2014-08

Time series transductive classification on imbalanced data sets: an experimental study

International Conference on Pattern Recognition, 22nd, 2014, Stockholm.
http://www.producao.usp.br/handle/BDPI/48634

2014 22nd International Conference on Pattern Recognition

Time series transductive classification on imbalanced data sets: an experimental study

Celso A. R. de Sousa, Vinícius M. A. Souza, Gustavo E. A. P. A. Batista
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo, São Carlos, Brazil
email: { sousa, vsouza, gbatista }@icmc.usp.br

***Abstract*—Graph-based semi-supervised learning (SSL) algorithms perform well on a variety of domains, such as digit recognition and text classification, when the data lie on a low-dimensional manifold. However, it is surprising that these methods have not been effectively applied on time series classification tasks. In this paper, we provide a comprehensive empirical comparison of state-of-the-art graph-based SSL algorithms with respect to graph construction and parameter selection. Specifically, we focus in this paper on the problem of time series transductive classification on imbalanced data sets. Through a comprehensive analysis using recently proposed empirical evaluation models, we confirm some of the hypotheses raised on previous work and show that some of them may not hold in the time series domain. From our results, we suggest the use of the Gaussian Fields and Harmonic Functions algorithm with the mutual k-nearest neighbors graph weighted by the RBF kernel, setting $k = 20$ on general tasks of time series transductive classification on imbalanced data sets.**

I. Introduction

In the last few years, we have witnessed a huge increase of interest on time series mining. Such an interest is justified by the innumerous applications that generate data across time. The recent advances in technology have made available a myriad of data acquisition equipment at a fraction of the cost of one decade ago. For instance, low-cost electrocardiogram and electroencephalogram equipment has reached the US$100 mark; motion tracking sensors, such as accelerometers, gyroscopes and GPS devices are available in virtually every mobile phone; etc. These digital devices are collecting huge amounts of time series data at increasing rates.

Unfortunately, most of these time series are unlabeled by nature. Due to time and cost constraints, human experts can label just a tiny fraction of these huge data volumes. However, semi-supervised methods can avail of labeled and unlabeled examples to leverage off classification performance. One of the most prominent approaches for *semi-supervised learning* (SSL) are the graph-based transductive classifiers. These methods learn from a weighted graph generated using both labeled and unlabeled examples without providing generalization for the entire sample space. Theoretically speaking, these methods are incapable to perform induction. However, by applying classifiers based on *k-nearest neighbors* (kNN) in the transductive solution, we can easily classify out-of-sample examples. For simplicity, we assume in this paper that *transductive classification* is a synonym of SSL; hence, the transductive classifiers are called here as SSL algorithms. A comprehensive discussion about transductive vs. semi-supervised learning can be found in [1].

Graph-based SSL algorithms have been effectively applied in a variety of domains [2]. However, it is surprising that these methods have not been extensively evaluated in the time series domain. Since similarity-based methods (using, e.g., the *Dynamic Time Warping* (DTW) distance) are effective for time series classification [3]–[4], the use of graph-based methods can also be effective in this domain because the weighted graph encodes similarities between neighbored examples.

In this paper, we show that graph-based SSL algorithms are effective in the time series domain. In order to do this, we provide experiments using kNN-based graphs with DTW, frequently beating the 1NN classifier with this same distance. Specifically, we provide a comprehensive empirical comparison of state-of-the-art graph-based SSL algorithms with respect to graph construction and parameter selection on time series transductive classification tasks on imbalanced data sets.

Although the literature has several studies comparing graph-based SSL algorithms [2], [5]–[8], very few articles draw conclusions for *imbalanced*[1] data sets. In general, conclusions concerning the performance of these methods on imbalanced data are made based on the observation of a very restricted number of data sets. Therefore, we ask in this paper whether some of these hypotheses hold for time series data, and analyze them in the light of a large number of data sets from different application domains. Specifically, we analyze the hypotheses raised on [2] for imbalanced data sets.

Hypothesis 1: The graphs generated by the mutual kNN (mutKNN) graph show high instability for relatively small values of k.

Hypothesis 2: The graphs generated by mutKNN tend to give better results than those generated by other adjacency graph construction methods.

Hypothesis 3: The Robust Multi-class Graph Transduction (RMGT) [8] algorithm may not be effective on imbalanced data sets.

Through a comprehensive analysis, we concluded that: (1) Hypothesis 1 may not hold in general because the graphs generated by mutKNN showed good stability in many data sets, even for small values of k; (2) Hypothesis 2 holds on most data sets; and (3) Hypothesis 3 may only holds for data sets with *high* imbalanced ratio because RMGT achieved competitive results in many data sets.

