Concise Papers

Local Kernel Regression Score for Selecting Features of High-Dimensional Data

Yiu-ming Cheung, *Senior Member*, *IEEE*, and
Hong Zeng, *Student Member*, *IEEE*

Abstract—In general, irrelevant features of high-dimensional data will degrade the performance of an inference system, e.g., a clustering algorithm or a classifier. In this paper, we therefore present a Local Kernel Regression (LKR) scoring approach to evaluate the relevancy of features based on their capabilities of keeping the local configuration in a small patch of data. Accordingly, a score index featuring applicability to both of supervised learning and unsupervised learning is developed to identify the relevant features within the framework of local kernel regression. Experimental results show the efficacy of the proposed approach in comparison with the existing methods.

Index Terms—Relevant features, feature selection, local kernel regression score, high-dimensional data.

1 Introduction

High-dimensional data consisting of a large number of features (also called *attributes* in community) are common in the inference problems of a variety of scientific areas, e.g., computer vision, pattern recognition, gene expression array analysis, and so forth. In general, some features may be noisy and irrelevant to the inference. The inclusion of them in an inference system may not only make the computational cost heavier, but also degrade the inference accuracy to a certain degree because of the curse of dimensionality. Undoubtedly, it is very crucial for an inference system to identify the relevant features and reduce the dimensionality of data then. Hereinafter, we concentrate on an inference system that is a clustering algorithm or a classifier only, both of which are learned to separate the data points into different clusters/classes so that the data points in the same cluster/class are similar under a metric, and those points in different clusters/classes are not.

To identify the relevant features, there are three kinds of the feature selection methods [1], namely, *wrapper*, *embedded*, and *filter* approaches. The *wrapper* repeatedly wraps the candidate feature subsets and utilizes the performance of the learning algorithm to evaluate the quality of them. Similarly, the *embedded* approach utilizes the intermediate outputs of the employed inference algorithm to evaluate the candidate feature subset and often requires a laborious iterative optimization process. Evidently, the process of evaluating the candidate feature subset in the *wrapper* and *embedded* methods is quite time-consuming. In contrast, the *filter* methods usually select the features independent of the succeeding inference phases. Hence, such a method is generally more efficient for feature selection.

In the literature, the supervised *filter* approaches have been extensively studied, e.g., χ^2-test, Fisher score, ReliefF [8], and so forth, but the unsupervised counterparts become more challenging because of no true class labels available in the feature selection. To the best of our knowledge, unsupervised feature selections have not been well studied yet in the past decades. Until most recently, several unsupervised *filter* methods have been proposed to select the relevant features based on the intrinsic properties of the data. For instance, Wolf and Shashua [12] proposed the Q-α algorithm, which builds on the spectral properties of the graph Laplacian of data on the candidate feature subset, and iteratively calculates the soft cluster indicator matrix and the feature weights. Although an interesting property of sparsity in feature weights naturally emerges, the computation of iterative optimization will be quite laborious, particularly in the presence of thousands of features. Under the circumstances, He et al. [5] proposed a more computation-efficiency method, namely Laplacian score, which also takes advantage of the graph Laplacian, but selects the features by ranking their capabilities of preserving the locality in the graph. Furthermore, Zhao and Liu [16] developed a more general spectral feature selection framework, which includes the Laplacian score as a special case.

Recently, the local regression technique that essentially minimizes the regression error of the dependent variable in a local space has been applied to a variety of learning problems. The promising results in [13], [14], [15] have shown that the local regression is effective in exploring the local structure of data. In this paper, we therefore present a new feature selection method named *Local Kernel Regression* (LKR) score, which is fundamentally based on the kernel ridge regression and the neighborhood graph, to select the relevant features from the high-dimensional data in both unsupervised and supervised manner. In our method, it is assumed that the feature value of each point from a small patch of a graph should be well estimated by using the feature values of its neighbors. The estimation error indicates the capability of a feature to preserve the local similarities of the data points. We herein adopt a local kernel regression model to make this estimation and further show the relationship between the proposed LKR score and the Laplacian one [5] from the perspective of local kernel regression. Experimental results demonstrate the efficacy of the proposed method.

The remainder of the paper is organized as follows: Section 2 overviews the kernel ridge regression and the neighborhood graph, respectively. Section 3 describes the proposed LKR score algorithm for feature selection in detail. Section 4 shows the relationship between the LKR score and the Laplacian one. The experimental results are presented in Section 5. Finally, we draw a conclusion in Section 6.

