Applied Soft Computing xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Incremental Semi-Supervised classification of data streams via self-representative selection

Q1 Zhixi Feng, Min Wang*, Shuyuan Yang, Licheng Jiao

Key Lab of National Radar Signal Processing, Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, International Collaboration Joint Lab in Intelligent Perception and Computation, Xidian University, Xi'an, Shaanxi 710071, China

ARTICLE INFO

Article history:
Received 20 April 2015
Received in revised form 29 January 2016
Accepted 12 February 2016
Available online xxx

Keywords:
Incremental learning
Semi-supervised classification
Self-representative selection
Data streams
Big data

ABSTRACT

Incremental learning has been developed for supervised classification, where knowledge is accumulated incrementally and represented in the learning process. However, labeling sufficient samples in each data chunk is of high cost, and incremental technologies are seldom discussed in the semi-supervised paradigm. In this paper we advance an Incremental Semi-Supervised classification approach via Self-Representative Selection (IS^{3}RS) for data streams classification, by exploring both the labeled and unlabeled dynamic samples. An incremental self-representative data selection strategy is proposed to find the most representative exemplars from the sequential data chunk. These exemplars are incrementally labeled to expand the training set, and accumulate knowledge over time to benefit future prediction. Extensive experimental evaluations on some benchmarks have demonstrated the effectiveness of the proposed framework.

1. Introduction

Q3 Data today is more deeply woven into the fabric of our daily lives than ever before due to the rapid improvement of digital technology of storage and information processing. Very recent few years have witnessed an explosive growth of data, where continuously collected data streams accounts for a large and important part [1,2]. From the perspectives of computation and machine intelligence, one should establish a data-driven machine that is capable of incrementally analyzing large-scale dynamic data stream, and accumulating knowledge incrementally over time to benefit future learning and decision-making process [3–11]. Consequently, a machine learning paradigm, Incremental Learning (InLe), is developed where the learning process takes place according to the newly emerged examples [12–21]. Compared with traditional supervised learning, InLe is capable of learning new information from sequential examples to facility the decision-making process. It is very suitable for applications where examples do not always arrive simultaneously, and the newly arriving data may bring a new perspective, may even change the statistical distribution of data. Moreover, from the biological viewpoint, InLe is more consistent with human learning where human beings already use possessed knowledge along with the experiences for learning and decision making.

Nowadays many incremental learning architectures [22,23] and algorithms [12–15,20,21,35] have been developed to deal with data streams, which can be categorized as Absolute Incremental Learning (AInLe) and Selective Incremental Learning (SInLe). In AInLe, new data are analyzed separately, and new features are formed and combined with the existing ones. In SInLe, the selected training set based on the proximity and impact of new data and new information are retrained in light of new information. Most of available InLe approaches are SInLe, which do not assume the availability of a sufficient labeled dataset before the learning, but the training examples appear over time. However, in real-life scenarios, new examples are not always labeled timely. In practical, massive amounts of data are collected dynamically in very rapid mode, resulting in the difficulty of offering labeled samples over time. For example, labeling examples from surveillance and mobile sensor network data streams is infeasible both in time and resource. On the other hand, preparing a sufficiently large number of labeled training samples at the very beginning is practically impossible, for the changing environment where new characteristic of samples or even new kind of samples are generated over time. Consequently, it is necessary to automatically update an existing training set in an incremental fashion to accommodate new information, by adding newly emerged samples to the training set.

Although the classification of data streams are characteristics of scarce labeled examples, enormous number of sequentially incoming samples are available. Because learning from labeled as well as unlabeled data is very useful for incremental learning,

Q2 * Corresponding author. Tel.: +86 88202312.
E-mail address: wangmin@xidian.edu.cn (M. Wang).

http://dx.doi.org/10.1016/j.asoc.2016.02.023
1568-4946/© 2016 Published by Elsevier B.V.

semi-supervised learning technologies can be developed by exploiting unlabeled data to modify and refine the classifier or discriminate criteria to improve classification accuracy [24–26]. Different with AInLe and SInLe, Semisupervised Incremental Learning (SSInLe) first builds knowledge base incrementally from the available labeled data. Then with the unlabeled data, SSInLe updates and restructures the knowledge incrementally. Finally it makes decisions about the new instance on the basis of the knowledge base and update the training set.

SSInLe is very important from various real-time learning perspectives, but few works have done on it. In order to explore both the labeled and dynamic unlabeled samples for a more accurate prediction of data streams, in this paper we advance an Incremental Semi-Supervised classification approach via Self-Representative Selection (IS^{3}RS), for data streams classification. In the SSInLe, an important issue is to identify relevant unlabeled data that can be added to the existing training set. In our method, an incremental self-representative data selection strategy is proposed to find the representative exemplars from the sequential data chunk. These exemplars are incrementally labeled to expand the training set, to accumulate knowledge over time to benefit future prediction. Inspired by the representation learning theory [27], we aim to find a subset of data that efficiently describe the entire data set. It assumes that each data in a dataset can be represented as a linear combination of a limited number of exemplars, which is regarded as a compact representation of data set. By adding some initial exemplars to the labeled set, a new training set can be obtained. Then we can acquire the labels of exemplars by co-training technique [28] via self-representation of each data chunk. The most confidently recovered testing data is added into training set to facilitate the learning.

