Relation, Transition and Comparison Between the Adaptive Nearest Neighbor Rule and the Hypersphere Classifier

Mauricio Orozco-Alzate[1](✉), Sisto Baldo[2], and Manuele Bicego[2]

[1] Departamento de Informática y Computación, Universidad Nacional de Colombia - Sede Manizales, km 7 vía al Magdalena, Manizales 170003, Colombia
morozcoa@unal.edu.co

[2] Dipartimento di Informatica, Università degli Studi di Verona, Strada Le Grazie, 15, Verona 37134, Italy
{sisto.baldo,manuele.bicego}@univr.it

Abstract. The Adaptive Nearest Neighbor (ANN) rule and the Hypersphere Classifier (HC) are two very simple and relatively new variants of the classical nearest neighbor (1NN) rule. Even if they share a similar formulation—they correct the query-to-prototype distance by taking into account the distance of the prototype to the nearest one from other classes—their relation has never been investigated. The main goal of this paper is studying this relation and providing an exhaustive performance comparison of both methods, highlighting occasions when their performances differ as well as identifying cases in which their application is advisable or leads to poorer results. Moreover, we propose a smooth transition between the two classifiers by studying the use of several convex combinations of their penalized distances. Experiments show that a combination is particularly helpful when both ANN and HC are worse than 1NN.

Keywords: Adaptive Nearest Neighbor · Convex combination · Comparison · Hypersphere Classifier · Relation · Transition

1 Introduction

The nearest neighbor rule (1NN) [1,2] represents a well known and widely applied classifier, which assigns an unknown object (query or test object) to the class of the object of the training set (prototype) whose distance to the testing object is minimum (i.e. the nearest neighbor). Over the years, numerous variants for improving this rule have been proposed. Some of them consist in either reducing the size of the set of prototypes [3] or generating new ones [4]; others focus on proposing novel dissimilarity measures and making them well-behaved in high dimensional spaces [5] or adaptive to particular local distributions. Two relatively recent and very similar approaches belong to the latter category, which have been independently proposed, namely the *Hypersphere Classifier* (HC) [6]

E. Ricci et al. (Eds.): ICIAP 2019, LNCS 11751, pp. 141–151, 2019.
https://doi.org/10.1007/978-3-030-30642-7_13

and the *Adaptive Nearest Neighbor rule* (ANN) [7]. Apparently, authors of HC—the most recently proposed method—were not aware of ANN since they do not refer to it in spite of the clear relationship between the two methods.

HC and ANN are both based on the rationale of penalizing 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 is defined by the distance to the prototype's nearest prototype which belongs to a different class. This radius measures how "inside" a class a given prototype is – a large radius indicates that the other classes are far away from it, thus it can be trusted more. Given this radius, both HC and ANN correct the distance of the testing point to the prototype: HC subtracts it from $||\mathbf{x} - \mathbf{x}_i||$ while ANN divides $||\mathbf{x} - \mathbf{x}_i||$ by the radius. In both cases, prototypes well inside their class have more importance (their distance to the testing object is decreased). Despite the idea behind the two approaches is very similar, a relation between them has not been analyzed yet, this representing the first goal of this manuscript. Actually, an empirical comparison of these methods would serve not just to judge whether there are significant performance differences between the two methods but also to better understand the overall effect of the corresponding penalizations.

The second goal of this paper originates from the fact that another way of improving the behavior of the (dis)similarity measures for classification is by combining them, such that the resulting measure outperforms the individual ones. In this paper we investigated a simple combination of the two penalized distances, in order to show if it is possible to improve even more the accuracies. One of the simplest possibilities is to use a convex linear combination. According to [8], such a combination of two distance functions is particularly useful when combining an overestimate and an underestimate of the Euclidean distance, provided that both are either suitable for non-Euclidean topological spaces or cheaper to be computed than the Euclidean distance itself, by, for instance, avoiding the computation of costly square root operations. Kernels—i.e. similarity functions—have also been interpolated by convex combinations. Gönen and Alpaydın [9], referring to [10], point out that the convex combination—or, more in general, a weighted average—is beneficial if both kernels exhibit similar classification performances but their class assignments rely sometimes on different support vectors.