2 Overview of the Kernel Ridge Regression and Neighborhood Graph

We first introduce the notations used in this paper. Given an input variable $\mathbf{x} = [x^{(1)}, x^{(2)}, \ldots, x^{(\mathrm{l})}, \ldots, x^{(\mathrm{d})}]^T$, where T is the transpose operator of a matrix, each element $x^{(\mathrm{l})}$ of $\mathbf{x}$ is called a feature. Suppose the N samples of $\mathbf{x}$ are $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, and their corresponding class labels are $y_1, y_2, \ldots, y_N$. We let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be the set of N samples, and $\mathbf{y} = [y_1, \ldots, y_N]^T$. $\mathcal{N}_i$ denotes the set of neighboring points of $\mathbf{x}_i$, $1 \leq i \leq N$, and $n_i = |\mathcal{N}_i|$ is the number of neighboring points of $\mathbf{x}_i$. Furthermore, $\mathbf{f}_l = [f_l^{(1)}, \ldots, f_l^{(N)}]^T = [x_1^{(l)}, \ldots, x_N^{(l)}]^T$ with $l = 1, 2, \ldots, d$, is the lth feature vector, i.e., the vector of the N samples of the lth feature, where $f_l^{(i)} = x_i^{(l)}$ is the ith element of $\mathbf{f}_l$. We let $\Phi_{LKR}(\mathbf{f}_l)$ denote the LKR score of the lth feature vector $\mathbf{f}_l$.

2.1 Kernel Ridge Regression

Kernel ridge regression is a simple yet very effective tool for building nonlinear regression model [9]. Given the training data

- *The authors are with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong SAR, China. E-mail: {ymc, hzeng}@comp.hkbu.edu.hk.*

Manuscript received 17 Feb. 2008; revised 5 Oct. 2008; accepted 8 Oct. 2008; published online 8 Jan. 2009.
Recommended for acceptance by S. Singh.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-02-0097.
Digital Object Identifier no. 10.1109/TKDE.2009.23.

1041-4347/09/$25.00 © 2009 IEEE Published by the IEEE Computer Society

$\{(\mathbf{x}_i, t_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ is an input sample data and $t_i \subset \mathbb{R}$ is the real-valued target value of the corresponding output, its task is to find a function to map $\mathcal{X}$ to $\mathbb{R}$ under the measurement of least square error. The predictive model of kernel ridge regression can be expressed as

$$g(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \mathcal{K}(\mathbf{x}, \mathbf{x}_i), \tag{1}$$

where $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a kernel function and α_is are the estimation coefficients. α_is can be solved by minimizing the following objective function that consists of the fitness item and the Tikhonov regularization item [11]:

$$\|\mathbf{K}\boldsymbol{\alpha} - \mathbf{t}\|^2 + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}, \tag{2}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T \in \mathbb{R}^N$, $\mathbf{t} = [t_1, \ldots, t_N]^T$, λ is a small positive regularization parameter, and $\mathbf{K} = [k_{ij}]_{N \times N}$ with $k_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel matrix over all the training samples $\mathcal{X}$. By minimizing the objective function with respect to $\boldsymbol{\alpha}$, the solution of (2) is given by

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{t}, \tag{3}$$

where $\mathbf{I}$ is an $N \times N$ unit matrix. As a result, the kernel ridge regression model of (1) can be expressed as

$$g(\mathbf{x}) = \mathbf{k}_x^T \boldsymbol{\alpha} = \mathbf{k}_x^T (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{t}, \tag{4}$$

where $\mathbf{k}_x = [\mathcal{K}(\mathbf{x}, \mathbf{x}_1), \ldots, \mathcal{K}(\mathbf{x}, \mathbf{x}_N)]^T \in \mathbb{R}^N$.

2.2 The Unsupervised and Supervised Neighborhood Graph

Given the data set $\mathcal{X}$, let $\mathbf{G}(\mathbf{V}, \mathbf{E})$ be the undirected graph constructed from $\mathcal{X}$, where the ith vertex v_i of $\mathbf{G}$ corresponds to $\mathbf{x}_i \in \mathcal{X}$, $\mathbf{V}$ is the set of vertices, and $\mathbf{E}$ is the set of edges. The $\mathbf{G}$ is constructed as follows: If v_i is the neighbor of v_j, or v_j is the neighbor of v_i $(i \neq j)$, an edge is drawn between vertex i and vertex j. For the *unsupervised* case, the *neighborhood* of v_i can be defined as its k nearest neighbors (excluding v_i itself) of a data according to a certain metric of distance, e.g., the euclidean distance used in this paper. Let $\mathbf{W}$ be the symmetric $N \times N$ weight matrix, where w_{ij} is the weight of the edge connecting vertex i and vertex j. The weight w_{ij} can be calculated by

$$w_{ij} = \begin{cases} \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_i \in \mathcal{N}_j \text{ } or \text{ } \mathbf{x}_j \in \mathcal{N}_i; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

