Exclusive lasso-based k-nearest-neighbor classification

Qiu, Lin; Qu, Yanpeng; Shang, Changjing; Yang, Longzhi; Chao, Fei; Shen, Qiang

Published in:
Neural Computing and Applications

DOI:
10.1007/s00521-021-06069-5

Publication date:
2021

Citation for published version (APA):
Qiu, L., Qu, Y., Shang, C., Yang, L., Chao, F., & Shen, Q. (2021). Exclusive lasso-based k-nearest-neighbor classification. *Neural Computing and Applications*, *33*(21), 14247-14261. https://doi.org/10.1007/s00521-021-06069-5

Noname manuscript No.
(will be inserted by the editor)

Exclusive Lasso-Based k-Nearest Neighbors Classification

Lin Qiu · Yanpeng Qu · Changjing Shang · Longzhi Yang · Fei Chao · Qiang Shen

Received: date / Accepted: date

Abstract Conventionally, the k nearest-neighbor (kNN) classification is implemented with the use of the Euclidean distance-based measures, which are mainly the one-to-one similarity relationships such as to lose the connections between different samples. As a strategy to alleviate this issue, the coefficients coded by sparse representation (SR) have played a role of similarity gauger for nearest neighbor classification as well. Although SR coefficients enjoy remarkable discrimination nature as a one-to-many relationship, it carries out variable selection at the individual level so that possible inherent group structure is ignored. In order to make the most of information implied in the group structure, this paper employs the Exclusive Lasso (EL) strategy to perform the similarity evaluation in two novel nearest neighbor classification methods. Experimental results on both benchmark data sets and the face recognition problem demonstrate that the EL-based kNN method outperforms certain state-of-the-art classification techniques and existing representation-based nearest neighbor approaches, in terms of both the size of feature reduction and the classification accuracy.

Keywords Exclusive Lasso · Sparse coefficient · kNN · Classification

Lin Qiu · Yanpeng Qu (Corresponding Author)
Information Science and Technology College, Dalian Maritime University, Dalian, 116026, China
E-mail: yanpengqu@dlmu.edu.cn

Changjing Shang · Qiang Shen
Department of Computer Science, Faculty of Business and Physical Sciences, Aberystwyth University, SY23 3DB, UK

Longzhi Yang
Department of Computer and Information Sciences, Northumbria University, London E1 7HT, UK

Fei Chao
Department of Computer Science, Xiamen University, Fujian 361005, China

1 Introduction

The k-nearest neighbor (kNN) algorithm [1] has enjoyed much attention since its inception as an intuitive and effective classification method. Conventionally, kNN is implemented with the use of the Euclidean distance-based measures. However, such measures only embrace the distance information between two samples, thus the sight of certain meaningful information, such as distribution information of training set and class structure information is ignored.

In order to comfort this defection, many efforts, such as mahalanobis distance [2], generalized mean distance [3], the nearest feature line (NFL) [4] and the center-based nearest neighbor (CNN) [5], have been devoted to improve the performance of the Euclidean distance. Specifically, the mahalanobis distance represents the covariance distance of the data and takes into account the relationships between the various samples. The generalized mean distance is defined as the generalized mean of the k distances between the query sample and each k nearest neighbors. The NFL classifier defined a new distance measure between a query sample and a straight "line" which passed through two samples of the same class. The classification by NFL is carried out with the use of the minimum distance between the feature point of the test and the feature line's. The CNN classifier is proposed to solve the classification task as an improvement of NFL. CNN defines a new distance between the test sample and a "line" called the center-based line

which passed through a sample point with known label and the center of the sample class.

Distinct from the one-to-one measures, the sparse representation (SR) [6] [7] approach reconstructs a query sample by selecting a small subset from a large data set, meanwhile making the reconstructing error as small as possible. In general, the sparse reconstructive coefficients can reflect inherent geometric similarity information and fully consider the neighborhood relationship of samples. In [8], it has been proved that the representation coefficients can effectively indicate the true nearest neighbors of a given query sample. Therefore, the sparse coefficients can serve as an effective similarity measure for searching the nearest neighbors. In most cases, the SR-based classification is implemented in the form of Least Absolute Shrinkage and Selection Operator (Lasso) [9] which exploits the classical ℓ_1-norm regularization [10–16]. Particularly, in [10], the SR-based classification (SRC) was proposed to generalize the classical nearest neighbor method. In [11], the sum of coefficients (SoC) plays a role of the indicator to obtain more discriminative information from sparse coefficients. In [12], a two-stage strategy was proposed to select the nearest neighbors by using sparse coefficients, first coarsely and then finely. Two novel kNN-based methods were proposed in [13], in which the classification decisions were made in light of two weighted voting strategies. In [14], an element-wise sparsity coefficient matrix was designed to identify exclusive k values for different test samples and the majority voting is used as the decision rule. Moreover, SR has been successfully applied in pattern recognition such as in face recognition [15, 16]. In [15], a general classification method based on SR is proposed for object recognition which can help handle errors due to occlusion and corruption uniformly. In [16], a new model which is much more robust to outliers is proposed, where the maximum likelihood estimation solution of the sparse coding problem is obtained.

To develop Lasso-based SR classification systems, Group Lasso (GL) [17] [18], which uses the ℓ_2-norm within a group and the ℓ_1-norm between groups, has been applied to obtain sparse coefficients [19–23]. In [19], a locality-constrained GL coding method is used for microvessel image classification and realizes the automatic "hot spot" detection of angiogenesis for human liver carcinoma. In [20], a novel collaborative double sparse period-GL algorithm, which is based on two main priors of the fault bearing signal provided by the resonance frequency and the fault characteristic frequency respectively, is proposed. The method proposed in [21] utilized an ℓ_1-norm regularization and an $\ell_{2,1}$-norm regularization to generate the element-wise sparsity for determining the value of k of each test sample and the row sparsity for determining the noisy training samples, respectively. In [22], GL is used to construct the objective function, which can make collaborative representation well-structured, to solve the lack of samples' problem in face recognition. In [23], the KSVD optimization method is used to obtain the sparse GL solution which can guarantee that k most relevant class groups are selected. In this way, those unrelated groups can be filtered out by using GL in group level, instead of individual sample level.

In the aforementioned SR-based classification methods, Lasso treats different coefficients equally and carries out variable selection at the individual level. However, this may result in excessive compression of parameters due to certain large absolute values and possible neglect of the inherent group structure of data samples. Whilst such defects may be reduced by GL through the exploitation of intra-group non-sparsity (via ℓ_2-norm) and inter-group sparsity (via ℓ_1-norm), each group of variables are either selected or discarded, entirely. In so doing, the results selected by GL may suffer from redundant information or deficient information. To address this deficiency, in this paper, the sparse coefficients gained by the use of Exclusive Lasso (EL) [24] are employed as the similarity measure to implement an improved kNN classification system. The EL regularization is designed to retain the group structure of the variables using the ℓ_1-norm and ℓ_2-norm to obtain intra-group sparsity and inter-group non-sparsity, respectively. From this, EL guarantees the selection of at least one variable from each group. In particular, with the use of distinct decision indicators, two EL-based kNN algorithms are proposed herein. The experimental results on benchmark and face recognition data sets are conducted to evaluate the classification performance of the proposed methods. The results of experiments manifest the proposed methods have promising performance.

The contributions of this paper are outlined below:

- Two SR-based kNN classification methods are presented, where the representation coefficients of EL are employed to act as the similarity gauger to support the nearest neighbor computation.
- The SR coefficients of EL are employed to help capture and reflect inherent geometric similarity information, enabling a full consideration of the neighborhood relationship of data samples, thereby forming a sharp contrast with conventional Euclidean distance-based kNN classification.
- SR is implemented from group level to take into consideration of possible inherent group structure between data samples, encouraging similar elements in different groups to co-exist, thereby differing from

approaches represented by existing Lasso-based kNN classification,
- Both proposed methods guarantee the selection of at least one variable from each group, entailing more meaningful information while avoiding redundant or deficient information, thereby improving performance over GL-based kNN classification.

