2016 IEEE 16th International Conference on Data Mining

Reliable Semi-supervised Learning

Junming Shao, Chen Huang, Qinli Yang and Guangchun Luo
School of Computer Science and Engineering, Big Data Research Center
University of Electronic Science and Technology of China, Chengdu, China
Email: {junmshao, gcluo}@uestc.edu.cn

***Abstract*—In this paper, we propose a Reliable Semi-Supervised Learning framework, called ReSSL, for both static and streaming data. Instead of relaxing different assumptions, we do model the reliability of cluster assumption, quantify the distinct importance of clusters (or evolving micro-clusters on data streams), and integrate the cluster-level information and labeled data for prediction with a lazy learning framework. Extensive experiments demonstrate that our method has good performance compared to state-of-the-art algorithms on data sets in both static and real streaming environments.**

***Keywords*-semi-supervised learning; data stream; reliability**

I. Introduction

During the past decade, semi-supervised learning has gained growing attentions and many approaches have been proposed from different points of view, such as co-training [1], graph representation [2], generative models [3] and SVM-based extension [4], [5]. In light of the scarcity of labeled data, these algorithms try to improve the performance by appropriately exploiting the available unlabeled data. However, it has found that the performance of traditional semi-supervised learning approaches may be even worse than using labeled data only in many real-world scenarios [5]. The main rationale is the current semi-supervised learning algorithms often heavily depend on some assumptions. If the assumption(s) does (do) not hold, unlabeled data cannot help and may actually hurt accuracy. To deal with these problems, the reliable semi-supervised learning approaches are highly needed. Currently, the main strategy is to relax the corresponding assumptions, such as conditional independence assumption relaxation [6] and low-density separation assumption [5]. Although the dependence of the assumption somehow is relaxed, the performance also cannot be promised.

In this paper, instead of relaxing assumptions, we do model the potential basic hypophysis for all current semi-supervised learning algorithms: *cluster assumption*. The basic idea is to construct clusters (or evolving micro-clusters in streaming data) with both limited labeled and abundant unlabeled data, model the degree of correctness of cluster assumption, and quantify the distinct importance of clusters (micro-clusters), and finally leverage the labeled data and reliabilities of clusters for prediction at the cluster level. To illustrate this idea, Figure 1 gives a simple example. For simplicity, a two-dimensional data which contains labeled and unlabeled instances is displayed. We can observe that some clusters (indicated by the blue dashed ellipsoids) violate the cluster assumption since different labeled instances exist in the same clusters. Therefore, using the cluster information for diffusing labels or finding the low-density separator may not yield good results as the cluster assumption does not hold anymore. Here, we model the reliability for each local cluster (e.g. $R_1 < R_2 < R_3$) and use them for improving classification.

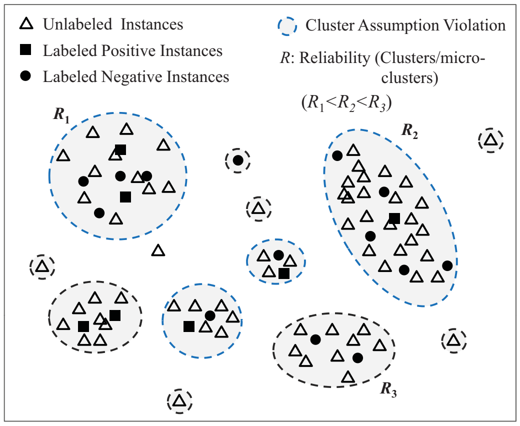

Unlabeled Instances
Labeled Positive Instances
Labeled Negative Instances
Cluster Assumption Violation
R: Reliability (Clusters/micro-clusters)
$(R_1<R_2<R_3)$
R_1
R_2
R_3

Figure 1. Illustration of cluster assumption modeling. Here traditional semi-supervised learning algorithms may produce a poor performance since the basic cluster assumption is violated. A good strategy may leverage the local cluster information, model the reliability of each cluster to build a reliable semi-supervised learning framework.

II. Related Work

During the past several decades, semi-supervised learning has attracted a huge volume of attentions, and many algorithms have been proposed [1], [7], [2], [4], [5]. Due to space limitation, we briefly survey on some important major research of semi-supervised learning directions.

