu = 1)$$

Using Bayes theorem, Importance weight W(x) can be written as

$$W(X) = \frac{p_{tr}}{p_{te}} = \frac{p(\mu = 0)\; p(\mu = 1|X)}{p(\mu = 1)\; p(\mu = 0|X)}$$

where $\frac{p(\mu=0)}{p(\mu=1)} \approx \frac{n_{tr}}{n_{te}}$ can be easily found.

The probability $p(\mu|X)$ can be approximated by *discriminating* $\{x_i\}_{i=1}^{n_{tr}}$ and $\{x_j\}_{j=1}^{n_{te}}$ using classifiers like Logistic Regression, Random Forests, SVM etc. It may also be noted here that the probability with which training samples can be segregated from test samples maybe used as a measure to detect if covariate shift exists in the dataset, called *discriminative testing* in this paper. However, training these models can be time consuming at times and thus efficient probabilistic classification methods like LSPC [7, 8] (least -squares probabilistic classifier) and IWLSPC (importance weighted LSPC which combines importance reweighting with LSPC) have been introduced [9, 10].

C. Kernel Mean Matching

Kernel mean matching (KMM) directly finds $W(X)$ without calculating p_{train} (X) and $p_{test}(X)$ [11, 12].The basic idea of KMM is to find $W(X)$ such that the means of the training and test points in a reproducing kernel Hilbert space (RKHS) [13] are close**.** The Gaussian kernel (equation 2) is an example of kernels that induce a universal RKHS, and it has been proved that the solution of the optimization problem given by equation 4 gives the true importance values:

$$\min_{w_i}\left[\frac{1}{2}\sum_{i,i'=1}^{n_{tr}} w_i w_{i'} K_\sigma(x_i^{tr}, x_{i'}^{tr}) - \sum_{i=1}^{n_{tr}} w_i\, \mathrm{K}_i\right]$$

(eq. 4)

Subject to $\left(\frac{1}{n_{tr}}\right)\left|\sum_{i=1}^{n_{tr}} w_i - n_{tr}\right| \leq \epsilon$ and

$0 \leq \omega_1, \omega_2, \omega_3, \cdots \omega_{n_{tr}} \leq B$

where $\mathrm{K}_i = \frac{n_{tr}}{n_{te}} \sum_{j=1}^{n_{te}} K_\sigma\left(x_i^{tr}, x_j^{te}\right)$

The solution to eq. 4 gives an estimate of importance at $\{x_i\}_{i=1}^{n_{tr}}$. The performance of KMM is solely determined by the values of the tuning parameters B, ϵ, and σ, hence, usual model selection methods like Cross validation can't find optimal values. An inductive variant of KMM [14] is a possible solution. σ is chosen as the median distance between samples [15].The work discussed in [12] gives a theoretical result which can be help in getting a correct value of σ. KMM has been experimentally proved to be superior to natural plug-in estimators [16].

D. Kullback Leblier Importance Estimation Procedure (KLIEP)

Model selection of algorithms like KMM done by cross validation may fail because of bias under covariate shift and hence an importance weighted version on CV called IWCV (Importance weighted Cross validation) [17] is used. However, in IWCV, model selection needs to be done by unsupervised learning inside the importance estimation step which is a major drawback. KLEIP finds an importance estimate $\widehat{w}$(x) such that the Kullback-Leibler divergence [18] between the true test input density$p_{te}(x)$and $\hat{p}_{te}(x)$ is minimized, where $\hat{p}_{te}(x) = \widehat{w}(x)\, p_{tr}(x)$. This is done without explicitly modelling $p_{te}(x)$ and $p_{tr}(x)$ [19, 20]. Model selection is carried out by using a variant of likelihood CV where the test samples are cross validated instead of training samples [21].

E. Least Squares Importance Fitting (LSIF), Unconstrained Least Squares Importance Fitting (uLSIF)

KLIEP used Kullback-Leibler divergence to find difference in densities between two functions. LSIF uses squared-loss instead. $\widehat{w}(x)$ is modelled as in KLIEP. Cross-validation is used for finding the optimal values of the tuning parameters like the regularization parameter and the kernel width σ. However, LSIF at times gives incorrect results due to the accumulation of numerical errors. To deal with this, an approximation version of LSIF has been proposed, called uLSIF which allows computation of solutions by simply solving a system of linear equations. Thus, uLSIF is numerically stable.