[1] In this paper, we assume that a data set is imbalanced if the majority class has at least two times more examples than the minority class.

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.649

TABLE I. DESCRIPTION OF THE DATA SETS.

Data sets	# labeled (%)	# unlabeled (%)	series length	# classes	% minority	% majority	ID
50words	450 (49.72)	455 (50.28)	270	50	0.22	11.56	D1
Adiac	390 (49.93)	391 (50.07)	176	37	1.03	3.85	D2
ChlorineConcentration	467 (10.84)	3840 (89.16)	166	3	19.49	56.10	D3
Cinc-ECG-torso	40 (2.81)	1380 (97.19)	639	4	12.5	32.5	D4
FacesUCR	200 (8.88)	2050 (91.12)	132	14	2.0	16.50	D5
Haptics	155 (33.47)	308 (66.53)	1092	5	11.61	23.23	D6
InlineSkate	100 (15.38)	550 (84.62)	1882	7	9.0	18.0	D7
Mallat	55 (2.29)	2345 (97.71)	1024	8	3.64	20.0	D8
MedicalImages	381 (33.39)	760 (66.61)	99	10	1.58	53.28	D9
OSULeaf	200 (45.24)	242 (54.76)	427	6	7.5	26.5	D10
SonyAIBORobotSurface	20 (3.22)	601 (96.78)	70	2	30.0	70.0	D11
StarLightCurves	1000 (10.82)	8236 (89.18)	1024	3	15.2	57.3	D12
Symbols	25 (2.45)	995 (97.55)	398	6	12.0	32.0	D13
Wafer	1000 (13.95)	6164 (86.05)	152	2	9.7	90.3	D14
WordsSynonyms	267 (29.50)	638 (70.50)	270	25	0.75	22.47	D15

The remainder of this paper is organized as follows. Section II describes our experimental design. Section III analyzes our results. Finally, Section IV provides our conclusions.

II. EXPERIMENTAL DESIGN

We used a slight variation of the experimental setup in [2] for time series transductive classification. We performed experiments using benchmark data sets widely used in the time series literature and publicly available in the UCR repository[2]. We chose the 15 most imbalanced data sets from the repository which covers different application domains such as medicine, astronomy, and robotics. These data sets are described in Table I. Since we provide an experimental study on transductive classification, the training (labeled) and test (unlabeled) data are used together during the classification process. Due to reasons concerning reproducibility, we used the data splits suggested in the UCR repository.

The SSL algorithms are compared based on their error rates on the test data. In order to provide a comprehensive analysis, we used the empirical evaluation models described in [2], which are: (1) *best case analysis*, in which we compare each combination of SSL algorithm and graph construction method based on their lowest error rates with respect to all parameter values; (2) *evaluation of classifier stability*, in which we evaluate how the classifiers' performance is affected on a given graph construction method with respect to k; (3) *evaluation of graph stability*, in which we evaluate how the performance of a given SSL algorithm is affected with respect to k for each graph; (4) *evaluation of regularization parameters*, in which we evaluate how the regularization parameters affect an SSL algorithm's performance on a fixed graph.

A. Graph construction

We generate a sparse, undirected, weighted graph using both labeled and unlabeled examples from the kNN graph using the DTW distance. In order to generate a symmetric adjacency matrix, we used the following adjacency graphs: *symmetric* kNN (symKNN), mutKNN, and *symmetry favored* kNN (symFKNN) [8]. Since mutKNN may generate a graph with isolated vertices, we created an undirected edge between each isolated vertex and its nearest neighbor, as suggested in [2]. The parameter k was chosen at range $\{4, 6, 8, \cdots, 40\}$.

[2]http://www.cs.ucr.edu/~eamonn/time_series_data/

From the adjacency matrix, we applied the following weighted matrix generation methods: RBF kernel; Hein & Maier's (HM) similarity function [9]; and *Local Linear Embedding* (LLE) [10]. In order to efficiently compute the LLE method, we used the *Local Anchor Embedding* (LAE) method [11] with the symmetrization process described in [2]. We estimate the value of the parameter in the RBF kernel as suggested in [12]. We generated the combinatorial and the normalized Laplacians using the definitions in [2], which are useful in an attempt to avoid numerical instabilities while executing the SSL algorithms.

