Supplement to "Domain Adaptation under Target and Conditional Shift"

This supplementary material provides the proofs and some details which are omitted in the submitted paper. The equation numbers in this material are consistent with those in the paper.

S1. Classification and Regression Machines Used in This Paper

In this paper we consider both the classification and regression problems. For the former problem, we adopt the support vector classification, and for the latter we use the penalized kernel ridge regression. All parameters in the learning machines (e.g., the kernel width and regularization parameter) are selected by cross-validation.

Reweighted support vector classification: Support vector classifiers can be extended to incorporate non-uniform importance weights of the training instances. Associated with each training instance is the importance weight $\beta^*(y_i)\gamma^*(x_i, y_i)$, which can be incorporated into (1) via the following minimization problem:

$$\underset{\theta,b,\xi}{\text{minimize}} \quad \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{n}\beta^*(y_i)\gamma^*(x_i,y_i)\xi_i \tag{13a}$$

$$\text{subject to} \quad y_i(\langle\theta,\phi(x_i)\rangle + b) \geq 1-\xi_i, \xi_i \geq 0 \tag{13b}$$

where $\phi(x)$ is a feature map from $\mathcal{X}$ to a feature space $\mathcal{F}$. The dual of (13) is

$$\underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k(x_i,x_j) - \sum_{i=1}^{n}\alpha_i \tag{14a}$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq \beta^*(y_i)\gamma^*(x_i,y_i)C, \tag{14b}$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \tag{14c}$$

Here $k(x,x') \triangleq \langle\phi(x),\phi(x')\rangle_{\mathcal{F}}$ denotes the inner product between the feature maps. We have modified the LIBSVM implementation[4] for reweighted instances.

Reweighted kernel ridge regression (KRR): The original kernel ridge regression (Saunders et al., 1998) represents the vector of fitted target values as $\mathbf{f} = Kc$, where K is the kernel matrix of $\mathbf{x}^{tr}$, and find the estimate of c by minimizing $(\mathbf{y}^{tr} - Kc)^\intercal(\mathbf{y}^{tr} - Kc) + \lambda_x c^\intercal Kc$. The estimate is $\hat{c} = (K + \lambda_x I)^{-1}\mathbf{y}^{tr}$ and consequently, the fitted target values are $\hat{\mathbf{f}} = K\hat{c} = K(K + \lambda_c I)^{-1}\mathbf{y}^{tr}$. Similarly, the reweighted kernel ridge regression minimizes $(\mathbf{y}^{tr} - Kc)^\intercal \cdot \text{diag}\{\beta^*(\mathbf{y}^{tr}) \odot \gamma^*(\mathbf{x}^{tr},\mathbf{y}^{tr})\} \cdot (\mathbf{y}^{tr} - Kc) + \lambda_x c^\intercal Kc$, where $\odot$ denotes the Hadamard (or entrywise) product. This gives $\hat{c} = [K + \lambda_x \text{diag}^{-1}\{\beta^*(\mathbf{y}^{tr}) \odot \gamma^*(\mathbf{x}^{tr},\mathbf{y}^{tr})\}]^{-1}\mathbf{y}^{tr}$ and hence, the fitted values are $\mathbf{f} = K[K + \lambda_x \cdot \text{diag}^{-1}\{\beta^*(\mathbf{y}^{tr}) \odot \gamma^*(\mathbf{x}^{tr},\mathbf{y}^{tr})\}]^{-1}\mathbf{y}^{tr}$.

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

S2. Proof of Theorem 1 in Sec. 3

Proof 8.1 *In (4), $\mathcal{U}[P^{tr}_{X|Y}]$ is a linear operator, $\mathbb{E}_{Y\sim P^{tr}_Y}[\beta(y)\phi(y)]$ is linear in β. Further note that the constraints are convex. We can see that the optimization problem (4) is convex in β.*

According to assumption A_1^{TarS}, we have $\mu[P^{te}_X] = \mathcal{U}[P^{tr}_{X|Y}]\mu[P^{te}_Y]$, and the function in (4) reduces to

$$\left|\left|\mathcal{U}[P^{tr}_{X|Y}] \cdot \{\mathbb{E}_{Y\sim P^{tr}_Y}[\beta(y)\phi(y)] - \mu[P^{te}_Y]\}\right|\right|.$$

It achieves zero, which is clearly a minimum, when $\mathbb{E}_{Y\sim P^{tr}_Y}[\beta(y)\phi(y)] = \mu[P^{te}_Y]$. It is equivalent to $\beta(y)P^{tr}_Y(y) = P^{te}_Y(y)$, since the kernel l is characteristic. Moreover, combining assumptions A_1^{TarS} and A_4^{TarS} implies that there is no other solution of $\beta(y)$ to (4).