The remained of this paper is organized as follows: In Section 2, the incrementally semi-supervised framework and self-representation are detailed. In Section 3, some experiments are taken on several datasets to validate the efficiency of our proposed method. The configurations, results and discussions of experiments are given. Conclusions and discusses are presented in Section 4.

2. Incremental semi-supervised learning via Self-Representative Selection (IS^{3}RS)

The proposed IS^{3}RS approach is illustrated in Fig. 1, which consists of three phases: self-representative selection, co-training, and finial decision. First each data chunk is self-represented to determine its exemplars. Under the framework of co-training, labels of these exemplars are predicted by the K-nearest neighbor (KNN) classifier. Then the training set is expanded by adding the most confident exemplars together with their predicted labels. Finally, the final classification is performed based on the expanded training set. In the following we describe each step in detail.

2.1. Self-Representative Selection of exemplars

As described in [27,36], the representative training data plays a key role in deciding the performance of learning algorithm. Therefore, learning representative data from vast amount of data is of great importance when building effective classifier or other prediction for data streams. In the data chunk classification, a key factor is whether the learning machine can take advantage of the representative testing data to construct a compact training set. Among various kinds of representative selection methods, sparsity inspired representation learning attracts a lot of interests because of its simple principle and feasibility. Moreover, it does not need to cast any distribution prior on data and present convincing performance. In this paper, we learn exemplars by a self-representation of data, under the assumption that there exist some exemplars, and each data in the dataset can be described as a linear combination of those exemplars. Mathematically, given a data set $\mathbf{X} \in \Re^{D\times N}$ with some D-dimensional data $\mathbf{x}_i$, where D is the dimensionality of data and N is the number of samples in the data set. We would like to select an informative data subset that can represent the whole dataset. Selecting exemplars can be reduced to the following optimization problem,

$$\begin{cases} \min\limits_{S} \left\|\mathbf{X} - \mathbf{XS}\right\|_F^2 \\ s.t. \left\|\mathbf{S}\right\|_{row,0} \le k \end{cases} \tag{1}$$

where $\mathbf{S} \in \Re^{N\times N}$ is the coefficient matrix and $\left\|\mathbf{S}\right\|_{row,0}$ counts the number of nonzero rows of $\mathbf{S}$. In other words, we expect to select at most $k(k \ll N)$ samples in $\mathbf{X}$ that can best represent $\mathbf{X}$. These k informative samples are called as exemplars. This is a self-representation model, where the dictionary is the data set itself. The property makes the obtained exemplars coincide with the actual data point which can be well revealed the whole data set. By minimizing the reconstruction error of each data point as a linear combination of the examples in the dataset and enforcing $\left\|\mathbf{S}\right\|_{0,q} \le k$, ($\|^{\bullet}\|_{0,q}$ norm is defined as $\left\|\mathbf{S}\right\|_{0,q} = \sum_{i=1}^{N} I(\left\|s^i\right\|_q > 0)$, and s^i is the i-th row of coefficient matrix $\mathbf{S}$ and $I(\bullet)$ denotes the

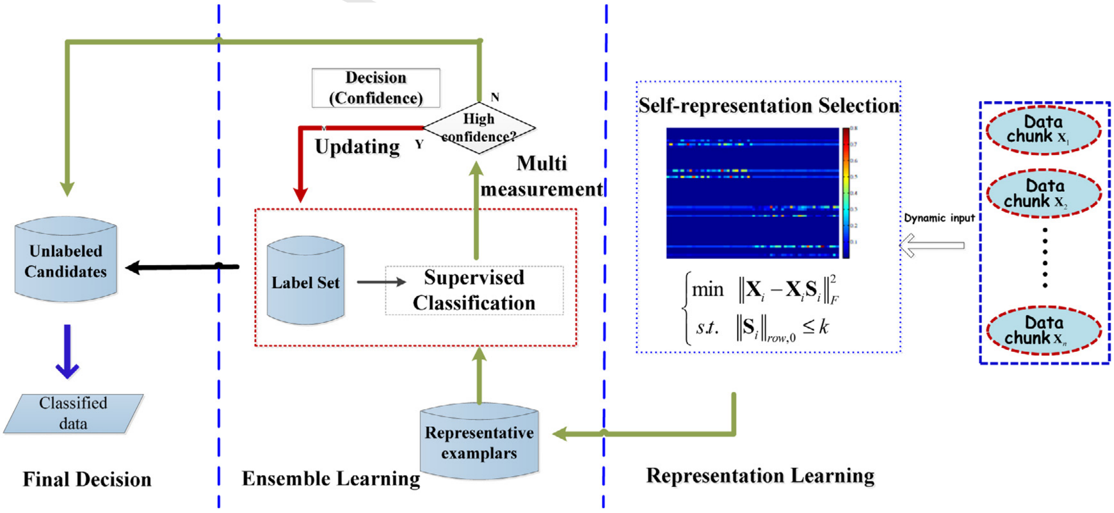

Fig. 1. An illustration of the proposed IS^{3}RS approach.

