

SUCCESS: A New Approach for Semi-Supervised Classification of Time-Series

Kristóf Marussy and Krisztian Buza

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
1117 Budapest, Magyar tudósok körútja 2.
kris7topher@gmail.com, buza@cs.bme.hu
<http://www.cs.bme.hu>

Abstract. The growing interest in time-series classification can be attributed to the intensively increasing amount of temporal data collected by widespread sensors. Often, human experts may only review a small portion of all the available data. Therefore, the available labeled data may not be representative enough and semi-supervised techniques may be necessary. In order to construct accurate classifiers, semi-supervised techniques learn both from labeled and unlabeled data. In this paper, we introduce a novel semi-supervised time-series classifier based on constrained hierarchical clustering and dynamic time warping. We discuss our approach in the framework of graph theory and evaluate it on 44 publicly available real-world time-series datasets from various domains. Our results show that our approach substantially outperforms the state-of-the-art semi-supervised time-series classifier. The results are also justified by statistical significance tests.

Keywords: time-series, semi-supervised classification, constrained clustering, hubs, dynamic time warping

1 Introduction

In the last decades, various types of sensors became cheaper and spread widely. Most of them record the values of some attributes continuously over time which results in extremely high number of very large time series. While such huge amounts of temporal data have never been seen before, they motivate the growing interest in time-series research. In the financial domain, for example, due to their huge volume, even storage of temporal data is challenging [14]. In general, one of the most prominent problems associated with temporal data is *classification* of time-series which is the common theoretical background of various recognition and prediction tasks ranging from handwriting, speech [20] and sign language recognition over signature verification [8] to problems related to medical diagnosis such as classification of electroencephalogram (EEG, "brain wave") and electrocardiograph (ECG) signals [2].

While the amount of temporal data grows drastically, in many cases, human experts only have the chance to review and label a small portion of all the

available data. Therefore, the labeled data may not be representative enough which may result in suboptimal classifiers. This problem is amplified by the high intrinsic dimensionality of time-series [16],[17]. As in high dimensional spaces the data becomes inherently sparse – a phenomenon often referred to as the curse of dimensionality – it is even more difficult to find a representative training set. In order to alleviate this problem, besides learning from the labeled data, we aim to use additional unlabeled data to construct more accurate time-series classifiers.

In this paper, we introduce a novel semi-supervised time-series classifier based on constrained hierarchical clustering [13] and dynamic time warping [20]. We call our approach **SUCCESS: Semi-sUpervised ClassifiCation of timE SerieS**. We discuss semi-supervised classification in the framework of graph theory: in particular, we show that semi-supervised classification is analogous to the minimal spanning tree problem. We explain our algorithm within this framework and explain its differences to the state-of-the-art semi-supervised time-series classifier. We evaluate our approach on 44 publicly available real-world time-series datasets from various domains. Our results show that our approach substantially outperforms the state-of-the-art semi-supervised time-series classifier. The results are also justified by statistical significance tests.

The remainder of the paper is organized as follows. In Section 2, we introduce the field of semi-supervised time-series classification and review the most important related works. Section 3 presents our approach followed by the experiments in Section 4. We conclude in Section 5.

2 Background

Both semi-supervised learning and time-series classification have been actively researched in the last decades. From the point of view of our current study, most relevant works deal with constrained clustering, cluster-and-label paradigm, self-training and semi-supervised classification of time-series. We will review these fields in the subsequent sections.

For an overview of further semi-supervised techniques and time-series classification approaches we refer to [3], [21], [25] and the references therein.

2.1 Constrained clustering

With clustering the data, we mean the automatic identification of groups of similar instances. Such groups are called clusters. In case of constrained clustering, the algorithm is provided with some pieces of *a priori* information in the form of *cannot-link-constraints* (or *must-link-constraints*) that describe that some instances *can not be* (or *must be*) in the same cluster. In case of *hierarchical agglomerative clustering* (HAC) algorithms, each instance initially belongs to a separate cluster. Clusters are then merged in an iterative process. In each iteration, the two most similar clusters are merged. The process is finished when the number of clusters has reached the expected number of clusters (or when