S3. Proof of Theorem 2 in Sec. 4

Proof 8.2 *This theorem is a special case of Theorem 3: in Theorem 3, setting $P^{new}_Y = P^{tr}_Y = P^{te}_Y$ gives this theorem.*

S4. Derivatives used in Sec. 4.2

The gradient of J^{ConS} w.r.t. $\tilde{K}$ and $\tilde{K}^c$ is

$$\frac{\partial J^{ConS}}{\partial \tilde{K}} = \frac{1}{m^2}(L+\lambda I)^{-1}L\mathbf{1}_m \cdot \mathbf{1}_m^\intercal L(L+\lambda I)^{-1}, \quad \text{and}$$

$$\frac{\partial J^{ConS}}{\partial \tilde{K}^c} = -\frac{2}{mn}\mathbf{1}_n\mathbf{1}_m^\intercal L(L+\lambda I)^{-1}.$$

Using the chain rule, we further have the gradient of J^{ConS} w.r.t. the entries of $\mathbf{G}$ and $\mathbf{H}$:

$$\frac{\partial J^{ConS}}{\partial G_{pq}} = \mathrm{Tr}\Big[\Big(\frac{\partial J^{ConS}}{\partial \tilde{K}}\Big)^\intercal \cdot (\mathbf{D}_{pq} \odot \tilde{K})\Big] - \mathrm{Tr}\Big[\Big(\frac{\partial J^{ConS}}{\partial \tilde{K}^c}\Big)^\intercal \cdot (\mathbf{E}_{pq} \odot \tilde{K}^c)\Big],$$
$$\frac{\partial J^{ConS}}{\partial H_{pq}} = \mathrm{Tr}\Big[\Big(\frac{\partial J^{ConS}}{\partial \tilde{K}}\Big)^\intercal \cdot (\tilde{\mathbf{D}}_{pq} \odot \tilde{K})\Big] - \mathrm{Tr}\Big[\Big(\frac{\partial J^{ConS}}{\partial \tilde{K}^c}\Big)^\intercal \cdot (\tilde{\mathbf{E}}_{pq} \odot \tilde{K}^c)\Big],$$

where

$$[\mathbf{D}_{pq}]_{ij} = -\frac{1}{l^2}(x^{new}_{jq} - x^{new}_{iq})(x^{tr}_{jq}R_{jp} - x^{tr}_{iq}R_{ip}),$$
$$[\mathbf{E}_{pq}]_{ij} = -\frac{1}{l^2}x^{tr}_{jq}R_{jp}(x^{new}_{jq} - x^{te}_{iq}),$$
$$[\tilde{\mathbf{D}}_{pq}]_{ij} = -\frac{1}{l^2}(x^{new}_{jq} - x^{new}_{iq})(R_{jp} - R_{ip}),$$
$$[\tilde{\mathbf{E}}_{pq}]_{ij} = -\frac{1}{l^2}R_{jp}(x^{new}_{jq} - x^{te}_{iq}).$$

The derivative of J^{reg} w.r.t. $\mathbf{G}$ and $\mathbf{H}$ is

$$\frac{\partial J^{reg}}{\partial \mathbf{G}} = \frac{2\lambda_{LS}}{m}R^\intercal(\mathbf{W} - \mathbf{1}_m\mathbf{1}_d^\intercal), \text{ and}$$
$$\frac{\partial J^{reg}}{\partial \mathbf{H}} = \frac{2\lambda_{LS}}{m}R^\intercal\mathbf{B}.$$

S5. Proof of Theorem 3 in Sec. 5

Combining assumption A^{ConS}, i.e., $P_X^{te} = \sum_i P_Y^{te}(y_i)P_{X|Y}^{(\mathbf{w}_i^*,\mathbf{b}_i^*)}(x|y_i)$, and the condition in Theorem 3, we have

$$\sum_i P_Y^{te}(y_i)P_{X|Y}^{(\mathbf{w}_i^*,\mathbf{b}_i^*)}(x|y_i) = \sum_i P_Y^{new}(y_i)P_{X|Y}^{(\mathbf{w}_i,\mathbf{b}_i)}(x|y_i)$$
$$\Rightarrow \sum_i \big[P_Y^{te}(y_i)P_{X|Y}^{(\mathbf{w}_i^*,\mathbf{b}_i^*)}(x|y_i) - P_Y^{new}(y_i)P_{X|Y}^{(\mathbf{w}_i,\mathbf{b}_i)}(x|y_i)\big] = 0.$$

