Transfer Metric Learning by Learning Task Relationships

Yu Zhang and Dit-Yan Yeung

Department of Computer Science and Engineering
Hong Kong University of Science and Technology

KDD 2010

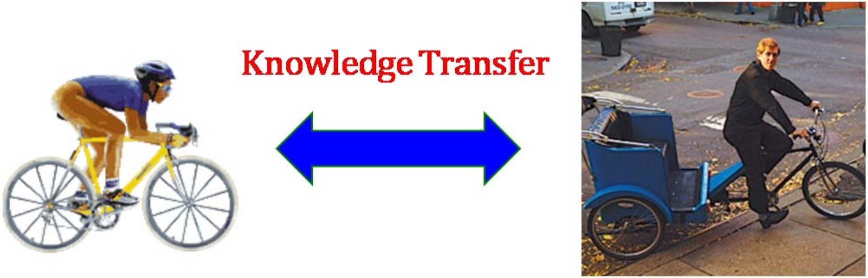
Outline

Metric Learning and Its Limitations

- Distance metric plays a very crucial role in many data mining algorithms.
  - k-means clustering, k-nearest neighbor classifier,...
- Its limitation:
  - With only limited labeled data, the metric learned is often unsatisfactory.
- Solutions:
  - Semi-Supervised Metric Learning
    - Utilize information in unlabeled data
  - Transfer Metric Learning
    - Utilize information in other related tasks

Transfer Learning

- Transfer learning is to improve the performance of the target task with the help of some source tasks.



Transfer Metric Learning

- There is only one work on transfer metric learning
  - [ref]: Robust distance metric learning with auxiliary knowledge, IJCAI'09.
- Some Limitations:
  - It only models positive task correlation
  - The optimization problem is non-convex
- Our Contributions:
  - Propose a convex formulation for transfer metric learning
  - Model the pairwise task relationships under the regularization framework
    - Positive task correlation
    - Negative task correlation
    - Task unrelatedness

Outline

Notations

- m learning tasks $\{T_i\}_{i=1}^m$
- The training set $\mathcal{D}_i$ in T_i consists of n_i data points $(\mathbf{x}_j^i, y_j^i)$, $j = 1, \ldots, n_i$
- $\mathbf{x}_j^i \in \mathbb{R}^d$ and its corresponding class label $y_j^i \in \{1, \ldots, C_i\}$.

The Objective Function

- The optimization problem for multi-task metric learning is formulated as follows:

$$\min_{\{\mathbf{\Sigma}_i\},\mathbf{\Omega}} \sum_{i=1}^{m} \frac{2}{n_i(n_i-1)} \sum_{j<k} g\left(y^i_{j,k}\left[1-\|\mathbf{x}^i_j-\mathbf{x}^i_k\|^2_{\mathbf{\Sigma}_i}\right]\right) + \frac{\lambda_1}{2}\sum_{i=1}^{m}\|\mathbf{\Sigma}_i\|_F^2 + \frac{\lambda_2}{2}\mathrm{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T)$$

$$\begin{aligned} \text{s.t.}\ & \mathbf{\Sigma}_i \succeq \mathbf{0}\ \forall i \\ & \tilde{\mathbf{\Sigma}} = (\mathrm{vec}(\mathbf{\Sigma}_1), \ldots, \mathrm{vec}(\mathbf{\Sigma}_m)) \\ & \mathbf{\Omega} \succeq 0,\ \mathrm{tr}(\mathbf{\Omega}) = 1. \end{aligned}$$

- From the probabilistic viewpoint, this is related to MAP solution of a probabilistic model where the prior on the metrics of all tasks is matrix-variate normal distribution.
- It has been proved that the optimization problem is a convex optimization problem.
- We propose an alternating method to solve the problem efficiently.

Outline

The Assumption

- Suppose we are given $m-1$ source tasks $\{T_i\}_{i=1}^{m-1}$ and one target task T_m.
- Each source task has enough labeled data and can learn an accurate model with no need to seek help from the other source tasks.
- We assume that the metric matrix $\mathbf{\Sigma}_i$ for the ith source task has been learned independently.