Traditional Semi-supervised learning. Early algorithms of semi-supervised learning focus on using some generative models (e.g. Gaussian mixture model) to propagate the labels of unlabeled data by maximizing the model fitness [3]. Co-training is another family of semi-supervised learning paradigm proposed by Blum and Mitchell [1], which is first developed for text mining and later extended in diverse applications. The key idea is to learn two or more classifiers

2374-8486/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDM.2016.11

from two views, and improves each classifier by predicting the unlabeled data based on the exploitation of disagreement of these classifiers. Building upon the label smoothness assumption, graph-based methods encode all data (including labeled and unlabeled data) as a weighted graph, and estimate the class memberships probability for the unlabeled data based on different strategies, such as Gaussian fields [7], manifold regularization [8] and max-cut [2]. In addition, the extension of support vector machines (SVMs) to unlabeled data is a popular strategy. For example, the widely-used algorithm, S3VMs [4], is to seek decision boundaries that pass through low density regions with maximum-margin. In summary, all algorithms depend on some assumptions, and each often comes with specific advantages and drawbacks.

Assumption Relaxation. To make semi-supervised learning algorithms to be more robust and safer, some relaxing-assumption algorithms have been proposed, which mainly focus on the conditional independence assumption in co-training. Balcan et al. [6] relax the conditional independence assumption with a much weaker expansion condition, and justify the iterative co-training procedure. Johnson and Zhang [9] propose a two-view model that relaxes the conditional independence assumption. Recently, based on the S3VM framework, Li and Zhou aim at building a safe semi-supervised learning algorithm, S4VM [5], by exploiting several low-density separators simultaneously to reduce the risk of identifying a poor separator with unlabeled data. However, S4VM still tends to fail if the cluster assumption (e.g. separators passing across low density regions) does not hold like S3VM.

Instead of relaxing assumptions, we do model the implicit cluster assumption in existing semi-supervised learning algorithms.

III. Reliable Semi-supervised Learning Framework

In this section, we present our semi-supervised learning framework ReSSL.

A. Cluster Assumption Modeling

To make use of unlabeled data, existing semi-supervised learning algorithms at least assume some structures to the underlying distribution of data, such as Gaussian distribution, label smoothness and manifold assumption. For all existing semi-supervised learning algorithms, they may explicitly or implicitly depend on the basic cluster assumption: "*instances in the same cluster are more likely to share the same label*". In the following, we will model the cluster assumption from two aspects: **Cluster Regularity** and **Cluster Priority**.

1) Cluster Regularity: Suppose that we have a training data set $D = \{x_1, \cdots, x_m, x_{m+1}, ..., x_n\}$ belonging to $L = \{1, \cdots, l\}$ classes, where $D_L = \{x_1, \cdots, x_m\}$ denotes the labeled data, and the remaining data $D_U = \{x_{m+1}, ..., x_n\}$ denotes the unlabeled data. Here, all instances with and without labels are first grouped into clusters with any existing clustering algorithm, e.g. K-Means [10], or Sync [11], [12]. In this study, K-Means algorithm is applied due to its simplicity. Given the derived clusters, the label distributions of different clusters may largely differ, ranging from complete consistent (all labeled instances share the same label) to inconsistent (all labeled instances with different classes are existed). To measure the consistency, we apply entropy to characterize the regularity of each cluster.

Definition 1 (**Cluster Regularity**) Given a clustering $C = \{C_1, \cdots, C_k\}$ on a training data set $D = \{X_L, X_U\}$, the regularity of a cluster C_i, denoted as $CR(C_i)$, is defined as:

$$CR(C_i) = \frac{H - H(C_i)}{H} \tag{1}$$

where H indicates the global entropy of all labeled data and $H(C_i)$ indicates the entropy of a given cluster C_i, which are computed as follows.

$$H = \sum_{l=1}^{|L|} -\frac{m_l}{m} log(\frac{m_l}{m}) \tag{2}$$

$$H(C_i) = \sum_{l=1}^{|L|} -\frac{m_l(C_i)}{m(C_i)} log(\frac{m_l(C_i)}{m(C_i)}) \tag{3}$$

where m is the total number of labeled instances and m_l is the number of instances associating with the label l, $m(C_i)$ is the number of labeled instances in the cluster C_i and $m_l(C_i)$ is the number of labeled instances belonging to label l in the cluster C_i. In a special case, the cluster regularity assigns to the global regularity H if all instances in a cluster are unlabeled.

