# Metadata of the chapter that will be visualized in SpringerLink


| | | |
|---|---|---|
| Book Title | Pattern Recognition and Image Analysis | |
| Series Title | | |
| Chapter Title | On the Impact of Distance Metrics in Instance-Based Learning Algorithms | |
| Copyright Year | 2015 | |
| Copyright HolderName | Springer International Publishing Switzerland | |



| | | |
|---|---|---|
| Corresponding Author | Family Name | **Lopes** |
| | Particle | |
| | Given Name | **Noel** |
| | Prefix | |
| | Suffix | |
| | Division | |
| | Organization | UDI, Polytechnic of Guarda |
| | Address | Guarda, Portugal |
| | Division | Department of Informatics Engineering, CISUC |
| | Organization | University of Coimbra |
| | Address | Coimbra, Portugal |
| | Email | noel@ipg.pt |
| Author | Family Name | **Ribeiro** |
| | Particle | |
| | Given Name | **Bernardete** |
| | Prefix | |
| | Suffix | |
| | Division | Department of Informatics Engineering, CISUC |
| | Organization | University of Coimbra |
| | Address | Coimbra, Portugal |
| | Email | bribeiro@dei.uc.pt |



| | |
|---|---|
| Abstract | In this paper we analyze the impact of distinct distance metrics in instance-based learning algorithms. In particular, we look at the well-known 1-Nearest Neighbor (NN) algorithm and the Incremental Hypersphere Classifier (IHC) algorithm, which proved to be efficient in large-scale recognition problems and online learning. We provide a detailed empirical evaluation on fifteen datasets with several sizes and dimensionality. We then statistically show that the Euclidean and Manhattan metrics significantly yield good results in a wide range of problems. However, grid-search like methods are often desirable to determine the best matching metric depending on the problem and algorithm. |



| | |
|---|---|
| Keywords (separated by '-') | Distance metrics - Instance-based learning - Incremental learning - Nearest Neighbor - Incremental Hypersphere Classifier (IHC) |

# On the Impact of Distance Metrics in Instance-Based Learning Algorithms

Noel Lopes[1,2](✉) and Bernardete Ribeiro[2]

[1] UDI, Polytechnic of Guarda, Guarda, Portugal
[2] Department of Informatics Engineering, CISUC, University of Coimbra, Coimbra, Portugal
noel@ipg.pt, bribeiro@dei.uc.pt

**Abstract.** In this paper we analyze the impact of distinct distance metrics in instance-based learning algorithms. In particular, we look at the well-known 1-Nearest Neighbor (NN) algorithm and the Incremental Hypersphere Classifier (IHC) algorithm, which proved to be efficient in large-scale recognition problems and online learning. We provide a detailed empirical evaluation on fifteen datasets with several sizes and dimensionality. We then statistically show that the Euclidean and Manhattan metrics significantly yield good results in a wide range of problems. However, grid-search like methods are often desirable to determine the best matching metric depending on the problem and algorithm.

**Keywords:** Distance metrics · Instance-based learning · Incremental learning · Nearest Neighbor · Incremental Hypersphere Classifier (IHC)

## 1 Introduction

Incremental learning algorithms embody the potential to deal with large scale datasets and data streams. Rather than requiring access to the complete dataset, they adjust their models continuously with upcoming data. One of such algorithms, recently proposed, is the Incremental Hypersphere Classifier (IHC), which possesses desirable characteristics in terms of multi-class support, complexity, scalability, interpretability and potential to handle concept drifts [8,9]. Moreover, it has been successfully used as an instance selection method for choosing a representative subset of the data that was later used to derive improved batch models [7].

Despite these advantages, IHC is a distance based learning method and naturally it is sensitive to the choice of the distance metric. In this context, in this paper, we analyze the impact of distinct distance metrics in both the 1-NN and the IHC algorithms. The reason to analyze the effects of the distance metrics also in the 1-NN is because we can look at IHC as a generalization of the former.

The remainder of this paper is organized as follows. The next section details the IHC algorithm and Sect. 3 describes the metrics that were analyzed in this study. Section 4 presents the experimental results and finally, in Sect. 5 the conclusions and future work are delineated.