The remainder of this paper is structured as follows. In Section 2, the preliminary of kNN-based methods, sparse representation-based classification and Exclusive Lasso are reviewed. Section 3 introduces the algorithms of Exclusive Lasso-based kNN classification. In Section 4, the comparative experimental results are presented and discussed. The paper is concluded in Section 5, with a brief discussion regarding important further work.

2 Theoretical Background

Notationally, in the following, $\mathbf{T} = \{(x_i, l_i)\}, i = 1, \ldots, n$ denotes a dataset which contains n distinct objects and is divided into M categories $\mathbf{C} = \{c_1, c_2, \ldots, c_M\}$, where $l_i \in \mathbf{C}$.

2.1 Distance-weighted k nearest neighbors

The kNN classification method [1] works by assigning a query sample to a decision class that is most common amongst its k nearest neighbors. By virtue of majority voting for decision-making, in kNN, the k neighbors of a query sample have an identical weight. In so doing, the classification performance is sensitive to the quality of the instances.

As a solution to this important issue, the distance-weighted k-nearest neighbor (DWkNN) rule has been proposed in the literature [25]. Given a query data sample y, let $\mathbf{T}^{NN}_{sort\_k} = \{(x^{NN}_{sort\_i}, l^{NN}_{sort\_i}) | i = 1, \ldots, k\}$, denote the k nearest neighbors arranged in an increasing order according to the distances $d(y, x^{NN}_{sort\_i})$. The corresponding weight of $x^{NN}_{sort\_i}$ is defined by

$$\widehat{w}_i = \frac{d(y, x^{NN}_{sort\_k}) - d(y, x^{NN}_{sort\_i})}{d(y, x^{NN}_{sort\_k}) - d(y, x^{NN}_{sort\_1})} \times \frac{d(y, x^{NN}_{sort\_k}) + d(y, x^{NN}_{sort\_1})}{d(y, x^{NN}_{sort\_k}) + d(y, x^{NN}_{sort\_i})}. \tag{1}$$

When the weights are set to be 1, DWkNN is reduced to the classical kNN.

Using DWkNN, the label of y is determined in light of the weighted majority vote of the k nearest neighbors such that

$$l_y = \arg\max_{c_j \in \mathbf{C}} \sum_{i=1}^{k} \widehat{w}_i^{(c_j)}, \tag{2}$$

where l_y is the label of the query sample y. Here, $\widehat{w}_i^{(c_j)}$ is induced as follows:

$$\widehat{w}_i^{(c_j)} = \begin{cases} \widehat{w}_i, & \text{if } \; l^{NN}_{sort\_i} = c_j, \\ 0, & \text{otherwise.} \end{cases} \quad i = 1, \ldots, k \text{ and } j = 1, \ldots, M. \tag{3}$$

Since the Euclidean distance used in such conventional kNN classification systems only embraces the distance information between two data samples, meaningful information embedded within the training dataset (such as the distribution of the data samples and the information on class structure) is ignored. In order to reduce this deficiency, the similarity measures produced by SR are often introduced to instance-based learning classification.

2.2 Sparse representation-based classification

Sparse representation-based classification (SRC) [10] approximately represents a query sample y by the following representation coefficients of a linear system:

$$\widehat{\alpha} = \arg\min_{\alpha} \{\|y - \mathbf{X}\alpha\|_2^2 + \lambda \|\alpha\|_1\}, \tag{4}$$

where $\mathbf{X} = [x_1, x_2, \ldots, x_n]$ is the matrix of all training samples; $\widehat{\alpha} = [\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_n]^T$ is the optimal solution to represent y. Moreover, the optimisation problem shown in Eq. (4) is also known as Lasso [9], which can effectively prevent overfitting and yield sparse solution to select the important variables and reduce the complexity of the model. In the light of the resulting $\widehat{\alpha}$, y is classified into the category which has the minimum class-given residual reconstructed by SRC:

$$l_y = \arg\min_{c_j \in \mathbf{C}} \|y - \mathbf{X}\widehat{\alpha}^{(c_j)}\|_2, \tag{5}$$

where $\widehat{\alpha}^{(c_j)}$ is the reconstruction vector induced by $\widehat{\alpha}$ as follow.

$$\widehat{\alpha}_i^{(c_j)} = \begin{cases} \widehat{\alpha}_i, & \text{if } \; l_i = c_j, \\ 0, & \text{otherwise.} \end{cases} \quad i = 1, \ldots, n \text{ and } j = 1, \ldots, M \tag{6}$$

An alternative decision strategy for SRC is the sum of coefficients (SoC) [11], which is defined as follows:

$$l_y = \arg\max_{c_j \in \mathbf{C}} \sum \widehat{\alpha}^{(c_j)}. \tag{7}$$

By this manner, y is classified into the category which has the maximal sum of coefficients. In general, when

the residual strategy suffers from small gaps between different classes, the resulting erroneous prediction may be accordingly corrected by SoC which is more discriminative and enjoys larger margins between classes [11].

2.3 Exclusive Lasso

Since SRC is proposed based on Lasso, it weights different coefficients with the identical degree. This may result in excessive compression for the samples that have large absolute value. In order to identify the diversity of the samples from group level, the Group Lasso (GL) algorithm [17] was proposed to maintain the inherent group structure of variables as follow [26].

$$\widehat{\alpha}_G = \arg\min_{\alpha}\{\|y - \mathbf{X}\alpha\|_2^2 + \lambda \sum_{g=1}^{G} \|\alpha_g\|_2^1\}, \tag{8}$$

where $\mathbf{X} = [x_1, x_2, \ldots, x_n]$ is the matrix of all training samples; the coefficients in α are divided into G groups and α_g represents the coefficient vector of the g-th group. The composite regularization in Eq. (8) is termed as $\ell_{2,1}$-norm [27] which achieves the intra-group non-sparsity via ℓ_2-norm and inter-group sparsity via ℓ_1-norm.

In general, GL forces the sparsity of variables at the inter-group level, so that the variables belong to different groups are competing to survive. Since each group of variables will be either selected or discarded entirely, the resulting selected groups by GL may suffer from redundant information or lack of partial useful information. To overcome this drawback of GL, the Exclusive Lasso (EL) algorithm was presented [24].

The optimal solution of EL is defined as follows.

$$\widehat{\alpha}_E = \arg\min_{\alpha}\{\|y - \mathbf{X}\alpha\|_2^2 + \lambda \sum_{g=1}^{G} \|\alpha_g\|_1^2\}, \tag{9}$$

where the regularization term is an $\ell_{1,2}$-norm that implements the intra-group sparsity via ℓ_1-norm and the inter-group non-sparsity via ℓ_2-norm. For instance, in a three-dimensional space, let the first two variables, $\alpha_{1,1}$ and $\alpha_{1,2}$, are in one group. The third variable, $\alpha_{2,1}$, is in another group. In Fig. 1(a), the relationship between $\ell_{1,2}$-norm and ℓ_1-norm is displayed. Relatively, Fig. 1(b) illustrates the relationship between $\ell_{1,2}$-norm and ℓ_2-norm. Fig. 1(c) concludes that $\ell_{1,2}$-norm inherits properties from both ℓ_1-norm and ℓ_2-norm.