2) Cluster Priority: The cluster regularity characterizes the label consistency, and provides a direct way to reflect the degree of cluster assumption violation. However, only the regularity is insufficient since the priority of each label is different. For example, considering a training data with two classes: positive and negative. The class distribution in the training data set is $c1 : c2(c1 < c2)$. Suppose we obtain two clusters C_1 and C_2, and the class distribution of the two classes in the two clusters are $c3 : c4(c3 > c4)$ and $c4 : c3$, respectively. In this case, we expect that the reliability of $C1$ is much higher than C_2. The reason is that it is more difficult to form a cluster C_1 than C_2 as the positive instances are fewer. The imbalanced class distribution problem is considered to better reflect the reliability of each cluster.

Definition 2 (**Cluster Priority**) Given a clustering $C = \{C_1, \cdots, C_k\}$ on a training data set $D = \{X_L, X_U\}$, the priority of a cluster C_i, denoted as $CR(C_i)$, is defined as:

$$CP(C_i) = \frac{1}{1 + exp(-\sum_{l=1}^{|L|} \frac{P(C_i^l) - P(D^l)}{P(D^l)})} \tag{4}$$

where $P(C_i^l)$ indicates the probability of labeled instances with label l in the class C_i, and $P(D^l)$ is the probability of labeled instances associated with label l in the data set D.

Finally, the reliability of a cluster is given as follows.

Definition 3 (**Cluster Reliability**) Given a clustering $C = \{C_1, \cdots, C_k\}$ on a training data set $D = \{X_L, X_U\}$, the reliability of each cluster C_i, denoted as $R(C_i)$, quantifying the degree of cluster assumption violation, is defined as:

$$R(C_i) = CR(C_i) \cdot CP(C_i) \tag{5}$$

B. Cluster-Level Lazy Learning

Here, a cluster-level lazy learning framework based on the cluster structure distance and cluster reliability, is used for prediction.

Definition 4 (**Cluster Structure Distance**) Given a test instance x and a cluster C_i, for each label l, we define the cluster structure distance $D^{cs}(x, C_i^l)$ as the average distance from the test instance x to all instances belonging to the label l in the cluster C_i.

$$D^{cs}(x, C_i^l) = \frac{\sum_{y \in C_i^l} dist(y, x)}{|C_i^l|} \tag{6}$$

where $dist(y, x)$ is a metric distance function, and the Euclidean distance is used in this study.

With the cluster structure distance, we can use cluster-level lazy learning framework for prediction. Specifically, first, for a given test instance x, we first search its k-nearest neighbors $N(x)$, and then find the corresponding cluster(s) (i.e. C_n) of these neighbors. Finally, the label of x is determined by the cluster structure distance, cluster reliability and cluster label distribution as follows.

$$x_p = \begin{cases} \underset{l}{argmax}\left(\frac{R(C_{n1}) \cdot P(C_{n1}^l)}{D^{cs}(x, C_{n1}^l)}\right) & \text{if } R(C_{n1}) > \mu + \tau \cdot \sigma \\ \underset{l}{argmax}\left(\sum\limits_{C_i \in C_n} \frac{R(C_i) \cdot P(C_i^l)}{D^{cs}(x, C_i^l)}\right) & \text{otherwise} \end{cases} \tag{7}$$

where C_{n1} is the nearest cluster including the nearest neighbor of x, C_n represent all clusters containing the k nearest neighbors of x, $P(C_i^l)$ indicates the probability of instances with label l in the cluster C_i, $D^{cs}(x, C_i^l)$ is the cluster structure distance from x to all instances belonging to the label l in the cluster C_i, and $R(C_i)$ is the reliability of the cluster C_i. μ and σ are the corresponding mean and standard deviation of the cluster reliabilities, respectively.

Here the prediction follows in two ways: if the reliability of the nearest cluster is high, it is confident to use the cluster information for prediction only. Otherwise, all nearest clusters are used to support a robust prediction. To determine whether the reliability of one cluster is high, the heuristic way is applied, where the reliability of the cluster is at least higher than the average reliability of all cluster plus τ number of standard deviations. Finally, the pseudocode of ReSSL for static data is given in Algorithm 1.

