Unified Framework for Representing and Ranking

Jim Jing-Yan Wang[a], Halima Bensmail[b,*]

[a] *Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia*
[b] *Qatar Computing Research Institute, Doha 5825,Qatar*

Abstract

In the database retrieval and nearest neighbor classification tasks, the two basic problems are to represent the query and database objects, and to learn the ranking scores of the database objects to the query. Many studies have been conducted for the representation learning and the ranking score learning problems, however, they are always learned independently from each other. In this paper, we argue that there are some inner relationships between the representation and ranking of database objects, and try to investigate their relationships by learning them in a unified way. To this end, we proposed the **U**nified framework for **R**epresentation and **R**anking (UR^2) of objects for the database retrieval and nearest neighbor classification tasks. The learning of representation parameter and the ranking scores are modeled within one single unified objective function. The objective function is optimized alternately with regarding to representation parameter and the ranking scores. Based on the optimization results; iterative algorithms are developed to learn the representation parameter and the ranking scores on a unified way. Moreover, with two different formulas of representation (feature selection and subspace learning), we give two versions of UR2. The proposed algorithms are tested on two challenging tasks - MRI image based brain tumor retrieval and nearest neighbor classification based protein identification. The experiments show the advantage of the proposed unified framework over

*Corresponding author.
Email addresses: jimjywang@gmail.com (Jim Jing-Yan Wang), hbensmail@qf.org.qa (Halima Bensmail)

the state-of-the-art independent representation and ranking methods.

Keywords: Database retrieval, Nearest neighbor classification, Data representation, Ranking score learning

1. Introduction

In the database retrieval and nearest neighbor classification tasks, given a query object, we try to find some relevant objects from a database [1, 2]. The relevant objects here are defined as the objects of the same semantical class. For example, in the brain tumors diagnosis problem, given a tumor region in a Magnetic Resonance Imaging (MRI) image as a query, it could be very helpful for the diagnosis to retrieve tumors of the same pathological category from a brain MRI scans database [3]. While in drug discovery problem, given a query protein, it could also be useful to find the proteins sharing the same specific chemical properties or similar structure as the query protein from a protein database, so that they can be used as sources for the treatment [4]. To this end, in a typical database retrieval system, the feature vectors are usually first extracted from both the query and database objects, and then the query is compared against each database object to compute the similarities or dissimilarities using their feature vectors. Finally, all the database objects will be ranked according to their similarities to the queries in the descending order, and a few number of them with the largest similarities will be returned to the user, or used to make a classification decision. Because the similarity is used for ranking the database objects, it is also called ranking score [5].

Two fundamental problems have been studied widely is the learning of the representations of the objects feature vectors, and the learning of the ranking scores of the database objects to the query, as listed as follows:

- **Representation**: The original features extracted from the objects are usually very high-dimensional, redundant, sometimes noisy, and only occupying a part of the input space. Thus the original features may not capture the semantical information and could not be used directly to retrieve

the relevant objects very well. In this case, it's necessary to represent the feature vectors to another dataspace so that they could be represented better for the retrieval task. Many representation methods can be considered, such as feature selection [6], subspace learning [7], sparse coding [8], nonnegative matrix factorization [9], hashing [10], etc. In this paper, we will focus on the feature selection and subspace learning problem.

  - To handle the redundant and noisy features, **feature selection** is desired. Feature selection assigns different feature weights to different features, so that the useful features will be emphasized while the redundant and noisy features will be restrained [6].
  - To handle the the high-dimension problem of the feature vectors, the **subspace learning** could be employed for dimensionality reduction. Subspace learning maps the input feature vectors into a lower dimensional space, by using an optimal linear mapping matrix [7].

Many feature selection and subspace learning methods have been proposed to refine the original features, which could be classified into two types — supervised and unsupervised representation methods. The supervised method uses the class labels to guide the learning procedure, however, in database retrieval problems, the objects are usually not annotated, thus unsupervised representation is more suitable in this task.

- **Ranking Score Learning**: To compute the ranking score of a database object to a query, a distance or similarity measure could be employed to compare them, such as Euclidean distance, cosine similarity, etc. This type of methods is called pairwise similarity, and they only consider the query and objects to compare, while neglecting the manifold structure of the database. To handle this problem, the manifold ranking (MR) has been proposed by Zhou et al. [11], so that the ranking score could be learned with respect to the manifold structure of the database, which is characterized by a nearest neighbor graph constructed from the database. More-

over, Yang et al. [5] proposed the Local Regression and Global Alignment (LRGA) based ranking method to further improve the manifold ranking by using the local linear regression model for the ranking score learning problem.

The representation parameter is usually learned first, and then used to represent both the query and database objects. Based on the new representation, some ranking score learning algorithm will be applied for the ranking problem. Thus the representation and the ranking are conducted sequentially and independently. An important assumption behind this strategy is that the representation and the ranking are independent from each other, thus the possible inner relationships between them, which is not clear yet, has been ignored. It's very interesting to notice that in [5], Yang et al. has applied the same LRGA model for both ranking and subspace learning. However, this model has been applied to the ranking and subspace learning respectively. In this paper, we argue that the representation and ranking should be considered in an unified way, so that we could investigate the possible relationships between them. Given a representation method, the ranking should be adjusted to the representation parameter. Moreover, given the ranking scores, the representation parameters should also be refined according to the ranking results.

To this end, we try to propose an unified framework for both the representation parameter learning and the ranking score learning, by constructing an unified objective function. The object representations parameterized by representation parameters will be used to compute the ground distances between query and database objects, and the the ground distances will be further used to regularize the ranking scores. At the same time, the ranking score will also be regularized by the manifold structure of the database. In this way, an unified objective function is built. The objective function will be optimized with regard to representation parameter and the ranking score alternately in an iterative algorithm. When the representation parameter is optimized, ranking score will be fixed, and then their role will be switched. Once the representation parameter

is learned in the training procedure, it will be used to represent the new query object and rank the database objects. The contribution of this paper is listed as follows:

1. An unified framework for representation and ranking is proposed. Though we only discuss the feature selection and subspace learning as examples of representation, it could be extended to other representation methods easily, such as sparse coding, nonnegative matrix factorization, etc.
2. An iterative algorithm is proposed for the learning of representation parameters and ranking scores.