R. Paredes et al. (Eds.): IbPRIA 2015, LNCS 9117, pp. 1–9, 2015.
DOI: 10.1007/978-3-319-19390-8_6

## 2 Incremental Hypersphere Classifier (IHC) Algorithm

Let us consider a training dataset, $\{(\mathbf{x_i}, y_i) : i = 1, \ldots, N\}$, composed by $N$ samples, each encompassing an input vector, $\mathbf{x_i} \in \mathbb{R}^D$, with $D$ features, and the associated class label, $y_i \in \{1, \ldots, C\}$, where $C$ is the number of classes.

For each training sample, $i$, IHC defines an hypersphere with center $\mathbf{x_i}$ and radius $\rho_i$ as follows:

$$\rho_i = \frac{\min(d(\mathbf{x_i}, \mathbf{x_j}))}{2}, \text{ for all } j \text{ where } y_j \neq y_i \tag{1}$$

where $d(\mathbf{x_i}, \mathbf{x_j})$ is the distance between $\mathbf{x_i}$ and $\mathbf{x_j}$ input vectors (see Sect. 3). The hypersphere's delineate the regions of influence of the associated samples and are used to classify new instances. Basically, given a new data point, $\mathbf{x_k}$, it is classified with the class associated to the nearest hypersphere (not the nearest sample). More precisely, $\mathbf{x_k}$ is associated to class $y_i$ (i.e. $y_k = y_i$) provided that:

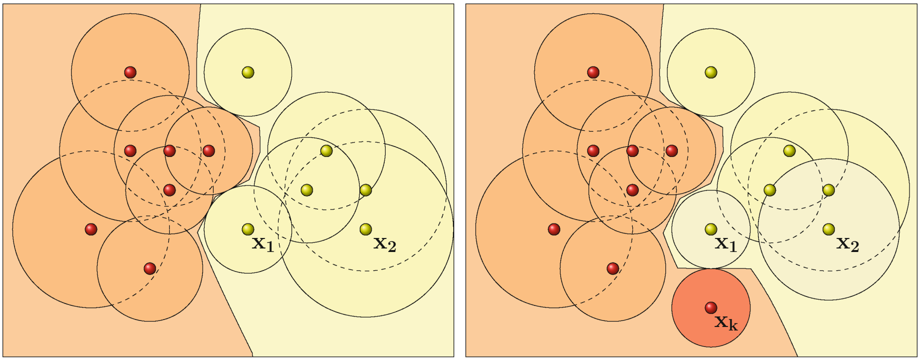
$$d(\mathbf{x_i}, \mathbf{x_k}) - g a_i \rho_i \leq d(\mathbf{x_j}, \mathbf{x_k}) - g a_j \rho_j, \text{ for all } j \neq i \tag{2}$$

where $g$ (gravity) controls the extension of the zones of influence, increasing or shrinking them and $a_i$ is the accuracy of sample $i$ when classifying itself and the forgotten training samples for which $i$ was the nearest sample in memory. A forgotten sample is one that either has been removed from memory or did not qualify to enter the memory in the first place. Hence, the accuracy is only updated when the memory is full. In such a scenario, at each iteration, the accuracy of a single (nearest) sample is updated, while the accuracy of all the others remains unchanged. The accuracy is the first mechanism of defense against outliers, reducing effectively their influence in the model.

Notice that for $g = 0$ the decision rule of the IHC is exactly the same as the one of the 1-NN (see Eq. 2). Hence, by fine-tuning $g$, IHC will always yield better or equal performance than 1-NN. This is important because Cover and Hart [3] demonstrated, in the limit $N \rightarrow \infty$, that the 1-NN error rate is never more than twice the minimum achievable error rate of an optimal classifier [2].