indicator function, we can determine the indices of nonzero rows correspond to the exemplars. Thus the above optimization problem can be represented as:

$$\begin{cases} \min\limits_{S} \left\| \mathbf{X} - \mathbf{XS} \right\|_F^2 \\ s.t. \left\| \mathbf{S} \right\|_{0,q} \le k \end{cases} \tag{2}$$

This is an NP-hard problem as it requires searching over every subset greedily. A standard relaxation of this problem is obtained as:

$$\begin{cases} \min\limits_{\mathbf{S}} \left\| \mathbf{X} - \mathbf{XS} \right\|_F^2 \\ s.t. \left\| \mathbf{S} \right\|_{1,q} \le \tau \end{cases} \tag{3}$$

where $\|\bullet\|_{1,q}$ norm is $\left\| \mathbf{S} \right\|_{1,q} = \sum_{i=1}^{N} \left\| s^i \right\|_q$, which is the sum of q-norm of the each rows in $\mathbf{S}$ and $\tau > 0$ is a positive threshold parameter. In this paper, we experimentally assign q as 2.

Using Lagarange multipliers, we rewrite the above optimization problem in (3) as:

$$\min_{\mathbf{S}} \lambda \left\| \mathbf{S} \right\|_{1,q} + \frac{1}{2} \left\| \mathbf{X} - \mathbf{XS} \right\|_F^2 \tag{4}$$

As the method of Multipliers, we introduce an auxiliary equivalent variable $\mathbf{J}$ for $\mathbf{S}$, that is $\mathbf{J} = \mathbf{S}$, which allows the optimization problem to be more easily solved. Thus the augmented Lagarange form of Eq. (4) can be formulated as:

$$\min_{\mathbf{S},\mathbf{J},Y_1} \lambda \left\| \mathbf{J} \right\|_{1,q} + \frac{1}{2} \left\| \mathbf{X} - \mathbf{XS} \right\|_F^2 + \langle Y_1, \mathbf{J} - \mathbf{S} \rangle + \frac{\mu}{2} \left\| \mathbf{J} - \mathbf{S} \right\|_F^2 \tag{5}$$

where $\langle \bullet \rangle$ denotes the trace operator, Y_1 denotes the Lagarange multiplier and μ is penalty parameter. The added last term in (5) is used to make the intermediate variable $\mathbf{J}$ equal to the variable $\mathbf{S}$, and the added third term in (5) is an augmented Lagarange regularizer term. Then we can use ADMM technique to alternately optimize these variables iteratively. The above optimization problem can be easily implemented in an alternating manner by using Alternating Directing Method of Multipliers (ADMM) optimization algorithm [28]. As soon as the sparse coefficients $\mathbf{S}$ are obtained, the exemplars can be determined as to the indices of nonzero row of $\mathbf{S}$. This self-representative selection can select some representative samples to reduce the redundancy of the data set.

2.2. Updating training set by co-training exemplars

As mentioned above, the classification performance largely depends on the training set, when the initial labeled set is limited and unlabeled samples are increasing chunk by chunk. In our work, we aim to construct a representative and informative training set during all the learning process. We attempt to find some informative samples to enhance the learning results. In this section, the exemplars are labeled by means of co-training techniques introduced in [29,30]. As discussed in [30], we simply split the features of each sample into two dependent parts (two views) randomly and use KNN classifiers to estimate the labels of these exemplars. The most confident exemplars that are classified into the same class by different classifiers, are added to the training set.

Specifically, an initial training set L_0 is given before the learning. Denote X_j as the j-th data chunk received between time t_{j-1} and t_j, X_j^{Rp} be the preliminary representative exemplars of data chunk X_j obtained via self-representation learning, L_{j-1} be the labeled set at time t_{j-1}. Then we perform a collaboration co-learning on X_j^{Rp} and the training set T_{j-1}, and add the most confident exemplars together with their labels into L_{j-1} to form a new labeled set L_j. Mathematically, $L_i = Y_i^{Rp} \cup L_{i-1}$.

Table 1
The main procedures of the proposed IS^{3}RS algorithm.

```
**Input:** data chunks $\mathbf{X} = [X_1 X_2 \ldots X_{i-1} X_i X_{i+1} \ldots X_N \ldots]$; class number $C$;
   parameter $\lambda$, $\tau$
**Output:** Classification results;
**Initialize** the labeled data set $L_0$;
***Repeat***
  Determine the exemplars $X_i^{Rp}$ from $X_i$ via self-representation learning;
  Choose the most confident exemplars $Y_i^{Rp}$ by performing a co-training on
   exemplars $X_i^{Rp}$ and $L_{i-1}$;
  Add $Y_i^{Rp}$ to the labeled set $L_{i-1}$ to form a new training set $T_i$, and let $L_i = T_i$;
***Until data stops***
**Do** classification of the data with KNN classifier.
```

Note that exemplars that are predicted as belonging to two or more classes will be excluded from the recovered exemplars. Finally, the recovered exemplars with their estimated labels are combined to formulate the representative training set. Based on this training set, we classify the testing data using KNN classifier. The objective of IS^{3}RS algorithm is to design an effective training set by exploiting the useful information from the testing data to improve the classification accuracy. The main procedure of IS^{3}RS is summarized in Table 1. By means of co-training technique, the exemplars that are selected by unsupervised representation learning, are prone to be confidently labeled to form an informative and representative training set. Since data come chunk by chunk, one can accumulate knowledge by a small number of exemplars with low storage and computation cost. Moreover, the selection can be extended to a distribution algorithm and taken on a parallel platform, if a large scale of data need to be processed.

