Hubness-based indicators for semi-supervised time-series classification

Kristof Marussy*

Dpt. of Computer Science and Inf. Theory
Budapest University of Techn. and Economics
1117 Budapest, Magyar tudósok körútja 2.,
Hungary
kris7topher@gmail.com

Krisztian Buza†

Dpt. of Computer Science and Inf. Theory
Budapest University of Techn. and Economics
1117 Budapest, Magyar tudósok körútja 2.,
Hungary
buza@cs.bme.hu

Abstract: Due to the decreasing cost and wide availability of sensors that measure the change of a quantity over time, machine learning methods focusing on time-series have gained increasing attention during the last few decades. In domains where time-series databases are only emerging, adequate amounts of data annotated by human experts might not be available due to the associated high expenses and required effort. Semi-supervised learning techniques are able to alleviate some of these problems. We exploit the relation between *hubness*—a phenomenon emergent in time-series data sets due to their high dimensionality—and the effectiveness of semi-supervised learning. We evaluate various hubness-based indicator values on 44 publicly available data sets whether they can predict the effectiveness of semi-supervised learning. We also investigate the use of hubness-based indicators in choosing between two semi-supervised learners for time series. Our results show that selection of the appropriate learning method is often possible based on indicators available *a priori*, without labelling the whole data set.

Keywords: semi-supervised learning, time-series classification, hubness, Dynamic Time Warping, nearest neighbour

1 Introduction

Due to the decreasing cost and wide availability of sensors that measure the change of a quantity over time, machine learning methods focusing on time-series have gained increasing attention during the last few decades. Huge amounts of time series data became available. For example, in the financial domain, even the storage of time-series data is challenging [10].

One of the most prominent tasks in machine learning is *classification*. Applications for time series include e.g. handwriting, speech [15] and sign language recognition, signature verification [6] and medical diagnosis based on electroencephalogram (EEG, "brain wave") and electrocardiograph (ECG) signals [2]. The construction of classifiers requires review and labelling of data by human experts. The sheer size of data sets allows such processing for only a fraction of available data. This fraction may not represent the whole data set very well, which can lead to suboptimal classifiers.

By combining both labelled and unlabelled data, semi-supervised learning can improve classification accuracy. However, semi-supervised learners usually require the data set to have a certain structure. Several learners are based on the *cluster assumption*: instances similar to each other have similar labels too. This property is related to *hubness*, the tendency of instances to cluster around a few "hubs" [12, 13].

*This research was developed in the framework of the project TÁMOP – 4.2.2.C-11/1/KONV-2012-0013 ("Infokommunikációs technológiák és a jövő társadalma"). We aknowledge the DAAD-MÖB Researcher Exchange Program.

†Krisztian Buza is on research term at the University of Warsaw, Poland.

In this paper, we investigate the ability of indicators derived from hubness to discriminate between data sets in which semi-supervised learning leads to classification accuracy improvement—i.e. the corresponding assumption holds—and those in which it does not. We especially focus on measures that are computable *a priori*, without first labelling the whole data set. We also attempt to choose between two semi-supervised learners for time series, Wei's algorithm [19] and SUCCESS [9]. We also propose modified hubness measures that try to capture the properties of these learners.

We performed experiments on 44 publicly available data sets [7]. Our results show that, in many cases, selection of the appropriate learning method is possible based on indicators available *a priori*. However, selection of the most descriptive indicator is still scope of future work.

2 Background

In this section, we briefly review some of the most closely related works concerning *self-training*, *k-nearest neighbour classification*, *semi-supervised time series classification* and *hubness*. For an in-depth review of semi-supervised learning techniques, we refer to [16, 23] and the references therein.

2.1 Semi-supervised learning

Classical machine learning tasks are said to be either *supervised* or *unsupervised*. In unsupervised problems—e.g. *clustering*—a set of *instances* $U = \{x_i\}_{i=1}^{n}$ is given to the learning algorithm. The output is a set of clusters of similar instances found. In contrast, supervised learners take *labelled* instances $L = \{(x_i, y_i)\}$. In *classification* problems the discrete label y_i signifies which class x_i belongs to. In the *training* stage a predictor $\hat{y}(\cdot)$ is constructed from L which is sought to estimate (class-) labels of instances with labels unknown to us.

Labelling instances is usually a tedious process which must be performed by a human expert with domain knowledge. Therefore, while unlabelled data is abundant, labelled data is often scarce and expensive. Semi-supervised algorithms attempt to learn from both labelled (L) and unlabelled (U) data even when the available labelled data does not represent the overall distribution and structure of labels. Thus it is possible to reduce the amount of labelled data—and effort of human expert—required for accurate prediction.

Self-training Self-training is one of the most common semi-supervised learning methods. It is a wrapper method around a supervised classifier.