As for the *supervised* case, the *neighborhood* can be defined as those vertices that share the same class labels, and the weight w_{ij} is calculated as

$$w_{ij} = \begin{cases} \dfrac{1}{N_l}, & \text{if } y_i = y_j = l, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where N_l denotes the number of data points in Class l. Furthermore, the degree matrix $\mathbf{D}$ of the graph $\mathbf{G}$ is defined as $D_{ij} = \sum_{m=1}^{N} w_{im}$, if $i = j$ and 0 otherwise, where D_{ij} is the (i, j)th element of $\mathbf{D}$. According to the spectral graph theory [2], the density around $\mathbf{x}_i$ can be approximated by D_{ii}. The more points are close to $\mathbf{x}_i$, the larger D_{ii} is.

3 The Local Kernel Regression Score for Feature Selection

Since the within-cluster and within-class similarities are very useful for the data discrimination, it is reasonable to select the features that keep such similarities or configurations within a small patch on the graph. We, therefore, measure the relevancy of features using this criterion. Practically, a quantitative measurement for such criterion can be realized by examining how well the feature value of each point can be estimated based on its neighbors. In this paper, we utilize a local kernel ridge regression to implement the estimation. Given the training data $\mathbf{x}_i$ and $\{(\mathbf{x}_j, f_l^{(j)})\}_{\mathbf{x}_j \in \mathcal{N}_i}$, we would like to train a kernel ridge regression model locally to approximate $f_l^{(i)}$, where $f_l^{(j)}$ plays the role as the real-valued target output of $\mathbf{x}_j$ for learning this kernel machine. Based on (4), we use the following equation to denote the local kernel ridge regression model at $\mathbf{x}_i$:

$$g_{\mathcal{N}_i}(\mathbf{x}_i) = \mathbf{k}_{\mathcal{N}_i}^T (\mathbf{K}_{\mathcal{N}_i} + \lambda \mathbf{I})^{-1} \mathbf{f}_l^{(\mathcal{N}_i)}, \tag{7}$$

where $g_{\mathcal{N}_i}(.)$ denotes the regression model learned with the training data $\{(\mathbf{x}_j, f_l^{(j)})\}_{\mathbf{x}_j \in \mathcal{N}_i}$, $\mathbf{k}_{\mathcal{N}_i} \in \mathbb{R}^{n_i}$ denotes the vector of $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$s for all $\mathbf{x}_j \in \mathcal{N}_i$, $\mathbf{f}_l^{(\mathcal{N}_i)} \in \mathbb{R}^{n_i}$ denotes the vector of $f_l^{(j)}$s for all $\mathbf{x}_j \in \mathcal{N}_i$, $\mathbf{K}_{\mathcal{N}_i} \in \mathbb{R}^{n_i \times n_i}$ is the kernel matrix over $\mathbf{x}_j \in \mathcal{N}_i$, i.e., $\mathbf{K}_{\mathcal{N}_i} = [\mathcal{K}(\mathbf{x}_p, \mathbf{x}_q)]$, for $\mathbf{x}_p$, $\mathbf{x}_q \in \mathcal{N}_i$, and $\mathbf{I}$ is an $n_i \times n_i$ unit matrix. We let

$$\boldsymbol{\beta}_{\mathcal{N}_i}^T = \mathbf{k}_{\mathcal{N}_i}^T (\mathbf{K}_{\mathcal{N}_i} + \lambda \mathbf{I})^{-1}, \tag{8}$$

where $\boldsymbol{\beta}_{\mathcal{N}_i} \in \mathbb{R}^{n_i}$ is independent of $\mathbf{f}_l^{(\mathcal{N}_i)}$. Equation (7) can then be rewritten in a linear form:

$$g_{\mathcal{N}_i}(\mathbf{x}_i) = \boldsymbol{\beta}_{\mathcal{N}_i}^T \mathbf{f}_l^{(\mathcal{N}_i)}. \tag{9}$$

We now introduce a new vector $\boldsymbol{\beta}_i \in \mathbb{R}^N$, whose jth $(j = 1, 2, \ldots, N)$ element, denoted as β_{ij}, is calculated as follows:

$$\beta_{ij} = \begin{cases} \textit{the corresponding element of } \boldsymbol{\beta}_{\mathcal{N}_i} \textit{ in } (8), & \text{if } \mathbf{x}_j \in \mathcal{N}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Note that $\mathbf{f}_l^{(\mathcal{N}_i)}$ is a subvector of $\mathbf{f}_l$, we rewrite (9) in a full vector form:

$$g(\mathbf{x}_i) = \boldsymbol{\beta}_i^T \mathbf{f}_l = \sum_{j=1}^{N} \beta_{ij} f_l^{(j)}. \tag{11}$$