Algorithm 1 ReSSL

```
Require:
    Training data $D$ and test data $T$
    The number of nearest neighbors: $k$
    Reliability control factor: $\tau$
Obtain a clustering $C$ on $D$ with K-Means;
// Cluster assumption modeling
for each $C_i \in C$ do
    Calculate its cluster regularity $CR(C_i)$ with Eq. (1);
    Calculate its cluster priority $CP(C_i)$ with Eq. (4);
    Measure its reliability $R(C_i)$ with Eq. (5);
end for
for each test instance $x \in T$ do
    Search $k$ nearest instances $N(x)$ from $D$;
    Identify the cluster(s) $C_n \subseteq C$ containing $N(x)$.
    if $C_{n1} = GetNearst(C_n) > \mu + \tau \cdot \sigma$ then
        Predict the label of $x$ with $C_{n1}$ using Eq. (7)
    else
        Predict the label of $x$ with $C_n$ using Eq. (7)
    end if
end for
```

IV. ReSSL on Data Streams

In this section, we extend the ReSSL framework in streaming environment and propose our reliable semi-supervised learning algorithm on data streams.

A. Semi-supervised Micro-clusters

Definition 5 (**Semi-supervised Micro-cluster**) We define a semi-supervised micro-cluster as a tuple MC^S = $(LS,\ SS, N_l, N_u, LD, LC, R)$. N_l and N_u denote the number of labeled and unlabeled instances in this micro-cluster, respectively, LD stores the number of instances for each label, LC maintains the center vector for each label and R is the reliability of the micro-cluster.

For each arriving instance, as ReSSL works on static data, the label is determined by the cluster structure distance, cluster reliability and the label distribution. The only difference is to search k nearest micro-clusters instead of searching k nearest instances. The reason is that we cannot maintain all instances, and only allow storing the statistic summary of the micro-clusters. Therefore, we search the k-nearest micro-clusters and use them for prediction in the same way.

B. Evolving Micro-clusters Maintenance

For semi-supervised learning on data stream, the crucial point is to maintain the micro-clusters to capture the current cluster structure. The update of the micro-clusters mainly from two aspects: forgetting historical data and accepting new incoming data. By considering the decreasing importance of historical data, as usual, the decay function is applied. Like DenStream[13], each micro-cluster is associated with a weight, denoted as $w(MC_i^S)$, indicated its importance over time.

$$w(MC_i^S) = \sum_{i=1}^{N} exp\big(-\lambda(T_{cur} - T_i)\big) \tag{8}$$

where T_{cur} is the current time and T_i is the arriving time stamp of instance x_i.

The update of micro-clusters with new incoming instances is similar with Denstream [13]. We maintain a set of semi-supervised micro-clusters and some potential outlier micro-clusters at the same time. For a new incoming instance, after prediction, the instance is inserted into an existing micro-cluster, is viewed as a potential outlier or we generate a new micro-cluster for the instance, depending on its distance to nearest micro-cluster. Specifically, when a new instance x arrives, the procedure is as follows.

1) First, we find its nearest semi-supervised micro-cluster MC_{n1}^S. If its distance is smaller than the radius of the micro-cluster (i.e. the instance is located inside the micro-cluster), we merge x into MC_{n1}^S, and update MC_{n1}^S based on the addition property of cluster feature. If its distance is larger than the diameter of the micro-cluster, it is defined as a potential outlier. And the instance is inserted into the nearest outlier micro-cluster, and then update the outlier micro-cluster accordingly.
2) Otherwise, the instance will be assigned into an existing micro-cluster or a new micro-cluster is generated to contain the instance according to its location against to the two nearest micro-clusters (i.e. MC_{n1}^S and MC_{n2}^S). The instance is merged into an existing nearest micro-cluster if the cover ratio r is large ($r = 0.5$ in this study), and a new micro-cluster is generated to include the instance, otherwise. The cover ratio is defined as follows.

$$r = \frac{(dist(x, MC_{n2}^S) - R_2) - (dist(x, MC_{n1}^S) - R_1)}{R_1} \tag{9}$$

