INSIGHT: Efficient and Effective Instance Selection for Time-Series Classification

Krisztian Buza, Alexandros Nanopoulos, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany,
{buza,nanopoulos,schmidt-thieme}@ismll.de

Abstract. Time-series classification is a widely examined data mining task with various scientific and industrial applications. Recent research in this domain has shown that the simple nearest-neighbor classifier using Dynamic Time Warping (DTW) as distance measure performs exceptionally well, in most cases outperforming more advanced classification algorithms. Instance selection is a commonly applied approach for improving efficiency of nearest-neighbor classifier with respect to classification time. This approach reduces the size of the training set by selecting the best representative instances and use only them during classification of new instances. In this paper, we introduce a novel instance selection method that exploits the hubness phenomenon in time-series data, which states that some few instances tend to be much more frequently nearest neighbors compared to the remaining instances. Based on hubness, we propose a framework for score-based instance selection, which is combined with a principled approach of selecting instances that optimize the *coverage* of training data. We discuss the theoretical considerations of casting the instance selection problem as a graph-coverage problem and analyze the resulting complexity. We experimentally compare the proposed method, denoted as INSIGHT, against FastAWARD, a state-of-the-art instance selection method for time series. Our results indicate substantial improvements in terms of classification accuracy and drastic reduction (orders of magnitude) in execution times.

1 Introduction

Time-series classification is a widely examined data mining task with applications in various domains, including finance, networking, medicine, astronomy, robotics, biometrics, chemistry and industry [11]. Recent research in this domain has shown that the simple nearest-neighbor (1-NN) classifier using Dynamic Time Warping (DTW) [18] as distance measure is "exceptionally hard to beat" [6]. Furthermore, 1-NN classifier is easy to implement and delivers a simple model together with a human-understandable explanation in form of an intuitive justification by the most similar train instances.

The efficiency of nearest-neighbor classification can be improved with several methods, such as indexing [6]. However, for very large time-series data sets, the execution time for classifying new (unlabeled) time-series can still be affected by the significant computational requirements posed by the need to calculate DTW distance between the new time-series and several time-series in the training data set ($O(n)$ in worst case, where n is the size of the training set). *Instance selection* is a commonly applied approach for speeding-up nearest-neighbor classification. This approach reduces the size of the training set by selecting the best representative instances and use only them during classification of new instances. Due to its advantages, instance selection has been explored for time-series classification [20].

In this paper, we propose a novel instance-selection method that exploits the recently explored concept of *hubness* [16], which states that some few instances tend to be much more frequently nearest neighbors than the remaining ones. Based on hubness, we propose a framework for score-based instance selection, which is combined with a principled approach of selecting instances that optimize the *coverage* of training data, in the sense that a time series x covers an other time series y, if y can be classified correctly using x. The proposed framework not only allows better understanding of the instance selection problem, but helps to analyze the properties of the proposed approach from the point of view of coverage maximization. For the above reasons, the proposed approach is denoted as *Instance Selection based on Graph-coverage and Hubness for Time-series* (INSIGHT). INSIGHT is evaluated experimentally with a collection of 37 publicly available time series classification data sets and is compared against FastAWARD [20], a state-of-the-art instance selection method for time series classification. We show that INSIGHT substantially outperforms FastAWARD both in terms of classification accuracy and execution time for performing the selection of instances.

The paper is organized as follows. We begin with reviewing related work in section 2. Section 3 introduces score-based instance selection and the implications of hubness to score-based instance selection. In section 4, we discuss the complexity of the instance selection problem, and the properties of our approach. Section 5 presents our experiments followed by our concluding remarks in section 6.

2 Related Work

Attempts to speed up DTW-based nearest neighbor (NN) classification [3] fall into 4 major categories: i) speed-up the calculation of the distance of two time series, ii) reduce the length of time series, iii) indexing, and iv) instance selection.

Regarding the calculation of the DTW-distance, the major issue is that implementing it in the classic way [18], the comparison of two time series of length l requires the calculation of the entries of an $l \times l$ matrix using dynamic programming, and therefore each comparison has a complexity of $O(l^2)$. A simple idea is to limit the warping window size, which eliminates the calculation of most of the entries of the DTW-matrix: only a small fraction around the diagonal remains. Ratanamahatana and Keogh [17] showed that such reduction does not negatively influence classification accuracy, instead, it leads to more accurate classification. More advanced scaling techniques include lower-bounding, like LB_Keogh [10].