The remainder of this paper is organized as follows: In Section 2, we present the unified framework for representing and ranking. In Section 3, we apply the proposed framework to the brain tumor retrieval and nearest neighbor protein classification applications and show the experimental results. The conclusions and future works are given in Section 4.

2. Unified Framework for Representing and Ranking

In this section, we will introduce the novel framework for data object representation and ranking in database retrieval and nearest neighbor classification tasks.

2.1. Objective Function

Suppose we have a database with N database objects, we denote it as $\mathcal{D} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \in \mathbb{R}^P$, where $\mathbf{x}_i = [x_{i1}, \cdots, x_{iP}]^\top \in \mathbb{R}^P$ is the P dimensional feature vector of the i-th database object. Given a query object, we denote it as $\mathbf{y} \in \mathbb{R}^P$, where $\mathbf{y} = [y_1, \cdots, y_P]^\top \in \mathbb{R}^P$ is the P dimensional feature vector the query object. The task of database retrieval is to rank the database objects in $\mathcal{D}$ according to the similarity between $\mathbf{y}$ and each $\mathbf{x}_i \in \mathcal{D}$, and then return then few top ranked ones as retrieval results. To this end, we need to learn the nonnegative ranking score for each $\mathbf{x}_i$, denoted as f_i, as the similarity measure between $\mathbf{y}$ and $\mathbf{x}_i$. The ranking scores of all the database objects are further

organized as a ranking score vector $\mathbf{f} = [f_1, \cdots, f_N]^\top \in \mathbb{R}_+^N$. Moreover, instead of using the original features of query object $\mathbf{y}$ and the database object $\mathbf{x}_i$, we also consider to represent them by feature selection or subspace learning. The represented query and database objects are denoted as $\mathbf{y}^\Theta \in \mathbb{R}^{P'}$ and $\mathbf{x}_i^\Theta \in \mathbb{R}^{P'}$, where Θ is the representation parameter, and P' is the dimension of the feature space of the new representation.

To learn the representation parameter Θ and the ranking score vector $\mathbf{f}$ in an unified way, we will formulate the learning problem by an unified objective function. We will consider the following two regularization terms when constructing the objective function:

Ground distance regularization : Given a query object represented as $\mathbf{y}^\Theta$, and a database object represented as $\mathbf{x}_i^\Theta$, parameterized by Θ, we could compute the squared Euclidean distance between them as the ground distance: $||\mathbf{y}^\Theta - \mathbf{x}_i^\Theta||_2^2$. If the ground distance of query to i-th database object is short, it's natural to expect the ranking score of i-th database objective is large; and vice versa. We model the regularization of ground distance with the following minimization problem:

$$\underset{\mathbf{f}\in\mathbb{R}_+^N,\Theta}{min} \sum_{i=1}^{N} \left\|\mathbf{y}^\Theta - \mathbf{x}_i^\Theta\right\|_2^2 f_i \tag{1}$$

Manifold regularization : Based on the manifold assumption [12], which assumes that all the database objects lie on a low-dimensional manifold, we also try to regularize the ranking scores by manifold information. The manifold can be approximated linearly in a local area of the feature space of the database objects. Therefore, we assume that a database object $\mathbf{x}_i$ can be approximated by linearly reconstructing from its K nearest neighbors $\mathbf{x}_j \in \mathcal{N}_i$, as $\mathbf{x}_i \approx \sum_{j:\mathbf{x}_j\in\mathcal{N}_i} A_{ij}\mathbf{x}_j$, where A_{ij} is the reconstruction coefficient which summarizes the contribution of $\mathbf{x}_j$ to the reconstruction of $\mathbf{x}_i$. Following Locally Linear Reconstruction (LLR) [13], the coefficients $A_{ij}, j = 1, \cdots, N$ could be obtained by minimizing the squared

reconstruction error as:

$$\begin{aligned} \underset{A_{i1},\cdots,A_{iN}}{min} & \left\| \mathbf{x}_i - \sum_{j=1}^{N} A_{ij}\mathbf{x}_j \right\|_2^2 \\ s.t. & \sum_{j=1}^{N} A_{ij} = 1, A_{ij} \geq 0, j = 1, \cdots, N, \\ & A_{ij} = 0, if\ \mathbf{x}_j \notin \mathcal{N}_i \end{aligned} \tag{2}$$

This problem could be solved as a Quadratic programming (QP) problem. The solved reconstruction coefficients are organized in a matrix $A = [A_{ij}] \in \mathbb{R}_+^{N \times N}$. With the reconstruction coefficient matrix, we could formulate the manifold assumption to ranking scores by

$$\underset{\mathbf{f} \in \mathbb{R}_+^N}{min} \sum_{i=1}^{N} \left\| f_i - \sum_{j=1}^{N} A_{ij} f_j \right\|_2^2 \tag{3}$$

By solving this problem, we imply that a ranking score f_i could also be recontracted from the ranking scores f_j of its neighbors $\mathbf{x}_j \in \mathcal{N}_i$. The manifold assumption is imposed to the ranking score by sharing the same local linear reconstruction coefficients A_{ij} between the feature space and the ranking score apace.

By combining the two regularization terms in (1) and (3), we could have the following objective function for the learning of $\mathbf{f}$ and Θ:

$$\underset{\mathbf{f} \in \mathbb{R}_+^N, \Theta}{min} \sum_{i=1}^{N} \left\| \mathbf{y}^{\Theta} - \mathbf{x}_i^{\Theta} \right\|_2^2 f_i + \alpha \sum_{i=1}^{N} \left\| f_i - \sum_{j=1}^{N} A_{ij} f_j \right\|_2^2 \tag{4}$$

where α is a trade-off parameter.