Because of assumption A_2^{ConS}, we know that $\forall\ i$,

$$P_Y^{te}(y_i)P_{X|Y}^{(\mathbf{w}_i^*,\mathbf{b}_i^*)}(x|y_i) - P_Y^{new}(y_i)P_{X|Y}^{(\mathbf{w}_i,\mathbf{b}_i)}(x|y_i) = 0.$$

Taking the integral of the above equation gives $P_Y^{new}(y_i) = P_Y^{te}(y_i)$. This further implies $P_{X|Y}^{(\mathbf{w}_i,\mathbf{b}_i)}(x|y_i) = P_{X|Y}^{(\mathbf{w}_i^*,\mathbf{b}_i^*)}(x|y_i) = P_{X|Y}^{te}(x|y_i)$.

S6. Algorithm for LS-GeTarS in Sec. 5

We iteratively alternate between the QP to minimize (11) w.r.t β and the SCG optimization procedure w.r.t. $\{\mathbf{W}, \mathbf{B}\}$. Algorithm 1 sunmmarizes this procedure for clarity. For details of the two optimization subprocedures, see Sections 3 and 4, respectively. After estimating the parameters, we train the learning machine by minimizing the weighted loss (2) on $(\mathbf{x}^{new}, \mathbf{y}^{tr})$.

Algorithm 1 Estimating weights $\boldsymbol{\beta}^*$, $\mathbf{W}$, and $\mathbf{B}$ under LS-GeTarS

Input: training data $(\mathbf{x}^{tr}, \mathbf{y}^{tr})$ and test data $\mathbf{x}^{te}$
Output: weights $\boldsymbol{\beta}$ and $\mathbf{x}^{new}$ corresponding to the training data points
$\boldsymbol{\beta} \leftarrow \mathbf{1}_m$, $\mathbf{W} \leftarrow \mathbf{1}_m\mathbf{1}_d^\intercal$, $\mathbf{B} \leftarrow \mathbf{0}$
repeat
  fix $\mathbf{W}$ and $\mathbf{B}$ and estimate $\boldsymbol{\beta}$ by minimizing (12) with QP, under the constraint on $\boldsymbol{\beta}$ given in Sec. 3;
  fix $\boldsymbol{\beta}$ and estimate $\mathbf{W}$ and $\mathbf{B}$ by minimizing (12) with SCG;
until convergence
$\boldsymbol{\beta}^* \leftarrow \boldsymbol{\beta}$, $\mathbf{x}^{new} = \mathbf{x}^{tr} \odot \mathbf{W} + \mathbf{B}$.

S7. Determination of Hyperparameters

As discussed in Sec. 2, all hyperparameters in the subsequent learning machines reweighted SVM and KRR are selected by importance weighted cross-validation (Sugiyama et al., 2007). In addition, there are three types of hyperparameters. One is the kernel width of X to construct the kernel matrix K. In our experiments we normalize all variables in X to unit variance, and use some empirical values for those kernel widths: they are set to $0.8\sqrt{d}$ if the sample size $m \leq 200$, to $0.3\sqrt{d}$ if $m > 1200$, or to $0.5\sqrt{d}$ otherwise, where d is the dimensionality of X. This simple setting always works well in all our experiments; for a more principled strategy, one might refer to Gretton et al. (2012).

The second type of hyperparameters are involved in the parameterization of β for regression under TarS (the kernel width for L_β and regularization parameter λ_β) and λ_{LS} for LS-GeTarS in (12). We set these parameters by cross-validation. (On some large data sets we simply set λ_{LS} to 0.001 to save computational load.) Although the objective functions (Eq. 5 for TarS, and Eq. 11 for LS-GeTarS) is the sum of squared errors, the corresponding problems are considered unsupervised, or in particular, as density estimation problems, rather than supervised. We treat P_X^{new} as the distribution given by the model, and $\mathbf{x}^{te}$ as the corresponding observed data points. They are different from the classical density estimation problem in that here we use the maximum mean discrepancy between P_X^{new} and P_X^{te} as the loss function. We divide $\mathbf{x}^{te}$ into five equal size subsamples, use four of them to estimate β or

W and **B**, and the remaining one for testing. Finally we find the values of these hyperparameters that give the smallest cross-validated loss, which is (5) for regression under TarS or (11) for LS-GeTarS. The last type of hyperparameters, including hyperparameters in L and the regularization parameter λ, are learned by the extension of Gaussian process regression in the multi-output case (Zhang et al., 2011).

