Inter-domain Cluster Mapping and GMCV Based Transformation for Domain Adaptation

Suranjana Samanta and Sukhendu Das

V.P. Lab, Dept. of CSE, IIT Madras, India
ssamanta@cse.iitm.ac.in, sdas@iitm.ac.in

Abstract. This paper describes an algorithm for a direct solution of domain adaptation (DA) to transform data in source domain to match the distribution in the target domain. This is achieved by formulating a transformation matrix based on the Geometric Mean of Co-Variances (GMCV), estimated from the covariance matrices of the data from both the domains. As a pre-processing step, we propose an iterative framework for clustering over data from both the domains, to produce an inter-domain mapping function of clusters. A closed form solution for direct DA is obtained from the GMCV formulation. Experimental results on real world datasets confirms the importance of clustering prior to transformation using GMCV for better classification accuracy. Results show the superior result of the proposed method of DA, when compared with a few state of the art methods.

1 Introduction

The basic assumption of the standard techniques of classification is that the training and the testing samples are drawn from the same distribution. This assumption may not always be true, specially for large-scale, real-world datasets, where we do not have clear idea about the distribution of the datasets. In certain situations, we have a very few number of training samples available from the domain of test samples for a classification task, but a large number of labeled samples are available from an auxiliary domain. During training, a very few number of training samples generally causes an over-fitting for any learning model, leading to inferior performance of classification.

The task of classification is done on test samples obtained from target domain, which often provides a few labeled/unlabeled training samples for training. On the other hand, the domain from which a large number of labeled training samples with a different distribution are available is termed as the source domain. Domain adaptation (DA) is the process of using training samples available from source domain to aid a statistical learning task to be done on testing samples obtained from target domain. There are mainly two types of DA techniques available in the literature - (a) supervised - where we have a very few number of training samples from the target domain and (b) unsupervised - where we have unlabeled training samples from target domain.

P. Maji et al. (Eds.): PReMI 2013, LNCS 8251, pp. 74–81, 2013.

There has been a lot of work in DA in recent past. One solution to this problem is to weigh each instance in the source domain appropriately such that, the weighted instances of the source domain are used for training to minimize the expected loss [1], [2]. In some works, instances from source and target domains are projected onto one or more intermediate domain(s), such that the difference in distribution in two domains are less in projected domain(s). Jiang et. al. [3] and Yang et. al. [4] have proposed methods of modifying the SVM trained on samples available from source domain by introducing a bias term between source and target domains during optimization in training. There has been some work on clustering using transfer learning [5], [6]. In [5], clustering is done using KL-Divergence in a common sub-space of features while in [6], a divergence measure based on inter-cluster and intra-cluster distances has been used. The number of clusters formed in both the domains are same and known apriori.

In this paper, we propose a direct method of supervised DA, where the instances from source domain is transformed to match the distribution of the target domain using a transformation matrix. Our work uses the concept of inter-domain clustering for successful transformation of data from one domain to another. The number of clusters are automatically determined during an iterative clustering framework. We use the geometric mean of covariance (GMCV) matrices of data, for direct closed-form solution of a transformation matrix exploiting the underlying distribution in both the domains. Results are shown on datasets obtained from UCI repository [7] and a remote sensing dataset [8].

The rest of the paper is organized as follows. Section 2 gives the description of the proposed method of clustering and domain transformation. Section 3 presents and discusses the performance of the proposed methodology on real-world datasets. Section 4 concludes the paper.

2 Proposed Solution of Domain Adaptation

This paper discusses a method of DA, where instances from the source domain are transformed to match the distribution of the target domain. The proposed method, consists of two stages - (a) clustering data in both the domains and (b) Calculating Transformation Matrix using GMCV - to transform a source domain data into target domain. We assume that a minimum of one training sample per class is available from both source and target domains.