We also suppose we have a query set with M query objects for the training procedure, denoted as $\mathcal{Q} = \{\mathbf{y}_1, \cdots, \mathbf{y}_M\} \in \mathbb{R}^P$, where $\mathbf{y}_k = [y_{k,1}, \cdots, y_{k,P}]^\top \in \mathbb{R}^P$ is the P dimensional feature vector of the k-th data object. When k-th query $\mathbf{y}_k$ is available in the training query set $\mathcal{Q}$, we denote the ranking score vector for the k-th query object as $\mathbf{f}_k = [f_{1k}, \cdots, f_{Nk}]^\top \in \mathbb{R}_+^N$, where y_{ik} is the

ranking score of i-th database object against k-th query object. We define the ranking score matrix as $F = [\mathbf{f}_1, \cdots, \mathbf{f}_M] = [f_{ik}] \in \mathbb{R}_+^{N \times M}$, with its k-th collum as the ranking score vector of k-th query. Then the objective function could be extended to the following one by applying the objective function to each query and summing them up:

$$\min_{F \in \mathbb{R}_+^{N \times M}, \Theta} \sum_{k=1}^{M} \left[\sum_{i=1}^{N} \left\| \mathbf{y}_k^{\Theta} - \mathbf{x}_i^{\Theta} \right\|_2^2 f_{ik} + \alpha \sum_{i=1}^{N} \left\| f_{ik} - \sum_{j=1}^{N} A_{ij} f_{jk} \right\|_2^2 \right] \tag{5}$$

By minimizing the objective function in (5), we try to find the optimal ranking scores for the queries in $\mathcal{Q}$, and the representation parameter Θ for both the query and databases objects in $\mathcal{Q}$ and $\mathcal{D}$ simultaneously.

2.2. Optimization

To optimize the objective function (5), we adopt the alternate optimization strategy. F and Θ will be optimized alternatively in an iterative algorithm, and in each iteration, one of them will be solved or updated, while the other fixed, then their role will be switched.

2.2.1. Optimizing F while fixing Θ

By fixing the representation parameter Θ, and defining the ground distance matrix $D = [d_{ik}^{\Theta}] \in \mathbb{R}^{N \times M}$ with $d_{ik}^{\Theta} = ||\mathbf{y}_k^{\Theta} - \mathbf{x}_i^{\Theta}||_2^2$, the problem (5) could be rewritten in matrix formula as,

$$\begin{aligned} \min_{F \in \mathbb{R}_+^{N \times M}} & \sum_{k=1}^{M} \sum_{i=1}^{N} d_{ik}^{\Theta} f_{ik} + \alpha \sum_{k=1}^{M} \sum_{i=1}^{N} \left\| f_{ik} - \sum_{j=1}^{N} A_{ij} f_{jk} \right\|_2^2 \\ & = Tr(F^\top D) + \alpha Tr\left[F^\top (I - A)^\top (I - A) F \right] \\ & = Tr(F^\top D) + \alpha Tr(F^\top L F) \end{aligned} \tag{6}$$

where $L = I - 2A + A^\top A \in \mathbb{R}^{N \times N}$. We introduce the lagrange multiplier matrix $\Phi = [\phi_{ik}] \in \mathbb{R}^{N \times N}$ for the constrain of $F \in \mathbb{R}_+^{N \times M}$, where ϕ_{ik} is the lagrange

multiplier for constraint $f_{ik} \geq 0$. The lagrange function $\mathcal{L}$ of the optimization problem is

$$\mathcal{L} = Tr(F^\top D) + \alpha Tr(F^\top LF) + Tr(F^\top \Phi) \tag{7}$$

By setting the derivative of L with respect to F to zero, we have

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial F} &= D + 2\alpha LF + \Phi \\ &= D + 2\alpha(I - 2A + A^\top A)F + \Phi = 0 \end{aligned} \tag{8}$$

Using the KKT condition $[\Phi] \circ [F] = 0$, where $[\cdot] \circ [\cdot]$ denotes the element-wise matrix product, we get the following equation:

$$\begin{aligned} &[D + 2\alpha(I + A^\top A)F - 4\alpha AF] \circ [F] = 0 \\ &\Rightarrow [D + 2\alpha(I + A^\top A)F] \circ [F] = [4\alpha AF] \circ [F] \end{aligned} \tag{9}$$

which leads to the following update rule for F

$$F \leftarrow \frac{[4\alpha AF]}{[D + 2\alpha(I + A^\top A)F]} \circ [F] \tag{10}$$

where $\frac{[\cdot]}{[\cdot]}$ denotes the element-wise matrix division.

2.2.2. Optimizing Θ while fixing F

To optimize Θ, we first need to specify the form of data representation which transfer a the original feature vector $\mathbf{x} \in \mathbb{R}^P$ to it newly represented feature vector $\mathbf{x}^\Theta \in \mathbb{R}^{P'}$, which is parameterized by Θ. Here we consider the feature selection and subspace learning as data representation methods, which are introduced as follows:

Feature Selection : Given a P dimensional feature vector $\mathbf{x} = [x_1, \cdots, x_P]^\top$ of a object, not all the features are relevant to the task in hand, and many of them might be noisy features. We try to assign each feature with different feature weight, so that the important features will be emphasized and the noisy features will be restrained. To this end, we introduce the nonnegative feature weight vector $\mathbf{t} = [t_1, \cdots, t_P]^\top \in \mathbb{R}_+^P$ to parameterize

the feature selection, where t_p is the weight for the p-th feature. The constrains $t_p \geq 0$ and $\sum_{p=1}^{P} t_p = 1$ are introduced to $\mathbf{t}$ to prevent the negative weight. The feature vector could then be represented as

$$\begin{aligned} \mathbf{x}^{\Theta} &= [t_1 x_1, \cdots, t_P x_P]^{\top} = diag(\mathbf{t})\mathbf{x}, \\ s.t.\ t_p &\geq 0,\ \sum_{p=1}^{P} t_p = 1,\ p = 1, \cdots, P. \end{aligned} \tag{11}$$