B. SSL algorithms

We provide an experimental evaluation using the following SSL algorithms: *Gaussian Fields and Harmonic Functions* (GFHF) [5], *Local and Global Consistency* (LGC) [6], RMGT [8], and *Laplacian Regularized Least Squares* (LapRLS) [7]. The regularization parameter μ in the LGC algorithm was chosen at range $\{0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10, 50, 100\}$. The regularization parameters γ_A and γ_I of LapRLS were chosen at range $\{10^{-6}, 10^{-4}, 10^{-2}, 10^{-1}, 1, 10, 100\}$, as suggested in [13]. We generated the kernel matrix for LapRLS using the RBF kernel with the same parameter estimation in [12]. For RMGT, we assumed a uniform class distribution, as suggested in [8]. We used the combinatorial Laplacian for RMGT and the normalized Laplacian for the other SSL algorithms, as suggested in [2] (see also [14]).

III. ANALYSIS OF THE RESULTS

In this section, we provide a comprehensive analysis of our results using the commonly used best case analysis as well as the empirical evaluation models proposed in [2].

A. Best case analysis

Table II shows the results for the best case analysis. Each result in this table corresponds to the best error rate for a combination of an SSL algorithm, graph construction method, and data set. The four worst results obtained by an SSL algorithm in each data set have a grey background while the best one is in bold. The best overall result on each data set is boxed. The symbol † in Table II indicates that numerical instabilities occurred for some parameter values, i.e., the algorithm found no solution for these parameter values.

TABLE II. ERROR RATES (%) OF THE SSL ALGORITHMS ON EACH DATA SET WITH RESPECT TO GRAPH CONSTRUCTION.

