Int J Software Informatics, Volume 7, Issue 3 (2013), pp. 391–405 Tel: +86-10-62661040
International Journal of Software and Informatics, ISSN 1673-7288 http://www.ijsi.org

Email: ijsi@iscas.ac.cn, ijsi2007@gmail.com

Manifold Ranking using Hessian Energy

Ziyu Guan[1,2], Jinye Peng[1], and Shulong Tan[3]

[1](College of Information and Technology, Northwest University of China, Xi'an, China)

[2](Dept. of Computer Science, University of California at Santa Barbara, Santa Barbara, USA)

[3](College of Computer Science, Zhejiang University, Hangzhou, China)

Abstract In recent years, learning on manifolds has attracted much attention in the academia community. The idea that the distribution of real-life data forms a low dimensional manifold embedded in the ambient space works quite well in practice, with applications such as ranking, dimensionality reduction, semi-supervised learning and clustering. This paper focuses on ranking on manifolds. Traditional manifold ranking methods try to learn a ranking function that varies smoothly along the data manifold by using a Laplacian regularizer. However, the Laplacian regularization suffers from the issue that the solution is biased towards constant functions. In this work, we propose using second-order Hessian energy as regularization for manifold ranking. Hessian energy overcomes the above issue by only penalizing accelerated variation of the ranking function along the geodesics of the data manifold. We also develop a manifold ranking framework for general graphs/hypergraphs for which we do not have an original feature space (i.e. the ambient space). We evaluate our ranking method on the COREL image dataset and a rich media dataset crawled from Last.fm. The experimental results indicate that our manifold ranking method is effective and outperforms traditional graph Laplacian based ranking method.

Key words: manifold ranking; hessian energy; regularization

Guan ZY, Peng JY, Tan SL. Manifold ranking using hessian energy. *Int J Software Informatics*, **Vol.7, No.3 (2013): 391–405. http://www.ijsi.org/1673-7288/7/i166.htm**

1 Introduction

Recently, learning on manifolds has attracted much attention from different research communities, e.g. machine learning, computer vision and information retrieval. The idea that real-life data generally lies on a low dimensional manifold embedded in the ambient space works quite well in practice, with applications such as ranking[22], dimensionality reduction[10], semi-supervised learning[1] and clustering[20].

Ranking is a fundamental problem in many research fields, such as information retrieval and data mining. A ranking problem can be abstracted as follows: given a query, we want to find a real valued function f which ranks the data instances according to their relevance to the query. For two data instances x_i and x_j, if x_i is more relevant to the query, an ideal ranking function should give $f(x_i) > f(x_j)$. A

This work is sponsored by Northwest University of China
Corresponding author: Jinye Peng, Email: pjy@nwu.edu.cn
Received 2012-11-26; Revised 2013-07-13; Accepted 2013-08-31.

major challenge in real-life ranking problems is that the dimensionality of the data is often very high, which generally leads to the so-called "curse of dimensionality" problem[9], i.e. the volume of the feature space grows so fast that the available data becomes very sparse. This sparsity is problematic for any method that requires statistical significance. Recently, manifold learning has gained increasing attention and the intuition that real-life high dimensional data may have a lower dimensional intrinsic geometric structure has showed promising performance in ranking problems[21,22,8]. In this work we focus on manifold-based ranking.

Most manifold ranking methods are based on a Laplacian regularization framework[22,23]. Specifically, they exploit the discrete Laplace operator on a similarity graph (i.e. graph Laplacian) among data instances, in order to learn a ranking function which varies smoothly along the manifold. The similarity graph, which can be regarded as an approximation of the data manifold, is usually constructed in a k-Nearest-Neighbor (kNN) manner. This regularization framework has shown promising performance on various data types, such as image[8], text[21] and even complex rich media relational data[6,19]. The Laplacian-based ranking methods can also be interpreted as spreading from query nodes the ranking scores on the graph iteratively until a stationary distribution is achieved, with a nice connection with the PageRank algorithm[17,22]. However, a recent theoretical study[11] showed that the Laplacian regularizer is biased towards constant functions, which could potentially blur the ranking results.

In this paper, we propose a novel manifold ranking framework called Hessian-Ranking which makes use of the second-order Hessian energy[11] to design the regularization functional on the data manifold. Unlike the Laplacian regularizer, the Hessian regularizer favors functions which vary linearly along the geodesics of the data manifold. In order words, linearity means that the output value of a function changes linearly with respect to geodesic distances on the manifold. This property makes the Hessian energy particularly suitable for ranking, since in ranking problems our goal is to differentiate data instances in terms of their relevance to the query instance. Intuitively, the relevance should be inversely proportional to the geodesic distances of the corresponding data instances to the query instance.

Our contributions are as follows: (1) we incorporate the Hessian energy into manifold ranking problems and demonstrate its effectiveness empirically; (2) we systematically develop a manifold ranking framework which can handle not only data represented by vectors in a Euclidean space but also general graph data (without coordinates for each node) and even hypergraphs; (3) we evaluate our ranking framework by two real-life ranking problems, content based image retrieval (CBIR) with the COREL dataset and music recommendation with a rich media relational dataset crawled from Last.fm. The experimental results indicate our ranking framework is effective and outperforms the traditional Laplacian-regularized ranking method.

The rest of the paper is organized as follows. In Section 2, we review the Laplacian regularized manifold ranking method and also give a brief introduction to Hessian energy. The proposed Hessian regularized ranking framework is presented in Section 3. In Section 4, we apply the proposed ranking framework on two real-life ranking problems, image retrieval and music recommendation in social media data, and show the experimental results. Finally, Section 5 concludes our work.