S8. Details of Simulation Settings in Sec. 6

The four simulation settings are

(a) a nonlinear regression problem $X = Y + 3\tanh(Y) + E$, where $E \sim \mathcal{N}(0, 1.5^2)$; $Y^{tr} \sim N(0, 2^2)$, and $Y^{te} \sim 0.8\mathcal{N}(1,1) + 0.2\mathcal{N}(0.2, 0.5^2)$,

(b) a classification problem under TarS, where $X|_{Y=-1} \sim \mathcal{N}\Big(\mathbf{0}, \begin{bmatrix} 0.21 & 0.09 \\ 0.09 & 0.21 \end{bmatrix}\Big)$, $X|_{Y=1} \sim \mathcal{N}\Big(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.31 & -0.06 \\ -0.06 & 0.31 \end{bmatrix}\Big)$, $P_Y^{tr}(y=-1) = 0.6$, and $P_Y^{te}(y=-1) = 0.2$,

(c) a classification problem approximately following location-scale GeTarS, where $X^{tr}|_{Y^{tr}=-1} \sim \mathcal{N}\Big(\mathbf{0}, \begin{bmatrix} 0.24 & 0.22 \\ 0.22 & 0.24 \end{bmatrix}\Big)$, $X^{tr}|_{Y^{tr}=1} \sim \mathcal{N}\Big(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.16 & -0.03 \\ -0.03 & 0.16 \end{bmatrix}\Big)$, $X^{te}|_{Y^{te}=-1} \sim \mathcal{N}\Big(\begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.12 & 0.11 \\ 0.11 & 0.12 \end{bmatrix}\Big)$, $X^{te}|_{Y^{te}=1} \sim \mathcal{N}\Big(\begin{bmatrix} 2 \\ 1.3 \end{bmatrix}, \begin{bmatrix} 0.27 & -0.04 \\ -0.04 & 0.27 \end{bmatrix}\Big)$, $P_Y^{tr}(y=-1) = 0.6$, and $P_Y^{te}(y=-1) = 0.3$, and

(d) a classification problem under non-location-scale GeTarS with slight change in the conditional, where $X^{tr}|_{Y^{tr}=-1} \sim \mathcal{N}\Big(\mathbf{0}, \begin{bmatrix} 0.16 & 0 \\ 0 & 0.16 \end{bmatrix}\Big)$, $X^{tr}|_{Y^{tr}=1} \sim \mathcal{N}\Big(\begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix}, \begin{bmatrix} 0.23 & 0 \\ 0 & 0.23 \end{bmatrix}\Big)$, $X^{te}|_{Y^{te}=-1} \sim \mathcal{N}\Big(\begin{bmatrix} -0.1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.10 & -0.03 \\ -0.03 & 0.10 \end{bmatrix}\Big)$, $X^{te}|_{Y^{te}=1} \sim \mathcal{N}\Big(\begin{bmatrix} 0.9 \\ 0.8 \end{bmatrix}, \begin{bmatrix} 0.11 & 0.05 \\ 0.05 & 0.11 \end{bmatrix}\Big)$, $P_Y^{tr}(y=-1) = 0.6$, and $P_Y^{te}(y=-1) = 0.2$.

S9. Results on Pseudo Real-world Data Sets in Sec. 7

Table 4 reports the results on pseudo real-world data sets. In these experiments, we split each data set into training set and test set. The percentage of training samples ranges from 60% to 80%. Then, we perform the biased sampling on the training data to obtain the shifted training set. Letting $P(s=1|y)$ be the probability of sample x being selected given that the its true output value is y, we consider the following two biased sampling schemes for selecting training data: (1) **Weighted Label** uses $P(s=1|y) = \exp(a+by)/(1+\exp(a+by))$ denoted by `label(a,b)`, and (2) **PCA** In this case, we generate biased sampling schemes over the features. Firstly, a kernel PCA is performed on the data. We select the first principal component and the corresponding projection values. The biased sampling scheme is then a normal distribution with mean $m + (\bar{m} - m)/a$ and variance $(\bar{m} - m)/b$ where m and $\bar{m}$ are the minimum value of the projection and the mean of the projection, respectively. We denote this sampling scheme by `PCA(a,b,`σ`)`, where σ is the bandwidth of the Gaussian RBF kernel. In summary, the LS-GeTarS outperforms Unweight, CovS, and TarS on 5 out of 6 data sets for classification problem. The TarS outperforms all other approaches on one of these data sets. For regression problem, TarS outperforms the Unweight and Covs on 7 out of 12 data sets.