Compared to GL, EL performs the variable selection by ensuring that at least one element will be selected from each group. In so doing, EL can support

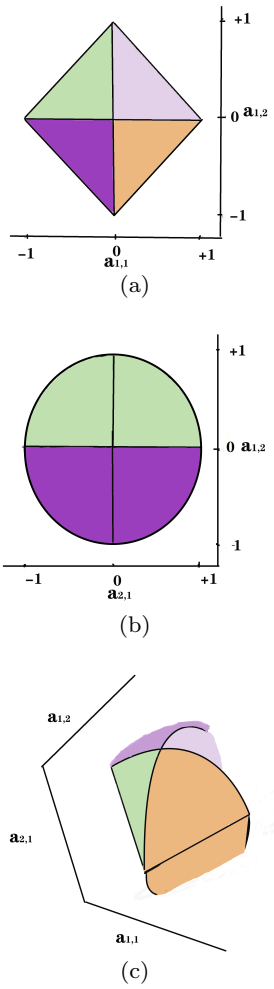


Fig. 1 Unit ball for exclusive lasso in a 3D space.

the coexistence of similar features in different groups. In [28], it is indicated that the EL penalty is virtually the tightest convex relaxation for the group regularization constraints which require the solution vector to contain at least one variable from each group.

As with SR regularization-based approaches, EL has also been used to perform the task of feature selection [29, 30], where it acts to capture the underlying descriptors for typically, a certain class predictor. As a result, features that are highly correlated to the decision classes are selected. For instance, in [8], it has proved that the SR coefficients can effectively indicate the true nearest neighbors of a given query sample. Reflecting this viewpoint, in this paper, EL is employed to play the role of a similarity gauger in kNN classification. Particularly, the resulting SR coefficients are used to implement two EL-based kNN algorithms as distinct decision indicators.

3 Exclusive Lasso-Based k-Nearest Neighbor Classification

Within the neighborhood located by EL, two kNN classification methods are implemented as EkNN-C and EkNN-R, which are with the coefficient and residual decision indicators, respectively.

By solving the optimization problem in Eq. (9), the samples $\mathbf{T}_k^{NN} = \{(x_i^{NN}, l_i^{NN})\}$, $i = 1, 2, \ldots, k$, which enjoy the k largest elements $\tilde{\alpha} = \{\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_k\}$ in the optimal solution $\widehat{\alpha}_E$, are assigned to be the k nearest neighbors of y. Within $\mathbf{T}_k^{NN}$, the decision indicators of EkNN-C is

$$l_y = \arg\max_{c_j \in \mathbf{C}} \sum_{i=1}^{k} \tilde{\alpha}_i^{(c_j)}. \tag{10}$$

Here, $\tilde{\alpha}^{(c_j)} \in \mathbb{R}^k$ is the reconstruction vector induced by $\tilde{\alpha}$ as follows.

$$\tilde{\alpha}_i^{(c_j)} = \begin{cases} \tilde{\alpha}_i, & \text{if } l_i^{NN} = c_j, \\ 0, & \text{otherwise.} \end{cases} \quad i = 1, \ldots, k \text{ and } j = 1, \ldots, M. \tag{11}$$

Following the above discussion, the EkNN-C algorithm can be summarized in Algorithm 1.

Algorithm 1: The EkNN-C algorithm

Input:
$\mathbf{C} = \{c_1, c_2, \ldots, c_M\}$: the set of class labels;
$\mathbf{T} = \{(x_i, l_i)\}$ $i = 1, \ldots, n$: the training set, $l_i \in \mathbf{C}$;
y : the test sample;
k : the number of nearest neighbors;
λ : a parameter of regularization.
Output: class label l_y

```
Get sparse coefficients: α̂_E ← Solve Eq.(9).
α̃ ← k largest elements of α̂_E;
for j=1,...,M do
    α̃^(c_j) = α̃;
    for i=1,...,k do
        if l_i^NN ≠ c_j then
            α̃_i^(c_j) = 0;
        end
    end
end
l_y ← max_{c_j ∈ C} Σ_{i=1}^{k} α̃_i^(c_j).
```

Analogously, within $\mathbf{T}_k^{NN}$, the residual distance between y and x_i^{NN} is defined as:

$$d_r(y, x_i^{NN}) = \|y - \tilde{\alpha}_i x_i^{NN}\|_2. \tag{12}$$

Considering the residual distance with respect to x_i^{NN} as its contribution to the sparse reconstruction representation of y, the residual distance-weighted function is defined as follow.

$$w_i = \begin{cases} \dfrac{d_{\max}^{NN} - d_r(y, x_i^{NN})}{d_{\max}^{NN} - d_{\min}^{NN}}, & \text{if } d_{\max}^{NN} \neq d_{\min}^{NN}, \\ 1, & \text{if } d_{\max}^{NN} = d_{\min}^{NN}. \end{cases} \tag{13}$$

Here, $d_{\max}^{NN}$ and $d_{\min}^{NN}$ are the maximum and minimum of all residual distances, respectively. With the use of both the sparse coefficients and the residual distance weights of the k nearest neighbors, EkNN-R predicts the class label l_y of y in light of the following equation.

$$l_y = \arg\max_{c_j \in \mathbf{C}} \sum_{i=1}^{k} w_i \times \tilde{\alpha}_i^{(c_j)}. \tag{14}$$

The EkNN-R algorithm is outlined in Algorithm 2.

Algorithm 2: The EkNN-R algorithm

Input:
$\mathbf{C} = \{c_1, c_2, \ldots, c_M\}$: the set of class labels;
$\mathbf{T} = \{(x_i, l_i)\}$ $i = 1, \ldots, n$: the training set, $l_i \in \mathbf{C}$;
y : the test sample;
k : the number of nearest neighbors;
λ : a parameter of regularization.
Output: class label l_y

```
Get sparse coefficients: α̂_E ← Solve Eq.(9).
α̃ ← k largest elements of α̂_E;
for j=1,...,M do
    α̃^(c_j) = α̃;
    for i=1,...,k do
        if l_i^NN ≠ c_j then
            α̃_i^(c_j) = 0;
        end
    end
end
for i=1,...,k do
    d_r(y, x_i^NN) ← ‖y − α̃_i x_i^NN‖_2
end
for i=1,...,k do
    if d_max^NN ≠ d_min^NN then
        w_i ← (d_max^NN − d_r(y, x_i^NN)) / (d_max^NN − d_min^NN);
    else
        w_i ←1;
    end
end
l_y ← max_{c_j ∈ C} Σ_{i=1}^{k} w_i × α̃_i^(c_j);
```

As reported in [8], the representation coefficients can effectively indicate the true nearest neighbors of a given query sample. Let y be a query sample which is represented as $y = \sum_{i=1}^{n} \bar{\alpha}_i x_i$, where $\bar{\alpha}=[\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_n]$

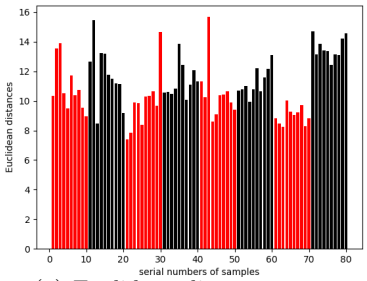
denotes the sparse solution returned by the SR-based classification system. The x_i is not the neighbor of y when $\bar{\alpha}_i = 0$, since $\bar{\alpha}_i = 0$ signifies x_i does not lie in the best representation linear subspace of y. Thus, y can be further represented as $y = \sum_{i=1}^{n'} \bar{\alpha}_i x_i$, where $\bar{\alpha}_i \neq 0$ and $n' \leq n$, with $\bar{\alpha}_i$ being computed by

$$\bar{\alpha}_i = \frac{1}{2}(d^2(\sum_{t=1,t\neq i}^{n'} \bar{\alpha}_t x_t, x_i) - d^2(y, x_i)). \quad (15)$$