Once the importance weights are found, various importance weighted learning methods can be used to learn the parameters for training the models. Instead of empirical risk minimization-ERM (A standard method to learn the parameter θ), it's importance weighted version- IWERM and it's adaptive (adaptive IWERM) and regularized versions (regularized IWERM) have been introduced. Similarly, weighted variants exist for regression methods like Least squares (IWLS), least absolute regression (IWLAR), support vector regression (IWSVR) etc. and for classification approaches like importance weighted logistic regressor (IWLR), SVM (IWSVM) and boosting (IWB).

Apart from these, few other aspects explored in this research-area include detection using two stages, namely- EWMA model and KS test [22], adaptation by applying Frank-Wolfe algorithm to KMM and KLIEP [23] and dealing with shifts in incremental learning environments [24].

IV. Applications of Covariate Shift Adaption on Real World Problems

In a few real world scenarios, an assumption is made that covariate shift exists in the training and test data distributions and adaptation techniques are applied. This has brought a significant improvement in the performance of machine learning algorithms.

One such example is 'semi-supervised speaker identification' [25], where a technique that can lighten the impact of non-stationarity such as session-dependent variations, changes in the recording scenario, and physical emotion has been proposed for identifying the speaker of a speech. An assumption is made that the covariate shift occurs where changes occur only in the sample distribution in the training and test phases. Covariate shift adaptation method with weighted versions of kernel logistic regression and cross-validation has been used which has the capability of reducing the effect of covariate shift. The proposed method is seen dealing with variations in session dependent variation units. Speech emotion recognition is also shown to be done better by adaptation.

Variations in scaling and rotation are modelled as shifts in the feature vector; these variations are represented as a covariate shift in the data [26]. the covariate shift is minimized by reducing the Kullback–Leibler divergence between the estimated and true distributions using importance weights. This approach gives a generalized solution that is applicable to any texture descriptor that attempts to model the transforms as a covariate shift of feature vectors.

Perceived Age Estimation from face images required for demographic analysis especially in public places like shopping malls to target advertisements based on prospective customers' genders and ages also takes advantage of covariate shift adaptation. Training and test data tend to have different distributions due to varying lighting conditions in the environment. Covariate shift adaptation is done for alleviating lighting condition change thereby contributing to the efficiency of age estimation under lighting conditions. The technique used here for minimizing the covariate shift is Kullback Leblier Importance Estimation Procedure, and the performance measure is the least Weighted Mean Square error.

Furthermore, Electroencephalogram (EEG) based Brain Computer Interfaces (BCIs) is a field that has widely benefited from covariate shift adaption approaches [27-29]. It has been observed that the non-stationary nature of the brain signals producing the EEG data may induce a time-variation over the input probability distribution, which often appears as a covariate shift [12]. It is proved that bagging combined with covariate shift adaptation increases stability when applied to a few real world dataset. Apart from this, EEG signals have also been used to detect emotions using PCA based covariate shift adaptation in deep learning networks.

There are many other fields where covariate shift adaptation has been applied to including social media text classification [30], software engineering prediction models used for estimating development and maintenance cost/effort, fault count, reliability of software projects [31], automatic sleep stage classification [32], improvement of prosthetic device monitoring and control [33, 34], medical and health insurance market risk assessment [35], remote-sensing [36], audio tagging and many more.

V. Performance Analysis of classification algorithms

As mentioned in section III there are various techniques for both detecting and handling covariate shift. Some of them have been implemented and the explanation of the same is presented here. For detection of Covariate shift, various Exploratory Data Analysis (EDA) techniques and Discriminative Testing have been used. Similarly, for handling covariate shift, Importance Reweighting, KLIEP, and Dropping of unimportant features that lead to covariate shift have been tried. The datasets used in the experiment are three synthetic datasets and one dataset is taken from Kaggle repository. Details of the dataset and the techniques are explained in the next section. Following are the classification algorithms used in the experiments:

- Linear Discriminant Analysis
- *k*-Nearest Neighbor
- Decision Tree Classifier
- Naive-Bayes Classifier

There is no specific reason behind selecting these classifiers. Any supervised machine learning algorithm can be selected, with an intend to find if there is any difference in the performance of that particular classifier on the dataset with covariate shift after applying the adaption algorithms. The performance of the classifiers is evaluated using accuracy score.