Therefore, the local estimation error for the lth feature at $\mathbf{x}_i$ is computed as

$$E_{local}\left(f_l^{(i)}\right) = \left(f_l^{(i)} - g(\mathbf{x}_i)\right)^2 = \left(f_l^{(i)} - \sum_{j=1}^{N} \beta_{ij} f_l^{(j)}\right)^2. \tag{12}$$

In general, the data density often varies over the whole data set. That is, some points may reside in a dense region, while the others may not. The importance of each point may not be the same. Under the circumstances, we compute the overall estimation error over the data manifold as a data-density weighted sum. That is,

$$\begin{aligned} E_{local}(\mathbf{f}_l) &\propto \sum_{i=1}^{N} \left(f_l^{(i)} - \sum_{j=1}^{N} \beta_{ij} f_l^{(j)}\right)^2 D_{ii} \\ &= \sum_{i=1}^{N} \left[\sqrt{D_{ii}} f_l^{(i)} - \sum_{j=1}^{N} \left(\beta_{ij} \sqrt{\frac{D_{ii}}{D_{jj}}}\right) \sqrt{D_{jj}} f_l^{(j)}\right]^2 \\ &= \mathbf{f}_l^T \mathbf{D}^{\frac{1}{2}} (\mathbf{I} - \mathbf{B})^T (\mathbf{I} - \mathbf{B}) \mathbf{D}^{\frac{1}{2}} \mathbf{f}_l, \end{aligned} \tag{13}$$

where $\mathbf{B} = [\beta_{ij} \sqrt{\frac{D_{ii}}{D_{jj}}}]_{N \times N}$ is a sparse matrix. Apparently, we should prefer the feature that makes the error value of (13) as small as possible. However, it can be seen that the zero vector is a trivial candidate that makes the error value of (13) the smallest. Furthermore, the feature vector, whose elements are nonzero constants, does not carry much information as well. Hence, a wide scattered band of the features is desirable in order to have enough representative power [5]. In general, it can be detected by investigating whether the feature has a large variance along the data manifold, and the density weighted variance is estimated as in [5]:

$$\begin{aligned} Var_{\mathcal{M}}(\mathbf{f}_l) &\propto \sum_{i=1}^{N}\left[f_l^{(i)} - \mu_{\mathcal{M}}(\mathbf{f}_l)\right]^2 D_{ii} \\ &= [\mathbf{f}_l - \mu_{\mathcal{M}}(\mathbf{f}_l)\mathbf{e}]^T\mathbf{D}[\mathbf{f}_l - \mu_{\mathcal{M}}(\mathbf{f}_l)\mathbf{e}] \\ &= \hat{\mathbf{f}}_l^T\mathbf{D}\hat{\mathbf{f}}_l, \end{aligned} \tag{14}$$

where $\mathbf{e} = [1, \ldots, 1]^T \in \mathbb{R}^N, \hat{\mathbf{f}}_l = \mathbf{f}_l - \mu_{\mathcal{M}}(\mathbf{f}_l)\mathbf{e}, \mu_{\mathcal{M}}(\mathbf{f}_l)$ is the weighted mean of the lth feature calculated by

$$\begin{aligned} \mu_{\mathcal{M}}(\mathbf{f}_l) &= \sum_{i=1}^{N}\frac{D_{ii}}{\sum_{j=1}^{N} D_{jj}} f_l^{(i)} \\ &= \frac{\sum_{i=1}^{N} D_{ii} f_l^{(i)}}{\sum_{j=1}^{N} D_{jj}} \\ &= \frac{\mathbf{f}_l^T\mathbf{D}\mathbf{e}}{\mathbf{e}^T\mathbf{D}\mathbf{e}}. \end{aligned} \tag{15}$$

Eventually, we formulate the local kernel regression score as an integration of the two requirements (i.e., (13) and (14)) for a feature:

$$\begin{aligned} \Phi_{LKR}(\mathbf{f}_l) &= \frac{E_{local}(\mathbf{f}_l)}{Var_{\mathcal{M}}(\mathbf{f}_l)} \\ &= \frac{\mathbf{f}_l^T\mathbf{D}^{\frac{1}{2}}(\mathbf{I}-\mathbf{B})^T(\mathbf{I}-\mathbf{B})\mathbf{D}^{\frac{1}{2}}\mathbf{f}_l}{\hat{\mathbf{f}}_l^T\mathbf{D}\hat{\mathbf{f}}_l}. \end{aligned} \tag{16}$$

We will seek the feature that makes the within-neighborhood estimation error as small as possible, meanwhile its variance cross all data points is as large as possible. In practice, we rank the features[1] in the ascending order of $\Phi_{LKR}(.)$ and choose the foremost features in the rank list as the relevant ones. The main steps of the proposed local kernel regression scoring algorithm are summarized in Algorithm 1.