Another way to speed-up time series classification is to reduce the length of time series by aggregating consecutive values into a single number [13], which reduces the overall length of time series and thus makes their processing faster.

Indexing [4], [7] aims at *fast* finding the most similar training time series to a given time series. Due to the “filtering” step that is performed by indexing, the execution time for classifying new time series can be considerable for large time-series data sets, since it can be affected by the significant computational requirements posed by the need to calculate DTW distance between the new time-series and several time-series in the training data set ($O(n)$ in worst case, where n is the size of the training set). For this reason, indexing can be considered complementary to instance selection, since both these techniques can be applied to improve execution time.

Instance selection (also known as *numerosity reduction* or *prototype selection*) aims at discarding most of the training time series while keeping only the most informative ones, which are then used to classify unlabeled instances. While instance selection is well explored for general nearest-neighbor classification, see e.g. [1], [2], [8], [9], [14], there are just a few works for the case of time series. Xi et al. [20] present the FastAWARD approach and show that it outperforms state-of-the-art, general-purpose instance selection techniques applied for time series.

FastAWARD follows an iterative procedure for discarding time series: in each iteration, the rank of all the time series is calculated and the one with lowest rank is discarded. Thus, each iteration corresponds to a particular number of kept time time series. Xi et al. argue that the optimal warping window size depends on the number of kept time series. Therefore, FastAWARD calculates the optimal warping window size for each number of kept time series.

FastAWARD follows some decisions whose nature can be considered as ad-hoc (such as the application of an iterative procedure or the use of tie-breaking criteria [20]). Conversely, INSIGHT follows a more principled approach. In particular, INSIGHT generalizes FastAWARD by being able to use several formulae for scoring instances. We will explain that the suitability of such formulae is based on the hubness property that holds in most time-series data sets. Moreover, we provide insights into the fact that the iterative procedure of FastAWARD is not a well-formed decision, since its large computation time can be saved by ranking instances only once. Furthermore, we observed the warping window size to be less crucial, and therefore we simply use a fixed window size for INSIGHT (that outperforms FastAWARD using adaptive window size).

3 Score functions in INSIGHT

INSIGHT performs instance selection by assigning a score to each instance and selecting instances with the highest scores (see Alg. 1). In this section, we examine how to develop appropriate score functions by exploiting the property of hubness.

3.1 The Hubness Property

In order to develop a score function that selects representative instance for nearest-neighbor time-series classification, we have to take into account the recently explored property of hubness [15]. This property states that for data with high (intrinsic) dimensionality, as most of the time-series data[1], some objects tend to become nearest neighbors much more frequently than others. In order to express hubness in a more precise way, for a data set $\mathscr{D}$ we define the *k-occurrence* of an instance $x \in \mathscr{D}$, denoted $f_N^k(x)$,

[1] In case of time series, consecutive values are strongly interdependent, thus instead of the length of time series, we have to consider the *intrinsic* dimensionality [16].

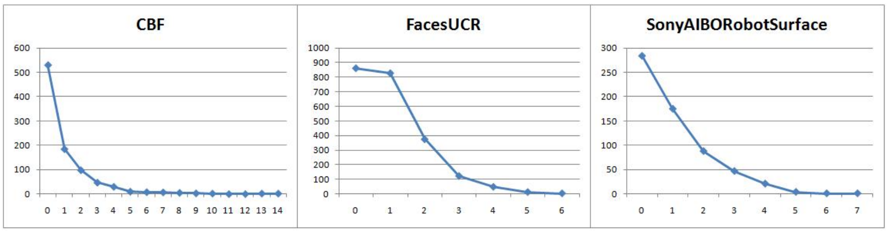

Fig. 1. Distribution of $f_G^1(x)$ for some time series datasets. The horizontal axis correspond to the values of $f_G^1(x)$, while on the vertical axis we see how many instance have that value.

that is the number of instances of $\mathscr{D}$ having x among their k nearest neighbors. With the term *hubness* we refer to the phenomenon that the distribution of $f_N^k(x)$ becomes significantly skewed to the right. We can measure this skewness, denoted by $\mathscr{S}_{f_N^k(x)}$, with the standardized third moment of $f_N^k(x)$:

$$\mathscr{S}_{f_N^k(x)} = \frac{E[(f_N^k(x) - \mu_{f_N^k(x)})^3]}{\sigma^3_{f_N^k(x)}} \tag{1}$$

where $\mu_{f_N^k(x)}$ and $\sigma_{f_N^k(x)}$ are the mean and standard deviation of $f_N^k(x)$. When $\mathscr{S}_{f_N^k(x)}$ is higher than zero, the corresponding distribution is skewed to the right and starts presenting a long tail.