Let $\mathbf{X}, \mathbf{Y} \in \Re^{D \times n_s}$ denote the source and the target data having n_s and n_t number of samples respectively. Let $\mathbf{X}_i$ and $\mathbf{Y}_j$ denote the i^{th} and j^{th} clusters of $\mathbf{X}$ and $\mathbf{Y}$ and let x_i and y_j be the i^{th} and j^{th} instance of $\mathbf{X}$ and $\mathbf{Y}$ respectively. The entire process has been explained in the following sub-sections.

2.1 Inter-domain Clustering and Mapping

An iterative process of clustering is performed in both the domains simultaneously, where the clusters formed in one domain is influenced by the clusters formed in another domain. The proposed clustering method can be seen as an extension of K-means clustering, where we cluster data in both domain simultaneously, where the clusters in the source domain helps to form appropriate clusters in target domain. At first, data is normalized in both the domains and $\mathbf{X}$ is then approximately aligned with the distribution of $\mathbf{Y}$ to obtain $\hat{\mathbf{X}}$, using: $\hat{\mathbf{X}} = (E_t^{-1} E_s)\mathbf{X}$, where E_s and E_t are matrices formed using the eigen-vectors of datasets $\mathbf{X}$ and $\mathbf{Y}$ respectively. $E_t^{-1} E_s$ is the higher dimensional rotation matrix which aligns the basis vectors (based on eigen-analysis) of $\mathbf{X}$ with that of $\mathbf{Y}$, such that the scatter of $\hat{\mathbf{X}}$ and $\mathbf{Y}$ are qualitatively similar to each other with significant overlap along each dimension of E_t. Let $\hat{x}_i$ be the i^{th} instance of $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}_i$ be the i^{th} cluster of $\hat{\mathbf{X}}$. Next, clusters are initialized in $\hat{\mathbf{X}}$ using K-Means clustering. The number of clusters for initialization must be considered high and it reduces in the successive iterations to give an appropriate value. Let at any iteration, K_s and K_t be the number of clusters in $\hat{\mathbf{X}}$ and $\mathbf{Y}$ respectively. Let, $\omega_s : \{1, \ldots, K_s\}$ and $\omega_t : \{1, \ldots, K_t\}$ be the sets of cluster-labels in $\hat{\mathbf{X}}$ and $\mathbf{Y}$ respectively. Let μ_s^i and μ_t^j be the mean of $\hat{\mathbf{X}}_i$ and $\mathbf{Y}_j$, $\forall i \in \omega_s$ and $\forall j \in \omega_t$.

The clusters are formed in $\mathbf{Y}$ based on the distance of each instance y_j from the means formed in $\hat{\mathbf{X}}$. The distribution of clusters are ignored in this case as it was observed to produce inferior results for classification using DA. A mapping function $\Phi : \omega_t \to \omega_s$ is calculated as,

$$\Phi(j) = i \text{ if } i = \arg\min_k dist(\mu_t^j, \mu_s^k), \forall k \in \omega_s \tag{1}$$

Here $dist(.,.)$ represents the Euclidean distance between two instances. The distribution of clusters are ignored in this case as it was observed to produce inferior results for classification in DA. Once the clusters in $\mathbf{Y}$ are formed, the clusters in $\hat{\mathbf{X}}$ are reformed based on the Bregman divergence [9] from the means of clusters of $\mathbf{Y}$. We consider the Bregman divergence of the convex function $x^T Ax$, which is given by $(x - y)^T A(x - y)$. Here, A is a positive definite matrix. For our experimentation, we consider A as $E_t^{-1} E_s$. To ensure the formation of compact clusters in $\hat{\mathbf{X}}$, inter-cluster distance is also considered. The iteration converges when the change in the cluster-labels of instances in $\hat{\mathbf{X}}$ and $\mathbf{Y}$ in successive iteration is less than 2%. The algorithm of cross-domain clustering and mapping is given in Algo. 1.

