Trading off Distance Metrics vs Accuracy in Incremental Learning Algorithms

Noel Lopes[1,2] and Bernardete Ribeiro[2]

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

Abstract. With the growth and development of data, the empirical evidence supporting a link between the distance metrics that are used in the instance-based algorithms and generalization has been mounting. In this paper, we look at distinct similarity measures to study its impact on the performance accuracy of incremental instance-based algorithms in pattern recognition problems. An in-depth analysis of the results of the proposed study for a variety of classification tasks (binary and multi-way) from various different domains shines light on the trade off between the distance metrics and yielded accuracy.

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

1 Introduction

In recent years there has been much interest in incremental learning algorithms, mainly due to their potential to deal with large scale datasets and data streams. Contrasting with batch learning algorithms, commonly designed with the emphasis on effectiveness (e.g. classification performance) and under the assumptions that data is static and its volume manageable, incremental algorithms are typically designed with emphasis on efficiency (e.g. time required to produce a model) [11]. Rather than requiring access to the complete dataset, incremental algorithms are designed to rapidly update their models to incorporate new information on a sample-by-sample basis and therefore suitable for high-throughput.

In previous work we presented a novel incremental instance-based learning algorithm which presents good properties in terms of multi-class support, complexity, scalability and interpretability. The algorithm named Incremental Hypersphere Classifier (IHC) algorithm [6] is extremely versatile and highly-scalable, being able to accommodate memory and computational restrictions, while creating the best possible model with the amount of given resources. Moreover, since the algorithm's execution time grows linearly with the number of samples stored in the memory, creating adaptive models and extracting information in real-time from large-scale datasets and data streams is feasible.

Experimental results, using well-known datasets, demonstrated that the IHC is able to handle concept drifts scenarios, while maintaining superior classification performance. Additionally, the resulting models are interpretable, making

Table 1. Distance metrics' formulas. Note that Euclidean, Manhattan and Chebychev are special cases of Minkowsky, obtained respectively for $p = 2$, $p = 1$ and $p \to \infty$.

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	$d(\mathbf{x_i}, \mathbf{x_j}) = \left(\sum_{k=1}^{D} \lvert x_{ik} - x_{jk} \rvert^p \right)^{\frac{1}{p}}$

this algorithm useful even in domains where interpretability is a key factor. Finally, since the IHC keeps samples that are at the odds of lying on the decision boundary while removing the noisy and less relevant ones, it represents a good choice for selecting a representative subset of the data for applying more sophisticated algorithms in a fraction of the time required for the complete dataset [7].

Despite these advantages, IHC is a distance based learning method and naturally sensitive to the choice of distance metrics. Therefore it is important to study their impact on IHC performance, in particular concerning incremental learning scenarios. Accordingly, in this paper we analyze the impact of distinct distance metrics in the 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.

The remainder of this paper is organized as follows. The next Section introduces the IHC algorithm. Section 3 presents and discusses the experimental results. Finally, in Section 4 the conclusions and future work are addressed.

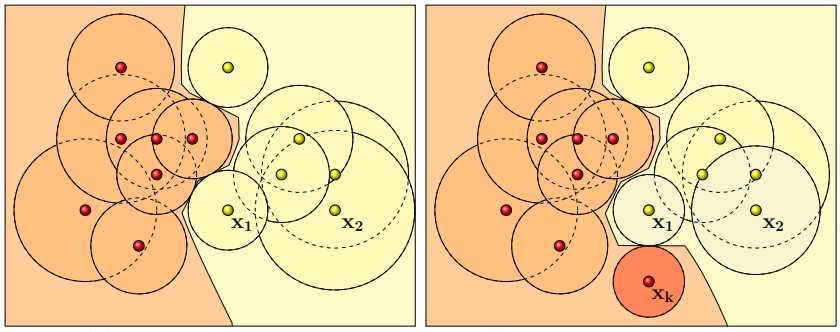
2 Incremental Hypersphere Classifier (IHC) algorithm

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 sample, i, IHC defines an hypersphere with center $\mathbf{x_i}$ and radius ρ_i:

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

where $d(\mathbf{x_i}, \mathbf{x_j})$ is the distance between $\mathbf{x_i}$ and $\mathbf{x_j}$ input vectors. 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 [4].

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



(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.

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 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.

Note that for $g = 0$ the decision rule of the IHC is exactly the same as the one of the 1-Nearest Neighbor (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 that for $N \to \infty$, 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 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. Note that adding a new sample might affect the radius of samples already in the model (in this case the ones with input vectors $\mathbf{x_1}$ and $\mathbf{x_2}$). Notice also that samples near the decision border have smaller radius than those far away (see Figure 1). Hence, when the memory is full, samples with smaller radius – that play the most significant role in the construction of the decision surface – are kept, while those with bigger radius – that have less or no impact in the model – are discarded. Unfortunately, outliers will most likely have a small radius and end-up occupying the limited memory resources. Thus, although their impact is diminished by the accuracy variable 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 IB3 algorithm [9, 1], which uses a significance test to remove all samples that are believed to be noisy.

