Pattern Recognition Letters 110 (2018) 36–43

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Redefining nearest neighbor classification in high-dimensional settings

Julio López [a], Sebastián Maldonado [b,*]

[a] Facultad de Ingeniería y Ciencias, Universidad Diego Portales, Av. Ejército 441, Santiago, Chile
[b] Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Mons. Álvaro del Portillo 12455, Las Condes, Santiago, Chile

ARTICLE INFO

Article history:
Received 21 July 2017
Available online 27 March 2018

Keywords:
Nearest neighbor classification
High-dimensional datasets
Distance metrics

ABSTRACT

In this work, a novel nearest neighbor approach is presented. The main idea is to redefine the distance metric in order to include only a subset of relevant variables, assuming that they are of equal importance for the classification model. Three different distance measures are redefined: the traditional squared Euclidean, the Manhattan, and the Chebyshev. These modifications are designed to improve classification performance in high-dimensional applications, in which the concept of distance becomes blurry, i.e., all training points become uniformly distant from each other. Additionally, the inclusion of noisy variables leads to a loss of predictive performance if the main patterns are contained in just a few variables, since they are equally weighted. Experimental results on low- and high-dimensional datasets demonstrate the importance of these modifications, leading to superior average performance in terms of Area Under the Curve (AUC) compared with the traditional k nearest neighbor approach.

1. Introduction

The k Nearest Neighbor (k-NN) classifier [5] is a well-known pattern recognition method that has been used widely in several applications. Simplicity is its main virtue, allowing the classification of two or more patterns based on a quite simple rule: a test sample will belong to the class that the majority of its k nearest neighbors belongs to Han and Kamber [9].

Due to this simplicity, the k-NN method has several issues to deal with. Two main shortcomings, which are related to high-dimensional applications, are discussed in this paper. First, metrics such as the Euclidean distance may not be suitable in this context, since the concepts of distance and proximity are ill defined [10,20]. A second issue is feature relevancy: in contrast to methods such as logistic regression or Support Vector Machines, the feature importance cannot be derived with the original version of k-NN, and all variables are assumed to be equally important in obtaining the neighbors [9]. This fact can cause poor prediction if most variables are irrelevant, 'diluting' the patterns present in the relevant variables. Nowadays, there are several applications that have hundreds or even thousands of potentially redundant or irrelevant variables. In most cases, all the information is collected at once, and it is not clear which variables are relevant a priori. For such applications, models are required for helping us disentangling the signal from the noise.

In this work, these two issues are taken into account in order to design robust k-NN classifiers for both low- and high-dimensional settings. Three different distance metrics (Euclidean, Manhattan, and Chebyshev) are studied and modified in order to incorporate only a subset of the available information. Filter methods for feature selection are embedded in the definition of the distance metric, in order to encourage sparsity based on only the most relevant variables for the problem.

The remainder of this paper is organized as follows: previous work on k-NN is discussed in Section 2. The proposed framework for k-NN classification based on novel distance metrics is described in Section 3. In Section 4, experimental results using binary classification datasets are given. Finally, the main conclusions of this study and ideas for future developments are presented in Section 5.

2. k-NN classification

The k Nearest Neighbor method is arguably the simplest pattern recognition method for classification [9]. Given a fixed value for k, i.e., the number of neighbors, this nonparametric approach assigns the class label y^* to an unlabeled sample $\mathbf{x}^*$, which occurs most frequently in its neighborhood of k closest examples from the training set [5].

Formally, given two sets of m training tuples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, where $\mathbf{x}_i \in \Re^n$ and $y_i \in \{-1, +1\}$ are its respective class labels, and given mt test samples $\{\mathbf{x}_l^t\}_{l=1}^{mt} \subset \Re^n$, this method first computes the distance between an unlabeled sample $\mathbf{x}_l^t$ and all training examples $\mathbf{x}_i$, for $i = 1, \ldots, m$. Assuming a set $\mathcal{S}$ that contains all variables

* Corresponding author.
E-mail address: smaldonado@uandes.cl (S. Maldonado).

https://doi.org/10.1016/j.patrec.2018.03.023
0167-8655/

($|\mathcal{S}| = n$), the squared Euclidean distance is usually used for this task [9]:

$$d_{l_2}(\mathbf{x}_l^t, \mathbf{x}_i) = \sum_{j \in \mathcal{S}} (x_{l,j}^t - x_{i,j})^2. \quad (1)$$

Next, the k training observations that are closest to $\mathbf{x}_l^t$ are selected, i.e., the k elements with lowest $d_{l_2}(\mathbf{x}_l^t, \mathbf{x}_i)$ for all $i = 1, \ldots, m$. The label assigned to $\mathbf{x}_l^t$ is the most frequent one among these k elements.

Several improvements to the traditional k-NN algorithm have been developed in recent years. One research line involves using alternative distance measures for improving performance [17,26,27] or dealing with different types of data [4,6]. For example, a penalty dissimilarity measure was proposed in Datta et al. [6] in order to deal with missing information. Cost and Salzberg [4] proposed a weighted measure for handling symbolic features. Variations of the Minkowski distance have been used previously in domains such as anomaly [19] and intrusion detection for preventing network attacks [17].

An important aspect of this research is adapting the k-NN distance metric for dealing with high dimensionality. Few studies have been proposed in this direction. Hastie and Tibshirani [10] proposed a locally adaptive strategy to try to ameliorate this course of dimensionality in k-NN classification, and Pal et al. [20] proposed a dissimilarity measure based on mean absolute differences between inter-point distances. The latter strategy reduces the negative effects caused by a high dimensionality, such as the concentration of pairwise distances, thus improving predictive performance.