2 Background

In this section, we first provide a review of the Laplacian-based manifold ranking method and discuss its properties. Then we present some background knowledge of the Hessian energy which will be used later as a regularizer in our ranking framework. We denote by $\mathcal{M}$ the m-dimensional data manifold embedded in $\mathbb{R}^n$, and use X and $\mathcal{X}$ to denote a data instance and the set of all data instances, respectively. Let $d = |\mathcal{X}|$ be the number of instances. In the follow, we use upper case letters in bold face to denote matrices and lower case letters in bold face to denote vectors.

2.1 Laplacian-regularized manifold ranking

Manifold ranking models are usually formulated in a regularized empirical error minimization framework:

$$\underset{f \in C^{\infty}(\mathcal{M})}{\arg\min} \sum_{i=1}^{l} (Y_i - f(X_i))^2 + \lambda S(f), \tag{1}$$

where f is the ranking function that we want to learn, $C^{\infty}(\mathcal{M})$ is the set of infinitely differentiable functions on $\mathcal{M}$ and $S(f)$ is the regularizer. The summation term is the least square error incurred by query instances (typically $l = 1$ and $y = 1$). It means that the ranking scores of query instances are forced to stick to their respective labels. The ranking scores of the remaining instances are controlled by $S(f)$. Equation (1) assumes that we have the complete data manifold. However, in practice we only have a sample of the data. Let $\mathbf{W}$ be the similarity matrix corresponding to the affinity graph G among all data instances constructed by connecting k nearest neighbors for each node (i.e. data instance):

$$W_{ij} = \begin{cases} \mathrm{sim}(X_i, X_j), & \text{if } X_i \text{ is among the } k \text{ nearest neighbors of } X_j, \\ & \text{or } X_j \text{ is among the } k \text{ nearest neighbors of } X_i \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

The Laplacian regularizer imposes a smooth constraint for the ranking function f on the affinity graph G

$$S_{\Delta}(\mathbf{f}) = \mathbf{f}^T \mathbf{L} \mathbf{f} = \frac{1}{2} \sum_{i,j} W_{ij} (f(X_i) - f(X_j))^2, \tag{3}$$

where $\mathbf{f}$ is the vector of function values for all data instances and $\mathbf{L}$ is the Laplacian matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ ($\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$). In practice, a normalized version $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ often generates better results[22]. Equation (3) indicates that if two data instances are similar, they should have similar ranking scores. Actually, the continuous analogue of the Laplacian regularizer is intrinsically trying to minimize the integration of the squared norm of f's gradient vector field over data manifold $\mathcal{M}$

$$S_{\Delta}(f) = \int_{\mathcal{M}} \|\nabla f\|^2 dV(x) = \int_{\mathcal{M}} f(\Delta f) dV(x) = \langle f, \Delta f \rangle, \tag{4}$$

where $\Delta = \sum_{i=1}^{m} \frac{\partial^2 f}{\partial x_i^2}$ is the Laplace operator and $dV(x)$ is the natural volume element[13]. As we can easily verify, constant ranking functions are the most "smooth" ones and incur a zero cost for $S_\Delta(f)$.

2.2 Hessian energy

For a real-valued function f, the Hessian energy is defined as

$$S_{Hess}(f) = \int_{\mathcal{M}} \|\nabla_a \nabla_b f\|^2_{T_x^* \mathcal{M} \otimes T_x^* \mathcal{M}} dV(x), \tag{5}$$

where $\nabla_a \nabla_b f$ is the second covariant derivative of f on $\mathcal{M}$. Note that the Hessian energy is by definition independent of the coordinate representation and depends only on the properties of $\mathcal{M}$. However, supposing we know the local normal coordinate system of $\mathcal{M}$ centered at point p, the second covariant derivative of f can be written in terms of the dual basis of the tangent space at p:

$$\nabla_a \nabla_b f \Big|_p = \sum_{r,s=1}^{m} \frac{\partial^2 f}{\partial x_r \partial x_s}\Big|_p dx_a^r \otimes dx_b^s. \tag{6}$$

Hence, the norm of the second covariant derivative of f at p is simply the Frobenius norm of the Hessian matrix of f at p

$$\|\nabla_a \nabla_b f\|^2_{T_p^* \mathcal{M} \otimes T_p^* \mathcal{M}} = \sum_{r,s=1}^{m} \left(\frac{\partial^2 f}{\partial x_r \partial x_s}\Big|_p \right)^2. \tag{7}$$

Unlike the Laplacian regularizer, the Hessian regularizer tries to minimize the integration of the squared norm of f's second order covariant derivative over $\mathcal{M}$. This means if f varies linearly with respect to the geodesics of $\mathcal{M}$, it will not be penalized by the Hessian regularizer. Formally speaking, for any geodesic function $\gamma : (-\varepsilon, \varepsilon) \to \mathcal{M}$ parameterized by arc length s, we have $\frac{\partial}{\partial s} f(\gamma(s)) = \text{constant}$[5]. This property is desirable in the ranking setting. Intuitively, the relevance of a data instance to the query instance should decrease linearly with respect to its geodesic distance to the query on the data manifold.

3 Manifold Ranking by Hessian Energy

In this section, we present our manifold ranking framework based on Hessian energy. Firstly, we discuss how to do ranking for data represented by vectors in a Euclidean space, which means we have a vector representation for $\mathcal{X}$. We then show our solution for general graph/hypergraph data where we do not have a vector representation for the data.