Algorithm 1. The LKR score for feature selection.

```
input : $\mathcal{X}$, the number of nearest neighbors $k$
    (for unsupervised feature selection) OR the label $\mathbf{y}$
    (for supervised feature selection)
output: the ranked feature list
1 Construct the kernel matrix $\mathbf{K}$ over $\mathcal{X}$;
2 Construct the neighborhood graph $\mathbf{G}$ and the weight
    matrix $\mathbf{W}$ (using (5) for unsupervised feature
    selection OR (6) for supervised feature selection);
3 Learn the local kernel regression model with
    $\mathbf{K}$, $\mathbf{G}$ by (10) and (11);
4 Compute LKR score for each feature by (16), then
    rank features in the ascending order of $\Phi_{LKR}(\mathbf{f}_l)$.
```

The complexity of Step 1 and Step 2 is both $O(N^2d)$. For Step 3, it is $O(Nk^3)$ and the complexity for Step 4 is $O(Nkd)$, where we denote k as the number of nearest neighbors (as for *unsupervised* LKR score) or the size of the largest class (as for *supervised* LKR score). For the *unsupervised* case, we often specify a considerable small size of neighborhood ($k \ll N, d$). Hence, the overall complexity is $O(N^2d)$, which is the same as the Laplacian score. Consequently, the overall complexity of *supervised* LKR score is $O(\max(Nk^3, N^2d))$.

4 Relationship between the LKR Score and Laplacian Score

The recently proposed Laplacian score [5] is an effective feature selection method, which is to seek the features that are able to preserve the locality. In essence, it prefers the features that minimize the following formula [5]:

$$\sum_{ij}\left(f_l^{(i)} - f_l^{(j)}\right)^2 w_{ij}, \tag{17}$$

where w_{ij} is the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (the Gaussian kernel function is adopted for w_{ij} in [5]). It can be seen that a larger value of $(f_l^{(i)} - f_l^{(j)})^2$ will lead to a larger value of the objective function in (17) if w_{ij} is large, thus indicating $\mathbf{f}_l$ is an undesirable feature. By setting the derivative of (17) with respect to $f_l^{(i)}$ to 0, it can be found that minimizing $\sum_{ij}(f_l^{(i)} - f_l^{(j)})^2 w_{ij}$ requires the elements of $\mathbf{f}_l$, satisfying the following harmonic property:

$$f_l^{(i)} = \frac{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij} f_l^{(j)}}{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij}}. \tag{18}$$

Thereby, the Laplacian score ranks the feature value at each point based on the estimation error using the nearest-neighbor regression (equivalent to the classical Nadaraya-Waston algorithm [6]), i.e., a weighted average of the values of its neighbors, and the weight of each neighbor is proportional to the proximity. That is, both the LKR score and the Laplacian score can be interpreted from the perspective of local regression and they both select the features capable of keeping the local information. Nevertheless, one key difference between them is that the LKR score *explicitly* estimates the feature value at each point by its neighbors using the kernel ridge regression approach, while the Laplacian score *implicitly* performs the estimation by the nearest-neighbor regression method. Furthermore, by investigating on the regression coefficients in (11) and (18), it can be found that the nearest-neighbor regression considers only the distance between $\mathbf{x}_i$ and its neighbors $\mathbf{x}_j \in \mathcal{N}_i$, and ignores the distance between the neighbors. Subsequently, $\mathbf{x}_i$ may be close to points that are far from each other, resulting in a weighted average of feature values from two distant but likely unrelated points. In contrast, kernel ridge regression considers the distance between the pairs of points in the neighbors to decide how heavily to weigh the influence of relevant neighboring points, which is embodied in the computation for the regression coefficients in (8). Hence, the kernel ridge regression is expected to be better for revealing the local relationship of data in comparison with the nearest-neighbor regression.