V. Parameter Setting

ReSSL needs to specify some parameters for learning static data and streaming data, respectively. For static data sets, the number of nearest neighbors k, the reliability control factor τ, and the number of K for clustering, are required. Like KNN, k determines the degree of data information ranging from the local to the global we expect to use for prediction. In this study, we set $k = 3$ as default value. For the parameter of reliability control factor τ, it characterizes the level when we allow predicting the label of a test instance using the most reliable cluster alone. Here, we set $\tau = 2$ according to the empirical rule (i.e. 68%-95%-99.7% rule), which indicates that the reliability of the cluster is significantly higher than those of others (p-value = 0.05 if normal distribution is assumed). For K-Means clustering embedded in ReSSL framework, unless specified otherwise, we specify the number of clusters as the number of five times of classes to get a reasonable number of clusters. For data streams, except for the parameters specified for static data learning, ReSSL needs to specify the maximum number of micro-clusters maintained ($maxClu$ = 1000 in this study).

VI. Experimental Evaluation

In this section, we evaluate our proposed reliable semi-supervised learning framework, ReSSL, on both real-world static data and streaming data. We also compare our framework to several representatives of semi-supervised learning approaches. For static data sets, we selected the co-training paradigm: Tri-train [14], ensemble co-training strategy Co-forest [15], the graph-based algorithm GFHF [7], the popular SVM-based approach meanS3VM [16], LapSVM [17], S4VM [5], and the baseline K-nearest neighbor classifier: KNN. For data streams, we compare ReSSL to IBLStream [18] and SPASC [19].

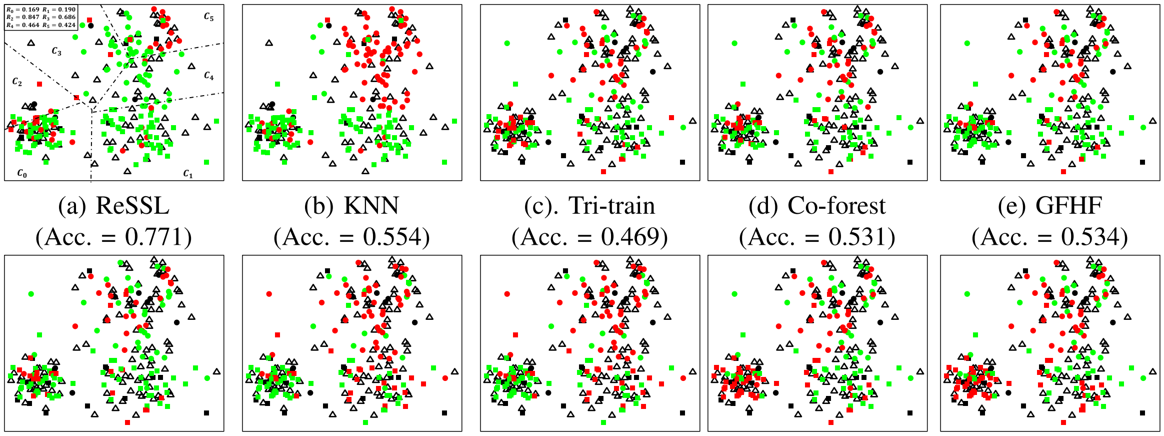
A. Proof of Concept

In this section, we start with the synthetic data sets to prove the concept of ReSSL. To generate the synthetic data sets, we first use Weka to create two-dimensional data sets with RandomRBF, and then 20% instances are labeled into two classes according to their cluster locations. Here a ratio α is further used to control the percentage of inconsistent labeled instances in the same clusters. Finally, all data sets are randomly split into two equal-size subsets, where one subset is used for training and the remaining is for testing.

We evaluate whether ReSSL allows modeling each cluster reliability well, and more importantly, supports a better prediction performance based on the reliable cluster-level information. Figure 2(a) depicts the cluster reliabilities and the predictions on a given synthetic data which we described above. Figure 2 (b) - (j) depicts the prediction performances for all comparing algorithms. From these plots, we can see that ReSSL achieves the best prediction accuracy. The reason lies in that ReSSL models the cluster assumption by cluster reliability, integrates them into prediction at the cluster-level, and thus avoids a large number of erroneous labeling if the cluster assumption is violated.