The Objective Function

- Based on multi-task metric learning, we formulate the optimization problem as follows:

$$
\begin{aligned}
&\min_{\boldsymbol{\Sigma}_m,\boldsymbol{\Omega}} \frac{2}{n_m(n_m-1)} \sum_{j<k} g\left(y_{j,k}^m\left[1-\|\mathbf{x}_j^m-\mathbf{x}_k^m\|_{\boldsymbol{\Sigma}_m}^2\right]\right)+\frac{\lambda_1}{2}\|\boldsymbol{\Sigma}_m\|_F^2+\frac{\lambda_2}{2}\mathrm{tr}(\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Omega}^{-1}\tilde{\boldsymbol{\Sigma}}^T) \\
&\text{s.t.} \boldsymbol{\Sigma}_m \succeq \mathbf{0} \\
&\quad \tilde{\boldsymbol{\Sigma}} = (\mathrm{vec}(\boldsymbol{\Sigma}_1),\ldots,\mathrm{vec}(\boldsymbol{\Sigma}_{m-1}),\mathrm{vec}(\boldsymbol{\Sigma}_m)) \\
&\quad \boldsymbol{\Omega} \succeq 0,\ \mathrm{tr}(\boldsymbol{\Omega})=1.
\end{aligned}
$$

- Since we assume that the source tasks are independent and of equal importance, we can express $\boldsymbol{\Omega}$ as

$$
\boldsymbol{\Omega} = \begin{pmatrix} \frac{1-\omega}{m-1}\mathbf{I}_{m-1} & \boldsymbol{\omega}_m \\ \boldsymbol{\omega}_m^T & \omega \end{pmatrix}.
$$

Optimization Procedure

- It can be proved that the problem is jointly convex with respect to all variables: $\boldsymbol{\Sigma}_m$, $\boldsymbol{\omega}_m$ and ω.
- However, it is not easy to optimize it with respect to all the variables simultaneously.
- We still use an alternating method to solve it.

Optimization Procedure - Optimizing w.r.t. $\mathbf{\Sigma}_m$

- The optimization problem with respect to $\mathbf{\Sigma}_m$ is formulated as

$$\begin{aligned} \min_{\mathbf{\Sigma}_m} \quad & \frac{2}{n_m(n_m-1)} \sum_{j<k} g\left(y_{j,k}^m \left[1 - \| \mathbf{x}_j^m - \mathbf{x}_k^m \|_{\mathbf{\Sigma}_m}^2 \right] \right) \\ & + \frac{\lambda_1'}{2} \| \mathbf{\Sigma}_m \|_F^2 - \lambda_2' \mathrm{tr}(\mathbf{\Sigma}_m^T \mathbf{M}) \\ \text{s.t.} \quad & \mathbf{\Sigma}_m \succeq \mathbf{0}. \end{aligned}$$

- Similar to regularized distance metric learning method, we use an online algorithm to solve this problem.

Optimization Procedure - Online Algorithm

Input: labeled data $(\mathbf{x}_j^m, y_j^m)$ $(j = 1, \ldots, n_m)$, matrix $\mathbf{M}$, λ_1', λ_2' and predefined learning rate η

Initialize $\boldsymbol{\Sigma}_m^{(0)} = \frac{\lambda_2'}{\lambda_1'}\mathbf{M}$;

for $t = 1, \ldots, T_{max}$ **do**

Receive a pair of training data points $\{(\mathbf{x}_j^m, y_j^m), (\mathbf{x}_k^m, y_k^m)\}$;

Compute y: $y = 1$ if $y_j^m = y_k^m$, and $y = -1$ otherwise;

if the training pair $(\mathbf{x}_j^m, \mathbf{x}_k^m, y)$ is classified correctly, i.e., $y(1 - \|\mathbf{x}_j^m - \mathbf{x}_k^m\|^2_{\boldsymbol{\Sigma}_m^{(t-1)}}) > 0$ **then**