VI. Experiments and results

Synthetic datasets have been used to conduct the experiments. The train and test datasets are made such that they follow different distributions. Details of the experiments and results obtained are presented in this section.

A. Train and Test data follow different Distributions

The dataset (Referred to as Dataset-I) has 1000 training samples and 1000 test samples. The distribution of the training samples is normal while that of test sample is taken to be binomial. The training samples contain feature variables X, Y. The training set is made by selecting 1000 random samples following uniform distribution with variance = 1 and the mean = 25 (referred to as 'data' in eq. 5a below). The features X, Y are computed using the model presented in equation 5:

$$X = 11 \times data - 6 \tag{5a}$$

$$Y = X^2 + 10 \times X - 5 \tag{5b}$$

Similarly the test set is made of 1000 random numbers following binomial distribution with the probability of being selected p=0.8 and the value ranges from 1 to 20. The features X, Y are computed using equation 5.

To confirm that there's a difference in the distributions of train and test sets, EDA is used where the graphs of $\hat{p}_{train}$ and $\hat{p}_{test}$ are plotted, i.e. Kernel Density Estimation (KDE) is done. Figure 1 shows the distribution of the training and the testing samples where the continuous curve shows the training samples and the dotted curve shows the testing samples.

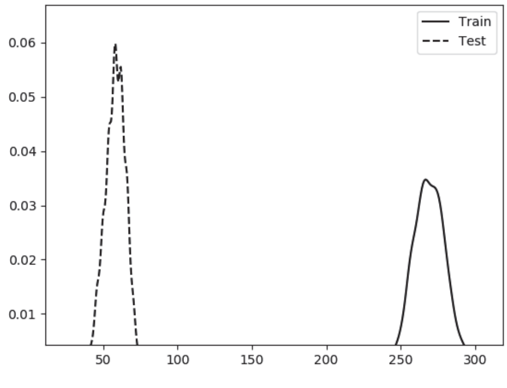

Fig. 1. KDE plot for initial distribution of Dataset-I

It is very clear from the distribution that there exists covariate shift distribution in this dataset. Hence the next step will be handling them. However before handling the covariate shift, classification algorithms have been applied to this dataset; four classification algorithms (LDA, *k*-NN, DT, NB) have been applied and their accuracy scores are recorded.

The technique used for handling covariate shift is discriminative learning. The steps followed for the same are: first, the training and testing sets are labelled as 1 and 0 respectively and are merged; next, a Logistic Regression classifier is applied on this newly created model to obtain the weights as described in section III. Once these values are calculated, they are multiplied to the same samples in training set; after handling of covariate shift, the performance of the machine learning algorithms on the new dataset are tested.

The accuracy of the algorithms is seen to be nearly 0.6 after handling covariate shift while the accuracy before handling was nearly only 0.2 for LDA, *k*-NN, Decision Tree, and approximately 0.7 for Naive Bayes Classifier. Thus, the accuracy scores of LDA, KNN, and Decision Tree classifiers has improved while that of Naive Bayes Classifier dropped after handling shift.

B. Train and Test data following same distribution with differing mean and variance

For the second experiment, the dataset created has the training and the testing samples of same distribution but with different mean and variance. The training and testing set individually has 1000 samples each. (Referred to as Dataset-II). The distribution of the training and the test sample are both uniform. The training set consists of two features - X, Y. 1000 random values are generated with mean as 25 and variance 1. To compute X, Y, the same approach presented in equation 5 is used. Similarly, the test data is created with normal distribution and comprises of two columns X, Y. 1000 random numbers are selected with mean equals 80 and variance as 1.

For covariate shift detection, the technique used is Kernel Density Estimation. The following figure shows the data distribution of the training and the testing dataset. The continuous curve shows the training set and dotted curve shows the test set.

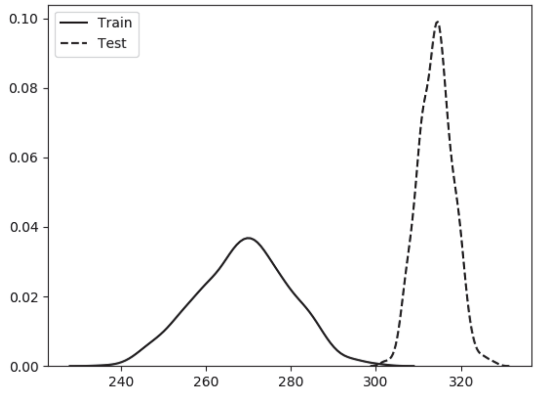

Fig. 2. Initial distribution of Dataset-II

It is clearly evident from figure 2 that there exists covariate shift between the distributions in dataset-2. Hence, Covariate shift adaption is done by using importance weighting, obtained by dividing test with train probability densities for the samples.