S10. Details of Remote Sensing Image Classification

Hyperspectral remote sensing images are characterized by a dense sampling of the spectral signature of different land-cover types. We used a benchmark data set in the literature which consists of data acquired by the Hyperion sensor of the Earth Observing 1 (EO-1) satellite in an area of the Okavango Delta, Botswana, with 145 features; for details of this data set, see (Ham et al., 2005). The labeled reference samples were collected on two different and spatially disjoint areas (Area 1 and Area 2), thus representing possible spatial variabilities of the spectral signatures of classes. The samples taken on each area were partitioned into a training set TR and a test set TS by random sampling. The numbers of labeled reference samples for each set and class are reported in Table 5. TR_1, TS_1, TR_2, and TS_2 have sample sizes 1242, 1252, 2621, and 627, respectively. One would expect that not only the prior probabilities of the classes Y, but also the conditional distribution of X given Y would change across them, due to physical factors related to ground (e.g., different soil moisture or composition), vegetation, and atmospheric conditions. Our target is to do domain adaptation from TR_1 to TS_2 and from TR_2 to TS_1.

Table 4. The results of different distribution shift correction schemes. The results are averaged over 10 trials for regression problems (marked *) and 30 trials for classification problems. We report the normalized mean squared error (NMSE) for regression problem and test error for classification problem.

Data Set	**Sampling Scheme**	**NMSE/test error ± std. error**			
		Unweight	**CovS**	**TarS**	**LS-GeTarS**
1. Abalone*	label(1,10)	0.4447±0.0223	0.4497±0.0125	0.4430±0.0208	–
2. CA Housing*	PCA(10,5,0.1)	0.4075±0.0298	0.3944±0.0346	0.4565±0.0422	–
3. Delta Alilerons (1)*	label(1,10)	0.3120±0.0040	0.3408±0.0278	0.3451±0.0280	–
4. Ailerons*	PCA(1e3,4,0.1)	0.1360±0.0350	0.1486±0.0264	0.1329±0.0174	–
5. haberman (1)	label(0.2,0.8)	0.2699±0.0304	0.2699±0.0315	0.2676±0.0287	0.2619±0.0352
6. Bank8FM*	PCA(3,6,0.1)	0.0477±0.0014	0.0590±0.0117	0.0452±0.0070	–
7. Bank32nh*	PCA(3,6,0.01)	0.5210±0.0318	0.5171±0.0131	0.5483±0.0455	–
8. cpu-act*	PCA(4,2,1e-12)	0.2026±0.0382	0.2042±0.0316	0.2000±0.0474	–
9. cpu-small*	PCA(4,2,1e-12)	0.1314±0.0347	0.2009±0.0849	0.0769±0.0100	–
10. Delta Ailerons(2)*	PCA(1e3,4,0.1)	0.4496±0.0236	0.3373±0.0596	0.3258±0.0274	–
11. Boston House*	PCA(2,4,1e-4)	0.5128±0.1269	0.4966±0.0970	0.5342±0.0777	–
12. kin8nm*	PCA(8,5,0.1)	0.5382±0.0425	0.5266±0.1248	0.6079±0.0976	–
13. puma8nh*	PCA(4,4,0.1)	0.6093±0.0629	0.5894±0.0361	0.5595±0.0297	–
14. haberman(2)	PCA(2,2,0.01)	0.2736±0.0374	0.2725±0.0422	0.2724±0.0367	0.2579±0.0241
15. Breast Cancer	label(0.3,0.7)	0.2699±0.0304	0.3196±0.1468	0.2670±0.0319	0.2609±0.0510
16. India Diabetes	label(0.3,0.7)	0.2742±0.0268	0.2797±0.0354	0.2846±0.0364	0.2700±0.0599
17. Ionosphere	label(0.3,0.7)	0.0865±0.0294	0.1079±0.0563	0.0846±0.0559	0.0938±0.0294
18. German Credit	label(0.2,0.8)	0.3000±0.0284	0.2802±0.0354	0.2846±0.0364	0.2596±0.0368

Table 5. Number of training (TR_1 and TR_2) and test (TS_1 and TS_2) patterns acquired in the two spatially disjoint areas for the experiment on remote sensing image classification.

