TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214
DOI: 10.26599/TST.2018.9010000

Ranking with Adaptive Neighbors

Muge Li, Liangyue Li, and Feiping Nie*

Abstract: Retrieving the most similar objects in a large-scale database for a given query is a fundamental building block in many application domains, ranging from web searches, visual, cross media, and document retrievals. State-of-the-art approaches have mainly focused on capturing the underlying geometry of the data manifolds. Graph-based approaches, in particular, define various diffusion processes on weighted data graphs. Despite success, these approaches rely on fixed-weight graphs, making ranking sensitive to the input affinity matrix. In this study, we propose a new ranking algorithm that simultaneously learns the data affinity matrix and the ranking scores. The proposed optimization formulation assigns adaptive neighbors to each point in the data based on the local connectivity, and the smoothness constraint assigns similar ranking scores to similar data points. We develop a novel and efficient algorithm to solve the optimization problem. Evaluations using synthetic and real datasets suggest that the proposed algorithm can outperform the existing methods.

Key words: Ranking; Adaptive neighbors; Manifold structure

arXiv:1803.05105v1 [cs.LG] 14 Mar 2018

1 Introduction

Retrieving the most similar objects in a large-scale database for a given query is a fundamental building block in many application domains, ranging from web search [1], visual retrieval [2–6], cross media retrieval [7], to document retrieval [8]. The most straightforward approach to such retrieval tasks is to compute the pairwise similarities between objects in the Euclidean space as the ranking scores. Nonetheless, high-dimensional data often lie on a nonlinear manifold [9, 10]. The Euclidean distance based approach largely ignores the intrinsic manifold structure and might degrade the retrieval performance.

State-of-the-art methods mainly focus on capturing the underlying geometry of the data manifold. The most common way is to first represent the data manifold using a weighted graph, wherein each vertex is a data object, and the edge weights are proportional to the pairwise similarities. All the vertices then repeatedly spread their affinities to their neighborhood via the weighted graph until a global stable state is reached. The various diffusion processes mainly differ in the transition matrix and the affinity update scheme [5]. Among others, the random walk transition matrix is widely used in PageRank [1], random walk with restart [11], self diffusion [12], label propagation [13] and graph transduction [14]. The random walk transition matrix is a row-stochastic matrix such that the transition probability is proportional to the edge weights.A slight variant is the symmetric normalized transition matrix used in the Ranking on Data Manifold method [15]. To reduce the effect of noisy nodes, random walks can be restricted to the k

• Muge Li is with Cixi Hanvos Yucai High School, Ningbo, China, 315300. E-mail: 1606024250@qq.com.
• Liangyue Li is with with the School of Computing, Informatics, Decision Systems Engineering, Arizona State University, Tempe, AZ, US, 85281. E-mail: liangyue@asu.edu.
• Feiping Nie is with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi?an, China, 710072. E-mail: feipingnie@gmail.com.
* To whom correspondence should be addressed.
Manuscript received: 2017-06-25; accepted: 2017-08-24

nearest neighbors by sparsifying the original weighted graph [16, 17]. For iterative update of the affinities, the random walk with restart allows for the random surfer to randomly jump to an arbitrary node. The modified diffusion process on the standard graph captures the high-order relations [17] and is equivalent to the diffusion process on the Kronecker product graph [18]. Despite success, graph-based ranking methods rely on fixed-weight graphs, making the ranking results sensitive to the input affinity matrix.

In this study, we propose the ranking with adaptive neighbors (RAN) algorithm simultaneously learns the data affinity matrix and the ranking scores. The proposed optimization explores two objectives. First, data points with smaller distance in the Euclidean space have high chance to be neighbors, i.e., more similar. In contrast to other graph-based ranking methods, the similarity is not computed a priori but is learned via optimizing the ranking scores. Consequently, the neighbors of each datum are adaptively assigned. Second, similar data points have similar ranking scores. This is essentially the smoothness constraint in graph transduction methods [19]. We develop a novel and efficient algorithm to solve the optimization problem. Evaluations using synthetic and real datasets suggest that the proposed ranking algorithm outperforms existing methods.

