2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

PowerHC: non linear normalization of distances for advanced nearest neighbor classification

Manuele Bicego
Università degli Studi di Verona
Dipartimento di Informatica
Strada le Grazie 15, Verona 37134, Italy

Mauricio Orozco-Alzate
Universidad Nacional de Colombia – Sede Manizales
Departamento de Informática y Computación
km 7 vía al Magdalena, Manizales 170003, Colombia

***Abstract*—In this paper we investigate the exploitation of non linear scaling of distances for advanced nearest neighbor classification. Starting from the recently found relation between the *Hypersphere Classifier* (HC) [1] and the *Adaptive Nearest Neighbor rule* (ANN) [2], here we propose *PowerHC*, an improved version of HC in which distances are normalized using a non linear mapping; non linear scaling of data, whose usefulness for feature spaces has been already assessed, has been hardly investigated for distances. A thorough experimental evaluation, involving 24 datasets and a challenging real world scenario of seismic signal classification, confirms the suitability of the proposed approach.**

I. Introduction

In very recent years, there is a scientific trend which is growing in importance in Pattern Recognition/Machine Learning/Artificial Intelligence, called in different ways, such as *explainable* artificial intelligence [3], [4], *interpretable* or *understandable* machine learning [5], [6] and so on. Disregarding the particular adjective that is used, the common motivation in this trend —as pointed out by [7]— is giving importance not just to the decision of a pattern recognition system (e.g. the assigned class label for classification – the "what") but also to the reason of the decision (the "why"). The former aspect is sought to build well performing PR systems (e.g. accurate classifiers); the second allows to enhance our understanding about the phenomena as well as about the algorithms themselves. From the perspective of this interpretability, an excellent choice is represented by the nearest neighbor rule (NN) [8]–[10]. This method implements an easy and human-understandable rule: a test object is assigned to the class of the training object which is most similar to it. More in general, the *K-nearest neighbor rule* (KNN) assigns an object to the *most frequent* class among the K objects of the training set which are nearest to the testing object (i.e. $K = 1$ in the nearest neighbor rule). Although this approach does not exploit density estimation or function optimization procedures – it entirely relies on the user defined distance measure –, its accuracy is often very competitive with respect to alternatives, provided that a suitable dissimilarity measure is chosen along with a proper training set.

Over the years, numerous variants of the basic Nearest Neighbor rule have been proposed. Some of them consist in either reducing the size of the set of prototypes [11] or generating new ones [12]; others focus on proposing novel dissimilarity measures which well behave in high dimensional spaces [13] or which are adaptive to particular local distributions. Two relatively recent and very similar approaches, belonging to the latter category, have been independently proposed: the *Hypersphere Classifier* (HC) [1] and the *Adaptive Nearest Neighbor rule* (ANN) [2]. HC and ANN are both based on the rationale of correcting the distance between the query point $\mathbf{x}$ and a prototype $\mathbf{x}_i$ by using the concept of a hypersphere, centered at $\mathbf{x}_i$, whose radius measures how "inside" a class the prototype $\mathbf{x}_i$ is. More precisely, the radius of a training object $\mathbf{x}_i$ is defined as the distance to the nearest prototype of $\mathbf{x}_i$ which belongs to a different class: a large radius indicates that the other classes are far away from $\mathbf{x}_i$, thus $\mathbf{x}_i$ can be trusted more.

The corrections to the distance implemented by these two techniques are rather different, one using a ratio, the other a difference. Recently [14], it has been shown that the relation between the two corrections is logarithmic: in other words, it has been shown that applying the ANN rule is equivalent to apply the HC rule on distances which are *non linearly normalized* (with a logarithm). Generally speaking, non linear normalization of data – as opposed to standard linear normalization such as z-score standardization – consists in applying a non linear mapping to every direction of the data representation; it has been shown in many different works that such scaling permits very often to highlight hidden structures and to improve the classification accuracies [15]–[20]. Moreover, it has been shown that this non linear scaling can be also beneficial when applied to distances [21], [22]. Clearly, if the non linear scaling is monotonic, it has not effect in those distance-based classifiers which only rely on rankings (such as the KNN methods). However, if the classifier uses more complex mechanisms, this non linear scaling can drastically change the results – see [21], [22] for an analysis in the dissimilarity-based representation. Finally, it is important to note that non linear scaling of distances may emphasize non linearities in the relation between objects: actually it is often claimed [23]–[25] that the good recognition of complex entities requires the consideration of non linear/non-Euclidean relations among them.

Non linear scaling can be implemented with non linear functions such as logarithm, sigmoid, power and others. Starting from the observation that the ANN classifier is the HC

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

classifier applied to distances which have been non linearly scaled with a logarithm, it seems very promising to investigate the effect of other non linear scalings, such as the power transformation – the superiority of the power transformation with respect to the logarithm, for particular feature spaces, has been demonstrated in [18]. This paper is devoted to this, and proposes a variant of the HC classifier, which we call *PowerHC*, in which the distances are scaled with a power transformation before applying the HC classifier. We show with a large scope empirical evaluation, involving 24 standard UCI-ML datasets, that this non linear scaling is rather beneficial with respect to HC (no scaling), also in comparison with the ANN classifier (i.e. the logarithm scaling). As a second contribution, we evaluate the behaviour of these advanced nearest neighbor schemes in a very recent and challenging real world scenario, which involves classification of seismic volcanic events [26]. In this context, the KNN rule has been applied almost always in its basic version, with most of the efforts put in defining the distance or the representation [27]. The analysis provided in this paper shows, on a very large dataset involving more than one thousand of signals gathered at the Nevado del Ruiz volcano in Colombia, that advanced NN techniques can be very useful to classify seismic signals.