In this case, the representation parameter Θ is $\mathbf{t}$. We apply the feature selection to both the query and the database objects, and then the ground distance between the k-th query object $\mathbf{y}_k$ and the i-th database object $\mathbf{x}_i$ will be computed as

$$||\mathbf{y}_k^{\Theta} - \mathbf{x}_i^{\Theta}||_2^2 \quad = ||diag(\mathbf{t})\mathbf{y}_k - diag(\mathbf{t})\mathbf{x}_i||_2^2 = \sum_{p=1}^{P} t_p^2 (y_{kp} - x_{ip})^2 \tag{12}$$

By replacing $\mathbf{t}$ by Θ, substituting (12) to (5), fixing F and removing the irrelevant term, (5) could be turned to the following optimization problem,

$$\begin{aligned} \min_{\mathbf{t}} & \sum_{k=1}^{M} \sum_{i=1}^{N} \left[\sum_{p=1}^{P} t_p^2 (y_{kp} - x_{ip})^2 \right] f_{ik} \\ &= \sum_{p=1}^{P} t_p^2 e_p \\ s.t. & t_p \geq 0,\ \sum_{p=1}^{P} t_p = 1,\ p = 1, \cdots, P. \end{aligned} \tag{13}$$

where $e_p = \sum_{k=1}^{M} \sum_{i=1}^{N} (y_{kp} - x_{ip})^2 f_{ik}$. This problem could be efficiently solve as a standard QP problem as well.

Subspace Learning : Given the feature vector a data object $\mathbf{x} \in \mathbb{R}^P$, subspace learning [7] tries to map it into a P'-dimension data space by a orthometric transformation matrix $W \in \mathbb{R}^{P \times P'}$ as

$$\begin{aligned} \mathbf{x}^{\Theta} &= W^{\top}\mathbf{x}, \\ s.t.\ & W^{\top}W = I \end{aligned} \tag{14}$$

where I is an identity matrix of order P'. In this case, the representation parameter is W. By applying the subspace learning to both query and database objects, we have the ground distance between $\mathbf{y}_k$ and $\mathbf{x}_i$ defined as

$$||\mathbf{y}_k^{\Theta} - \mathbf{x}_i^{\Theta}||_2^2 \quad = ||W^{\top}\mathbf{y}_k - W^{\top}\mathbf{x}_i||_2^2 = Tr\left[W^{\top}(\mathbf{y}_k - \mathbf{x}_i)(\mathbf{y}_k - \mathbf{x}_i)^{\top}W\right] \tag{15}$$

By replacing Θ by W, substituting (15) to (5), fixing F, and removing the term irrelevant to W, (5) could be turned to the following optimization problem,

$$\begin{aligned} \min_{\mathbf{t}} & \sum_{k=1}^{M}\sum_{i=1}^{N} Tr\left[W^{\top}(\mathbf{y}_k - \mathbf{x}_i)(\mathbf{y}_k - \mathbf{x}_i)^{\top}W\right] f_{ik} \\ & = Tr(W^{\top}EW) \\ s.t. & W^{\top}W = I \end{aligned} \tag{16}$$

where $E = \sum_{k=1}^{M}\sum_{i=1}^{N}(\mathbf{y}_k - \mathbf{x}_i)(\mathbf{y}_k - \mathbf{x}_i)^{\top} f_{ik}$. This problem could be obtained by solving the generalized eigenvalue decomposition problem,

$$E\mathbf{w} = \lambda\mathbf{w} \tag{17}$$

where λ is a eigenvalue and $\mathbf{w} \in \mathbb{R}^P$ is its corresponding eigenvector. Assume that the P' smallest eigenvalues are ranked in a ascending order, as $\lambda_1, \cdots, \lambda_{P'}$, and the corresponding eigenvectors are denoted as $\mathbf{w}_1, \cdots, \mathbf{w}_{P'}$. Then the solution of (16) could be obtained as $W = [\mathbf{w}_1, \cdots, \mathbf{w}_{P'}] \in R^{P\times P'}$.

2.3. Algorithm

Based on the optimization results, we could develop the iterative algorithm for the training procedure of unified object representation parameter Θ and the ranking score matrix F. The algorithm is summarized in Algorithm 1.

Algorithm 1 UR^2: off-line learning algorithm.

Input: Database object set $\mathcal{D} = \{\mathbf{x}_i, \cdots, \mathbf{x}_N\}$.

Input: Query object set $\mathcal{Q} = \{\mathbf{y}_i, \cdots, \mathbf{y}_M\}$.

Construct the nearest neighbor graph for $\mathcal{D}$ and compute its reconstruction coefficient matrix A.

Initialize the ranking score matrix F^0.

Initialize the representation parameter Θ^0 and compute the initial ground distance matrix D^0.

for $t = 1, \cdots, T$ **do**

Update the ranking score matrix F^t based on the previous ground distance matrix D^{t-t} and ranking score matrix F^{t-1}, as in (10).

Update the representation parameter Θ^t by fixing F^t, as in (13) or (17).

Update the ground distance matrix D^t based on the newly updated representation parameter Θ^t.

end for

Output: The ranking score matrix F^T, and the representation parameter Θ^T.