Self-training trains a sequence of supervised models by the following iterative process:

1. Let $L_0 = L$ the initial training set and $t = 0$.
2. Train the supervised learner on L_t.
3. Assess the *certainty* of classification by the supervised learner for each instance.
4. Construct the next training set L_{t+1} taking the certainty estimates into account. If an instance is added to L_{t+1} that was not in L_t—i.e. its label is yet unknown—it takes whatever label is predicted by the model trained in Step 2.
5. Let $t = t + 1$ and go to Step 2 unless some *stopping criterion* is satisfied.

The supervised classifier is gradually trained using its own output. In the simplest case, L_{t+1} is constructed by appending some of the most certainly classified unlabelled instances along with their predicted labels to L_t. More elaborate methods include e.g. Yarowsky's algorithm [21].

2.2 Time-series classification

Time-series are temporal sequences of scalar- or vector-valued measurements.

In time-series classification problems, a time-series x is associated with a class label $y(x)$. Prediction of this class label is sought. A competitive—and in some cases, even better than many more complicated

approaches—method of time-series classification is the nearest-neighbour classifier with Dynamic Time Warping [5]. Dynamic Time Warping (DTW) [15] is a *distance measure* for time-series: a function $d_{\mathrm{DTW}}(\cdot,\cdot)$ assigns a numeric value of dissimilarity to a pair of time-series.

***k*-nearest neighbour classifier** We can predict the label of a time series x with the k-nearest neighbour (k-NN) classifier by looking at the class of its k nearest (according to Dynamic Time Warping) neighbours in the set of time-series with known labels—the training set in supervised learning—L. Specifically, 1-nearest neighbour (1-NN) sets the prediction to the class label of the instance in L which is the least dissimilar to x.

***k*-nearest neighbour graph** A useful device in k-nearest neighbour classification is the nearest neighbour graph [1], which gives insight to what instances influence the classification of other instances most.

Definition 1 *Let us define the* k-nearest neighbour coverage *graph of a set of instances* $X = \{x_i\}$ *as the directed graph* $G_N^k = (X, E)$ *which contains the edge* (x_i, x_j) *between two instances if and only if* x_j *has* x_i *among its* k *nearest neighbours w.r.t. some distance function* $d(\cdot,\cdot)$.

2.3 Semi-supervised methods for time-series classification

There are surprisingly few works on semi-supervised classification of time-series. Wei and Keogh proposed a method based on *self-training* with 1-nearest neighbour [19], which was extended by Ratanamahatana et al. [14] with a new stopping criterion. In our recent work [9], we applied similar principles to construct a classifier based on *constrained classification*, which performs better on many data sets. We call this approach SUCCESS: Semi-sUpervised ClassifiCation of timE-SerieS.

Other, more complicated semi-supervised learners include k-means and principal component analysis by Nguyen et al. [11] and self-training with Hidden Markov Models by Zhong [22].

We will now illustrate the learners considered in this study in more detail.

Wei's algorithm Wei's algorithm [19] applies 1-nearest neighbour for time-series with a self-training protocol. Classification certainty is determined by the DTW-distance of the unlabelled instance and the closest labelled time-series. The closer is an instance to its labelled neighbour, the more certain classification is expected to be. The training set is grown by one instance at a time. With Ratanamahatana's stopping criterion [14] for multiple—i.e. more than two—classes, the algorithm stops when all available unlabelled time-series become labelled.

SUCCESS Our classifier, SUCCESS is based on *constrained hierarchical agglomerative clustering*. Hierarchical clustering with constrains was shown to improve clustering accuracy compared to traditional hierarchical clustering [8]. The approach can be summarized in the following steps:

1. When the algorithm starts, each instance—labelled or unlabelled—forms a cluster of one element.
2. Clusters are iteratively merged. At each step the two closest clusters are merged into a new cluster. The distance of clusters is determined by *single link*: it is the DTW-distance of the closest pair of elements in the two clusters. Formally, $d(C_1, C_2) = \min_{x_1 \in C_1, x_2 \in C_2} d_{\mathrm{DTW}}(x_1, x_2)$.
3. If two clusters both contain a labelled instance, they will not be merged with each other. The search for the closest cluster pair continues until two mergeable clusters are found. Effectively, we have a *cannot-link* (CL) constraint [18] for each pair of labelled time-series that forbids them from being in the same cluster.
4. When no more clusters can be merged, each resulting cluster contains exactly one labelled instance. We refer to these labelled time-series as *seeds*. Prediction output for unlabelled time-series in each cluster is the class label of their associated seed.
5. The initially labelled instances and the unlabelled instances enhanced with predicted class labels form the training set for 1-nearest neighbour. The resulting classifier, which can predict labels for *unseen* instances, is hence analogous with the final classifier in Wei's self-training algorithm.

