A Comparison between Optimum-Path Forest and k-Nearest Neighbors Classifiers

Roberto Souza
DCA
UNICAMP
Campinas, Brazil
Email: roberto.medeiros.souza@gmail.com

Roberto Lotufo, Letícia Rittner
DCA
UNICAMP
Campinas, Brazil
Email: lotufo@unicamp.br, lrittner@gmail.com

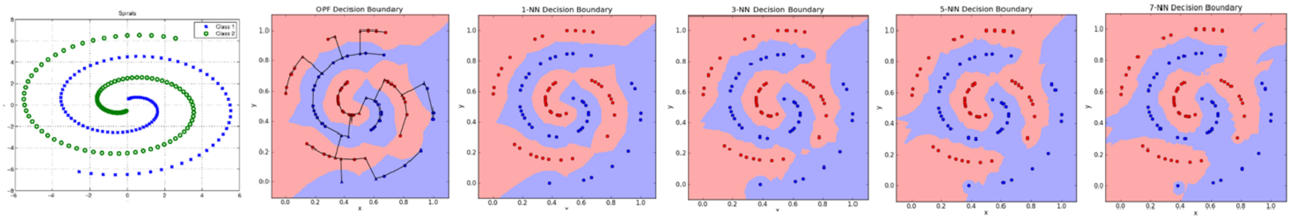


Fig. 1. From left to right: synthetic bidimensional dataset, OPF decision boundary, 1-NN decision boundary, 3-NN decision boundary, 5-NN decision boundary, 7-NN decision boundary.

***Abstract*—This paper presents a comparison between the k-Nearest Neighbors, with an especial focus on the 1-Nearest Neighbor, and the Optimum-Path Forest supervised classifiers. The first was developed in the 1960s, while the second was recently proposed in the 2000s. Although, they were developed around 40 years apart, we can find many similarities between them, especially between 1-Nearest Neighbor and Optimum-Path Forest. This work shows that the Optimum-Path Forest classifier is equivalent to the 1-Nearest Neighbor classifier when all training samples are used as prototypes. The decision boundaries generated by the classifiers are analysed and also some simulations results for both algorithms are presented to compare their performances in real and synthetic data.**

***Keywords*-k-Nearest Neighbors; Optimum-Path Forest; decision boundaries; classification;**

I. Introduction

Classification problems are present in many fields of engineering, for example, a medic may want to be able to tell from patterns present in a brain image, if a patient has brain cancer or not. Other example is a market research, where an enterprise tries to segment the market aiming to increase their sells. The first is an example of supervised classification, because the medic can assign the patient to only one of two possible classes: “He has brain cancer” or “he doesn’t have brain cancer”. The second example is an unsupervised classification problem, since the market research can group the market in any arbitrary number of segments, which are not known a priori.

In this context, the k-Nearest Neighbors (k-NN) and the Optimum-Path Forest (OPF) supervised classifiers are simple non-parametric classification methods presented in the literature, that often provide competitive results when compared with other classifiers [1], [2]. The k-NN classifier was proposed in the 1960s and it is still very used to this very day. It can be shown that for a large number of samples in a problem with M classes the 1-NN is bounded below by the Bayes Error Rate (BER) and above by twice the BER, which is the reason why it is said that half the classification information in an infinite sample set is contained in the nearest neighbor [3] . The OPF classifier was developed in the 2000s and it has shown good results in many classification problems, but it lacks a mathematical treatment to explain why it works well in many problems and what are its limitations.

Contributions: The goal of this paper is to compare both theoretically and with simulations the k-NN and the OPF supervised classifiers and show that they share many common aspects (OPF is equivalent to 1-NN when all training samples are used as prototypes). This comparison tries to give some intuition for the reason why such good results are achieved by OPF and also propose possible research lines that may be used to improve both methods.

Paper Organization: Section II formulates the supervised classification problem. Section III and IV present the operation of k-NN and OPF classifiers, respectively. Section V presents some simulations results for both methods using synthetic and real databases, for the bidimensional cases the decision boundaries are also analysed. In Section VI the conclusions are presented.

II. Supervised Classification Problem Formulation

Let’s suppose that we have a classification problem in which there are M possible classes and there are N

i.i.d. (independent and identically distributed) samples $Z = \{(X_1, \theta(X_1)), (X_2, \theta(X_2)), ...(X_N, \theta(X_N))\}$, where X_i is a vector in the feature space and θ corresponds to the class that sample belongs to, i.e. $\theta(X_i) \in \{\omega_1, \omega_2, ..., \omega_M\}$. The supervised classification problem consists in using that prior knowledge to classify new samples Xs to one of the M possible classes in a manner to minimize the classification error, which is given by the following expression:

$$\begin{aligned} p(error) &= \int_{-\infty}^{+\infty} p(error, X) dX \\ &= \int_{-\infty}^{+\infty} p(error \mid X) p(X) dX. \end{aligned} \quad (1)$$

Bayesian decision theory is able to find an optimum solution to (1). It starts from the principle that $p(\omega_i)$ and $p(X \mid \omega_i)$ are known distributions. From Bayes' Theorem, $p(\omega_i \mid X)$ can be written as:

$$p(\omega_i \mid X) = \frac{p(X|\omega_i) \times p(\omega_i)}{p(X)}, \quad (2)$$

where:

$$p(X) = \sum_{i=1}^{M} p(X \mid \omega_i) \times p(\omega_i). \quad (3)$$

Bayesian decision theory arrives to the conclusion that the optimal decision rule, known as Bayes' Decision Rule is given by:

$$p(error \mid X) = 1 - max[p(\omega_1 \mid X), ..., p(\omega_M \mid X)]. \quad (4)$$

Replacing (4) in (1), we have what is known as the Bayes' Error Rate (BER), which is a theoretical lower bound to the minimum error that can be achieved in a supervised classification problem. This BER is often used as a guideline to determine if a classifier is good or not.