2.4. Ranking new query object

We have introduced the off-line training procedure of Θ given a set of training query objects. In this subsection, we will discuss how to represent and rank a new query object $\mathbf{y}$ in the on-line retrieval procedure. In fact, we assume that the new arrived query won't effect the representation parameter, and we use the parameter Θ learned using the training query objects to represent it as $\mathbf{y}^{\Theta}$, based on feature selection or subspace learning. To learn its ranking score

vector $\mathbf{f}$, we simply solve the optimization problem in (4) while fixing Θ as learned by Algorithm 1. We define a ground distance vector for $\mathbf{y}^{\Theta}$ against all the represented database objects as $\mathbf{d} = [d_1, \cdots, d_N]^\top \in \mathbb{R}^N$, where $d_i = ||\mathbf{y}^{\Theta} - \mathbf{x}_i^{\Theta}||_2^2$. (4) then could be rewritten as

$$\min_{\mathbf{f} \in \mathbb{R}_+^N} \mathbf{f}^\top \mathbf{d} + \alpha \mathbf{f}^\top L \mathbf{f} \tag{18}$$

Its lagrange function $\mathcal{L}$ of is

$$\mathcal{L} = \mathbf{f}^\top \mathbf{d} + \alpha \mathbf{f}^\top L \mathbf{f} + \mathbf{f}^\top \boldsymbol{\phi} \tag{19}$$

where $\boldsymbol{\phi} \in \mathbb{R}^N$ is the lagrange multiplier vector for constrain $\mathbf{f} \geq 0$. By setting the derivative of $\mathcal{L}$ with respect to $\mathbf{f}$ to zero, we have

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{f}} &= \mathbf{d} + 2\alpha L \mathbf{f} + \boldsymbol{\phi} \\ &= \mathbf{d} + 2\alpha(I - 2A + A^\top A)\mathbf{f} + \boldsymbol{\phi} = 0 \end{aligned} \tag{20}$$

Using the KKT condition $[\boldsymbol{\phi}] \circ [\mathbf{f}] = 0$, we get the following equation:

$$\begin{aligned} &[\mathbf{d} + 2\alpha(I + A^\top A)\mathbf{f} - 4\alpha A \mathbf{f}] \circ [\mathbf{f}] = 0 \\ &\Rightarrow [\mathbf{d} + 2\alpha(I + A^\top A)\mathbf{f}] \circ [\mathbf{f}] = [4\alpha A \mathbf{f}] \circ [\mathbf{f}] \end{aligned} \tag{21}$$

which leads to the following update rule for $\mathbf{f}$

$$\mathbf{f} \leftarrow \frac{[4\alpha A \mathbf{f}]}{[\mathbf{d} + 2\alpha(I + A^\top A)\mathbf{f}]} \circ [\mathbf{f}] \tag{22}$$

Based on the update rule, we could have the on-line ranking algorithm for query $\mathbf{y}$, as summarized in Algorithm 2.

3. Experiments

In this experiment, we will evaluate the proposed methods for the brain tumor retrieval task and the nearest neighbor protein identification task.

Algorithm 2 UR^2: on-line ranking algorithm.

Input: Database object set $\mathcal{D} = \{\mathbf{x}_i, \cdots, \mathbf{x}_N\}$ with its Laplacian matrix L.

Input: Query object $\mathbf{y}$.

Input: The representation parameter Θ.

Initialize the ranking score vector $\mathbf{f}^0$.

Compute the ground distance vector $\mathbf{d}$ based on Θ.

for $t = 1, \cdots, T$ **do**

Update the ranking score vector $\mathbf{f}^t$ based on the ground distance vector $\mathbf{d}$ and previous ranking score vector $\mathbf{f}^{t-1}$ as in (22).

end for

Output: The ranking score vector $\mathbf{f}^T$.

3.1. Experiment I: Brain Tumor Retrieval

MRI has been one of the the most popular means for the diagnose of human brain tumors. However, the diagnosis of a brain tumor relies strongly on the experience of radiologists. In clinical practice, it would be significant helpful to have a retrieval system for brain tumors in MRI image which could return the tumors of the same pathological category as the query image. The doctors then can use the relevant MRI images returned by the retrieval system and the diagnosis information associated to these relevant images for the diagnosis for the current case [3]. In this experiment, we will evaluate the proposed method as MRI image representation and ranking method for the brain tumor retrieval system.

3.1.1. Dataset and Setup

Three types of brain tumors have been studied widely due to their high incidence rate in clinics, which are gliomas, meningiomas, and pituitary tumors. In this experiment, we use a dataset of 1014 MRI slices of the three types of brain tumors. There are 220 MRI slices of meningiomas, 475 MRI slices of gliomas, and 319 MRI slices of pituitary tumors in the dataset. The tumor regions in the images were manually outlined by drawing the the tumor boundaries. In

this experiment, we define two tumor region as relevant if they contains tumors of the same type, otherwise, they are defined irrelevant. Given a query tumor region, the brain tumor retrieval task is to retrieve relevant tumor regions from the database. To this end, we extract visual features from the tumor region, including the following ones:

- **Intensity Features**: To extract the intensity features from the tumor region, we calculate the mean and variance of the normalized intensities of the tumor region pixels.

- **Texture Features**: To extract the texture feature from the tumor region, we first calculate the Gray Level Co-occurrence Matrix (GLCM) and wavelet coefficients, and then some statistical parameters including mean, variance, entropy, correlation, etc, are estimated and used as texture features.

- **Shape Features**: To extract the shape features from the tumor region, we first calculate the shape signature from the points of the tumor boundary by using the radial distance, then perform the wavelet decomposition to the shape signature, and finally compute the mean and variance of the wavelet coefficients in each sub-band as shape features.

- **Bag-of-Words Features**: We also employ the bag-of-words model to extract the visual features from the tumor region. The key points are first detected, then the Scale-Invariant Feature Transform (SIFT) descriptor of each key points are calculated as "words", and finally they are quantized to a dictionary and the quantization histogram is used as the bag-of-words feature.