A related research line uses the k-NN principles for performing feature selection. Navot et al. [18] proposed a feature-weighted k-NN version for simultaneous regression and feature selection. This strategy was used to model cortical neural activity. Li et al. [13] developed an ensemble strategy based on various k-NN classifiers, which were constructed based on random subsets of variables. This approach, which resembles the reasoning behind random forest, can be used as feature ranking, and subsequently for performing backward feature elimination. Another feature selection method that uses the ideas behind random forest and k-NN was proposed by Park and Kim [21].

Other heuristics have been used for performing feature selection and k-NN classification. For example, Tahir et al. [23] proposed a hybrid approach based on a Tabu search for simultaneous feature weighting and classification. Lee et al. [12] used genetic algorithms for dealing with the issue of having various scales in the datasets. The authors proposed an efficient k-NN reference set editing strategy for maximizing accuracy, while reducing running times and memory resources.

Efficiency has also been a relevant topic in the k-NN literature. Beliakov and Li [1] proposed an efficient strategy for replacing the sort operation, by using order statistics and parallel computing via GPUs. Li et al. [14] developed a strategy for reducing the number amount of target samples to be considered by creating partial sets of the nearest neighbors.

3. Proposed strategy for nearest neighbor classification

Dimensionality reduction is quite an important topic in pattern recognition. A low-dimensional data representation reduces the risk of overfitting by constructing simple models with few parameters, yielding to a better predictive performance. It also provides a better understanding of the outcome of the model while reducing storage and acquisition costs [15]. In pattern recognition and image processing, dimensionality reduction is related with feature extraction, which corresponds to the process of constructing new features from the original dataset, aiming at reducing redundancy and identifying latent dimensions of the image that describe the data with sufficient accuracy.

Most methods are able to deal with noisy/irrelevant features by either removing them during the model training, or weighting them down when constructing a separating hyperplane. Decision trees fall in the first category, while methods such as logistic regression, SVM, or ANN on the second. In contrast, k-NN weights all variables equally, and it is usually outperformed by these alternative methods for this reason.

The main idea of the proposed approach is to adapt the classic k-NN method in order to deal with the two main issues pointed out in the introduction: the course of dimensionality faced by distance metrics such as the Euclidean norm, and the problem of having equal weights for all variables, which may lead to poor prediction if redundant or irrelevant variables, are included in the k-NN classification task.

Our contribution is twofold: first, we propose variations of the Minkowski distance that are more suitable under conditions of high-dimensionality, such as the Chebyshev distance or l_∞-norm. Additionally, we propose a modification of the Minkowski metric as the distance of two samples based only on a subset of the available variables, demonstrating that this modified Minkowski distance can, indeed, be considered as a distance measure.

Formally, the following distance metric is proposed for a given set of variables $\mathcal{U} \subset \mathcal{S}$, which is a subset of the full set of features $\mathcal{S}$, and two data objects $\mathbf{x}_k \in \Re^{|\mathcal{S}|}$ for $k = \{1, 2\}$:

$$d_{l_p,\mathcal{U}}(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|_{p,\mathcal{U}} = \left(\sum_{j \in \mathcal{U} \subset \mathcal{S}} |x_{1,j} - x_{2,j}|^p \right)^{1/p}, \quad (2)$$

for $p \geq 1$. This distance is designed to be used with $p \in \{1, 2, \infty\}$, i.e., the Manhattan, Euclidean, and Chebyshev distances, respectively. The proof that the proposed modified Minkowski distance satisfies the various properties required for being a distance measure is presented in Appendix A (see online supplementary material).

Next, the modified k-NN algorithm is proposed. The inputs of the model are the training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, the (unlabeled) test objects $\{\mathbf{x}_l^t\}_{l=1}^{mt}$, a predefined number of nearest neighbors k, a predefined number of selected attributes r, and the Minkowski distance parameter $p \in \{1, 2, \infty\}$. The output is the label vector for the test samples. The proposed strategy is formalized in Algorithm 1.

Algorithm 1 Modified k-NN method.

Input: Training tuples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$; Test samples $\{\mathbf{x}_l^t\}_{l=1}^{mt}$; Number of nearest neighbors k; Feature ranking strategy FR; Number of selected attributes r; Distance parameter $p \in \{1, 2, \infty\}$.

Output: Test labels $\{\mathbf{y}_l^t\}_{i=1}^{mt}$.

1: $\mathbf{R} \leftarrow$ *Feature ranking resulting from using strategy FR on the training samples.*
2: $\mathcal{U} \leftarrow$ *subset of attributes corresponding to the r largest values of rank* $\mathbf{R}$.
3: **for** $l = 1, \ldots, mt$ **do**
4: Compute the distance between the sample $\mathbf{x}_l^t$ and all the training samples $\mathbf{x}_i$, $i = 1, \ldots, m$, using the modified Minkowski distance $d_{l_p,\mathcal{U}}(\mathbf{x}_l^t, \mathbf{x}_i)$.
5: $\mathcal{N}_l \leftarrow$ *Subset of the k nearest neighbors from the training set of the test sample* $\mathbf{x}_l^t$.
6: $\mathbf{y}_l^t \leftarrow$ *Label corresponding to the mode in* $\mathcal{N}_l$.
7: **end for**

The first step of the algorithm corresponds to the construction of the feature ranking $\mathbf{R}$. This ranking is constructed by sorting the variables according to its relevancy using, e.g. statistical measures.