III. k-Nearest Neighbors Classifier

This section will first present the 1-NN, which is simpler to understand, and then it will generalize to the k-NN case.

A. 1-NN

The 1-Nearest Neighbor is a non-parametric, sub-optimum classifier [4], and it has an operational heuristic very simple to implement and understand. It assigns each new sample X to the class ω_i of its nearest labeled sample X_i. There are many metrics that can be used to calculate distance, like the Manhattan distance and the Minkowski distance, but the most commonly used is the Euclidean distance [4]. The 1-NN classifier procedure is illustrated in Fig. 2. There are four labeled samples, X_1, X_2 represented by blue squares belonging to Class 1, and X_3, X_4 represented by red circles belonging to Class 2. The green triangle represents the test sample to be classified. Its distance to each labeled sample is calculated, in this example using the Euclidean distance, and then this sample is assigned to the class of its closest labeled sample. The test sample was assigned to Class 2, because it was closer to X_4. It is important to point out that the 1-NN classifier does not update its initial set of labeled samples, i.e. each new sample that arrives is classified based solely on the initial set of labeled samples. The 1-NN classifier can be formulated in terms of mathematical equations. The label $\theta(X)$ of each new sample X is given by the following equation:

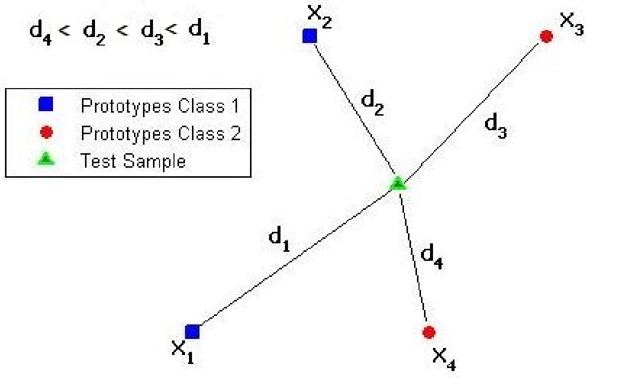


Fig. 2. Illustration of the 1-NN operation. There are 4 training samples, two blue squares (X_2, X_4) belonging to Class 1 and two red circles (X_1, X_3) belonging to Class 2. The test sample represented by the green triangle is assigned to Class 2, because it is closer to X_4.

$$\theta(X) = \theta(X_{NN}), \quad (5)$$

where X_{NN} is given by:

$$X_{NN} = arg \min_{\forall X_i \in Z} \{d_X(X_i)\}, \quad (6)$$

and $d_X(X_i)$ is the distance between X and X_i in the chosen metric.

It was shown in [3] that as the number of labeled samples N tends to infinity in a M-class classification problem, the 1-Nearest Neighbor Error Rate (1NNER) is bounded by the following expression:

$$BER \leq 1NNER \leq BER \times (2 - \frac{M}{M-1} \times BER). \quad (7)$$

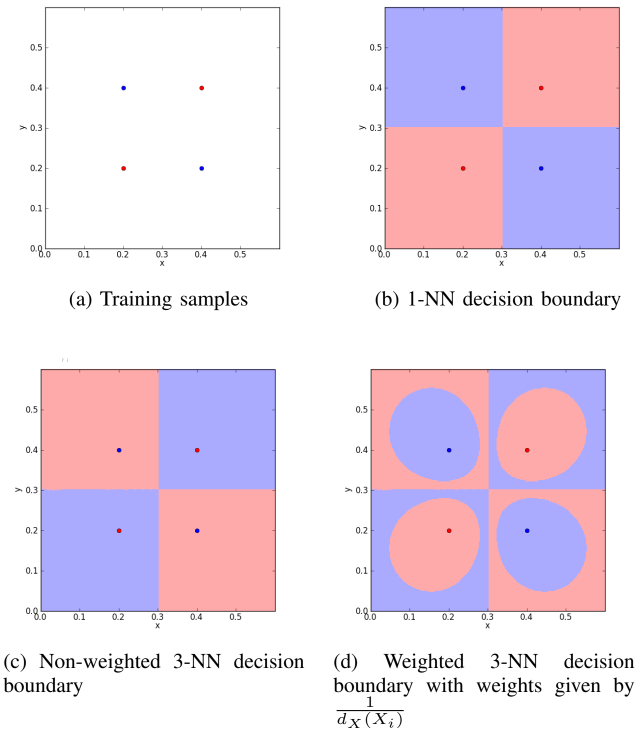
The lower bound of (7) is the BER and can only be achieved with the complete knowledge of the problem statistics, i.e. $p(X)$ and $p(X \mid \omega_i)$, and the upper bound is in its worst case two times the BER, which is achieved when the BER tends to zero [3]. This is an very interesting result, since the worst 1NNER can only be achieved when the BER is made arbitrarily small and the double of an arbitrarily small number is often acceptable from a practical point of view.