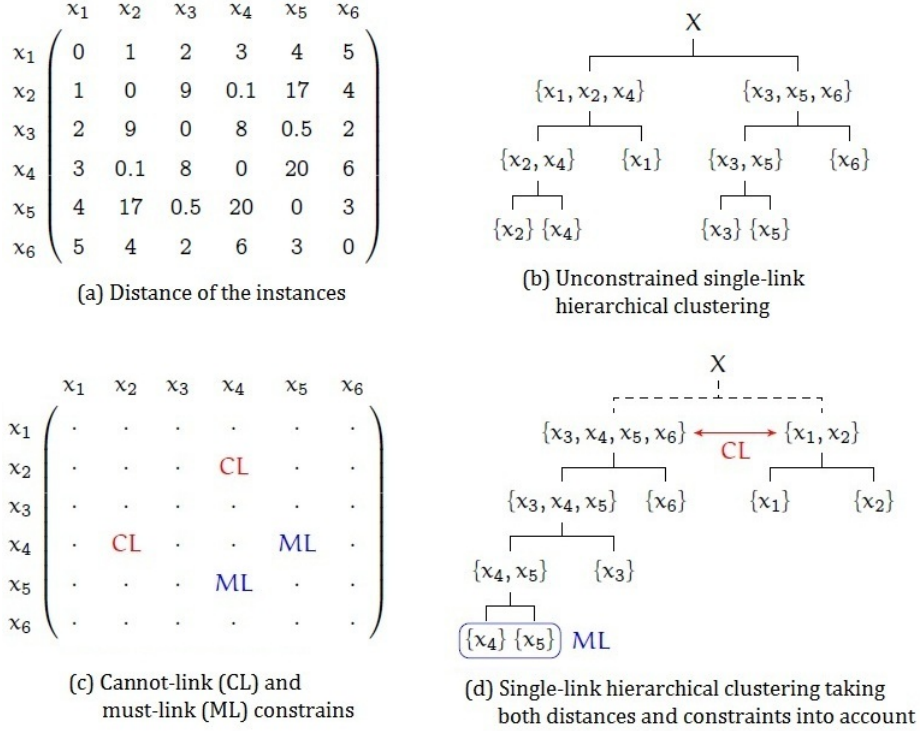


Fig. 1. Unconstrained and constrained single-link hierarchical agglomerative clustering with the dendrograms illustrating the merge steps performed during the iterative process

the two most similar clusters are too far from each other respectively). The expected number of clusters (or the distance threshold) is an external parameter set by the user. In case of *single link*, the similarity of two clusters is determined by the distance of their closest instances. Must-link (ML) and cannot-link (CL) constraints were shown to improve clustering accuracy and robustness [11], [13] compared to the case of unconstrained hierarchical clustering. Unconstrained and constrained single-link hierarchical agglomerative clustering algorithms are illustrated in Figure 1.

2.2 Cluster-and-label

In the cluster-and-label approach, unconstrained or constrained clustering is performed first. Clusters are then mapped to classes by some algorithm. A possible mapping can be constructed by majority vote, i.e., each cluster gets mapped to the class of which the most labeled instances it contains.

Cluster-and-label performs well if the particular clustering algorithm captures the true structure of the data. Dara et al. [5] and Demiriz et al. [6] applied

```

SELF-TRAINING( $L, U$ )
1  $L_0 = L$ 
2  $U_0 = U$ 
3  $t = 0$ 
4 repeat
5    $M = \text{SUPERVISED-LEARNING}(L_t)$ 
6    $x_{\text{best}} = \arg \max_{x \in U_t} \text{CERTAINTY}(M, x)$ 
7    $\hat{y} = \text{CLASSIFY}(M, x_{\text{best}})$ 
8    $L_{t+1} = L_t \cup \{(x_{\text{best}}, \hat{y})\}$ 
9    $U_{t+1} = U_t \setminus \{x_{\text{best}}\}$ 
10   $t = t + 1$ 
11 until  $|U_t| = 0$ 
12 return  $M$ 

```

Fig. 2. Simple self-training algorithm.

the cluster-and-label paradigm with self-organizing maps and genetic algorithms for semi-supervised classification.