The rest of the paper is organized as follows. The original formulations of ANN and HC are presented in Sec. II. Our proposal —the PowerHC rule— is described in Sec. III. The experimental results and comparison with the baseline methods are shown in Sec. IV. Finally, our concluding remarks are given in Sec. V.

II. The ANN and the HC rules

This section presents the two different variants of the Nearest Neighbor rule from which we start our analysis, i.e. the Hypersphere Classifier (HC – [1]) and the Adaptive Nearest Neighbor rule (ANN – [2]).

A. The Hypersphere Classifier

The Hypersphere classifier was originally proposed in [1]: one of its main original characteristics was its ability to reduce the number of prototypes in the training set, to deal with memory restrictions. Obviously, this method can also be exploited without any constraint: actually, in this study, we do not exploit the incremental nature of HC, relying on the basic scheme. Before presenting the version which uses the radius, let us present the version introduced in [1]: the first step is to define as ρ_i the region of influence of a given training point $\mathbf{x}_i$; once defined this ρ_i, the HC rule proposes to compute the distance from a testing object $\mathbf{x}$ to the training point $\mathbf{x}_i$ as follows:

$$d_{HC}(\mathbf{x}, \mathbf{x}_i) = d(\mathbf{x}, \mathbf{x}_i) - g\rho_i, \tag{1}$$

The idea is to give more importance to those training points which have a large “region of influence”, i.e. points on which we can trust more: this is accomplished by reducing their distance to the testing object. The link with the radius can be found in the way the region of influence ρ_i is defined: actually ρ_i is $1/2$ the radius of the largest hypersphere having as center $\mathbf{x}_i$ and not containing any training object belonging to a different class. The radius r_i of this hypersphere, which represents the distance to the nearest training object of $\mathbf{x}_i$ belonging to a different class, can be formally defined as:

$$r_i = \min_{\mathbf{x}_j \in OT(\mathbf{x}_i)} d(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

with

$$OT(\mathbf{x}_i) = \{\mathbf{x}_k \text{ such that } label(\mathbf{x}_k) \neq label(\mathbf{x}_i)\} \tag{3}$$

In Eq. (1), g is a free parameter, which has to be defined by the user. In the definition of ρ_i, the idea is to define the region of influence as half of the radius of the hypersphere in order to avoid the overlap between hyperspheres of different classes. However, in [1] authors suggested that the optimal value for g would be 2: to simplify our notation, here we only consider this setting, and rewrite Eq. (1) by using Eq. (2):

$$d_{HC}(\mathbf{x}, \mathbf{x}_i) = d(\mathbf{x}, \mathbf{x}_i) - r_i. \tag{4}$$

Please note that Eq. (4) may produce negative distances in those cases when the testing object is inside the hypersphere associated to the training object $\mathbf{x}_i$; nevertheless, when using nearest neighbor rules, which are based on ranking, this does not represent a problem – the nearest neighbor rule simply extracts the minimum of the distances to all training objects, independently from the sign.

B. The Adaptive Nearest Neighbor Rule

Apparently, the proposers of the Hypersphere classifier were not aware of a very similar technique which was presented some years before: the Adaptive Nearest Neighbor Rule (ANN – [2]). ANN exploits a reasoning similar to that of [1], since corrects the distance of a testing point $\mathbf{x}$ to a training object $\mathbf{x}_i$ using the radius of the hypersphere associated to that training object, where the hypersphere is defined exactly as in the Hypersphere Classifier. Again, the goal is to give more importance to training points which are well inside their class, i.e. training points with large radius: to realize that, given a testing object, its distances to training points with large hyperspheres are diminished (thus making them “more near”), whereas its distances to training points with small hyperspheres are enlarged (thus making them “more far away”). To get this effect the ANN technique proposes to divide the distances between the testing object $\mathbf{x}$ and the training object $\mathbf{x}_i$ by the radius, as follows:

$$d_{ANN}(\mathbf{x}, \mathbf{x}_i) = \frac{d(\mathbf{x}, \mathbf{x}_i)}{r_i}. \tag{5}$$

Even if implementing the same idea, the behaviour of the corrections in Eq. (5) and Eq. (4) is rather different, since the penalization is much stronger in the former rule (ratio versus difference). Please note that Eq. (5), differently than Eq. (4), does not generate negative values. However, the distance might diverge if $r_i \to 0$, leading to numerical inaccuracies. This problem is solved in [2] by adding an arbitrarily small ϵ to the radius. In general, the numerical problem is unlikely to occur for real-world data satisfying the compactness hypothesis [28].