Relationship between Wei's algorithm and SUCCESS Let us now consider the *complete* graph of all instances, labelled or unlabelled, G with edge weights $w_{i,j} = d_{\mathrm{DTW}}(x_i, x_j)$. Without loss of generality, let us denote the labelled time-series—seeds—$L = \{x_i\}_{i=1}^{\ell}$, i.e. the first ℓ known instances are labelled. The rest of the instances in the training set $U = \{x_i\}_{i=\ell+1}^{n}$ are unlabelled.

Definition 2 *A set of trees $\mathcal{T} = \{T_i\}$ is a* spanning forest *of G if the following properties hold: (i) There are exactly as many trees as labelled time-series: $|\mathcal{T}| = \ell$. (ii) For each labelled time-series there is exactly one corresponding tree which contains it: $\forall 1 \leq i \leq \ell : x_i \in V(T_i)$. (iii) The sets $V(T_1), V(T_2), \ldots, V(T_\ell)$ together form a disjoint partition of $V(G) = L \cup U$.*

Given a spanning forest, *transductive* classification of U may be performed by assigning the class label of the corresponding seed to all the unlabelled instances of trees.

Definition 3 *The* weight *of a spanning forest is the sum of their edge weights,*

$$W(\mathcal{T}) = \sum_{i=1}^{\ell} \sum_{e \in E(T_i)} w(e) \tag{1}$$

Definition 4 *A* minimum spanning forest *is a spanning forest of minimal weight.*

Now let us extend G with super-vertex $\star$ to form $G^\star$, i.e. $V(G^\star) = V(G) \cup \{\star\}$. The super-vertex is connected to the labelled time-series with zero-weight edges,

$$E(G^\star) = E(G) \cup \{\{x_i, \star\} : 1 \leq i \leq \ell\}, \quad w(\{x_i, \star\}) = 0 \tag{2}$$

If all the edges of G have strictly positive weight, a minimum spanning tree of $G^\star$ that contains all 0-weight edges can be converted into a minimum spanning forest of G by removing $\star$: If there were two seeds in the same tree of the resulting forest, the minimum spanning tree would contain a circle; which is a contradiction. Moreover, the resulting tree cannot have greater weight than a minimum spanning forest of G.

Wei's algorithm is analogous to running Prim's algorithm [3] on $G^\star$ and classifying the unlabelled instances by the corresponding minimum spanning forest of G. In contrast, SUCCESS uses Kruskal's algorithm to achieve the same goal.

2.4 Hubness

Hubness is a phenomenon emergent in data sets of high dimensionality [12]. In such data sets some instances, which are called *hubs*, are among the nearest neighbours of surprisingly many other instances and therefore influence classification of those.

The presence of *good hubs* with class labels equal to their neighbours' may aid classification [1]. In contrast, *bad hubs*, which have a different class label than their neighbours, are especially problematic because they are responsible for a large number of misclassifications. Hubness occurs in many real-world time-series data sets [13].

Measuring hubness Radovanovic et al. [13] developed a framework for characterizing hubness:

Definition 5 *The k-*occurrence score *g_N^k of an instance x is the number of (other) instances which have it among their k nearest neighbours, i.e. the out-degree of x in G_N^k.*

The extent to which hubness is present in a data set may be determined by observing the k-occurrence scores. In the presence of strong hubness, the distribution of g_N^k is highly skewed: a handful of instances have exceptionally large k-occurrence. Therefore, the *skewness*—or *standardized third moment*—$S[g_N^k]$ is and indicator of overall hubness.

Definition 6 *The* bad *k-*occurrence score *$g_B^k(x)$ of an instance x is the number of instances that have a different class label than x but have x among their k nearest neighbours.*

Definition 7 *The* normalised total bad k-occurrence score $\widetilde{BN}_k$ *is the number of bad k-occurrences divided by the number of all k-occurrences, i.e.*

$$\widetilde{BN}_k = \frac{\sum_{x \in X} g_B^k(x)}{k \cdot |X|}. \tag{3}$$

$\widetilde{BN}_k$ measures bad hubness in the data set.

3 Hubness and semi-supervised learning for time-series

When a semi-supervised learning algorithm is selected, assumptions are implicitly taken about the relationship of labelled and unlabelled data. If our assumptions are wrong, unlabelled data may decrease classification accuracy [17]. For example, the *cluster assumption* states that two instances that are similar to each other have similar labels. High hubness and high bad hubness indicates *cluster assumption violation* (CAV) [13]. This violation hurts both supervised and semi-supervised learners.

One could determine whether semi-supervised learning improves classification accuracy by splitting the available labelled data into a *training set* and *test set*. By leaving out the test set from classifier training phase, classifier output can be fairly compared to the true class labels of the test set. However, if the quantity of labelled data available is small, this comparison may not be a good estimate of true classification accuracy. Therefore, other devices are needed for *a priori* estimation of semi-supervised classification accuracy and selection of appropriate learner.

We will now examine if hubness is related to the success of semi-supervised learning strategies and introduce two new measures of hubness, which also take semi-supervised nearest-neighbour learning methods such as Wei's self-training algorithm [19].