B. Evaluation on Static Real-world Data Sets

In this section, we evaluate our ReSSL framework and compare its performance to other state-of-the-art semi-



(a) ReSSL (Acc. = 0.771) (b) KNN (Acc. = 0.554) (c). Tri-train (Acc. = 0.469) (d) Co-forest (Acc. = 0.531) (e) GFHF (Acc. = 0.534)

(f) S4VM (Acc. = 0.594) (g) MeanS3VM(MKL) (Acc. = 0.400) (h) MeanS3VM(Iter) (Acc. = 0.400) (i) LapSVM(Primal) (Acc. = 0.343) (J) LapSVM(Newton) (Acc. = 0.314)

Figure 2. Illustration the cluster reliability modeling on an artificial data set. Here the triangles and squares with black color indicate the training instances while the shapes with green or red colors mean the testing instances. The hollow shapes are unlabeled instances in training data set and the different filled shapes indicate different classes. In addition, the shapes with green color mean correct predictions and vice versa.

Table I
SUMMARY OF PREDICTION PERFORMANCE OF DIFFERENT SEMI-SUPERVISED LEARNING ALGORITHMS ON EIGHT REAL-WORLD DATA SETS. HERE LP INDICATES THE PERCENTAGES OF INSTANCES ARE LABELED.ALL ALGORITHMS ARE RUN FOR TEN TIMES AND THE CLASSIFICATION ACCURACIES ARE AVERAGED, WHERE $\mu \pm \sigma$ INDICATES THE MEAN μ PLUS STANDARD DEVIATION σ.

Data	LP	ReSSL	KNN	Co-forest	Tri-train	GFHF	LapSVM(NewTon)	LapSVM(PCG)	MeanS3VM(Iter)	MeanS3VM(MKL)	S4VM
Wdbc	1%	**0.851±0.036**	0.556±0.129	0.760±0.100	0.689±0.143	0.641±0.005	0.432±0.233	0.509±0.217	0.756±0.118	0.756±0.118	0.811±0.158
#Class=2	3%	**0.871±0.043**	0.711±0.153	0.842±0.071	0.792±0.089	0.657±0.049	0.255±0.158	0.420±0.235	0.718±0.157	0.718±0.156	0.825±0.166
Dim.=30	5%	0.896±0.068	0.781±0.084	0.882±0.047	0.857±0.043	0.639±0.000	0.173±0.148	0.264±0.179	0.793±0.129	0.793±0.129	**0.915±0.047**
#Inst.=569	10%	**0.913±0.018**	0.852±0.055	0.871±0.044	0.856±0.044	0.656±0.043	0.129±0.032	0.227±0.200	0.651±0.164	0.651±0.164	0.906±0.032
Breast	1%	**0.943±0.032**	0.500±0.152	0.747±0.208	0.742±0.111	0.650±0.001	0.620±0.266	0.399±0.193	0.772±0.268	0.772±0.268	0.906±0.115
#Class=2	3%	**0.957±0.015**	0.839±0.125	0.909±0.040	0.854±0.072	0.650±0.000	0.431±0.330	0.439±0.201	0.700±0.276	0.700±0.277	0.930±0.072
Dim.=9	5%	**0.958±0.017**	0.888±0.079	0.895±0.031	0.857±0.073	0.650±0.000	0.561±0.299	0.525±0.146	0.706±0.341	0.706±0.341	0.943±0.052
#Inst.=699	10%	**0.958±0.013**	0.951±0.022	0.932±0.020	0.899±0.029	0.650±0.000	0.626±0.257	0.515±0.250	0.880±0.058	0.880±0.059	0.930±0.033
Banknote	1%	0.749±0.065	0.570±0.119	0.768±0.143	0.688±0.122	0.731±0.216	0.534±0.058	0.503±0.069	0.645±0.083	0.645±0.083	**0.824±0.138**
#Class=2	3%	0.858±0.055	0.749±0.154	0.813±0.061	0.796±0.083	0.626±0.186	0.489±0.065	0.437±0.084	0.569±0.151	0.568±0.151	**0.878±0.129**
Dim.=4	5%	**0.950±0.033**	0.869±0.035	0.871±0.024	0.832±0.027	0.712±0.217	0.477±0.053	0.387±0.082	0.429±0.148	0.432±0.152	0.934±0.050
#Inst.=1372	10%	**0.964±0.013**	0.904±0.042	0.891±0.024	0.873±0.028	0.698±0.175	0.490±0.0590	0.373±0.083	0.411±0.240	0.410±0.243	0.950±0.045
USPS	1%	0.730±0.144	0.710±0.199	0.719±0.152	0.706±0.166	**0.806±0.001**	0.746±0.191	0.745±0.193	0.630±0.131	0.631±0.135	0.729±0.120
#Class=2	3%	**0.836±0.084**	0.810±0.029	0.798±0.039	0.699±0.067	0.806±0.001	0.805±0.004	0.806±0.001	0.650±0.077	0.651±0.081	0.773±0.064
Dim.=241	5%	**0.892±0.036**	0.823±0.014	0.808±0.004	0.739±0.053	0.806±0.000	0.805±0.002	0.806±0.001	0.689±0.033	0.689±0.032	0.823±0.036
#Inst.=1500	10%	**0.907±0.035**	0.869±0.021	0.816±0.012	0.740±0.064	0.806±0.001	0.797±0.012	0.801±0.010	0.695±0.074	0.695±0.074	0.847±0.030
COIL2	1%	0.559±0.056	0.483±0.045	0.505±0.025	0.512±0.027	0.529±0.025	0.496±0.021	0.493±0.012	0.543±0.053	0.541±0.054	**0.560±0.056**
#Class=2	3%	**0.654±0.067**	0.525±0.072	0.562±0.042	0.564±0.068	0.563±0.043	0.504±0.025	0.500±0.024	0.544±0.080	0.543±0.080	0.625±0.055
Dim.=241	5%	0.712±0.048	0.599±0.041	0.639±0.027	0.581±0.020	0.590±0.048	0.513±0.023	0.514±0.024	0.542±0.063	0.542±0.063	**0.716±0.042**
#Inst.=1500	10%	**0.776±0.038**	0.659±0.053	0.668±0.046	0.627±0.039	0.602±0.058	0.510±0.023	0.508±0.021	0.525±0.049	0.525±0.049	0.767±0.030