III. The PowerHC rule

The proposed approach starts from the mathematical relation between ANN and HC which has been recently assessed in [14]: the ANN rule can be seen as equivalent to the application of the HC rule to the logarithm of the original distances. In few words, if we take the logarithm of our input distance $d(\mathbf{x}, \mathbf{x}_i)$, we get a novel distance $\tilde{d}(\mathbf{x}, \mathbf{x}_i)$:

$$\tilde{d}(\mathbf{x}, \mathbf{x}_i) = \log d(\mathbf{x}, \mathbf{x}_i) \tag{6}$$

Note that this operation has no effect on the KNN rule, since the ranking does not change. However, in more complex distance-based classifiers, this operation may have an impact. If we apply the HC rule using this normalized distance $\tilde{d}(\mathbf{x}, \mathbf{x}_i)$ we have (using again the notation that was introduced in Eq. (3)):

$$\tilde{d}_{HC}(\mathbf{x}, \mathbf{x}_i) = \tilde{d}(\mathbf{x}, \mathbf{x}_i) - \tilde{r}_i, \tag{7}$$

where $\tilde{r}_i$ is the radius using the normalized distance. It is easy to see that

$$\tilde{r}_i = \log r_i \tag{8}$$

and that

$$\tilde{d}_{HC}(\mathbf{x}, \mathbf{x}_i) = \log d(\mathbf{x}, \mathbf{x}_i) - \log r_i = \log d_{ANN}(\mathbf{x}, \mathbf{x}_i) \tag{9}$$

i.e. the application of the HC correction to the logarithm of the original distances is equivalent to the application of the logarithm to the ANN correction computed on the original distances.

The PowerHC classifier starts from this relation, and substitutes the logarithm scaling with the power transformation. This transform has been widely used to normalize data, especially in a slightly different variant called Box-Cox transform [29], [30]: this transform, introduced in the 60s, was mainly used to transform a set of points in order to make their distribution approximately Gaussian. Recently, it has been also shown that it can be useful also in classification contexts, especially when used in parameter ranges which are very far from those optimal for Gaussianity [20]. Its usefulness with distances, however, has been poorly investigated – i.e. only for the particular case of dissimilarity-based representation [21], [22].

The PowerHC rule starts by defining, for every object in the training set $\mathbf{x}_i$, the power-radius p_i, which represents the radius computed using the power of the distances:

$$p_i = \min_{\mathbf{x}_j \in OT(\mathbf{x}_i)} d(\mathbf{x}_i, \mathbf{x}_j)^\rho, \quad \rho > 0 \tag{10}$$

with $OT(\mathbf{x}_i)$ as defined in Eq. (3). Given this radius, the PowerHC method computes the new distance between a testing point $\mathbf{x}$ and a prototype $\mathbf{x}_i$ as:

$$d_{PHC}(\mathbf{x}, \mathbf{x}_i) = d(\mathbf{x}, \mathbf{x}_i)^\rho - p_i. \tag{11}$$

Given this corrected distance, as in the case of ANN and HC, the classification is then performed using the NN rule (or the KNN rule).

Depending on the value of ρ, the power transformation can have different effects on the distances, which can be divided mainly in two classes (see Fig. 1). Suppose that the distance takes values in $[0, 1]$: when using $\rho < 1$, the non linear mapping is convex, and small distances are increased, whereas large distances are reduced. The effect is that relations between objects become more balanced, since points tend to have all the same distance to the others. On the contrary, when $\rho > 1$, the non linear mapping is concave: small distances become even smaller, whereas large distances are increased: in this way relations are driven to extreme (neighbor objects become even closer). These two opposite effects may be beneficial or not, depending on the distribution of the classes inside the problem – this will be evident in the experimental section, where we will show that some problems benefit from the concave transformation, whereas for others the convex transformation is more useful.

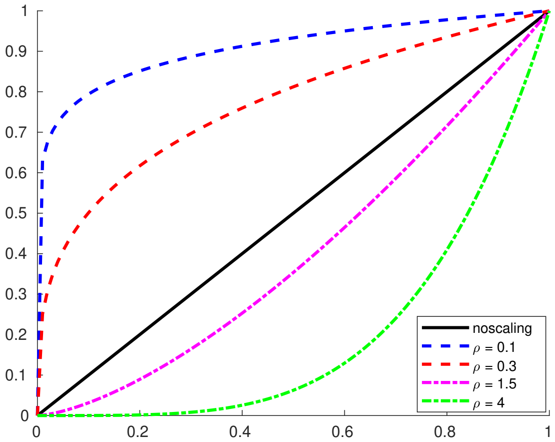

Fig. 1. Effect of the power transformation

IV. Experimental Evaluation

A. Experimental Setup

We used a heterogeneous collection of datasets for the experiments (see Table I), including those that were used in [1], [2], [14]. The heterogeneity of this collection is aimed at evaluating the PowerHC method under several conditions of dimensionality, cardinality and number of classes. In all these cases we used Euclidean distance between vectorial representations.

Moreover, we tested the proposed scheme in a real world challenging scenario, involving the classification of seismic-volcanic signals into a number of predefined categories: this problem is definitely challenging, since distinguishing among the different categories of seismic signals is not easy even for the most experienced analysts [26]. In this context, the nearest-neighbor rule has been applied almost always in its basic version, with most of the efforts put in defining the distance or the representation: we will show that the application of advanced nearest neighbor schemes may be beneficial for the problem. In particular we used a rather large dataset of 1065 signals collected at the Nevado del Ruiz volcano in Colombia, and pre-processed by the Observatorio Vulcanológico y

Sismológico de Manizales, Colombia[1]; there are five different seismic events, which represent the classes: volcano tectonic (VT) events, long period (LP) events, tremors (TR), hybrid (HB) events, and screw-like (TO) earthquakes. Signals are characterized using spectrograms, using 1-second frames, a 128-point FFT, a 64-point Hamming window and an overlap of 50%. From spectrograms we derived two problems: in the first (`volcano_Eucl`), we derived a vectorial representation by averaging the spectrograms, computing the distance using again the Euclidean distance. In the second (`volcano_DTW`) we directly used the spectrograms, characterizing their similarity with the dynamic time warping (DTW) distance [31], [32], a widely used not metric (dis)similarity measure able to take into account the intrinsic temporal and sequential nature of the seismic signals [27].