$\boldsymbol{\Sigma}_m^{(t)} = \boldsymbol{\Sigma}_m^{(t-1)}$;

else if $y == -1$

$\boldsymbol{\Sigma}_m^{(t)} = \boldsymbol{\Sigma}_m^{(t-1)} + \eta(\mathbf{x}_j^m - \mathbf{x}_k^m)(\mathbf{x}_j^m - \mathbf{x}_k^m)^T$;

else

$\boldsymbol{\Sigma}_m^{(t)} = \pi_{S_+}\left(\boldsymbol{\Sigma}_m^{(t-1)} - \eta(\mathbf{x}_j^m - \mathbf{x}_k^m)(\mathbf{x}_j^m - \mathbf{x}_k^m)^T\right)$ where $\pi_{S_+}(\mathbf{A})$ projects matrix $\mathbf{A}$ into the positive semidefinite cone;

end if

end for

Output: metric $\boldsymbol{\Sigma}_m^{(T_{max})}$

Optimization Procedure - Optimizing w.r.t. $\boldsymbol{\omega}_m$ and ω

- The optimization problem with respect to $\boldsymbol{\omega}_m$ and ω is formulated as

$$\begin{aligned} \min_{\boldsymbol{\omega}_m,\omega,\boldsymbol{\Omega}} \quad & \mathrm{tr}(\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Omega}^{-1}\tilde{\boldsymbol{\Sigma}}^T) \\ \text{s.t.} \quad & \boldsymbol{\Omega} = \begin{pmatrix} \frac{1-\omega}{m-1}\mathbf{I}_{m-1} & \boldsymbol{\omega}_m \\ \boldsymbol{\omega}_m^T & \omega \end{pmatrix} \\ & \omega(1-\omega) \geq (m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m. \end{aligned}$$

- Then we can reformulate it as a second-order cone programming problem:

$$\begin{aligned} \min_{\boldsymbol{\omega}_m,\omega,\mathbf{f},t,\{h_j\},\{r_j\}} \quad & -t \\ \text{s.t.} \quad & \frac{1-\omega}{m-1} \geq t\lambda_1,\ \mathbf{f} = \mathbf{U}^T(\boldsymbol{\omega}_m - t\boldsymbol{\Psi}_{12}) \\ & r_j = \frac{1-\omega}{m-1} - t\lambda_j,\ \left\| \begin{pmatrix} f_j \\ \frac{r_j-h_j}{2} \end{pmatrix} \right\|_2 \leq \frac{r_j+h_j}{2}\ \forall j \\ & \sum_{j=1}^{m-1} h_j \leq \omega - t\Psi_{22},\ \left\| \begin{pmatrix} \sqrt{m-1}\boldsymbol{\omega}_m \\ \frac{\omega-1}{2} \\ \omega \end{pmatrix} \right\|_2 \leq \frac{\omega+1}{2}. \end{aligned}$$

Outline

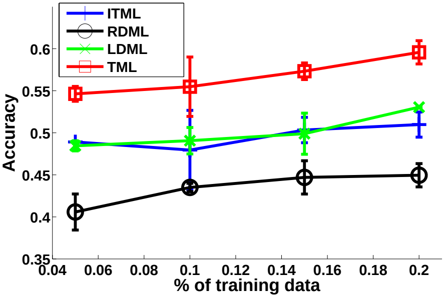
Experimental Setup

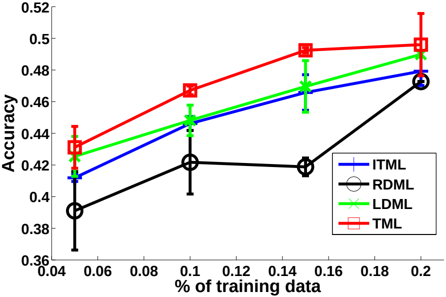
- Three baseline methods are compared:
  - Information-Theoretic Metric Learning (ITML)
    - [ref]: Information-theoretic metric learning, ICML'07.
  - Regularized distance metric learning (RDML)
    - [ref]: Regularized distance metric learning: Theory and algorithm, NIPS'09.
  - Existing transfer metric learning method - LDML
    - [ref]: Robust distance metric learning with auxiliary knowledge, IJCAI'09.
- CVX solver is used to solve the second-order cone programming problem.
- The learning rate η is set to be 0.01.

Wine Quality Classification

- This is to classify wine into different grades from 0 to 10.
- There are two tasks:
  - One for red wine classification
  - The other for white wine classification
- Each task is treated as the target task and the other task as the source task.
- To see the effect of varying the size of the training set, we vary the percentage of the training data used from 5% to 20%.
- Each configuration is repeated 10 times.