In section 2, we present the proposed RAN algorithm. Next, in section 3 we discuss the empirical evaluation results and, in section 4, we summarize the conclusions.

Notations: Throughout the paper, the matrices are written as upper-case letters. For matrix M, the i-th row and (i,j)-th element of M are denoted by m_i and m_{ij}, respectively. An identity matrix is denoted by I, and $\mathbf{1}$ denotes the column vector with all elements as one. For vector v and matrix M, $v \geq 0$ and $M \geq 0$ represent all the elements of v and M are nonnegative.

2 Ranking with adaptive neighbors

In this section, we discuss RAN algorithm and then the optimization approach for solving the objective function.

2.1 Proposed Formulation

Given a set of data points $\mathcal{X} = \{x_1, x_2, \ldots, x_N\} \subseteq \mathbb{R}^d$ with a query indicator vector $y = [y_1, y_2, \ldots, y_N]^T \in \{0,1\}^N$, where $y_1 = 1$ if x_i is the query and $y_1 = 0$ otherwise, the task is to find a function f that assigns each point in the data x_i a ranking score $f_i \in \mathbb{R}$ according to its relevance to the queries. We explore the local connectivity of each point for ranking purposes and in particular consider the k-nearest points as the neighbors of a specific node.

Data points separated by small distances in the Euclidean space have high chance to be neighbors. We denote the probability that the i-th data point x_i, and the j-th data point x_j are neighbors by s_{ij}. Intuitively, if the two data points are separated by a small distance, i.e., $\|x_i - x_j\|_2^2$ is small, then their probability s_{ij} of being connected is likely high. One way to find such probabilities $s_{ij}|_{j=1}^N$ is to solve the following optimization problem:

$$\min_{s_i^T \mathbf{1}=1, 0 \leq s_i \leq 1} \sum_{j=1}^{N} \|x_i - x_j\|_2^2 s_{ij} \tag{1}$$

where $s_i \in \mathbb{R}^N$ is a vector with the j-th element as s_{ij}. Nonetheless, the above optimization problem has a trivial solution, that is, $s_{ij} = 1$ for the nearest data point x_j of x_i, otherwise $s_{ij} = 0$. This can be addressed by adding a l_2-norm regularization on s_i to drag s_i closer to the center of mass of the simplex defined by $s_i^T \mathbf{1} = 1, 0 \leq s_i \leq 1$. This slight modification gives us the following optimization problem:

$$\min_{s_i^T \mathbf{1}=1, 0 \leq s_i \leq 1} \sum_{j=1}^{N} (\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2) \tag{2}$$

where the second term is the regularization term and γ is the regularization parameter.

For each data point x_i, we compute its probability of connecting to other data points using Eq. (2). As a result, we assign the neighbors of all the data points by solving the following problem:

$$\min_{\forall i, s_i^T \mathbf{1}=1, 0 \leq s_i \leq 1} \sum_{i,j=1}^{N} (\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2) \tag{3}$$

Similar data points have similar ranking scores, essentially a smoothness constraint over the data graph. We assume the matrix $S \in \mathbb{R}^{N \times N}$ is the similarity matrix obtained from assigning the neighbors, where each row is s_i^T. We write the smoothness constraint as,

$$\sum_{i,j=1}^{N} (f_i - f_j)^2 s_{ij} = 2 f^T L_S f \tag{4}$$

where f is the vector of ranking scores for all the data points, $L_S = D_S - \frac{S^T + S}{2}$ is the Laplacian matrix of the affinity matrix, and the degree matrix D_S is a diagonal matrix with the i-th diagonal element defined as $\sum_j (s_{ij} + s_{ji})/2$.

Combining the above and using the information from

the query, we derive the final objective function:

$$\begin{aligned}\min_{S,f} &\sum_{i,j=1}^{n}(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2) + 2\lambda f^T L_S f \\ &+ (f-y)^T U (f-y) \\ s.t. \quad &\forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1\end{aligned} \tag{5}$$

where U is a diagonal matrix with $U_{ii} = \infty$ (a large constant) if x_i is the query, otherwise $U_{ii} = 1$. The last term is equivalent to $\sum_{i=1}^{n} U_{ii}(f_i - y_i)^2$ to make the ranking results consistent with the queries. The queries are given much more weights as they reflect the user's search intentions. In non-queried examples, we do not know a priori whether they meet the user's intentions and give them lower weights. It is not easy to solve Eq. (5) because $L_S = D_S - \frac{S^T+S}{2}$ and D_S both depend on the similarity matrix S. In the next subsection, we propose a novel and efficient algorithm to solve this problem.