3. Experimental results and discussions

In our experiments, we use Synthetic dataset, USPS digital dataset and some UCI datasets (http://archive.ics.uci.edu/ml/) to evaluate the proposed IS^{3}RS method. Some aspects are investigated in our experiments, including: (1) an investigation on the efficiency of the proposed self-representative selection strategy; (2) an investigation on the performance of the proposed ISSC approach; (3) an investigation on the classification results of IS^{3}RS algorithm, and a comparison of IS^{3}RS with some related incremental approaches, including: ADAIN.MLP [5], ADAIN.SVR [5], Learn^{++} [33] and IMORL [34]; (4) an investigation on the computational complexity of IS^{3}RS algorithm. All experimental simulations are performed with MATLAB R2013a on a personal computer with 3.2 GHz Intel Core i5-3470 CPU and 4.0GB RAM.

3.1. Investigation on the proposed self-representative selection strategy

In this experiment, we use two datasets (one synthetic data set and one USPS digital data set) to demonstrate the efficiency of the proposed representative selection strategy.

1) ***Synthetic dataset:*** We first construct 3 independent subspaces whose bases $\{U_i\}_{i=1}^3$ are computed by $U_{i+1} = TU_i$, where T is a random rotation and U_1 is a random orthogonal matrix of dimension 100 × 4. Then we generate 100 × 120 data matrix $\mathbf{X} = [X_1 X_2 X_3]$ by randomly sample 40 data points from each subspace by $\mathbf{S}_i = \mathbf{U}_i \mathbf{C}_i$, $1 \le i \le 3$ being a 4 × 40 with $\mathbf{C}_i$ being a i.i.d. $N(0, 1)$ matrix.
2) ***USPS digital dataset:*** The USPS digital data set contains 10 classes of hand draft characters. Each sample is a digital gray scale image with size 16 × 16.

In this test, we first use the self-representation learning to find the exemplars in the Synthetic data set. Fig. 2 illustrates the

Please cite this article in press as: Z. Feng, et al., Incremental Semi-Supervised classification of data streams via self-representative selection, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.02.023

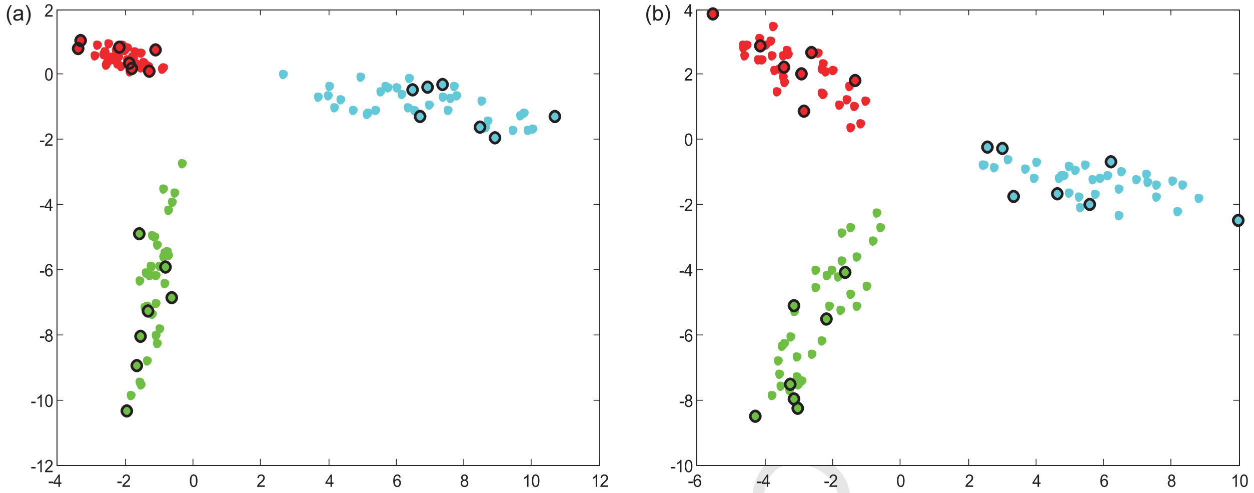

Fig. 2. An illustrative example of self-representation learning. Three colors indicate three classes, and the point with black circle is the selected exemplar (a) Clean data points and their exemplars. (b) Noisy data points with additive noise (zero mean and variance 0.2).

exemplars. The samples are represented as points in a 2D feature space after a dimensionality reduction operator by Principle Component Analysis (PCA). Some data points are randomly chosen to be corrupt, the observed data are formulated by adding Gaussian noise with zero mean and 0.2 $\|x_2\|$ variance respectively ($\|x\|$ most range from 0.1 to 0.4 in this experiment).