Summarizing, the main contributions of this paper are the following: (i) first, we highlight the affinity and discuss the relation between HC and ANN; (ii) we compare their behaviors in terms of accuracy; (iii) we propose a modified convex combination of them in order to give further insights on the transition from one to the other. The remaining part of the paper is organized as follows. HC and ANN are explained in more detail in Sect. 2. Afterwards, in Sect. 3, their relation is analyzed and four linear transitions between HC and ANN by convex combinations are proposed. Experimental results and their discussions are given in Sect. 4. Finally, our concluding remarks are provided in Sect. 5.

2 Methods

In this section we introduce the two variants of 1NN, namely the Hypersphere Classifier and the Adaptive Nearest Neighbor Rule. Then, we study their relation in terms of a logarithmic scaling and, afterwards, present a simple model for the transition between the two.

2.1 The Hypersphere Classifier

This classifier was originally proposed [6] as an incremental method, usable to reduce the number of prototypes. Clearly it can also be used without memory restrictions and, therefore, without forgetting prototypes. In this study, we do not make use of the incremental property of HC. Let us present the approach starting from [6], coming later to the formulation with the radius. In [6] authors define as ρ_i the region of influence of $\mathbf{x}_i$; given that, the distance from $\mathbf{x}$ to $\mathbf{x}_i$ is computed as follows:

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

The region of influence ρ_i is defined as 1/2 of the radius of the hypersphere associated to $\mathbf{x}_i$, namely the hypershpere having as center $\mathbf{x}_i$ and as radius (r_i) the distance to the nearest prototype of $\mathbf{x}_i$ belonging to a different class. The radius r_i 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

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

In Eq. (1), g is a free parameter. Even though ρ_i is defined as half of the radius of the hypersphere in order to avoid overlapping between hyperspheres from different classes, in [6] it is shown that the best value for g is 2 which, in words, means that the best configuration is considering the region of influence as the whole volume of the hypersphere in spite of the overlapping, i.e. $r_i = 2\rho_i$. For the sake of simplification, here we only consider that recommended configuration and use Eq. (2) to rewrite Eq. (1) as:

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

Notice that Eq. (4) produces negative distances when a query point is inside the hypersphere associated to $\mathbf{x}_i$; this is not a practical problem with the nearest neighbor rule, which simply takes the minimum of the distances to all prototypes (no matter this value is negative).

2.2 The Adaptive Nearest Neighbor Rule

Similarly to HC, this classifier [7] weights distances of a testing point to a prototype according to the size of the hypersphere associated to that prototype—the hypersphere is defined as in the HC method. Similarly to HC, the effect is to promote prototypes well inside their class: distances to points having small hyperspheres are enlarged while distances to points having large hyperspheres are diminished. This effect is simply obtained by dividing the distances by the radius, as follows:

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

Notice that Eq. (5) does not generate negative values but has a much stronger penalization than the one of Eq. (4). However, the distance might diverge if $r_i \rightarrow 0$. In order to avoid the uncontrolled increase of d_{ANN}, in [7] it is proposed to add an arbitrarily small ϵ to the radius. In general, the numerical problem is unlikely to occur for real-world data satisfying the compactness hypothesis [11].

3 Relation and Transition Between HC and ANN

Relation. It has been shown in some recent works [12,13] that scaling the distance with a convex non linear transformation can be beneficial for distance-based classifiers. One example of such non linear scaling is to raise the distance to a power less than one. Another possibility, which has been investigated for the feature space but not for distances [14], is to use the logarithm, which has the same convex monotonic behavior of the power transformation (for feature spaces, the power transformation corresponds to the well known Box-Cox transform [15,16]).

Clearly, in distance-based classifiers, such monotonic transformation has no effect if the classifier only relies on rankings (such as the K-Nearest Neighbor methods). However, if the classifier uses more complex mechanisms, this non linear scaling can drastically change the results – see [12,13] for an analysis in the dissimilarity-based representation.