2.2 Optimization Solutions

We propose to solve Eq. (5) via an alternative optimization approach. We first fix S and then the problem transforms to:

$$\min_f 2\lambda f^T L_S f + (f-y)^T U (f-y) \tag{6}$$

We take the derivative of the above objective function w.r.t. f and set it to 0, obtaining the following linear equation:

$$(2\lambda L_S + U) f = Uy \tag{7}$$

The solution is easily obtained as $f = (2\lambda L_S + U)^{-1} Uy$.

When f is fixed, Eq. (5) transforms to:

$$\min_S \sum_{i,j=1}^{n}(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2) + 2\lambda f^T L_S f \tag{8}$$

$$s.t. \quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1 \tag{9}$$

And based on Eq. (4), it is written

$$\begin{aligned}&\min_S \sum_{i,j=1}^{n}(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 + \lambda(f_i - f_j)^2 s_{ij}) \\ &s.t. \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1\end{aligned} \tag{10}$$

Because the summations are independent of each other given i, we can solve the following sub-problem individually for each i:

$$\begin{aligned}&\min_{s_i} \sum_{j=1}^{n}(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 + \lambda(f_i - f_j)^2 s_{ij}) \\ &s.t. s_i^T \mathbf{1} = 1, 0 \le s_i \le 1\end{aligned} \tag{11}$$

We denote $d_{ij}^x = \|x_i - x_j\|_2^2$ and $d_{ij}^f = (f_i - f_j)^2$, and denote $d_i \in \mathbb{R}^N$ as a vector with the j-th element as $d_{ij} = d_{ij}^x + \lambda d_{ij}^f$. Then Eq. (11) is reformulated as:

$$\min_{s_i^T \mathbf{1} = 1, 0 \le s_i \le 1} \|s_i + \frac{d_i}{2\gamma}\|_2^2 \tag{12}$$

Next, we will show how to solve this equation in a closed form using the Lagrange multipliers method. The Lagrangian function of the problem is

$$\mathcal{L}(s_i, \eta, \beta_i) = \frac{1}{2}\|s_i + \frac{d_i}{2\gamma_i}\|_2^2 - \eta(s_i^T \mathbf{1} - 1) - \beta_i^T s_i \tag{13}$$

where η and β_i are non-negative Lagrangian multipliers.

According to the KKT condition, the optimal solution is

$$s_{ij} = (-\frac{d_{ij}}{2\gamma_i} + \eta)_+ \tag{14}$$

where $(x)_+$ is the shorthand for $\max\{x, 0\}$.

It is often desirable to focus on the locality of each point, as it can reduce the effect of noisy data and boost the performance in practice [20]. In this study, we will learn the sparse vector s_i and allow x_i to connect to its k-nearest neighbors. Such sparsification of S would minimize the computational cost.

We sort d_{ij} in ascending order such that $d_{i1} \le d_{i2} \le \ldots \le d_{iN}$. We want to learn the sparse s_i with only k nonzero elements, from Eq. (14); thus we have $s_{ik} > 0$ and $s_{i,k+1} = 0$. Therefore

$$\begin{cases} -\frac{d_{ik}}{2\gamma_i} + \eta > 0 \\ -\frac{d_{i,k+1}}{2\gamma_i} + \eta \le 0 \end{cases} \tag{15}$$

Considering the constraint $s_i^T \mathbf{1} = 1$, we obtain

$$\sum_{j=1}^{k}(-\frac{d_{ij}}{2\gamma_i} + \eta) = 1 \Rightarrow \eta = \frac{1}{k} + \frac{1}{2k\gamma_i}\sum_{j=1}^{k} d_{ij} \tag{16}$$