supervised clustering algorithms on seven real-world data set, which are all publicly available at the UCI machine learning repository (http://archive.ics.uci.edu/ml) and the benchmark data sets (http://olivier.chapelle.cc/ssl-book/benchmarks.html).

To check the classification performance against different levels of label scarcity, these real-world data sets are first randomly split into training data and test data. Afterwards, for the training data, we randomly remove the labels of instances, resulting in the percentage of label instances ranging from 1% to 10%. Table I gives a summary of the prediction performance for different algorithms. From this table, we can see that ReSSL yields good results on all these real-world data sets, and the results are also relatively stable with different percentages of labeled instances available. Interestingly, even if only 1% labeled instances are used for training, ReSSL does achieve amazing results.

C. Evaluation on Real-world Data Streams

In this section, we also evaluate ReSSL on public real-world data streams including shuttle[1], electricity[2], covtype[2] and sensor[2], to demonstrate its generality.

Table I summaries the performances of different data stream classification algorithms at different levels of label scarcity ranging from 70% to 99%. From the table, we can see that ReSSL yields a successful results, and outperforms the baseline algorithms IBLStream and SPASC. In summary, regardless of static real-world data sets or streaming data sets, ReSSL supports a reliable prediction, and shows its superiority on the state-of-the-art algorithms.