The iterative process (steps 3-9) in Algo. 1 helps to reduce the number of clusters formed in both the domains. The cluster in $\hat{\mathbf{X}}$, whose mean is not close to any of the instances in $\mathbf{Y}$ will remain outside the range of Φ (steps 4 & 5). The number of clusters formed in $\hat{\mathbf{X}}$ in step 8, will then be less than or equal to the cardinality of the range of Φ. This condition reduces the K_s which in turn will reduce K_t in the next iteration (step 4). While re-clustering $\hat{\mathbf{X}}$ (step 8), α_2 ensures that an instance $\hat{x}_i$ is assigned to a cluster with minimum Bregman divergence [9] from a cluster in $\mathbf{Y}$. On the other hand, α_1 ensures that the cluster assigned to $\hat{x}_i$ yields a compact cluster. Hence, clusters formed in both the domains will be compact due to the distance criterion (steps 4 & 6) used.

Algorithm 1. Cross-Domain Clustering and Mapping.
INPUT: Source domain: $\mathbf{X}$ and target domain: $\mathbf{Y}$.
OUTPUT: Cluster-label of instances in $\mathbf{X}$ and $\mathbf{Y}$, i.e., $\delta(x_i)$ and $\delta(y_j) \forall i = 1, 2, \ldots, n_s$ and $\forall j = 1, 2, \ldots, n_t$ and the mapping function $\varPhi : \omega_t \rightarrow \omega_s$.

```
Normalize $\mathbf{X}$ and $\mathbf{Y}$ and align source domain to obtain $\hat{\mathbf{X}} = (E_t^{-1} E_s)\mathbf{X}$.
Cluster $\hat{\mathbf{X}}$ with $K_s$ (large) number of clusters using K-means, forming means
  of clusters $\mu_s^i$ for $i \in \omega_s$.
repeat
    Cluster instances in $\mathbf{Y}$ as: $\delta(y_j) = \arg\min_k dist(y_j, \mu_s^k)$ and obtain $\mu_t^j$ $\forall j$.
    Calculate the cross-domain mapping function, $\varPhi$, as:
    $\varPhi(j) = \arg\min_k dist(\mu_t^j, \mu_s^k), \forall k \in \omega_s$ and $\forall j \in \omega_t$.
    Calculate cluster compactness of $\hat{\mathbf{X}}_k$ as:
    $\alpha_1(k) = \frac{1}{n_s^k} \sum_{\hat{x} \in \hat{\mathbf{X}}_k} dist(\hat{x}, \mu_s^k), \forall k \in \omega_s$.
    Calculate Bregman divergence [9] of each instance of $\hat{\mathbf{X}}$ with respect to
    each of the clusters formed in $\mathbf{Y}$, as:
    $\alpha_2(i, k) = (\hat{x}_i - \mu_t^k)^T (E_t^{-1} E_s)(\hat{x}_i - \mu_t^k), \forall i = 1, 2, ..., n_s$ and $k \in \omega_t$.
    Re-cluster $\hat{\mathbf{X}}$, after normalizing $\alpha_1$ and $\alpha_2$, as:
    $\delta(\hat{x}_i) = \arg\min_k [\lambda \alpha_2(i, k) + (1 - \lambda)\alpha_1(\varPhi(k))]$, $\forall k \in \omega_t$ and $\forall i$. ($\lambda = 0.6$,
    found empirically).
    Recompute means of all clusters: $\mu_s^k$, $\forall k \in \omega_s$.
until Convergence (i.e., $\delta(\hat{x}_i)$ & $\delta(y_j)$ are unchanged)
```

2.2 Estimation of Transformation Matrix Using GMCV

We transform data from $\mathbf{X}$ considering each cluster at a time. We consider a transformation, which ensures that clusters formed in $\hat{\mathbf{X}}$ are same as that in $\hat{\mathbf{X}}$, i.e., $\delta(x_i) = \delta(\hat{x}_i), \forall x_i \in \mathbf{X}$. If $\varPhi(j) = i$, then $\mathbf{X}_i$ is transformed to match the distribution of $\mathbf{Y}_j$. Each of the clusters formed in $\mathbf{X}$ and $\mathbf{Y}$ are hyper-ellipsoidal in nature, which can be considered as clusters following a Gaussian distribution. It becomes less erroneous to calculate the transformation matrix if the given source and the target distributions can be modeled by a similar distribution but with different parameters. Let, $\mathbf{X}_i$ be transformed using a transformation matrix W_i to match the distribution of $\mathbf{Y}_j$ ($\varPhi(j) = i$). We use geometric mean of covariance matrices of data from $\mathbf{X}$ and $\mathbf{Y}$ to calculate W_i. Let, $\tilde{\mathbf{X}}_i$ be the transformed source domain cluster: $\tilde{\mathbf{X}}_i = W_i \mathbf{X}_i$.