Substituting Eq. (16) into Eq. (15), we obtain the following inequality for γ_i

$$\frac{k}{2}d_{ik} - \frac{1}{2}\sum_{j=1}^{k} d_{ij} < \gamma_i \le \frac{k}{2}d_{i,k+1} - \frac{1}{2}\sum_{j=1}^{k} d_{ij} \tag{17}$$

For the objective function in Eq. (12) to have an optimal solution s_i, we set γ_i to

$$\gamma_i = \frac{k}{2}d_{i,k+1} - \frac{1}{2}\sum_{j=1}^{k} d_{ij} \tag{18}$$

The overall γ is set as the mean of all γ_i:

$$\gamma = \frac{1}{n}\sum_{i=1}^{n}(\frac{k}{2}d_{i,k+1} - \frac{1}{2}\sum_{j=1}^{k} d_{ij}) \tag{19}$$

The algorithm for solving the optimization problem in Eq. (5) is summarized in Algorithm 1.

Algorithm 1 Algorithm to solve problem in Eq. (5)

Input: (1) Data matrix $X \in \mathbb{R}^{n \times d}$,
(2) Query indicator vector y,
(3) parameters γ, λ.
Output: The ranking scores f.

```
Initialize S and compute L_S accordingly;
while not converged do
    Define the diagonal matrix U as: U_ii = ∞ if y_i = 1 and U_ii = 1 otherwise;
    Update f by solving Eq. (7) as f = (2λL_S + U)^{-1} U y;
    for i = 1, ..., N do
        Update i-th row of S by solving Eq. (12)
    end for
end while
```

3 Experiments

In this section, we show the performance of the proposed ranking algorithm RAN (Algorithm 1) on synthetic and real world datasets.

3.1 Synthetic datasets

We randomly generate two synthetic datasets constructed as two moons (Fig. 1) and three rings (Fig. 2) patterns. A query is given in the upper moon and the innermost ring marked in red cross. The task is to rank the remaining data points according to their relevance to the query. We represent the ranking scores returned by RAN using the diameter of the data points such that larger points are more relevant. From Fig. 1, we observe that the ranking scores gradually decrease along the upper moon. The same decreasing trend is also observed in the lower moon. In addition, the ranking scores in the upper moon are generally much higher than in the lower moon. Such ranking outcome is intuitively expected. We make similar observations for the three rings in Fig. 2. The data points in the innermost ring are more relevant than those in the middle ring, which are more relevant than those in the outermost ring. These results clearly show that the proposed RAN can capture the underlying manifold pretty well.

3.2 Real dataset

We compare the retrieval performance on three real image datasets: Yale [21], ORL [22] and USPS [23].

YALE: Yale contains face images of subjects at different poses and illumination conditions. We extract 11 images at different conditions for 15 subjects. Each image is down-sampled and normalized to zero mean and unit variance. The bandwidth for constructing the weighted graph for the graph based baselines is $\sigma = 0.021$. We set $k = 5$ and $\lambda = 90$ for RAN.

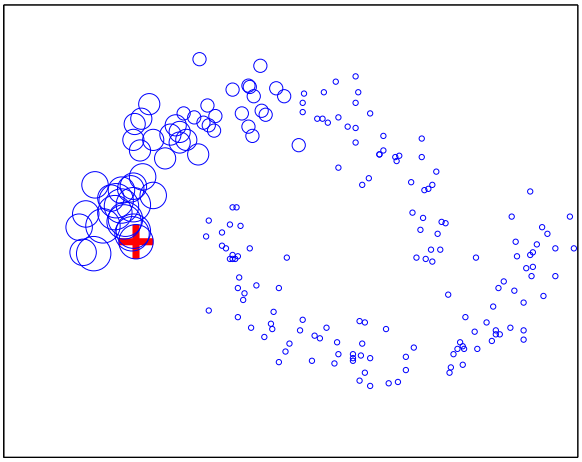

Fig. 1 Ranking Example using Two Moon.