TABLE I
SUMMARY OF THE DATASETS PROPERTIES

Dataset	# features	# objects	# classes
`german-credit`	20	1000	2
`wine`	13	178	3
`pima`	8	768	2
`sonar`	60	208	2
`wdbc`	30	569	2
`soybean1`	35	266	15
`tic-tac-toe`	9	958	2
`chromo`	8	1143	24
`yeast`	8	1484	10
`vehicles`	18	846	4
`ecoli`	7	336	8
`malaysia`	8	291	20
`arrhythmia`	278	420	12
`imox`	8	192	4
`heart`	13	297	2
`x80`	8	45	3
`haberman`	3	306	2
`soybean2`	35	136	4
`ionosphere`	34	351	2
`iris`	4	150	3
`liver`	6	345	2
`glass`	9	214	6
`wpbc`	32	194	2
`spirals`	2	194	2
`volcano_Eucl`	65	1065	5
`volcano_DTW`	65 × # frames	1065	5

In all problems, accuracies were evaluated by using a repeated training and testing protocol. At each repetition, the dataset was randomly split into two parts of approximately the same size: 50% for training the classifier and the remaining part for estimating the performance. We compute accuracies under two scenarios: (i) when using the NN rule for the final decision, (ii) when using the KNN rule. Averaged accuracies, along with their corresponding standard errors, are shown in Tables II and III, for NN and KNN, respectively. Notice that all accuracies are presented as percentages in order to facilitate the visualization of significant figures in both the accuracies and their corresponding standard errors. In order to distinguish the methods when applied under these scenarios, either NN or KNN is used as a prefix for the name of the compared methods. To have a more clear analysis of the differences between PowerHC and both ANN and HC, we used a paired t-test at a 5% of significance to test the accuracies. Our null hypothesis is that "the performance of PowerHC and the baseline method (either ANN or HC) is effectively the same". When the hypothesis is rejected —that is, when PowerHC and the baseline are significantly different— in Tables II and III we point with an arrow to the name of the method exhibiting the highest performance (e.g., in table II, first line, column "B vs D", "Reject $\nearrow$" indicates that B (NN-ANN) and D (NN-PowerHC) are different with a statistical significance, and that the best method is D (NN-PowerHC)).

The most important parameter involved in our experiments is the power ρ in PowerHC. We explored the effect of its values in the range $\rho \in \{0.2, 0.4, \ldots, 9.8, 10\}$; to completely understand the potentialities of the proposed approach, for the comparisons we selected for every experiment the parameter values that yield the best classification results – we report such values in the tables below. An analysis on how to select ρ in an automatic way is reported in the final part of this section (namely Sect. IV-D). Regarding the results with KNN, we repeated the tests with $K \in \{1, 3, \ldots, 27, 29\}$, reporting again in Table III the best obtained result.

B. Comparison of NN-PowerHC vs NN-ANN and NN-HC

According to the acceptance or rejection of the t-tests, results in Table II were grouped and marked with a symbol preceding the name of the dataset. The first group, marked with a ★, corresponds to cases were NN-PowerHC is better than both NN-ANN and NN-HC. Notice that, in this group the best ρ for each dataset is always larger than or equal to 2.0; that is, in these cases, diminishing small distances and, at the same time, increasing the large ones is beneficial. This is somehow in contrast with other studies [17], [18], [22], in which better results were obtained with the power less than 1. Accuracy differences of NN-PowerHC with respect to NN-ANN range from 0.78 to 5.51, which correspond to `yeast` and `arrhythmia`, respectively. However, even though the improvement for `arrhythmia` is the largest one, we must take into account that, in that case, NN-ANN is worse than NN. Similarly, improvements of NN-PowerHC with respect to NN-HC range from 0.53 to 2.56, which correspond to the same datasets.

The second and the third groups correspond to cases where NN-PowerHC is better than one of the baseline methods but equal to the other one. In particular, when NN-PowerHC is better than NN-ANN but equal to NN-HC (symbol ■), we notice that improvements range from 0.28 for `wdbc` to 1.41 for `volcano_DTW`. In this group, it can be highlighted that the best ρ values are around 1.55. The subsequent group, the one marked with ▲, shows improvements of NN-PowerHC with respect to NN-HC that range from 0.12 to 0.54 for `vehicles` and `iris`. In contrast with the second group, in this one all the best ρ values are smaller than 1.0; that is, homogenizing the distances in this group (remember that the transformation

[1] We thank John Makario Londoño-Bonilla and the Observatorio Vulcanológico y Sismológico de Manizales, Colombia for providing the data.

TABLE II
ACCURACIES, AS PERCENTAGES, ALONG WITH STANDARD ERRORS FOR 50 REPETITIONS AND T-TESTS AT 5% OF SIGNIFICANCE FOR THE COMPARED METHODS WHEN USING NN FOR DECISION. ARROWS POINT TO THE BEST METHOD WHEN DIFFERENCES ARE SIGNIFICANT.