Data sets	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
GFHF-symKNN-RBF	19.78	37.34	7.01	0.07	3.61	55.19	59.27	5.07	20.39	35.95	2.66	6.85	2.11	0.26	**23.35**
GFHF-mutKNN-RBF	**19.34**	36.57	6.22	**0.0**	4.59	54.87	56.55	5.07	20.39	**35.54**	**0.5**	5.43	**1.51**	0.32	24.45
GFHF-symFKNN-RBF	19.56	36.06	6.59	0.07	**3.56**	56.82	59.09	**4.18**	**20.0**	35.95	2.16	6.33	2.11	0.26	23.51
GFHF-symKNN-HM	23.3	41.18	13.83	0.43	5.17	52.6	59.09	6.27	22.37	38.84	23.29	6.28	3.02	0.28	25.86
GFHF-mutKNN-HM	21.98	36.83	11.41	**0.0**	4.93	52.6	**56.36**	5.93	21.97	37.19	0.83	5.1	1.61	0.41	25.71
GFHF-symFKNN-HM	22.2	37.6	13.2	0.29	4.78	52.27	59.09	5.29	21.71	38.02	14.14	5.8	2.81	**0.23**	25.86
GFHF-symKNN-LLE	26.37	33.25	0.73	0.65	6.2	51.3	59.64	5.33	23.42	37.6	21.8	4.89	2.31	0.36	29.62
GFHF-mutKNN-LLE	23.74	33.76	**0.52**	**0.0**	5.27	52.6	56.55	6.61	22.24	39.67	1.16	5.04	1.61	0.52	28.21
GFHF-symFKNN-LLE	25.27	**31.97**	0.7	0.07	5.51	**50.97**	59.82	4.39	22.89	37.6	11.81	**4.4**	2.31	0.29	27.9
LGC-symKNN-RBF	**18.9**	39.9	7.5	**0.0**	**3.85**	55.52	57.64	4.26	**19.21**	36.78	2.0	6.22	2.41	**0.26**	**22.26**
LGC-mutKNN-RBF	**18.9**	35.29	6.67	**0.0**	4.59	55.19	**55.09**	4.26	19.61	**35.12**	**0.5**	5.23	**1.51**	0.32	23.51
LGC-symFKNN-RBF	**18.9**	37.08	7.01	**0.0**	**3.85**	55.52	57.27	**4.01**	**19.21**	35.95	2.16	5.8	2.11	**0.26**	22.73
LGC-symKNN-HM	22.64	39.39	14.11	0.14	5.41	52.27	57.27	4.82	21.58	38.84	7.65	5.65	3.12	0.28	26.02
LGC-mutKNN-HM	20.0	35.29	11.41	**0.0**	5.32	51.62	**55.09**	4.39	21.18	36.78	0.83	5.16	**1.51**	0.39	25.24
LGC-symFKNN-HM	21.1	37.85	13.44	0.14	4.78	51.62	56.55	4.35	21.32	37.6	5.99	5.08	3.02	0.28	25.24
LGC-symKNN-LLE	25.27	32.74	1.07	0.72	6.2	**50.97**	58.91	4.56	23.42	37.19	7.15	4.4	2.61	0.31	28.37
LGC-mutKNN-LLE	21.54	33.25	**0.55**	**0.0**	4.93	52.27	56.55	4.69	21.97	36.36	1.0	5.01	**1.51**	0.44	28.06
LGC-symFKNN-LLE	23.74	**31.97**	0.89	0.43	5.37	**50.97**	58.18	4.09	22.63	36.36	6.82	**3.97**	2.31	0.29	26.96
LapRLS-symKNN-RBF	18.9	36.06	7.03	**0.0**	3.61	54.55	58.36	4.61	20.26	35.54	2.0	6.39	2.11	0.26	**23.2**
LapRLS-mutKNN-RBF	**18.24**	33.76	6.25	**0.0**	3.95	54.55	58.36	2.86	**20.0**	**35.12**	**0.33**	5.38	**1.51**	**0.11**	23.51
LapRLS-symFKNN-RBF	18.9	34.53	6.64	**0.0**	**3.46**	54.55	58.55	3.92	20.13	35.54	2.16	6.31	2.11	0.26	23.67
LapRLS-symKNN-HM	20.66	37.08	13.75	0.14	5.12	52.6	57.64	5.07	22.11	35.95	8.32	6.22	3.02	0.19	24.29
LapRLS-mutKNN-HM	20.0	34.53	11.3	**0.0**	4.2	52.6	**56.36**	**2.39**	20.39	35.95	0.83	5.48	1.61	0.19	24.45
LapRLS-symFKNN-HM	20.22	36.83	13.05	0.14	4.78	51.95	57.45	4.73	21.05	35.54	6.82	6.23	2.81	0.19	24.45
LapRLS-symKNN-LLE	21.1	28.64	0.73	0.72	5.9	50.97	57.09	4.9	22.11	35.95	8.32	6.29	2.31	0.39	24.29
LapRLS-mutKNN-LLE	20.22	30.43	**0.52**	**0.0**	4.49	51.3	56.55	2.77	20.79	35.95	1.0	**5.09**	1.61	0.23	24.14
LapRLS-symFKNN-LLE	20.88	**27.88**	0.7	0.07	5.37	**50.0**	57.82	4.22	21.97	35.95	7.15	6.20	2.31	0.37	24.45
RMGT-symKNN-RBF	20.22	46.04†	7.01	0.14	4.78	58.77	60.55	9.34	22.63	40.5	9.82	13.93	3.92	**0.73**†	24.61
RMGT-mutKNN-RBF	**19.34**	38.87†	6.35	0.07	**4.44**	57.47	60.0	7.21	**21.58**	38.43	1.16	**7.72**	**1.51**	0.83†	24.61
RMGT-symFKNN-RBF	19.56	42.97†	6.69	0.14	4.59	58.12	60.73	8.53	21.97	39.26	7.99	13.57	3.72	0.78†	24.29
RMGT-symKNN-HM	21.32	37.08	18.75	0.07	4.98	53.25	**55.64**	5.33	31.18	38.84	1.5	10.59	2.91	27.84	23.51
RMGT-mutKNN-HM	20.88	34.02	15.99	**0.0**	4.88	52.6	56.36	3.45	29.34	37.6	**0.17**	10.48	1.61	10.85	23.82
RMGT-symFKNN-HM	**19.34**	36.32	17.66	0.07	**4.44**	52.27	**55.64**	3.84	31.18	37.6	1.16	11.15	2.81	28.88	**22.41**
RMGT-symKNN-LLE	25.05	31.46	6.07	**0.0**	5.46	52.27	58.18	3.03	33.82	35.95	1.5	9.64	2.11	31.98	28.68
RMGT-mutKNN-LLE	22.64	31.97	**5.7**	**0.0**	4.49	51.95	**55.64**	3.41	31.05	**35.54**	0.5	10.26	1.61	30.39	27.59
RMGT-symFKNN-LLE	24.84	**28.9**	5.81	**0.0**	4.88	**51.3**	58.18	**2.22**	32.24	35.95	1.0	10.27	2.01	33.42	27.74

TABLE III. AVERAGE RANKINGS FOR THE GRAPH CONSTRUCTION METHODS FOR EACH SSL ALGORITHM.