TABLE I. CLASSIFIER PERFORMANCE BEFORE AND AFTER HANDLING COVARIATE SHIFT ON DATASET-II

	Accuracy Score Before Handling	Accuracy Score After Handling
LDA	0.0745	0.7056
k-NN	0.0840	0.2022
DT	0.0844	0.7345
NB	0.0622	0.3218

The performance of the classifier after handling is shown in the table 1. It can be observed that the classifiers show significant improvement in their performances.

C. *Train and Test data has same distribution with increased number of attributes*

For the third experiment, two datasets with normal distribution are generated. The training and the testing set have around 500 samples and 3 attributes - X, Y and Z where X, Y are the input attributes and Z is the predicted label. For the training set mean is taken as 0 and the scale is 0.5 (Dataset – III). Equation 6 presents the computation of Z.

$$Z = \sin(Y \times \pi) + X \tag{6}$$

where X, Y are the random values generated. The same approach is used for the test dataset as well, but the X, Y values for the test dataset are generated such that the distribution is normal but with a mean of 1 and the scale is still 0.5.

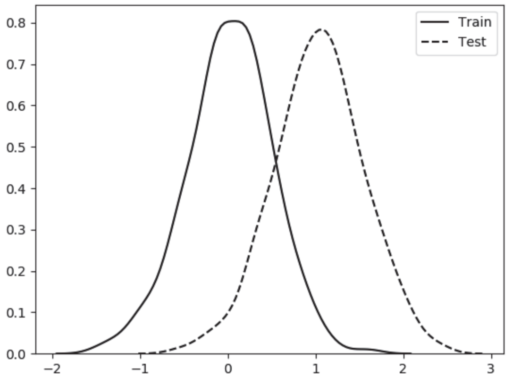

Fig. 3. Initial distribution of Dataset-III

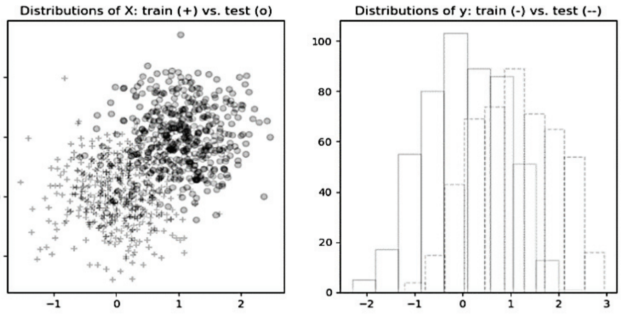

Fig.4. Scatter plot and histogram intersection of Dataset-III

For covariate shift detection, Kernel Density Estimation is used. Using this technique, the variation in the distribution of the datasets can be observed as shown in Fig.3.The scatter plot and the histogram intersection for the dataset is also shown in Fig.4

As is clearly evident from the distribution curve there exists a covariate shift in this dataset. To handle the shift, Importance Reweighting using KLIEP (Kullback Liebler Importance Estimation Procedure) is employed.

This experiment has shown a peculiar behavior where it is seen that the performance of the Decision Tree Classifier has improved by about 30% while that of other three classifiers has reduced by nearly 20%.

D. *Real World Dataset*

This experiment is conducted on the ‘Russian Housing Market’ dataset taken from Kaggle repository (Dataset-IV). This dataset has 291 attributes and one label 'price_doc'. The training set comprises of 30.5k samples and the test set consists of 7662 samples. Only basic preprocessing was required for this dataset which included filling the missing values with the mean value of that attribute in case of numeric attribute and with the mode values for the categorical attributes. Also the categorical attributes which were in string initially were converted into categorical codes like 0, 1 and 2.

To check for covariate shift in this dataset, the idea is taken from Discriminative Testing described in section III. A random forest classifier is applied on each attribute to compute each of their ROC-AUC scores. Attributes having an ROC-AUC score greater than 0.8 are considered to contribute to covariate shift because those are the attributes which contribute highly in discriminating the training samples from the test samples. One of these features in the dataset is found to be the “timestamp” variable. Fig.5 shows the distribution of timestamp variable for training and testing datasets.