3.1 Hessian ranking for vector data

In real-life applications, we do not have the complete view of the data manifold $\mathcal{M}$, but only have a sample from $\mathcal{M}$, i.e. $\mathcal{X}$. Therefore, we need to approximate the Hessian energy from the data sample. In order to estimate the Hessian regularizer of f, we need to first estimate the local structure of the data manifold, i.e. the local tangent space $T_X\mathcal{M}$ centered at each data instance (point) X. After we obtain the estimates of local normal coordinates, the Hessian energy can then be estimated by Eq. (7).

The local structure of $\mathcal{M}$ at X_i can be estimated from its k nearest neighbors $N_k(X_i)$ (including X_i itself). When the sampled data is dense enough, the k nearest neighbors of a data point can provide a reasonable estimate of the local structure. In order to estimate the local tangent space $T_{X_i}\mathcal{M}$ of X_i, we perform PCA on $N_k(X_i)$ and treat the m leading eigenvectors $\{u_r\}_{r=1}^m$ as the basis of the tangent space[11]. The data points in $N_k(X_i)$ are then centered at X_i and projected onto the basis vectors.

After obtaining the local normal coordinates for each data point in $N_k(X_i)$, the next step is to estimate the Hessian of f at each point X_i which can be approximated as follows

$$\left.\frac{\partial^2 f}{\partial x_r \partial x_s}\right|_{X_i} \approx \sum_{j=1}^{k} H_{rsj}^{(i)} f(X_j), \tag{8}$$

where H is an operator we need to derive which establishes the relationship between function values and their second order derivatives. H can be computed by fitting the second-order Taylor expansion of f at each point X_i. In particular, for each $X_j \in N_k(X_i)$, we have

$$f(X_j) = f(X_i) + \sum_{r=1}^{m} B_r x_r(X_j) + \sum_{r=1}^{m}\sum_{s=r}^{m} A_{rs} x_r(X_j) x_s(X_j), \tag{9}$$

where $x_r(X_j)$ represents the r-th coordinate of X_j in the local normal coordinate system of X_i and B_r and A_{rs} correspond to the first order and second order derivatives of f at X_i respectively

$$B_r = \left.\frac{\partial f}{\partial x_r}\right|_{X_i}, \qquad A_{rr} = \left.\frac{1}{2}\frac{\partial^2 f}{\partial x_r^2}\right|_{X_i}, \qquad A_{rs} = \left.\frac{\partial^2 f}{\partial x_r \partial x_s}\right|_{X_i}. \tag{10}$$

To fit the above polynomial we use standard linear least squares,

$$\arg\min_{\mathbf{w}\in\mathbb{R}^P} \|\mathbf{f}^{(i)} - f(X_i)\mathbf{e} - \mathbf{\Phi}\mathbf{w}\|^2, \tag{11}$$

where $\mathbf{f}^{(i)}$ is a $k \times 1$ vector containing function values of data instances in $N_k(X_i)$, $\mathbf{e}$ is a $k \times 1$ vector with all elements equal to 1 and $\mathbf{\Phi} \in \mathbb{R}^{k\times P}$ represents the design matrix with $P = m + \frac{m(m+1)}{2}$. The j-th row of $\mathbf{\Phi}$ are the monomials for X_j, i.e. $[x_1(X_j), \ldots, x_m(X_j), x_1(X_j)x_1(X_j), \ldots, x_m(X_j)x_m(X_j)]$. Since this optimization problem is convex, its solution can be easily obtained by differentiating the objective function with respect to $\mathbf{w}$ and setting the derivative to 0:

$$2\mathbf{\Phi}^T(f(X_i)\mathbf{e} - \mathbf{f}^{(i)}) + 2\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w} = 0 \quad \Rightarrow \quad \mathbf{w} = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T(\mathbf{f}^{(i)} - f(X_i)\mathbf{e}). \tag{12}$$

The last $\frac{m(m+1)}{2}$ components of $\mathbf{w}$ correspond to the coefficients A_{rs} in Eq. (9). According to Eq. (10) and Eq. (8) we obtain the operator H. Consequently, the Forbenius norm of the Hessian of f at X_i can be estimated as

$$\|\nabla_a \nabla_b f\big|_{X_i}\|^2$$

$$\approx \sum_{r,s=1}^{m} \left(\sum_{j=1}^{k} H_{rsj}^{(i)} f(X_j) \right)^2 = \sum_{j,h=1}^{k} f(X_j) f(X_h) B_{jh}^{(i)} = \left(\mathbf{f}^{(i)}\right)^T \mathbf{B}^{(i)} \mathbf{f}^{(i)}, \tag{13}$$

where $B_{jh}^{(i)} = \sum_{r,s=1}^{m} H_{rsj}^{(i)} H_{rsh}^{(i)}$. Finally, the estimate of the Hessian energy $S_{Hess}(f)$ is simply the sum of local Hessian over all data instances:

$$\begin{aligned} & S_{Hess}(f) \\ \approx & \sum_{i=1}^{d} \sum_{r,s=1}^{m} \left(\left. \frac{\partial^2 f}{\partial x_r \partial x_s} \right|_{X_i} \right)^2 = \sum_{i=1}^{d} \sum_{j \in N_k(X_i)} \sum_{h \in N_k(X_i)} f(X_j) f(X_h) B_{jh}^{(i)} = \mathbf{f}^T \mathbf{B} \mathbf{f}, \end{aligned} \tag{14}$$

where $\mathbf{B}$ is the matrix that sums up the $\mathbf{B}^{(i)}$'s for all data instances.