3.1 Transitive (semi-supervised) hubness

In Subsection 2.3 we have seen that if a labelled instance can influence the classification of its unlabelled neighbours. If a labelled hub has some unlabelled neighbours in the training set that are themselves hubs, it influences the classification of those unlabelled hubs' neighbours. The labelled hub hence can *indirectly* influences the classification of a lot more time-series than in traditional supervised classification.

This phenomenon can be illustrated by the transitive closure G_{TN}^k of the k-nearest neighbour graph G_N^k. In G_{TN}^k, hubs are connected to all those instances of which the classification they influence.

Definition 8 *The* transitive k-occurrence score $g_{TN}^k(x)$ *of an instance x is its out-degree in G_{TN}^k, i.e. the number of time-series that are accessible from x in G_N^k.*

The skewness $S[g_{TN}^k]$ of g_{TN}^k quantifies the extent of "transitive" hubness in a data set. The indicators of "transitive" bad hubness can also be constructed analogously to the non-transitive case.

Definition 9 *The* transitive bad k-occurrence score $g_{TB}^k(x)$ *of an instance x is the number of time-series that are accessible from x in G_N^k but have a different class label.*

Definition 10 *The* normalised total transitive bad k-occurrence score $\widetilde{TBN}_k$ *is the number of bad transitive k-occurrences divided by the number of all transitive k-occurrences, i.e.*

$$\widetilde{TBN}_k = \frac{\sum_{x \in X} g_{TB}^k(x)}{\sum_{x \in X} g_{TN}^k(x)}. \tag{4}$$

3.2 Hubness-based indicators

By a hubness-based indicator of semi-supervised classification accuracy, we mean a quantity derived from the hubness present in the data set that helps to differentiate between two groups of data sets. We will refer to these group as "better" and "worse".

Our first research question is whether semi-supervised learning for a data set is preferable over traditional supervised learning. In this case, semi-supervised learners outperform supervised one on data sets in the "better" group, while the converse is true for the "worse" group.

Our second research question refers to the choice of a semi-supervised learner. Here the "better" group is defined to contain data sets where SUCCESS outperforms Wei's algorithm. Due to the relatively small number (44) of data sets on which we conducted our experiments, data sets are included regardless semi-supervised learning is preferable for them at all.

We use skewness indicators $S[g_N^k]$ and $S[g_{TN}^k]$ may be calculated *a priori* without knowing any class labels as they are based on k-occurrence scores. In contrast, normalised bad hubness indicators $\widetilde{BN}_k$ and $\widetilde{TBN}_k$ can only be determined when class labels for all the instances are known and therefore, in practical applications, are less useful for estimating semi-supervised learning accuracy.

3.3 Measuring indicator quality

We selected two methods—which we refer to as "*meta-measures*" for they quantify properties of hubness measures—of determining hubness-based indicator quality.

We determined whether the indicator value for the "better" and "worse" groups has significantly different mean by statistical two-tailed t-test. This simulates discrimination between the "better" and "worse" groups by fuzzy methods, e.g. Gaussian mixture models.

Another method we employed simulates discrimination based by simple *thresholding*. In other words, if we denote the indicator value for a particular data set X by $I(X)$ and the threshold by T, X is predicted to belong to the "better" when $I(X) \leq T$. We evaluated such clusterings generated by some threshold T with *normalised mutual information*, which is also called *symmetric uncertainty* [20]. Formally,

$$U(T) = 2\frac{H(\Omega) + H(\omega) - H(\Omega, \omega)}{H(\Omega) + H(\omega)}, \tag{5}$$

where Ω is the distribution of the "better" and "worse" groups and ω is the distribution associated with the threshold, while $H(\Omega)$, $H(\omega)$ and $H(\Omega, \omega)$ are the entropies of Ω, ω and their joint distribution.

For each evaluated indicator, we calculated the threshold T_{best} with the largest mutual information $U(T)$. We use the maximal value $U_{\max} = U(T_{\text{best}})$ as the second evaluation criterion for hubness-based indicators.

4 Experiments

We used a collection of 44 publicly available time-series data sets [7] from various real-world domains to demonstrate how hubness effects semi-supervised time-series classification. These data sets have been widely used in the literature, see e.g. [1, 4, 5, 9, 13].

We measured performance of three algorithms. Each of these algorithms used Dynamic Time Warping as the distance function and operated in an instance-based manner: (i) We selected the ordinary 1-nearest neighbour (1-NN) supervised classifier as a baseline. Thus it is possible to take into account that some data sets are more difficult to classify by instance-based methods because of cluster assumption violation phenomena, such as bad hubness. In cases where semi-supervised learning is effective, it should at least outperform this simple, albeit state-of-the-art algorithm. (ii) The first selected semi-supervised learner was the self-training method suggested by Wei [19]. (iii) We also considered SUCCESS, a semi-supervised time-series classifier with a hybrid approach of constrained clustering and nearest neighbour classification.