Suppose now that we apply the non linear scaling logarithm to our input distance $d(\mathbf{x}, \mathbf{x}_i) = ||\mathbf{x} - \mathbf{x}_i||$, getting a novel distance $\tilde{d}(\mathbf{x}, \mathbf{x}_i)$:

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

Consider again the notation that was introduced in Eq. (3). Now the HC rule redefines the distance with $\tilde{d}_{HC}(\mathbf{x}, \mathbf{x}_i)$:

$$\tilde{d}_{HC}(\mathbf{x}, \mathbf{x}_i) = \tilde{d}(\mathbf{x}, \mathbf{x}_i) - \tilde{r}_i, \qquad \text{where} \quad \tilde{r}_i = \min_{\mathbf{x}_j \in OT(\mathbf{x}_i)} \tilde{d}(\mathbf{x}_i, \mathbf{x}_j) \tag{7}$$

This radius can be expressed in terms of original distances, as follows:

$$\begin{aligned}\tilde{r}_i &= \log\left(\min_{\mathbf{x}_j \in OT(\mathbf{x}_i)} e^{\tilde{d}(\mathbf{x}_i,\mathbf{x}_j)}\right)\\ &= \log\left(\min_{\mathbf{x}_j \in OT(\mathbf{x}_i)} e^{\log d(\mathbf{x}_i,\mathbf{x}_j)}\right)\\ &= \log\left(\min_{\mathbf{x}_j \in OT(\mathbf{x}_i)} d(\mathbf{x}_i,\mathbf{x}_j)\right)\\ &= \log r_i \end{aligned} \tag{8}$$

Where the first step is possible since the exponential does not change the argument of the minimum of the distance. Now,

$$\begin{aligned}\tilde{d}_{HC}(\mathbf{x},\mathbf{x}_i) =&\ \tilde{d}(\mathbf{x},\mathbf{x}_i) - \tilde{r}_i\\ =&\ \log d(\mathbf{x},\mathbf{x}_i) - \log r_i\\ =&\ \log \frac{d(\mathbf{x},\mathbf{x}_i)}{r_i}\\ =&\ \log d_{ANN}(\mathbf{x},\mathbf{x}_i)\end{aligned} \tag{9}$$

Therefore, 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.

Transition. The above-mentioned affinity of HC and ANN motivated us to propose a link between the two classifiers. Let us call $s = ||\mathbf{x} - \mathbf{x}_i||$ and $t = r_i$. Given that, the two distances d_{ANN} and d_{HC} can be rewritten as s/t and $s - t$, respectively. In order to combine d_{ANN} and d_{HC}, we propose four variants of their convex combination:

$$d_\lambda(s,t) = (1-\lambda)\frac{s}{t} + \lambda(s-t) \tag{10}$$

$$d_\lambda(s,t) = \frac{(1-\lambda)s}{(t+\lambda)} + \lambda(s-t) \tag{11}$$

$$d_\lambda(s,t) = \frac{(1-\lambda)s}{(t+\lambda^2)} + \lambda(s-t) \tag{12}$$

$$d_\lambda(s,t) = \frac{(1-\lambda)s}{(t+\sqrt{\lambda})} + \lambda(s-t) \tag{13}$$

Equation (10) corresponds to the canonical convex combination of s and t, where $\lambda \in [0,1]$ controls the transition from ANN to HC. In order to cope with a possible singularity, Eq. (11) might be preferred instead, as well as other variants that damp faster or slower the singularity, e.g. Eqs. (12) and (13). Please note that also in these variants $\lambda \in [0,1]$ controls the transition from ANN to HC (for $\lambda = 0$ we have the d_{ANN} distance, whereas for $\lambda = 1$ we have the d_{HC} one).

4 Experimental Results and Discussion

For the sake of reproducible research and fair comparison, we consider the union of the two collections of data sets that were used for the experiments in the original papers of HC and ANN. In [6], results were computed for the following data sets: WDBC, Ecoli, German credit data, Glass, Haberman, Heart, Ionosphere, Iris, Pima, Sonar, Tic-Tac-Toe, Vehicles, Wine and Yeast. In [7], experiments were performed for WDBC, Ionosphere, Pima, Liver and Sonar. Besides, with the aim of considering a wider range of data conditions, we included additional data sets to the collection; namely: Arrhytmia, WPBC, Soybean1, Soybean2, Malaysia, x80, Imox, Chromo and Spirals. The main properties of the collection of 24 data sets are summarized in Table 1.