Another advantage of IHC is that it can accommodate restrictions in terms of memory and computational power, creating the best model possible for the amount of resources given, instead of requiring systems to comply with its own

requirements. Since we can control the amount of memory and computational power required by the algorithm and due to its scalability creating up-to-date models in real-time is feasible [7]. A more detailed description of the IHC can be found elsewhere [6, 7] and a working version of the algorithm, including its source code, can be found at `http://sourceforge.net/projects/ihclassifier/`.

3 Experimental Results

Our goal consists of determining the impact of distance metrics in the IHC classification performance. Recently, we have analyzed the impact of distance metrics in batch scenarios, for both the NN and IHC algorithms [8]. Among the conclusions, we have found that the No-Free-Lunch theorem [10] still applies and the best distance metric is problem dependent. Accordingly, in batch learning configurations, it is desirable to perform a grid search both for the distance metric and g parameters in order to determine favorable parameter configurations [8]. Unfortunately, in incremental scenarios, performing a grid search is not feasible and knowing beforehand which distance metrics are likely to yield quality models becomes a fundamental aspect. Moreover, typically in incremental learning configurations, IHC must work with limited memory settings, being able to store only a small fraction of the samples. Therefore, adequate distance metrics play a vital role in choosing the core samples that delineate the decision borders.

In order to analyze the performance of distance metrics in incremental learning scenarios, we carried out extensive experiments in the same fifteen UCI databases [5] that were previously investigated in [8], comprehending distinct data distributions and characteristics (see Table 2). Altogether, five distinct memory configurations were considered, allowing IHC to store approximately 20%, 40%, 60%, 80% and 100% of the training samples. 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 memory configuration. Overall, 2250 runs per benchmark (dataset) were performed. Given the large number of runs (33,750 in total) the experiments were performed only for $g = 1$. The results were compiled both for the unseen test data and for all the data (encompassing both training and test data). The latter, reflects the IHC performance on forgotten data and it is important because real-world databases often present a high-degree of redundancy with similar records being common [7]. Figure 2 presents the IHC results, obtained for the different memory settings. Note that, in general, higher memory configurations correspond to better results.

On average Euclidean and Manhattan metrics present the best performance results for most memory settings (except for the 100% memory configuration, in which case Canberra yields better results for the test data). In fact, there is strong statistical evidence compelling the choice of these two distance metrics (see Table 3). Overall, these two metrics attained competitive and in many cases top classification performance results for most benchmarks (Breast cancer, Ecoli, German, Heart-statlog, Ionosphere, Pima, Sonar, Vehicle, Wine and Yeast).

Table 2. Experimental dataset characteristics.

Database	Samples	Inputs	Classes
Balance	625	4	3
Breast cancer	569	30	2
Ecoli	336	7	8
German	1000	59	2
Glass	214	9	6
Haberman	306	3	2
Heart-statlog	270	20	2
Ionosphere	351	34	2
Iris	150	4	3
Pima	768	8	2
Sonar	208	60	2
Tic-tac-toe	958	9	2
Vehicle	946	18	4
Wine	178	13	3
Yeast	1484	8	10

Moreover, Manhattan also attained good results in the Glass dataset, achieving top results for the 20% and 100% memory configurations. In the remaining configurations, Canberra yielded the highest F-Scores. Additionally, Manhattan outperforms all other metrics for the Breast cancer and wine datasets.

Concerning performance in the individual datasets, for the Vehicle dataset both Manhattan and Euclidean yield superior classification performance, with Manhattan presenting better results when less memory is available. These two metrics also present good results in the Sonar dataset, with the Manhattan attaining the top performance for the 20%, 40% and 80% memory configurations and Euclidean and Canberra yielding the highest results respectively for the 60% and 40% configurations. In the German dataset, overall both Manhattan and Euclidean attained competitive results. Moreover, Manhattan achieved the top results on the test data for memory configurations of 20%, 40% and 60%, while the highest results for 80% and 100% were yielded by Canberra. Interestingly, despite Chebychev yielding the worst results for the test datasets, this metric attained some of the best results when considering all data, evidencing that the model is overfitting the training data. Concerning the Heart-statlog dataset, with the exception of the 100% memory configuration, once again both Manhattan and Euclidean attained competitive results. Moreover, Manhattan yielded top results on the test dataset for the 20% and 40% memory configurations, while Canberra attained the top results for the remaining configurations. Manhattan and Euclidean also yielded good results for the Ecoli dataset, with the highest F-Score being obtained by Euclidean for the 20% memory setting, by Manhattan for the 40% memory configuration and by Minkowsky for the remaining configurations. With respect to the Haberman dataset, Manhattan yielded the highest results for the 60% and 80% memory settings, while Euclidean, Minkowsky and Chebychev attained the best F-Scores respectively for the 100%, 20% and 40% configurations. Regarding the Ionosphere, Manhattan yielded the dominant classification performance in the 40%, 60% and 80% memory settings, while the 20% and 100% were attained respectively by Canberra