The following ranking strategies are proposed for the first step of Algorithm 1:

- **Fisher Score:** This approach assesses relevancy by computing the difference between the mean values of the two classes, assuming a binary classification problem, and dividing it by a combined standard deviation [8]. For a given variable j, its Fisher Score $FS(j)$ follows:

$$FS(j) = \left| \frac{\mu_j^+ - \mu_j^-}{(\sigma_j^+)^2 + (\sigma_j^-)^2} \right|, \tag{3}$$

  where μ_j^+ and μ_j^- are the averages for the j-th variable in the positive and negative classes, respectively, while σ_j^+ and σ_j^- are their respective standard deviations.
- **Mutual Information:** Another well-known strategy for assessing relevance is using the Mutual Information (MI) measure [25]. Loosely speaking, this metric evaluates the amount of information obtained about one random variable by observing the other. In order to use MI as a feature selection strategy, the label variable is compared with all the covariates to assess dependency. In contrast with the Fisher Score, MI was originally developed for categorical variables, therefore numerical inputs have to be discretized by binning. For a given variable j, its Mutual Information $MI(j)$ follows:

$$MI(j) = \sum_{y \in \mathbf{y}} \sum_{x \in \mathbf{x}_j} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right), \tag{4}$$

  where x and y are the various levels related to variable $\mathbf{x}_j$ and the label vector $\mathbf{y}$, respectively; and $p(x)$ and $p(y)$ are their marginal probability distributions, with $p(x, y)$ being their joint distribution.
- **Eigenvector Centrality:** The reasoning behind Eigenvector Centrality (EC) is computing a measure that defines pairwise relationships among variables, by constructing an affinity graph and weighting the edges formed by the various covariates [22]. The EC metric combines the Fisher Score, the Mutual Information, and the variables' standard deviation in order to define an adjacency matrix A, whose edges $a_{jj'}$ can be interpreted as the discriminative power of attributes j and j', when they are taken into account jointly. A feature ranking can be constructed by computing the eigenvector related to the largest eigenvalue of A.
- **Correlation Score**: Unlike previous measures, the Correlation score (CFS) aims at taking the redundancy among the variables into account, rather than the dependency between the label vector and the covariates [8]. In a first step, the pairwise Pearson's correlation is computed for each pair of variables j and j':

$$\rho_{j,j'} = \frac{\Sigma_{j,j'}}{\sigma_j \sigma_{j'}}, \tag{5}$$

  where $\Sigma_{j,j'}$ corresponds to the covariance of j and j'. A feature ranking can be constructed by finding the lowest absolute correlation for each variable, and sorting them in ascending order:

$$CFS(j) = \min_{j'} |\rho_{j,j'}|. \tag{6}$$

Notice that there is a plethora of ranking methods that can be used as an alternative for this step. We refer the reader to the book by Guyon et al. [8], chapter 3: Filter methods. Good alternatives for this step are methods rank the variables by balancing redundancy and relevance. For example, the Scalar Selection Technique (SST) [24] ranks the variables by balancing redundancy and relevance. For a given attribute, SST computes a weighted sum by combining the Fisher Score and the sum of the bivariate correlations between it and the rest of the variables.

The second step of the algorithm simply identifies the top r variables in terms of relevancy according to the ranking $\mathbf{R}$. These variables define the subset $\mathcal{U} \subset \mathcal{S}$, which is used to define the neighborhood for each new sample. Notice that the definition of the neighborhood is not fully unsupervised because of this step, in contrast with the traditional k-NN approach.

Steps 3 to 7 in Algorithm 1 correspond to the modified k-NN method: Step 4 computes the distance between each unlabeled object and all the training samples using the modified Minkowski distance (Eq. (2)), i.e., only the attributes in $\mathcal{U}$ are used. Step 5 identifies the k nearest neighbors, i.e., the k samples in the training set that have the shortest distance to the test sample l based on the distance matrix computed in Step 4. Finally, a label is assigned to each test sample based on the most frequent one among the k nearest neighbors (Step 6).

4. Experiments

The proposed k-NN strategy was applied to 14 datasets of various dimensionality: eight well-known benchmark data sets from the UCI Repository (Australian Credit -AUSTRALIAN-, Wisconsin Breast Cancer -WISCONSIN-, BUPA Liver -LIVER-, German Credit -GERMAN-, Pima Indians Diabetes -DIABETES-, Heart/Statlog -HEART-, IONOSPHERE, and SONAR), and six microarray datasets (Alon's colon cancer data -ALON-, Gravier's breast cancer data -GRAVIER-, Alizadeh's lymphoma data -ALIZADEH-, Pomeroy's central nervous system embryonal tumor data -POMEROY-, West's breast cancer data -WEST-, and Shipp's lymphoma data -SHIPP-). Table 1 presents the number of variables and sample size for each dataset [see 15, for more details on these datasets].

We compared the proposed strategy with the standard k-NN method using $k \in \{1, 3, 5, 9, 15\}$. For the proposed approach, we explored the following values for r (cardinality of $\mathcal{U}$) and Minkowski distance parameter p: $r \in \{1, 3, 5, 10, 15\}$ (UCI datasets), $r \in \{1, 5, 10, 20, 50, 100, 500, 1000\}$ (microarray datasets), and $p \in \{1, 2, \infty\}$. Results with all available variables are also reported for completeness. Model validation was performed using 10-fold cross-validation for all the datasets. Results in terms of Area Under the Curve (AUC) multiplied by 100 are reported.

Tables 2 and 3 compare k-NN and the proposal empirically to assess the influence of the two important aspects of it: the selection of a subset of relevant variables, and the variation of the Minkowski distance (parameter p). Table 2 summarizes the performance for each subset selection method (FS, MI, EC, and CFS), using all available variables (k-NN all). The best configuration for parameters k, r, and p is reported. Table 3 summarizes the performance for each variation of the Minkowski distance (Manhattan, Euclidean, and Chebyshev), for which the best configuration for parameters k, r, and the subset selection method (all variables, FS, MI, EC, or CFS) are reported. The largest AUC is highlighted in bold type for each dataset.

It can be observed in Table 2 that the largest AUC is always achieved with the Fisher Score, Mutual Information, or Eigenvector Centrality. This result demonstrates the importance of using an adequate subset of relevant attributes for computing the distance metric; using all variables or selecting a subset based on redundancy leads to worse results in terms of predictive performance.