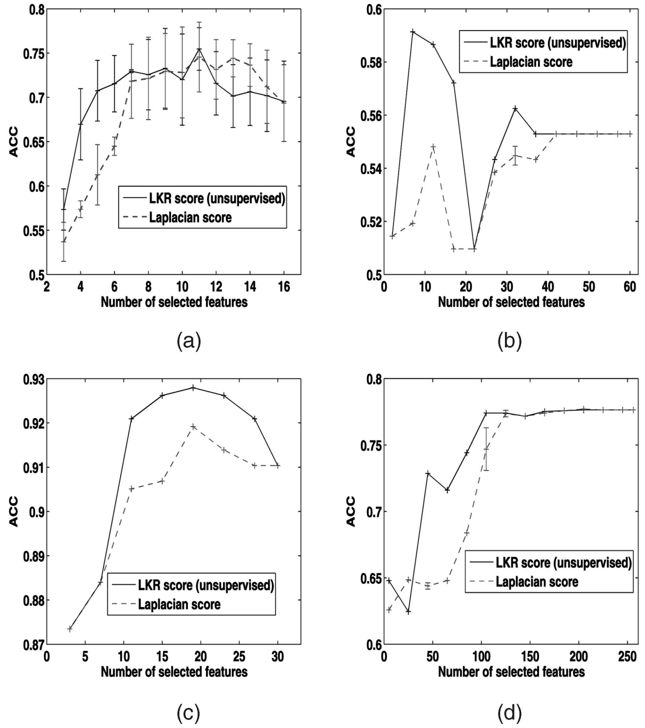
5 Experimental Results

Six benchmark data sets were used to investigate the performance of the LKR score to select informative features either for clustering or classification. The characteristics of these six data sets are summarized in Table 1. We compared the *unsupervised* LKR score with the Laplacian score to select features for clustering on the first four data sets in Table 1. For identifying features for classification on the last two data sets in Table 1, the *supervised* LKR score was compared with the popular supervised *filter* methods: Fisher score and ReliefF [8]. In all the experiments we have tried in Sections 5.1 and 5.2, the popular Gaussian kernel, $\mathcal{K}(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2/h)$, was utilized for both the

[1] If $\mathbf{f}_l$ is a constant vector, e.g., $\mathbf{0}$ or $\mathbf{e}$, we have $Var_{\mathcal{M}}(\mathbf{f}_l) = 0$. This trivial candidate can be easily excluded from the selection.

TABLE 1
Characteristics of the Data Sets Used in Experiments

Data Set	Dimension (d)	No. of Data Points (N)	No. of Clusters/Classes (C)
pendigit [7]	16	3498	10
wdbc [7]	30	569	2
sonar [7]	60	208	2
USPS49 [4]	256	1673	2
PIEC27 [10]	1024	1428	68
UMist [3]	10304	575	20



(a) (b)

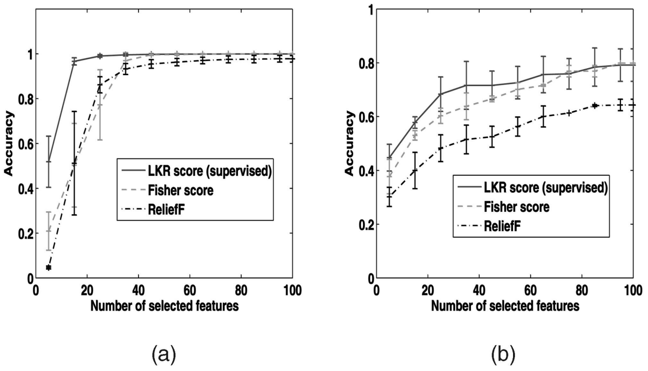
(c) (d)

Fig. 1. Clustering accuracy versus different number of top-ranked features obtained via the *unsupervised* LKR score and the Laplacian score, respectively. (a) pendigit, (b) sonar, (c) wdbc, and (d) USPS49.

LKR score and the Laplacian score with the same kernel parameter h. Also, the ridge parameter λ was set at 0.1. We shall empirically study the sensitivity of choosing λ and h, as well as the neighborhood size, in Section 5.3.

5.1 Feature Selection for Clustering

The pendigit, wdbc, and sonar data sets are from the UCI repository [7]. The USPS49 data set[2] contains the gray-scale images of digits "4" and "9" from the USPS ZIP code handwritten digits database [4], in which each image is represented by a 256-dimensional vector. No preprocessing was performed except on the wdbc data set, for which each feature was normalized to zero mean and unit variance so as to make the scales of different features roughly equal a priori. The performance of k-means clustering, with top-ranked features for each data set obtained by the unsupervised LKR score and the Laplacian score respectively, was utilized to assess the efficacy of the two approaches. For both methods, a neighborhood graph was built with the neighborhood size of 10 and the parameter of the Gaussian kernel $h = 100$. For the k-means clustering, the number of clusters was set at the number of classes C in each data set. We started from 10 different random initializations and chose the solution with the lowest objective function value of the k-means. The clustering accuracy index (ACC) [5] was then computed with this solution. The results over different number of selected features are summarized in Fig. 1, where the mean and the standard deviation of ACC are obtained over 20 repeats of the above process.