In the presence of labeled data, we distinguish between *good hubness* and *bad hubness*: we say that the instance y is a *good (bad) k-nearest neighbor* of the instance x if (i) y is one of the k-nearest neighbors of x, and (ii) both have the same (different) class labels. This allows us to define *good (bad) k-occurrence* of a time series x, $f_G^k(x)$ (and $f_B^k(x)$ respectively), which is the number of other time series that have x as one of their good (bad) k-nearest neighbors. For time series, both distributions $f_G^k(x)$ and $f_B^k(x)$ are usually skewed, as is exemplified in Figure 1, which depicts the distribution of $f_G^1(x)$ for some time series data sets (from the collection used in Table 1). As shown, the distributions have long tails, in which the good hubs occur.

We say that a time series x is a good (bad) hub, if $f_G^k(x)$ (and $f_B^k(x)$ respectively) is exceptionally large for x. For the nearest neighbor classification of time series, the skewness of good occurrence is of major importance, because a few time series (i.e., the good hubs) are able to correctly classify most of the other time series. Therefore, it is evident that instance selection should pay special attention to good hubs.

3.2 Score functions based on Hubness

Good 1-occurrence score — In the light of the previous discussion, INSIGHT can use scores that take the good 1-occurrence of an instance x into account. Thus, a simple score function that follows directly is the *good 1-occurrence score* $f_G(x)$:

$$f_G(x) = f_G^1(x) \tag{2}$$

Henceforth, when there is no ambiguity, we omit the upper index 1.

While x is being a good hub, at the same time it may appear as bad neighbor of several other instances. Thus, INSIGHT can also consider scores that take bad occurrences into account. This leads to scores that relate the good occurrence of an instance x to either its total occurrence or to its bad occurrence. For simplicity, we focus on the following *relative score*, however other variations can be used too:
Relative score $f_R(x)$ of a time series x is the fraction of good 1-occurrences and total occurrences plus one (plus one in the denominator avoids division by zero):

$$f_R(x) = \frac{f_G^1(x)}{f_N^1(x)+1} \tag{3}$$

Xi's score — Interestingly, $f_G^k(x)$ and $f_B^k(x)$ allows us to interpret the ranking criterion of Xi et al. [20], by expressing it as another form of score for relative hubness:

$$f_{Xi}(x) = f_G^1(x) - 2f_B^1(x) \tag{4}$$

4 Coverage and Instance Selection

Based on scoring functions, such as those described in the previous section, INSIGHT selects top-ranked instances (see Alg. 1). However, while ranking the instances, it is also important to examine the interactions between them. For example, suppose that the 1st top-ranked instance allows correct 1-NN classification of almost the same instances as the 2nd top-ranked instance. The contribution of the 2nd top-ranked instance is, therefore, not important with respect to the overall classification. In this section we describe the concept of *coverage graphs*, which helps to examine the aforementioned aspect of interactions between the selected instances. In Section 4.1 we examine the general relation between coverage graphs and instance-based learning methods, whereas in Section 4.2 we focus on the case of 1-NN time-series classification.

4.1 Coverage graphs for Instance-based Learning Methods

We first define coverage graphs, which in the sequel allow us to cast the instance-selection problem as a graph-coverage problem:

Definition 1 (Coverage graph). *A coverage graph $G_c = (V,E)$ is a directed graph, where each vertex $v \in V_{G_c}$ corresponds to a time series of the (labeled) training set. A directed edge from vertex v_x to vertex v_y denoted as $(v_x,v_y) \in E_{G_c}$ states that instance y contributes to the correct classification of instance x.*

Algorithm 1 INSIGHT

```
Require: Time-series dataset $D$, Score Function $f$, Number of selected instances $N$
Ensure: Set of selected instances (time series) $D'$
1: Calculate score function $f(x)$ for all $x \in D$
2: Sort all the time series in $D$ according to their scores $f(x)$
3: Select the top-ranked $N$ time series and return the set containing them
```

We first examine coverage graphs for the general case of instance-based learning methods, which include the k-NN ($k \geq 1$) classifier and its generalizations, such as adaptive k-NN classification where the number of nearest neighbors k is chosen adaptively for each object to be classified [12], [19].[2] In this context, the *contribution* of an instance y to the correct classification of an instance x refers to the case when y is among the nearest neighbors of x and they have the same label.