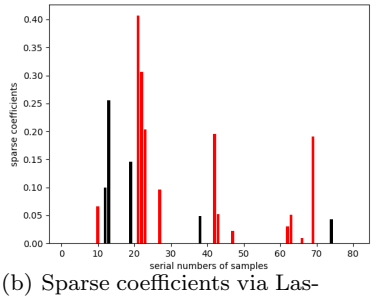
where $d(y, x_i)$ is the Euclidean distance between y and x_i. The detailed derivation of Eq. (15) is beyond the scope of this paper but can be consulted in [8]. Yet, generally, the $d^2(\sum_{t=1,t\neq i}^{n'} \bar{\alpha}_t x_t, x_i)$ denotes the weighted sum of Euclidean distance from x_i to any other $x_t, t \neq i$, with $d(y, x_i)$ expressing the Euclidean distance between y and x_i. The larger $d^2(\sum_{t=1,t\neq i}^{n'} \bar{\alpha}_t x_t, x_i)$ and the smaller $d(y, x_i)$ (equivalently, the larger $\bar{\alpha}_i$), the more similar y is to x_i. Hence, a larger $\|\bar{\alpha}_i\|$ implies that x_i is more likely to be in the neighborhood of y. Conversely, the interpretation os $\bar{\alpha}_i < 0$ is that there is a high probability that x_i does not fall into the same class as y. Thus, data samples whose corresponding coefficients are less than 0 are out of the scope of consideration (which can therefore be set to 0). It can be seen from Eq.(15) that sparse coefficients cover not only the Euclidean distance between two samples, but also linearly reflects the contribution of x_i to its neighbor structure.

To intuitively illustrate the advantage of the SR coefficients produced by EL as similarity measures, comparisons with the Euclidean distance and Lasso on the datasets *sonar* and *wpbc* [31] are shoiwn in Figs. 2 and 3, respectively. For each data set, the 80 training samples (40 from class 0 and 40 from class 1) and one query sample from class 0 are randomly chosen. In Figs. 2 and 3, the similarity measure index corresponding to samples from class 0 is marked with red bar and that from class 1 is marked with black bar. It can be seen from Figs. 2 and 3 that sparse coefficients of EL can retain sufficient information to entail high discriminating ability than Euclidean distances. In most cases, the sparse coefficients of samples from the same class (class 0) as the query sample are always predominant and the ones from other class (class 1) are small or tend to be 0. Thus, sparse coefficients are the better similarity metric to determine the nearest neighbors of a given query sample. Moreover, EL can select more similar samples than Lasso. Therefore, EL is better to deal with the sensitivities to k in kNN-based methods.

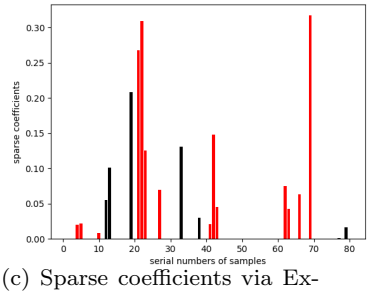
Computationally, there are two loops in the optimization algorithm of EL: one to cover the iterative optimization process and another to iterate through the groups. Let the maximum number of iterations and the



(a) Euclidean distances



(b) Sparse coefficients via Lasso

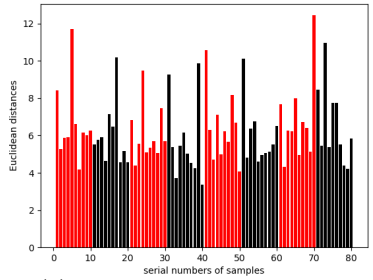


(c) Sparse coefficients via Exclusive Lasso

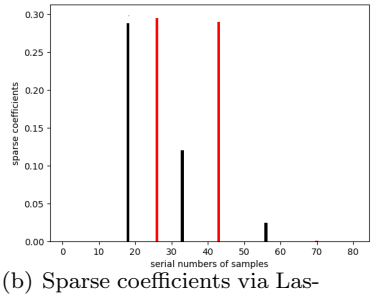
Fig. 2 Illustrative examples on dataset *sonar*.

quantity of the groups be denoted by I and G, respectively. In the worst case (where the optimization process reaches I), the complexity of EL is $\mathrm{O}(I \cdot G)$. Moreover, the complexity of making decision by EkNN-C is $\mathrm{O}(Mk)$ and that by EkNN-R is $\mathrm{O}(Mk + 2k)$. Thus, the complexity of EkNN-C is $\mathrm{O}(I \cdot G + Mk)$ and the complexity of EkNN-R is $\mathrm{O}(I \cdot G + Mk + 2k)$.

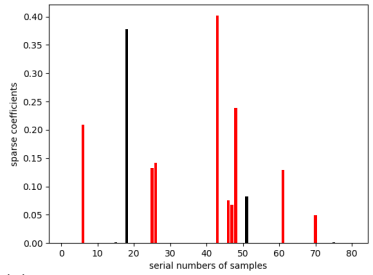
Although the decision framework of the proposed EL-based methods are similar to those of Euclidean distance and Lasso-based kNN methods, the distinguished distribution of neighborhood will grant the proposed approaches a significant discriminative nature. In general, EL results in an intervening performance between Lasso and Euclidean distance, in terms of sparsity. Thus, sparse coefficients of EL enjoy more meaningful information than those of Lasso and more discriminative information than those of Euclidean distance. Such merit can endow the EL-based kNN methods with an abundance of choices of neighborhood and a hindrance of the existence of redundant instances.



(a) Euclidean distances



(b) Sparse coefficients via Lasso



(c) Sparse coefficients via Exclusive Lasso

Fig. 3 Illustrative examples on dataset *wpbc*.

4 Experimental evaluation

This section presents a systematic evaluation of EkNN-C and EkNN-R experimentally. The results and discussions are divided into three different parts, after an introduction to the experimental set-up. The first part compares EkNN-C and EkNN-R with other nearest-neighbor methods in term of classification accuracy. The second part compares EkNN-C and EkNN-R against the state of the art in term of classification accuracy. The last part investigates the performance of EkNN-C and EkNN-R on the application of face recognition.

4.1 Experimental set-up

Twenty-four benchmark data sets [31] [32] [33] are used for the experimental evaluation. The basic information about the data sets as summarized in Table 1. Moreover, three face recognition data sets, including AR [34], Yale [35] and IMM [36] are used to verify the performance of EkNN-C and EkNN-R, compared to their Lasso-based parallels.

Table 1 Benchmark data sets used for evaluation

No.	Data set	Samples	Attributes	Classes
1	LSVT	127	310	2
2	wpbc	197	32	2
3	arcene	200	10000	2
4	sonar	208	60	2
5	ionosphere	230	34	2
6	spectfheart	267	44	2
7	setap	275	85	2
8	bupa	345	6	2
9	liver	345	7	2
10	ILPD	583	11	2
11	Hill_Valley	606	101	2
12	transfusion	748	4	2
13	QSAR	1054	39	2
14	steel	1941	33	2
15	coil2000	9822	85	2
16	wine	178	13	3
17	seeds	210	7	3
18	vehicle	846	13	4
19	cleveland	297	14	5
20	warpAR10P	130	2400	10
21	led7digit	500	7	10
22	multifeat	2000	650	10
23	optdigits	5620	64	10
24	penbased	10992	16	10

The experiments conducted on these date sets apply stratified 10-fold cross-validation (10-FCV) for data validation. In 10-FCV, the original data set is partitioned into 10 subsets. Of these 10 subsets, a single subset is retained as the testing data for the classifier, and the remaining 9 subsets are used for training. The cross-validation process is then repeated 10 times (the number of folds). The 10 sets of results are then aggregated to produce a single classifier estimation. The advantage of 10-FCV over random sub-sampling is that all objects are used for both training and testing, and each object is used for testing only once per fold. The stratification of the data prior to its division into folds ensures that each class label (as far as possible) has equal representation in all folds, thus helping to alleviate bias/variance problems [37].

In addition, the Wilcoxon Signed Rank (WSR) Test is utilized to provide statistical analysis of the resulting classification accuracy. This is done in order to ensure that results are not discovered by chance. In the WSR test, the null-hypothesis should be rejected with a 0.95 confidence interval (or a 0.05 level of significance), when the value of the statistics is lower than 1.96. In Tables

2 and 3, the results on statistical significance by WSR test are summarized in the last three lines, where Z represents the standard score and the p-value reflects the information of significant degree. The true or false significant difference (SD) is recorded in the final line of each table.