A major advantage of the IHC algorithm relies on the possibility of building models incrementally on a sample-by-sample basis. Figure 1 presents the hypersphere's generated by IHC and the resulting decision surface, (a) prior to and (b) after the addition of a new sample, for a toy problem. Notice that the samples near the decision border have smaller radius than those furthest, providing a simple method for determining the relevance of each sample. Hence, when the memory is full, the samples with smaller radius are kept, while those with bigger radius are discarded. By doing so, we keep the samples that play the most significant role in the construction of the decision surface (given the available memory) while removing those that have less or no impact in the model.

Unfortunately, outliers will most likely have a small radius and end-up occupying our limited memory resources. Thus, although their impact is diminished by the use of the accuracy in Eq. 2, it is still important to identify and remove them from memory. To address this problem IHC mimics the process used by the




(a) Before adding sample $k$. (b) After adding a new sample, $k$.

**Fig. 1.** Hypersphere's and decision surface generated by IHC ($g = 1$) for a toy problem.

IB3 algorithm [1,11], which consists of removing all samples that are believed to be noisy by employing a significance test.

A more detailed description of the IHC can be found elsewhere [8,9] and a working version of the algorithm, including its source code, can be found at http://sourceforge.net/projects/ihclassifier/.

## 3 Distance Metrics

A distance metric is a function that measures the similarity between two vectors, $\mathbf{x_i} = [x_{i1}, x_{i2}, \ldots, x_{iD}]$ and $\mathbf{x_j} = [x_{j1}, x_{j2}, \ldots, x_{jD}]$, yielding a non-negative real number, representing the degree of discrepancy between the two data points.

Although a large number of distance metrics have been proposed in the literature, the most widely used and well known metric is still the Euclidean distance, stated by Euclid more than two thousand years ago. Another extensively used metric, is the Manhattan, also known as the city-block distance [6].

Table 1 presents the distance metrics used in this study. For the Minkowsky metric, $p$ was set to the number of features, $D$, in order to give more weight to the individual distance components as the space dimensionality increases [5].

## 4 Experimental Results

Our goal consists of analyzing the impact of distinct distance metrics in both the 1-NN and the IHC algorithms. With that purpose in mind, we carried out extensive experiments on fifteen UCI databases [4] with distinct characteristics (number of samples, features and classes). For statistical significance, each experiment was executed using repeated 5-fold stratified cross-validation. Altogether 30 different random cross-validation partitions were created, accounting for a total of 150 runs per benchmark and algorithm settings. The experiments were conducted using the 1-NN and the IHC algorithm with both $g = 1$ and $g = 2$

**Table 1.** Distance metrics' formulas.

| Metric | Formula |
|---|---|
| Euclidean | $d(\mathbf{x_i}, \mathbf{x_j}) = \left(\sum_{k=1}^{D} (x_{ik} - x_{jk})^2\right)^{\frac{1}{2}}$ |
| Manhattan | $d(\mathbf{x_i}, \mathbf{x_j}) = \sum_{k=1}^{D} \lvert x_{ik} - x_{jk}\rvert$ |
| Canberra | $d(\mathbf{x_i}, \mathbf{x_j}) = \sum_{k=1}^{D} \frac{\lvert x_{ik} - x_{jk}\rvert}{\lvert x_{ik}\rvert + \lvert x_{jk}\rvert}$ |
| Chebychev | $d(\mathbf{x_i}, \mathbf{x_j}) = \max(\lvert x_{ik} - x_{jk}\rvert)$ |
| Minkowsky | $d(\mathbf{x_i}, \mathbf{x_j}) = \left(\sum_{k=1}^{D} \lvert x_{ik} - x_{jk}\rvert^p\right)^{\frac{1}{p}}$ |