In Table 3 the three distance metrics studied in this paper are compared: Manhattan ($p = 1$), Euclidean ($p = 2$), and Chebyshev ($p = \infty$). We conclude with the information on this table that the Manhattan distance has the best overall performance, achieving the maximum AUC in eight of the 14 datasets, while the Euclidean and Chebyshev distances worked best in five and four of the datasets, respectively.

Besides k-NN, we compared the proposed approach with the following binary classification approaches:

Table 1
Number of variables and sample size for all data sets.

Dataset	#features	#examples	Dataset	#features	#examples
LIVER	6	345	SONAR	60	208
DIABETES	8	768	ALON	2,000	62
HEART	13	270	GRAVIER	2,905	168
AUSTRALIAN	14	690	ALIZADEH	4,026	96
GERMAN	24	1000	POMEROY	7,128	60
WISCONSIN	30	569	WEST	7,129	49
IONOSPHERE	34	351	SHIPP	7,129	77

Table 2
Predictive performance summary (AUC×100) for the various subset selection methods.

	k-NN all	*k*-NN FS	*k*-NN MI	*k*-NN EC	*k*-NN CFS
LIVER	66.3	66.3	66.3	**67.2**	66.3
DIABETES	70.4	72.5	**73.5**	72.8	70.4
HEART	71.5	82.1	**82.8**	71.5	72.3
AUSTRALIAN	86.3	**86.9**	**86.9**	86.3	86.3
GERMAN	63.6	**65.4**	63.9	64.2	63.6
WISCONSIN	96.8	96.8	96.8	**97.0**	96.8
IONOSPHERE	89.1	**90.5**	89.1	89.1	89.1
SONAR	**84.0**	**84.0**	**84.0**	**84.0**	**84.0**
ALON	81.3	**90.0**	88.8	**90.0**	81.3
GRAVIER	61.7	**74.1**	65.3	66.2	62.8
ALIZADEH	87.5	**97.1**	95.0	82.1	91.3
POMEROY	60.4	72.9	60.4	**73.3**	65.8
WEST	54.2	**89.2**	88.3	61.7	67.5
SHIPP	84.0	**96.3**	94.0	92.5	84.7

Table 3
Predictive performance summary (AUCx100) for various distance metrics: Manhattan ($p = 1$), Euclidean ($p = 2$), and Chebyshev ($p = \infty$).

	k-NN $p = 1$	*k*-NN $p = 2$	*k*-NN $p = \infty$
LIVER	**67.2**	66.8	65.8
DIABETES	72.8	71.7	**73.5**
HEART	**82.8**	82.4	79.6
AUSTRALIAN	**86.9**	**86.9**	**86.9**
GERMAN	**65.4**	65.1	64.8
WISCONSIN	**97.0**	96.7	96.5
IONOSPHERE	**90.5**	86.7	86.3
SONAR	**84.0**	82.6	83.1
ALON	**90.0**	87.5	**90.0**
GRAVIER	73.3	**74.1**	72.1
ALIZADEH	96.3	**97.1**	90.4
POMEROY	72.9	68.3	**73.3**
WEST	86.7	**89.2**	88.3
SHIPP	96.2	**96.3**	92.2

- *Logistic regression* (Logit): This method constructs a classification function which is linear in terms of the variables via maximum likelihood estimation [9].
- *Naïve Bayes* (NB): This approach uses the Bayes theorem to compute the *a posteriori* probability for each class, under the assumption that all variables are independent of each other. A new sample is assigned to the class with maximum probability [9].
- *Artificial Neural Networks* (ANN): This machine learning technique computes a nonlinear classifier using a network architecture inspired by the functioning of neurons in an animal brain [9]. In this work, we used a single layer ANN with 10 nodes in the hidden layer. Although more complex architectures could have been used, such as deep learning approaches, a single hidden layer neural network usually suffices when dealing with structured datasets, such as the ones studied in this section [9]. Deep learning is particularly useful for tasks in which the preprocessing step requires a complex featurization process, such as in pattern recognition with images or videos [11].
- *Support Vector Machines* (SVM): This method constructs a maximum-margin hyperplane that aims at classifying the tow training patterns accurately. The linear and nonlinear soft-margin SVM were implemented using the LIBSVM toolbox [3].

Table 4
Predictive performance summary (AUC×100) for the various classification methods.

	Prop.	*k*-NN	Logit	NB	ANN	SVM
LIVER	67.2	66.3	67.0	58.3	54.8	**67.5**
DIABETES	73.5	70.4	72.5	71.8	**73.7**	72.2
HEART	82.8	71.5	**84.8**	84.5	80.7	84.1
AUSTRALIAN	**86.9**	86.3	85.8	79.3	86.2	86.2
GERMAN	65.4	63.6	**70.0**	69.4	**70.0**	68.6
WISCONSIN	**97.0**	96.8	95.2	92.7	96.4	96.9
IONOSPHERE	90.5	89.1	83.6	83.2	86.5	**90.9**
SONAR	**84.0**	**84.0**	75.8	69.2	74.6	76.6
ALON	**90.0**	81.3	70.0	83.8	72.5	83.8
GRAVIER	74.1	61.7	60.5	66.9	71.4	**76.0**
ALIZADEH	**97.1**	87.5	59.5	85.8	90.0	94.6
POMEROY	**73.3**	60.4	38.8	65.4	52.9	61.3
WEST	**89.2**	54.2	40.0	50.0	47.5	61.7
SHIPP	**96.3**	84.0	61.8	67.8	85.0	94.2

The results of this comparison are reported in Table 4, in which the best performance in terms of AUCx100 is presented for the various subset selection techniques and distance metrics. The best performance among all methods is highlighted in bold type. The goal of these experiments is to show that *k*-NN is indeed outperformed by the alternative classification techniques when all variables are used due to its inability for dealing with noisy/irrelevant features, but the inclusion of the proposed modifications ("Prop.") makes this approach competitive with the same available information.