Table 1. Main properties of the considered data sets

Dataset	# feat	# obj	# class
German-credit	20	1000	2
Pima	8	768	2
WDBC	30	569	2
Tic-Tac-Toe	9	958	2
Yeast	8	1484	10
Ecoli	7	336	8
Arrhythmia	278	420	12
Heart	13	297	2
Haberman	3	306	2
Ionosphere	34	351	2
Liver	6	345	2
WPBC	32	194	2

Dataset	# feat	# obj	# class
Wine	13	178	3
Sonar	60	208	2
Soybean1	35	266	15
Chromo	8	1143	24
Vehicles	18	846	4
Malaysia	8	291	20
Imox	8	192	4
x80	8	45	3
Soybean2	35	136	4
Iris	4	150	3
Glass	9	214	6
Spirals	2	194	2

4.1 First Experiment: Classifier Comparison

All the results reported in Table 2 were computed for repeated train and test with 50 repetitions. In each repetition, data sets were split into two random equal-sized parts, one used for training and the other for testing. Classification accuracies are computed as the number of correctly classified elements in the testing set. In the second, third and fourth columns of Table 2 we reported such accuracies, together with the standard errors. In order to have a statistically robust pairwise comparison between the three methods, we performed a two-tailed t-test, at the 5% of significance, to compare the 50 repetitions of each pair of methods (namely 1NN vs. ANN, 1NN vs. HC and ANN vs. HC). This permits to judge whether the observed differences are statistically significant or not [17]. The null hypothesis was that the performances of the two examined techniques are equivalent: when it is rejected, a statistically significant difference is found. Results of the t-tests are reported in the last three columns of Table 2. In case of rejection of the null hyphothesis, a slanted arrow points to the best

Table 2. Accuracies and t-tests

Dataset	Accuracies			t-tests		
	1NN	ANN	HC	1NN vs. ANN	1NN vs. HC	ANN vs. HC
◇ German-credit	68.59 ± 0.29	70.92 ± 0.29	71.27 ± 0.29	Reject↗	Reject↗	Reject↗
◇ Pima	69.03 ± 0.33	71.91 ± 0.32	72.19 ± 0.32	Reject↗	Reject↗	Reject↗
◇ WDBC	95.27 ± 0.18	96.01 ± 0.16	96.23 ± 0.16	Reject↗	Reject↗	Reject↗
◇ Tic-Tac-Toe	79.07 ± 0.26	80.79 ± 0.25	82.51 ± 0.25	Reject↗	Reject↗	Reject↗
◇ Yeast	50.81 ± 0.26	52.87 ± 0.26	53.54 ± 0.26	Reject↗	Reject↗	Reject↗
◇ Ecoli	79.42 ± 0.44	81.95 ± 0.42	82.71 ± 0.41	Reject↗	Reject↗	Reject↗
□ Arrhythmia	56.42 ± 0.48	55.8 ± 0.48	58.07 ± 0.48	Accept	Reject↗	Reject↗
▲ Heart 2	76.48 ± 0.49	78.28 ± 0.48	78.59 ± 0.48	Reject↗	Reject↗	Accept
▲ Haberman	66.3 ± 0.54	68.39 ± 0.53	68.24 ± 0.53	Reject↗	Reject↗	Accept
▲ Ionosphere	85.27 ± 0.38	92.92 ± 0.27	92.82 ± 0.28	Reject↗	Reject↗	Accept
▲ Liver	60.35 ± 0.53	62.08 ± 0.52	62.15 ± 0.52	Reject↗	Reject↗	Accept
▲ WPBC	66.25 ± 0.68	71.11 ± 0.65	70.99 ± 0.65	Reject↗	Reject↗	Accept
▲ Wine	94.47 ± 0.34	95.37 ± 0.31	95.28 ± 0.32	Reject↗	Reject↗	Accept
♦ Sonar	82.96 ± 0.52	83.77 ± 0.51	83.75 ± 0.51	Accept	Accept	Accept
♦ Soybean1	84.24 ± 0.45	83.32 ± 0.46	83.41 ± 0.46	Accept	Accept	Accept
♦ Chromo	54.32 ± 0.29	54.06 ± 0.29	53.95 ± 0.29	Accept	Accept	Accept
△ Vehicles	68.56 ± 0.32	67.93 ± 0.32	68.0 ± 0.32	↖Reject	↖Reject	Accept
△ Malaysia	66.07 ± 0.55	64.96 ± 0.56	64.64 ± 0.56	↖Reject	↖Reject	Accept
△ Imox	91.77 ± 0.4	90.81 ± 0.42	90.67 ± 0.42	↖Reject	↖Reject	Accept
■ x80	90.17 ± 0.88	87.48 ± 0.98	87.83 ± 0.96	↖Reject	Accept	Accept
■ Soybean2	83.68 ± 0.63	82.5 ± 0.65	82.62 ± 0.65	↖Reject	Accept	Accept
▽ Iris	93.79 ± 0.39	94.32 ± 0.38	93.76 ± 0.39	Reject↗	Accept	↖Reject
▼ Glass	66.62 ± 0.64	64.99 ± 0.65	66.07 ± 0.65	↖Reject	Accept	Reject↗
★ Spirals	72.95 ± 0.64	68.76 ± 0.67	68.02 ± 0.67	↖Reject	↖Reject	↖Reject