	Accuracies				t-tests	
Method Dataset	A NN	B NN-ANN	C NN-HC	D NN-PowerHC	B vs D	C vs D
★ `german-credit`	68.72±0.29	71.32±0.29	71.58±0.29	72.59±0.28 ($\rho = 6.2$)	Reject↗	Reject↗
★ `pima`	69.70±0.33	72.43±0.32	72.70±0.32	73.54±0.32 ($\rho = 6.0$)	Reject↗	Reject↗
★ `tic-tac-toe`	79.52±0.26	80.86±0.25	83.04±0.24	84.36±0.23 ($\rho = 5.8$)	Reject↗	Reject↗
★ `yeast`	51.02±0.26	53.53±0.26	53.78±0.26	54.31±0.26 ($\rho = 3.4$)	Reject↗	Reject↗
★ `arrhythmia`	57.88±0.48	55.07±0.49	58.02±0.48	60.58±0.48 ($\rho = 2.0$)	Reject↗	Reject↗
★ `heart`	76.55±0.49	78.23±0.48	78.50±0.48	79.28±0.47 ($\rho = 3.2$)	Reject↗	Reject↗
★ `haberman`	66.32±0.54	68.70±0.53	68.70±0.53	69.70±0.53 ($\rho = 9.6$)	Reject↗	Reject↗
■ `wdbc`	95.06±0.18	96.16±0.16	96.36±0.16	96.44±0.16 ($\rho = 1.6$)	Reject↗	Accept
■ `ecoli`	81.79±0.42	83.52±0.40	84.14±0.40	84.30±0.40 ($\rho = 1.8$)	Reject↗	Accept
■ `volcano_DTW`	72.55±0.27	78.41±0.25	79.75±0.25	79.82±0.25 ($\rho = 1.6$)	Reject↗	Accept
■ `glass`	68.50±0.64	66.91±0.64	67.66±0.64	67.79±0.64 ($\rho = 1.2$)	Reject↗	Accept
▲ `sonar`	83.44±0.52	84.82±0.50	84.49±0.50	84.85±0.50 ($\rho = 0.2$)	Accept	Reject↗
▲ `iris`	93.33±0.41	94.40±0.38	93.89±0.39	94.43±0.37 ($\rho = 0.2$)	Accept	Reject↗
▲ `liver`	61.45±0.52	61.40±0.52	61.06±0.52	61.40±0.52 ($\rho = 0.6$)	Accept	Reject↗
▲ `vehicles`	69.11±0.32	68.79±0.32	68.74±0.32	68.86±0.32 ($\rho = 0.8$)	Accept	Reject↗
▲ `malaysia`	70.64±0.53	69.05±0.54	68.72±0.54	69.08±0.54 ($\rho = 0.2$)	Accept	Reject↗
◊ `ionosphere`	85.21±0.38	93.36±0.27	93.19±0.27	93.36±0.27 ($\rho = 0.2$)	Accept	Accept
◊ `wpbc`	65.59±0.68	71.46±0.65	71.18±0.65	71.65±0.65 ($\rho = 3.4$)	Accept	Accept
◊ `wine`	95.00±0.33	95.93±0.30	96.00±0.29	96.05±0.29 ($\rho = 0.6$)	Accept	Accept
◊ `chromo`	55.34±0.29	55.24±0.29	55.28±0.29	55.35±0.29 ($\rho = 0.8$)	Accept	Accept
◊ `volcano_Eucl`	73.91±0.27	75.73±0.26	75.69±0.26	75.75±0.26 ($\rho = 0.4$)	Accept	Accept
◊ `soybean1`	85.16±0.44	84.24±0.45	84.41±0.44	84.59±0.44 ($\rho = 1.6$)	Accept	Accept
◊ `imox`	92.94±0.37	91.52±0.40	91.73±0.40	91.79±0.40 ($\rho = 0.8$)	Accept	Accept
◊ `x80`	94.38±0.68	88.29±0.95	88.29±0.95	88.95±0.92 ($\rho = 3.6$)	Accept	Accept
◊ `soybean2`	82.03±0.66	81.62±0.66	81.62±0.66	81.79±0.66 ($\rho = 1.2$)	Accept	Accept
△ `spirals`	74.25±0.63	68.56±0.67	67.44±0.67	68.21±0.67 ($\rho = 0.2$)	↖Reject	Reject↗

in this case is convex) enhances HC but the improvement is not enough to reach the performance of NN.

In 9 out of 26 cases, there are no significant differences between the performance of NN-PowerHC and the baseline methods; see the group marked with ◊. Finally, the △ singleton composed by `spirals` is an atypical case in which none of the advanced methods is better than NN. This dataset was intentionally included in the collection of [14] for the sake of illustrating a case where ANN and HC are counterproductive. In fact, notice that even though NN-PowerHC improves over NN-HC for `spirals`, its accuracy is significantly lower than the one of NN-ANN and, in turn, both are far from the performance of NN.

Finally, for what concerns the volcanic datasets, two observations should be done: i) PowerHC is again slightly better than HC and ANN, but only when using the DTW distance, ii) more importantly, advanced nearest neighbor techniques largely improve the Nearest Neighbor rule, especially when we use the non metric DTW distance.

C. Comparison of KNN-PowerHC vs KNN-ANN and KNN-HC

The same marking conventions that were used in Table II are also followed in the analysis of Table III. With respect to the first group (★), we noticed that the best improvement obtained by KNN-PowerHC over both KNN-ANN and KNN-HC still corresponds to `arrhythmia`, but again, KNN-ANN is worse in this dataset than KNN. The span of the best ρ values in this group is large: from 1.8 to 9.6; however, as in the previous case, they correspond to concave transformations.