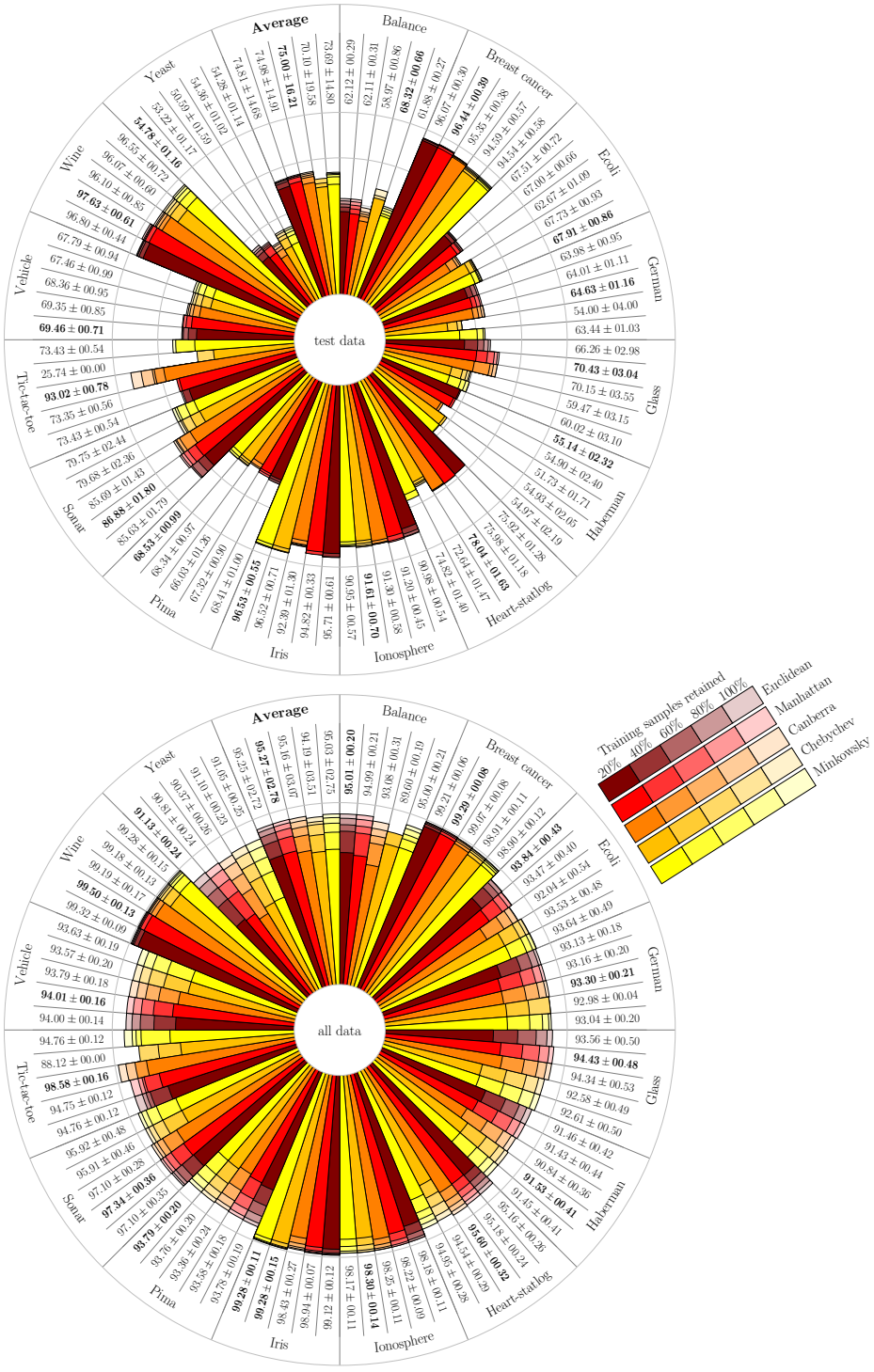


Fig. 2. IHC performance (macro-average F-Score).

Table 3. Null hypotheses (H_0 : F-Score$_{\mathbf{X}}$ >= F-Score$_{\mathbf{Y}}$) rejected using the Wilcoxon signed rank test, for the (a) test data and for (b) all data.