(a) (b)

Fig. 2. Classification accuracy versus different number of top-ranked features obtained via the *supervised* LKR score, Fisher score, and ReliefF, respectively. (a) PIEC27 and (b) UMist.

From Fig. 1, it can be seen that the clustering performance varies with the number of features. The best performance is often obtained with less features than with all the features. This indicates that feature selection is capable of improving the clustering performance. Furthermore, the LKR score outperforms the Laplacian score when a small number of features are selected. This implies that the relevant features are authentically ranked foremost in the LKR score ranking list. Therefore, it uses less features than the Laplacian score while achieving the same accuracy (see Fig. 1). As the number of selected features approaches to the original full dimensionality of each data set, their performance naturally becomes comparable.

5.2 Feature Selection for Classification

The PIEC27 data set [10] contains 68 human subjects of the frontal poses (C27) but under different illumination conditions, with each subject having 21 faces. We used the cropped images[3] of 32×32 pixels. The pixel values were scaled to $[0, 1]$, and each image was represented by a 1,024-dimensional vector. The UMist face data set[4] [3] was also scaled and represented in vector space. For the two data sets, two-thirds of the whole samples in each class were randomly selected to form the training set, and the remaining ones were the test set. We utilized the *supervised* LKR score, fisher score (equivalent to the supervised extension of the Laplacian score as shown in [5]), and the ReliefF [8] to select the features on the training set, and the 1-nearest neighbor (1NN) classification accuracy on the test set using the selected features was used to evaluate the quality of selected features by the three methods. The Gaussian kernel parameter h was still set at 100 for the *supervised* LKR score. The random split was repeated 20 times. Fig. 2 shows the mean and the standard deviation of the classification accuracy versus the different numbers of selected features. It can be seen that the LKR score can significantly improve the performance of Fisher score and ReliefF on the PIEC27 data set when less than 40 features were selected. The LKR score outperforms the other two in terms of the average classification accuracy on the UMist data set. A plausible reason is that, while the Fisher score and the ReliefF only utilize the linear information of data for ranking, the *supervised* LKR score is able to extract the nonlinear relationship within data set due to the use of nonlinear Gaussian kernel.

[2] http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/data.html.

[3] http://www.cs.uiuc.edu/homes/dengcai2/Data/data.html.

[4] http://images.ee.umist.ac.uk/danny/database.html.

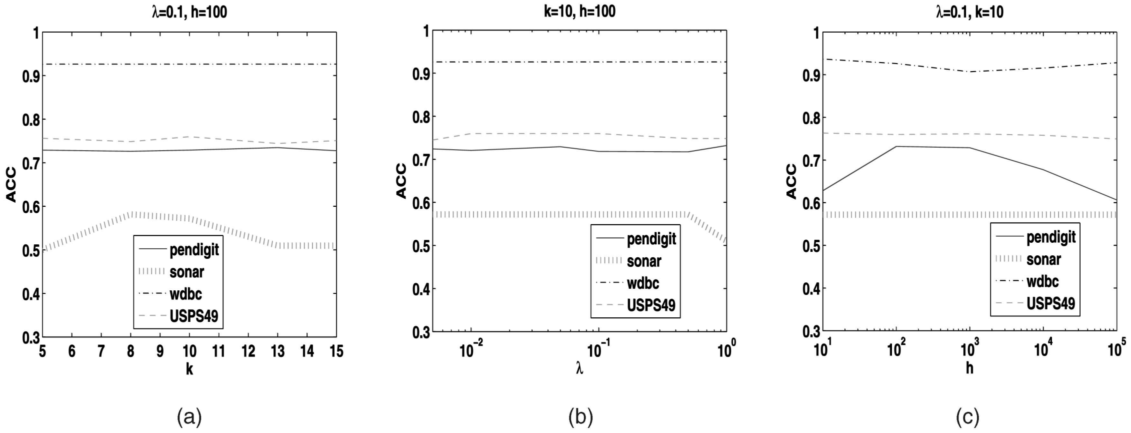


Fig. 3. The impacts of parameters on the performance of the *unsupervised* LKR score. (a) Size of neighborhood, (b) regularization parameter, and (c) Gaussian kernel parameter.

5.3 Studies of the Sensitivity of Parameters

For the *unsupervised* LKR score, there are three parameters: the size of neighborhood k, the ridge regression constant λ, and the Gaussian kernel parameter h. For the *supervised* LKR score, there are two parameters: λ and h. In order to study their impacts on the performance of proposed method, we fixed the number of selected features (pendigit: 6 features; sonar: 17 features; wdbc: 15 features; USPS49: 100 features; PIEC27: 15 features; UMist: 25 features) and investigated the performance of LKR score as a function of each parameter. When assessing any one of the parameters, the remaining parameters were kept unchanged. Figs. 3 and 4 show the average performance indices over 20 trials for each value of these parameters in the unsupervised and supervised learning environments, respectively.

