Applying general-purpose Data Reduction Techniques for fast time series classification

Stefanos Ougiaroglou[1⋆], Leonidas Karamitopoulos[2], Christos Tatoglou[2], and Georgios Evangelidis[1]

[1] Department of Applied Informatics, University of Macedonia, 156 Egnatia St, GR-54006 Thessaloniki, Greece
{stoug, gevan}@uom.gr
[2] Information Technology Department, Alexander TEI of Thessaloniki, GR-57400 Sindos, Greece
lkaramit@it.teithe.gr, xtatty@gmail.com

Abstract. The one-nearest neighbour classifier is a widely-used time series classification method. However, its efficiency depends on the size of the training set as well as the data dimensionality. Although many speed-up methods for fast time series classification have been proposed, state-of-the-art, non-parametric data reduction techniques have not been exploited on time series data. This paper presents an experimental study where known prototype selection and abstraction data reduction techniques are evaluated on the original and a representation form of seven time series datasets. The results show that data reduction can even improve the classification accuracy and at the same time reduce the computational cost.

Keywords: time series classification, nearest neighbor, data reduction

1 Introduction

Classification methods based on similarity search have been proven to be effective approaches for time series data. More specifically, the one-Nearest Neighbour (1NN) classifier is a widely-used method. It works by assigning to an unclassified time series the class label of its most similar training time series. The main drawback of similarity-based classifiers is that all similarities between an unclassified time series item and the training time series items must be estimated. For large and high dimensional time series training sets, the high computational cost renders the application of such classifiers prohibitive. Time series classification can be sped-up using indexing, representation and/or data reduction.

Indexing can speed-up classification tremendously, but works well only in low dimensions. Thus, one must first use a dimensionality reduction technique to aquire a representation of the original data in lower dimensions. A representation may be considered as a transformation technique that maps a time series from

⋆ S. Ougiaroglou is supported by the Greek State Scholarships Foundation (I.K.Y.)

the original space to a feature space, retaining the most important features. There have been several time series representations proposed in the literature, mainly on the purpose of reducing the intrinsically high dimensionality of time series [7].

The main goal of data reduction is to reduce the computational cost of the kNN classifier and the storage requirements of the Training Set (TS). Data Reduction Techniques (DRTs) try to build a small representative set of the initial training data. This set is called the Condensing Set (CS) and has the benefits of low computational cost and storage requirements while maintaining the classification accuracy at high levels. DRTs can be divided into two algorithm categories: (i) Prototype Selection (PS) [4], and, (ii) Prototype Abstraction (PA) (or generation) [10]. Although both categories have the same motivation, they differ on the way they build the CS. PS algorithms select some TS items and use them as representatives, whereas, PA algorithms generate representative items by summarizing similar TS items.

Data reduction has been recently exploited for fast time series classification. More specifically, [2] and [11] propose PS algorithms for speeding-up 1NN time series classification. The disadvantage of these methods is that they are parametric. The user must define the CS size through trial-and-error procedures.

The present work has been motivated by the following observations. State-of-the-art non-parametric PS and PA algorithms have not been evaluated neither on original time series nor on their reduced dimensionality representations. Also, a PA algorithm we have previously proposed (RHC [8]) has not been evaluated on time series data. The contribution of this paper is the experimental evaluation of two PS algorithms, namely, CNN-rule [5] and IB2 [1], and two PA algorithms, namely, RSP3 [9] and RHC [8] on time series data. Our experimental study adopts the time series representation method Piecewise Aggregate Approximation (PAA) [6, 12] in order to test the effect the combination of data reduction on dimensionality reduced time series has on classification. PAA is a very simple dimensionality reduction technique that segments a time series into h consecutive sections of equal-width and calculates the corresponding mean for each one. The series of these means is the new representation of the original data.

The rest of the paper is organized as follows. Section 2 discusses the details of the four aforementioned DRTs. Section 3 describes the experimental study and the obtained results, and Section 4 concludes the paper.

2 Data Reduction Techniques

In this section, we present the four DRTs we use in our experimentation. All DRTs are based on a simple idea. The data items that do not define decision boundaries between classes are useless for the classification process. Therefore, they can be discarded. Thus, they try to select or generate a sufficient number of items that lie in data areas close to decision boundaries. The DRTs we deal with in this Section are non-parametric. They automatically determine the size of CS based on the level of noise and the number of classes in the data (the more

the classes, the more boundaries exist and, thus, the more items are selected or generated).

