# Ranking with Semi-Supervised Distance Metric Learning and Its Application to Housing Potential Estimation*

Yangqiu Song[†§] Bin Zhang[§] Wenjun Yin[§] Changshui Zhang[‡] Jin Dong[§]
[†‡]State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing, China
[§]IBM China Research Lab, Beijing, China.
[†]songyq99@mails.tsinghua.edu.cn, [‡]zcs@mail.tsinghua.edu.cn
[§]{zbin,yinwenj,dongjin}@cn.ibm.com

## ABSTRACT

This paper proposes a semi-supervised distance metric learning algorithm for the ranking problem. Instead of giving the computer what are the important factors that affect the final rank value, we only give several most certainly ranked points which implicitly contain the knowledge of the ranking factors. Then the computer can automatically use the most certain points and plenty of unlabeded data to learn an informative metric for ranking. This metric not only can help to regress an order in the observed data, but also can be used to retrieve the data by querying new test points. Moreover, the lower-rank distance metric can be used to visualize high-dimensional data. We also present an application to the housing potential estimation problem. It is shown that the algorithm is efficient to help consultants to refine their consulting work.

## Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.5.4 [**Pattern Recognition**]: Applications; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

## General Terms

Algorithms, Experimentation

## Keywords

Metric Learning, Dimensionality Reduction, Ranking, Information Retrieval, Semi-Supervised Learning.

## 1. INTRODUCTION

Ranking is a common way to find an order of real world data. It is not only used in computer science society, but also used in other domains such as business and finance. Scoring the multi-variate examples by adding a weight to each feature is usually done in business consulting and analysis. However, it is simple and intuitive, and the results may be not good enough. Another method to get the weights of features is to learn them from data which contain implicit domain knowledge.

Learning weights of features is regarded as a kind of distance metric learning problem [9]. For ranking, learning an informative distance metric is helpful to reduce the computational complexity and improve the ranking accuracy. For high-dimensional data, it is also helpful to visualize them in a low-dimensional coordinates system. Many problems, such as text and content based multimedia retrieval, need to learn a good distance metric.

In general, we should utilize some prior assumption or domain knowledge to learn a distance metric. For example, for classification the domain knowledge is the class label. For ranking, we can not make the cluster assumption because there is no clear class boundary. Thus, other domain knowledge should be proposed to learn a more reasonable ranking metric. It is impossible to sort a large amount of data as an ordered sequence manually. Conversely, it is effortlessly for human to provide some examples as the most certainly "good" and "bad" points, which can be seen as a kind of domain knowledge. Based on the domain knowledge of most certain points, there are two jobs: (1) to obtain the order of the observed data points. (2) for new test point, to find what is its position in the whole sequence, or to find what are the most similar points in the already observed data set.

In this paper, we design an algorithm that can deal with this type of domain knowledge and can do the two types of ranking tasks. Both the human hints and the geometric information provided by observed data should be used. The essence of the proposed algorithm is a semi-supervised learning method. However, it is different from the start point of traditional semi-supervised classification and clustering [11]. Generally, we do not make the assumption of existence of classes and do not want to find the classification boundaries. Instead, we only assume that the data point cloud can construct a graph which describes the manifold structure, and there are multiple concepts on different parts of the manifold. By maximizing the distance between different con-

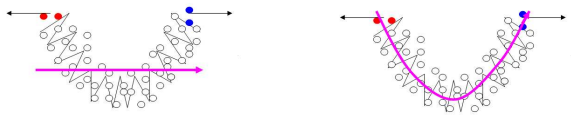
*This work was done when the first author visited IBM China Research Lab (CRL).


*CIKM'07,* November 6–8, 2007, Lisboa, Portugal.

cepts and simultaneously preserving the local structure on the manifold, the learned metric can indeed give good ranking results. Besides giving the test results on the benchmark data, we also show an application to business consulting.

## 2. MOTIVATION AND SOLUTION

The intuitive start is to both preserve intrinsic geometry of the data point cloud and use the information provided by user. We consider a toy problem of "U" shape which is shown in Fig. 1 (a) and (b).