**Table 2.** Best distance metric depending on the database and chosen algorithm.

| Database | Samples | Inputs | Classes | 1-NN | IHC ($g = 1$) | IHC ($g = 2$) |
|---|---|---|---|---|---|---|
| Balance | 500 | 4 | 3 | Chebychev | Chebychev | Canberra |
| Breast cancer | 569 | 30 | 2 | Manhattan | Manhattan | Euclidean |
| Ecoli | 336 | 7 | 8 | Euclidean | Minkowsky | Minkowsky |
| German | 1000 | 59 | 2 | Euclidean | Canberra | Manhattan |
| Glass | 214 | 9 | 6 | Manhattan | Manhattan | Manhattan |
| Haberman | 306 | 3 | 2 | Minkowsky | Euclidean | Euclidean |
| Heart-statlog | 270 | 20 | 2 | Canberra | Canberra | Canberra |
| Ionosphere | 351 | 34 | 2 | Manhattan | Chebychev | Chebychev |
| Iris | 150 | 4 | 3 | Chebychev | Minkowsky | Chebychev |
| Pima | 768 | 8 | 2 | Euclidean | Minkowsky | Euclidean |
| Sonar | 208 | 60 | 2 | Manhattan | Manhattan | Euclidean |
| Tic-tac-toe | 958 | 9 | 2 | Canberra | Euclidean | Minkowsky |
| Vehicle | 946 | 18 | 4 | Euclidean | Euclidean | Euclidean |
| Wine | 178 | 13 | 3 | Manhattan | Manhattan | Manhattan |
| Yeast | 1484 | 8 | 10 | Manhattan | Euclidean | Euclidean |

settings. Table 2 presents the main characteristics of the experimental databases as well as the best distance metric for each algorithm. Moreover, Fig. 2 presents the results for the 1-NN and Figs. 3 and 4 the results for the IHC algorithm, using respectively $g = 1$ and $g = 2$ settings. In addition, Fig. 5 reports the average F-score results for each distance metric. Note that, with the exception of the Iris problem, the best results were obtained with the IHC algorithm.

Using the Wilcoxon signed rank test, the null hypothesis of the 1-NN having an equal or better F-score than the IHC (considering $g = 1$) is rejected at a significance level of 0.005 for the Euclidean, Manhattan, Canberra and Minkowsky distance metrics and rejected at a significance level of 0.01 for the Chebychev