After we obtain the estimate of the Hessian energy, the final objective function of Hessian Ranking is designed similarly as that of Laplacian-regularized manifold ranking

$$\arg \min_{\mathbf{f} \in \mathbb{R}^d} (\mathbf{y} - \mathbf{f})^T \mathbf{I}_q (\mathbf{y} - \mathbf{f}) + \lambda \mathbf{f}^T \mathbf{B} \mathbf{f}, \tag{15}$$

where $\mathbf{I}_q$ is a diagonal matrix with (i,i)-th element equal to 1 if i is a query and zero otherwise, and $\mathbf{y}$ is the query label vector. Differentiating the objective function with respect to $\mathbf{f}$ and set the derivative to zero, we get the following linear system

$$(\mathbf{I}_q + \lambda \mathbf{B}) \mathbf{f} = \mathbf{y} \tag{16}$$

Since $\mathbf{B}$ is usually sparse[11], this linear system can be solved efficiently by using the left division operator in MATLAB. Once $\mathbf{f}$ is computed, we can rank data instances by their corresponding scores in $\mathbf{f}$.

3.2 Hessian ranking for graph data

In the last subsection, we show how to do Hessian Ranking when the data has an original feature space (Euclidean space). Nevertheless, in many real world settings, we only have the relational information between data, i.e. graphs. For example, in a social tagging service we have the relationships between users, tags and resources, but not their feature representations[6,7]. Furthermore, some relationships are triplets or even higher order ones which are modeled by hypergraphs[19]. Here we discuss how to implement Hessian Ranking in the general graph/hypergraph settings.

In order to estimate the Hessian energy, we need to estimate the local structure of the data manifold. However, without a feature representation, we cannot apply PCA in local neighborhoods to estimate the local tangent spaces. We can treat edges in a graph as representing the affinity relationships between the corresponding nodes. Therefore, the graph can be viewed as an approximation of the data manifold, just as the similarity graph in the graph Laplacian case. Then our goal is to learn a representation for nodes in the graph which best reflects the geometric structure of the data manifold approximated by the graph. In other words, the goal is to learn a Euclidean space which best preserves the local structure of the manifold.

Let us first consider a normal undirected graph G. Let $\mathbf{W}$ be the weighted adjacency matrix for G. Let $\mathbf{q}_i$ be the Euclidean representation for X_i that we want to derive and $\mathbf{Q} \in \mathbb{R}^{d \times k}$ be the matrix that contains all $\mathbf{q}_i$'s as row vectors. The central idea is that if two instances are similar, they should be near to each other in the learned space:

$$\underset{\mathbf{Q}\in\mathbb{R}^{d\times k}}{\arg\min}\,\frac{1}{2}\sum_{i,j}W_{ij}\left\|\mathbf{q}_i-\mathbf{q}_j\right\|^2, \tag{17}$$

With simple transformations, we can write Eq. (17) as

$$\begin{aligned}\frac{1}{2}\sum_{i,j}W_{ij}\left\|\mathbf{q}_i-\mathbf{q}_j\right\|^2&=\frac{1}{2}\sum_{i,j}W_{ij}\left(\mathbf{q}_i^T\mathbf{q}_i+\mathbf{q}_j^T\mathbf{q}_j-2\mathbf{q}_i^T\mathbf{q}_j\right)\\&=\sum_i\|\mathbf{q}_i\|^2De(i)-\sum_{i,j}W_{ij}\mathbf{q}_i^T\mathbf{q}_j\\&=tr(\mathbf{Q}^T\mathbf{D}\mathbf{Q})-tr(\mathbf{Q}^T\mathbf{W}\mathbf{Q})\\&=tr(\mathbf{Q}^T\mathbf{L}\mathbf{Q}),\end{aligned}$$

where $tr(\cdot)$ denotes the trace of a matrix, $De(i)=\sum_j W_{ij}$ is the degree of X_i in graph G, $\mathbf{D}$ is a diagonal matrix with $D_{ii}=De(i)$ and $\mathbf{L}$ is the graph Laplacian matrix. In addition, we maximize the global variance in the target space in order to maintain the discrimination power of the space. In the discrete graph setting, the probability of observing a node can be estimated by the node's degree[4]. Therefore, the total variance of $\mathcal{X}$ in the target space can be estimated by $tr(\mathbf{Q}^T\mathbf{D}\mathbf{Q})$ (treating the origin as the mean). The final optimization formulation is as follows

$$\underset{\mathbf{Q}\in\mathbb{R}^{d\times k}}{\arg\min}\,\frac{tr(\mathbf{Q}^T\mathbf{L}\mathbf{Q})}{tr(\mathbf{Q}^T\mathbf{D}\mathbf{Q})},\quad \text{s.t. }\ \mathbf{e}^T\mathbf{Q}=\mathbf{0}. \tag{18}$$

The constraint means that the trivial feature $\mathbf{e}$ is removed from the solution. By the Rayleigh-Ritz theorem[15], the solution of this optimization problem is given by the first k non-trivial generalized eigenvectors (as column vectors of $\mathbf{Q}$) corresponding to the smallest eigenvalues of $(\mathbf{L},\mathbf{D})$. $\mathbf{Q}$ contains the vector representation $\mathbf{q}_i$ for each X_i as a row vector which best preserves the local structure of the manifold approximated by the graph.

