Expert Systems With Applications 91 (2018) 364–373

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Sparse coding based classifier ensembles in supervised and active learning scenarios for data classification

Göksu Tüysüzoğlu [a], Yusuf Yaslan [b,*]

[a] Department of Computer Engineering, Engineering Faculty, Dokuz Eylül University, İzmir, Turkey
[b] Computer Engineering Department, Faculty of Computer and Informatics Engineering, Istanbul Technical University, Maslak, Istanbul 34469, Turkey

ARTICLE INFO

Article history:
Received 7 April 2017
Revised 24 June 2017
Accepted 9 September 2017
Available online 11 September 2017

Keywords:
Active learning
Dictionary learning
Ensemble classifiers
Random subspace feature selection
Bagging

ABSTRACT

Sparse coding and dictionary learning has recently gained great interest in signal, image and audio processing applications through representing each problem instance by a sparse set of atoms. This also allows us to obtain different representations of feature sets in machine learning problems. Thus, different feature views for classifier ensembles can be obtained using sparse coding. On the other hand, nowadays unlabelled data is abundant and active learning methods with single and classifier ensembles received great interest. In this study, Random Subspace Dictionary Learning (RDL) and Bagging Dictionary Learning (BDL) algorithms are examined by learning ensembles of dictionaries through feature/instance subspaces. Besides, ensembles of dictionaries are evaluated under active learning framework as promising models and they are named as Active Random Subspace Dictionary Learning (ARDL) and Active Bagging Dictionary Learning (ABDL) algorithms. Active learning methods are compared with their Support Vector Machines counterparts. The experiments on eleven datasets from UCI and OpenML repositories has shown that selecting instance and feature subspaces for dictionary learning model increases the number of correctly classified instances for the most of the data sets while SVM has superiority over all of the applied models. Furthermore, using an active learner generally increases the chance of improved classification performance as the number of iterations is increased.

1. Introduction

Nowadays, an abundant amount of latent information is available in databases, web pages or data repositories to be exploited for intelligent decision making. In supervised learning tasks, these databases contain data that is related to a specific category or class. The process to investigate to which class these data points should belong by using the training data samples whose class/category information are known is called classification. There are a number of problem domains where classification takes place such as text categorization (Tang, Kay, & He, 2016), optical character recognition (Mehta, Singla, & Mahajan, 2016), fraud detection (Sharma & Panigrahi, 2013), face detection (Wan, Chen, Zhang, Zhang, & Wong, 2016), classification of proteins (Cao & Xiong, 2014) etc.

In order to obtain a good classification accuracy finding a suitable feature representation plays a fundamental role. In literature, there is a vast amount of research to represent the features in other dimensional spaces to enhance the classification performance such as kernels (Wang, Zhang, Zhou, Tang, & Li, 2015), wavelet transformation (Wang, Song, & Liu, 2016), frequency representation of time domain signals (Sejdić, Djurović, & Jiang, 2009).

On the top of feature representation, image, audio and video types can be sparsely represented by applying transform-domain methods (Elad, 2010). A lot of significant tasks related to such media can be handled by finding sparse solutions to underdetermined systems of linear equations. Regarding this issue, sparse coding and dictionary learning have recently aroused much interest by representing feature vectors as linear combinations of basis element of a dictionary.

Dictionary learning has been applied in many problem areas such as signal processing applications (Tosic & Frossard, 2011), image segmentation (Dahl A B, 2015), music genre classification (Yeh & Yang, 2012) and saliency detection (Zhu, Chen, & Zhao, 2014). One of the major application areas is in data representation and classification/clustering applications.

Sprechmann and Sapiro (2010) proposed a clustering framework based on a set of dictionaries which forms each cluster by providing the best representation for the signals of that cluster and giving the sparsest solution. The experimental results obtained for

* Corresponding author.
E-mail addresses: goksu.tuysuzoglu@ceng.deu.edu.tr (G. Tüysüzoğlu), yyaslan@itu.edu.tr (Y. Yaslan).

http://dx.doi.org/10.1016/j.eswa.2017.09.024
0957-4174/

the model's classifier counterpart were conducted on three standard datasets, the MNIST and USPS. According to the results, the proposed dictionary learning model provides remarkable classification performance comparable with other sophisticated classification algorithms such as SVM and k-NN in terms of reconstruction and discrimination power.

In order to classify different music genres (Yeh & Yang, 2012), a dictionary learning based technique was developed to summarize short-time features (codebook) of recorded music over time, where codebook is made up of sub-dictionaries for each class. The proposed method was shown superior to other existing codebook generation methods such as conventional VQ-based and exemplar-based methods.

Tosic and Frossard (2011) presented dictionary learning and sparse approximation as a dimensionality reduction tool to find a representation adaptive to the proper inference of causes of the observed data. In addition, supervised dictionary learning was examined in a face recognition application by using the discriminative power of the sparse representation.