From Fig. 2, we can find that the self-representation learning selects the samples near the boundary which can well represent the corresponding class. Even for the noisy data, the representation learning can learn the noisy data, which indicated that seriously corrupted data are often incoherent with clean data, and they prone to be classified to a new class that does not appear in the previous training set. Because a sample belongs to a new class can be regarded as an outlier or serious noisy data, when compared with the other samples in the training set, it is expected to be selected. Consequently, all the classes in the dataset can be found. Fig. 3 shows two USPS digital number exemplars obtained by self-representation, from which we can see that the exemplars are informative.

An important question still remains for the proposed algorithm, i.e. to what extent or under what assumption that the proposed method can benefit the finial decision-making process? In this simulation, we discussion it and take an experiment to demonstrate how the proposed algorithm generates the representative labeled set. We first randomly choose 50 samples from the whole dataset for each class and then divided them into five chunks with identical size of 100 samples per chunk, to form a subset named SubUSPS. Each chunk is enforced to contain at most four classes of digital characters. The detailed description of each chunk is given in Table 2. The size of the chunk and the class of each chunk can be randomly initialized. In our simulation, we set the number of samples in each chunk as 100, and the classes as 3 or 4.

Firstly, self-representation learning is performed for each data chunk to obtain its exemplars. (The second and third columns in Table 3 show the class indexes and the corresponding number of exemplars in each chunk). For the Chunk 1, we query the labels of initial exemplars by means of initial labeled set. The remaining 4 exemplars of chunks are sequentially fed to the labeled set to update the training set. Meanwhile, we predict the subsequent exemplars that are selected from the training set until 10 classes are found.

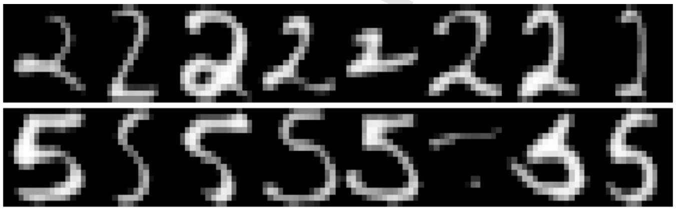

Fig. 3. The self-representative exemplars of number 2 and 5.

From the Table 3, we can find that the query classes and their corresponding number. (class – #. No.): (0/3/5 – 4/5/7)→(0/3/2/7 – 1/1/2/3)→(2/7/1/4 – 3/2/2/2)→(1/4/6/9 – 2/2/3/2)→(6/9/8 – 3/2/5). The bold class number in the sixth column denotes the new classes that are selected in the previous active annotation. The results demonstrate that new patterns are easily chosen as representative exemplars since it cannot be classified as the other classes. Thus a complete training set can be built, and we can use the learned labeled set to classify the SubUSPS. Finally a classification accuracy 99.20% can be obtained with SVM classifier.

3.2. Experiments on the proposed incremental algorithm

To validate the performance of the proposed incremental algorithm, four real-world data sets with varied size and number of classes from UCI machine learning repository [http://archive.ics.uci.edu/ml/] are employed for empirical study in the following test [5]. A detail description of the four data sets can be found in Table 4. In this simulation, each data set is sliced into chunks with size between 150 and 300. At each run, one chunk is selected to be added to the training set according to its arriving order, and the subsequent chunks are fed to the classifier according to its arriving order.

In this experiment, we have included some of state-of-art incremental learning algorithms including: ADAIN.MLP [5], ADAIN.SVR [5], Learn^{++} [33] and IMORL [34]. Our major focus here is to demonstrate that the proposed IS^{3}RS algorithm can learn the informative and representative training set and labeled set, by predicting the

Table 2
A detail description of SubUSPS chunks.

Chunk	Classes	# Number
Chunk 1	0; 3; 5	25; 25; 50
Chunk 2	0; 3; 2; 7	25; 25; 25; 25
Chunk 3	2; 7; 1; 4	25; 25; 25; 25
Chunk 4	1; 4; 6; 9	25; 25; 25; 25
Chunk 5	6; 9; 8	25; 25; 50

Please cite this article in press as: Z. Feng, et al., Incremental Semi-Supervised classification of data streams via self-representative selection, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.02.023

Table 3
Q8 An example of the labeled set update.