The 1-NN classifier has some weaknesses. The first is that it is sensible to noise and outliers. The second is that although it is nice to determine lower and upper bounds to the 1NNER, this result is only valid on an infinite labeled samples space. The convergence of the method on a finite space can be arbitrarily slow and the 1NNER may not even decrease monotonically with the increase of N [4]. It is also important to point out that the computational complexity of the method increases with N, since you have to compute and find the minimum distance between all possible distances $d_X(X_i)$ for each test sample. There are in the literature some alternative methods to cope with these weaknesses cited [5], [6], [7].

B. k-NN

k-NN is a natural extension of the 1-NN classifier. k-NN classifies X by assigning it to the label most frequently present in the k nearest neighbors. k-NN takes into account k neighbors, so it is less sensible to noise and outliers than 1-NN. It can be shown that for an infinite number of samples, N, as k tends to infinity the k-NN Error Rate (kNNER) tends to the BER [4]. Although it seems that k-NN for $k > 1$ is a better classifier than 1-NN, this may not always be true, since the 1NNER and the kNNER bounds were developed based on the hypothesis of an infinite number of samples.

An anomalous example comparing 1-NN and 3-NN occurs when there are 4 equidistant training samples from 2 different classes, as shown in Fig. 3(a). The decision boundaries resulting from 1-NN and 3-NN, Fig. 3(b) and Fig. 3(c), respectively, are exactly the opposite of each other. Clearly the 3-NN boundary does not seem to be a good decision boundary, this occurred because the number of training samples used was too small. To improve k-NN results, the nearest neighbors votes are multiplied by weights. Although this technique makes k-NN more robust for the cases where the number of test samples are not so high, it also turns k-NN into a parametric classifier, since we have to appropriately choose the values of the weights. A more reasonable decision boundary for 3-NN in this anomalous example is shown in Fig. 3(d), using weights inversely proportional to their distances to the test sample, i.e. $\frac{1}{d_X(X_i)}$.



(a) Training samples

(b) 1-NN decision boundary

(c) Non-weighted 3-NN decision boundary

(d) Weighted 3-NN decision boundary with weights given by $\frac{1}{d_X(X_i)}$

Fig. 3. Anomalous example comparing 1-NN and 3-NN.

IV. Optimum-Path Forest Classifier

The Optimum-Path Forest is a graph based classifier that was devleoped as a generalization of the Image Foresting Transform (IFT) [8]. OPF is simple, multi-class, parameter independent, does not make any prior assumption about the shapes of classes and can handle some degree of overlapping [1]. It was developed in the years 2000s and it has shown good results in many classification problems [1], [2], [9]. Classification through the OPF consists of two steps: fit and predict. In the fit step, OPF chooses what it considers to be the most meaningful training samples to become prototypes. The predict step consists in assigning to the test samples the label of the prototype that offers the lower path cost, which is given by a cost function, f_{cost}, defined a priori.

A. OPF Fit

In the fit step a complete undirected graph $A = (Z \times Z)$ is built. In this graph, the training samples X_i represent the nodes and the edges weights, d's, are the distances between training samples calculated using an appropriate metric. The next step is to find the Minimun Spanning Tree (MST) of A using one of the many algorithms available, such as Kruskal's and Prim's algorithm. The K nodes connected in the MST that belong to different classes ω_i are selected and together they form what is called the prototypes set, Z_p. The $L = N - K$ non-prototypes nodes form the non-prototypes set, Z_{np}. These sets will be represented as follows:

$$Z_p = \{(X_{p1}, \theta(X_{p1})), ..., (X_{pK}, \theta(X_{pK}))\}, \tag{8}$$

$$Z_{np} = \{(X_{np1}, \theta(X_{np1})), ..., (X_{npL}, \theta(X_{npL}))\}, \tag{9}$$

and $Z_p \cup Z_{np} = Z$.

A path, $\pi_{s,t} = \langle s, a_i, a_{i+1}, ..., t \rangle$, between nodes s and t is defined as a sequence of nodes, such that from each of its nodes there is an edge to the next node and nodes s and t are the extremes of the path. A path $\pi_{t,t} = \langle t \rangle$ is said to be a trivial path and its path cost is 0. The next step in the fitting process is to calculate the path cost of every training sample to every prototype and assign each training sample to a tree rooted in the prototype that offered the minimum cost. Suppose the path cost function is $C(X)$, then the minimum cost of a training sample X_l is:

$$C(X_l) = \min_{\forall X_{pi} \in Z_p} \{f_{cost}(\pi_{X_l, X_{pi}})\}. \tag{10}$$

In case of ties the training sample is assigned to the tree with the lower number of edges in the path. OPF cost function most commonly used is given by:

$$f_{cost}(\pi_{s,t}) = \max_{\forall j \in \pi_{s,t}} \{d_j\}. \tag{11}$$

Replacing (11) in (10), the result is:

$$C(X_l) = \min_{\forall X_{pi} \in Z_p} \{ \max_{\forall j \in \pi_{X_l, X_{pi}}} \{d_j\}\}, \tag{12}$$

which is the usual OPF classification problem formulation and can be solved using Maximum Dijkstra's algorithm adapted for the multi-source case [8], where the sources are the prototypes.