In Table 4, it can be observed that no method is able to outperform the others, although our proposal achieved best performance on the high-dimensional datasets. This is mainly because of the use of only a subset of variables, since there are previous studies that confirm that feature selection improves performance for these datasets [see e.g. 15]. Next, we report the training times for all the previous methods in Table 5, including the various ranking strategies discussed in the previous section. All methods were implemented on MATLAB 2014a, on an HP Envy dv6 with 16 GB RAM, 750 GB SSD, a i7-2620M processor with 2.70 GHz, and using Microsoft Windows 8.1 Operating System (64-bits). We used $p = 1$ for the proposed method since the running times for the various p values were quite similar.

It can be concluded from Table 5 that all methods have tractable running times, being all below 10 s. Notice that a high dimensionality causes larger training times, especially for the proposed approach using Eigenvector Centrality or Correlation Score. In contrast, the use of the Fisher Score leads to running times below one second, being actually faster than *k*-NN with all variables on high-dimensional settings. In other words, the computation of

Table 5
Training times (seconds) for the various classification methods.

	Prop. FS	Prop. MI	Prop. EC	Prop. CFS	*k*-NN	Logit	NB	ANN	SVM
LIVER	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.34	0.02
DIABETES	0.02	0.02	0.02	0.01	0.03	0.00	0.00	0.11	0.01
HEART	0.02	0.01	0.02	0.00	0.00	0.01	0.00	0.10	0.01
AUSTRALIAN	0.02	0.02	0.02	0.02	0.02	0.03	0.01	0.32	0.02
GERMAN	0.11	0.18	0.20	0.10	0.04	0.03	0.01	0.31	0.07
WISCONSIN	0.01	0.02	0.03	0.01	0.03	0.24	0.01	0.35	0.01
IONOSPHERE	0.05	0.07	0.10	0.01	0.01	0.17	0.02	0.23	0.01
SONAR	0.03	0.08	0.10	0.01	0.02	0.24	0.03	0.32	0.00
ALON	0.75	0.48	1.06	0.35	0.88	0.15	0.11	0.80	0.05
GRAVIER	0.74	0.86	2.31	0.94	0.48	0.57	0.21	1.27	0.48
ALIZADEH	0.54	0.93	3.26	1.49	0.93	0.33	0.22	1.65	0.23
POMEROY	0.75	1.56	7.28	3.04	1.16	0.21	0.29	2.01	0.22
WEST	0.75	1.54	7.48	3.03	1.28	0.16	0.29	1.54	0.17
SHIPP	0.77	1.58	7.54	3.10	1.21	0.31	0.28	5.48	0.25

Table 6
Holm's post-hoc test for pairwise comparisons. Various subset selection methods.

Method	Mean Rank	Mean AUCx100	*p* value	$\alpha/(k-i)$	Action
k-NN FS	1.86	83.15	-	-	not reject
k-NN MI	2.61	81.08	0.209	0.0500	not reject
k-NN EC	2.68	78.42	0.169	0.0250	not reject
k-NN CFS	3.64	77.30	0.003	0.0167	reject
k-NN all	4.21	75.51	0.001	0.0125	reject

the distance matrix with few attributes is able to compensate the computational effort that requires the ranking step.

Next, a statistical analysis is performed to further exploration of the influence of the various subset selection methods and distance metrics. Based on the information provided in Tables 2 and 3, a ranking is constructed for the subset selection methods and distance metrics, respectively. The average ranking is subsequently computed, and the Holm's test is used to assess whether or not one strategy outperforms the others in terms of AUC. This test was suggested by Demšar for comparing various machine learning methods statistically [7], and performs pairwise comparisons between each strategy and the one with the best performance. This analysis is presented in Tables 6 and 7 for the various subset selection methods (including the use of all available variables) and for the various distance metrics, respectively.

It can be concluded from the results of the experiments presented in Table 6, that our proposal outperforms *k*-NN when Fisher Score, Mutual Information, or Eigenvector Centrality are used for selecting the subset of variables. There are no significant differences among these three approaches. By contrast, the Correlation Score and the use of all variables are significantly worse than FS, MI, and EC (*p*-value below the threshold defined by the Holm's test). In Table 7, it can be seen that no distance metric is able to outperform the others statistically.

For the next set of experiments, the performance is studied for the various values of *r*, the cardinality of $\mathcal{U}$, when varying parameters *k*, *p*, and the subset selection method (Figs. 1, 2, and 3, respectively). The goal of these experiments is to analyze the stability of the proposed method, and to understand whether or not performance is affected by the size of $\mathcal{U}$.

The stability analysis is presented using the AUSTRALIAN and ALON datasets for illustrative purposes. Fig. 1 presents the AUCx100 for an increasing number of *r* and *k*, for the best configuration of *p* and the subset selection method (AUSTRALIAN: $p = 1$ and FS, ALON: $p = \infty$ and FS). It can be observed in this figure that *r* has a strong influence on performance, showing an inverted-U shape where the worst performance is achieved with either just few ($r \leq 3$) or all attributes ($r = n$). This phenomenon is particularly strong on the ALON dataset, for which the gain of using a subset of approximately 100 attributes is noticeable compared with using the 2000 features. Regarding the influence of *k*, it can be observed that predictive results are quite poor for $k = 1$, being all performances relatively similar when $k > 1$.

Fig. 2 presents the AUCx100 for an increasing number of *r* and *p*, for the best configuration of *k* and the subset selection method (AUSTRALIAN: $k = 9$ for $p = 1$, $k = 5$ for $p = 2$, $k = 15$ for $p = \infty$; ALON: $k = 5$ for $p = 1$, $k = 9$ for $p = 2$, and $k = 5$ for $p = \infty$). The three distance metrics show relatively similar results in both datasets and, although the Chebyshev distance has the single best performance for the ALON dataset for $r = 100$, this measure is not able to outperform the other distances for the various *r* values. In terms of the influence of *r*, the inverted-U shape pattern can again be observed.