2.3 Self-training

Self-training is one of the most commonly used semi-supervised algorithm. Self-training is a wrapper method around a supervised classifier, i.e., one may use self-training to enhance various classifiers. To apply self-training, for each instance x to be classified, besides its predicted class label, the classifier must be able to output a certainty score, i.e., an estimation of how likely the predicted class label is correct.

Self-training is an iterative process during which the set of labeled instances is grown until all the instances become labeled. Let L_1 denote the set of initially labeled instances, and, more generally, let L_t denote the set of labeled instances in the t -th iteration ($t \geq 1$). In each iteration of self-training, the base classifier is trained on the labeled set L_t . Then, the base classifier is used to classify the unlabeled instances. Finally, the instances with highest certainty scores are selected. These instances, together with their predicted labels, are added to the set of labeled instances, in order to construct L_{t+1} the set of labeled instance for the next iteration. In the simplest case, one instance is added in each iteration, the pseudocode of this algorithm is shown in Figure 2. In context of nearest neighbor classification, the algorithm are illustrated in Figure 3. Other variants of self-training include e.g. Yarowsky’s algorithm [23].

2.4 Semi-supervised classification of time-series

One of the most surprising recent results in the time-series classification domain is that simple nearest neighbor classifiers using a special distance measure called dynamic time warping (DTW) are generally competitive with, if not better than, many complicated approaches [7]. Therefore, we build our approach on

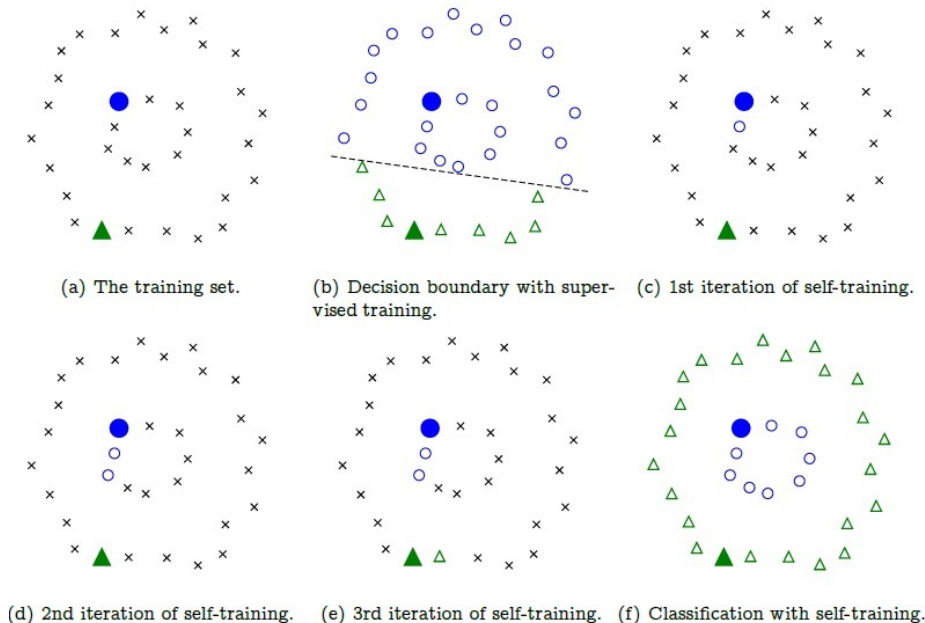


Fig. 3. Self-training with nearest neighbor. There are two classes, circles and triangles. Bold symbols correspond to instances of the initially labeled training set L_1 , while unlabeled instances are marked with crosses, see Subfigure (a). Subfigures (c) – (e) show the first three iterations of Self-Training. The final output of self-training is shown in Subfigure (f).

the DTW-based nearest neighbor classification of time-series. DTW was originally introduced for speech recognition [20]. The key feature of DTW is that it allows for siftings and elongations while it compares two time-series. We refer to [3] for a detailed description of DTW.