Let C_s^i and C_t^j be the covariance matrices of $\mathbf{X}_i$ and $\mathbf{Y}_j$ respectively. The covariance matrix of $\tilde{\mathbf{X}}_i$ can be represented as $W_i C_s^i W_i^T$, which should be equal to C_t^j as the distribution of $\mathbf{Y}_j$ should be same as $\tilde{\mathbf{X}}_i$. This yields a quadratic expression of W_i: $W_i C_s^i W_i^T = C_t^j$. If W_i is symmetric, we get $W_i C_s^i W_i = C_t^j$ and W_i becomes the geometric mean of covariances (GMCV): $(C_s^i)^{-1}$ and C_t^j [10]. Let $C_s^i = U^T U$, where U is an upper triangular matrix obtained by performing Cholesky's decomposition of C_s^i. Let, $V = U^{-1}$ and $R^2 = U C_t^j U^T$, i.e., R is the matrix square-root of $UC_t^j U^T$. Then, the solution to W_i based on GMCV, is obtained as the following expression (substitution yields the proof ☒):

$$W_i = VRV^T \tag{2}$$

Eqn. 2 gives a direct solution to DA, to obtain $\tilde{\mathbf{X}}$ based on GMCV.

2.3 Result on a Toy Dataset for Illustration

To explain the steps of the proposed algorithm, we consider a simple example of data distribution in source ($\mathbf{X}$) and target ($\mathbf{Y}$) domains in $\Re^2$. In Fig. 1 (a) the cyan and the green points denote the instances from $\mathbf{X}$ and $\mathbf{Y}$. Fig. 1 (b) shows the overlap of $\hat{\mathbf{X}}$ with $\mathbf{Y}$, where the approximate alignment is performed using $E_t^{-1}E_s$ (step 1 in Algo. 1). The result of cross-domain clustering is shown in Fig. 1 (c). The yellow, light blue and orange points denote the data in three clusters formed in $\mathbf{X}$, while blue, magenta and black points denote the data in three clusters formed in $\mathbf{Y}$. Fig. 1 (d) shows the transformed source domain data, $\tilde{\mathbf{X}}$, in red points obtained using the proposed method (Eqn. 2). Fig. 1 (e) shows the transformed source domain data when clustering is avoided as the pre-processing step, producing inferior transformation compared to that shown in Fig. 1 (d). Transformed source domain data marked in red and violet in Figs. 1 (d) & (e), overlap the data in target domain, marked in green points. Fig. 1 (f) gives two plots shown against the number of clusters formed in each iteration, showing the change in: (i) ratio of cluster-labels in $\hat{\mathbf{X}}$ and (ii) the average KL-Divergence between the clusters formed in $\hat{\mathbf{X}}$ and $\mathbf{Y}$, for cross-domain clustering. Both monotonically decrease with increase in iterations (Algo. 1). The KL divergence measure with respect to the target domain, for the original, aligned and transformed source domains in Figs. 1 (a), (b), (d) and (e) are 14.8191, 0.5451, 0.0250 and 0.6240 respectively. Least value of KL-Divergence is obtained for Fig. 1 (d), which shows the efficiency of the proposed method, specifically when inter-domain clustering is used.

3 Experimental Results and Performance Analysis

We evaluate the performance of the proposed method on real world datasets obtained from UCI repository [7] and landmine dataset [8]. We describe the two sets of experimentation done to exhibit the efficiency of the proposed method.