(a) Ranking in Linear Space. (b) Ranking in Kernel Space.

**Figure 1: Toy example of metric learning for ranking. Red points (left solid points): labeled as "best"; blue points (right solid points): labeled as "worst". Blank points are unlabeled data, which reveal the manifold geometry.**

In general, if no human knowledge is available, the computer does not know which point is "good" and which point is "bad". The two coordinates in Fig. 1 may have the same weight. Contrarily, if we allow the user to provide the most certain points, it will lead to a more meaningful ranking result. In Fig. 1 (a), we see that, labeling the points in the left as "best" and the points in the right as "worst", we may probably select $x$ axis as the important feature. Note that the projection direction of the linear transformation may be similar to the one of Fisher. However, there are many unlabeled data which show that the "U" shape is a non-linear manifold. Ideally, the good ranking result should be along the manifold which is shown in Fig. 1 (b). Using only linear transformation can not discover the intrinsic geometry. Thus, we need to embedded the data into a non-linear space. This can be solved by using the *kernel trick* [6, 8].

We denote the input points as $D = \{\mathbf{X}, \mathbf{Y}\}$. There are $l$ points having been appointed by human as the most certain points and $u$ unlabeled points. Then, the observed input points can be written as $\mathbf{X} = (\mathbf{X}_L, \mathbf{X}_U)$, where $\mathbf{X}_L = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l)$ and $\mathbf{X}_U = (\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, ..., \boldsymbol{x}_{l+u})$. Each point $\boldsymbol{x} \in \mathbb{R}^d$ is a $d$-dimensional vector. The goal of our linear transformation algorithm is to find a projection matrix $\mathbf{W} \in \mathbb{R}^{d\times m}$ that transforms the original Euclidean space to a more informative space for ranking.

To maximize the distances between the projected values of the most certainly "good" and "bad" points, we have the following objective function:

$$\boldsymbol{w}_j^T \mathbf{X}_L \mathbf{L}_b \mathbf{X}_L^T \boldsymbol{w}_j. \tag{1}$$

where $\boldsymbol{w}_j$ is the $jth$ projection direction. The graph Laplacian $\mathbf{L}_b$ is defined based on the graph which is constructed by the labeled data. Specifically it is $\mathbf{L}_b = \mathbf{D}_b - \mathbf{A}_b$, where $(\mathbf{D}_b)_{ii} = \sum_j (\mathbf{A}_b)_{ij}$ and

$$(\mathbf{A}_b)_{ij} = \begin{cases} \frac{1}{l} - \frac{1}{l_k} & \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong to the } kth \text{ concept} \\ \frac{1}{l} & \text{otherwise} \end{cases}. \tag{2}$$

$l_k$ is the number of the points labeled as the $kth$ concept. Actually, $\mathbf{D}_b$ is a zero matrix since the sum of rows of $\mathbf{A}_b$ are zeros.

To formulate the information provided by unlabeled data, we define $G = (V, E)$ as a weighted neighborhood graph to describe point cloud $\mathbf{X}$. $V$ is the vertex set of graph. $E$ is the edge set which contains the pairs of neighboring vertices $(\boldsymbol{x}_i, \boldsymbol{x}_j)$. A typical adjacency matrix $\mathbf{A}$ of neighborhood graph is:

$$\mathbf{A}_{ij} = \begin{cases} \exp\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\} & \text{if } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in E \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

Then the normalized graph Laplacian of a neighborhood graph [2] is:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}, \tag{4}$$

where the diagonal matrix $\mathbf{D}$ satisfies $\mathbf{D}_{ii} = d_i$, and $d_i = \sum_{j=1}^{l+u} \mathbf{A}_{ij}$ is the degree of vertex $\boldsymbol{x}_i$.