2.1 Prototype Selection algorithms

Hart's Condensing Nearest Neighbour rule (CNN-rule). CNN-rule [5] is the earliest and the best known PS algorithm. It uses two sets, S and T. Initially, a TS item is placed in S, while all the other TS items are placed in T. Then, CNN-rule tries to classify the content of T by using the 1NN classifier on the content of S. When an item is misclassified, it is considered to lie in a data area close to decision boundaries. Thus, it is transferred from T to S. The algorithm terminates when there are no transfers from T to S during a complete pass of T. The final set S constitutes the CS. The multiple passes on data ensure that the remaining items in T are correctly classified by the 1NN classifier on the CS. The algorithm is based on the following simple idea: items that are correctly classified by 1NN, are considered to lie in a central-class data area and thus, they are ignored. In contrast, items that are misclassified, are considered to lie in a close-class-border data area, and thus, they are placed in CS. The weak point of the CNN-rule is that the resulting CS depends on the order of items in TS. This means that different CSs are build by examining the same data in a different order.

IB2 algorithm. IB2 belongs to the well-known family of IBL algorithms [1]. It is based on CNN-rule. Actually, IB2 is a simple one pass variation of CNN-rule. Each TS item x is classified using 1NN on the current CS. If x is classified correctly, it is discarded. Otherwise, x is transferred to CS. Contrary to CNN-rule, IB2 does not ensure that all discarded items can be correctly classified by the final content of CS. However, since it is a one-pass algorithm, it is very fast. In addition, IB2 builds its CS incrementally. New TS items can be taken into consideration after the CS creation. Thus, IB2 is appropriate for dynamic (streaming) environments where new TS items may gradually arrive. Also, contrary to CNN-rule and many other DRTs, IB2 does not require that all TS data reside into the main memory. Therefore, it can be applied in devices whose memory is insufficient for storing all the TS data. Like CNN-rule, IB2 is a data order dependent algorithm.

2.2 Prototype Abstraction algorithms

RSP3 algorithm. The RSP3 algorithm belongs to the popular family of Reduction by Space Partitioning (RSP) algorithms [9]. This family includes three PA algorithms. All of them are based on the idea of the early PA algorithm of Chen and Jozwik (CJ) [3] that works as follows. Initially, it retrieves the most distant items A and B of TS that define its diameter. Then, it divides TS into two sets. S_A includes TS items that are closer to A, while S_B includes TS items that are closer to B. CJ selects to divide the set with the larger diameter. This

procedure continues until the number of sets becomes equal to a user defined number. In the end, for each set S, CJ averages the items that belong to the most common class in S and creates a mean item. The created mean items constitute the final CS.

RSP1 is a simple variation of CJ that for each final set creates as many mean items as the number of distinct classes in the set. RSP2 differs on how it selects the next set that will be divided. Instead of the criterion of the largest diameter, RSP2 uses the criterion of overlapping degree. RSP3 is based on the concept of homogeneity. A set is homogeneous when it includes items of only a specific class. The algorithm continues dividing the created sets until all of them became homogeneous. Considering RSP3, we observe that the algorithm generates more prototypes for the close borders data areas and less for the "central" data areas. RSP3 is the only non-parametric algorithm of the RSP family (CJ included). All these algorithms do not depend on the order of the data items in TS.

Reduction through Homogeneous Cluster (RHC) algorithm. RHC [8] is also based on the concept of homogeneity. Initially, the whole TS is considered as a non-homogeneous cluster C. RHC begins by computing a mean item for each class (class centroid) in C. Then, it applies k-means clustering on C using the class centroids as initial means. The clustering procedure builds as many clusters as the number of classes in C. The aforementioned clustering procedure is applied recursively on each non-homogeneous cluster. In the end, the centroids of the homogeneous clusters are stored in CS. By using the class centroids as initial means for the k-means clustering, the algorithm attempts to quickly find homogeneous clusters and achieve high reduction rates. RHC is independent on the order of data in TS. The results of the experimental study in [8] show that RHC achieves higher reduction rates (smaller CSs) and is faster than RSP3 and CNN-rule, while the classification accuracy remains at high levels.

3 Experimental Study

3.1 Experimental setup

The four presented DRTs were evaluated on seven time series datasets distributed by the UCR time-series classification/clustering page[3]. Table 1 summarizes the datasets used. All datasets are available in a training/testing form. We merged the training and testing parts and then we randomized the resulting datasets. No other data transformation was performed. All algorithms were coded in C and as a similarity measure we used the Euclidean distance.

We report on the experiment we contacted with a certain value for the parameter of the PAA representation due to space restrictions. We applied the PAA representation on time series by setting the number of dimensions equal to twelve (h=12). Most of the research work provides experimental results with

[3] http://www.cs.ucr.edu/∼eamonn/time_series_data/

Table 1: Time-series datasets description

Time-series dataset	**Size (time-series)**	**Length (Attr.)**	**Classes**
Synthetic Control (SC)	600	60	6
Face All (FA)	2250	131	14
Two-Patterns (TP)	5000	128	4
Yoga (YG)	3300	426	2
Wafer (WF)	7164	152	2
Sweadish Leaf (SL)	1125	128	15
CBF	930	128	3

values of h ranging from 2 to 20. We found that lower values of h have a negative effect on the classification accuracy, whereas higher values give time series that cannot be efficiently indexed by multi-dimensional indexing methods. In our future work, we plan to further investigate the effect the dimensionality of time series has on the performance of classification.