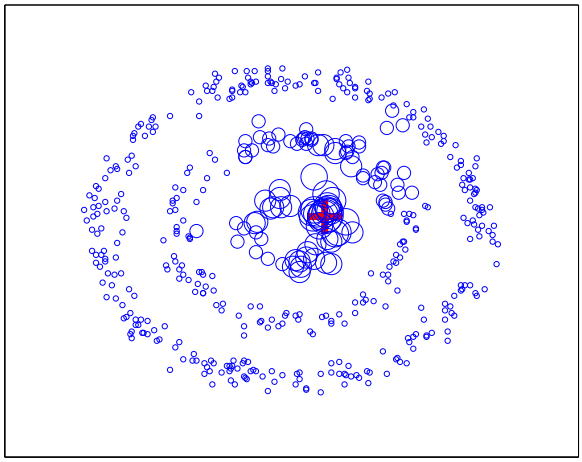

Fig. 2 Ranking Example using Three Ring.

ORL: ORL contains contains 400 images with ten different images for 40 different subjects each. The bandwidth for constructing the weighted graph for the graph based baselines is $\sigma = 20$. We set $k = 5$ and $\lambda = 0.1$ for RAN.

USPS: This dataset collects images of handwritten digits (0-9) from envelopes of the U.S. Postal Service. We extract 40 images for each digit and normalize them to 16×16 pixels in gray scale. The bandwidth for constructing the weighted graph for the graph based baselines is $\sigma = 0.8$. We set $k = 10$ and $\lambda = 1.0$ for RAN.

On all the datasets, we use each image as query and measure the retrieval accuracy by ranking all the other images. We compare the proposed RAN algorithm with the Euclidean distance based baseline and several other diffusion methods, including self-

diffusion (SD) [12], Personalized PageRank (PPR) [24], Manifold Ranking [15] and Graph Transduction (GT) [14]. The results are shown in Tables 1, 2 and 3. From the results, we can see that the proposed RAN algorithm consistently outperforms all other methods. The straightforward Euclidean distance based baseline is the worst because it ignores the manifold structure in the data. The various diffusion based methods capture the manifold information to a certain extent, but they assume the weighted data graph is fixed. We instead adaptively learn the localized weighted graph optimized for the ranking. To study how the locality of the graph, i.e., the number of neighbors k, affects the retrieval performance, we show (Fig. 3) the retrieval performance by varying the number of neighbors on USPS dataset. As it can be seen, it is important to select a reasonable value for k for the retrieval. For USPS, the best performance can be achieved at $k = 15$.

Table 1 Retrieval performance (%) for YALE.

Methods	Precision@10	Recall@10
Euclidean Distance	66.61	60.55
SD [12]	69.03	62.75
PPR [24]	69.03	62.75
Manifold Ranking [15]	68.85	62.59
GT [14]	68.91	62.65
RAN (ours)	**72.00**	**65.45**

Table 2 Retrieval performance (%) for ORL.

Methods	Precision@15	Recall@15
Euclidean Distance	41.56	62.35
SD [12]	46.87	70.30
PPR [24]	47.15	70.73
Manifold Ranking [15]	47.35	71.02
GT [14]	48.97	73.45
RAN (ours)	**49.02**	**73.53**

Table 3 Retrieval performance (%) for USPS.

Methods	Precision@50	Recall@50
Euclidean Distance	45.53	56.91
SD [12]	47.42	59.27
PPR [24]	47.39	59.24
Manifold Ranking [15]	47.42	59.28
GT [14]	46.18	57.72
RAN (ours)	**56.19**	**70.23**

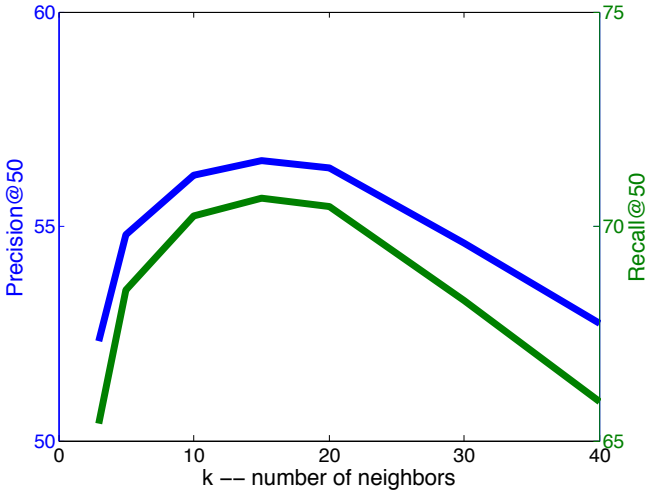

Fig. 3 Retrieval Performance (%) v.s. the number of neighbors on USPS.