Based on the information provided by user and the observed data points, the learning work is to find a projection that can balance two terms. The first term is the force that makes the most certain points be mostly separated. The second term is considered as a spring that preserves the intrinsic structure of the data point cloud. The two forces are also shown in Fig. 1. Generalizing to the multi-dimensionality case, we have the following objective function:

$$\mathbf{W}^* = \arg \max_{\mathbf{W}\in\mathbb{R}^{d\times m}} \frac{|\mathbf{W}^T \mathbf{X}_L \mathbf{L}_b \mathbf{X}_L^T \mathbf{W}|}{|\mathbf{W}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{W}|}. \tag{5}$$

where $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_c) \in \mathbb{R}^{d\times m}$ is the projection matrix. Then the objective function (5) has the solution:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T \boldsymbol{w}_j^* = \eta_j \mathbf{X}_L \mathbf{L}_b \mathbf{X}_L^T \boldsymbol{w}_j^* \quad j = 1, ..., m, \tag{6}$$

where $\boldsymbol{w}_j^{*\prime}s$ $(j = 1, ..., m)$ are the eigenvectors corresponding to the $m$ largest eigenvalues $\eta_j{}'s$ of $(\mathbf{X}\mathbf{L}\mathbf{X}^T)^{-1}\mathbf{X}_L\mathbf{L}_b\mathbf{X}_L^T$.

## 3. EXPERIMENTS ON DIGITS IMAGE DATA

We test the algorithms with the USPS digits image data set.[1] User submit a query example and the computer retrieves and ranks the points in the observed data base. The retrieval accuracy is defined as:

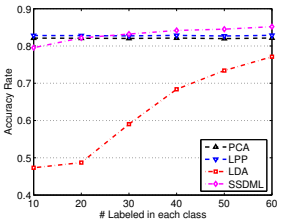
$$Accuracy = \frac{relevant\ examples\ in\ top\ N\ returns}{N}. \tag{7}$$

The original database contains 7291 training data and 2007 test data, and each data point is an image with $16 \times 16$ resolution. All data are "0"-"9" digit representations.

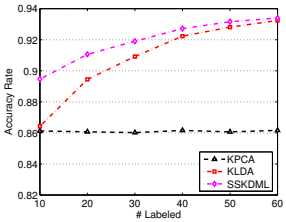
We test our algorithms with PCA [3], LDA [4], LPP [5] and the existent corresponding kernel versions. We randomly select 3000 data points as seen and 3000 data points unseen. Then seen data is randomly splitted into labeled and unlabeled data. For unsupervised methods PCA and LPP, we use the seen data to find the projection vectors. For supervised method LDA, we use the labeled data in the first $d-1$ dimensions in PCA sub-space as input features.

[1] *http://www.kernel-machines.org*

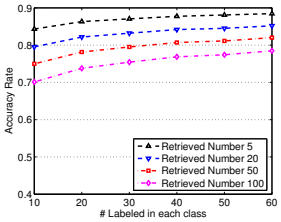
For our semi-supervised distance metric learning algorithm (SSDML), we make use of the partially labeled seen data in the $d-1$ dimensions in PCA sub-space as input features. For kernelized version, we directly input the original vector of digit images. New distance metric is adopted to test the retrieval accuracy on the unseen set to find the first $K$ nearest neighbors in the training set. The retrieved number $K$ is set to 20. Results are shown in Fig. 2 (a) and (b). Each test accuracy is an average of 50 random trials. We see that SSDML is competitive with PCA and LPP, and significantly better than LDA. SSKDML (kernelized SSDML) even outperforms KPCA and KLDA. It is shown that the accuracy rate is near 90% when we only label 10 points in each class. We also vary the retrieved numbers to show the accuracy differences. We select retrieved numbers as 5, 20, 50 and 100 and the average accuracy rates of 50 random trials are shown in Fig. 2 (c) and (d). As expectation, the accuracy decreases when the retrieved number increases.



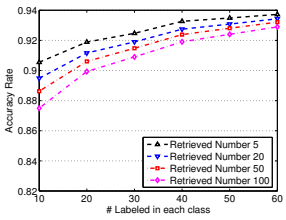

(a) Different labeled numbers, linear.