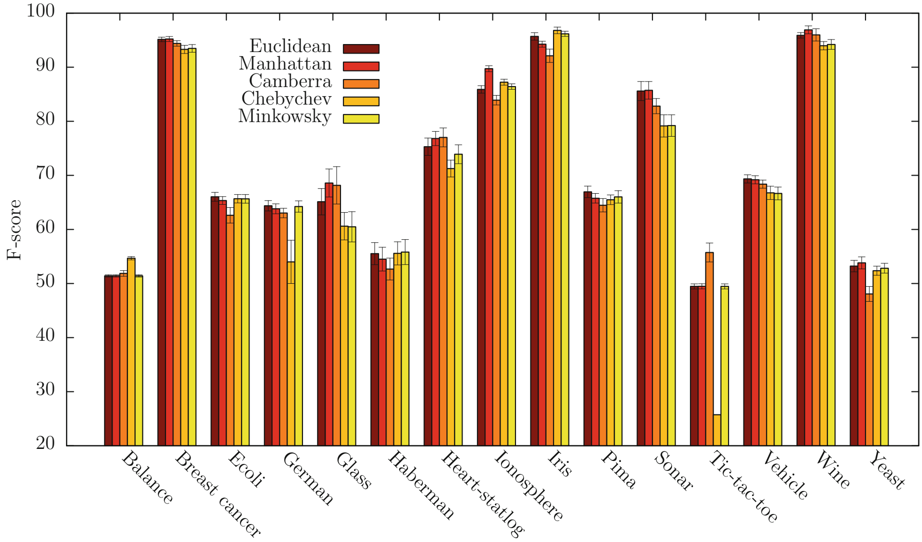



**Fig. 2.** Benchmark results for the 1-NN, according to the distance metric used.

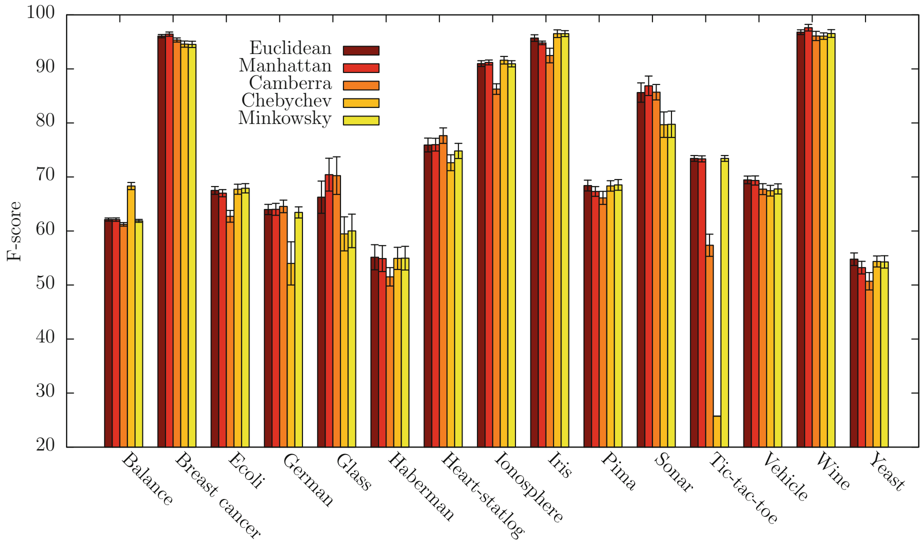



**Fig. 3.** Benchmark results for the IHC ($g = 1$), according to the distance metric used.

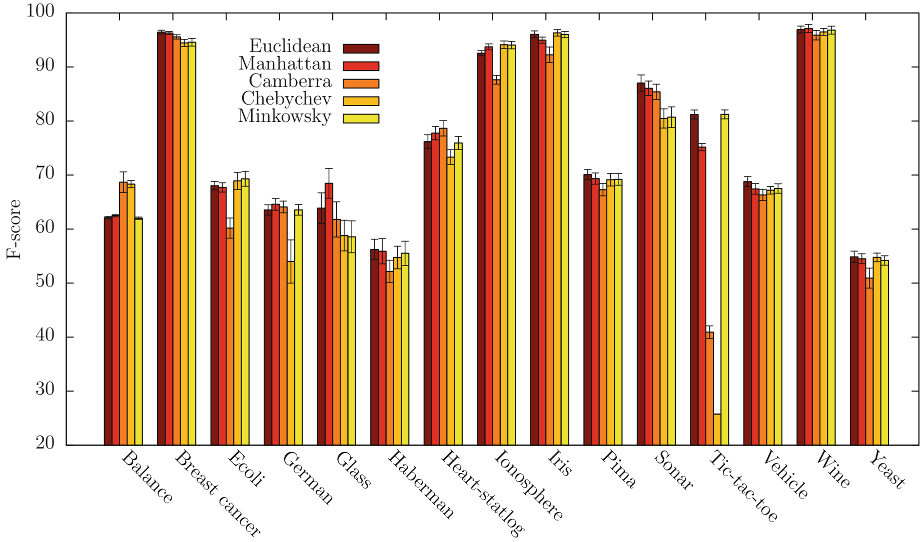



**Fig. 4.** Benchmark results for the IHC ($g = 2$), according to the distance metric used.

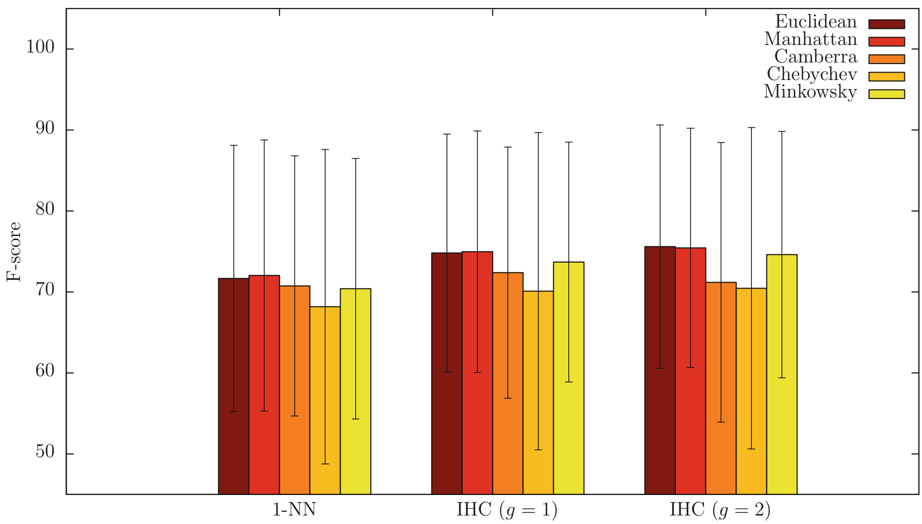



**Fig. 5.** Average performance for the 1-NN and IHC algorithms, according to the distance metric used.

metric. The same holds true when setting $g = 2$, except for the Canberra metric. Thus, these results corroborate the ones in Lopes and Ribeiro [8] and strongly evidence that the IHC significantly outperforms the 1-NN independently of the distance metric used (see Figs. 2, 3, 4 and 5). On average the Chebychev distance metric yielded the poorest results, both for the 1-NN and for the IHC algorithms (see Fig. 5). In particular, concerning the Tic-tac-toe problem, its performance is very poor (25.74 %) for all of the algorithms analyzed. However, considering the Ionosphere and Iris databases, this metric actually performed better than the remaining ones in most of the cases (see Table 2 and Figs. 2, 3 and 4). Excluding the Chebychev metric, in general Canberra yielded, on average, the worst results as compared to the remaining metrics (see Fig. 5). Nevertheless, it consistently yielded the best results for the Heart-statlog problem (see Table 2, and Figs. 2, 3 and 4). Minkowsky performs, on average, better than the Chebychev and Canberra and therefore it appears to be a better choice than these distance metrics (see Fig. 5), although there is no compelling statistical evidence to support this decision. This metric performed particularly well on the Ecoli problem (see Table 2 and Figs. 2, 3 and 4). The average performance of Manhattan and Euclidean is similar, with slightly advantage for the Manhattan distance metric, concerning the 1-NN and IHC with $g = 2$ (respectively +0.37 % and +0.16 %) and slightly advantage for the Euclidean metric, considering the IHC algorithm with $g = 1$. In the case of the NN algorithm, the results confirm the