Based on the definition of the coverage graph, we can next define the coverage of a specific vertex and of set of vertices:

Definition 2 (Coverage of a vertex and of vertex-set). *A vertex v covers an other vertex v' if there is an edge from v' to v; $C(v)$ is the set of all vertices covered by v: $C(v) = \{v' | v' \neq v \wedge (v', v) \in E_{G_c}\}$. Moreover, a set of vertices S_0 covers all the vertices that are covered by at least one vertex $v \in S_0$: $C(S_0) = \bigcup_{\forall v \in S_0} C(v)$.*

Following the common assumption that the distribution of the test (unlabeled) data is similar to the distribution of the training (labeled) data, the more vertices are covered, the better prediction for new (unlabeled) data is expected. Therefore, the objective of an instance-selection algorithm is to have the selected vertex-set S (i.e., selected instances) covering the entire set of vertices (i.e., the entire training set), i.e., $C(S) = V_{G_c}$. This, however, may not be always possible, such as when there exist vertices that are not covered by any other vertex. If a vertex v is not covered by any other vertex, this means that the out-degree of v is zero (there are no edges going from v to other vertices). Denote the set of such vertices with by $V^0_{G_c}$. Then, an ideal instance selection algorithm should cover all *coverable* vertices, i.e., for the selected vertices S an ideal instance selection algorithm should fulfill:

$$\bigcup_{\forall v \in S} C(v) = V_{G_c} \setminus V^0_{G_c} \tag{5}$$

In order to achieve the aforementioned objective, the trivial solution is to select all the instances of the training set, i.e., chose $S = V_{G_c}$. This, however is not an effective instance selection algorithm, as the major aim of discarding less important instances is not achieved at all. Therefore, the natural requirement regarding the ideal instance selection algorithm is that it selects the *minimal* amount of those instances that together cover all coverable vertices. This way we can cast the instance selection task as a coverage problem:

Instance selection problem (ISP) — We are given a coverage graph $G_c = (V, E)$. We aim at finding a set of vertices $S \subseteq V_{G_c}$ so that: i) all the coverable vertices are covered (see Eq. 5), and ii) the size of S is minimal among all those sets that cover all coverable vertices.

Next we will show that this problem is NP-complete, because it is equivalent to the set-covering problem (SCP), which is NP-complete [5]. We proceed with recalling the set-covering problem.

[2] Please notice that in the general case the resulting coverage graph has no regularity regarding both the in- and out-degrees of the vertices (e.g., in the case of k-NN classifier with adaptive k).

Set-covering problem (SCP) — ”An instance $(X,\mathscr{F})$ of the set-covering problem consists of a finite set X and a familiy $\mathscr{F}$ of subsets of X, such that every element of X belongs to at least one subset in $\mathscr{F}$. (...) We say that a subset $F \in \mathscr{F}$ covers its elements. The problem is to find a minimum-size subset $\mathscr{C} \subseteq \mathscr{F}$ whose members cover all of X”[5]. Formally: the task is to find $\mathscr{C} \subseteq \mathscr{F}$, so that $|\mathscr{C}|$ is minimal and $X = \bigcup\limits_{\forall F \in \mathscr{C}} F$

Theorem 1. *ISP and SCP are equivalent. (See Appendix for the proof.)*

4.2 1-NN coverage graphs

In this section, we introduce 1-nearest neighbor (1-NN) coverage graphs which is motivated by the good performance of the 1-NN classifier for time series classification. We show the optimality of INSIGHT for the case of 1-NN coverage graphs and how the NP-completeness of the general case (Section 4.1) is alleviated for this special case.

We first define the specialization of the coverage graph based on the 1-NN relation:

Definition 3 (1-NN coverage graph). *A 1-NN coverage graph, denoted by G_{1NN} is a coverage graph where $(v_x, v_y) \in E_{G_{1NN}}$ if and only if time series y is the first nearest neighbor of time series x and the class labels of x and y are equal.*

This definition states that an edge points from each vertex v to the nearest neighbor of v, only if this is a good nearest neighbor (i.e., their labels match). Thus, vertexes are *not* connected with their bad nearest neighbors.