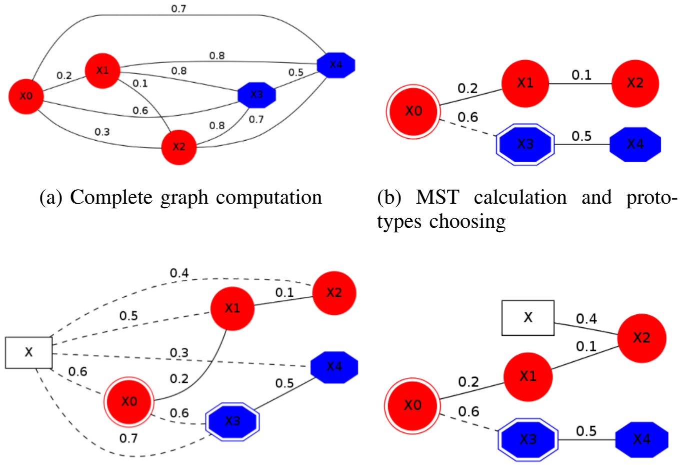
B. OPF Predict

The predict phase of OPF's algorithm consists in assigning a label to every new sample X. This is done by assigning the label of the root prototype of the tree this sample would belong if it was added to the graph A, and can be formalized by the following equations:

$$X_{pi} = arg \min_{\forall X_{pi} \in Z_p} \{f_{cost}(\pi_{X, X_{pi}})\}, \tag{13}$$

$$\theta(X) = \theta(X_{pi}). \tag{14}$$

OPF operation is illustrated in Fig. 4. There are five training samples, X_0, X_1, X_2 from class "circle" and X_3, X_4 from class "hexagon", represented as a complete undirected graph, Fig. 4(a). The MST of the graph is calculated and X_0 and X_3 are chosen as prototypes, Fig. 4(b). These steps correspond to the fitting phase. A test sample X arrives and its distance is calculated in relation to every sample in the train set, Fig. 4(c). Then X is assigned to the tree rooted in X_0 since it offered the lower cost, therefore X was classified as class "circle", Fig. 4(d). The last 2 steps corresponds to the prediction phase. For this particular example shown, it is important to point out that 1-NN would classify X as being from class "hexagon", since X_4 is the nearest neighbor to X.



(a) Complete graph computation

(b) MST calculation and prototypes choosing

(c) New sample X arrives

(d) New sample X is classified as class "circle"

Fig. 4. Illustration of OPF operation. (a)-(b) OPF fit. (c)-(d) OPF predict.

OPF is also sensible to noise and outliers, since the prototypes choosing based on the MST will choose noisy samples or outliers to become prototypes and these samples have great influence on OPF's classification decision.

C. Equivalence between 1-NN Classifier and OPF Classifier

OPF's usual formulation using (11) is equivalent to 1-NN when all training samples are used as prototypes. This is easily shown, since $Z = Z_p$ and all test samples are directly connected to the prototypes, this implies that $f_{cost}(\pi_{X, X_{pi}}) = d_X(X_{pi}) = d_X(X_i)$. Replacing this result in (13), leads to:

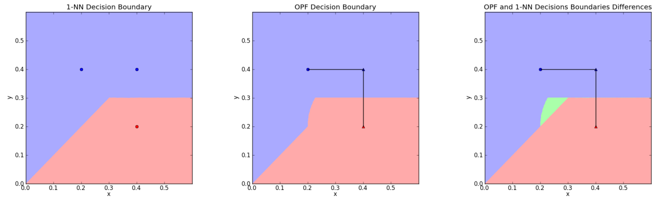
$$X_{pi} = X_i = arg \min_{\forall X_i \in Z} \{d_X(X_i)\}, \tag{15}$$

which is exactly the 1-NN mathematical formulation.

D. OPF and 1-NN Decision Boundaries

The 1-NN decision boundary is defined by the Voronoi diagram, which divides the feature space in clusters where the distances of all points in a given cluster defined by a labeled sample are not greater than their distance to the other labeled samples. A simple example of a 1-NN decision boundary, where two labeled samples are from class blue and a third sample is from class red, is shown in Fig. 5(a). This decision boundary results from the composition of two lines, the horizontal line displays the interaction between the red sample and the blue sample right above it. The second line results from the interaction between the red sample and the blue sample on the top left of the image.

OPF's decision boundary in its usual formulation divides the feature space in clusters defined by trees where the distances of all points in a given cluster to the prototype sample of the tree are not greater than the cost to the other prototypes samples of the other trees. OPF's decision boundary for the same simple example used before is shown in Fig. 5(b). In this image the black lines joining the samples represent the MST and the triangular samples represent the prototypes. Notice that OPF and 1-NN have very similar decision boundaries, differing only on a small portion of the decision space as illustrated in Fig. 5(c).



(a) 1-NN decision boundary

(b) OPF decision boundary

(c) OPF and 1-NN differences shown in green

Fig. 5. Simple comparison between 1-NN and OPF decision boundaries.

V. Simulations

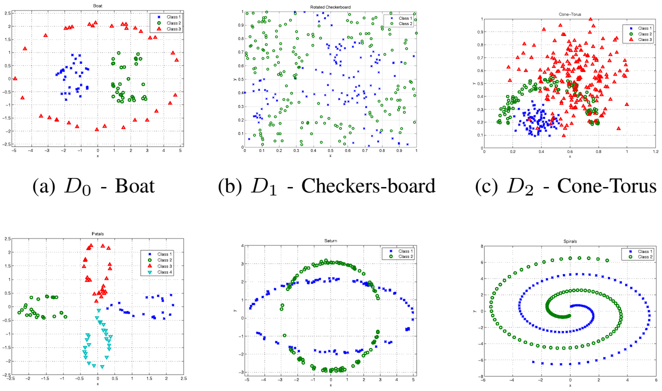
In this section k-NN and OPF supervised classifiers are compared through simulations. The k-NN was simulated using $k = 1$, 3, 5 and 7, and the weights were set as $\frac{1}{d_X(X_i)}$ in all tests. Experiments were made using synthetic data and real data. In all tests the metric used was the Euclidean distance. Table I summarizes the information concerning the datasets.