4.2 Part 1 - Comparison with alternative nearest neighbor methods

Here, EkNN-C and EkNN-R are compared with five nearest-neighbour classification methods: DWkNN [25], generalized mean distance-based k nearest neighbors (GMDkNN) [3], multi-local means-based k-harmonic nearest neighbor (MLMkHNN) [38], CWkNN [13] and RWkNN [13]. In order to comprehensively evaluate the performance of the proposed classification approaches, k is increasingly set between 1 and 11 in different runs performed by 10-FCV. The results can been seen in Figs. 4 and 5.

From the experimental results presented, it can be seen that for most cases, EkNN-C and EkNN-R outperform kNN, DWkNN, GMDkNN, MLMkHNN, CWkNN and RWkNN. In particular, the performance of EkNN-C transcends those of the alternative competitors, consistently. Furthermore, the classification accuracy of both proposed classification systems remains on a generally upward trend as the value of k grows. These experimental results manifest that, compared with Euclidean distance-based and Lasso-based kNN strategies, the proposed methods are more robust (or less sensitive) regarding the number of nearest neighbors and hence, embrace less redundant information in each neighborhood of interest.

4.3 Part 2 - Comparison with the state of the art: use of different aggregators

This section experimentally compares EkNN-C and EkNN-R with several leading classifier learners that represent a cross section of the most popular approaches, including DGC [39], NB [40], JRip [41], J48 [42], AdaJ48 [43], RF [44], Bagging [45], and DNN [46]. For completeness, a brief summary of these methods is provided below:

- Data Gravitation-based Classification (DGC) [39] is based on the concept of data gravitation. Its main principle is to classify data samples by comparing the data gravitation between the different data classes. A larger gravitation from a class means the data sample belongs to a particular class.
- Naive Bayes (NB) [40] is a classification method based on Bayes' theorem and strong (naive) independence assumptions. For a given training data set, the joint probability distribution of input/output is first learned based on the independent assumption of feature conditions. Then based on this model, the output with the maximum posterior probability is obtained by using Bayes' theorem for the given input.
- JRip [41] learns propositional rules by repeatedly growing rules and pruning them. During the growth phase, features are added greedily until a termination condition is satisfied. Features are then pruned in the next phase subject to a pruning metric. Once the rule set is generated, a further optimization is performed where classification rules are evaluated and deleted based on their performance on randomized data. In this paper, JRip is set with folds=3, minNo=2, optimazations=2 and seed=1.
- J48 is based on ID3 [42] and creates decision trees by choosing the most informative features and recursively partitioning a training data table into subtables based on the values of such features. Each node in the tree represents a feature, with the subsequent nodes branching from the possible values of this node according to the current subtable. Partitioning stops when all data items in the subtable have the same classification. In this paper, J48 is set with the pruning confidence threshold C=0.25.
- In AdaBoostM1 (AdaM1) [43] algorithm , each training sample is given a weight and the weight represents the probability that the sample will be listed in the training sample set by the next weak learn. If a sample can be accurately classified by the current weak learn, the probability that the sample will be selected will be reduced when constructing the training sample set of the next weak learn; on the contrary, If a sample fails to be classified correctly by the current classifier, its weight will be increased accordingly, which strengthens the classification ability of the harder samples. The final classification output depends on the comprehensive effect of all classifiers. In this paper, AdaBoostM1 is set with J48 classifier, num Iterations=10, seed=1 and weight Threshold=100.
- Random Forests (RF) [44] is an algorithm that integrates multiple trees with the idea of ensemble learning. Its basic unit is decision tree. For each tree, the training set used is sampled from the total training set in a way that is put back. When the nodes of each tree are trained, the features used are extracted from all the features in a certain proportion in