(a) Test data

Samples retained	Significance level	X distance metric	Y distance metric
20%	0.010	Canberra	Manhattan
	0.025	Chebychev	
	0.025	Minkowsky	
40%	0.025	Canberra	
	0.025	Chebychev	
	0.010	Minkowsky	
60%	0.025	Canberra	
	0.025	Chebychev	
	0.050	Minkowsky	
80%	0.025	Canberra	
	0.050	Chebychev	
100%	0.050	Canberra	
20%	0.025	Chebychev	Euclidean
	0.010	Minkowsky	
40%	0.025	Chebychev	
	0.005	Minkowsky	
60%	0.005	Chebychev	
	0.010	Minkowsky	
80%	0.025	Chebychev	
	0.010	Minkowsky	
100%	0.050	Chebychev	
	0.025	Minkowsky	
20%	0.010	Chebychev	Minkowsky

(b) All data

Samples retained	Significance level	X distance metric	Y distance metric
20%	0.050	Canberra	Manhattan
40%			
60%			
80%	0.025	Canberra	
	0.050	Chebychev	
100%	0.050	Canberra	
	0.050	Chebychev	
20%	0.025	Chebychev	Euclidean
	0.025	Minkowsky	
40%	0.050	Chebychev	
	0.050	Minkowsky	
80%	0.025	Chebychev	
	0.050	Minkowsky	
100%	0.005	Chebychev	
	0.005	Minkowsky	
20%	0.005	Chebychev	Minkowsky
100%	0.050	Minkowsky	

and Chebychev. The Euclidean distance metric excel all the others in the Yeast dataset. Moreover, concerning the Pima dataset, it also outperformed the other distance metrics, except in the 100% memory configuration for which Minkowsky yielded the best F-Score. The Canberra distance metric performed particularly well on the Tic-tac-toe problem, excelling by far the remaining distance metrics. Chebychev, on the other hand, yielded the worst results for this dataset and its performance significantly dropped with increase of available memory. As in the case of the German, the model overfits the training data, indicating that Chebychev-based IHC models are prone to overfitting the training data. In fact, on average Chebychev yielded the worst F-Scores for all memory settings. Nevertheless, this metric excelled all the others performance in the Balance dataset. Moreover, Chebychev also performed quite well on the Iris dataset attaining together with the Minkowsky distance metric the top classification performances.

4 Conclusions and Future Work

We are seeing a torrent of data coming in from sensors everywhere. This data is compounding daily, creating what is called "fast data". In this context, incremental algorithms are part of the solution for dealing with the data explosion that is happening at a massive scale. The big challenge is now speed and agility when building systems, in particular for dealing with anomaly detection and concept drifts that occur in many fields of science and society in general.

In this paper, we looked at the importance of distance metrics for assessment of similarity of patterns in incremental learning. To reinforce this idea, we interpreted the distance metric as a pivotal parameter for the success of many machine learning algorithms and models. We extended our previous research on the impact of distance metrics on batch learning to incremental scenarios. In the latter using grid-search like methods for determining favorable metrics is not feasible. Therefore, distance metrics play a vital role in the choice of core samples, which are expected to be representative of the whole dataset and will in practice shape the boundary decisions. To analyze the performance of distance metrics in incremental scenarios, we carried out extensive experiments using fifteen UCI databases, with distinct data distributions and characteristics. Altogether, five memory configurations were considered, allowing IHC to store approximately 20%, 40%, 60%, 80% and 100% of the samples and for statistical significance, each experiment was executed using repeated 5-fold stratified cross-validation.

This study 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 as shown in the experimental tests, 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 focus as building ensembles using distinct distance metrics.

References

1. Aha, D., Kibler, D., Albert, M.: Instance-based learning algorithms. Machine Learning 6(1), 37–66 (1991)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
3. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
4. de Geer, J.P.V.: Some Aspects of Minkowski Distance. Leiden University, Department of Data Theory (1995)
5. Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
6. Lopes, N., Ribeiro, B.: An incremental class boundary preserving hypersphere classifier. In: International Conference on Neural Information Processing (ICONIP 2011), Part II, LNCS 7063. pp. 690–699. Springer Berlin / Heidelberg (2011)
7. Lopes, N., Ribeiro, B.: Machine Learning for Adaptive Many-Core Machines – A Practical Approach, Studies in Big Data, vol. 7. Springer (2014)
8. Lopes, N., Ribeiro, B.: On the impact of distance metrics in instance-based learning algorithms. In: Pattern Recognition and Image Analysis, pp. 48–56. LNCS 9117, Springer International Publishing (2015)
9. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Machine Learning 38(3), 257–286 (2000)
10. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural Computation 8(7), 1341–1390 (1996)
11. Zhou, Z.: Three perspectives of data mining. Artificial Intelligence 143(1), 139–146 (2003)