	GFHF	LGC	LapRLS	RMGT	overall
symKNN-RBF	4.567	4.6	4.423	6.933	5.129
mutKNN-RBF	**3.167**	**3.433**	**3.039**	4.2	**3.46**
symFKNN-RBF	3.767	3.767	4.269	6.233	4.509
symKNN-HM	7.033	7.2	7.5	5.933	6.916
mutKNN-HM	4.433	4.433	4.192	4.133	4.298
symFKNN-HM	5.567	5.433	6.231	4.533	5.441
symKNN-LLE	6.333	6.5	6.231	5.233	6.074
mutKNN-LLE	5.133	4.7	3.539	**3.4**	4.193
symFKNN-LLE	5.0	4.933	5.577	4.4	4.978

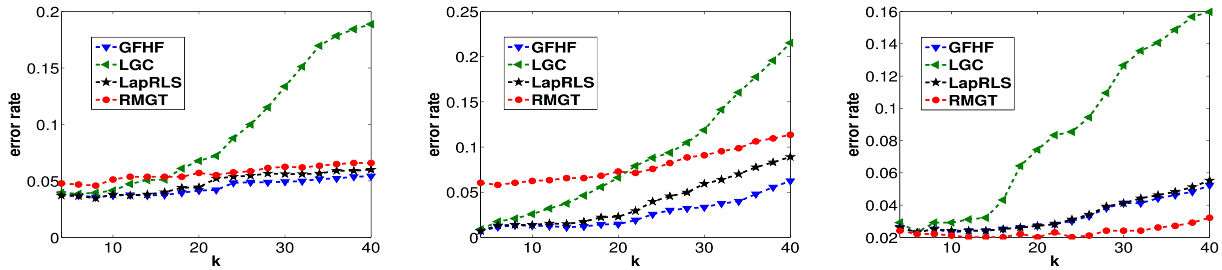
For most combinations of graph construction method, the SSL algorithms showed similar performances. However, we can see some interesting results in Table II. For ChlorineConcentration (D3), we see that the graphs generated by LLE outperformed those generated by HM and the RBF kernel for GFHF, LGC, and LapRLS. However, for RMGT, this high superiority was not evidenced if we compare these results with those for the graphs generated by LLE and the RBF kernel.

For most combinations of SSL algorithm and weighted matrix generation method, we see that the graphs generated by mutKNN outperformed those generated by symKNN and symFKNN. Moreover, for SonyAIBORobotSurface (D11), we see that the graphs generated by mutKNN outperformed those generated by symKNN and symFKNN by a large margin.

RMGT showed competitive performance with the other SSL algorithms in most data sets. However, for the data sets with high imbalanced ratio (e.g. MedicalImages (D9), StarLightCurves (D12), and Wafer (D14)), this method achieved poor results for some graph construction methods. For Mallat (D8), RMGT showed no competitive results with the other methods when using the graphs generated by the RBF kernel. In addition, for StarLightCurves (D12), RMGT showed poor results for all graph construction methods. Moreover, for Wafer (D14), this method showed poor results using the graphs generated by HM and LLE.

Table III shows the average rankings for the graph construction methods for each SSL algorithm as well as the overall average rankings. The results that were statistically[3] outperformed by the best ranked method have a grey background. We see in Table III that mutKNN-RBF achieved the best average ranking for GFHF, LGC, and LapRLS as well as the best overall average ranking. We also note in Table III that the graphs generated by mutKNN achieved better average rankings than those generated by symKNN and symFKNN for all weighted matrix generation methods. In addition, we see that symKNN-RBF and symFKNN-RBF achieved good average rankings for GFHF, LGC, and LapRLS. However, these graphs achieved the worst average rankings for RMGT.

[3]We used the Friedman's test with the Nemenyi's post test using a significance level of 0.05 (see [15] for a review on statistical tests).



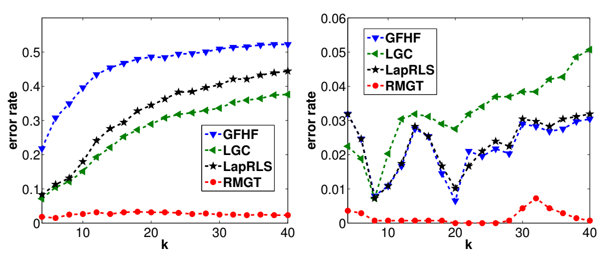
(a) FacesUCR (b) ChlorineConcentration (c) Symbols

Fig. 1. Error rates of the SSL algorithms with respect to k on the FacesUCR (using the symFKNN-RBF graph), ChlorineConcentration (using the symFKNN-LLE graph), and Symbols (using the symFKNN-LLE graph) data sets.

B. Evaluation of classifier stability

For most combinations of graph construction method and data set, the SSL algorithms showed similar behaviors with respect to k. However, we found some interesting results, which are discussed in the following.