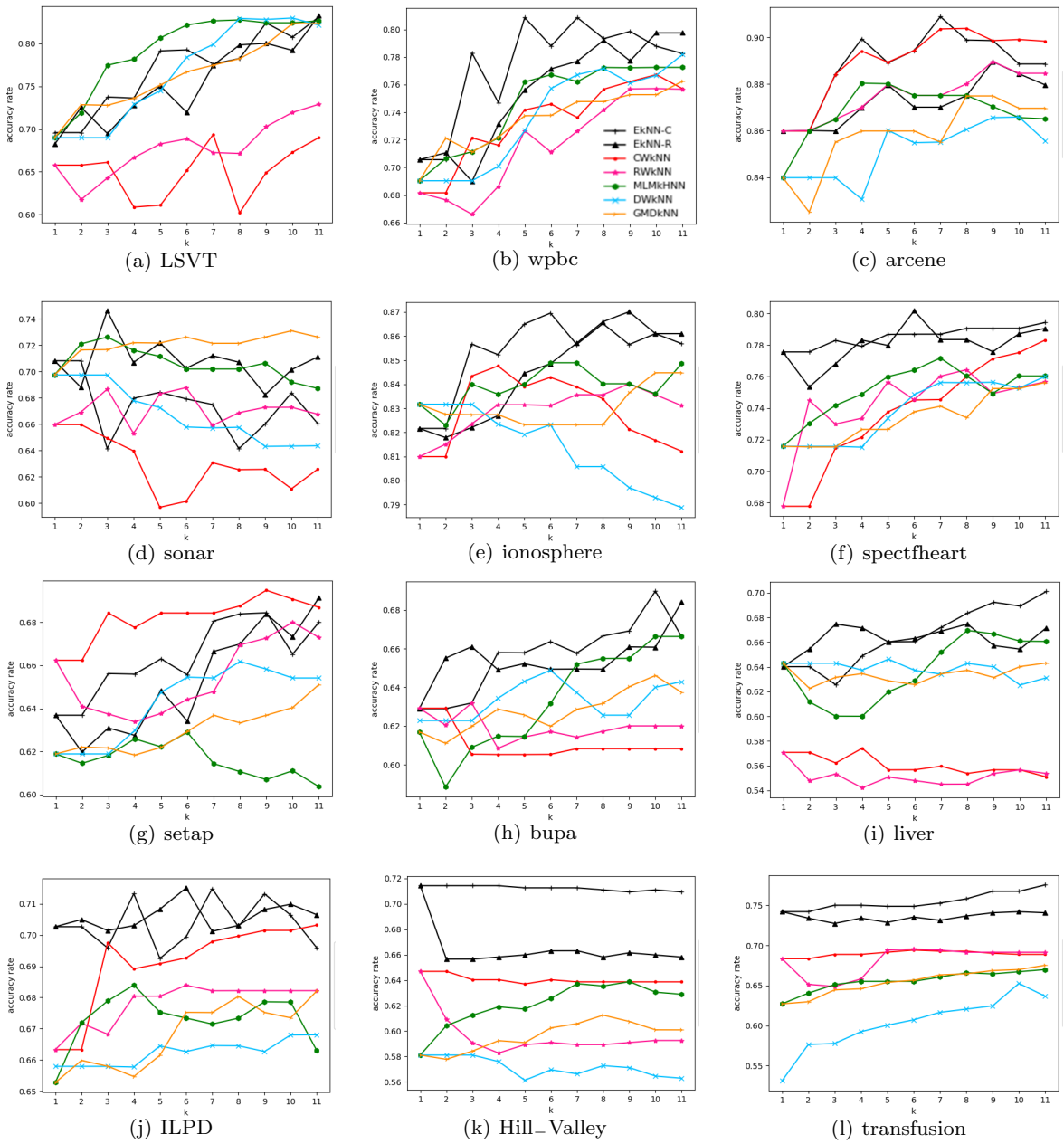


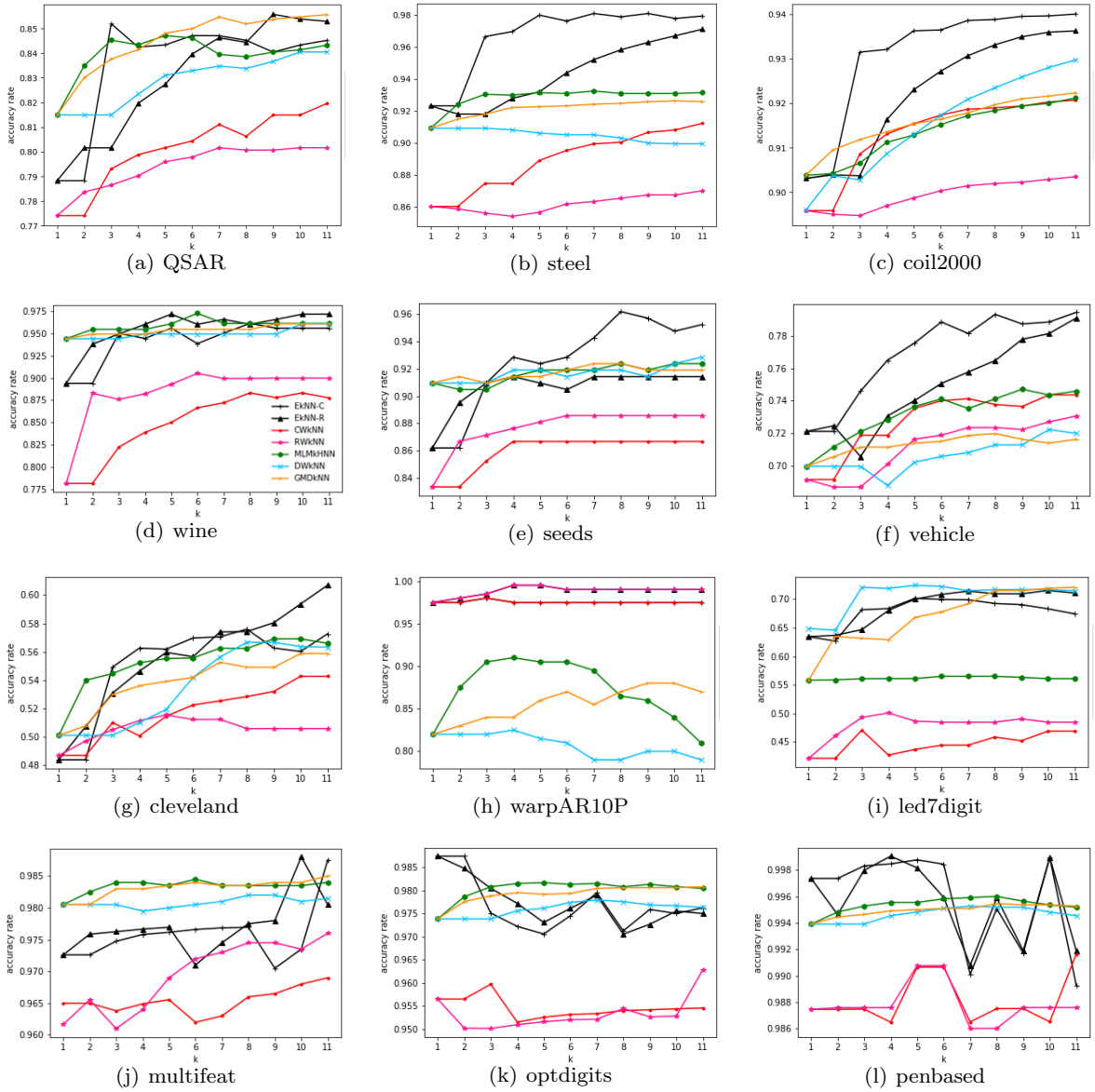
Fig. 4 Comparisons with other alternative nearest-neighbour algorithms on data sets 1-12.

a random way without putting back. In this paper, RF is set with numTrees=100 and seeds=1.

– Bagging [45] is one of the most basic integration technologies. It is based on a bootstrapping statistical approach, which makes many statistical evaluations of complex models feasible. It can reduce overfitting.

– Deep Neural Networks (DNN) [46] is a feedforward artificial neural network model, which belongs to nonparametric estimation and can be used to solve classification and regression problems. In this paper, DNN is set with activation='relu', batchSize=100, trainingTime=500, learningRate=0.3 and 2 hiddenLayers, where the numbers of the hidden nodes are set as the summation and the average of the values of Attributes and Classes as shown in Table 1, respectively.

The results are listed in Tables 2 and 3, together with a statistical comparison of each method against EkNN-C and EkNN-R, respectively. The baseline references for the p-tests carried out are the highest classification accuracies obtained by the proposed classifiers for each data set as shown in Fig. 4 and 5. From the results of this experimentation, across all data sets used,



(a) QSAR (b) steel (c) coil2000

(d) wine (e) seeds (f) vehicle

(g) cleveland (h) warpAR10P (i) led7digit

(j) multifeat (k) optdigits (l) penbased

Fig. 5 Comparisons with other alternative nearest-neighbour algorithms on data sets 13-24.

it can be seen that the resulting average accuracies of the proposed algorithms systematically beat the rest. Particularly, EkNN-C and EkNN-R statistically outperform DGC, NB, JRip, J48, RF and DNN, since the p-values are smaller than 0.05, and the performance improvement over that achievable by DGC, NB, JRip and J48 is highly significant, given that p-values$<$ 0.01. Occasionally, the differences between the resultant average accuracies obtained by the proposed methods and those attainable by AdaJ48, RF and Bagging are not statistically significant. Nonetheless, the results returned by the proposed are still greater than those achievable by the rest numerically.

4.4 Part 3 - Face Recognition

The data sets employed in this experimental evaluation are IMM [36] (as shown in Fig. 6), Yale [35] (as shown in Fig. 7) and AR [34] (as shown in Fig. 8). Specifically, the IMM data set contains 240 images of faces, involving a total of 40 people (7 women and 33 men) and each people with 6 face images. The Yale data set consists of 15 classes, each of which includes 11 samples. And the AR data set consists of 2600 images of faces from 100 people (50 women and 50 men), each of which has 26 face images.

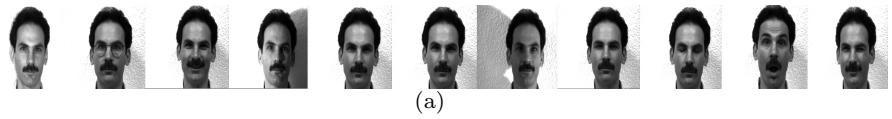
The respective experimental results on these three data sets are shown in Figs. 9, 10 and 11. The highest accuracies of each classification method on these