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/shuttle/

[2] http://moa.cms.waikato.ac.nz/datasets/

Table II
THE PERFORMANCES OF DIFFERENT DATA STREAM CLASSIFICATIONS ON REAL-WORLD DATA SETS

Data	#Cla.	Dim.	#Inst.	LP	ReSSL Stream	IBL Stream	SPASC (Heuristic)	SPASC (Bayes)
Shuttle	9	9	43500	1%	**0.924**	0.780	0.776	0.778
				3%	**0.941**	0.869	0.786	0.793
				5%	**0.965**	0.907	0.818	0.817
				10%	**0.978**	0.935	0.847	0.851
				30%	**0.990**	0.946	0.856	0.858
Electric	2	8	45312	1%	**0.557**	0.430	0.477	0.477
				3%	**0.599**	0.513	0.527	0.561
				5%	**0.615**	0.541	0.526	0.552
				10%	**0.668**	0.642	0.523	0.561
				30%	**0.724**	0.705	0.526	0.526
CovType	7	54	581012	1%	**0.726**	0.527	0.614	0.549
				3%	**0.814**	0.628	0.588	0.562
				5%	**0.849**	0.682	0.583	0.585
				10%	**0.879**	0.710	0.582	0.572
				30%	**0.915**	0.770	0.627	0.590
Sensor	57	5	2219803	1%	**0.203**	0.099	0.032	0.029
				3%	**0.357**	0.150	0.035	0.036
				5%	**0.437**	0.229	0.038	0.040
				10%	**0.547**	0.366	0.038	0.039
				30%	**0.667**	0.571	0.033	0.037

VII. CONCLUSION

In this paper, we propose a reliable semi-supervised learning framework, called ReSSL, for the classification of both static and streaming data with partial labeled data. While existing approaches focus on exploiting the available unlabel data from different viewpoints, explicitly or implicitly depend on some assumptions, and the classification performance may suffer if the assumptions do not hold in real-world scenarios. Instead of relaxing assumptions or proposing new criterion to utilize unlabeled data, we do model the basic cluster assumption, quantify the degree of assumption violation for each cluster, use the cluster-level information and labeled data for final prediction.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (61403062, 61433014, 41601025), China Postdoctoral Science Foundation (2014M552344, 2015M580786, 2015T80973), Science-Technology Foundation for Young Scientist of SiChuan Province (2016JQ0007), Fundamental Research Funds for the Central Universities (ZYGX2014J053, ZYGX2014J091) and the National Key Research and Development Program of China (2016YFB0502303).

REFERENCES

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

[2] J. Wang, T. Jebara, and S.-F. Chang. Semi-supervised learning using greedy max-cut. *The Journal of Machine Learning Research*, 14(1):771–800, 2013.

[3] A. Fujino, N. Ueda, and K. Saito. A hybrid generative/discriminative approach to semi-supervised classifier design. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 764, 2005.

[4] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the tenth international workshop on artificial intelligence and statistics*, volume 1, pages 57–64, 2005.

[5] Y.-F. Li and Z.-H. Zhou. Towards making unlabeled data never hurt. *IEEE PAMI*, 37(1):175–188, 2015.

[6] M.-F. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*, pages 89–96, 2004.

[7] X. Zhu, Z. Ghahramani, J. Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.

[8] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006.

[9] R. K. Ando and T. Zhang. Two-view feature generation model for semi-supervised learning. In *ICML*, pages 25–32, 2007.

[10] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967.

[11] J. Shao, X. He, C. Böhm, Q. Yang and C. Plant. Synchronization-inspired partitioning and hierarchical clustering. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):893–905, 2013.

[12] J. Shao et al. Scalable Clustering by Iterative Partitioning and Point Attractor Representation. *ACM Transactions on Knowledge Discovery from Data 11(1): 5, 2016.*

[13] W. Q. F. Cao, M. Ester and A. Zhou. Density-based clustering over an evolving data stream with noise. In SDM*, pages 328–339, 2006.*

[14] Z.-H. Zhou and M. Li. Tri-training: Exploiting unlabeled data using three classifiers. IEEE TKDE*, 17(11):1529–1541, 2005.*

[15] N. Settouti, M. El Habib Daho, M. El Amine Lazouni, and M. A. Chikh. Random forest in semi-supervised learning (co-forest). In WoSSPA*, pages 326–329, 2013.*

[16] Y.F. Li , J.T. Kwok, Z.H. Zhou. Semi-supervised learning using label mean. ICML*, pages 633–640, 2009.*

[17] S. Melacci, and B. Mikhail. Laplacian support vector machines trained in the primal. The Journal of Machine Learning Research*, 12, 1149–1184, 2011.*

[18] A. Shaker and E. Hllermeier. IBLStreams: a system for instance-based classification and regression on data streams. Evolving Systems*, 3(4): 235–249, 2012.*

[19] M. J. Hosseini, A. Gholipour, and H. Beigy. An ensemble of cluster-based classifiers for semi-supervised classification of non-stationary data streams. Knowledge and Information Systems*, pages 1–31, 2015.*