classifier. To better clarify this notation, for example, in the "1NN vs. ANN" column, "German-credit" row, the arrow following the "Reject" indicates that the ANN rule was statistically significantly better than the 1NN rule on the German-credit dataset.

By looking at the table, different observations can be derived. According to the performances, a number of groups of data sets can be identified. The first group (denoted with ◇) corresponds to six data sets for which HC is better than ANN and both, in turn, are better than 1NN. A slightly different behavior is exhibited by Arrhytmia (denoted with □) for which there is no statistical difference between 1NN and ANN. Another large group (denoted by ▲) is composed by data sets for which there is no difference between ANN and HC but both are better than 1NN. Subsequently, we find a group of three data sets (♦) for which there is no statistical difference between the three classifiers.

Continuing with the descending reading of the table, there is a group of three data sets (△) for which 1NN is better than both ANN and HC while there is no difference between the latter. A slightly different behavior is shown by x80 and Soybean2 (denoted with ■), for which—in contrast to the previous case—HC is equivalent to 1NN. The last group (▽, ▼ and ★) contains three data sets whose results are special: Iris is the only case in which ANN is better than both 1NN and HC; for Glass, HC is better than ANN, even though the former is not significantly different than 1NN while the latter is worse than 1NN. Finally, results for the Spirals data set show an artificial case—deliberately included by us for illustration purposes, see Fig. 1—in which 1NN is significantly better in accuracy (by 4.19% and 4.93%, respectively) than ANN and HC. Notice that the spheres defined for Spirals would occupy the space between the spiral arms and their corresponding radii are the half of the width of the inter-arm corridors. Penalizations by the radius, that are so beneficial in other cases, appear to be counterproductive for this data set due to its particular configuration.

In general we can see that in many cases the correction of both HC and ANN is beneficial with respect to 1NN, but there are some other cases where this correction is not useful at all. Concerning the two techniques, the HC method seems to be slightly superior to the ANN variant. We tried to derive a relation between such classification accuracies and data set properties (in terms of dimensionality, number of classes and so on): however, it was not possible to derive many regularities, apart from the facts that (i) the behavior for Arrhythmia—the highest-dimensional data set—is special; (ii) the large group of datasets for which there is no difference between ANN and HC but both are better than 1NN is homogeneous with respect to the number of classes (five two-class problems and a three-class one) and (iii) two of the three data sets with more classes (Soybean1 and Chromo) do not exhibit any profit from the use of ANN and HC.

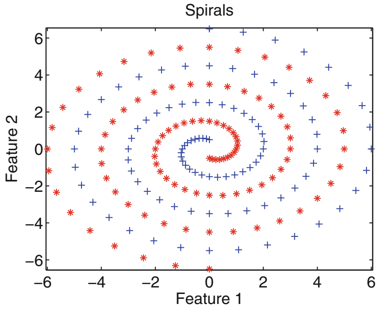

Fig. 1. Scatter plot of Spirals data set

4.2 Second Experiment: Transition Between ANN and HC

In this second experiment we tested if and how much helpful is to employ a smooth transition between ANN and HC. Actually, the penalizations of the distances implemented by these two rules have different nature, due to the two different mathematical operations involved (subtraction vs. division). Therefore, it seems reasonable to try to employ a combination of the two, as explained in Sect. 3. To test this aspect we repeated the classification experiments on the 24 datasets of before, by using the convex combinations of the two modified distances (in all the variants proposed in Sect. 3). The parameter λ has been varied from 0 (ANN rule) to 1 (HC rule) with step 0.1.