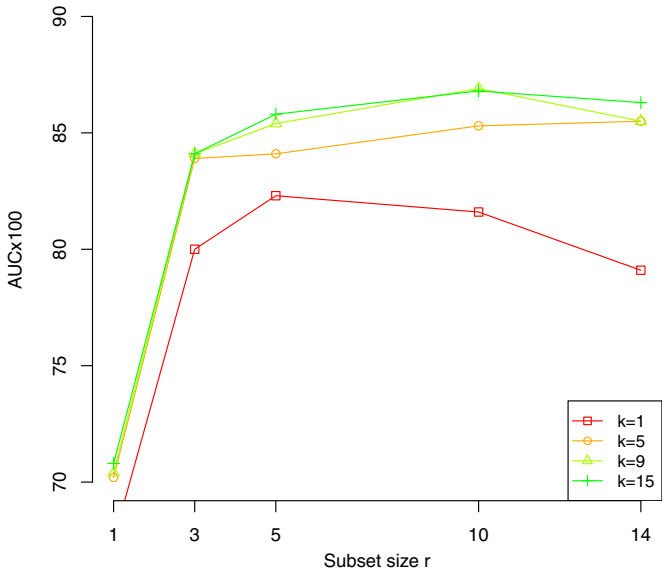
Finally, Fig. 3 illustrates the AUCx100 for an increasing number of parameter *r* and for the various subset selection methods (FS, MI, EC, and CFS), for the best configuration of *k* and the best subset selection method (AUSTRALIAN: $k = 9$ and $p = 1$ for FS, $k = 15$ and $p = \infty$ for MI, $k = 15$ and $p = 1$ for EC, and $k = 15$ and $p = 1$ for CFS). The methods FS, MI, and EC show remarkably better performance than CFS, demonstrating that the latter approach fails at identifying the right subset for the proposed model. Regarding the influence of *r*, the results coincide with those shown in Figs. 1 and 2 in the sense that the best performance is achieved with approximately half of the variables for the AUSTRALIAN dataset, and 100 attributes for the ALON dataset.

The main findings of this experimental section are:

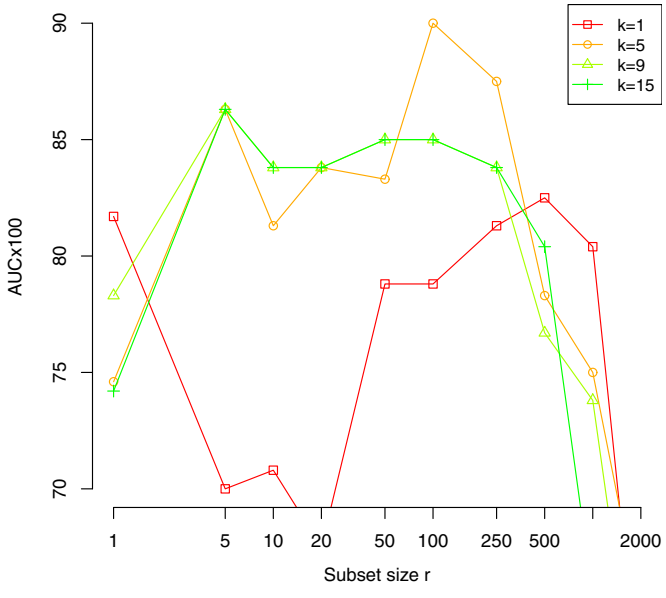
- Among the various distance metrics related to our proposal, the Manhattan norm showed best performance in comparison with the Euclidean and Chebyshev distances. Although these differences are not significant, we recommend using $p = 1$ for low-dimensional datasets, and $p = 2$ for microarray data with a subset of 100–200 variables. All norms are relatively similar in terms of computational time, and therefore we recommend the ones that showed best predictive performance. Although the theory suggests that the Euclidean norm is not suitable in high-dimensional settings, the use of a feature subset for computing the neighborhood leads to robust results with any norm.
- In high-dimensional settings, the definition of a subset of ranked features was crucial for achieving good predictive performance. This confirms the importance of variable selection in datasets such as microarray data. For these experiments,our approach not only outperformed *k*-NN, but also all the alternative binary classification approaches studied.
- Among the various ranking methods related to our proposal, the Fisher Score showed best performance, especially in high-

Table 7
Holm's post-hoc test for pairwise comparisons. Various distance metrics: Manhattan ($p = 1$), Euclidean ($p = 2$), and Chebyshev ($p = \infty$).

Method	Mean Rank	Mean AUCx100	p value	$\alpha/(k-i)$	Action
k-NN $p = 1$	1.61	83.00	-	-	not reject
k-NN $p = 2$	2.00	82.24	0.30	0.0500	not reject
k-NN $p = \infty$	2.39	81.63	0.04	0.0250	not reject