From the practical point of view, to account for the fact that the size of selected instances is defined apriori (e.g., a user-defined parameter), a slightly different version of the Instance Selection Problem (ISP) is the following:

***m*-limited Instance Selection Problem (*m*-ISP)** — If we wish to select exactly m labeled time series from the training set, then, instead of selecting the minimal amount of time series that ensure total coverage, we select those m time series that maximize the coverage. We call this variant *m-limited Instance Selection Problem* (*m*-ISP). The following proposition shows the relation between 1-NN coverage graphs and *m*-ISP:

Proposition 1. *In 1-NN coverage graphs, selecting m vertices v_1, ..., v_m that have the largest covered sets $C(v_1)$, ..., $C(v_m)$ leads to the optimal solution of m-ISP.*

The validity of this proposition stems from the fact that, in 1-NN coverage graphs, the out-degree of all vertices is 1. This implies that each vertex is covered by *at most* one other vertex, i.e., the covered sets $C(v)$ are mutually disjoint for each $v \in V_{G_{1NN}}$.

Proposition 1 describes the optimality of INSIGHT, when the good 1-occurrence score (Equation 2) is used, since the size of the set $C(v_i)$ is the number of vertices having v_i as first good nearest neighbor. It has to be noted that described framework of coverage graphs can be extended to other scores too, such as relatives scores (Equations 3 or 4). In such cases, we can additionally model bad neighbors and introduce weights on the edges of the graph. For example, for the score of Equation 4, the weight of an edge e is $+1$, if e denotes a good neighbor, whereas it is -2, if e denotes a bad neighbor. We can define the coverage score of a vertex v as the sum of weights of the incoming edges to v and aim to maximize this coverage score. The detailed examination of this generalization is addressed as future work.

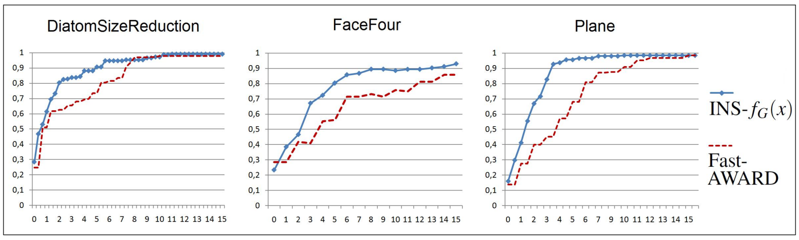

Fig. 2. Accuracy as function of the number of selected instances (in % of the entire training data) for some datasets for FastAWARD and INSIGHT.

5 Experiments

We experimentally examine the performance of INSIGHT with respect to effectiveness, i.e., classification accuracy, and efficiency, i.e., execution time required by instance selection. As baseline we use FastAWARD [20].

We used 37 publicly available time series datasets[3] [6]. We performed 10-fold-cross validation. INSIGHT uses $f_G(x)$ (Eq. 2) as the default score function, however $f_R(x)$ (Eq. 3) and $f_{Xi}(x)$ (Eq. 4) are also being examined. The resulting combinations are denoted as INS-$f_G(x)$, INS-$f_R(x)$ and INS-$f_{Xi}(x)$, respectively.

The distance function for the 1-NN classifier is DTW that uses warping windows [17]. In contrast to FastAWARD, which determines the optimal warping window size r_{opt}, INSIGHT sets the warping-window size to a constant of 5%. (This selection is justified by the results presented in [17], which show that relatively small window sizes lead to higher accuracy.) In order to speed-up the calculations, we used the LB_Keogh lower bounding technique [10] for both INSIGHT and FastAWARD.

Results on Effectiveness — We first compare INSIGHT and FastAWARD in terms of classification accuracy that results when using the instances selected by these two methods. Table 1 presents the average accuracy and corresponding standard deviation for each data set, for the case when the number of selected instances is equal to 10% of the size of the training set (for INSIGHT, the INS-$f_G(x)$ variation is used). In the vast majority of cases, INSIGHT substantially outperforms FastAWARD. In the few remaining cases, their difference are remarkably small (in the order of the second or third decimal digit, which are not significant in terms of standard deviations).