The results showed that when the HC and the ANN rules were both outperforming the 1NN rule (namely in the first fourteen data sets, from German-credit until Sonar), there are no improvements by their convex combinations, with a smooth transition between the accuracies of the two methods.

More interesting are the situations where ANN, HC or both are worse than 1NN. Such situations are shown in Fig. 2, where accuracies are shown when varying λ. In all plots, the red line represents the 1NN result, whereas the four variants defined by Eqs. (10), (11), (12) and (13) are represented by the blue, cyan, black and magenta lines, respectively.

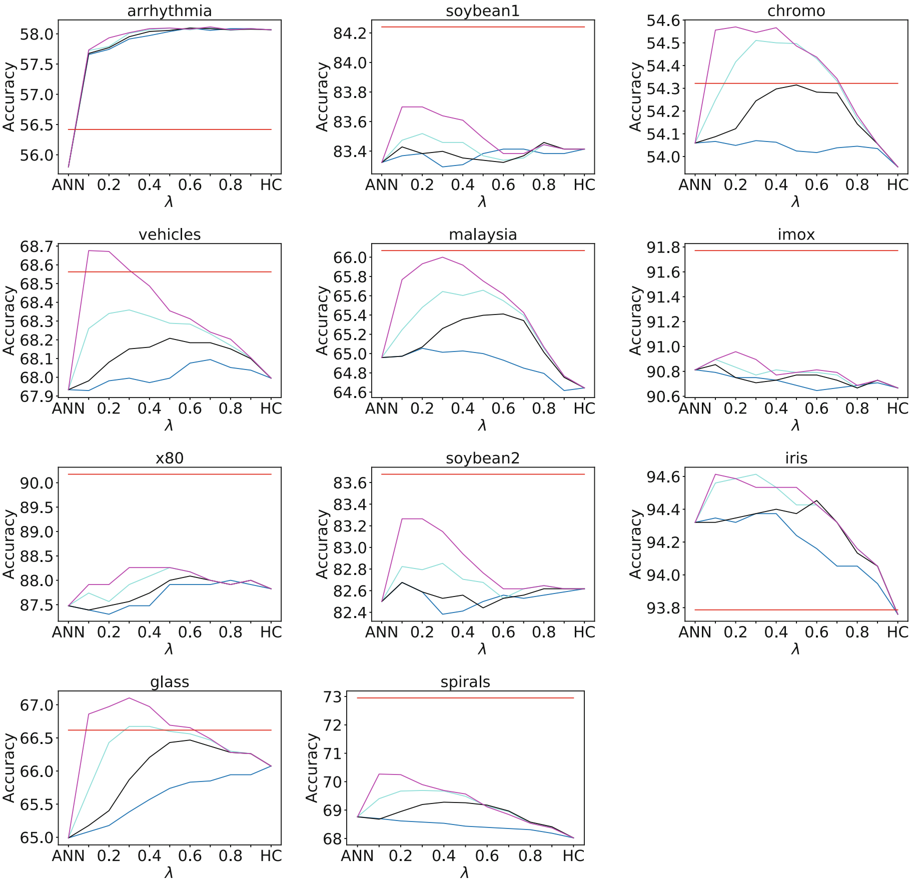

Fig. 2. (Best viewed in color) Analysis of the transition. Red line represents the 1NN result, whereas the four variants defined by Eqs. (10), (11), (12) and (13) are represented by the blue, cyan, black and magenta lines, respectively. (Color figure online)

For these data sets, it is interesting to observe that the convex combinations improve over both ANN and HC, except in the `Arrhytmia` case. Equation 13 is consistently the best for all these cases. Notice, in addition, that in four out of eleven occasions—for `Chromo`, `Vehicle`, `Iris` and `Glass`—at least one of the convex combinations outperforms 1NN for either some values of λ or all its range (cf. `Iris`). This represents a valuable result, since it supports the idea that the combination can be really useful when ANN and HC both fail. Concerning the parameter λ, we observed that in general the best value lies in the interval $[0.1, 0.3]$.

A ranking of the variants of the convex combinations, according to their effects, is clearly observed in some of the subfigures; see, for instance, results for `Chromo`, `Vehicles`, and `Malaysia`. In such cases, the sequence of the variants, starting from the best one, is: Eqs. (13), (11), (12) and (10).