When comparing Table III against the first group in Table II, we can notice that `pima` and `yeast` were slightly degraded from ★ to ■. In the opposite relation, we can see that `ecoli`, `volcano_DTW`, `liver` and `x80` were upgraded to the first group in Table III. The case of the latter dataset is the most significant upgrade (from ◊ to ★) but also, as for `arrhythmia`, in `x80` KNN-ANN is worse than KNN. Moreover, `x80` is similar to `spirals` in the sense that KNN is, overall, the best option. Moving to the second and third groups, `glass`, `sonar`, `malaysia` and `iris` were kept in the same groups as in Table II. In contrast, for `wdbc` and `vehicles`, KNN-PowerHC lost its advantage with respect to the baseline that had when considering NN-PowerHC. Notice that `ionosphere` accompanies now `spirals` in the △ group. However, in `ionosphere`, the improvement of all the advanced methods is significant in comparison to KNN.

Finally, when considering volcano datasets, we can observe that with PowerHC and DTW we obtain a remarkable 82.19% of accuracy, which is largely higher than that obtained with basic NN (72.55% with DTW or 73.91% with Euclidean distance) or KNN (73.62 with DTW or 74.99% with Euclidean distance). This suggests that non linear - non metric characterization of relations may be fundamental in highly complex real problems.

TABLE III
ACCURACIES, AS PERCENTAGES, ALONG WITH STANDARD ERRORS FOR 50 REPETITIONS AND T-TESTS AT 5% OF SIGNIFICANCE FOR THE COMPARED METHODS WHEN USING KNN FOR DECISION. ARROWS POINT TO THE BEST METHOD WHEN DIFFERENCES ARE SIGNIFICANT.

	Accuracies				t-tests	
Method Dataset	A KNN	B KNN-ANN	C KNN-HC	D KNN-PowerHC	B vs D	C vs D
★ german-credit	73.82±0.28	72.75±0.28	73.22±0.28	74.22±0.28 ($\rho = 8.8$)	Reject↗	Reject↗
★ tic-tac-toe	83.37±0.24	82.66±0.24	83.04±0.24	84.36±0.23 ($\rho = 5.8$)	Reject↗	Reject↗
★ arrhythmia	63.00±0.47	61.57±0.47	64.98±0.47	68.89±0.45 ($\rho = 1.8$)	Reject↗	Reject↗
★ haberman	75.03±0.49	74.64±0.50	74.76±0.50	75.26±0.49 ($\rho = 2.0$)	Reject↗	Reject↗
★ liver	63.74±0.52	62.48±0.52	64.15±0.52	65.08±0.51 ($\rho = 2.8$)	Reject↗	Reject↗
★ volcano_DTW	73.62±0.27	78.78±0.25	80.95±0.24	82.19±0.23 ($\rho = 2.2$)	Reject↗	Reject↗
★ ecoli	86.64±0.37	84.80±0.39	85.60±0.38	86.20±0.38 ($\rho = 3.4$)	Reject↗	Reject↗
★ heart	83.26±0.43	81.01±0.45	82.12±0.44	83.07±0.43 ($\rho = 9.6$)	Reject↗	Reject↗
★ x80	94.38±0.68	88.29±0.95	88.29±0.95	90.57±0.86 ($\rho = 2.8$)	Reject↗	Reject↗
■ pima	74.90±0.31	75.52±0.31	75.88±0.31	75.90±0.31 ($\rho = 0.8$)	Reject↗	Accept
■ yeast	58.22±0.26	57.29±0.26	58.44±0.26	58.72±0.26 ($\rho = 1.8$)	Reject↗	Accept
■ glass	68.50±0.64	66.91±0.64	67.66±0.64	67.79±0.64 ($\rho = 1.2$)	Reject↗	Accept
▲ sonar	83.44±0.52	84.82±0.50	84.49±0.50	84.85±0.50 ($\rho = 0.2$)	Accept	Reject↗
▲ malaysia	70.64±0.53	69.05±0.54	68.72±0.54	69.08±0.54 ($\rho = 0.2$)	Accept	Reject↗
▲ iris	95.25±0.35	94.40±0.38	93.89±0.39	94.43±0.37 ($\rho = 0.2$)	Accept	Reject↗
◊ wdbc	96.14±0.16	96.30±0.16	96.39±0.16	96.44±0.16 ($\rho = 1.6$)	Accept	Accept
◊ wpbc	76.33±0.61	76.37±0.61	76.31±0.61	76.41±0.61 ($\rho = 8.2$)	Accept	Accept
◊ chromo	55.34±0.29	55.24±0.29	55.28±0.29	55.35±0.29 ($\rho = 0.8$)	Accept	Accept
◊ volcano_Eucl	74.99±0.27	75.73±0.26	75.73±0.26	75.79±0.26 ($\rho = 3.0$)	Accept	Accept
◊ wine	96.91±0.26	95.93±0.30	96.00±0.29	96.05±0.29 ($\rho = 0.6$)	Accept	Accept
◊ soybean1	85.16±0.44	84.24±0.45	84.41±0.44	84.59±0.44 ($\rho = 1.6$)	Accept	Accept
◊ vehicles	69.90±0.32	68.79±0.32	68.74±0.32	69.3±0.32 ($\rho = 9.8$)	Accept	Accept
◊ imox	92.94±0.37	91.52±0.40	91.73±0.40	91.79±0.40 ($\rho = 0.8$)	Accept	Accept
◊ soybean2	82.03±0.66	81.62±0.66	81.62±0.66	81.79±0.66 ($\rho = 1.2$)	Accept	Accept
△ ionosphere	85.21±0.38	94.02±0.25	93.43±0.26	93.92±0.25 ($\rho = 0.2$)	↖Reject	Reject↗
△ spirals	74.25±0.63	68.56±0.67	67.44±0.67	68.21±0.67 ($\rho = 0.2$)	↖Reject	Reject↗