We selected $k = 10$ for hubness measures. This value was also used in a previous study by Radovanovic et al. [13]. Because transitive closures of such nearest-neighbour graphs would be extremely dense, we

opted for $k = 1$ in the case of transitive hubness measures. Therefore, the following hubness-based indicators were tested: $S[g_N^{10}]$, $\widetilde{BN}_{10}$, $S[g_{TN}^1]$, $\widetilde{TBN}_1$.

4.1 Comparison protocol

Experiments were ran separately on each of the 44 data sets.

The instances were partitioned into three disjoint subsets: (i) Labelled data L was 10% of the instances, the initially labelled training set of the semi-supervised algorithm. (ii) Unlabelled data U was 80% of the instances. The instances themselves were available at training time, however, their labels were not. The baseline 1-nearest neighbour learner, being a supervised algorithm, did not take advantage of this data. (iii) Test data was the remaining 10%. We evaluated the misclassification ratio of the learners using the these instances, i.e. the fraction of instances that had their class labels predicted wrongly.

This partitioning simulates a scenario in which a vast amount of time-series are available, but—due to shortage of resources—class labels can be produced for just a fraction of them. Semi-supervised learning may exploit the knowledge in the instances with missing class labels. The evaluated task was *inductive*, i.e. we wished to classify time-series entirely absent from the training stage.

We repeated each experiment 10 times with a different partitioning each time. We report the average misclassification ratio on each data set in Table 1, along with hubness measurements. The "winning" classifier with the smallest misclassification ratio is shown in bold type. We checked statistical significance of differences with a two-tailed paired t-test with confidence value $\alpha = 0.05$.

Both in the problem of choosing between supervised and semi-supervised learning and choosing between Wei's algorithm and SUCCESS data sets were sorted into "better" and "worse" groups depending on misclassification ratio comparison. We produced additional groupings with only significant differences.

4.2 Results

Table 3 shows meta-measure values for the tested data set groupings and hubness-based indicators. Skewness measures $S[g_N^{10}]$ and $S[g_{TN}^1]$ generally have lower p-values and competitive normalised mutual information values compared to bad hubness measures $\widetilde{BN}_{10}$ and $\widetilde{TBN}_1$. An exception is the case of supervised versus semi-supervised learning with only statistically significant differences, where $\widetilde{TBN}_1$ was better in discriminating between the "better" and "worse" group than any other indicator.

Interpreting the results is somewhat difficult, because in some cases the p-value based and mutual information based evaluation criteria selected different indicators as the best discriminator. Additionally, the exclusion of not highly significant differences changed meta-measures considerably. The latter effect may be attributed to the drastic reduction of data points—i.e. time series databases with classification accuracy differences—by the exclusion. A larger number of labelled time-series databases from various real-world domains would be needed for drawing stronger conclusions.

5 Conclusion and future work

We performed experiments on 44 publicly available time-series data sets concerning the relationship of hubness—a phenomenon emergent in time-series databases—and the accuracy of semi supervised learning—a learning method which could partially alleviate the need tedious and costly manual labelling of data in classification problems. We tested the use of measures derived from hubness as indicators to choose between supervised and semi-supervised learning and to choose between two time-series semi-supervised learning techniques. We found that with the exception of a single scenario, indicators that can be calculated *a priori* without manually labelling the whole data set are competitive with or better than indicators of bad hubness that can be determined only *a posteriori*.

We also proposed two novel measures of *transitive* hubness in a data set that take the properties of time-series semi-supervised learning methods into account. We found these measures competitive with the respective traditional hubness measures.

Table 1: Indicators of hubness along with misclassification ratios for a supervised baseline (1-NN) and two semi-supervised (Wei, SUCCESS) learning algorithms. **Boldface** shows absolute "winner" with smallest misclassification ration, while *italics* show the better semi-supervised algorithm. The + sign indicates statistically significance. The + sign left of the slash (/) corresponds to significant improvement over 1-NN, while + sign on the right corresponds to significant difference between the two semi-supervised algorithms.