Table 2 Classification accuracy: EkNN-C versus others

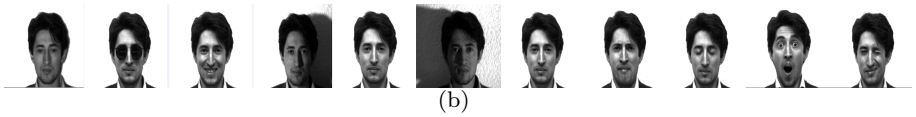
dataset	EkNN-C	DGC	NB	Jrip	J48	AdaJ48	RF	Bagging	DNN
LSVT	83.13	62.37	56.41	74.68	75.45	79.83	81.73	66.67	84.61
wpbc	80.86	76.64	68	69.87	73.13	76.18	74.24	79.18	76.7
arcene	90.89	57	68.7	71.5	75	43.58	75.7	75.9	85
sonar	70.83	66.22	67.88	76.95	71.17	79.13	71.67	76.9	72.02
ionosphere	86.96	90.63	83.91	86.52	87.83	90.09	88.26	87.83	88.20
spectfheart	79.43	68.86	68.63	78.25	74.91	79.04	79.83	81.65	77.11
setap	68.45	67.56	66.51	73.82	66.65	73.38	69.43	68.02	63.63
bupa	68.97	56.25	53.86	65.84	67.87	68.37	68.71	71.93	69.22
liver	70.12	59.12	55.39	64.64	68.71	69.42	69.89	69.58	66.06
ILPD	71.48	58.16	55.74	70.15	68.79	70.46	71.85	71.52	68.26
Hill_Valley	71.43	53.98	50.67	49.34	50.33	50.33	58.76	59.83	84.82
transfusion	77.54	63.87	75.4	77.14	77.81	77.44	72.86	79.15	72.62
QSAR	85.2	67.09	76.84	81.69	83.01	85.69	84.82	84.91	82.24
steel	98.09	56.28	55.48	100	100	100	99.33	100	97.42
coli2000	94	91.97	78.08	93.91	93.95	91.64	92.82	93.87	89.94
wine	96.11	98.30	97.22	92.68	94.41	97.19	98.3	94.97	97.25
seeds	96.19	62.38	91.43	86.67	91.9	92.67	92.86	92.86	91.90
vehicle	79.43	70.69	44.8	69.39	72.47	75.59	74.94	72.1	83.34
cleveland	57.59	52.48	56.6	52.18	51.87	53.18	54.9	57.6	47.99
warpAR10P	98	70.32	72.15	59.69	70.31	83.92	74.38	74.31	46
led7digits	70.16	50.62	70.8	71	71.2	72.1	70.4	72	71.99
multifeat	98.75	80.84	95.35	93.1	94.75	97.7	96.55	96.25	98.24
optdigits	98.74	83.66	91.33	91.25	90.68	97.35	96.92	95.32	97.54
penbased	99.54	79.04	85.86	96.39	96.52	98.92	98.97	97.75	98.91
Average	82.99	68.51	70.29	76.94	77.86	79.30	79.92	80.00	79.61
Z	-	-4.09	-4.17	-3.17	-3.6	-1.71	-2.37	-1.63	-2.11
p-value	-	0.00004	0.00003	0.00152	0.00032	0.08647	0.01771	0.1034	0.03448
SD	-	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE

(a) (b)

Fig. 6 Two samples of dataset *IMM*



(a)



(b)

Fig. 7 Two samples of dataset *Yale*

(a) (b)

Fig. 8 Two samples of dataset *AR*

data sets, along with the corresponding values of k, are concluded in Table 4. Overall, for most values of k, the EkNN-C and EkNN-R outperform their Lasso-based parallels, respectively. Only for 6 cases of k, i.e.,

Table 3 Classification accuracy: EkNN-R versus others

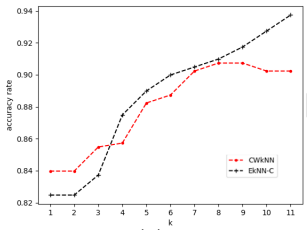
dataset	EkNN-R	DGC	NB	Jrip	J48	AdaJ48	RF	Bagging	DNN
LSVT	83.25	62.37	56.41	74.68	75.45	79.83	81.73	66.67	84.61
wpbc	79.75	76.64	68	69.87	73.13	76.18	74.24	79.18	76.7
arcene	88.96	57	68.7	71.5	75	43.58	75.7	75.9	85
sonar	74.63	66.22	67.88	76.95	71.17	79.13	71.67	76.9	72.02
ionosphere	87.01	90.63	83.91	86.52	87.83	90.09	88.26	87.83	88.2
spectfheart	80.17	68.86	68.63	78.25	74.91	79.04	79.83	81.65	77.11
setap	69.15	67.56	66.51	73.82	66.65	73.38	69.43	68.02	63.63
bupa	68.42	56.25	53.86	65.84	67.87	68.37	68.71	71.93	69.22
liver	67.47	59.12	55.39	64.64	68.71	69.42	69.89	69.58	66.06
ILPD	71.51	58.16	55.74	70.15	68.79	70.46	71.85	71.52	68.26
Hill_Valley	71.43	53.98	50.67	49.34	50.33	50.33	58.76	59.83	84.82
transfusion	74.21	63.87	75.4	77.14	77.81	77.44	72.86	79.15	72.62
QSAR	85.57	67.09	76.84	81.69	83.01	85.69	84.82	84.91	82.24
steel	97.12	56.28	55.48	100	100	100	99.33	100	97.42
coli2000	93.63	91.97	78.08	93.91	93.95	91.64	92.82	93.87	89.94
wine	97.15	98.30	97.22	92.68	94.41	97.19	98.3	94.97	97.19
seeds	91.43	62.38	91.43	86.67	91.9	92.67	92.86	92.86	91.90
vehicle	79.08	70.69	44.8	69.39	72.47	75.59	74.94	72.1	83.34
cleveland	60.68	52.48	56.6	52.18	51.87	53.18	54.9	57.6	47.99
warpAR10P	99.5	70.32	72.15	59.69	70.31	83.92	74.38	74.31	46
led7digits	71.42	50.62	70.8	71	71.2	72.1	70.4	72	71.99
multifeat	98.79	80.84	95.35	93.1	94.75	97.7	96.55	96.25	98.24
optdigits	98.74	83.66	91.33	91.25	90.68	97.35	96.92	95.32	97.54
penbased	99.89	79.04	85.86	96.39	96.52	98.92	98.97	97.75	98.91
Average	82.87	68.51	70.29	76.94	77.86	79.30	79.92	80.00	79.61
Z	-	-4.11	-4.17	-3.14	-3.11	-1.17	-2.31	-1.28	-2.06
p-value	-	0.00004	0.00005	0.00167	0.00184	0.24142	0.02065	0.19854	0.03967
SD	-	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE

$k = 1, 2, 3$ by EkNN-C on the IMM data set, $k = 3$ by EkNN-R on the Yale data set and $k = 4, 5$ by EkNN-R on the AR data set, the proposed methods endure a inferior performance compared to the Lasso-based approaches.

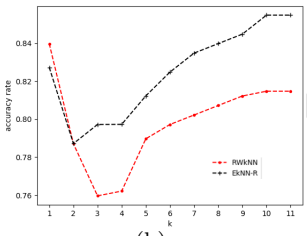
Table 4 Highest classification accuracies of different nearest neighbour algorithms

dataset	IMM	Yale	AR
EkNN-C	0.9374(11)	0.9531(9)	0.9905(11)
CWkNN	0.9073(9)	0.9400(6)	0.9883(11)
EkNN-R	0.8548(11)	0.9333(1)	0.9566(11)
RWkNN	0.8397(11)	0.9298(1)	0.9500(1)

In summary, examining all of the results obtained, it has been experimentally shown that with the use of the coefficients of EL, the kNN classification offers a better and more robust performance than the other classifiers.



(a)



(b)

Fig. 9 Classification accuracy with respect to different k values for dataset *IMM*

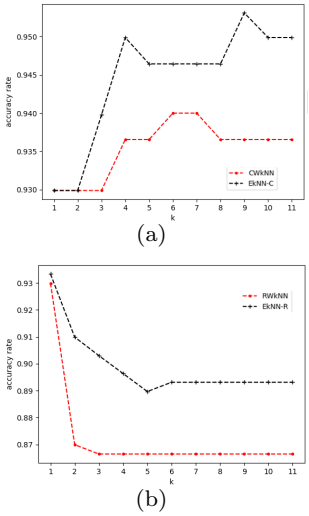


Fig. 10 Classification accuracy with respect to different k values for dataset *Yale*

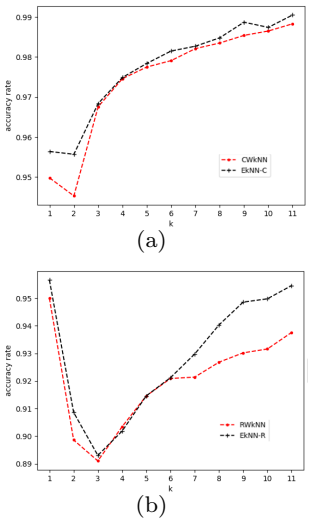


Fig. 11 Classification accuracy with respect to different k values for dataset *AR*

5 Conclusions

In this paper, two novel k nearest neighbor classification strategies are proposed by the EL regularization. The sparse coefficients of EL are employed to perform the similarity evaluation which carries out variable selection at the group level. To demonstrate the efficacy of the proposed methods, systematic experiments have been carried out from the perspective of classification accuracy on both benchmark data sets and face recognition applications. The results of the experimental evaluation show that the EL-based k nearest neighbors methods generally outperform a range of state-of-the-art learning classifiers for these classification tasks.