Despite its relevance, there are just a few works on the semi-supervised classification of time-series. Wei and Keogh proposed a self-training based approach [22] which was enhanced by Ratanamahatana et al. [18] by the introduction of a new stopping criterion. Nguyen et al. used k -Means and principal component analysis for semi-supervised time-series classification [15]. All these works focused on the case when labeled instances are only available for one of the classes. In contrast, we assume that there are some labeled instances for each class, like in the example shown in Figure 3. Furthermore, our approach is much simpler than that of Nguyen et al., as we do not use dimensionality reduction. Instead, we compare time-series directly by DTW. Zhong used self-training with Hidden Markov Models [24] for the semi-supervised classification of small time-series datasets. In contrast to previous works, we base our approach on the cluster-and-label paradigm and use a constrained hierarchical single link clustering algorithm.

3 Our approach: SUCCESS

We consider the semi-supervised classification problem, in which a set of labeled time-series $L = \{(x_i, y_i)\}_{i=1}^l$ and a set of unlabeled time-series $U = \{x_i\}_{i=l+1}^n$ is available as train data to a classifier. The labeled time series (elements of L) are called *seeds*. We wish to construct a classifier that can accurately classify any time-series, i.e., not only elements of U . For this problem, we propose a novel semi-supervised time-series classification approach, called SUCCESS. SUCCESS has the following phases:

1. The labeled and unlabeled instances of the training set are clustered with constrained single-linkage hierarchical agglomerative clustering. While doing so, we measure the distance of two instances (time-series) as their DTW-distance and we include cannot-link constraints for *each* pair of labeled seeds even if the both seeds have the same class labels.
2. The resulting top-level clusters are labeled by their corresponding seeds.
3. The final classifier is 1-nearest neighbor trained on the resulting labeled data. This classifier can be applied to *unseen* test data.

While the components of the algorithm (like DTW or single-link clustering) are well-known, we emphasize that the algorithm as a whole is new for semi-supervised time-series classification. Next, we explain the difference between self-training and our approach using the framework of graph theory.

3.1 A graph-theoretic view of semi-supervised time-series classification

The presented semi-supervised time-series classification algorithms can be considered as algorithms that aim at finding the minimum spanning tree of a graph.

Consider the set of all, labeled and unlabeled, train instances $X = L \cup U = \{x_i\}_{i=1}^n$. Let $G = (X, V)$ be an undirected *complete* graph, the vertices of which correspond the instances of the database, and the weights of the edges correspond the distance of two instances (DTW-distance in our case): $w_{i,j} = d(x_i, x_j)$.

We define a *spanning forest* as a set of trees $\mathcal{T} = \{T_i\}_{i=1}^l$ that satisfy the following properties:

- The trees are disjoint, i.e. $\forall i : x_i \in V(T_a) \wedge x_i \in V(T_b) \Rightarrow a = b$.
- The trees together span the entire set of instances, i.e. $\bigcup_{i=1}^l V(T_i) = X$.
- The i th tree contains the i labeled instances, i.e. $\forall 1 \leq i \leq l : x_i \in V(T_i)$.

Note that we consider forests where the number of trees equals the number of labeled instances, and each tree corresponds to a labeled instance.

A spanning forest is a *minimum spanning forest* if the sum of its edge weights $W(\mathcal{T}) = \sum_{i=1}^l \sum_{e \in E(T_i)} w(e)$ is minimal.

Let us define the graph $G^* = (X \cup \{\star\}, E^*)$, which is an extension of G with a super-vertex \star . This super-vertex \star is connected to the labeled examples with

0-weight edges, i.e., $E^* = E \cup \{\{x_i, \star\} : 1 \leq i \leq l\}$, and $w(\{x_i, \star\}) = 0$ for all vertices x_i .

Consider the tree T^* which contains \star and the new edges from \star , and the union of the trees in a minimum spanning forest \mathcal{T} of G . The sum of edge-weights in T^* is not greater than that of a minimum spanning tree of G^* , therefore, T^* is a minimum spanning tree of G^* .

The self-training algorithm with the 1-nearest neighbor classifier can be viewed as a specific way of finding a minimum spanning tree of G^* . In particular, if all the edge weights $w_{i,j}$ in G are strictly positive, except the weights of edges that connect \star and the seeds, instance based self-training is equivalent to running Prim’s algorithm [4] with \star as the root node. In the first l iterations, the algorithm adds the labeled instances $\{x_i\}_{i=1}^l$ to the tree. In every subsequent iteration, the set nodes in the growing tree equals the set of already labeled instances.

