# Supplementary Material: Transductive Learning Via Improved Geodesic Sampling

Youshan Zhang
Sihong Xie
Brian D. Davison
{yoz217,six316,bdd3}@lehigh.edu

Computer Science and Engineering
Lehigh University
Bethlehem, PA, USA

## 1 Choosing a Manifold

Table 1 shows all tasks using Office+ Caltech 10 with SURF feature. We can easily find that spherical manifold has the highest accuracy with less computation time. The computation time of Kendall's manifold is close to the time of spherical manifold since only Log map requires the singular value decomposition; while both Log map and Exp map require SVD on Grassmannian manifold. Hence, classification using Grassmannian manifold will takes longer computation time. Therefore, the spherical manifold appears best for classification.

Table 1: Accuracy and time of three manifolds using SURF feature

| Task | Sphere accuracy | Sphere time | Kendall accuracy | Kendall time | Grassmannian accuracy | Grassmannian time |
|---|---|---|---|---|---|---|
| C $\longrightarrow$ A | **55.6%** | **3.3s** | 55.4% | 3.6s | **55.6%** | 10.6s |
| C $\longrightarrow$ W | **49.8%** | **2.8s** | 48.1% | 3.0s | **49.8%** | 5.4s |
| C $\longrightarrow$ D | 49.6% | **3.0s** | **50.3%** | 3.1s | 49.7% | 5.3s |
| A $\longrightarrow$ C | **40.8%** | **1.8s** | **40.8%** | 1.9s | **40.8%** | 4.5s |
| A $\longrightarrow$ W | 34.6% | **1.3s** | **38.6%** | 1.4s | 33.2% | 4.1s |
| A $\longrightarrow$ D | **34.4%** | **1.3s** | 32.5% | 1.4s | 34.3% | 4.1s |
| W $\longrightarrow$ C | **26.9%** | **0.7s** | 26.0% | 0.8s | **26.9%** | 3.8s |
| W $\longrightarrow$ A | **35.6%** | **0.78s** | 25.8% | 0.8s | 27.7% | 3.6s |
| W $\longrightarrow$ D | **74.5%** | **0.6s** | **74.5%** | 0.7s | **74.5%** | 3.4s |
| D $\longrightarrow$ C | **30.0%** | **0.7s** | 29.5% | 0.7s | **30.0%** | 3.6s |
| D $\longrightarrow$ A | 32.9% | **0.6s** | **33.1%** | 0.8s | 32.9% | 3.5s |
| D $\longrightarrow$ W | 70.5% | **0.5s** | **70.8%** | 0.6s | 70.5% | 3.6s |
| Average | **44.6%** | **1.4s** | 43.8% | 1.6s | 43.8% | 4.6s |

## 2 GSM with Manifold Embedded Distribution Alignment (MEDA)

It has three fundamental steps: 1) learn features from the manifold based on Gong et al. [1]; in our case, the features will be generated from our GSM method (from Alg. 1); 2) MEDA

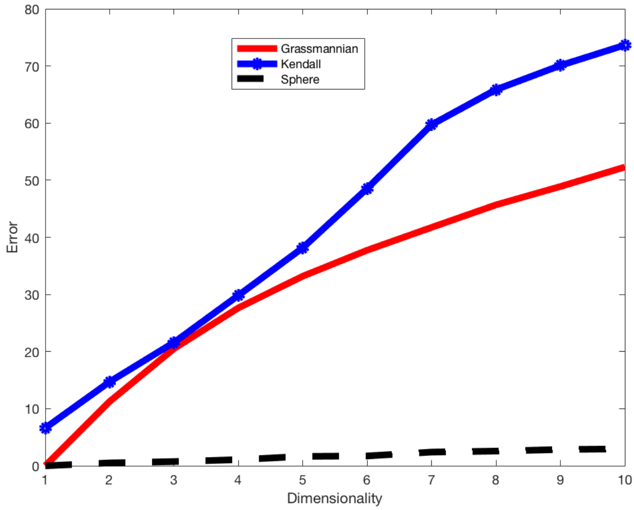


Figure 1: The error of three manifolds as column dimensionality grows.

uses dynamic distribution alignment to estimate the marginal and conditional distributions of data; and, 3) construct a new classifier from previous steps. Please refer to Wang et al. [2] for more details. The classifier ($fr$) is defined as:

$$\begin{aligned} fr = &\underset{\in \sum_{i=1}^{n} \mathcal{H}_k}{\arg\min}\, l(fr(X_{k_i}), y_i) + \eta ||fr||_K^2 + \\ &\lambda \overline{D_{fr}}(\mathcal{D}_s, \mathcal{D}_t) + \rho R_{fr}(\mathcal{D}_s, \mathcal{D}_t) \end{aligned} \tag{1}$$

where $X_k$ is the learned features from GSM, $||fr||_2^K$ is the squared norm of $fr$; $\overline{D_{fr}}(\cdot,\cdot)$ represents the dynamic distribution alignment; $R_{fr}(\cdot,\cdot)$ is a Laplacian regularization; $\eta, \lambda$, and $\rho$ are regularization parameters. By training the classifier from Eq. 1, we can predict labels of test data. we then combine our GSM model with MEDA model as shown in Alg. 2.

**Algorithm 1** Principal Component Analysis for GSM

**Input:** $X_S'^{\top}$ with $(d \times N_1)$and $X_T'^{\top}$ with $(d \times N_2)$
**Output:** New projected matrix $X_S$ and $X_T$

$\mu_{S/T} = \frac{1}{N_1/\,N_2} \sum_{i=1}^{N_1/\,N_2} X_{S_i/\,T_i}'^{\top}$
**If** $d <= N_1/\,N_2$
  $S = \frac{1}{N} \sum_{i=1}^{N} (X_{S_i/\,T_i}'^{\top} - \mu_{S/T})(X_{S_i/\,T_i}'^{\top} - \mu_{S/T})^{\top}$
  $X_S'/X_T'$ = eigenvectors of S
**Else**
  $S = \frac{1}{N} \sum_{i=1}^{N} (X_{S_i/\,T_i}'^{\top} - \mu_{S/T})^{\top}(X_{S_i/\,T_i}'^{\top} - \mu_{S/T})$
  $Vec_S/Vec_T$ = eigenvectors of $S$
  $X_S'/X_T' = X_S^{\top}/X_T^{\top} \times Vec_S/Vec_T$
**End**

**Algorithm 2** Classification using GSM_MEDA

**Input:** $X'_S, Y_S, X'_T, Y_T$, Sample size: $N$

**Output:** Accuracy of $predict_{Y_T}$

1: Get $X_S$ and $X_T$ according to Alg. 1.

2: Sample $(S_t)$ $N$ times between $X_S$ and $X_T$

3: $New\_X_S = X'_S \times [S_0, \cdots, S_t, \cdots, S_1]$
$New\_X_T = X'_T \times [S_0, \cdots, S_t, \cdots, S_1]$

4: Train MEDA model ($fr$ classifier) using $New\_X_S$ and $Y_S$, then predict the labels of $New\_X_T$ using trained $fr$ classifier, and calculate the accuracy of $predict_{Y_T}$.

where $Y_S$ is the vector of labels of $X'_S$, and $Y_T$ is the vector of labels of $X_T$.

# References

[1] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073. IEEE, 2012.

[2] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S. Yu. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 402–410, 2018. doi: 10.1145/3240508.3240512.