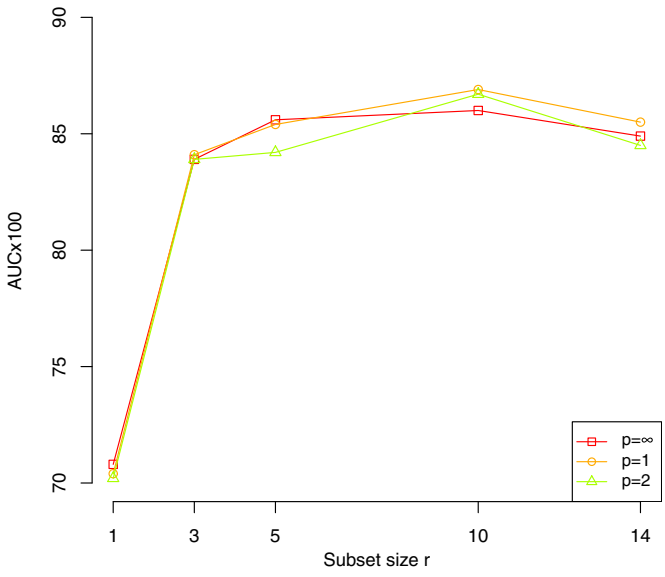


(a) AUSTRALIAN dataset

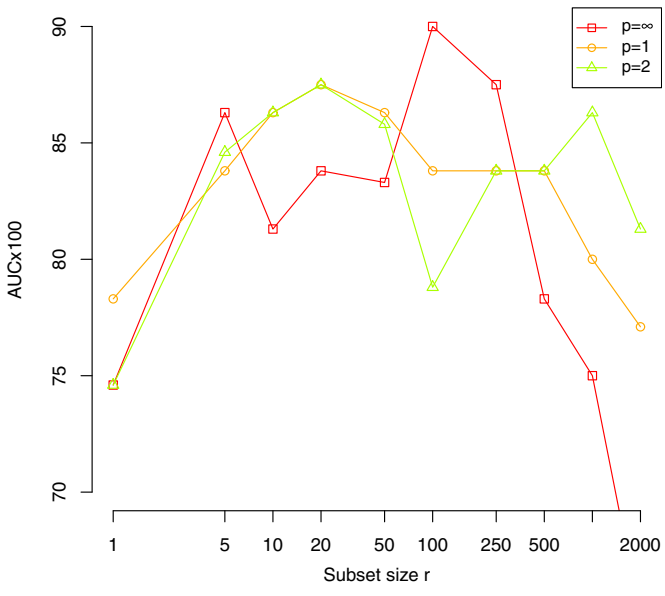


(b) ALON dataset

Fig. 1. Performance (AUC×100) for an increasing number of r and k.



(a) AUSTRALIAN dataset

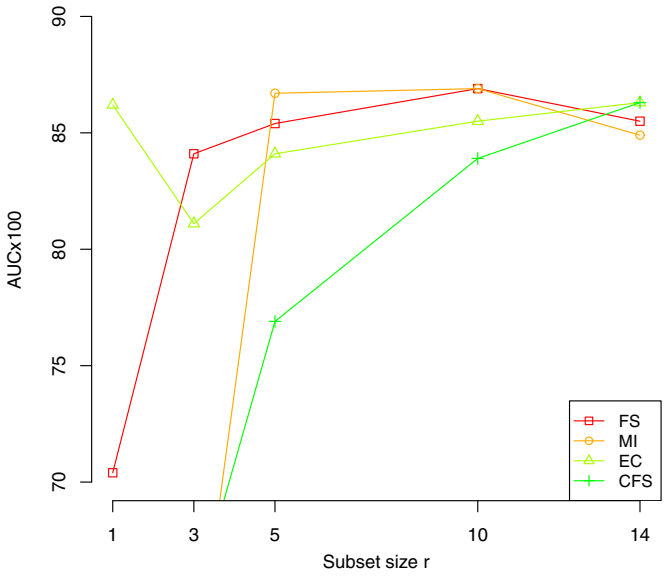


(b) ALON dataset

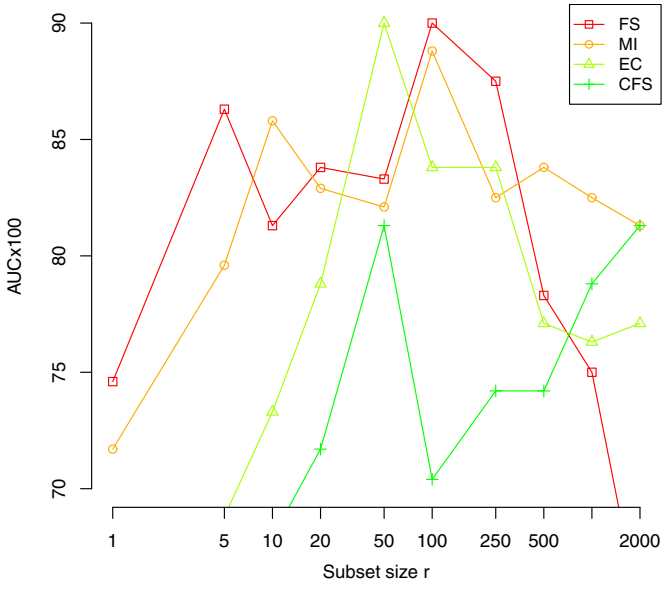
Fig. 2. Performance (AUC×100) for an increasing number of r and p.

dimensional settings. Since this technique is also the one with the fastest running time, we strongly recommend this approach for feature ranking. Notice that Fisher Score assesses only feature relevancy; approaches such as Eigenvector Centrality can be useful in applications that face a high level of redundancy, such as computer vision [2,8].

- Although each dataset shows a different behavior, there are some noticeable results: among the low-dimensional datasets, HEART is the only one in that our method shows significant gains in terms of performance compared with standard k-NN. Interestingly, this dataset is also the smallest in terms of samples, showing that our approach may be useful to avoid overfitting when few examples are available. This can be confirmed



(a) AUSTRALIAN dataset



(b) ALON dataset

Fig. 3. Performance (AUCx100) for an increasing number of r. Various subset selection methods.

with the results obtained on high-dimensional datasets since they also have few samples. Our method leads to at least a 10% increase in terms of AUC for all those datasets.

- Despite the fact we believe that the "no free lunch" theorem holds for our proposal, in the sense that no configuration seems to work best in every dataset [28], our proposal with $p = 1$ and Fisher Score as ranking method achieved robust results, being a good alternative for k-NN classification and machine learning in general. Nevertheless, we strongly suggest trying multiple models and find one that works best for a particular problem. In high-dimensional settings, feature selection is strongly recommended [8].