For hypergraphs, we can compute such a Euclidean space similarly. Let $G=(V,E)$ be a hypergraph with node set V and edge set E. A hyperedge $e\in E$ can be regarded as a subset of vertices. e is said to be incident with a vertex v if $v\in e$. Each hyperedge e is associated with a weight denoted by $w(e)$ which encodes the strength of the connection. Let $\mathbf{H}$ be a $|V|\times|E|$ weighted incidence matrix where an entry $H(v,e)=1$ if $v\in e$ and 0 otherwise. The degree of a node i is defined as $De(i)=\sum_{e\in E}w(e)H(i,e)$. The degree of an edge e is defined as $\delta(e)=\sum_{i\in V}H(i,e)$, i.e. the number of nodes in e. To learn the Euclidean space that best preserves the hypergraph structure, we minimize the following cost function:

$$\underset{\mathbf{Q}\in\mathbb{R}^{d\times k}}{\arg\min}\,\frac{1}{2}\sum_{e\in E}\frac{1}{\delta(e)}\sum_{i,j\in e}w(e)\left\|\mathbf{q}_i-\mathbf{q}_j\right\|^2, \tag{19}$$

which can be re-written as

$$\frac{1}{2}\sum_{e\in E}\frac{1}{\delta(e)}\sum_{i,j\in e}w(e)\left\|\mathbf{q}_i-\mathbf{q}_j\right\|^2$$

$$
\begin{aligned}
&= \frac{1}{2}\sum_{e\in E}\frac{1}{\delta(e)}\sum_{i,j\in V} w(e)H(i,e)H(j,e)\left\|\mathbf{q}_i-\mathbf{q}_j\right\|^2\\
&= \frac{1}{2}\sum_{e\in E}\frac{1}{\delta(e)}\sum_{i,j\in V} w(e)H(i,e)H(j,e)(\mathbf{q}_i^T\mathbf{q}_i+\mathbf{q}_j^T\mathbf{q}_j-2\mathbf{q}_i^T\mathbf{q}_j)\\
&= \sum_{i\in V}\|\mathbf{q}_i\|^2 De(i)-\sum_{e\in E}\sum_{i,j\in V}\frac{w(e)H(i,e)H(j,e)}{\delta(e)}\mathbf{q}_i^T\mathbf{q}_j\\
&= tr(\mathbf{Q}^T\mathbf{D}_v\mathbf{Q})-tr(\mathbf{Q}^T\mathbf{H}\mathbf{W}_e\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{Q})=tr(\mathbf{Q}^T\tilde{\mathbf{L}}\mathbf{Q}), \qquad (20)
\end{aligned}
$$

where $\mathbf{D}_v$, $\mathbf{D}_e$ and $\mathbf{W}_e$ are three diagonal matrices containing node degrees, edge degrees and edge weights respectively. Similarly, we also need to maximize the variance in the target space which can be estimated by $tr(\mathbf{Q}^T\mathbf{D}_v\mathbf{Q})$. The final optimization problem is

$$
\arg\min_{\mathbf{Q}\in\mathbb{R}^{d\times k}}\frac{tr(\mathbf{Q}^T\tilde{\mathbf{L}}\mathbf{Q})}{tr(\mathbf{Q}^T\mathbf{D}_v\mathbf{Q})}, \quad \text{s.t. } \mathbf{e}^T\mathbf{Q}=\mathbf{0}. \qquad (21)
$$

Again, the solution is obtained by solving the generalized eigenvector problem $\tilde{\mathbf{L}}\mathbf{q}=\lambda\mathbf{D}_v\mathbf{q}$ and selecting the first k non-trivial generalized eigenvectors corresponding to the smallest eigenvalues.

If the sampled data is dense enough, the k nearest neighbors of instance X_i in the learned space $\mathbf{Q}$ can well approximate the tangent space at X_i. Then we can do Hessian Ranking with the learned space.

4 Experiments

In this section we apply Hessian Ranking on two real-life ranking problems, image retrieval and music recommendation in social media data. In image retrieval, we have a feature representation for the data instances (i.e. images), while for the music recommendation problem we only have the relations between different entities (e.g. users, music tracks, tags). In the following, we first show the experimental results for the image retrieval problem and then discuss the results for the music recommendation problem.

4.1 Image retrieval

4.1.1 Dataset

Our dataset for the image retrieval experiment contains 5,000 images of 50 semantic categories from the COREL database. Each image in the dataset is described by a 297-dimensional feature vector which consists of the following information

- Grid Color Moment: each image is partitioned into 3 × 3 grids. The color moments (i.e. mean, variance and skewness) in each color channel (R, G, B) for each grid are extracted. Totally, we have 81 color moment features.

- Edge: the Canny edge detector[3] is used to obtain the edge map for the edge orientation histogram. Each histogram contains 36 bins with 10 degree for each bin. An additional bin is used to count the number of pixels without edge information. This category has 37 features.

- Gabor Wavelets Texture: each image is first scaled to size 64 × 64. The Gabor wavelet transform[12] is then applied on the scaled image with 5 levels and 8 orientations, which results in 40 subimages. For each subimage, mean, variance and skewness are computed. This type of feature contributes 120 dimensions.

- Local Binary Pattern (LBP): a gray-scale texture measure derived from a general texture definition in a local neighborhood[16]. This forms a 59-dimensional LBP histogram vector.