Dataset	Number of classes	Hubness				Misclassification ratio		
		$S[g_N^{10}]$	$\widetilde{BN}_{10}$	$S[g_{TN}^{1}]$	$\widetilde{TBN}_1$	1-NN	Wei	SUCCESS
50words	50	0.659	0.362	2.535	0.197	0.439	0.436	***0.414***
Adiac	37	0.357	0.518	1.946	0.377	**0.571**	0.601	*0.595*
Beef	5	-0.248	0.620	-0.125	0.350	0.617	0.617	***0.600***
Car	4	1.567	0.392	1.768	0.243	**0.417**	0.458	*0.450*
CBF	3	1.435	0.001	4.235	0.000	**0.001**	0.005	*0.003*
ChlorineConcentration	3	0.503	0.316	2.954	0.004	0.369	0.350	***0.101***+/+
CinC ECG torso	5	0.079	0.011	1.232	0.000	0.031	0.019	***0.001***+/+
Coffee	2	-0.271	0.361	1.004	0.051	0.520	0.500	***0.460***
Cricket_X	12	0.380	0.331	1.800	0.192	0.450	0.465	***0.444***
Cricket_Y	12	0.457	0.349	1.600	0.189	**0.379**	0.433	*0.396* · /+
Cricket_Z	12	0.374	0.332	1.529	0.184	0.429	0.459	***0.423*** · /+
DiatomSizeReduction	4	0.357	0.014	1.101	0.006	0.038	0.031	***0.025***
ECG200	2	0.241	0.197	1.949	0.118	**0.180**	*0.190*	0.195
ECGFiveDays	2	-0.005	0.036	0.730	0.009	0.078	0.053	***0.030***+/ ·
FaceFour	4	0.402	0.141	1.719	0.048	0.236	***0.182***	0.200
FacesUCR	14	0.751	0.053	2.081	0.018	0.079	0.083	***0.070*** · /+
Fish	7	0.831	0.328	2.684	0.189	**0.354**	*0.403*	0.434
GunPoint	2	0.307	0.052	0.820	0.024	0.085	0.075	***0.045***
Haptics	5	0.851	0.609	3.654	0.523	**0.652**+	*0.704*	0.730
InlineSkate	7	0.420	0.593	1.979	0.380	**0.651**	0.683	*0.663*
ItalyPowerDemand	2	0.831	0.051	2.163	0.040	**0.051**	*0.066*	0.076
Lighting2	2	0.355	0.288	1.877	0.173	**0.308**	0.342	*0.317*
Lighting7	7	0.392	0.391	1.558	0.182	**0.493**	0.536	*0.529*
MALLAT	8	1.479	0.027	2.805	0.017	**0.032**	0.042	*0.037*
MedicalImages	10	0.352	0.316	1.219	0.184	0.399	0.394	***0.393***
MoteStrain	2	0.732	0.093	1.714	0.063	**0.098**	0.115	*0.107*
OliveOil	4	0.382	0.280	2.037	0.156	**0.367**	***0.367***	0.383
OSULeaf	6	0.626	0.448	1.545	0.246	0.473	0.532	***0.466*** · /+
Plane	7	-0.049	0.009	1.489	0.000	0.062	***0.038***	***0.038***
SonyAIBORobotS.	2	0.799	0.044	2.093	0.026	0.069	***0.060*** · /+	0.110
SonyAIBORobotS.II	2	0.815	0.065	2.050	0.022	**0.076**	*0.079*	0.088
StarLightCurves	3	1.050	0.090	4.353	0.126	**0.098**+	*0.140* · /+	0.200
SwedishLeaf	15	0.969	0.235	3.451	0.220	**0.315**+	*0.364*	0.379
Symbols	6	0.832	0.030	1.852	0.017	0.032	0.025	***0.019***+/ ·
SyntheticControl	6	1.400	0.020	3.940	0.010	**0.042**	0.065	*0.045*
Trace	4	0.035	0.025	1.556	0.000	0.185	0.050	***0.000***+/ ·
TwoLeadECG	2	0.399	0.003	1.846	0.001	0.012	0.003	***0.001***+/ ·
TwoPatterns	4	1.007	0.001	4.901	0.000	0.008	***0.000***+/ ·	***0.000***+/ ·
uWaveGestureX	8	0.773	0.234	2.501	0.222	**0.254**+	*0.284*	0.286
uWaveGestureY	8	0.627	0.314	2.068	0.258	**0.336**+	*0.377*	0.377
uWaveGestureZ	8	0.660	0.317	2.100	0.279	**0.333**+	*0.368* · /+	0.385
Wafer	2	0.307	0.005	1.154	0.004	**0.008**	0.009	*0.009*
WordsSynonyms	25	0.659	0.346	2.536	0.182	0.406	0.410	***0.382***+/ ·
Yoga	2	0.595	0.119	1.752	0.058	**0.141**	0.152	*0.151*

Table 2: Summary of experiment outcomes. Underlined algorithm is compared to the other.

		Better	Equal	Worse
Supervised versus	All	21	1	22
semi-supervised	Significant only	8	30	6
Wei versus	All	29	2	13
SUCCESS	Significant only	8	29	7

Figure 1: Data sets with differences between the considered supervised and semi-supervised learning algorithms. A single data point is a data set, while the horizontal and vertical axes show the values of k-occurrence and bad k-occurrence based indicators, respectively. **Bold** symbols indicate statistical significance ($\alpha = 0.05$). Semi-supervised learning is considered better than supervised if at least one semi-supervised algorithm performed better than supervised 1-NN. *Optimal threshold* provides maximum normalised mutual information selection of data sets where the underlined learner is beneficial (thresholds are selected one by one, i.e. only one indicator is used for separation at a time).