We also compared INSIGHT and FastAWARD in terms of the resulting classification accuracy for varying number of selected instances. Figure 2 illustrates that INSIGHT compares favorably to FastAWARD. Due to space constraints, we cannot present such results for all data sets, but analogous conclusion is drawn for all cases of Table 1 for which INSIGHT outperforms FastAWARD.

Besides the comparison between INSIGHT and FastAward, what is also interesting is to examine their relative performance compared to using the entire training data (i.e., no instance selection is applied). Indicatively, for 17 data sets from Table 1 the accuracy

[3] For StarLightCurves the calculations have not been completed for FastAWARD till the submission, therefore we omit this dataset.

Table 1. Accuracy ± standard deviation for INSIGHT and FastAWARD (bold font: winner).

Dataset	FastAWARD	INS-$f_G(x)$	Dataset	FastAWARD	INS-$f_G(x)$
50words	0.526±0.041	**0.642±0.046**	Lighting7	0.447±0.126	**0.510±0.082**
Adiac	0.348±0.058	**0.469±0.049**	MALLAT	0.551±0.098	**0.969±0.013**
Beef	**0.350±0.174**	0.333±0.105	MedicalImages	0.642±0.033	**0.693±0.049**
Car	0.450±0.119	**0.608±0.145**	Motes	0.867±0.042	**0.908±0.027**
CBF	0.972±0.034	**0.998±0.006**	OliveOil	0.633±0.100	**0.717±0.130**
Chlorine[a]	0.537±0.023	**0.734±0.030**	OSULeaf	0.419±0.053	**0.538±0.057**
CinC	0.406±0.089	**0.966±0.014**	Plane	0.876±0.155	**0.981±0.032**
Coffee	0.560±0.309	**0.603±0.213**	Sony[d]	0.924±0.032	**0.976±0.017**
Diatom[b]	**0.972±0.026**	0.966±0.058	SonyII[e]	**0.919±0.015**	0.912±0.033
ECG200	0.755±0.113	**0.835±0.090**	SwedishLeaf	0.683±0.046	**0.756±0.048**
ECGFiveDays	0.937±0.027	**0.945±0.020**	Symbols	0.957±0.018	**0.966±0.016**
FaceFour	0.714±0.141	**0.894±0.128**	SyntheticControl	0.923±0.068	**0.978±0.026**
FacesUCR	0.892±0.019	**0.934±0.021**	Trace	0.780±0.117	**0.895±0.072**
FISH	0.591±0.082	**0.666±0.085**	TwoPatterns	0.407±0.027	**0.987±0.007**
GunPoint	0.800±0.124	**0.935±0.059**	TwoLeadECG	0.978±0.013	**0.989±0.012**
Haptics	0.303±0.068	**0.435±0.060**	Wafer	0.921±0.012	**0.991±0.002**
InlineSkate	0.197±0.056	**0.434±0.077**	WordsSynonyms	0.544±0.058	**0.637±0.066**
Italy[c]	**0.960±0.020**	0.957±0.028	Yoga	0.550±0.017	**0.877±0.021**
Lighting2	**0.694±0.134**	0.670±0.096			

[a] ChlorineConcentration, [b] DiatomSizeReduction, [c] ItalyPowerDemand,
[d] SonyAIBORobotSurface, [e] SonyAIBORobotSurfaceII

resulting from INSIGHT (INS-$f_G(x)$) is worse by less than 0.05 compared to using the entire training data. For FastAward this number is 4, which clearly shows that INSIGHT select more representative instances of the training set than FastAward.

Next, we investigate the reasons for the presented difference between INSIGHT and FastAward. In Section 3.1, we identified the skewness of good k-occurrence, $f_G^k(x)$, as a crucial property for instance selection to work properly, since skewness renders good hubs to become representative instances. In our examination, we found that using the iterative procedure applied by FastAWARD, this skewness has a decreasing trend from iteration to iteration. Figure 3 exemplifies this by illustrating the skewness of $f_G^1(x)$ for two data sets as a function of iterations performed in FastAWARD. (In order to quantitatively measure skewness we use the standardized third moment, see Equation 1.) The reduction in the skewness of $f_G^1(x)$ means that FastAWARD is not able to identify in the end representative instances, since there are no pronounced good hubs remaining.