Landmine dataset has been used for DA by Shi et. al. [8]. This dataset has 5 source domains (Dataset 20 - Dataset 24) with different distributions than the target domain and one source domain (Dataset 6-10) with similar distribution with the target domain. The dataset has two classes and the feature set has nine dimensions. Further details of the dataset is given in [8]. We conduct a 30-fold experimentation and consider one instance from every class of the target domain randomly for training purpose, which is similar to the experimental setup used by Shi et al. [8]. In this case, due to the presence of only one training sample

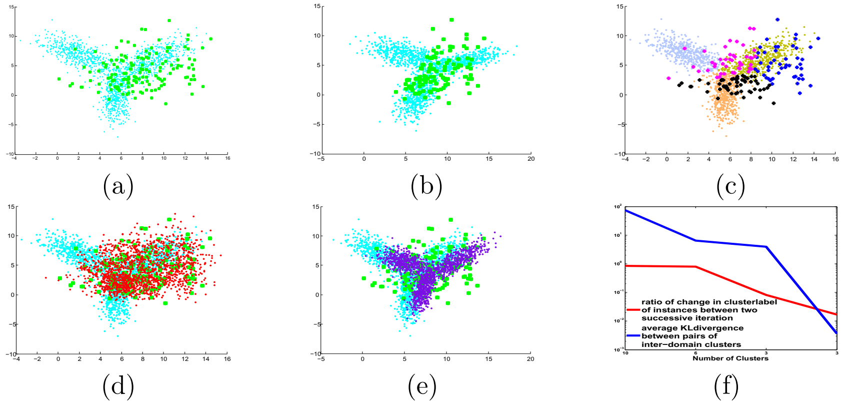

Fig. 1. Scatter plots (a)-(e) showing the effect of proposed method using a toy dataset. (a) cyan - original source domain, green - target domain; (b) cyan - aligned source domain (step 1 in Algo. 1), green - normalized target domain; (c) three clusters (shown in light blue, orange and mustard) formed in source domain and three clusters (shown in blue, magenta and black) formed in target domain using Algo. 1; (d) red - transformed source domain using the proposed method of clustering and GMCV transformation which completely overlaps with the distribution of target domain; (e) violet - transformed source domain using only GMCV transformation which partly overlaps with the distribution of the target domain; (f) Plots showing changes in average KL-Divergence between pair of inter-domain clusters with each iteration in blue and the ratio of change in cluster-labels in successive iterations in red, shown against the number of clusters in each iteration.

in target domain per class, application of the process of iterative clustering was not meaningful. Hence, we transform source data considering samples from all classes together, without using cross-domain clustering as the pre-processing step. The average accuracy of 30-fold experimentation using SVM classifier with Gaussian kernel is given in Table 1. We report the results in [8] (columns 2, 3 & 4), whereas the performance of ASVM [4] and CD-SVM [3] (column 5 & 6) are obtained using the code given in [11]. Results shows that our method (column 7) performs better than the state of the arts as reported, for all datasets having signification disparity of distribution between two domains (Dataset 20-24).

In the 2^{nd} set of experimentation, we consider 3 datasets - ionosphere, mushroom and WDBC from UCI repository [7]. We split each dataset into source and target domains using the method explained in [2]. We observe the accuracy of SVM classifier with Gaussian kernel by varying the fraction of training samples obtained from the target domain (0.3 to 0.7). Comparative studies are done using ASVM [4], CD-SVM [3] and KLIEP [1]. Two methods of experiments are performed using GMCV based transformation: (i) C-GMCV - with inter-domain clustering & (ii) GMCV - without inter-domain clustering. Results in Fig. 2 shows the classification accuracy using different techniques of DA for varying number of training samples obtained from the target domain. The red and the green curves show the classification accuracy using the proposed

Table 1. Classification accuracy (in %-age) of Landmine dataset [8] using different techniques of domain adaptation. Best classification accuracy is highlighted in bold.