All experiments were run twice, once on the original time series and once on their 12-dimensional representations. We wanted to test how the combination of data reduction and dimensionality reduction affects the performance of 1NN classification.

We evaluated the four DRTs by estimating four measurements, namely, accuracy (ACC), classification cost (CC), reduction rate (RR), and, preprocessing cost (PC). Cost measurements were estimated by counting the distance computations multiplied by the number of time series attributes (time series length). Of course, the RR and CC measurements are related to each other: the lower the RR, the higher the CC. However, CC measurements can express the cost introduced by the data dimensionality. We report the average values of these measurements obtained via five-cross-fold validation.

3.2 Comparisons

Table 2 presents the experimental results. The table includes two parts, one for the original datasets and one for the datasets obtained after applying PAA on them. Both table parts include the measurements obtained by applying the 1NN classifier on the non-reduced data (conventional 1NN). Each table cell includes the four measurements obtained by first applying a DRT on the original or 12-dimensional time series datasets (preprocessing step) and then by using 1NN on the resulting CS (classification step). The cost measurements are in million distance computations. The PC measurements do not include the small cost overhead introduced by PAA.

At a glance, we observe that 1NN classification on the 12-dimensional datasets is very fast. In most cases, the preprocessing and classification cost are extremely low, while classification accuracy remains at high, acceptable levels. Therefore, we conclude that one can obtain efficient time series classifiers by combining

Table 2: Experimental results on accuracy, classification cost, reduction rate and preprocessing cost

Dataset		Original dimensionality					12 dimensions				
		Conv. 1NN	**CNN**	**IB2**	**RSP3**	**RHC**	**Conv. 1NN**	**CNN**	**IB2**	**RSP3**	**RHC**
SC	Acc:	91.67	90.17	89.00	98.33	98.67	98.50	97.00	95.83	98.83	98.17
	CC:	3.46	0.67	0.53	1.38	0.09	0.69	0.06	0.05	0.12	0.03
	RR:	-	80.50	84.67	60.08	97.29	-	90.75	93.13	82.96	95.75
	PC:	-	7.77	1.31	16.22	1.31	-	0.89	0.13	3.45	0.13
FA	Acc:	95.07	91.60	91.02	95.46	93.02	87.91	83.78	82.31	87.07	84.49
	CC:	106.11	19.87	18.38	51.65	12.93	9.72	2.89	2.53	4.80	2.08
	RR:	-	81.28	82.68	51.32	87.81	-	70.23	74.01	50.58	78.59
	PC:	-	216.36	48.96	533.70	48.96	-	30.36	5.95	50.91	5.95
TP	Acc:	98.50	94.68	93.60	98.10	93.72	97.56	93.52	91.38	96.66	94.34
	CC:	512.00	85.66	76.83	243.51	55.50	48.00	8.22	6.86	20.42	6.69
	RR:	-	83.27	85.00	52.44	89.16	-	82.89	85.72	57.45	86.06
	PC:	-	1.169.75	205.95	2085.42	205.95	-	103.86	17.34	196.00	17.34
YG	Acc:	93.76	91.58	89.55	92.85	90.94	92.36	90.39	88.03	91.03	90.03
	CC:	742.26	138.56	108.92	229.82	93.85	20.91	4.41	3.50	6.71	3.13
	RR:	-	81.33	85.33	69.04	87.36	-	78.91	83.26	67.90	85.02
	PC:	-	1854.74	254.41	4072.30	254.41	-	52.23	8.04	110.56	8.04
WF	Acc:	99.87	99.69	99.62	99.82	99.55	99.79	99.62	99.51	99.40	99.25
	CC:	1248.30	13.59	11.72	26.88	9.37	98.55	1.21	1.01	1.86	1.01
	RR:	-	98.91	99.06	97.85	99.25	-	98.77	98.97	98.11	98.97
	PC:	-	165.88	31.42	7196.75	31.42	-	15.63	2.57	495.63	2.57
SL	Acc:	52.36	49.87	48.18	52.00	52.80	52.62	49.07	48.62	51.20	51.20
	CC:	25.92	15.94	14.80	19.00	12.80	2.43	1.54	1.37	1.78	1.32
	RR:	-	38.51	42.89	26.69	50.60	-	36.76	43.67	26.69	45.69
	PC:	-	112.17	31.39	1537.07	31.39	-	11.33	2.86	56.00	2.86
CBF	Acc:	98.39	98.17	97.63	99.78	98.60	100.00	99.57	99.35	99.68	99.57
	CC:	17.71	1.29	1.15	1.97	0.40	1.66	0.06	0.06	0.12	0.04
	RR:	-	92.74	93.49	88.87	97.74	-	96.34	96.56	92.63	97.47
	PC:	-	15.06	3.50	78.48	3.50	-	0.66	0.19	7.32	0.19
Avg	Acc:	89.94	87.97	86.94	90.91	89.62	89.82	87.57	86.43	89.12	88.15
	CC:	379.40	39.37	33.19	82.03	26.42	25.99	2.63	2.20	5.12	2.04
	RR:	-	79.51	81.87	63.76	87.03	-	79.24	82.19	68.05	83.94
	PC:	-	505.96	82.42	2217.03	83.37	-	30.71	5.30	131.44	6.57