Chunk	Exemplars		Training set		Labeled set	
	Classes	# No.	Classes	# No.	Classes	# No.
Chunk 1	0; 3; 5	4; 5; 7	0; 3; 5	4; 5; 7	∅	∅
Chunk 2	0; 3; 2; 7	3; 4;4;4	0; 3; 5;2;7	7;9;7;4;4	**0; 3; 5**	4; 5; 7
Chunk 3	2; 7; 1; 4	6; 3; 2;5	0; 3; 5; 2; 7; 1; 4	5; 6; 7; 8; 6; 2; 5	0; 3; 5; **2; 7**	5; 6; 7; 2; 3
Chunk 4	1; 4; 6; 9	2; 5; 4;2	0; 3; 5; 2; 7; 1; 4; 6; 9	5; 6; 7; 5; 5; 4; 7; 4; 2	0; 3; 5; 2; 7; **1; 4**	5; 6; 7; 5; 5; 2; 2
Chunk 5	6; 9; 8	5; 4; 7	0; 3; 5; 2; 7; 1; 4; 6; 9; 8	5; 6; 7; 5; 5; 4; 6; 8; 6; 7	0; 3; 5; 2; 7; 1; 4; **6; 9**	5; 6; 7; 5; 5; 4; 4; 3; 2
. . .	. . .	. . .	. . .	. . .	0; 3; 5; 2; 7; 1;4; 6; 9; **8**	5; 6; 7; 5; 5; 4; 6; 6; 4; 5

Table 4
A detail description of four UCI data sets.

Class name	# Features	# Samples	# Class
Spambase	57	4601	2
Magic	10	19,020	2
Waveform	40	5000	3
Statlog	36	6435	6

most informative exemplars. By using the accumulated knowledge over time, we can subsequently add the most confident samples to update the label. For KNN classifier, we use Euclid distance and L1-Norm as a distance measure, and the number of neighbors is set as 1 and 3 respectively. Table 5 gives the numerical results of these data sets, including the Overall Accuracy (OA, the total classification accuracy that is defined as the ratio of the number of correctly classified examples to total examples), Average Accuracy (AA, the average value of classification accuracies for each class) as well as Kappa Coefficient (KC, an accuracy assessment that is defined as the ratio of (*Po* − *Pe*) to 1 − *Pe*, where *Po* and *Pe* are the observed label agreement and expected label agreement respectively). In essence, for classification tasks, the Kappa Coefficient measures the association between the ground truth labels to the labels that acquired by classifiers and helps to evaluate the predicted labels.

It can be observed from Table 5 that, in most cases, the proposed algorithm obtained the best numerical results compared to other methods. But in some cases, the classification result is not the best. This is perhaps due to the fact that not all the informative exemplars are included in the labeled set.

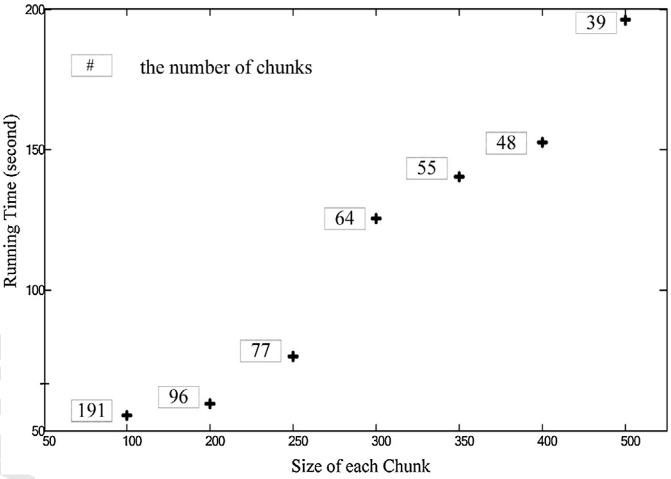

Fig. 4. The running time of our algorithm under different chunk size (Magic dataset).

3.3. Investigation on the computational complexity

In this experiment, we test the Magic data set to analyze the computational complexity of the proposed framework, when different size of chunk is used. The running time of the initial exemplars selection procedure for all chunks is shown in Fig. 4 with different predefined chunk size. The number in the box indicates the number of samples in the chunk. Q4 Though the number of chunks decreases with respect to the increase of the chunks size, the time is mainly decided by the chunk size not the number of

Table 5
A comparison of the proposed IS^{3}RS algorithm with some state-of-art methods.

Data set	Method	Classification accuracy								
		Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	OA	AA	KC
Spambase	ADAIN.MLP [5]	0.8820	0.9352	–	–	–	–	0.9142	0.9086	0.8197
	ADAIN.SVR [5]	0.8990	0.9205	–	–	–	–	0.9120	0.9098	0.8164
	Learn^{++} [33]	0.8532	**0.9561**	–	–	–	–	0.9143	0.9046	0.8209
	IMORL [34]	**0.9106**	0.8929	–	–	–	–	0.9000	0.9018	0.7929
	IS^{3}RS	0.9072	0.9344	–	–	–	–	**0.9237**	**0.9208**	**0.8405**
Magic	ADAIN.MLP [5]	0.9315	0.7137	–	–	–	–	0.8459	0.8226	0.6697
	ADAIN.SVR [5]	0.9319	0.7395	–	–	–	–	0.8644	0.8357	0.6928
	Learn^{++} [33]	**0.9523**	0.6786	–	–	–	–	0.8547	0.8155	0.6665
	IMORL [34]	0.8404	0.7836	–	–	–	–	0.8205	0.8120	0.6130
	IS^{3}RS	0.9117	**0.7980**	–	–	–	–	**0.8717**	**0.8549**	**0.7162**
Waveform	ADAIN.MLP [5]	0.7843	0.8230	0.8193	–	–	–	0.8132	0.8089	0.7223
	ADAIN.SVR [5]	0.7576	0.8198	0.8474	–	–	–	0.8077	0.8083	0.7204
	Learn^{++} [33]	0.7870	0.8360	0.9072	–	–	–	**0.8547**	**0.8434**	0.7694
	IMORL [34]	0.7575	0.8000	0.8009	–	–	–	0.7814	0.7861	0.6902
	IS^{3}RS	**0.8776**	**0.8466**	0.8050	–	–	–	0.8384	0.8431	**0.7720**
Statlog	ADAIN.MLP [5]	0.9602	0.9131	0.9169	0.4837	0.6417	0.8494	0.8387	0.8005	0.8040
	ADAIN.SVR [5]	0.9584	0.8889	0.9305	0.5180	0.7235	0.8345	0.8471	0.8144	0.8163
	Learn^{++} [33]	0.9696	0.8860	**0.9327**	0.5651	0.6958	**0.8545**	**0.8558**	0.8228	**0.8272**
	IMORL [34]	0.9000	0.8918	0.8566	**0.5653**	0.6841	0.7897	0.8079	0.7851	0.7687
	IS^{3}RS	**0.9708**	**0.9803**	0.8837	0.5217	**0.8143**	0.7862	0.8519	**0.8298**	0.8186