Source Data	Source only [8]	TrAdaboost [2], [8]	AcTraK [8]	ASVM [4]	CD-SVM [3]	Proposed method
Dataset 20	57	89.76	94.49	94.31	90.77	**94.58**
Dataset 21	57	86.04	94.48	94.18	94.02	**94.58**
Dataset 22	57	90.50	94.49	94.44	94.56	**94.60**
Dataset 23	57	88.42	94.49	94.08	83.97	**94.58**
Dataset 24	57	90.70	94.49	94.34	93.57	**94.58**
Dataset 6-10	57	**94.76**	94.70	94.55	94.58	94.66

methods: C-GMCV and GMCV respectively. When the training samples from both the domains are taken together for training, we term it as the naive combination, whose performance is shown by the black curve. The blue curves show the performances when training samples from only one of the domains (**X** or **Y**) is used for training the classifier. The cyan, brown and magenta curves show the classification accuracy using different methods published for DA: KLIEP [1], ASVM [4] and CDSVM [3]. Plots for three datasets show that the proposed method of C-GMCV gives the best result. The results also show that the inclusion of proposed method of inter-domain clustering enhances the classification performance of DA.

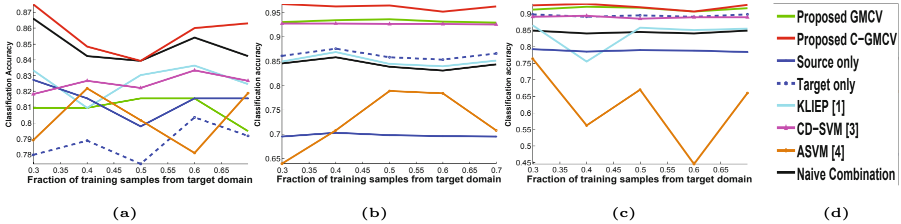

Fig. 2. Classification accuracy of (a) Ionosphere, (b) Mushroom and (c) Wdbc using different techniques of DA with varying fraction of training samples obtained from target domain. (d) shows the color indices indicating the methods used to get different performance curves. The proposed C-GMCV (in red curve) gives the best accuracy.

4 Conclusion

We propose an algorithm for a direct solution of domain adaptation based on geometric mean of covariance matrices (GMCV) of data in both the domains, which exploits the cluster information present in the data. Results demonstrated in the paper show that the proposed method is better than that reported in published literature in almost all the cases. The work can be extended for better object categorization and face recognition.

References

1. Sugiyama, M., Nakajima, S., Kashima, H., von Bünau, P., Kawanabe, M.: Direct importance estimation with model selection and its application to covariate shift adaptation. In: Neural Information Processing Systems, pp. 1962–1965 (2007)
2. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: International Conference on Machine Learning, pp. 193–200 (2007)
3. Jiang, W., Zavesky, E., Fu Chang, S., Loui, A.: Cross-domain learning methods for high-level visual concept classification. In: International Conference on Image Processing, pp. 161–164 (2008)
4. Yang, J., Yan, R., Hauptmann, A.G.: Cross-domain video concept detection using adaptive SVMs. In: International Conference on Multimedia, pp. 188–197 (2007)
5. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Self-taught clustering. In: International Conference on Machine Learning, pp. 200–207 (2008)
6. Bhattacharya, I., Godbole, S., Joshi, S., Verma, A.: Cross-guided clustering: Transfer of relevant supervision across domains for improved clustering. In: International Conference on Data Mining, pp. 41–50 (2009)
7. Asuncion, A., Newman, D.H.: UCI machine learning repository (2007)
8. Shi, X., Fan, W., Ren, J.: Actively transfer domain knowledge. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 342–357. Springer, Heidelberg (2008)
9. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman Divergences. Journal of Machine Learning Research 6, 1705–1749 (2005)
10. Lawson, J.D., Lim, Y.: The geometric mean, matrices, metrics, and more. The American Mathematical Monthly 108(9), 797–812 (2001)
11. Duan, L., Xu, D., Tsang, I.W.H.: Domain adaptation from multiple sources: A domain-dependent regularization approach. IEEE Transaction Neural Network Learning System 23(3) (2012), `http://vc.sce.ntu.edu.sg/transfer-learning-domain-adaptation/domain-adaptation-home.html`