prototype selection and abstraction algorithms with time-series dimensionality reduction representations.

It is worth mentioning that, in three datasets, the two PA algorithms, RSP3 and RHC, achieved higher classification accuracy than the conv-1NN. In the case of SC dataset, the accuracy improvements were very high. Almost in all cases, RSP3 achieved the highest accuracy. However, it is the slowest method in terms of both preprocessing and classification (RSP3 had the lowest reduction rates). The high PC measurements are attributed to the costly procedure for finding the most distant items in each created subset (see Subsection 2.2 or [9] for details).

RHC and IB2 had much lower preprocessing cost than the other two methods. This happened because IB2 is a one-pass algorithm and RHC is based on a version of k-Means that is sped-up by the class centroid initializations (see Subsection 2.2 or [8] for details). In addition, RHC builds the smallest CSs. In all cases, RHC achieved higher reduction rates than the other DRTs. Thus, the corresponding classifiers had the lowest classification costs. The classification accuracy achieved by RHC was usually higher than IB2 and CNN-rule. In some cases, RHC achieved accuracy even higher than RSP3. One can conclude that RHC is an efficient speed-up method that can deal with all comparison criteria.

No DRT can be considered as the best speed-up choice. If classification accuracy is the most critical criterion, RSP3 may be preferable. On the other hand, if fast classification and/or fast construction of the CS are more critical than accuracy, RHC may be a better choice.

4 Conclusions

Fast time series classification is a crucial data mining issue. This paper proposed the use of non-parametric state-of-the-art prototype selection and abstraction algorithms for fast time series classification.

The experimental study showed that by combining prototype selection and abstraction algorithms with dimensionality reduction, one can obtain accurate and extremely fast time series classifiers. In addition, our study showed that the abstraction algorithms can achieve even higher accuracy that the conventional 1NN classifier.

References

1. Aha, D.W.: Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. Int. J. Man-Mach. Stud. 36(2), 267–287 (Feb 1992), http://dx.doi.org/10.1016/0020-7373(92)90018-G
2. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: Insight: efficient and effective instance selection for time-series classification. In: Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part II. pp. 149–160. PAKDD'11, Springer-Verlag, Berlin, Heidelberg (2011)
3. Chen, C.H., Jóźwik, A.: A sample set condensation algorithm for the class sensitive artificial neural network. Pattern Recogn. Lett. 17, 819–823 (July 1996)

4. Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Trans. Pattern Anal. Mach. Intell. 34(3), 417–435 (Mar 2012)
5. Hart, P.E.: The condensed nearest neighbor rule. IEEE Transactions on Information Theory 14(3), 515–516 (1968)
6. Keogh, E.J., Pazzani, M.J.: A simple dimensionality reduction technique for fast similarity search in large time series databases. In: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications. pp. 122–133. PADKK '00, Springer-Verlag, London, UK, UK (2000)
7. Lin, J., Keogh, E.J., Lonardi, S., chi Chiu, B.Y.: A symbolic representation of time series, with implications for streaming algorithms. In: DMKD. pp. 2–11 (2003)
8. Ougiaroglou, S., Evangelidis, G.: Efficient dataset size reduction by finding homogeneous clusters. In: Proceedings of the Fifth Balkan Conference in Informatics. pp. 168–173. BCI '12, ACM, New York, NY, USA (2012)
9. Sánchez, J.S.: High training set size reduction by space partitioning and prototype abstraction. Pattern Recognition 37(7), 1561–1564 (2004)
10. Triguero, I., Derrac, J., andFrancisco Herrera, S.G.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Transactions on Systems, Man, and Cybernetics, Part C 42(1), 86–100 (2012)
11. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: Proceedings of the 23rd international conference on Machine learning. pp. 1033–1040. ICML '06, ACM, New York, NY, USA (2006)
12. Yi, B.K., Faloutsos, C.: Fast time sequence indexing for arbitrary lp norms. In: Proceedings of the 26th International Conference on Very Large Data Bases. pp. 385–394. VLDB '00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2000)