TABLE I
DESCRIPTION OF THE DATASETS.

Dataset Code	Dataset Name	Samples	Features	Classes
D_0	Boat	100	2	3
$D1$	Checkerboard	300	2	2
$D2$	Cone-Torus	400	2	3
$D3$	Petals	100	2	4
$D4$	Saturn	200	2	2
$D5$	Spirals	200	2	2
$D6$	Digits	1797	64	10
$D7$	Iris	150	4	3
$D8$	WBC	683	9	2

Datasets D_0 to D_5 are constituted of synthetic data and they are available on professor Kuncheva, from Bangor University, online repository[1]. The geometry of many of these synthetic datasets are highly non-linear and with some superposition like in sets D_1, D_2 and D_4. The bidimensional datasets are depicted in Fig. 6. Datasets D_6, D_7 and D_8 correspond to real data and are available on the UC Irvine Machine Learning Repository[2].

The datasets were normalized to perform the simulations. The experiments were done varying the training sets from 20% to 50% of the whole dataset with steps of 5%. For each training percentual, it were performed 50 simulations with each one of the classifiers, and at the end the mean accuracies and the mean kappa coefficients were calculated.



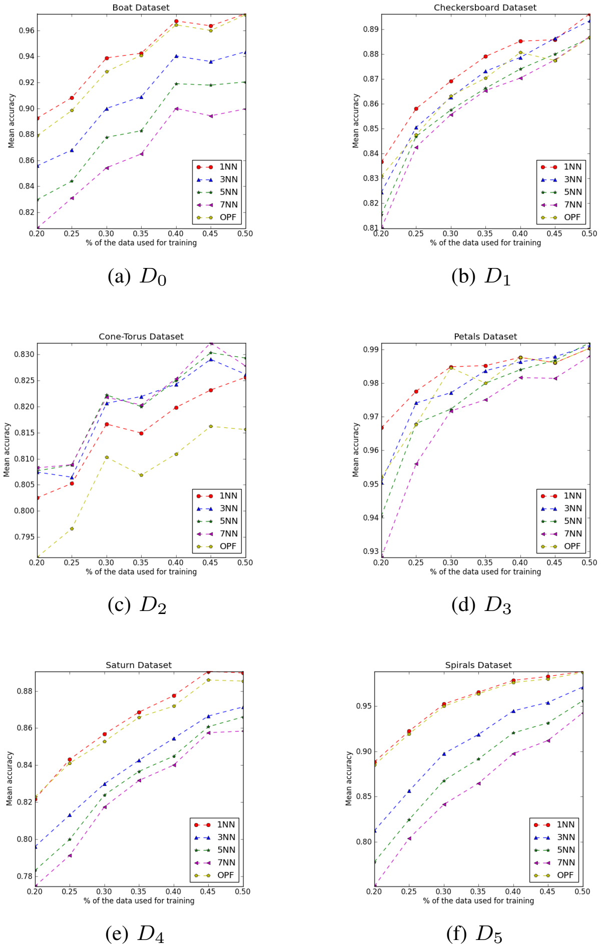
(a) D_0 - Boat (b) D_1 - Checkers-board (c) D_2 - Cone-Torus

(d) D_3 - Petals (e) D_4 - Saturn (f) D_5 - Spirals

Fig. 6. Synthetics Datasets $D_0 - D_5$.

A. Synthetic Datasets Results

The mean accuracy and the mean kappa plots found for each synthetic dataset are summarized in Fig. 7 and Fig. 8, respectively. In most simulations, we can see that OPF and 1-NN accuracy and kappa curves are very similar with 1-NN results being slightly better than OPF results. The only dataset where their results diverge a little more is dataset D_3, where their kappa and accuracy curves start 0.015 apart and in the transition from 30% to 35% OPF accuracy and kappa curves have a little more accentuated drop, but then they start growing again. 3-NN, 5-NN and 7-NN achieve worst results than 1-NN and OPF in datasets D_0, D_4 and D_5 and a slightly better result with dataset D_2. For the other datasets, their results become very similar to 1-NN and OPF, when the percentual of data used for training is increased.