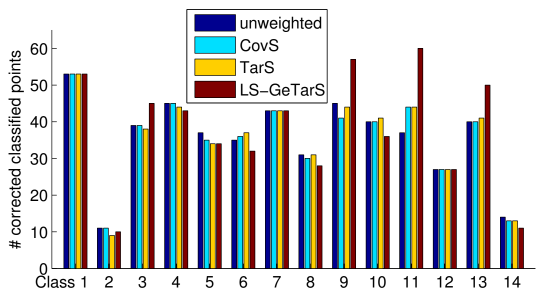
Class	Number of patterns			
	Area 1		Area 2	
	TR_1	TS_1	TR_2	TS_2
Water	69	57	213	57
Hippo grass	81	81	83	18
Floodplain grasses1	83	75	199	52
Floodplain grasses2	74	91	169	46
Reeds1	80	88	219	50
Riparian	102	109	221	48
Firescar2	93	83	215	44
Island interior	77	77	166	37
Acacia woodlands	84	67	253	61
Acacia shrublands	101	89	202	46
Acacia grasslands	184	174	243	62
Short mopane	68	85	154	27
Mixed mopane	105	128	203	65
Exposed soil	41	48	81	14
Total	1242	1252	2621	627

After estimating the weights and/or the transformed training points, we applied the multi-class classifier with a RBF kernel, provided by LIBSVM, on the weighted or transformed data. Each time, the kernel size and parameter C were chosen by five-fold cross-validation over the sets $\{2^{5/2}, 2^{3/2}, 2^{1/2}, 2^{-1/2}, 2^{-3/2}, 2^{-5/2}\} \cdot \sqrt{d}$ and $\{2^6, 2^8, 2^{10}, 2^{12}, 2^{14}, 2^{16}, 2^{18}\}$, respectively. (We found that the selected values always belonged to the interior of the sets.)

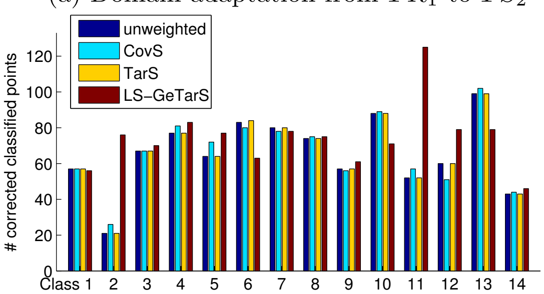
Table 3 shows the overall classification error (i.e., the fraction of misclassified points) obtained by different approaches for each domain adaptation problem. We can see that in this experiment, correction for target shift does not significantly improve the performance; in fact, the β values for most classes are rather close to one. However, correction for conditional shift with LS-GeTarS reduces the overall classification error from 20.73% to 11.96% for domain adaptation from TR_1 to TS_2, and from 25.32% to 13.56% for that from TR_2 to TS_1. Covariate shift helps slightly for $TR_2 \rightarrow TS_1$, probably because our classifier is rather simple in that all dimensions have the same kernel size.

Correction for conditional shift with LS-GeTarS reduces the overall classification error (fraction of misclassified points), as seen from Table 3. In addition to the overall classification error, we also report the number of correctly classified points from each class;

see Fig. 7. One can see that for both domain adaptation problems, LS-GeTarS improves the classification accuracy on classes 11, 9, and 3. It also leads to significant improvement on class 13 for the problem $TR_1 \rightarrow TS_2$, and on class 2 for $TR_1 \rightarrow TS_2$. Note that this is a multi-class classification problem and we aim to improve the overall classification accuracy; to achieve that, the accuracy on some particular classes, such as classes 10 and 6, could be worse. Fig. 8 plots some of the estimated scale transformation coefficients $\mathbf{w}(y^{tr})$ and location transformations $\mathbf{b}(y^{tr})$ that are significant (i.e., $\mathbf{w}(y^{tr})$ is significantly different from one, and $\mathbf{b}(y^{tr})$ different from zero). One can see that roughtly speaking, the transformation learned for the domain adaptation problem $TR_2 \rightarrow TS_1$ is the inverse of that for the problem $TR_1 \rightarrow TS_1$.