4 Conclusions

We study the data ranking problem by capturing the underlying geometry of the data manifold. Instead of relying on the fixed-weight data graphs, we propose a new ranking algorithm that is able to learn the data affinity matrix and the ranking scores simultaneously. The proposed optimization formulation assigns adaptive neighbors to each data point based on the local connectivity and the smoothness constraint assigns similar ranking scores to similar data points. An efficient algorithm is developed to solve the optimization problem. Evaluations using synthetic and real datasets demonstrates the superior performance of the proposed algorithm.

References

[1] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[2] Jingrui He, Mingjing Li, Hong-Jiang Zhang, Hanghang Tong, and Changshui Zhang. Manifold-ranking based image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 9–16. ACM, 2004.

[3] Hanghang Tong, Jingrui He, Mingjing Li, Wei-Ying Ma, Hong-Jiang Zhang, and Changshui Zhang. Manifold-ranking-based keyword propagation for image retrieval. *EURASIP Journal on Advances in Signal Processing*, 2006(1):079412, 2006.

[4] Song Bai, Xiang Bai, Qi Tian, and Longin Jan Latecki. Regularized diffusion process for visual retrieval. In *AAAI*, pages 3967–3973, 2017.

[5] Michael Donoser and Horst Bischof. Diffusion processes for retrieval revisited. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 1320–1327, 2013.

[6] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *CVPR*, 2017.

[7] Yi Yang, Dong Xu, Feiping Nie, Jiebo Luo, and Yueting Zhuang. Ranking with local regression and global alignment for cross media retrieval. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 175–184. ACM, 2009.

[8] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 186–193. ACM, 2006.

[9] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[10] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[11] Hanghang Tong, Christos Faloutsos, and Jia-yu Pan. Fast random walk with restart and its applications. In *ICDM*, pages 613–622. IEEE, 2006.

[12] Bo Wang and Zhuowen Tu. Affinity learning via self-diffusion for image segmentation and clustering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2312–2319. IEEE, 2012.

[13] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

[14] Xiang Bai, Xingwei Yang, Longin Jan Latecki, Wenyu Liu, and Zhuowen Tu. Learning context-sensitive shape similarity by graph transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):861–874, 2010.

[15] Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on data manifolds. In *NIPS*, pages 169–176, 2003.

[16] Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In *NIPS*, NIPS'01, pages 945–952, 2001.

[17] X. Yang, S. Koknar-Tezel, and L. J. Latecki. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 357–364, 2009.

[18] Xingwei Yang, Lakshman Prasad, and Longin Jan Latecki. Affinity learning with diffusion on tensor product graph. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):28–38, 2013.

[19] Jun Wang, Tony Jebara, and Shih-Fu Chang. Graph transduction via alternating minimization. In *Proceedings of the 25th international conference on Machine learning*, pages 1144–1151. ACM, 2008.

[20] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *KDD*, KDD '14, pages 977–986, 2014.

[21] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001.

[22] Ferdinando S Samaria and Andy C Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 138–142. IEEE, 1994.

[23] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[24] Taher H Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002.

Feiping Nie received the Ph.D. degree in Computer Science from Tsinghua University, China in 2009, and currently is full professor in Northwestern Polytechnical University, China. His research interests are machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing and information retrieval. He has published more than 100 papers in the following top journals and conferences: TPAMI, IJCV, TIP, TNNLS/TNN, TKDE, Bioinformatics, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, ACM MM. His papers have been cited more than 7000 times and the H-index is 48. He is now serving as Associate Editor or PC member for several prestigious journals and conferences in the related fields.

Liangyue Li received the BEng degree in computer science from Tongji University in 2011. He is currently a Ph.D. student in the School of Computing, Informatics and Decision System Engineering, Arizona State University. His current research interests include large scale data mining and machine learning, especially for large graph data with application to social network analysis.