Therefore, we can see that self-training corresponds to Prim’s minimal spanning tree algorithm. Next, we show that our approach, SUCCESS, in contrast, corresponds Kruskal’s algorithm [4].

Notice that the forest which is gradually joined by Kruskal’s algorithm is a set of clusters at some level of a *single-linkage* (SLINK) *hierarchical agglomerative clustering* dendrogram.

Due to \star and the 0-weight edges connecting \star with the labeled instances, in the first l iterations, Kruskal’s greedy algorithm will select all the labeled instances into the minimal spanning tree. Therefore, after the l -th iteration, the tree has l branches, each one corresponding to a labeled instance. In the subsequent iterations the tree grows along these branches, however, no new branch is created from node \star as all of the edges of \star are already contained in the tree after the l -th iteration. We call the aforementioned branches *main branches*. When the algorithm terminates, each of the main branches corresponds to a cluster. This is analogous to having cannot-link constraints in the hierarchical clustering between each pair of labeled instances.

4 Experiments

In order to assist reproducibility, we provide a detailed description of the experiments we performed.

Methods – We compared our approach, SUCCESS, against Wei’s approach [22], which is one the most prominent state-of-the-art semi-supervised time-series classifiers. While Wei’s approach is based on self-training, SUCCESS is based on the cluster-and-label paradigm as explained before.

Datasets – We evaluated both Wei’s approach and SUCCESS on 44 publicly available real-world datasets from the UCR time-series repository [10]. These datasets originate from various domains ranging from handwriting recognition [19] and user identification with graphical passwords [1] over biological

Table 1. Summary of the results. The number of datasets on which our approach wins/looses against Wei’s approach. The numbers in parenthesis show how many times the difference is statistically significant.

	Unlabeled Train	Test
Wins	29 (14)	30 (6)
Ties	-	2
Looses	15 (5)	12 (3)

shape recognition [9] and electrocardiograph classification to gesture recognition [12]. The names of these datasets as well as the number of classes are shown in the first two columns of Table 2.

Comparison Protocol – We run experiments separately on each of the 44 datasets. For each experiment, we split the data into 3 disjoint subsets: the first one, denoted as L , contains around 10% of the instances. The second split, U , contains around 80% of the instances while the remaining instances are in the third split. We used the first split, L , as the initially labeled instances of the semi-supervised algorithm. U served as the set of unlabeled training instances the labels of which were unavailable to the algorithm but the instances themselves were available at training time. The third split was used as test data that was completely unavailable to the algorithm at the training time. The instances of the test set were classified one by one without updating the classification model. We measured the performance both on the set of unlabeled training instances (U) and on the test set. This allowed us to simulate two, slightly different, real-world situations. Measuring the performance on U corresponds to the case of having a large set of unlabeled instances and a small set of labeled instances with the goal of correctly classifying the unlabeled instances. Measuring the performance on the test set simulates the situation where we have a large set of unlabeled instances and a small set of labeled instances and we aim at constructing a classifier that should be used to classify *new* instances that may be different from the unlabeled instances available at training time.

We used misclassification ratio to measure the performance of the baseline and our approach. For each dataset, we repeated all experiments 10 times, i.e., we split the data into the above three splits 10 times by random and measured the performances of our approach and the baseline. In Table 2, we report average performances. In order to check whether the differences are statistically significant, we used t-tests at significance level $\alpha = 0.05$.

Results – We show the average misclassification ratios on the 44 datasets in Table 2. We use the + symbol to denote that an approach statistically significantly outperformed its competitor. The results of our experiments are summarized in Table 1. As it can be seen, in clear majority of the datasets, our approach, SUCCESS, outperforms Wei’s approach. For each dataset, we also performed

Table 2. Misclassification ratio of Wei’s approach and SUCCESS. Bold font denotes the winner, + denotes statistically significant difference. (While determining the winner, we took the non-shown digits into account as well.)