As for the size of neighborhood, it can be seen from Fig. 3a that the performance of LKR score is almost stable within the range $\{5, 6, \ldots, 15\}$ for all the data sets. Further, from Figs. 3b, 3c, 4a, and 4b, we have found that the LKR score is generally not very sensitive to the regularization parameter λ and the Gaussian kernel parameter h on most data sets, when they are neither too small ($\lambda < 10^{-2}, h < 10^{1}$) nor too large ($\lambda > 10^{0}, h > 10^{4}$). By a rule of thumb, λ around 10^{-1} and h around 10^{2} are often an appropriate choice to produce the satisfactory results.

6 Conclusion

In this paper, we have proposed the LKR score to select the features based on the local kernel regression and the neighborhood graph. This score ranks the features based on their capabilities of keeping the local similarities in a small patch of the high-dimensional data. Essentially, this score is applicable to both of supervised learning and unsupervised learning by adopting different definitions of the neighborhood. Experimental results have shown the efficacy of the proposed approach in comparison with the existing methods.

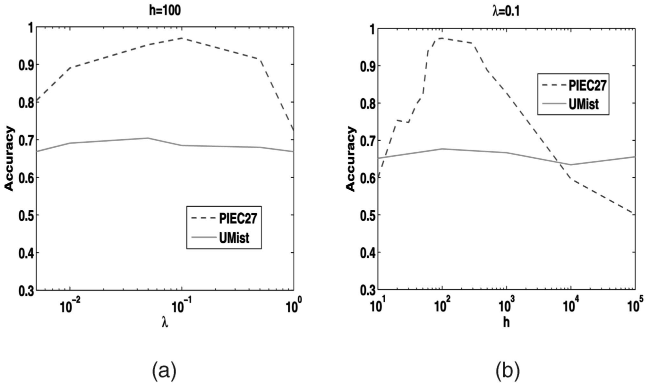


Fig. 4. The impacts of parameters on the performance of the *supervised* LKR score. (a) Regularization parameter and (b) Gaussian kernel parameter.

Acknowledgments

The work in this paper was supported by a grant from the Research Grant Council of the Hong Kong SAR (Project No: HKBU 210306) and a Faculty Research Grant of Hong Kong Baptist University (Project Code: FRG/07-08/II-54).

References

[1] A. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning," *Artificial Intelligence,* vol. 97, nos. 1/2, pp. 245-271, 1997.

[2] F. Chung, *Spectral Graph Theory.* Am. Math. Soc., 1997.

[3] D. Graham and N. Allinson, "Characterizing Virtual Eigensignatures for General Purpose Face Recognition," *Face Recognition: From Theory to Applications,* Springer, 1998.

[4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer, 2003.

[5] X. He, D. Cai, and P. Niyogi, "Laplacian Score for Feature Selection," *Advances in Neural Information Processing Systems,* vol. 18, pp. 507-514, 2005.

[6] E. Nadaraya, *Nonparametric Estimation of Probability Densities and Regression Curves.* Kluwer Academic, 1989.

[7] D. Newman, S. Hettich, C. Blake, and C. Merz, *UCI Repository of Machine Learning Databases.* Univ. of California, 1998.

[8] M. Robnik-Šikonja and I. Kononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," *Machine Learning,* vol. 53, no. 1, pp. 23-69, 2003.

[9] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis.* Cambridge Univ. Press, 2004.

[10] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 12, pp. 1615-1618, Dec. 2003.

[11] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems.* John Wiley, 1977.

[12] L. Wolf and A. Shashua, "Feature Selection for Unsupervised and Supervised Inference: The Emergence of Sparsity in a Weight-Based Approach," *J. Machine Learning Research,* vol. 6, pp. 1855-1887, 2005.

[13] M. Wu and B. Schölkopf, "A Local Learning Approach for Clustering," *Advances in Neural Information Processing Systems,* vol. 19, pp. 1529-1536, 2007.

[14] M. Wu and B. Schölkopf, "Transductive Classification via Local Learning Regularization," *Proc. 11th Int'l Conf. Artificial Intelligence and Statistics,* pp. 628-635, 2007.

[15] M. Wu, K. Yu, S. Yu, and B. Schölkopf, "Local Learning Projections," *Proc. 24th Int'l Conf. Machine Learning,* pp. 1039-1046, 2007.

[16] Z. Zhao and H. Liu, "Spectral Feature Selection for Supervised and Unsupervised Learning," *Proc. Int'l Conf. Machine Learning,* pp. 1151-1158, 2007.