Wine Quality Classification (Cont'd)

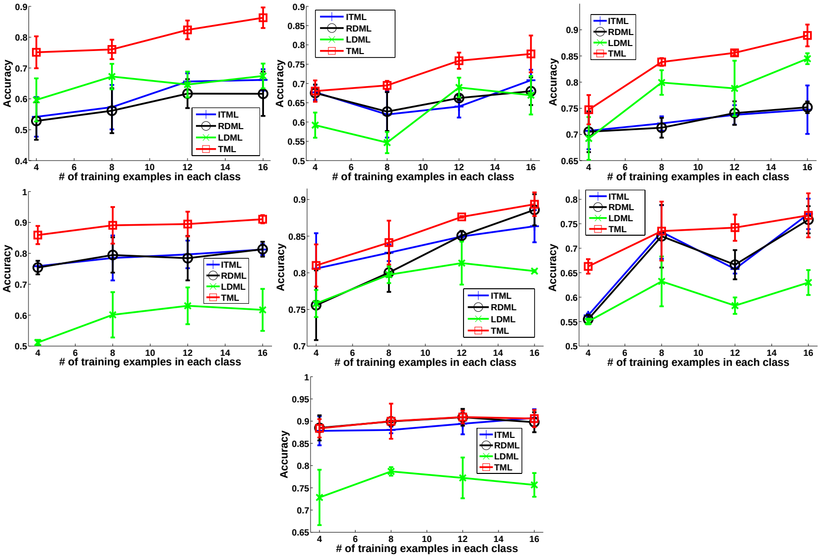




Handwritten Letter Classification

- The handwritten letter classification applicaton consists of seven tasks.
- Each task is a binary letter classification problem.
  - The corresponding letters for each task are: c/e, g/y, m/n, a/g, a/o, f/t and h/n.
- For each task, there are about 1000 positive and 1000 negative data points.

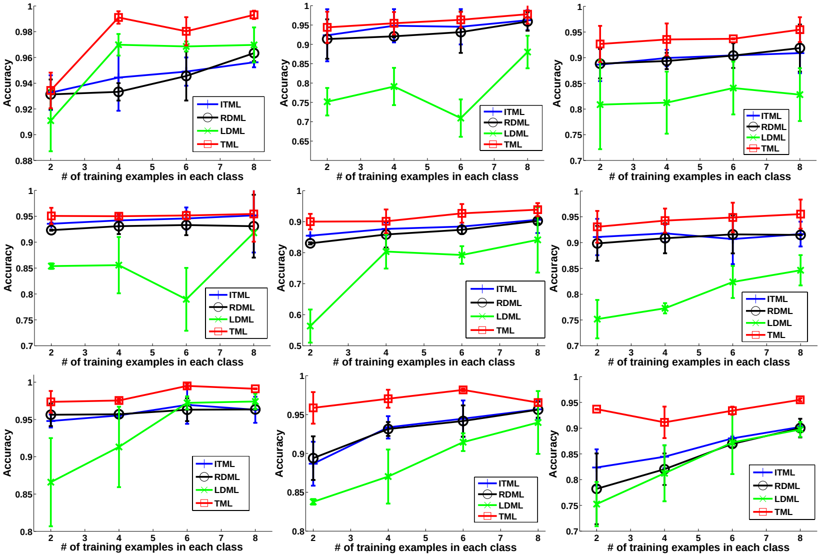
Handwritten Letter Classification (Cont'd)



USPS Digit Classification

- There are nine classification tasks.
  - Each task corresponding to the classification of two successive digits.
- The experimental settings are the same as those for handwritten letter classification.

USPS Digit Classification (Cont'd)



Outline

Conclusion

- We have proposed a transfer metric learning method to alleviate the labeled data deficiency problem in the target learning task by exploiting useful information from some source tasks.
- The learning of the distance metric in the target task and the relationships between the source tasks and the target task is formulated as a convex optimization problem.
- Future work:
    - We will extend our method to semi-supervised setting by exploiting useful information contained in the unlabeled data as well.

Thank you for your attention!