(a) Domain adaptation from TR_1 to TS_2

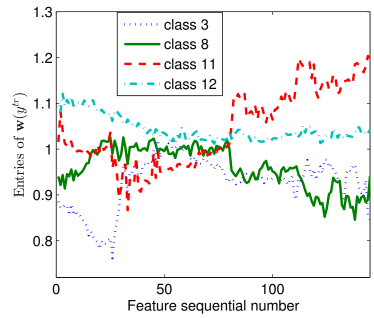


(a) Domain adaptation from TR_2 to TS_1

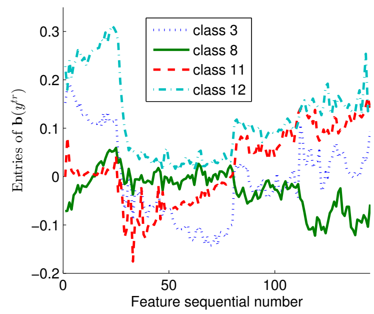
Figure 7. The number of correctly classified data points for each class and each approach. (a) $TR1$ as training set and $TS2$ as test set. (b) $TR2$ as training set and TS_1 as test set.

S11. Experiment on TRECVID Concept Detection

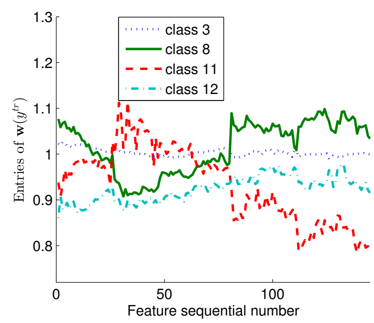
In this experiment, we consider automatic assignment of semantic tags to video segments, which can be a fundamental technology for content-based video search (Smeaton et al., 2009). For each semantic concept, classifiers can be obtained from annotated training data (source domain) and used to determine the presence of the concept for each segment in test data



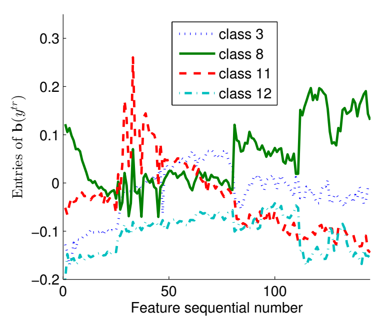
(a) Estimated scale transformation coefficient for selected classes for domain adaptation $TR_1 \rightarrow TS_2$.



(b) Estimated location transformation for selected classes for domain adaptation $TR_1 \rightarrow TS_2$.



(c) Estimated scale transformation coefficient for selected classes for domain adaptation $TR_2 \rightarrow TS_1$.



(d) Estimated location transformation for selected classes for domain adaptation $TR_2 \rightarrow TS_1$.

Figure 8. Estimated scale transformation coefficient $\mathbf{w}(y^{tr})$ and location transformation $\mathbf{b}(y^{tr})$ for selected classes by correction for LS-GeTarS. (a, b) For domain adaptation from TR_1 to TS_2. (c, d) For domain adaptation from TR_2 to TS_1.