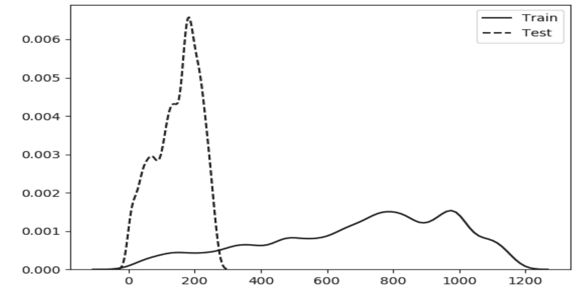

Fig. 5. Distribution of timestamp variable for training and testing datasets

TABLE II. CLASSIFIER PERFORMANCE BEFORE AND AFTER HANDLING COVARIATE SHIFT ON DATASET-IV

	Accuracy Score Before Handling	Accuracy Score After Handling
LDA	0.27	0.78800
k-NN	0.36	0.89840
DT	0.345	0.88730
NB	0.3423	0.87675

To handle covariate shift, all the features which contribute to covariate shift and are not important in the dataset are dropped. For this, a Random forest regressor model based ranking of feature importance is done. Leaving behind the top 20 important attributes, all the other attributes which contribute to covariate shift (i.e. attributes which are unimportant and contribute to covariate shift) are dropped. After dropping the features, classifiers are trained again only to see a significant improvement in the result as shown in table 2.

VII. Conclusion

The performance of a machine learning algorithms is an important factor to be considered for their implementation into real world scenario; it greatly depends on the datasets and the distribution of the data. When the machine learning model such as a decision tree or a neural network, is trained to one scenario and is exploited to improve generalization in another scenario, then the domain adaptation that occurs is called Transfer Learning. But in supervised learning algorithms, to make sure that a model works well in both the training and testing scenario it is important to ensure that the distribution of train and test samples is same. If the distribution of the testing dataset is seen to vary from the training set, it is the role of the knowledge engineer or developer to detect this variation and also handle them to thereby tune the performance of the model so that it generalizes well over unseen samples. This work presents a general overview of the techniques to detect and deal with this issue along with their real world applications and performance analysis. It also analyzes the results experimentally via synthetic and real datasets. The vision is that this fast growing area of research – covariate shift adaptation – will significantly catalyze the performance of all branches of data analysis. It would be interesting from the research perspective particularly to extend these approaches thoughtfully and gain better insights about covariate shift and associated data shift problems.

References

[1] Hidetoshi Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," Journal of statistical planning and inference, vol. 90, no. 2, pp. 227-244, 2000.

[2] Hardle and Wolfgang, Non Parametric and semi parametric models.: Spinger sceince and business media, 2012.

[3] Vanpik, Statistical Learning Theory. NewYork: Wiley-Interscience, 1998.

[4] Cheng, Kuang Fu, and ChihKang Chu, Semiparametric density estimation under two-sample density ratio model, 2004.

[5] Steffen Bickel, Brückner Michael, and Scheffe Tobias, "Discriminative learning under covariate shift," Journal of Machine Learning Research, vol. 10, pp. 2137-2155, 2009.

[6] Bickel, Briickner M, and Scheffer T, "Discriminative Learning for differing training and test distributions," in 24th International conference on Machine Learning, 2007, pp. 81-88.

[7] Sugiyama M, Superfast trainable multi-class probabilistic classifier by least-squares posterior fitting, 2010.

[8] Kanamori, Takafumi, and Shohei Hido, "A Least-Squares approach to direct importance estimation," Journal of Machine Learning Research, pp. 1391-1445, 2009.

[9] Hachiya, Hirotaka, Sugiyama M, and Naonori Ueda, Importance weighted least-squares probabilistic classifier for covariate shift adaption with application to human activity recognition, 2012.

[10] Yamada and Makoto, "Improving the accuracy of least-squares probablistic classifiers," in IEICE transactions on information and systems, 2011, pp. 1337-1340.

[11] Gretton and Artur, "Covariate shift by Kernel Mean Matching," in Dataset shift in machine learning., 2009.

[12] Huang and Jiayuan, "Correlating sample selection bias by unlabeled data," 2007.