4.1.2 Evaluation and results

We compare Hessian Ranking (abbreviated as *HessRanking* hereafter) with the traditional Laplacian-regularized manifold ranking method which is denoted by *LapRanking*. *Euclidean Distance* is also employed as a baseline, which ranks images according to their Euclidean distances to the query image in the 297-dimensional feature space.

Specifically, we treat each of the 5000 images as a query image and rank the remaining images. The relevant set for each image is the remaining 99 images in the corresponding category. We use Precision, Recall, Normalized Discount Cumulative Gain (NDCG) and Mean Average Precision (MAP) to evaluate these ranking methods. For a ranking position n, Precision is defined as the number of relevant images in the ranking list up to ranking position n divided by n. Recall is defined as the number of relevant images in the ranking list up to ranking position n divided by the number of all relevant images (i.e. 99). NDCG at position n is defined as

$$\text{NDCG@n} = Z_n \sum_{i=1}^{n} \frac{2^{r_i} - 1}{\log_2(i+1)}, \tag{22}$$

where r_i is the rating of the image at rank i. In our case, r_i is 1 if the image is a relevant image and 0 otherwise. Z_n is chosen so that the perfect ranking has a NDCG value of 1. Average Precision (AP) is the average of precision scores after each relevant image in the ranked list:

$$\text{AP} = \frac{\sum_i \text{Precision@}i \times \text{corr}_i}{\text{No. of relevant images in the list}}, \tag{23}$$

where Precision@i is the precision at ranking position i and $\text{corr}_i = 1$ if the image at position i is a relevant image, otherwise $\text{corr}_i = 0$. MAP is the mean of average precision scores over all query instances. In our experiments, we compute MAP with respect to ranked lists of length 100. For the two manifold ranking methods, each time we take one image as the query and set its query label to 1. The performance results are averaged over all 5000 query images.

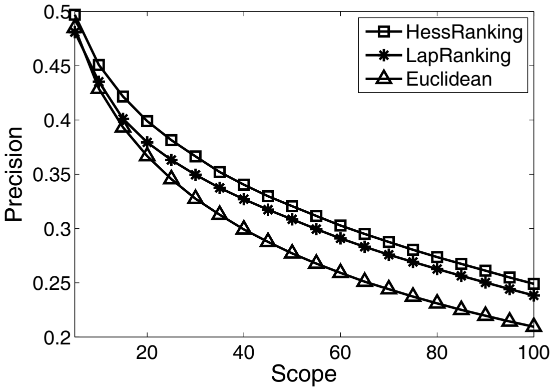


Figure 1. Performance comparison of the three ranking methods on the COREL dataset in terms of Precision under different scopes. The results are averaged over 5000 query instances.

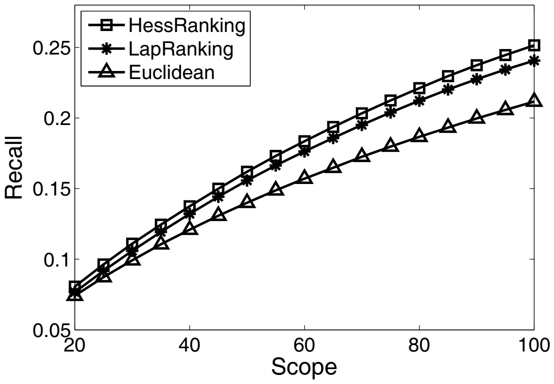
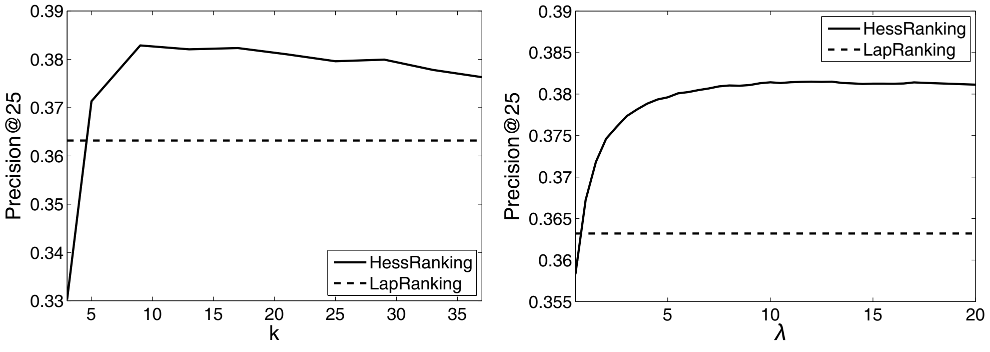


Figure 2. Performance comparison of the three ranking methods on the COREL dataset in terms of Recall under different scopes. The results are averaged over 5000 query instances.

Figures 1 and 2 show the performance comparison of the three ranking methods on the COREL dataset. We show their Precision and Recall as a function of the scope. We tune the parameters of LapRanking and show its best performance. For HessRanking, we set $m = k = 20$ and $\lambda = 11.5$. How to set these parameters will be discussed later. As can be seen, HessRanking consistently outperforms the other two baseline methods in terms of Precision and Recall. The performance superiority is significant according to one-tailed paired t-test with significance level $\alpha = 0.01$. We also show the performance comparison in terms of NDCG@n and MAP in Table 1. One can see that HessRanking also outperforms the baseline methods over a wide range of ranking positions. For NDCG@n, the superiority of HessRanking is significant according to Wilcoxon test with significance level $\alpha = 0.05$. This comparison demonstrates that our intuition that the Hessian energy is suitable for ranking problems is correct in practical problems. The reason should be that the Hessian energy favors ranking functions which change linearly with the geodesic distance to the query instance, while the traditional Laplacian-based ranking method has a bias towards constant functions.