(a) D_0 (b) D_1

(c) D_2 (d) D_3

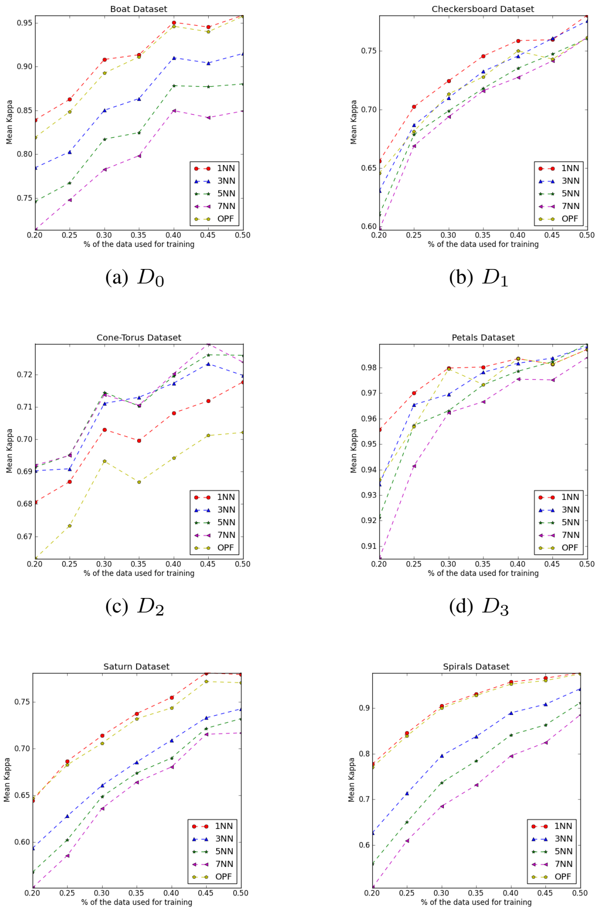
(e) D_4 (f) D_5

Fig. 7. Mean accuracies found for the synthetic datasets.

Fig. 9 through Fig. 14 display the decison boundaries found by the classifiers for one of the simulations using 40% of the dataset for training. In OPF's decision boundaries the black lines joining the samples represent the MST and the triangular samples represent the prototypes. The green regions represent OPF and 1-NN decision differences. The images indicate that 1-NN and OPF produce very similar decision boundaries with OPF's boundaries being smoother due to its cost function. The similarities between OPF and 1-NN decision boundaries explain why they achieve very similar accuracy results. 3-NN, 5-NN and 7-NN also produce smoother decision boundaries than 1-NN, but their shapes are very different from each other and in many cases they present a few isolated decision regions.

[1] http://pages.bangor.ac.uk/~mas00a/activities/artificial_data.htm

[2] http://archive.ics.uci.edu/ml/datasets.html

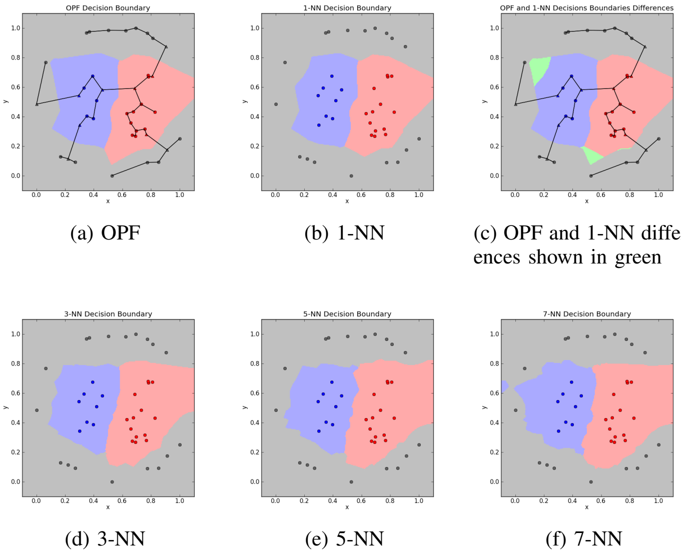


(a) D_0 (b) D_1

(c) D_2 (d) D_3

(e) D_4 (f) D_5

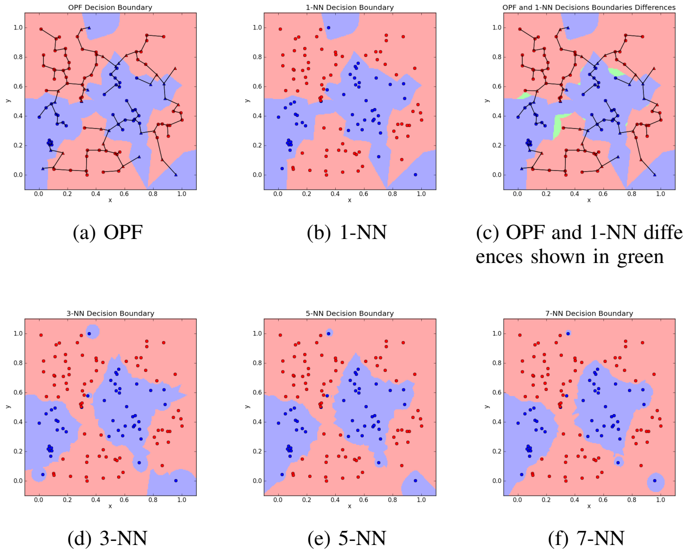
Fig. 8. Mean kappa coefficients found for the synthetic datasets.



(a) OPF (b) 1-NN (c) OPF and 1-NN differences shown in green

(d) 3-NN (e) 5-NN (f) 7-NN

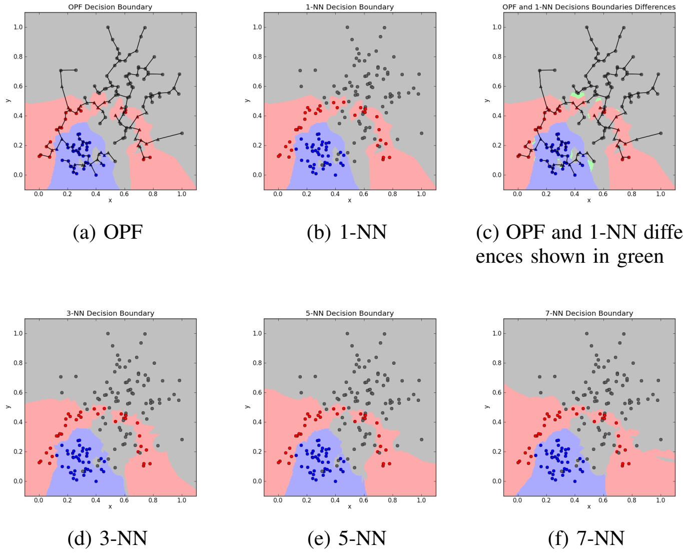
Fig. 9. Boundaries found for dataset D_0.