[13] Steinwart and Ingo, "On the influence of the kernel on the consistency of support vector machines," Journal of machine learning research, pp. 67-93, 2001.

[14] Kanamri, Suzuki, Taiji Takafumi, and Sugiyama Masashi, Condition number analysis of kernel-based density ratio estimation, 2009.

[15] Scholkopf, Bernhard, and Alexander Smola, Learning with kernels: Support Vector Machines, regularization, optimization, and beyond.: MIT Press, 2001.

[16] Yu, Yaoliang, and Szepesvari Csaba, Analysis of Kernel mean matching under Covariate shift, 2012.

[17] Moreno Torres, Jose Garcia, Jose A.Saez, and Fransisco, "Study on the impact of partition-induced dataset shift on K-fold cross-validation," in IEEE Transactions on Neural Networks and Learning Systems, 2012, pp. 1304-1312.

[18] Kullback S and A, Leibler R, "On information and sufficiency," Annals of Mathematical Statistics, pp. 79-86, 1951.

[19] Sugiyama Masashi and et al., "Direct importance estimation for covariate shift adaption," Annals of the Institute of Statistical Mathematics, pp. 699-746, 2008.

[20] Tsuboi and Yuta, "Direct density ratio estimation for large-scale covariate shift adaption," Journal of Information Processing, pp. 138-155, 2009.

[21] Masashi Sugiyama, Nakajima Shinichi, Kashima Hisashi, Buenau Paul V., and Kawanabe Motoaki, "Direct importance estimation with model selection and its application to covariate shift adaptation," Advances in neural information processing systems, pp. 1433-1440, 2008.

[22] Raza, Haider, Girijesh, and Yuhua, "EWMA model based shift-detection methods for detecting covariate shifts in non-stationary ," Pattern Recognition, pp. 659-669, 2015.

[23] Wen, Junfeng, Russell Greiner, and Dale, "Correcting covariate shift with Frank-Wolfe algorithm," in Twenty-fourth International Joint Conference on Artificial Intelligence, 2015.

[24] Yamauchi and Koichiro, Optimal Incremental Learning under Covariate Shift, 2009.

[25] Makoto Yamada, Masashi Sugiyama, and Tomoko Matsui," Covariate shift adaptation for semi-supervised speaker identification," in IEEE, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing.

[26] Ali Hassan, Riaz Farhan, and Shaukat Arslan, "Scale and rotation invariant texture classification using covariate shift methodology," IEEE Signal Processing Letters, vol. 21, no. 3, pp. 321-324, 2014.

[27] Raza and Haider, "Adaptive learning with Covariate shift detection for motor imagery based brain-computer interface," Soft Computing, pp. 3085-3096, 2016.

[28] Spuler, Martin, Wolfgang Rosenstiel, and Martin Bogdan, " Principal component based Covariate shift adaption to reduce non-stationarity in a MEG-based brain-computer interface," EURASIP Journal on Advances in Signal Processing, 2012.

[29] Chowdhury and Anirban, Cortico-Muscular-Coupling and Covariate Shift Adaptation based BCI for personalized NeuroRehabilitation of Stroke Patients, 2016, Proc. of BCI Meeting.

[30] Fei, Geli, and Bing Liu, "Social media text classification under negative covariate shift," in Conference on Emperical Methods in Natural Language Processing, 2015.

[31] Turhan and Burak, "On the dataset shift problem in software engineering prediction models," Emperical Software Engineering, pp. 62-74, 2012.

[32] Khalighi, Sirvan, Teresa Sousa, and Urbano Nunes. "Adaptive automatic sleep stage classification under covariate IEEE Engineering in Medicine and Biology Society, 2012.

[33] Vidovic and Marinna M-C, "Improving the robustness of mypelectric pattern recognition for upper limb prostheses by Covariate shift adaptation," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2016.

[34] Vidivic and Marina M-c, "Covariate shift adaptation in EMG pattern recognition for prosthetic device control," in 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Soceity, 2014.

[35] Wei, Dennis, Karthikeyan Ramamurthy, and Kush Varshney, "Health insurance market risk assessment: Covariate shift and k-anonymity," in SIAM International Conference on Data Mining, 2015.

[36] Tuia, Devis, Pasolli, and William J.Emry, "Using active learning to adpt remote sensing image classifiers," Remote Sensing of Environment, pp. 2232-2242, 2011.