Table 1 Comparison of the three ranking methods in terms of NDCG@n and MAP on the COREL dataset

Methods	NDCG@10	NDCG@20	NDCG@30	NDCG@50	NDCG@100	MAP
HessRanking	**0.484**	**0.457**	**0.451**	**0.467**	**0.544**	**0.378**
LapRanking	0.465	0.438	0.434	0.451	0.528	0.366
Euclidean	0.467	0.431	0.420	0.437	0.525	0.342



(a) (b)

Figure 3. Performance of HessRanking on the COREL dataset when varying parameters (a) k (i.e. k nearest neighbors) and (b) λ. We fix the other parameter when tuning one of k and λ. The performance of the best baseline is also shown for comparison purpose.

Next we explore the impact of different parameter settings on the performance of HessRanking. The free parameters of HessRanking are k for kNN, m for the dimensionality of the manifold and λ, the regularization weight of $\mathbf{B}$. We first set $m = k$ and vary k and λ. Figure 3 show the corresponding results. When varying k, λ is fixed at $\lambda = 11.5$; when tuning λ, we fix k at 20. The dotted lines represent the performance of LapRanking which is the best result baselines can achieve. We can see that HessRanking outperforms LapRanking in a wide range of parameter values. Then we investigate the relationship between k and m. We vary m with different fixed k values and the results are shown in Figure 4. Generally speaking, when $(m+1)m/2 > k$, the model tends to overfit the data[11]. However, we find that the performance of HessRanking increases with increasing m (and seems to converge when $m = k$). This could be because that our ranking function is learned from and applied to the same data. Therefore, overfitting does not affect the performance of HessRanking.

4.2 Music recommendation

4.2.1 Dataset

Our dataset is collected from Last.fm[1] which is a popular social platform for music listening and sharing. In Last.fm, users can add keyword tags to music related objects and join different groups. The specific collection procedure can be found in

[1] http://last.fm/

Ref. [19]. The Last.fm dataset contains six types of objects, whose notations and statistics are summarized in Table 2.

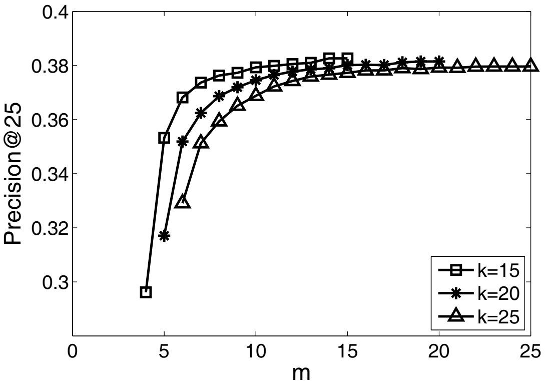


Figure 4. Model selection with respect to the dimensionality of the data manifold (m).

Table 2 Objects in our data set

Objects	Notations	Count
Users	U	2596
Groups	G	1124
Tags	Ta	3255
Tracks	Tr	16055
Albums	Al	4694
Artists	Ar	371

The relations among these objects are given in Table 3. The relations are divided into four categories, social relations, actions on resources, inclusion relations among resources, and acoustic-based music similarity relations. Social relations include friendship relations and membership relations (e.g., an interest group), denoted by R_1 and R_2, respectively. Actions on resources involve four types of relations, i.e., listening relations (R_3), and tagging relations on tracks, albums and artists (R_4, R_5 and R_6). Inclusion relations among resources are the inclusion relations between tracks and albums, albums and artists (R_7 and R_8). Acoustic-based music similarity relations are denoted by R_9. To compactly represent the music content, We derive features from Mel-frequency cepstral coefficients (MFCCs)[2]. MFCCs are prevalent in audio classification. A given music track is segmented into short frames and the MFCC is computed for each frame. Similar to Ref. [14], we use K-means to group all the frames of each track into several clusters. For all the clusters, the means, covariances, and weights are computed as the signature of the music track. To compare the signatures for two different tracks, we employ the Earth-Mover's Distance (EMD)[18]. The six types of objects and nine types of relations among these objects are employed to construct the hypergraph which will be used in the experiment. The readers could refer to Ref. [19] for construction details of the hypergraph.

4.2.2 Results

Table 3 Relations in our data set

Relations	Notations	Count
Friendship relations	R_1	4503
Membership relations	R_2	1124
Listening relations	R_3	304860
Tagging relations on tracks	R_4	10936
Tagging relations on albums	R_5	730
Tagging relations on artists	R_6	36812
Track-album inclusion relations	R_7	4694
Album-artist inclusion relations	R_8	371
Similarities between tracks	R_9	-

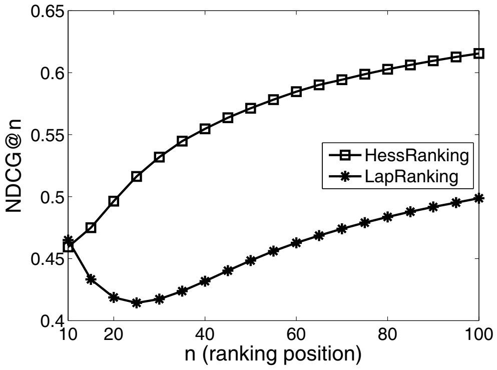


Figure 5. Performance comparison of HessRanking and LapRanking in terms of NDCG@n on the Last.fm dataset.