To further understand that the reduced effectiveness of FastAWARD stems from its iterative procedure and not from its score function, $f_{Xi}(x)$ (Eq. 4), we compare the accuracy of all variations of INSIGHT including INS-$f_{Xi}(x)$, see Tab. 2. Remarkably, INS-$f_{Xi}(x)$ clearly outperforms FastAWARD for the majority of cases, which verifies our previous statement. Moreover, the differences between the three variations are not large, indicating the robustness of INSIGHT with respect to the scoring function.

Results on Efficiency — The computational complexity of INSIGHT depends on the calculation of the scores of the instances of the training set and on the selection of the top-ranked instances. Thus, for the examined score functions, the computational complexity is $O(n^2)$, n being the number of training instances, since it is determined by the calculation of the distance between each pair of training instances. For FastAWARD, its first step (leave-one-out nearest neighbor classification of the train instances) already requires $O(n^2)$ execution time. However, FastAWARD performs additional computationally expensive steps, such as determining the best warping-window size and the iterative procedure for excluding instances. For this reason, INSIGHT is expected to require reduced execution time compared to FastAWARD. This is verified by the results presented in Table 3, which shows the execution time needed to perform instance selection with INSIGHT and FastAWARD. As expected, INSIGHT outperforms FastAWARD drastically. (Regarding the time for classifying new instances, please notice that both methods perform 1-NN using the same number of selected instances, therefore the classification times are equal.)

6 Conclusion and Outlook

We examined the problem of instance selection for speeding-up time-series classification. We introduced a principled framework for instance selection based on coverage graphs and hubness. We proposed INSIGHT, a novel instance selection method for time series. In our experiments we showed that INSIGHT outperforms FastAWARD, a state-of-the-art instance selection algorithm for time series.

In our future work, we aim at examining the generalization of coverage graphs for considering weights on edges. We also plan to extend our approach for other instance-based learning methods besides 1-NN classifier.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine learning. 6(1), 37–66 (1991)
2. Brighton, H., Mellish, C.: Advances in Instance Selection for Instance-Based Learning Algorithms. Data Mining and Knowledge Discovery 6, 153–172 (2002)
3. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: Time-Series Classification based on Individualised Error Prediction. In: IEEE CSE'10 (2010)

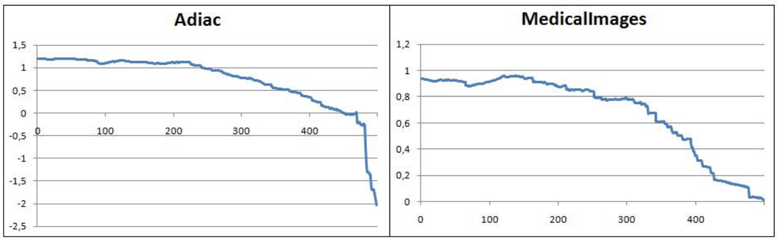

Fig. 3. Skewness of the distribution of $f_G^1(x)$ as function of the number of iterations performed in FastAWARD. On the trend, the skewness decreases from iteration to iteration.

4. Chakrabarti, K., Keogh, E., Sharad, M., Pazzani, M.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Transactions on Database Systems 27, 188–228 (2002)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge, Massachusetts (2001)
6. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. In: VLDB '08 (2008)
7. Gunopulos, D., Das, G.: Time series similarity measures and time series indexing. ACM SIGMOD Record. 30, 624 (2001)
8. Jankowski, N., Grochowski, M.: Comparison of instance selection algorithms I. Algorithms survey. In: ICAISC, LNCS, Vol. 3070/2004, 598–603, Springer (2004)
9. Jankowski, N., Grochowski, M.: Comparison of instance selection algorithms II. Results and Comments. In: ICAISC, LNCS, Vol. 3070/2004, 580–585, Springer (2004)
10. Keogh, E.: Exact indexing of dynamic time warping. In: VLDB'02 (2002)
11. Keogh, E., Kasetty, S.: On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. In: SIGKDD (2002)

Table 2. Number of datasets where different versions of INSIGHT win/lose against FastAWARD.