All these features will be concatenated to obtain the visual feature vector of each brain tumor region in the MRI image. Using the proposed method, we perform the feature selection or subspace learning to the visual feature vector of query and database tumor regions to obtain the new representations, and learn the ranking scores of the database tumor regions according to the query tumor

region for the ranking problem. Based on the ranking scores, the database tumor regions are ranked in a descending order of the ranking score, and the top few ones will be returned as relevant ones.

To conduct the experiment, we need a database, a training query set used to learn the representation, and a test query set to evaluate the retrieval performance. To this end, we randomly split the entire dataset into three subsets, one with 50% slices as database, one with with 25% slices as training query set, and another one with 25% slices as test query set. The database training query test qeury set split will be repeated randomly for ten times to reduce the bias of each split.

To evaluate the retrieval performances, we used the Receiver Operating Characteristic (ROC) and the recall-recision curves. The ROC curve is created by plotting True Positive Rates (TPR) against the False Positive Rates (FPR) of different numbers of returned tumors. The recall-precision curve is created by plotting precision against recall of different numbers of returned tumors. The TPR, FPR, precision and recall are defined as follows:

$$\begin{aligned} TPR = \frac{TP}{TP+FN}, FPR = \frac{FP}{FP+TN}, \\ precision = \frac{TP}{TP+FP}, recall = \frac{TP}{TP+FN} \end{aligned} \tag{23}$$

where TP is the number of returned tumors relevant to the query, TN is the number non-returned tumors irrelevant to the query, FP is the number of returned tumors irrelevant to the query, while FN is the non-returned tumors relevant to the query. Besides the curves, we also employ the Area Under the ROC Curve (AUC) and the Mean Average Precision (MAP) as the single measures for the retrieval task.

3.1.2. Results

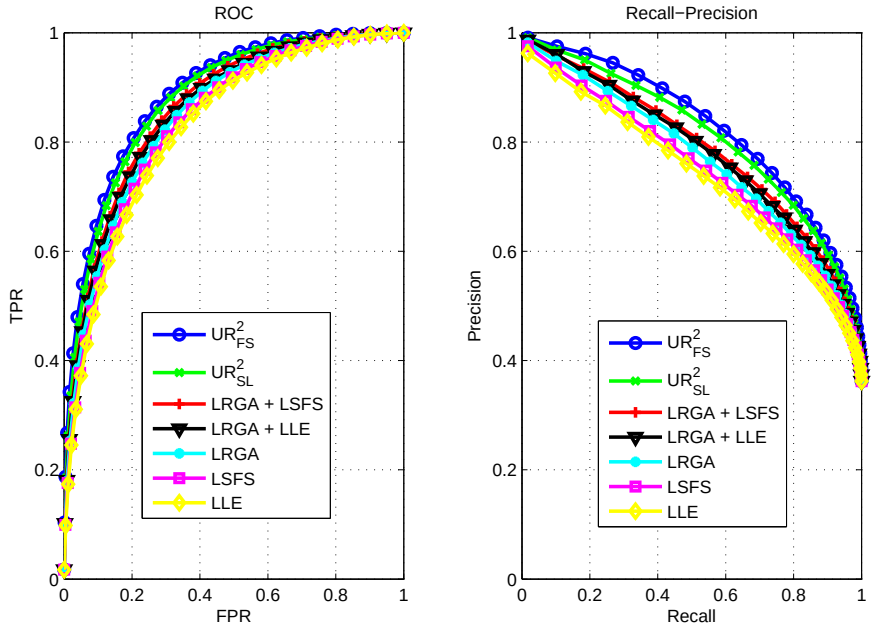
In the experiments, we compare our unified framework for both representation and ranking of tumor region against several representation and ranking methods. The UR^2 method with Feature Selection is donated as UR^2_{FS}, and UR^2 method with Subspace Learning is donated as UR^2_{SL}. Since our methods

are based on manifold learning of ranking score and representation parameters, we compare them against several manifold-based ranking and presentation methods, including:

- a feature selection method, Laplacian Score for Feature Selection (LSFS) [14],
- a subspace learning method, Locally Linear Embedding (LLE) [15],
- a ranking score learning method, LRGA [16], and
- the naive combinations of LRGA with LSFS and LLE respectively, denoted as "LRGA+LSFS" and "LRGA+LLE".

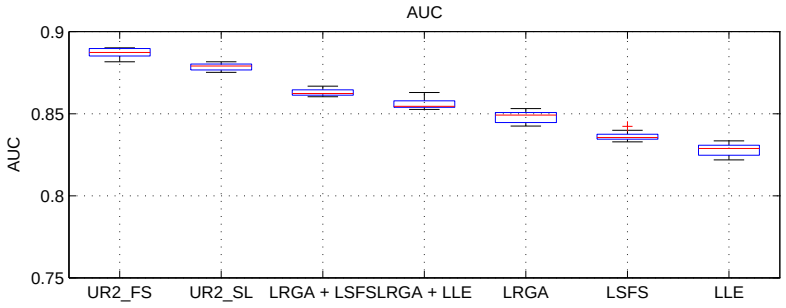
Figure. 1 show the results (average ROC and recall-precision curves) obtained by applying our methods UR^2_{FS} and UR^2_{SL} to the tumor region retrieval problem compared to other manifold-based representation and ranking score methods with tensity, texture, shape and bag-of-word histogram features. LLE has been chosen as a baseline since it has been extensively used in previous manifold learning works. Figure. 1 confirms the advantages of unfired representation and ranking approaches w.r.t. competing methods. For example, in the case of ROC our UR^2_{FS} outperforms other methods consistently with different FPR values, which is followed by UR^2_{SL}. In the case of recall-precision curve, UR^2_{FS} is more closer to the top right corner of the figure than any other methods. We should note that the proposed unified framework outperform not only the independent presentation and ranking methods (LRGA, LSFS and LLE), but also their naive combinations (LRGA+LSFS and LRGA+LLE). We explain this with the fact that our approaches, differently from other independent representation and ranking methods, take into account both representation and ranking problems simultaneously, so that the representation parameters and ranking scores could be learned optimally. Moreover, it is worth noting that the manifold ranking method (LRGA) outperforms the feature selection and subspace learning methods (LSFS and LLE) with pairwise distance as similarities, which highlights the importance of considering the manifold structure of the

database when ranking. It's also interesting to notice that for this task in hand, feature selection works better than subspace learning. The possible reason is that we have extracted many visual features from the tumor region while only few of them are relevant to the pathological type of the tumors. Similar conclusions can be made for the AUC and MAP values of the methods (see boxplots of AUC and MAP in Figure. 2). Also in this case the unified approaches of representation and ranking outperform independent representation and ranking methods.