Please cite this article in press as: Z. Feng, et al., Incremental Semi-Supervised classification of data streams via self-representative selection, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.02.023

chunks, which can be seen from Fig. 4. The reason lies in the fact that the optimization needs to update variables by using a singular value decomposition of a matrix, whose computational complexity relies on the size of chunks. Therefore, if the computational time is limited, we should limit the chunk size.

As mentioned at the beginning, we are now in the era of big data, since the data are large both in the dimensionality and volume. When dealing with large scale data, we can extend the proposed algorithm to a distributed version and realize it on a parallel platform. For each data chunk, we can find its representative individually on a slave machine, and then synthesize the exemplars to a master machine to update the training set by adding the most confident exemplars.

4. Conclusion

In this paper, we proposed a new incremental semi-supervised learning framework via representation learning for stream data classification. The key idea of this new algorithm is to improve the classification performance based on the information incrementally learned from the testing data. Representative learning is used to obtain informative exemplars of the stream data, and co-training technique is used to label the exemplars. We investigate the effectiveness of the proposed algorithm on some benchmark datasets, and compare it with some state-of-the-art results on incremental learning. The results show that our method can find informative exemplars to enlarge the training set and gradually find new classes. Moreover, our method can achieve higher classification results than its counterparts. The proposed algorithm has potential business applications in stock forecasting and other data mining tasks. So it can be embedded into a business forecasting software to deal with large scale data streams or "big" dataset. Future work will be taken on an extension of our method to a distributed version and a realization on a parallel computing platform.

Q5 Uncited references

[31,32].

Acknowledgements

Q6 This work was supported by the National Basic Research Program of China (973 Program) under Grant no.2013CB329402, the Fundamental Research Funds for the Central Universities BDY021429, the Huawei Innovation Research Program, the Kunshan Innovation Institute of Xidian University, National Science Foundation of China under Grant no. 91438103, 91438201, 61072108, 61173090, NCET-10-668, the Foreign Scholars in University Research and Teaching Programs (No. B07048), and the fundamental research funds for the Central Universities under grant no. BDY021429, 2013KJXX-63.

References

[1] J. Podesta, P. Pritzker, E. Moniz, J. Holdren, J. Zients, Big Data: Seizing Opportunities, Preserving Values, Exec. Off. Pres. White House Washington, Study, 2014.

[2] R. Ranjan, Streaming big data processing in datacenter clouds, IEEE Cloud Comput. 1 (May (1)) (2014) 78–83.

[3] S. Pang, S. Ozawa, N. Kasabov, Incremental linear discriminant analysis for classification of data streams, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 35 (5) (2005) 905–914. Q7

[4] S.R. Rangari, M.S.S. Dongre, L.G. Malik, Machine learning and knowledge discovery in databases, in: Classification of Stream Data in the Presence Of Drifting Concept, 2013, pp. 79–94.

[5] H. He, S. Chen, Incremental learning from stream data, IEEE Trans. Neural Netw. 22 (12) (2011) 1901–1914.

[6] P.M. Hall, A.D. Marshall, R.R. Martin, Incremental eigenanalysis for classification, BMVC 98 (1998) 286–295.

[7] M. Pratama, S.G. Anavatti, P.P. Angelov, E. Lughofer, PANFIS: a novel incremental learning machine, IEEE Trans. Neural Netw. Learn. Syst. 25 (January (1)) (2014) 55–68.

[8] E. Rehn, D. Maltoni, Incremental learning by message passing in hierarchical temporal memory, Neural Comput. 26 (August (8)) (2014) 1763–1809.

[9] S.L. Ho, Y. Shiyou, Y. Bai, J. Huang, A robust metaheuristic combining clonal colony optimization and population-based incremental learning methods, IEEE Trans. Magn. 50 (February (2)) (2014) 677–680.