(b) Different labeled numbers, kernel.




(c) Different retrieved numbers, linear.




(d) Different retrieved numbers, kernel.

**Figure 2: Query Results: USPS data set, digits "0"-"9". Each test accuracy is an average of 50 random trials.**

## 4. APPLICATION TO HOUSING POTENTIAL ESTIMATION

In this section, we present another application, which is the computer aided housing potential estimation and location recommendation system. As we know, the factors that can affect the housing value are numerous. Therefore, to rank the value of different housing location is difficult for human. The most usually adopted foregoing method for business consultants is simple: they multiply a weight to each feature and then combine them together. However, the results may be not sufficiently good. In this experiment, we show that our algorithm can efficiently solve this problem and it has been embedded in a real system for consultants' work.

To estimate whether there is big value at a location for housing, consultants should investigate several factors around

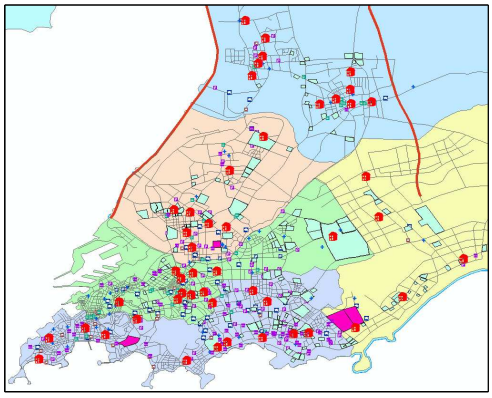
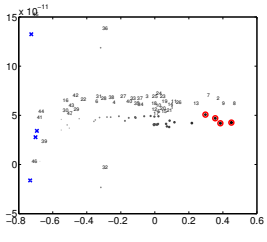
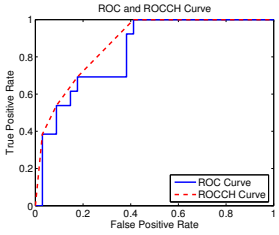


**Figure 3: An example of map with housing locations and their impact factors. Factors around the housing location within a million square meters should be considered as features for ranking.**

the housing location within a million square meters[2]. First, they count the commercial services sites such as shopping centers, banks, supermarkets, carnies and amusement parks and so on. Second, they count the social service sites such as hospitals, hotels, schools and colleges. Third, they evaluate traffics such as bus and subway stations, even the railway and air stations. Other sites such as restaurants and bars are also considered. After counting the units, they normalize them as probabilities. Moreover, environmental and social conditions around the housing location should also be evaluated. Finally, we obtain a vector that contains 32 features to represent housing potential factors. The main features are shown in Fig. 3. Only 47 housing locations are plotted.



(a) 2D visualization.




(b) ROC curve of classification

**Figure 4: An example of a small set (47 points). The areas under the ROC curve and convex hull of ROC curve are: AUROC=0.8303, AUROCCH=0.8744.**

We first present a result that uses the 47 locations plotted in Fig. 3, since only these samples has manually ranking values. We appoints the first four most certainly "good" locations and last four most certainly "bad" locations as the labeled points. After running our algorithm SSKDML, a 2D visualization of these 47 points is plotted in Fig. 4 (a). Each location has an ID for distinguishing. We can see that the data is reduced to one dimension, since the second dimension is scaled to $10^{-11}$. The ROC (receiver operating characteristic) curve and its convex hull are plotted in Fig. 4 (b). The