Dataset	Number of classes	Unlabeled train		Test	
		Wei	SUCCESS	Wei	SUCCESS
50 Words	50	0.432	0.398+	0.436	0.414
Adiac	37	0.607	0.582+	0.601	0.595
Beef	5	0.683	0.656	0.617	0.600
Car	4	0.484	0.457	0.458	0.450
CBF	3	0.007	0.002	0.005	0.003
ChlorineConcentration	3	0.373	0.062+	0.350	0.101+
CinC ECG Torso	4	0.021	0.001+	0.019	0.001+
Coffee	2	0.429	0.368+	0.460	0.440
Cricket_X	12	0.477	0.425+	0.465	0.444
Cricket_Y	12	0.463	0.405+	0.433	0.396+
Cricket_Z	12	0.443	0.395+	0.459	0.423+
DiatomSizeReduction	4	0.018	0.017	0.031	0.025
ECG200	2	0.237	0.225	0.239	0.195
ECGFiveDays	2	0.051	0.021+	0.053	0.030
FaceFour	4	0.201	0.191	0.182	0.200
FacesUCR	14	0.080	0.062+	0.083	0.070+
Fish	7	0.424	0.449	0.403	0.434
GunPoint	2	0.089	0.039	0.075	0.045
Haptics	5	0.671+	0.706	0.704	0.730
InlineSkate	7	0.693	0.679	0.683	0.663
ItalyPowerDemand	2	0.063	0.073	0.066	0.076
Lighting2	2	0.355	0.322	0.342	0.317
Lighting7	7	0.463	0.477	0.536	0.529
Mallat	8	0.042	0.041	0.042	0.037
MedicalImages	10	0.379	0.386	0.394	0.393
MoteStrain	2	0.124	0.129	0.115	0.107
OliveOil	4	0.300	0.315	0.367	0.383
OSULeaf	6	0.550	0.512+	0.532	0.466+
Plane	7	0.050	0.049	0.038	0.038
SonyAIBORobotS.	2	0.052+	0.090	0.060+	0.110
SonyAIBORobotS.II	2	0.088	0.094	0.079	0.087
StarLightCurves	3	0.119+	0.200	0.140+	0.200
SwedishLeaf	15	0.330+	0.369	0.364	0.379
Symbols	6	0.033	0.022+	0.025	0.019
SyntheticControl	6	0.051	0.029	0.065	0.045
Trace	4	0.054	0.001+	0.050	0.000
TwoLeadECG	2	0.004	0.001	0.003	0.001
TwoPatterns	4	0.000	0.000	0.000	0.000
uWaveGestureX	8	0.276	0.284	0.284	0.286
uWaveGestureY	8	0.356	0.368	0.377	0.377
uWaveGestureZ	8	0.359+	0.378	0.368+	0.385
Wafer	2	0.009	0.009	0.009	0.009
WordsSynonyms	25	0.414	0.378+	0.410	0.382
Yoga	2	0.148	0.149	0.152	0.151

experiments with 20% of the data being labeled train data L (and 70% being the unlabeled train data U respectively) and we observed very similar results.

5 Conclusion

In this paper, we proposed SUCCESS, a novel semi-supervised time-series classifier. We discussed the relation between the minimal spanning tree problem and semi-supervised classification. We pointed out the analogy between a state-of-the-art semi-supervised time-series classifier and Prim's algorithm as well as our approach and Kruskal's greedy algorithm. We performed exhaustive experimental evaluation that showed that our approach is able to outperform that state-of-the-art semi-supervised time-series classifier on many real-world datasets.

Besides time-series, huge amounts of other types of sequential data are being collected, e.g., DNA-sequence of a persons and other organisms. Therefore, as future work, one may consider to use similar approaches for the semi-supervised classification of other types of sequential data.

Acknowledgments. The work reported in the paper has been developed in the framework of the project "Talent care and cultivation in the scientific workshops of BME". This project is supported by the grant TÁMOP-4.2.2.B-10/1-2010-0009. We acknowledge the DAAD-MÖB Researcher Exchange Program.