5 Conclusion

In this paper we presented an empirical comparison and analysis of two related techniques, namely the Adaptive Nearest Neighbor Rule and the Hypersphere Classifier. Both approaches improve 1NN by correcting the distance query-prototype with information related to the distance of the prototype to the other classes, the difference consists in the way such correction is implemented. The relation between them is that 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. We also performed a thorough experimental comparison between the two methods, also investigating how to integrate them via convex combinations.

Results lead us to conclude that HC, overall, should be preferred over ANN. However, since ANN does not yield negative distances, it might be considered as a processing step to apply, afterwards, alternative decision rules that are not necessarily based on the smallest dissimilarity values. We also showed that the convex combination of the two approaches is useful when both methods are worse than the original 1-Nearest Neighbor. In these cases, in general, $0.1 \leq \lambda \leq 0.3$ seems to be a convenient interval to select the parameter for the convex combination. Its proper tuning, however, is a matter for further study.

Acknowledgements. Part of this work was done while the first author was visiting the Computer Science Department at Università degli Studi di Verona, Italy, during his Sabbatical Year 2017–2018 granted by Universidad Nacional de Colombia - Sede Manizales.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2001)
2. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, Boston (1990)
3. Pekalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recogn. **39**(2), 189–208 (2006). Part Special Issue: Complexity Reduction
4. Triguero, I., Derrac, J., García, S., Herrera, F.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **42**(1), 86–100 (2012)

5. Pal, A.K., Mondal, P.K., Ghosh, A.K.: High dimensional nearest neighbor classification based on mean absolute differences of inter-point distances. Pattern Recogn. Lett. **74**, 1–8 (2016)
6. Lopes, N., Ribeiro, B.: Incremental hypersphere classifier (IHC). Machine Learning for Adaptive Many-Core Machines - A Practical Approach. SBD, vol. 7, pp. 107–123. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-06938-8_6
7. Wang, J., Neskovic, P., Cooper, L.N.: Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recogn. Lett. **28**(2), 207–213 (2007)
8. Mukherjee, J.: Linear combination of norms in improving approximation of Euclidean norm. Pattern Recogn. Lett. **34**(12), 1348–1355 (2013)
9. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12**, 2211–2268 (2011)
10. Joachims, T., Cristianini, N., Shawe-Taylor, J.: Composite kernels for hypertext categorisation. In: Brodley, C.E., Danyluk, A.P. (eds.) Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williamstown, MA, USA, pp. 250–257. Morgan Kaufmann, June 2001
11. Duin, R.P.W.: Compactness and complexity of pattern recognition problems. In: Perneel, C. (ed.) Proceedings of the International Symposium on Pattern Recognition "In Memoriam Pierre Devijver", Brussels, Belgium, Royal Military Academy, pp. 124–128, February 1999
12. Orozco-Alzate, M., Duin, R.P.W., Bicego, M.: Unsupervised parameter estimation of non linear scaling for improved classification in the dissimilarity space. In: Robles-Kelly, A., Loog, M., Biggio, B., Escolano, F., Wilson, R. (eds.) S+SSPR 2016. LNCS, vol. 10029, pp. 74–83. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49055-7_7
13. Duin, R.P.W., Bicego, M., Orozco-Alzate, M., Kim, S.-W., Loog, M.: Metric learning in dissimilarity space for improved nearest neighbor performance. In: Fränti, P., Brown, G., Loog, M., Escolano, F., Pelillo, M. (eds.) S+SSPR 2014. LNCS, vol. 8621, pp. 183–192. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44415-3_19
14. Bicego, M., Baldo, S.: Properties of the Box-Cox transformation for pattern classification. Neurocomputing **218**, 390–400 (2016)
15. Sakia, R.M.: The Box-Cox transformation technique: a review. J. Roy. Stat. Soc. Ser. D (The Statistician) **41**(2), 169–178 (1992)
16. Box, G.E.P., Cox, D.R.: An analysis of transformations. J. Roy. Stat. Soc.: Ser. B (Methodol.) **26**(2), 211–243 (1964)
17. Bramer, M.: 15: comparing classifiers. In: Principles of Data Mining. Undergraduate Topics in Computer Science, 2nd edn, pp. 221–236. Springer, London (2013). https://doi.org/10.1007/978-1-4471-7307-6_15