	INS-$f_G(x)$	INS-$f_R(x)$	INS-$f_{Xi}(x)$
Wins	32	33	33
Loses	5	4	4

Table 3. Execution times (in seconds, averaged over 10 folds) of instance selection using INSIGHT and FastAWARD

Dataset	FastAWARD	INS-$f_G(x)$	Dataset	FastAWARD	INS-$f_G(x)$
50words	94 464	203	Lighting7	5 511	8
Adiac	32 935	75	Mallat	4 562 881	19 041
Beef	1 273	3	MedicalImages	13 495	55
Car	11 420	18	Motes	17 937	55
CBF	37 370	67	OliveOil	3 233	5
ChlorineConcentration	16 920	1 974	OSULeaf	80 316	118
CinC	3 604 930	16 196	Plane	1 527	4
Coffee	499	1	SonyAIBORobotS.	4 608	11
DiatomSizeReduction	18 236	44	SonyAIBORobotS.II	10 349	23
ECG200	634	2	SwedishLeaf	37 323	89
ECGFiveDays	20 455	60	Symbols	165 875	514
FaceFour	4 029	6	SyntheticControl	3 017	8
FacesUCR	150 764	403	Trace	3 606	11
FISH	59 305	93	TwoPatterns	360 719	1 693
GunPoint	1 107	4	TwoLeadECG	12 946	45
Haptics	152 617	869	Wafer	923 915	4 485
InlineSkate	906 472	4 574	WordsSynonyms	101 643	203
ItalyPowerDemand	1 855	6	Yoga	1 774 772	6 114
Lighting2	15 593	23			

12. Ougiaroglou, S., Nanopoulos, A., Papadopoulos, A. N., Manolopoulos, Y., Welzer-Druzovec, T.: Adaptive k-Nearest-Neighbor Classification Using a Dynamic Number of Nearest Neighbors, In: ADBIS, LNCS Vol. 4690/2007, Springer (2007)
13. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (2003)
14. Liu, H., Motoda, H.: On Issues of Instance Selection. Data Mining and Knowledge Discovery. 6, 115–130 (2002)
15. Radovanovic, M., Nanopoulos, A., Ivanovic, M.: Nearest Neighbors in High-Dimensional Data: The Emergence and Influence of Hubs. In: ICML'09 (2009)
16. Radovanovic, M., Nanopoulos, A., Ivanovic, M.: Time-Series Classification in Many Intrinsic Dimensions. In: 10th SIAM International Conference on Data Mining (2010)
17. Ratanamahatana, C.A., Keogh, E.: Three myths about Dynamic Time Warping. In SDM (2005)
18. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoustics, Speech, and Signal Proc. 26, 43–49 (1978)
19. Wettschereck, D., Dietterich, T.: Locally Adaptive Nearest Neighbor Algorithms, Advances in Neural Information Processing Systems 6, Morgan Kaufmann (1994)
20. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast Time Series Classification Using Numerosity Reduction. In: ICML'06 (2006)

Appendix: Proof of Theorem 1

We show the equivalence in two steps. First we show that ISP is a subproblem of SCP, i.e. for each instance of ISP a corresponding instance of SCP can be constructed (and the solution of the SCP-instances directly gives the solution of the ISP-instance). In the second step we show that SCP is a subproblem of ISP. The both together imply equivalence.

For each ISP-instance we construct a corresponding SCP-instance: $X := V_{G_c} \setminus V^0_{G_c}$ and $\mathscr{F} := \{C(v) | v \in V_{G_c}\}$ We say that vertex v is the *seed* of the set $C(v)$. The solution of SCP is a set $F \subseteq \mathscr{F}$. The set of seeds of the subsets in F constitute the solution of ISP: $S = \{v | C(v) \in F\}$

While constructing an ISP-instance for an SCP-instance, we have to be careful, because the number of subsets in SCP is not limited. Therefore in the coverage graph G_c there are two types of vertices. Each first-type-vertex v_x corresponds to one element $x \in X$, and each second-type-vertex v_F correspond to a subset $F \in \mathscr{F}$. Edges go only from first-type-vertices to second-type-vertices, thus only first-type-vertices are coverable. There is an edge (v_x, v_F) from a first-type-vertex v_x to a second-type-vertex v_F if and only if the corresponding element of X is included in the corresponding subset F, i.e. $x \in F$. When the ISP is solved, all the coverable vertices (first-type-vertices) are covered by a minimal set of vertices S. In this case, S obviously consits only of second-type-vertices. The solution of the SCP are the subsets corresponding to the vertices included in S: $\mathscr{C} = \{F | F \in \mathscr{F} \wedge v_F \in S\}$