References

1. B. Malek, M.O., Saddik, A.E.: Novel shoulder-surfing resistant haptic-based graphical password. In: Proceedings of EuroHaptics06 (2006)
2. Buza, K., Nanopoulos, A., Schmidt-Thieme, L., Koller, J.: Fast Classification of Electrocardiograph Signals via Instance Selection. In: First IEEE Conference on Healthcare Informatics, Imaging, and Systems Biology (HISB) (2011)
3. Buza, K.A.: Fusion Methods for Time-Series Classification. Ph.D. thesis (2011)
4. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms. The MIT Press (2001)
5. Dara, R., Kremer, S., Stacey, D.: Clustering unlabeled data with soms improves classification of labeled real-world data. In: Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on. vol. 3, pp. 2237–2242 (2002)
6. Demiriz, A., Bennett, K., Embrechts, M.J.: Semi-supervised clustering using genetic algorithms. In: In Artificial Neural Networks in Engineering (ANNIE-99). pp. 809–814. ASME Press (1999)
7. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.J.: Querying and mining of time series data: experimental comparison of representations and distance measures. PVLDB 1(2), 1542–1552 (2008)
8. Gruber, C., Coduro, M., Sick, B.: Signature Verification with Dynamic RBF Networks and Time Series Motifs. In: 10th International Workshop on Frontiers in Handwriting Recognition (2006)

9. Jalba, A., Wilkinson, M., Roerdink, J., Bayer, M., Juggins, S.: Automatic diatom identification using contour analysis by morphological curvature scale spaces. *Machine Vision and Applications* 16, 217–228 (2005), <http://dx.doi.org/10.1007/s00138-005-0175-8>
10. Keogh, E.J., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR Time Series Classification/Clustering Homepage (2006), http://www.cs.ucr.edu/~eamonn/time_series_data/
11. Kestler, H.A., Kraus, J.M., Palm, G., Schwenker, F.: On the effects of constraints in semi-supervised hierarchical clustering. In: Schwenker, F., Marinai, S. (eds.) *ANNPR. Lecture Notes in Computer Science*, vol. 4087, pp. 57–66. Springer (2006)
12. Ko, M.H., West, G., Venkatesh, S., Kumar, M.: Using dynamic time warping for online temporal fusion in multisensor systems. *Information Fusion* 9(3), 370 – 388 (2008), <http://www.sciencedirect.com/science/article/pii/S1566253506000674>, special Issue on Distributed Sensor Networks
13. Miyamoto, S., Terami, A.: Semi-supervised agglomerative hierarchical clustering algorithms with pairwise constraints. In: *FUZZ-IEEE*. pp. 1–6. IEEE (2010)
14. Nagy, G., Buza, K.: Sohac: Efficient storage of tick data that supports search and analysis. In: Perner, P. (ed.) *Advances in Data Mining. Applications and Theoretical Aspects, Lecture Notes in Computer Science*, vol. 7377, pp. 38–51. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-31488-9_4
15. Nguyen, M.N., Li, X., Ng, S.K.: Positive unlabeled learning for time series classification. In: Walsh, T. (ed.) *IJCAI*. pp. 1421–1426. *IJCAI/AAAI* (2011)
16. Radovanovic, M., Nanopoulos, A., Ivanovic, M.: Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research* 11, 2487–2531 (2010)
17. Radovanovic, M., Nanopoulos, A., Ivanovic, M.: Time-series classification in many intrinsic dimensions. In: *SDM*. pp. 677–688. *SIAM* (2010)
18. Ratanamahatana, C.A., Wanichsan, D.: Stopping criterion selection for efficient semi-supervised time series classification. In: Lee, R.Y. (ed.) *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Studies in Computational Intelligence*, vol. 149, pp. 1–14. Springer (2008)
19. Rath, T., Manmatha, R.: Word Image Matching using Dynamic Time Warping. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. vol. 2, pp. II–521. IEEE (2003)
20. Sakoe, H., Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *Acoustics, Speech and Signal Processing* 26(1), 43–49 (1978)
21. Seeger, M.: Learning with labeled and unlabeled data. Tech. rep., University of Edinburgh (2001)
22. Wei, L., Keogh, E.J.: Semi-supervised time series classification. In: Eliassi-Rad, T., Ungar, L.H., Craven, M., Gunopoulos, D. (eds.) *KDD*. pp. 748–753. *ACM* (2006)
23. Yarowsky, D.: Word-sense disambiguation using statistical models of roget’s categories trained on large corpora. In: *COLING*. pp. 454–460 (1992)
24. Zhong, S.: Semi-supervised sequence classification with hmms. *IJPRAI* 19(2), 165–182 (2005)
25. Zhu, X.: Semi-supervised learning literature survey (2007)