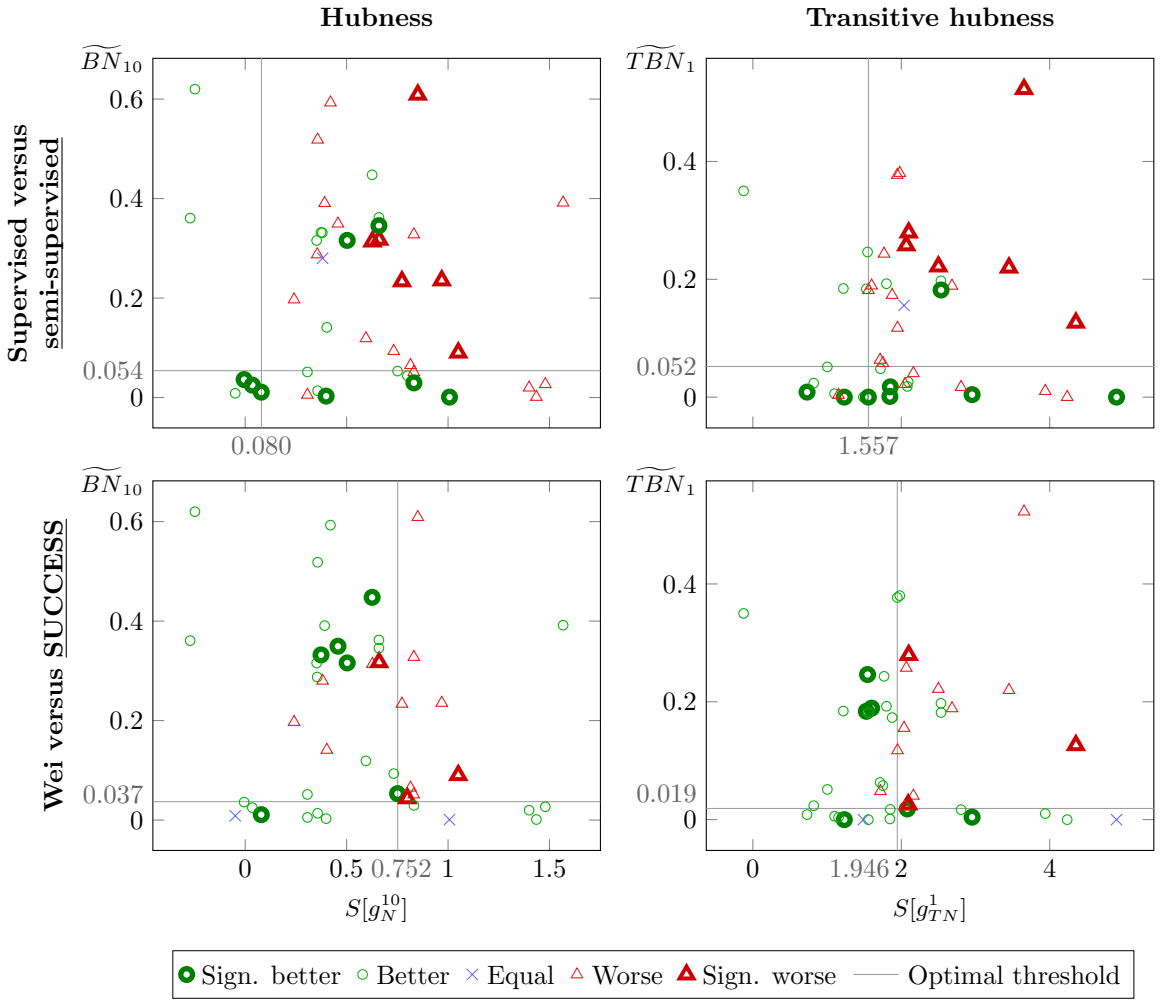

Table 3: Hubness-based indicators' power of discrimination between the "better" and "worse" group. The values in **boldface** are the best meta-measure values for a given group of experiments.

(a) Supervised verus semi-supervised

	All differences				Significant only			
	$S[g_N^{10}]$	$\widetilde{BN}_{10}$	$S[g_{TN}^1]$	$\widetilde{TBN}_1$	$S[g_N^{10}]$	$\widetilde{BN}_{10}$	$S[g_{TN}^1]$	$\widetilde{TBN}_1$
Mean "better"	0.378	0.183	1.734	0.083	0.439	0.096	2.201	0.027
Mean "worse"	0.779	0.238	2.423	0.168	0.822	0.300	3.021	0.271
p-value	**0.001**	0.347	0.024	0.030	0.030	0.042	0.194	**0.005**
Optimal threshold	0.080	0.054	1.557	0.052	0.503	0.037	1.852	0.018
Norm. mutual inf.	0.204	0.071	**0.248**	0.116	0.410	0.529	0.410	**0.695**