D. The tuning of ρ

As mentioned in Sec. IV-A, to inspect the whole potentialities of the proposed PowerHC we select the optimal ρ, i.e. the parameter leading to the optimal accuracy. Clearly, in realistic scenarios, such parameter should be set in advance. In this section we performed a comparison between the optimal accuracy and that obtained with an automatic tuning approach based on Cross validation (as done in [22]). In particular, the optimal ρ is the one which minimizes the Leave One Error of the Nearest Neighbor rule on the training set. In Fig. 2 we report the comparisons for NN-PowerHC (left) and KNN-PowerHC (right).

Notice that the automatic method is only slightly worse than the best value. For some datasets, the difference is more notorious than for others as can be observed for arrhythmia and malaysia; in contrast, in other cases the result is in practice the same: see for instance ionosphere, iris and wdbc. No particular difference, with respect to the tuning approach, is observed between NN-PowerHC and KNN-PowerHC. Please note that, once given the distances, this automatic selection rule may be implemented in a very efficient way (three lines of code), thus resulting in a viable approach to automatic selection of ρ.

V. CONCLUSIONS

In this paper we investigated the suitability of non linear scaling of distances to improve standard as well as advanced Nearest Neighbor approaches. In particular we studied PowerHC, a method which normalizes distances with a power transformation prior to applying the HC classifier. The performed experimental evaluation, conducted on a rather large set of datasets, confirms the suitability of the proposed approach. Remarkably, we have shown that on a real world challenging application related to the classification of volcano-seismic signals, advanced nearest neighbor techniques – and especially the PowerHC – can be very beneficial.

REFERENCES

[1] N. Lopes and B. Ribeiro, “Incremental Hypersphere Classifier (IHC),” in *Machine Learning for Adaptive Many-Core Machines - A Practical Approach*, ser. Studies in Big Data. Cham: Springer International Publishing, 2015, vol. 7, ch. 6, pp. 107–123.

[2] J. Wang, P. Neskovic, and L. N. Cooper, “Improving nearest neighbor rule with a simple adaptive distance measure,” *Pattern Recognition Letters*, vol. 28, no. 2, pp. 207 – 213, 2007.

[3] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (XAI),” *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[4] A. Holzinger, “From machine learning to explainable AI,” in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, Aug. 2018, pp. 55–66.

[5] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” *arXiv*, 2017. [Online]. Available: https://arxiv.org/abs/1702.08608

[6] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, “Interpretable machine learning: definitions, methods, and applications,” *arXiv*, 2019. [Online]. Available: http://arxiv.org/abs/1901.04592

[7] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Christoph Molnar, 2019. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[8] E. Fix and J. L. Hodges Jr, “Discriminatory analysis-nonparametric discrimination: consistency properties,” DTIC Document, Tech. Rep., 1951.

[9] ——, “Discriminatory analysis-nonparametric discrimination: Small sample performance,” DTIC Document, Tech. Rep., 1952.

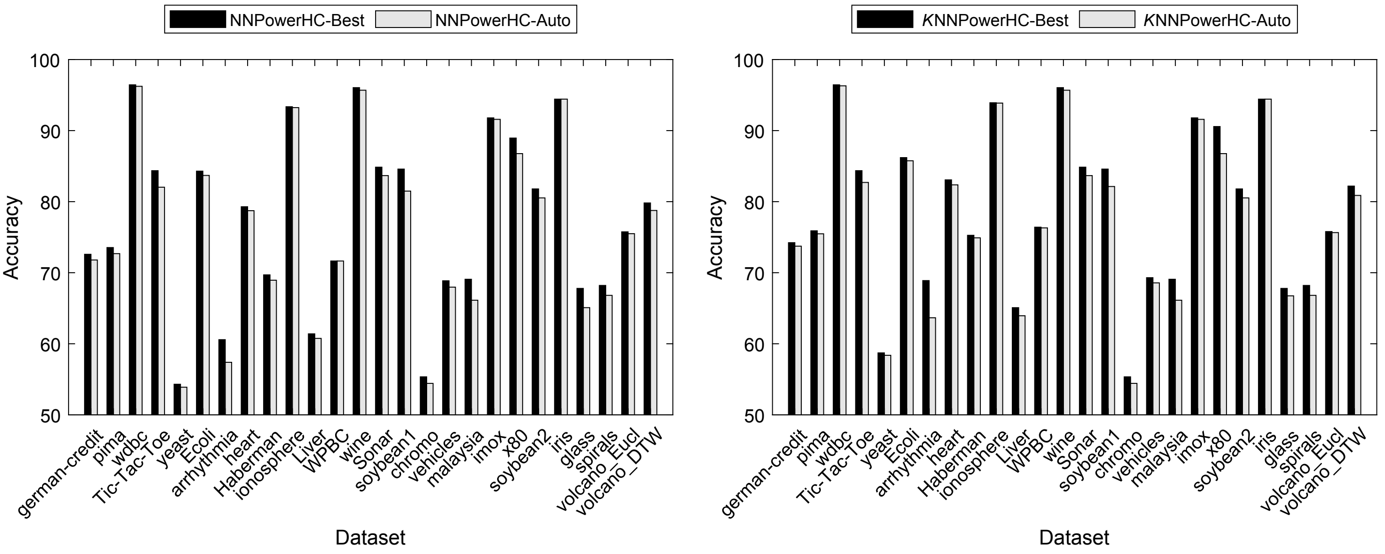

Fig. 2. Comparison of Optimal (Best) vs Automatic Approach for tuning ρ in: (left) NN-PowerHC and (right) KNN-PowerHC

[10] T. Cover and P. Hart, "The nearest neighbor decision rule," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 21–27, 1967.