Fig. 1 shows the error rates of the SSL algorithms on the FacesUCR (D5), ChlorineConcentration (D3), and Symbols (D13) data sets with respect to k. Although LGC performed very well in the best case analysis in these data sets, we see that this method showed high instability while the competing methods achieved good as well as stable results. Fig. 1 is an excellent example to show why the best case analysis alone may not be effective to choose the best classifiers for a given application; this analysis may hide useful information related to the SSL algorithms' performance with respect to graph construction and parameter selection [2].



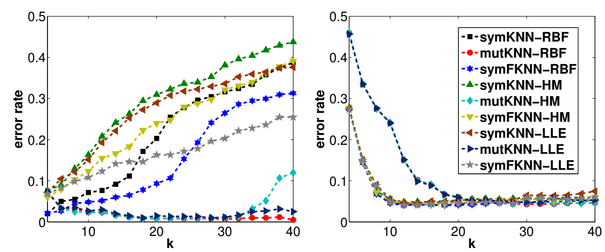
(a) SonyAIBORobotSurface (b) Cinc-ECG-torso

Fig. 2. Error rates of the SSL algorithms with respect to k on the SonyAIBORobotSurface and Cinc-ECG-torso data sets using the symKNN-LLE graph.

Fig. 2 shows the error rates of the SSL algorithms with respect to k on the SonyAIBORobotSurface (D11) and Cinc-ECG-torso (D4) data sets using the symKNN-LLE graph. Although Hypothesis 3 says that RMGT may not be effective on imbalanced data sets, we see that this method showed exceptional performance as well as good stability on SonyAIBORobotSurface and Cinc-ECG-torso while the competing methods showed moderate to high instability. However, for the data sets with high imbalanced ratio (e.g. MedicalImages (D9), StarLightCurves (D12), and Wafer (D14)), RMGT showed poor results for some graph construction methods.

C. Evaluation of graph stability

In this section, we evaluate the stability of the graph construction methods with respect to k for each SSL algorithm. We analyze in the following some specific though important results concerning graphs' stability with respect to k.



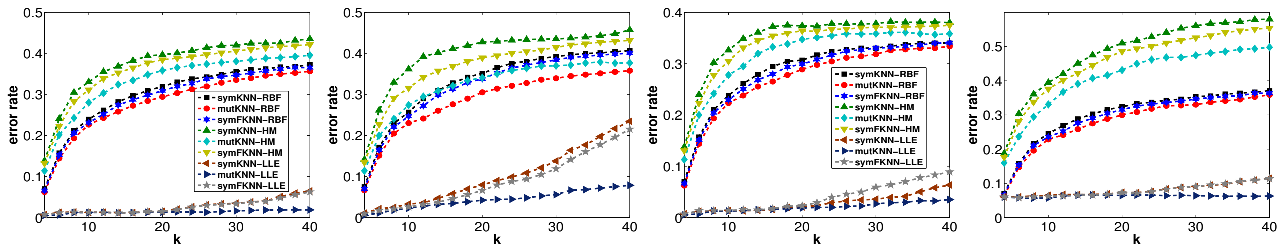
(a) SonyAIBORobotSurface (b) Mallat

Fig. 3. Error rates of LGC with respect to k on the SonyAIBORobotSurface and Mallat data sets using a variety of graph construction methods.

Fig. 3 shows the error rates of LGC with respect to k on the SonyAIBORobotSurface (D11) and Mallat (D8) data sets using a variety of graph construction methods. The legends for Fig. 3 and 4 can be found in Fig. 3(b), 4(a), and 4(c). Although the graphs generated by mutKNN achieved similar results in the best case analysis on SonyAIBORobotSurface for most SSL algorithms, Fig. 3(a) shows that these graphs achieved good as well as stable results while the other graphs showed high instability.

Fig. 3(b) shows that the graphs generated by mutKNN achieved the highest instabilities in comparison to the other graphs on Mallat (D8) while achieving similar results in the best case analysis in this data set. Although the graphs generated by mutKNN showed high instability in this data set, this behavior was only evidenced in a few data sets.

Fig. 4 shows the error rates of the SSL algorithms with respect to k on the ChlorineConcentration (D3) data set using a variety of graph construction methods. We see that the graphs generated by LLE outperformed the other graphs. The graphs generated by HM and the RBF kernel achieved high instability. We also note that the graphs generated by mutKNN performed slightly better than those generated by symKNN and symFKNN.