findings of Salzberg [10], which suggested that the differences between these two metrics were not significant, from the point of view of the NN algorithm. Notwithstanding, the Manhattan performs particularly well on the Glass problem regardless of the algorithm and settings considered (see Figs. 2, 3 and 4). Overall, the performance of these two distance metrics is usually superior to the remaining ones. In fact, in general, there is statistical evidence compelling the choice of the Manhattan and Euclidean distance metrics over the other ones. Using the Wilcoxon signed rank test, the null hypothesis of Chebychev having an equal or better F-score than the Euclidean metric is rejected at a significance level of 0.025 for the 1-NN and at a significance level of 0.05 for the IHC algorithm. Moreover, the null hypothesis of Chebychev having an equal or better F-score than Manhattan is rejected at a significance level of 0.025 for the 1-NN. Concerning the Canberra distance metric, the null hypothesis of Canberra having an equal or better F-score than the Euclidean is rejected at a significance level of 0.025 for the IHC algorithm. In addition the null hypothesis of Canberra having an equal or better F-score than Manhattan is rejected at a significance level of 0.01 both for the NN and IHC algorithms (0.005 for $g = 1$). Finally, concerning the Minkowsky distance metric, the null hypothesis of Minkowsky having an equal or better F-score than Euclidean is rejected at a significance level of 0.025 for the 1-NN algorithm and respectively at a significance level of 0.025 and 0.05 for the IHC algorithm using $g = 1$ and $g = 2$ settings. Moreover, the null hypothesis of Minkowsky having an equal or better F-score than Manhattan is rejected at a significance level of 0.05 for the 1-NN algorithm. Nevertheless, the No-Free-Lunch theorem [12] still applies and using the appropriate distance metric is fundamental for improving the generalization capabilities of distance based Machine Learning (ML) algorithms. Therefore, performing a grid search with the distance metric and $g$ parameters (in the case of the IHC), using the training data, is vital to enhance the algorithms' generalization capabilities.