In this experiment, we treat each user as a query and rank music tracks accordingly. The top ranked music tracks which have not been listened to by the user are then recommended to the user. For each user, we randomly select 20% listening relations as test data as well as the ground truth[19]. The final performance results are averaged over all 2596 users. Figure 5 shows the performance comparison of HessRanking and LapRanking in terms of NDCG. Since NDCG favors ranked lists which put relevant items at high positions, the results indicate that HessRanking can achieve better ranking results than LapRanking. The performance superiority is significant by Wilcoxon test with significance level $\alpha = 0.05$ (except for $n = 10$). The MAP scores for HessRanking and LapRanking are 0.399 and 0.295, respectively, which also indicates that HessRanking generates better ranked lists than LapRanking. The reason should be that LapRanking has a bias towards constant functions and consequently tends to blur the ranking results. Unfortunately, HessRanking does not show significant better performance than LapRanking in terms of Precision and Recall. As aforementioned, we learn the Euclidean space which best preserves the local geometric structure of the manifold approximated by the relational graph. Such a transformation leads to information loss and noises, which could explain why HessRanking cannot beat LapRanking on Precision and Recall.

5 Conclusions

In this work, we propose a novel manifold ranking framework based on the Hessian energy regularization. The critical difference between Hessian regularization and Laplacian regularization is that the Hessian regularizer favors ranking functions which change linearly along the geodesics of the data manifold, while the Laplacian regularizer has a bias towards constant functions and could potentially blur the ranked lists. The experimental results on two real-life ranking problems, content based image retrieval and music recommendation in social media data, demonstrate that HessRanking outperforms LapRanking significantly. In future work, we plan to develop new Hessian energy based learning methods for graph data which can eliminate the information loss and noises incurred by projecting graph nodes into Euclidean spaces.

References

[1] Belkin M, Niyogi P. Semi-supervised learning on riemannian manifolds. Machine Learning, 2004, 56(1): 209–239.

[2] Berenzweig A, Logan B, Ellis DPW, Whitman B. A large-scale evaluation of acoustic and subjective music-similarity measures. Computer Music Journal, 2004, 28(2): 63–76.

[3] Canny J. A computational approach to edge detection. IEEE Trans. on Pattern Analysis and Machine Intelligence , 1986, (6): 679–698.

[4] Chung FRK. Spectral Graph Theory. American Mathematical Society. 1997.

[5] Eells J, Lemaire L. Selected Topics in Harmonic Maps. Amer Mathematical Society. 1983.

[6] Guan Z, Bu J, Mei Q, Chen C, Wang C. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. Proc. of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2009. 540–547.

[7] Guan Z, Wang C, Bu J, Chen C, Yang K, Cai D, He X. Document recommendation in social tagging services. Proc. of the 19th International Conference on World Wide Web. 2010. 391–400.

[8] He J, Li M, Zhang H, Tong H, Zhang C. Manifold-ranking based image retrieval. Proc. of the 12th Annual ACM International Conference on Multimedia. 2004. 9–16.

[9] He X, Cai D, Han J. Learning a maximum margin subspace for image retrieval. IEEE Trans. on Knowledge and Data Engineering, 2008, 20(2): 189–201.

[10] He X, Niyogi P. Locality preserving projections. Advances in Neural Information Processing Systems 16. 2003.

[11] Kim K, Steinke F, Hein M. Semi-supervised regression using hessian energy with an application to semi-supervised dimensionality reduction. Advances in Neural Information Processing Systems, 2009, 22: 979–987.

[12] Lades M, Vorbruggen J, Buhmann J, Lange J, von der Malsburg C, Wurtz R, Konen W. Distortion invariant object recognition in the dynamic link architecture. IEEE Trans. on Computers, 1993, 42(3):300–311.

[13] Lee J. Riemannian Manifolds: an Introduction to Curvature. Springer. 1997.

[14] Logan B, Salomon A. A music similarity function based on signal analysis. Proc. of IEEE International Conference on Multimedia and Expo. 2001. 745–748.

[15] H. Lütkepohl. Handbook of Matrices. Wiley. 1996.

[16] Ojala T, Pietik?inen M, Harwood D. A comparative study of texture measures with classification based on featured distributions. Pattern Recognition, 1996, 29(1):51–59.

[17] Page L, Brin S, Motwani R, Winograd T. The Pagerank citation algorithm: bringing order to

the web. [Technical report]. Stanford Digital Library Technologies Project. 1998.

[18] Rubner Y, Tomasi C, Guibas LJ. The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision, 2000, 40(2): 99–121.

[19] Tan S, Bu J, Chen C, Xu B, Wang C, He X. Using rich social media information for music recommendation via hypergraph model. ACM Trans. on Multimedia Computing, Communications, and Applications (TOMCCAP), 2011, 7(1): 22.

[20] von Luxburg U. A tutorial on spectral clustering. Statistics and Computing, 2007, 17(4): 395–416.

[21] Wan X, Yang J, Xiao J. Manifold-ranking based topic-focused multi-document summarization. Proc. of the 20th International Joint Conference on Artifical Intelligence. 2007. 2903–2908.

[22] Zhou D, Weston J, Gretton A, Bousquet O, Scholkopf B. Ranking on data manifolds. 18th Annual Conference on Neural Information Processing Systems. 2003.

[23] Zhou X, Belkin M, Srebro N. An iterated graph laplacian approach for ranking on manifolds. Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2011. 877–885.