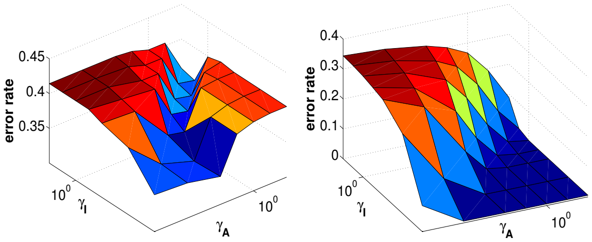


(a) GFHF (b) LGC (c) LapRLS (d) RMGT

Fig. 4. Error rates of the SSL algorithms with respect to k on the ChlorineConcentration (D3) data set using a variety of graph construction methods.

D. Evaluation of regularization parameters

In this section, we evaluate how the performance of LapRLS is affected with respect to γ_A and γ_I. We see in Table II that LapRLS achieved the best overall result in 8 out of 15 data sets and competitive results in the other data sets for most graph construction methods. Although it appears to be a surprising result, we have to notice that LapRLS has two regularization parameters and, to our knowledge, the process of parameter selection for this method remains unclear. Since these regularization parameters may drastically affect the performance of LapRLS [2], we have to analyze the error surfaces generated by this method in order to verify whether it is strongly or weakly dependent of parameter selection with respect to γ_A and γ_I on a given data set. We provide this analysis in the following.



(a) Adiac (b) ChlorineConcentration

Fig. 5. Error surfaces for LapRLS with respect to γ_A and γ_I on the Adiac (using the mutKNN-RBF graph) and ChlorineConcentration (using the mutKNN-LLE graph) data sets.

Fig. 5(a) shows the error surface generated by LapRLS on the Adiac (D2) data set using the mutKNN-RBF graph. We see that the optimal results are achieved when $\gamma_A = \gamma_I$, which is a pattern found in [2] for a data set of text classification. However, this pattern occurred only in a few combinations of graph construction method and data set.

Fig. 5(b) shows the error surface generated by LapRLS on the ChlorineConcentration (D3) data set using the mutKNN-LLE graph. Although LapRLS achieved the best overall performance in this setting (with an error rate of 0.52%), we see that this method can be considered strongly dependent of parameter selection with respect to γ_A and γ_I on ChlorineConcentration. This high variance on the error rate (almost 35%) of LapRLS may hinder the effective use of this method on real applications of time series transductive classification.

Fig. 6 shows the error surfaces generated by LapRLS on the Cinc-ECG-torso (D4) data set using the graphs generated by the RBF kernel. We see that the optimal results occur only when $\gamma_A \neq \gamma_I$, which is an opposite pattern to that reported in Fig. 5(a). Unfortunately, by analyzing the error surfaces generated by LapRLS, we found no evident explorable pattern that could help parameter selection.

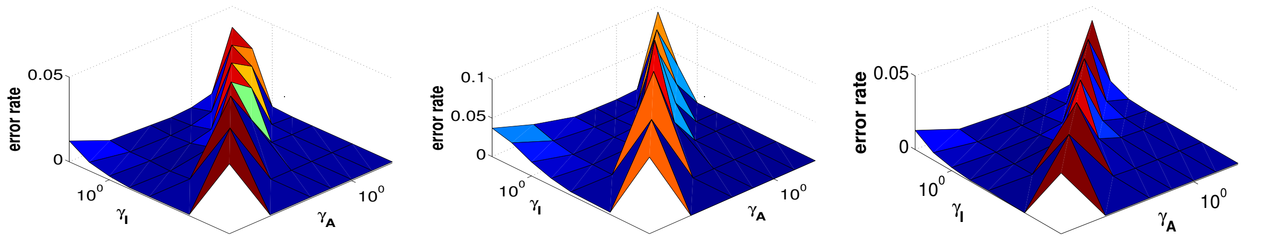
IV. Conclusion

In this paper, we provided a comprehensive empirical comparison of state-of-the-art graph-based SSL algorithms combined with a variety of graph construction methods in order to compare them on time series transductive classification tasks on imbalanced data sets. Through a comprehensive and detailed analysis using recently proposed empirical evaluation models, we observed the following:

- the graphs generated by mutKNN showed high instability in some data sets for relatively small values of k. However, these graphs showed good performance and stability in a variety of data sets, even for small values of k. Therefore, Hypothesis 1 may not hold in general. Although we found situations in which the graphs generated by symKNN and symFKNN achieved high instability for small values of k, such an instability is more probable to occur when we use the graphs generated by mutKNN because they have less edges than the other graphs for the same value of k;
- for most data sets, the graphs generated by mutKNN achieved better results than those generated by symKNN and symFKNN for all weighted matrix generation methods. In addition, for some data sets, mutKNN outperformed the competitors by a large margin. Therefore, mutKNN tends to be the best adjacency graph for time series data; hence, Hypothesis 2 holds for time series transductive classification;
- by analyzing the average rankings of the graph construction methods in Table III, we see that mutKNN-RBF achieved the best average ranking for GFHF, LGC, and LapRLS as well as the best overall average ranking. Therefore, this graph may be the best graph for time series transductive classification on imbalanced data sets for general tasks. In the absence of any other criteria to choose a graph construction method for real applications, we suggest, from our results, the use of mutKNN-RBF;

TABLE IV. Error rates (%) for GFHF using mutKNN-RBF ($k = 20$), the best results for GFHF, and the best overall results.