[10] C. Wei-Yuan, J. Chia-Feng, A fuzzy model with online incremental SVM and margin-selective gradient descent learning for classification problems, IEEE Trans. Fuzzy Syst. 22 (April (2)) (2014) 324–337.

[11] G. Ditzler, P. Robi, Incremental learning of concept drift from streaming imbalanced data, IEEE Trans. Knowl. Data Eng. 25 (October (10)) (2013) 2283–2301.

[12] R. Polikar, L. Upda, S.S. Upda, V. Honavar, Learn^{++}: an incremental learning algorithm for supervised neural networks, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 31 (November (4)) (2001) 497–508.

[13] F. Guorui, H. Guang-Bin, L. Qingping, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Trans. Neural Netw. 20 (August (8)) (2009) 1352–1357.

[14] M.H.C. Law, A.K. Jain, Incremental nonlinear dimensionality reduction by manifold learning, IEEE Trans. Pattern Anal. Mach. Intell. 28 (March (3)) (2006) 377–391.

[15] D.L. Wang, B. Yuwono, Incremental learning of complex temporal patterns, IEEE Trans. Neural Netw. 7 (November (6)) (1996) 1465–1481.

[16] R. Elwell, R. Polikar, Incremental learning of concept drift in nonstationary environments, IEEE Trans. Neural Netw. 22 (October (10)) (2011) 1517–1531.

[17] S. Ozawa, P. Shaoning, N. Kasabov, Incremental learning of chunk data for online pattern classification systems, IEEE Trans. Neural Netw. 19 (June (6)) (2008) 1061–1074.

[18] S. Wan, L.E. Banta, Parameter incremental learning algorithm for neural networks, IEEE Trans. Neural Netw. 17 (November (6)) (2006) 1424–1438.

[19] M. Karasuyama, I. Takeuchi, Multiple incremental decremental learning of support vector machines, IEEE Trans. Neural Netw. 21 (July (7)) (2010) 1048–1059.

[20] Y. Cao, H. He, H. Huang, LIFT: a new framework of learning from testing data for face recognition, Neurocomputing 74 (February (6)) (2011) 916–929.

[21] Z.H. Zhou, Z.Q. Chen, Hybrid decision tree, Knowl.-Based Syst. 15 (8) (2002) 515–528.

[22] E.C. Wang, A. Kuh, A smart algorithm for incremental learning, in: International Joint Conference on Neural Networks, vol. 3, 1992, pp. 121–126.

[23] B.T. Zhang, An incremental learning algorithm that optimizes network size and sample size in one trial, in: IEEE World Congress on Computational Intelligence, vol. 1, 1994, pp. 215–220.

[24] C. Yang, L. Bruzzone, R. Guan, L. Lu, Y. Liang, Incremental and decremental affinity propagation for semisupervised clustering in multispectral images, IEEE Trans. Geosci. Remote Sens. 51 (March (3)) (2013) 1666–1679.

[25] P. Kulkarni, Incremental Learning and Knowledge Representation, Reinforcement and Systemic Machine Learning for Decision Making, 2012, pp. 177–208.

[26] V. Bhatnagar, R. Dobariyal, P. Jain, A. Mahabal, Data understanding using semi-supervised clustering, in: Conference on Intelligent Data Understanding (CIDU), 24–26 October, 2012, pp. 118–123.

[27] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[28] Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Comput. Math. Appl. 2 (1) (1976) 17–40.

[29] M.-F. Balcan, A. Blum, K. Yang, Co-training and expansion: towards bridging theory and practice, in: Advances in Neural Information Processing Systems, 2004, pp. 89–96.

[30] K. Nigam, R. Ghani, Analyzing the effectiveness and applicability of co-training, in: Proceedings of the Ninth International Conference on Information and Knowledge Management, 2000, pp. 86–93.

[31] F. Chung, S. Wang, Z. Deng, et al., Clustering analysis of gene expression data based on semi-supervised visual clustering algorithm, Soft Comput. 10 (11) (2006) 981–993.

[32] Y. Shi, F. Chung, S. Wang, An improved TA-SVM method without matrix inversion and its fast implementation for nonstationary datasets, IEEE Trans. Neural Netw. Learn. Syst. 26 (9) (2015) 2005–2018.

[33] R. Polikar, L. Udpa, S. Member, S.S. Udpa, V. Honavar, Learn^{++}: an incremental learning algorithm for supervised neural networks, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 31 (4) (2001) 497–508.

[34] H. He, S. Chen, IMORL: incremental multiple-object recognition and localization, IEEE Trans. Neural Netw. 19 (10) (2008) 1727–1738.

[35] M. Wang, S. Yang, B. Wu, Hierarchical representation learning based spatio-temporal data redundancy reduction, Neurocomputing 173 (2016) 298–305.

[36] S. Yang, M. Wang, Y. Chen, Y. Sun, Single-image super-resolution reconstruction via learned geometric dictionaries and clustered sparse coding, IEEE Trans. Image Process. 21 (9) (2012) 4016–4028.

Please cite this article in press as: Z. Feng, et al., Incremental Semi-Supervised classification of data streams via self-representative selection, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.02.023