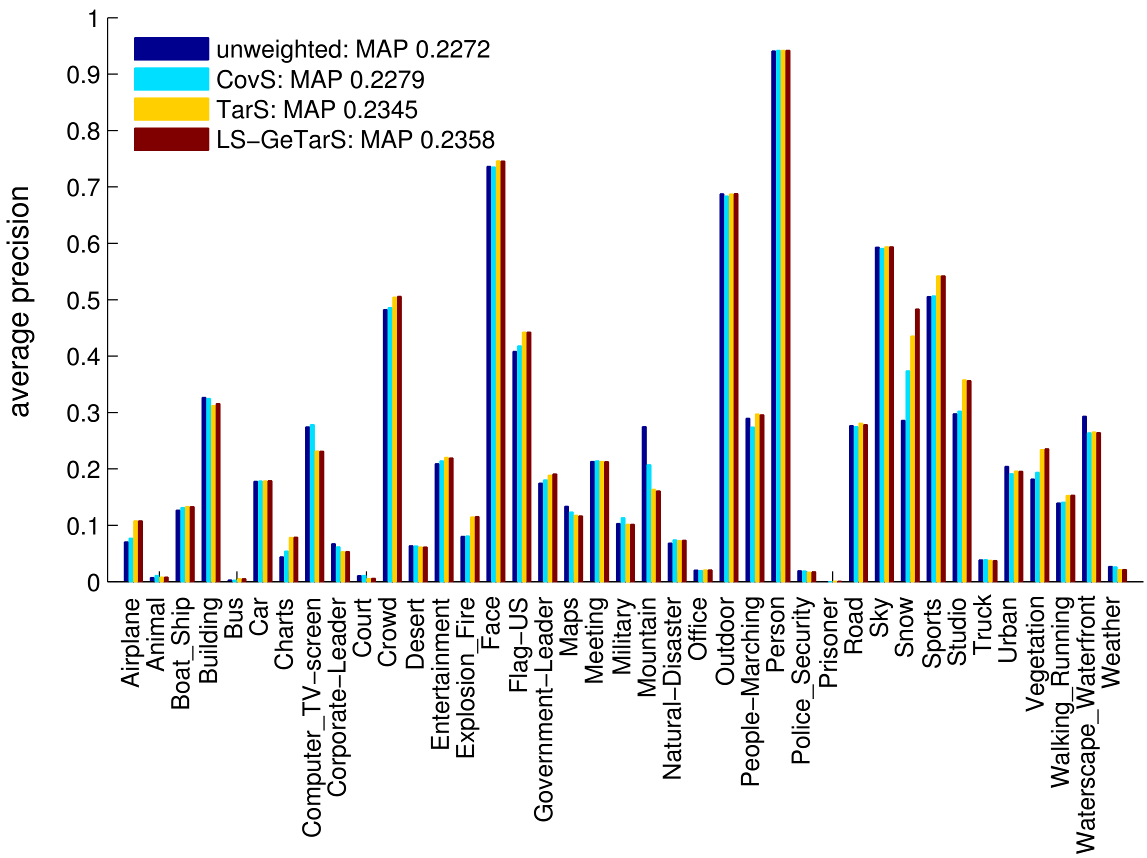


Figure 9. The performance of the baseline, CovS, TarS, and LS-GeTarS on all concepts.

(target domain). We show that the proposed TarS and LS-GeTarS can improve the performance of concept detection when the training and test data are from different domains, for example different TV channels.

We consider the 39 semantic concepts from the LSCOM-lite lexicon (Naphade et al., 2005), with annotation on the TRECVID 2005 data set. The data set contains $61,901$ segmented video shots from 108 hours of television programmes from six different broadcast channels, including three English channels (CNN, MSNBC and NBC), two Chinese channels (CCTV and NTDTV) and one Arabic channel (LBC). For each shot, 346 low-level features were extracted from its keyframe (Yang et al., 2007), including Grid Color Moment (225 dim.), Gabor Texture (48 dim.), and Edge Detection Histogram (73 dim.). We split the data set into a source domain that consists of video shots from the English and Chinese channels, and a target domain that contains shots from the Arabic channel.

We apply asymmetric bagging to handle the scarcity of positive training instances (Tao et al., 2006). For each concept, five SVM classifiers were trained using up to 1000 positive training instances and the randomly sampled same amount of negative instances. The overall rank list on the test data was obtained from the average classification confidence. We used the default parameters for training the SVM classifiers, as suggested by Tao et al. (2006)

The average precision of all concepts is shown in Fig. 9. Overall, TarS achieved a Mean Average Precision (MAP) of 0.2345 across all concepts, and outperformed the baseline method (MAP: 0.2272). TarS achieved substantial improvements on concepts such as *Snow*, *Vegetation*, and *Flag-US*, where P_Y varies significantly. LS-GeTarS further improved the performance and achieved an MAP of 0.2358. As shown in Fig. 9, LS-GeTarS worked particularly well for the concept *Snow*, where considerable conditional shift is expected. Note that our methods should be distinguished from previous work by Duan et al. (2009), as we do not use any annotation from the target domain.

References

Duan, L., Tsang, I. W., Xu, D., and Chua, T. S. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML*, 2009.

Gretton, A., Sriperumbudur, B., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., and Fukumizu, K. Optimal kernel choice for large-scale two-sample tests. In *NIPS 25*. 2012.

Naphade, M. R., Kennedy, L., Kender, J. R., Chang, S. F., Smith, J. R., Over, P., and Hauptmann, A. *A Light Scale Concept Ontology for Multimedia Understanding for TRECVID 2005*, 2005. IBM Research Technical Report.

Smeaton, A. F., Over, P., and Kraaij W. High-Level Feature Detection from Video in TRECVid: A 5-Year Retrospective of Achievements. In Divakaran A. (eds.), *Multimedia Content Analysis, Theory and Applications*, pp. 151–174. Springer Verlag, Berlin, 2009.

Sugiyama, M., Krauledat, M., and Müller, K. R. Covariate shift adaptation by importance weighted cross validation. *JMLR*, 8:985–1005, December 2007.

Tao, D., Tang, X., Li, X., and Wu, X. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE T-PAMI*, 28(7):1088–1099, 2006.

Yang, J., Yan, R., and Hauptmann, A. G. Cross-domain video concept detection using adaptive svms. In *ACM MM*, 2007.