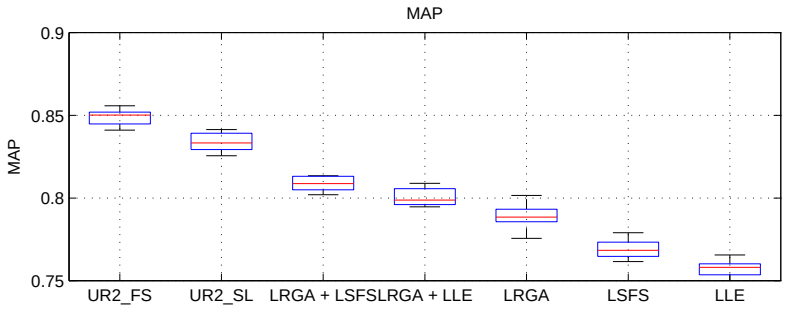


(a) ROC (b) Recall-Precision

Figure 1: The ROC and recall-precision curves on brain tumor retrieval problem.



(a) AUC



(b) MAP

Figure 2: The AUC and MAP values on brain tumor retrieval problem.

3.2. Experiment II: Protein Identification

Identification the protein sample by using bio-sensor is very important for biochemical research and disease diagnose. In this experiment, we will evaluate the usage of proposed methods for the nearest neighbor classification based identification using the bio-sensor array data.

3.2.1. Dataset and setup

In this experiment, we collect a dataset of 100 protein samples, belonging to 9 different proteins. The 9 proteins are SubtilisinA (Sub), Fibrinogen (Fib), Hemoglobin (Hem), Cytochrome C (Cyt), Lysozyme (Lys), Horseradish perox-

idase (Hor), Bovine serum albumin (Bov), Lipase (Lip) and Casein (Cas). The sample number of each protein varies from 6 to 16. The distribution of sample number of different proteins is shown in Figure 3.

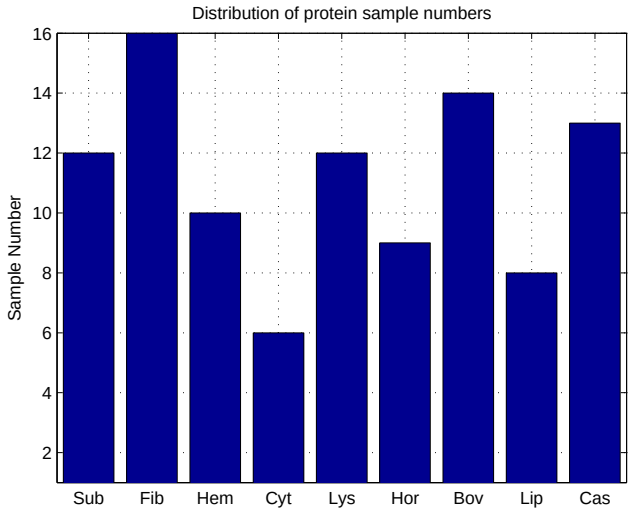


Figure 3: Distribution of protein sample numbers in the protein identification dataset.

Given an unknown sample, the task of protein identification is to classify the sample into one of the nine proteins in the training set. To this end, each sample will be tested against a bio-sensor array developed by Pei et al [17], called adaptive ensemble aptamers (ENSaptamers) which exploits the collective recognition abilities of a small set of rationally designed, nonspecific DNA sequences. The seven fluorescence intensities of a sample generated by seven ENSaptamers of the bio-sensor array are used as the original features and organized as a seven-dimensional feature vector. Then the feature vector of the query sample will be compared against all the feature vectors of the training samples in the datable and the most similar ones will be used for nearest neighbor classification.

To test the proposed methods, we employ the leave-one-out protocol to conduct the experiment. Each sample in the dataset will be used as a query sample in turns, while the remaining ones as training set. The training set will be

further divided into training query set and database to learn the representation parameter. The training query set will contains 40% samples of the entire training set, while the database will contains 60% of the training samples. Once the representation parameter is learned by using the training set, it will be used to represent the query and the training samples. For the nearest neighbor classification of the query, the entire training set will be used as database. The ranking score of the database samples will be learned w.r.t the query, the ones with largest ranking scores will be returned and the query's class label will be obtained by major voting of the returned samples.

The classification results are evaluated by the average classification accuracies of all the queries, which is defined as

$$Accuracy = \frac{Number\ of\ correctly\ classified\ queries}{Total\ Number\ of\ queries} \tag{24}$$

By varying the number of returned samples from the database, we could have different accuracies. The classification results will be reported using the curves of the accuracies against the returned sample numbers.