(a) OPF (b) 1-NN (c) OPF and 1-NN differences shown in green

(d) 3-NN (e) 5-NN (f) 7-NN

Fig. 10. Boundaries found for dataset D_1.



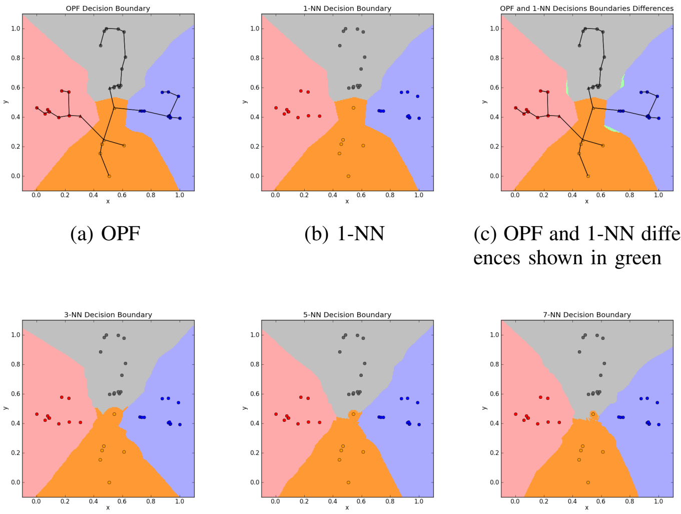
(a) OPF (b) 1-NN (c) OPF and 1-NN differences shown in green

(d) 3-NN (e) 5-NN (f) 7-NN

Fig. 11. Boundaries found for dataset D_2.

B. Real Datasets Results

The mean accuracy and the mean kappa plots found for each real dataset are summarized in Fig. 15 and Fig. 16, respectively. In all simulations 3-NN, 5-NN and 7-NN obtained slightly better results than 1-NN and OPF. Proabably because the larger the k from the k-NN classifier, less sensible to noise and outliers it becomes. OPF and 1-NN still obtain similar results but the behavior of their mean accuracy and mean kappa curves differ in a few transitions, such as in the transistion from 35% to 40% with dataset D_6, where the 1-NN mean accuracy increases while OPF's decreases.



(a) OPF

(b) 1-NN

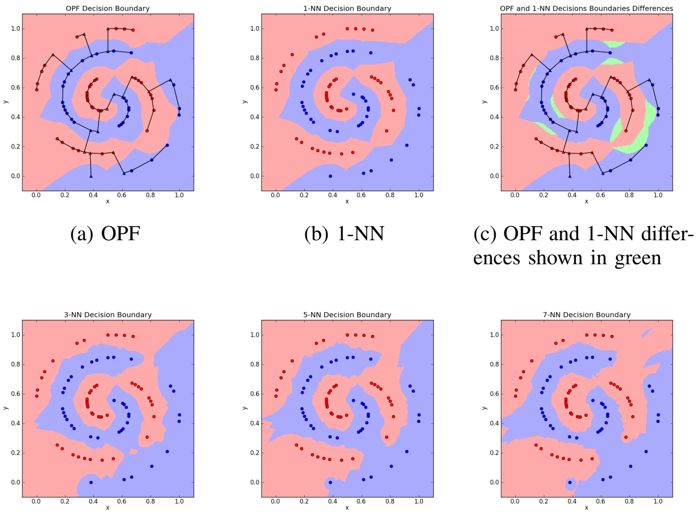
(c) OPF and 1-NN differences shown in green

(d) 3-NN

(e) 5-NN

(f) 7-NN

Fig. 12. Boundaries found for dataset D_3.



(a) OPF

(b) 1-NN

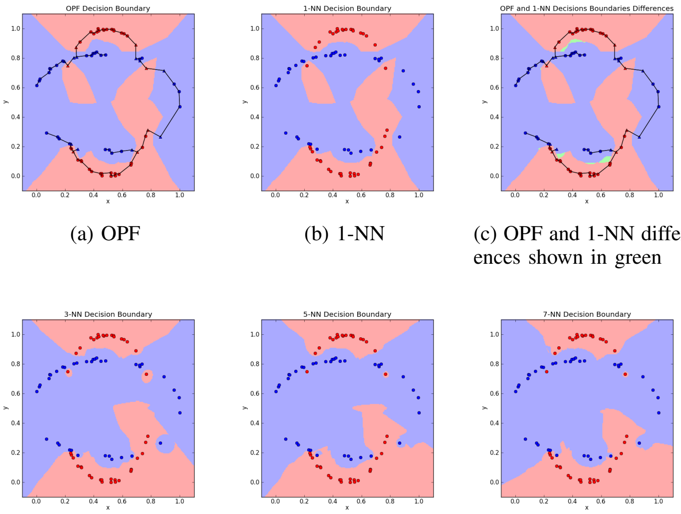
(c) OPF and 1-NN differences shown in green

(d) 3-NN

(e) 5-NN

(f) 7-NN

Fig. 14. Boundaries found for dataset D_5.