[11] E. Pekalska, R. P. W. Duin, and P. Paclík, "Prototype selection for dissimilarity-based classifiers," *Pattern Recognition*, vol. 39, no. 2, pp. 189 – 208, 2006, part Special Issue: Complexity Reduction.

[12] I. Triguero, J. Derrac, S. García, and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, no. 1, pp. 86–100, 2012.

[13] A. K. Pal, P. K. Mondal, and A. K. Ghosh, "High dimensional nearest neighbor classification based on mean absolute differences of inter-point distances," *Pattern Recognition Letters*, vol. 74, pp. 1–8, 2016.

[14] M. Orozco-Alzate, S. Baldo, and M. Bicego, "Relation, transition and comparison between the adaptive nearest neighbor rule and the hypersphere classifier," in *Proc. Int. Conf. on Image Analysis and Processing (ICIAP2019)*, 2019.

[15] R. V. D. Heiden and F. Groen, "The box-cox metric for nearest neighbour classification improvement," *Pattern Recognition*, vol. 30, no. 2, pp. 273–279, 1997.

[16] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques," *Pattern Recognition*, vol. 37, no. 2, pp. 265–279, 2004.

[17] A. Carli, M. Bicego, S. Baldo, and V. Murino, "Non-linear generative embeddings for kernels on latent variable models," in *Proc. of ICCV09 Workshop on Subspace Methods*, 2009, pp. 154–161.

[18] ——, "Nonlinear mappings for generative kernels on latent variable models," in *Proc. Int. Conf. on Pattern Recognition*, 2010, pp. 2134–2137.

[19] M. Pan, L. Du, P. Wang, and Z. B. H. Liu, "Multi-task hidden markov modeling of spectrogram feature from radar high-resolution range profiles," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, p. 86, 2012.

[20] M. Bicego and S. Baldo, "Properties of the Box-Cox transformation for pattern classification," *Neurocomputing*, vol. 218, pp. 390 – 400, 2016.

[21] M. Orozco-Alzate, R. P. W. Duin, and M. Bicego, "Unsupervised parameter estimation of non linear scaling for improved classification in the dissimilarity space," in *Proc. Joint Int. Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, 2016, pp. 74–83.

[22] R. P. W. Duin, M. Bicego, M. Orozco-Alzate, S.-W. Kim, and M. Loog, "Metric learning in dissimilarity space for improved nearest neighbor performance," in *Proc. Joint Int. Workshop on Structural, Syntactic and Statistical Pattern Recognition*, vol. 8621, 2014, pp. 183–192.

[23] W. Scheirer, M. Wilber, M. Eckmann, and T. Boult, "Good recognition is non-metric," *Pattern Recognition*, vol. 47, no. 8, pp. 2721–2731, 2014.

[24] S. Huang, J. Lu, J. Zhou, and A. Jain, "Nonlinear local metric learning for person re-identification," 2015. [Online]. Available: arXiv: http://arxiv.org/abs/1511.05169

[25] D. Kedem, S. Tyree, K. Weinberger, F. Sha, and G. Lanckriet, "Non-linear metric learning," in *NIPS 2012*, 2012, pp. 2582–2590.

[26] M. Orozco-Alzate, C. Acosta-Muñoz, and J. M. Londoño-Bonilla, "The Automated Identification of Volcanic Earthquakes: Concepts, Applications and Challenges," in *Earthquake Research and Analysis - Seismology, Seismotectonic and Earthquake Geology*, S. D'Amico, Ed. Rijeka, Croatia: InTech, feb 2012, ch. 19, pp. 345–370.

[27] M. Orozco-Alzate, P. A. Castro-Cabrera, M. Bicego, and J. M. Londoño-Bonilla, "The DTW-based representation space for seismic pattern classification," *Computers & Geosciences*, vol. 85(B), pp. 86–95, 2015.

[28] R. P. W. Duin, "Compactness and complexity of pattern recognition problems," in *Proc. Int. Symposium on Pattern Recognition "In Memoriam Pierre Devijver"*, C. Perneel, Ed., Feb. 1999, pp. 124–128.

[29] R. M. Sakia, "The Box-Cox transformation technique: A review," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 41, no. 2, pp. 169–178, 1992.

[30] G. E. P. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 26, no. 2, pp. 211–243, 1964.

[31] D. Lemire, "Faster retrieval with a two-pass dynamic-time-warping lower bound," *Pattern Recognition*, vol. 42, no. 9, pp. 2169 – 2180, 2009.

[32] J. Lin, S. Williamson, K. D. Borne, and D. DeBarr, "Pattern recognition in time series," in *Advances in Machine Learning and Data Mining for Astronomy*, 2012, ch. 28, pp. 617–646.