3.2.2. Results

The accuracies of different methods with different different returned sample numbers are shown in Figure 4. It can be seen that both UR^2_{FS} and UR^2_{SL} perform better than the best results of other methods at most cases, with UR^2_{SL} getting the overall best results. The combination of LLE/ LSFS and LRGA performs better than using individual representation or ranking methods, but could not beat the proposed unified framework. It indicates that using presentation and ranking methods together could boost the nearest neighbor classification performance, but the way to combine them is also very important. It's also interesting to notice that UR^2_{SL} outperforms UR^2_{FS} in this experiment, indicating that all the seven features of seven ENSaptamers are useful for the protein identification problem. This fact could also be verified by the fact that LLE outperforms LSFS. Moreover, it could be observed that when the returned sample number is small, the classifications are a kind of stable. However, when the re-

turned sample number is larger than 20, the classifications decrees significantly. This is because that for each query, there are at most 15 samples of the same protein in the database, which is defined as relevant to the query. When more than 15 samples are returned, the irrelevant samples will increase significantly and dominate the major voting of the nearest neighbor classification.

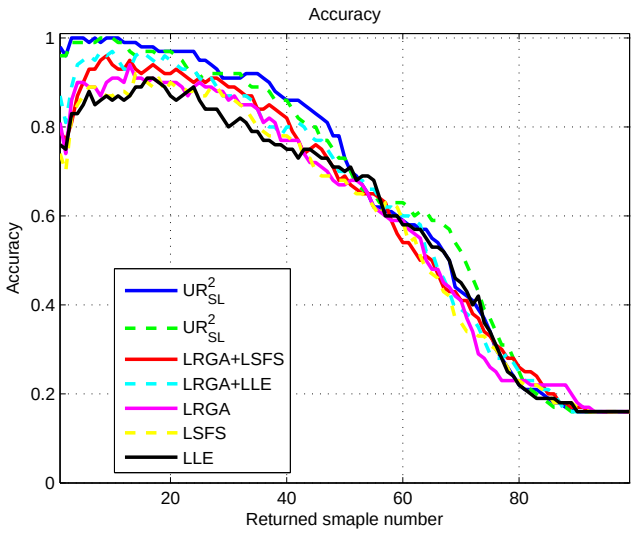


Figure 4: Curves of Accuracies against different returned sample numbers.

4. Conclusion and Future works

Representation learning and ranking score learning are two foundational problems for similar neighbor finding with many significant applications including database retrieval and nearest classification. Most research in the machine learning community have been focussed on the learning of representation parameters and ranking score respectively, which ignores the possible relationships between these two issues at all. In this paper, for the first time, we propose the unified framework for representation and ranking objects in database retrieval and nearest classification problems. It is shown in this work that using the proposed unified framework to learn the representation and raking parameters

works well in this scenario. A significant advantage of the proposed method, as compared to methods to represent and rank objects respectively, is that, with different representation parameter to define the ground distance, the optimal ranking scores could be learned according to the representation parameter. Moreover, the representation parameter could also be adjusted according to the ranking scores.

For the future works, we would consider using sparse coding as the representation method instead of features selection and subspace learning, which is the stat-of-the-art representation method. Moreover, the optimization of the ranking score could possibly has close form, which is another direction desired to explore.

References

[1] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1349–1380.

[2] T. Denoeux, k-nearest neighbor classification rule based on dempster-shafer theory, IEEE Transactions on Systems, Man and Cybernetics 25 (5) (1995) 804–813.

[3] W. Yang, Q. Feng, M. Yu, Z. Lu, Y. Gao, Y. Xu, W. Chen, Content-based retrieval of brain tumor in contrast-enhanced MRI images using tumor margin information and learned distance metric, Medical Physics 39 (11) (2012) 6929–6942. doi:10.1118/1.4754305.

[4] K. Marsolo, S. Parthasarathy, On the use of structure and sequence-based features for protein classification and retrieval, KNOWLEDGE AND INFORMATION SYSTEMS 14 (1) (2008) 59–80. doi:10.1007/s10115-007-0088-0.

[5] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, Y. Pan, A Multimedia Retrieval Framework Based on Semi-Supervised Ranking and Relevance Feedback,

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 34 (4) (2012) 723–742. doi:10.1109/TPAMI.2011.170.

[6] Y. Sun, S. Todorovic, S. Goodison, Local-learning-based feature selection for high-dimensional data analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1610–1626.

[7] F. De La Torre, M. Black, A framework for robust subspace learning, International Journal of Computer Vision 54 (1-3) (2003) 117–142.

[8] H. Lee, A. Battle, R. Raina, A. Ng, Efficient sparse coding algorithms, 2007, pp. 801–808.

[9] J.-Y. Wang, I. Almasri, X. Gao, Adaptive graph regularized Nonnegative Matrix Factorization via feature selection, in: 2012 21st International Conference on Pattern Recognition (ICPR 2012), 2012, pp. 963–6.

[10] M. Datar, P. Indyk, N. Immorlica, V. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, 2004, pp. 253–262.

[11] D. Zhou, J. Weston, A. Gretton, O. Bousquet, B. Scholkopf, Ranking on data manifolds, in: Thrun, S and Saul, K and Scholkopf, B (Ed.), ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 16, Vol. 16, 2004, pp. 169–176.

[12] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, Journal of Machine Learning Research 7 (2006) 2399–2434.

[13] L. Zhang, C. Chen, J. Bu, D. Cai, X. He, T. S. Huang, Active Learning Based on Locally Linear Reconstruction, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 33 (10) (2011) 2026–2038. doi:10.1109/TPAMI.2011.20.

[14] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems 18, 2005.

[15] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323+. doi:10.1126/science.290.5500.2323.

[16] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, Y. Pan, A Multimedia Retrieval Framework Based on Semi-supervised Ranking and Relevance Feedback, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 723–42.

[17] H. Pei, J. Li, M. Lv, J. Wang, J. Gao, J. Lu, Y. Li, Q. Huang, J. Hu, C. Fan, A Graphene-Based Sensor Array for High-Precision and Adaptive Target Identification with Ensemble Aptamers, Journal of the American Chemical Society 134 (33) (2012) 13843–13849. doi:10.1021/Ja305814u.