Topics for further research include a more comprehensive study of how EL could be used for other tasks such as attribute reduction. Also, due to the feature selection capacity of EL, the concept of data reliability based on the proposed nearest neighbor strategies is a worthwhile avenue of exploration. An investigation into how such work may be combined with the alternative classification indicators, in order to reinforce the potential of these approaches, remains active research.

Conflicts of interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Acknowledgments

This work was jointly supported by the Innovation Support Plan for Dalian High-level Talents (No. 2018RQ70) and partly by two awards under the Sêr Cymru II COFUND Fellowship scheme, UK. The authors are grateful to the anonymous reviewers for their constructive comments, which have helped improve this work significantly.

References

1. T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
2. Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.
3. Jianping Gou, Hongxing Ma, Weihua Ou, Shaoning Zeng, Yunbo Rao, and Hebiao Yang. A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications*, 115:356–372, 2019.
4. S. Z. Li and Juwei Lu. Face recognition using the nearest feature line method. *IEEE Transactions on Neural Networks*, 10(2):439–443, 1999.
5. Qing-Bin Gao and Zheng-Zhi Wang. Center-based nearest neighbor classifier. *Pattern Recognition*, 40(1):346 – 349, 2007.
6. David, L., and Donoho. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 2006.
7. Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang. A survey of sparse representation:

Algorithms and applications. *IEEE Access*, 3:490–530, 2017.

8. Jian Zhang and Jian Yang. Linear reconstruction measure steered nearest neighbor classification framework. *Pattern Recognition*, 47(4):1709 –1720, 2014.
9. Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statal Society, Series B*, 58(1), 1996.
10. J. Wright, A. Y. Yang, A. Ganesh, S. S.Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.
11. Jiang Li and Can-Yi Lu. A new decision rule for sparse representation based classification for face recognition. *Neurocomputing*, 116:265 – 271, 2013.
12. Yong Xu, Qi Zhu, Zizhu Fan, Minna Qiu, Yan Chen, and Hong Liu. Coarse to fine k nearest neighbor classifier. *Pattern Recognition Letters*, 34(9):980 – 986, 2013.
13. H. Ma, J. Gou, X. Wang, J. Ke, and S. Zeng. Sparse coefficient-based k -nearest neighbor classification. *IEEE Access*, 5:16618–16634, 2017.
14. Shichao Zhang, Debo Cheng, Zhenyun Deng, Ming Zong, and Xuelian Deng. A novel knn algorithm with data-driven k parameter computation. *Pattern Recognition Letters*, 109:44 – 54, 2018.
15. Wright, John, Yang, Allen, Y., Ganesh, Arvind, Sastry, S., and Shankar and. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.
16. Meng Yang, Lei Zhang, Jian Yang, and David Zhang. Robust sparse coding for face recognition. *CVPR 2011*, 42(7):625–632, 2011.
17. Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statal Society*, 68(1):49–67, 2006.
18. Laurent Jacob, Guillaume Obozinski, and Jean Philippe Vert. Group lasso with overlap and graph lasso. In *International Conference on Machine Learning*, 2009.
19. Juan Chen A, Shijie Zhou B, Zhao Kang A, and Quan Wen A. Locality-constrained group lasso coding for microvessel image classification - sciencedirect. *Pattern Recognition Letters*, 130:132–138, 2020.
20. Zhenkun Diwu, Hongrui Cao, Lei Wang, and Xuefeng Chen. Collaborative double sparse period-group lasso for bearing fault diagnosis. *IEEE Trans. Instrum. Meas.*, 70:1–10, 2021.
21. Shichao Zhang, Ming Zong, K. Sun, Yue Liu, and Debo Cheng. Efficient knn algorithm based on graph sparse reconstruction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8933:356 – 369, 2014.
22. Tang Yufang, Li Xueming, Xu Yan, and Liu Shuchang. Group lasso based collaborative representation for face recognition. In *2014 4th IEEE International Conference on Network Infrastructure and Digital Content*.
23. Shuai Zheng and Chris Ding. A group lasso based sparse knn classifier. *Pattern Recognition Letters*, 131:227–233, 2020.
24. Yang Zhou, Rong Jin, and Steven C. H. Hoi. Exclusive lasso for multi-task feature selection. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 988–995. JMLR.org, 2010.
25. Jianping Gou, Lan Du, Yuhong Zhang, and Taisong Xiong. A new distance-weighted k-nearest neighbor classifier. *Journal of Information and Computational Science*, 9(6), 2012.
26. Frederick Campbell and Genevera I. Allen. Within group variable selection through the exclusive lasso. *Electronic Journal Of Statistics*, 11(2):4220–4257, 2017.
27. Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *Annals Of Statistics*, 37(6A):3468–3497, 2009.
28. Guillaume Obozinski and Francis Bach. Convex relaxation for combinatorial penalties. *Eprint Arxiv*, 2012.
29. Deguang Kong, Ryohei Fujimaki, Ji Liu, Feiping Nie, and Chris Ding. Exclusive feature learning on arbitrary structures via l(1,2)-norm. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, volume 27, 2014.
30. Yuxin Sun, Benny Chain, Samuel Kaski, and John Shawe-Taylor. Correlated feature selection with extended exclusive group lasso. *CoRR*, abs/2002.12460, 2020.
31. Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
32. Isaac Triguero, Sergio Gonzlez, Jose M. Moyano, Salvador Garca, and Francisco Herrera. Keel 3.0: An open source software for multi-stage analysis in data mining. *International Journal of Computational Intelligence Systems*, 10(1):1238–1249, 2017.
33. Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2018.

34. A. Martinez and Robert Benavente. The ar face database. *Tech. Rep. 24 CVC Technical Report*, 01 1998.
35. A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 23(6):643–660, 2002.
36. M. B. Stegmann, B. K. Ersboll, and R. Larsen. Fame-a flexible appearance modeling environment. *IEEE Transactions on Medical Imaging*, 22(10):1319–1331, 2003.
37. Y. Bengio and Yves Grandvalet. *Bias in Estimating the Variance of K-Fold Cross-Validation*, pages 75–95. 2005.
38. Zhibin Pan, Yidi Wang, and Weiping Ku. A new k-harmonic nearest neighbor classifier based on the multi-local means. *Expert Systems with Application*, 67(jan.):115–125, 2017.
39. Lizhi Peng, Bo Yang, Yuehui Chen, and Ajith Abraham. Data gravitation based classification. *Information Sciences*, 179(6):809–819, 2009.
40. GH John and P Langley. Estimating continuous distributions in bayesian classifiers. In *Eleventh conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.
41. William W. Cohen. Fast effective rule induction. *Machine Learning Proceedings*, 95:115–123, 1995.
42. J Ross Quinlan. *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers Inc., 1992.
43. Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm draft-please do not distribute. In *Thirteenth International Conference on International Conference on Machine Learning*, 1996.
44. Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.
45. Breiman and Leo. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
46. David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back propagating errors. *Nature*, 323(6088):533–536, 1986.