Data sets	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
GFHF (mutKNN-RBF)	20.44	37.34	29.43	0.0	4.78	56.49	56.54	6.14	21.18	35.54	0.83	7.24	1.81	0.67	25.24
best results (GFHF)	19.34	31.97	0.52	0.0	3.56	50.97	56.36	4.18	20.0	35.54	0.5	4.4	1.51	0.23	23.35
best results (overall)	18.24	27.88	0.52	0.0	3.46	50.0	55.09	2.22	19.21	35.12	0.17	3.97	1.51	0.23	22.26



(a) symKNN-RBF (b) mutKNN-RBF (c) symFKNN-RBF

Fig. 6. Error surfaces for LapRLS with respect to γ_A and γ_I on the Cinc-ECG-torso data set using graphs generated by the RBF kernel.

- RMGT showed exceptional performance on most data sets. However, for the data sets with high imbalanced ratio, this algorithm achieved poor results. Therefore, Hypothesis 3 may only hold for data sets with *high* imbalanced ratio;
- although LapRLS achieved the best overall performance in most data sets in the best case analysis, the error surfaces generated by this method showed high variance in the error rate with respect to γ_A and γ_I. Therefore, at least for imbalanced data sets, LapRLS can be considered strongly dependent of parameter selection with respect to γ_A and γ_I, which may hinder the effective use of this method on real applications of time series transductive classification. However, this method can naturally perform induction through kernel expansions, which may be an advantage over other methods, depending on the application;
- LGC, LapRLS, and RMGT may not be effective for time series transductive classification due to the following observations: (1) LGC achieved high instability on some data sets (see Fig. 1); (2) RMGT showed no competitive results in some data sets (those with high imbalanced ratio); and (3) LapRLS can be considered strongly dependent of parameter selection with respect to γ_A and γ_I in many data sets;
- we note that GFHF achieved competitive results in most combinations of graph construction method and data set. In addition, this method is *parameter-free*; hence, we just have to choose the parameter k of the graph. In Table IV, we show the results for GFHF using mutKNN-RBF with $k = 20$. We see that this setting achieved good results in comparison to the best result reported for GFHF in most data sets. The performance for this setting can be considered ineffective only for ChlorineConcentration (D3). As shown in Fig. 4, only the graphs generated by LLE were effective in this data set. From our results, we suggest $k = 20$ as initial choice for real applications on time series transductive classification.

Acknowledgments

Supported by São Paulo Research Foundation (FAPESP) under grants 2011/17698-5 and 2012/50714-7.

References

[1] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*. The MIT Press, 2006.

[2] C. A. R. de Sousa, S. O. Rezende, and G. E. A. P. A. Batista, "Influence of graph construction on semi-supervised learning," in *ECML PKDD*, 2013, pp. 160–175.

[3] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *PVLDB*, vol. 1, no. 2, pp. 1542–1552, 2008.

[4] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *DMKD*, vol. 26, no. 2, pp. 275–309, 2013.

[5] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[6] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NIPS 16*, 2004, pp. 321–328.

[7] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from labeled and unlabeled examples." *JMLR*, vol. 7, pp. 2399–2434, 2006.

[8] W. Liu and S.-F. Chang, "Robust multi-class transductive learning with graphs," in *CVPR*, 2009, pp. 381–388.

[9] M. Hein and M. Maier, "Manifold denoising," in *NIPS 19*, 2007, pp. 561–568.

[10] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[11] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *ICML*, 2010, pp. 679–686.

[12] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *ICML*, 2009, pp. 441–448.

[13] S. Melacci and M. Belkin, "Laplacian support vector machines trained in the primal," *JMLR*, vol. 12, pp. 1149–1184, 2011.

[14] R. Johnson and T. Zhang, "On the effectiveness of laplacian normalization for graph semi-supervised learning," *JMLR*, vol. 8, pp. 1489–1517, 2007.

[15] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *JMLR*, vol. 7, pp. 1–30, 2006.