## 5 Conclusions and Future Work

The distance metric is a pivotal parameter of distance based ML algorithms and models. The empirical results, obtained in this paper, evidence that the best metric depends on the problems' data distribution (see Table 2) and therefore grid-search like methods are crucial to potentially determine the most-advantageous metric for a given problem and algorithm. This study also demonstrates that the Euclidean and Manhattan, two of the most commonly used distance metrics, which consistently yield good results over a wide range of problems (see Figs. 2, 3 and 4), are probably the best choices for distance based learning methods when performing a grid-search method is not a viable option. In this scenario, the Manhattan distance is preferred, in particular for large datasets, since it is computationally less demanding. Future work will analyze combining different distance metrics as well as building ensembles using distinct distance metrics.

**Acknowledgments.** This work is partly funded by iCIS (CENTRO-07-ST24-FEDER-002003).

## References

1. Aha, D., Kibler, D., Albert, M.: Instance-based learning algorithms. Mach. Learn. **6**(1), 37–66 (1991)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theor. **13**(1), 21–27 (1967)
4. Frank, A., Asuncion, A.: UCI machine learning repository. http://archive.ics.uci.edu/ml
5. de Geer, J.P.V.: Some aspects of Minkowski distance. Department of Data Theory, Leiden University (1995)
6. Hassanat, A.B.A.: Dimensionality invariant similarity measure. J. Am. Sci. **10**(8), 221–226 (2014)
7. Lopes, N., Correia, D., Pereira, C., Ribeiro, B., Dourado, A.: An incremental hypersphere learning framework for protein membership prediction. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part III. LNCS, vol. 7208, pp. 429–439. Springer, Heidelberg (2012)
8. Lopes, N., Ribeiro, B.: An incremental class boundary preserving hypersphere classifier. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part II. LNCS, vol. 7063, pp. 690–699. Springer, Heidelberg (2011)
9. Lopes, N., Ribeiro, B.: Adaptive many-core machines. In: Lopes, N., Ribeiro, B. (eds.) Machine Learning for Adaptive Many-Core Machines – A Practical Approach. SBD, vol. 7, pp. 189–200. Springer, Heidelberg (2015)

10. Salzberg, S.: Distance metrics for instance-based learning. In: Ras, Z., Zemankova, M. (eds.) Methodologies for Intelligent Systems. LNCS, vol. 542, pp. 399–408. Springer, Heidelberg (1991)
11. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Mach. Learn. **38**(3), 257–286 (2000)
12. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural Comput. **8**(7), 1341–1390 (1996)

# MARKED PROOF

## Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

| *Instruction to printer* | *Textual mark* | *Marginal mark* |
|---|---|---|
| Leave unchanged | • • • under matter to remain | |
| Insert in text the matter indicated in the margin | | New matter followed by or |
| Delete | through single character, rule or underline or through all characters to be deleted | or |
| Substitute character or substitute part of one or more word(s) | through letter or through characters | new character or new characters |
| Change to italics | under matter to be changed | |
| Change to capitals | under matter to be changed | |
| Change to small capitals | under matter to be changed | |
| Change to bold type | under matter to be changed | |
| Change to bold italic | under matter to be changed | |
| Change to lower case | Encircle matter to be changed | |
| Change italic to upright type | (As above) | |
| Change bold to non-bold type | (As above) | |
| Insert 'superior' character | through character or where required | or under character e.g. or |
| Insert 'inferior' character | (As above) | over character e.g. |
| Insert full stop | (As above) | ⊙ |
| Insert comma | (As above) | , |
| Insert single quotation marks | (As above) | or and/or or |
| Insert double quotation marks | (As above) | or and/or or |
| Insert hyphen | (As above) | |
| Start new paragraph | | |
| No new paragraph | | |
| Transpose | | |
| Close up | linking characters | |
| Insert or substitute space between characters or words | through character or where required | |
| Reduce space between characters or words | between characters or words affected | |