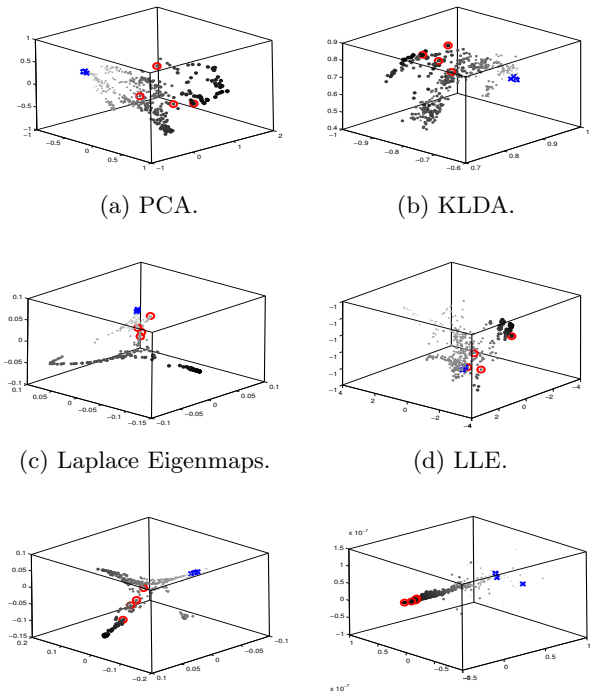
[2]While different countries may have different factors, we only present a case study of a city in China.

**Table 1: Numerical Comparison Results.**

| | Methods | AUROC | AUROCCH |
|---|---|---|---|
| Linear | PCA | 0.7783 | 0.8248 |
| | LDA | 0.7534 | 0.8066 |
| | SSDML | 0.7805 | 0.8326 |
| Non-Linear | LapEigs | 0.8032 | 0.8529 |
| | LLE | 0.8077 | 0.8586 |
| | LTSA | 0.7534 | 0.8179 |
| | KLDA | 0.7828 | 0.8484 |
| | SSKDML | 0.8303 | 0.8744 |

areas under the ROC curve and convex hull of ROC curve are: AUROC=0.8303, AUROCCH=0.8744. It is acceptable since even manually labeled rank value has mistakes. We also compare the proposed algorithm with some linear and non-linear metric learning algorithms: PCA, LDA, kernel LDA (KLDA), Laplace Eigenmaps (LapEigs) [1], LLE [7] an LTSA [10]. Table 4 shows that SSKDML gives the best result.

To show the efficiency of our algorithm, we also plot a visualization results of 497 points with the same labeled data as the 47 points case. In Fig. 5 (f), we can see that the points is also reduced to one dimension for our algorithm SSKDML, since there are two coordinates scaled to $10^{-7}$. The other linear and non-linear methods also give visualization results in Fig. 5 (a)-(e). However, they show less ranking information.



(a) PCA. (b) KLDA.

(c) Laplace Eigenmaps. (d) LLE.

(e) LTSA. (f) SSKDML.

**Figure 5: 3D visualization of 497 points. The gray level and size of the points indicate the final rank values. The rad circle and blue cross indicate the labeled points.**

## 5. CONCLUSIONS

In this paper, we present a novel type of domain knowledge for ranking, where only most certain points are provided by user. Having this domain knowledge, we can (1) give an order of the observed data points, (2) retrieve from an observed data base by querying new points. We propose a distance metric learning algorithm to deal with the new domain knowledge and can be used both for regression and for retrieval by querying. Experiments show that the semi-supervised method can improve the retrieval accuracy.

Besides giving the test results on the benchmark set, we also show an application to computer aided housing potential estimation and location recommendation problem. The implemented algorithm has been used in practice. The computer ranking results are satisfactory compared to the manually ranking value. It saved much time for consultants.

## 6. ACKNOWLEDGMENTS

We would like to thank Feiping Nie and Shiming Xiang for their helpful comments and discussions.

## 7. REFERENCES

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] F. Chung. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997.

[3] R. O. Duda., P. E. Hart., and D. G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2000.

[4] K. Fukunaga. *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press, Boston, MA, 1990.

[5] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. *IEEE Trans. on PAMI*, 27(3):328–340, 2005.

[6] K.-R. muller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12(2):181–201, 2001.

[7] S. T. Roweis and K. S. Lawrance. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[8] B. Schölkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[9] L. Yang and R. J. and. Distance metric learning: A comprehensive survey. Technical report, Michigan State University. http://www.cse.msu.edu/ yangliu1/frame_survey_v2.pdf.

[10] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing*, 26(1):313–338, 2004.

[11] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.