(b) Wei versus SUCCESS

	All differences				Significant only			
	$S[g_N^{10}]$	$\widetilde{BN}_{10}$	$S[g_{TN}^1]$	$\widetilde{TBN}_1$	$S[g_N^{10}]$	$\widetilde{BN}_{10}$	$S[g_{TN}^1]$	$\widetilde{TBN}_1$
Mean "better"	0.527	0.222	1.812	0.116	0.465	0.252	1.824	0.107
Mean "worse"	0.710	0.223	2.525	0.171	0.837	0.150	2.848	0.144
p-value	0.101	0.986	**0.016**	0.229	**0.056**	0.405	0.304	0.694
Optimal threshold	0.752	0.037	1.946	0.019	0.626	0.318	2.082	0.019
Norm. mutual inf.	0.154	0.168	**0.325**	0.210	**0.607**	0.274	**0.607**	0.274

A more extensive study with a larger number of databases from various other domains could be more decisive concerning which indicator to apply when choosing an appropriate semi-supervised time-series learner. Another scope of possible future work is the combination of different indicators, both based on hubness and other properties of time-series databases (e.g. number of instances, number of classes, or average length of time-series).

While hubness-based indicators could potentially enhance semi-supervised learning in any database with high (intrinsic) dimensionality, in this study we focused on time-series data sets. Future work could address hubness-based indicators in various other databases.

References

[1] K. Buza, A. Nanopoulos, and L. Schmidt-Thieme, INSIGHT: Efficient and effective instance selection for time-series classification, in *PAKDD (2)*, Lecture Notes in Computer Science, Springer (2011) **6635**, pp. 149–160

[2] K. Buza, A. Nanopoulos, L. Schmidt-Thieme, and J. Koller, Fast classification of electrocardiograph signals via instance selection, in *HISB*, IEEE (2011), pp. 9–16

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press 3rd ed. (2009)

[4] B. Csatári and Z. Prekopcsák, Class-based attribute weighting for time series classification (2010)

[5] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh, Querying and mining of time series data: experimental comparison of representations and distance measures, *PVLDB* (2008) **1**, pp. 1542–1552

[6] C. Gruber, M. Coduro, and B. Sick, Signature verification with dynamic RBF networks and time series motifs, in *Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule (France) Université de Rennes 1 Suvisoft (2006)

[7] E. J. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana, The UCR Time Series Classification/Clustering Homepage, `http://www.cs.ucr.edu/~eamonn/time_series_data/` (2006)

[8] H. A. Kestler, J. M. Kraus, G. Palm, and F. Schwenker, On the effects of constraints in semi-supervised hierarchical clustering, in *ANNPR*, Lecture Notes in Computer Science, Springer (2006) **4087**, pp. 57–66

[9] K. Marussy and K. Buza, SUCCESS: A new approach for semi-supervised classification of time-series, in *ICAISC* (2013), Paper in press.

[10] G. I. Nagy and K. Buza, SOHAC: Efficient storage of tick data that supports search and analysis, in *ICDM*, Lecture Notes in Computer Science, Springer (2012) **7377**, pp. 38–51

[11] M. N. Nguyen, X. Li, and S.-K. Ng, Positive unlabeled leaning for time series classification, in *IJCAI*, IJCAI/AAAI (2011), pp. 1421–1426

[12] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, Hubs in space: Popular nearest neighbors in high-dimensional data, *Journal of Machine Learning Research* (2010) **11**, pp. 2487–2531

[13] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, Time-series classification in many intrinsic dimensions, in *SDM*, SIAM (2010), pp. 677–688

[14] C. A. Ratanamahatana and D. Wanichsan, Stopping criterion selection for efficient semi-supervised time series classification, in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Studies in Computational Intelligence, Springer (2008), pp. 1–14

[15] H. Sakoe and S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, *Acoustics, Speech and Signal Processing, IEEE Transactions on* (1978) **26**, pp. 43 – 49

[16] M. Seeger, Learning with labeled and unlabeled data, tech. rep., University of Edinburgh (2001)

[17] A. Singh, R. D. Nowak, and X. Zhu, Unlabeled data: Now it helps, now it doesn't, in *NIPS*, Curran Associates, Inc. (2008), pp. 1513–1520

[18] K. Wagstaff and C. Cardie, Clustering with instance-level constraints, in *ICML*, Morgan Kaufmann (2000), pp. 1103–1110

[19] L. Wei and E. J. Keogh, Semi-supervised time series classification, in *KDD*, ACM (2006), pp. 748–753

[20] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Burlington, MA 3 ed. (2011)

[21] D. Yarowsky, Word-sense disambiguation using statistical models of Roget's categories trained on large corpora, in *COLING* (1992), pp. 454–460

[22] S. Zhong, Semi-supervised sequence classification with HMMs, *IJPRAI* (2005) **19**, pp. 165–182

[23] X. Zhu, Semi-supervised learning literature survey (2006)