5. Conclusions

The present study provides a modified k-NN algorithm which computes the distances between samples based only on a subset of the available attributes. Three different variations of the Minkowski distance are also explored, namely the Manhattan, the Euclidean, and the Chebyshev distances. Our proposal aims at solving two major problems found in standard k-NN classification: the first is the course of dimensionality which has a strong influence when defining a neighborhood in high-dimensional settings [16], and the second is that irrelevant features affect predictive performance negatively, since the traditional Euclidean distance measure used to obtain the k nearest neighbors weights each variable equally, regardless of its dependency on the target variable. A novel metric is proposed, and the proof that it satisfies the properties required for being a distance measure is included.

In our experiments, various strategies for defining an adequate subset of variables were explored. In particular, the Fisher score, mutual information, eigenvector centrality, and correlation score were studied in order to discard noisy attributes that affect performance negatively. The proposed method was applied on low- and high-dimensional datasets, outperforming the classical k-NN one when relevancy is considered as a feature selection criterion. Correlation-based selection, in contrast, failed at identifying an adequate feature subset. It was found that the proposed method worked best with $k > 3$ and a subset size of about half of the variables for low-dimensional datasets, and approximately 100 of them for the microarray datasets.

There are several directions for future work. The proposed approach can be extended to other k-NN variations, such as regression or multi-class classification. Furthermore, the Big Data era has opened new challenges for dealing with the three V's: velocity, volume, and variety. The proposed approach not only enhances predictive performance; it also confers efficiency, since it discards irrelevant information quickly, allowing faster computation of the distances between points. There is, however, room for improvement in other aspects of the algorithm, such as instance selection [14] or parallel computing [1] for an efficient definition of the nearest neighbors.

Acknowledgments

This research was partially funded by FONDECYT projects 1160894 and 1160738, and by the Complex Engineering Systems Institute (CONICYT, PIA, FB0816). The authors are grateful to the anonymous reviewers who contributed to improving the quality of the original paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patrec.2018.03.023.

References

[1] G. Beliakov, G. Li, Improving the speed and stability of the k-nearest neighbors method, Pattern Recognit. Lett. 33 (2012) 1296–1301.
[2] S. Biasotti, D. Giorgi, S. Marini, M. Spagnuolo, B. Falcidieno, MRCS 2006: Multimedia Content Representation, Classification and Security, vol. 4105, Springer, pp. 314–321.
[3] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[4] S. Cost, S. Salzberg, A weighted nearest neighbor algorithm for learning with symbolic features, Mach. Learn. 10 (1) (1993) 57–78.
[5] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[6] S. Datta, D. Misra, S. Das, A feature weighted penalty based dissimilarity measure for k-nearest neighbor classification with missing features, Pattern Recognit. Lett. 80 (2016) 231–237.

[7] J. Demšar, Statistical comparisons of classifiers over multiple data set, J. Mach. Learn. Res. 7 (2006) 1–30.
[8] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature Extraction, Foundations and Applications, Springer, Berlin, 2006.
[9] J. Han, Data Mining: Concepts and Techniques, Elsevier, 2011.
[10] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 18 (6) (1996) 607–616.
[11] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[12] H. Lee, S. Hong, I.F. Nizami, E. Kim, An efficient design of a nearest neighbor classifier for various-scale problems, Pattern Recognit. Lett. 31 (2010) 1020–1027.
[13] S. Li, E. Harner, D. Adjeroh, Random knn feature selection - a fast and stable alternative to random forests, BMC Bioinform. 12 (2011) 450.
[14] Z. Li, G. Ding, R. Li, S. Qin, A new extracting algorithm of k nearest neighbors searching for point clouds, Pattern Recognit. Lett. 49 (2014) 162–170.
[15] J. López, S. Maldonado, Group-penalized feature selection and robust twin svm classification via second-order cone programming, Neurocomputing 235 (2017) 112–121.
[16] R. Marimont, M. Shapiro, Nearest neighbour searches and the curse of dimensionality, IMA J. Appl. Math. 24 (1) (1979) 59–70.
[17] P. Mulak, N. Talhar, Analysis of distance measures using k-nearest neighbor algorithm on kdd dataset, Int. J. Sci. Res. 4 (7) (2015) 2101–2104.
[18] A. Navot, L. Shpigelman, N. Tishby, E. Vaadia, Nearest neighbor based feature selection for regression and its application to neural activity, Advances in neural information processing systems (NIPS), 18, 2005.
[19] H.-L. Ooi, S.-C. Ng, E. Lim, Ano detection with k-nearest neighbor using minkowski distance, Int. J. Signal Process. Syst. 1 (2) (2013) 208–211.
[20] A.K. Pal, P.K. Mondal, A.K. Ghosh, High dimensional nearest neighbor classification based on mean absolute differences of inter-point distances, Pattern Recognit. Lett. 74 (2016) 1–8.
[21] C.-H. Park, S.-B. Kim, Sequential random k-nearest neighbor feature selection for high-dimensional data, Expert Syst. Appl. 42 (2015) 2336–2342.
[22] G. Roffo, S. Melzi, New frontiers in mining complex patterns, Fifth International workshop, nfMCP2016. Lecture Notes in Computer Science, Springer, pp. 19–35.
[23] M. Tahir, A. Bouridane, F. Kurugollu, Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier, Pattern Recognit. Lett. 28 (4) (2007) 438–446.
[24] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 4th, Cambridge: Academic Press, 2008.
[25] J. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (2014) 175–186.
[26] J. Wang, P. Neskovic, L.N. Cooper, Improving nearest neighbor rule with a simple adaptive distance measure, Pattern Recognit. Lett. 28 (2007) 207–213.
[27] K. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2009) 207–244.
[28] D. Wolpert, The lack of a priori distinctions between learning algorithms, Neural Comput. 8 (7) (1996) 1341–1390.