(a) OPF

(b) 1-NN

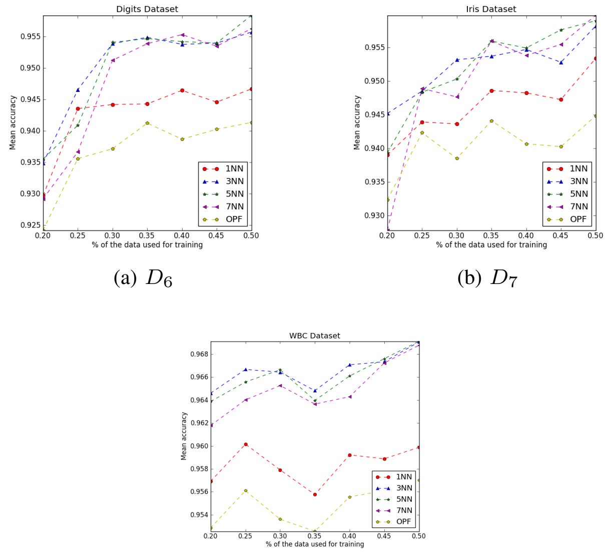
(c) OPF and 1-NN differences shown in green

(d) 3-NN

(e) 5-NN

(f) 7-NN

Fig. 13. Boundaries found for dataset D_4.



(a) D_6

(b) D_7

(c) D_8

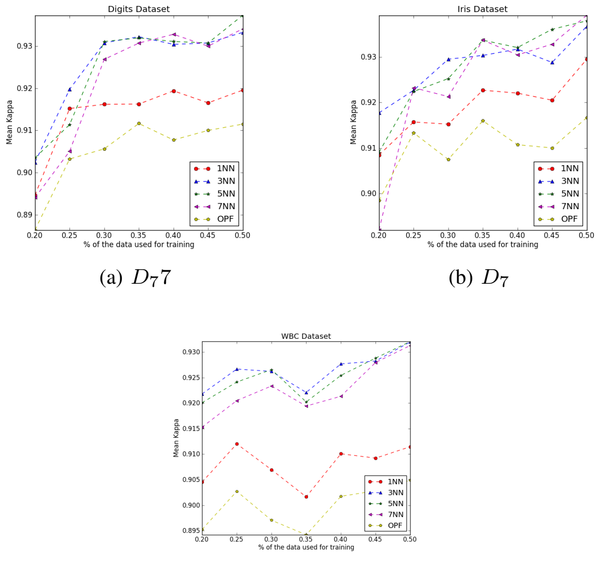
Fig. 15. Mean accuracies found for the real datasets.

VI. Conclusions

This paper presented a brief theoretical description of k-NN and OPF classifiers and a comparison through simulations. During the theoretical description it was shown that 1-NN and OPF are equivalent to each other when all test samples are used as prototypes. Also, the simulation results and the decision boundaries analysis showed a similar behavior between 1-NN and OPF, buth with OPF boundaries being smoother. The simulation results also showed that although k-NN, for $k > 1$, is theoretically a better classifier than 1-NN, this may not be true if the number of training samples is not large enough. The results also comproved that in presence of noise k-NN with $k > 1$ is less sensible than 1-NN and OPF.

The results obtained are not enough to achieve general performance conclusions, but it indicates that OPF and 1-NN are similar classifiers and when working with sparse training sets they usually achieve better results than k-NN for $k > 1$.

OPF classifier introduced the notion of optimum path trees rooted in prototypes and it has a structure that may support improvements both in time and accuracy performance of the classifier. Also, the method based on the MST for finding meaningful labeled samples to become OPF prototypes may be an alternative method for reducing samples in Condensed Nearest Neighbors (CNN) methods. Another possible investigation is implementing the k-OPF, where, instead of using only the distance to the nearest prototype to perform a classification decision, the distances to the k nearest prototypes are used,



(a) D_77

(b) D_7

(c) D_8

Fig. 16. Mean kappa coefficients found for the real datasets.

and analyze if k-OPF and k-NN with the same values of k present similar behaviors such as 1-NN and OPF.

REFERENCES

[1] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *Int. J. Imaging Syst. Technol.*, vol. 19, no. 2, pp. 120–131, Jun. 2009.

[2] J. Papa, A. Spadotto, A. Falcao, and J. Pereira, "Optimum path forest classifier applied to laryngeal pathology detection," in *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, june 2008, pp. 249 –252.

[3] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21 –27, january 1967.

[4] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley, 2001.

[5] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 14, pp. 515–516, 1968.

[6] F. Angiulli, "Fast condensed nearest neighbor rule," in *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, ser. ACM International Conference Proceeding Series, L. D. Raedt and S. Wrobel, Eds., vol. 119. ACM, 2005, pp. 25–32.

[7] C. Kier and T. Aach, "Predicting the benefit of sample size extension in multiclass k-nn classification," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, 0-0 2006, pp. 332 –335.

[8] A. X. Falcão, J. Stolfi, and R. A. Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, Jan 2004.

[9] J. P. Papa, A. X. Falcao, G. M. de Freitas, and A. M. H. de Avila, "Robust Pruning of Training Patterns for Optimum-Path Forest Classification Applied to Satellite-Based Rainfall Occurrence Estimation," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, pp. 396–400, 2010.