Recently, in order to improve the accuracy of a single classifier ensemble methods have gained interest. Ensemble classifiers can be obtained by training different classifiers on datasets which are obtained using data/feature resampling methods or trained on a single training dataset by different classifiers or single classifier with different parameters (Tuysuzoglu, Moarref, & Yaslan, 2016). Ensemble learning methods are used for classification problems as well as regression. Classifier ensembles can be obtained either in feature space, instance space or classifier level. Boosting, bootstrap aggregating (bagging), stacking, random subspace feature selection, random forests and adaboost are among the most applied ensemble learning methods (Polikar, 2006).

Bagging is an instance-based ensemble learning method which generates subspaces of instances by applying random selection method with replacement. Each ensemble classifier produces a decision and the final prediction is their combined output. On the other hand, random subspace feature selection is a feature-based counterpart of bagging model, where a sub-group of features are randomly selected with replacement to form ensemble classifiers. Taking advantage of the strengths of these two ensemble learning methods, classification problems can be solved more accurately and the variance of the individual classifiers are reduced.

Obtaining labelled training examples for classification problems is an expensive task while a massive chunk of unlabelled data is available to process. For instance, let us think of a case where we want to predict which web pages a person can find interesting. In order to do this, we need the data of web pages which were marked as favourite by this person. The more we know about the labelling information, we can predict better and present more appropriate pages to recommend. On the other side, people are generally not willing to hand-label all the pages they like even if there are a lot. Active learning is a largely used framework for these kind of situations. It has the ability to choose the most informative unlabelled examples automatically for human annotation.

Up to the present, active learning framework has been applied with many different classifiers for text classification (Hu, Mac Namee, & Delany, 2016), image retrieval (Qi & Zhang, 2016), advertisement removal (Sun & Hardoon, 2010), visual object detection (Abramson & Freund, 2006), natural language processing (Olsson, 2009) etc. In this paper, we extend our previous works (Tüysüzoğlu & Yaslan, 2016; Tuysuzoglu et al., 2016) by using dictionary learning algorithm as a base classifier for active learning and classifier ensembles. In Tüysüzoğlu and Yaslan (2016), random dictionary active learning algorithm was compared with SVM based classifier ensembles without best parameter search. In this paper in addition to RDL, we also propose to use bagging algorithm with dictionary learning both in supervised learning and active learning scenarios. In the new experimental results in order to have fair comparisons, we also optimized SVM parameters with grid search.

The remainder of the paper is structured as follows. Section 2 introduces the theoretical motivation for the applied methodology. In the first step, sparse signal representation and dictionary learning models are explained. Then, ensemble learning methods in general is discussed and detailed knowledge on random subspace feature selection and bagging ensemble classifiers are provided. Furthermore, active learning framework is stated by expanding different sampling scenarios used throughout literature. In the last part of the applied methodology, sparse coding based ensemble classifiers and their counterparts under active learning framework are proposed. Section 3 discusses datasets and toolboxes which have been used to obtain classifier models. In Section 4 experimental results achieved are explained while Section 5 concludes the paper with some discussion of the potential significance of our results and some directions for future work.

2. Materials and methods

2.1. Notations

Throughout the paper, bold uppercase letters are used to denote matrices, bold lowercase letters to denote vectors and italics are used to display scalars. Let $\mathbf{X} = \{\mathbf{x_i}\}_{i=1\ldots m} \in R^{m\ x\ n}$ be a matrix including training data where $\mathbf{x_i} \in R^{m\ x\ 1}$ is an input signal, and $\mathbf{X'} \in R^{wxn}$ is the test set, $\mathbf{D = [D_1, D_2, \ldots, D_k]} \in R^{n\times k}$ be the dictionary matrix and $\boldsymbol{\alpha} = \{\alpha_i\}_{\boldsymbol{i}=1\ldots\boldsymbol{m}} \in R^{m\ x\ k}$ is the sparse coefficient vector of the signal $\mathbf{X}$ where each α_i is the representation of the signal $\mathbf{x_i}$. $\mathbf{y} \in R^m$ indicates training class labels, $\mathbf{y'}$ points predicted class labels for test instances, K is the number of ensemble dictionaries, c is the number of classes, m is the number of instances in the training set, n is the number of features and k is the number of atoms in the initial dictionary. $||\boldsymbol{\alpha}||_0$ is the l_0 norm of sparse vector $\boldsymbol{\alpha}$. ε is the noise parameter and λ is the penalty parameter of the dictionary learning model to balance the sparsity of the decomposition and the reconstruction error. In the ensemble learning, for random subspace part, the whole feature subspace is displayed as $\mathbf{X_{rs}}$, and the selected feature subspace at ith iteration as $\mathbf{X_{rs\_i}}$, s is the number of selected feature/instances and for bagging, $\mathbf{X_{Bagged}}$ is instance subspace whose instances are randomly drawn from the original dataset $\mathbf{X}$ and h denotes the predictor model.

2.2. Dictionary learning and sparse signal approximation

2.1.1. Sparse signal approximation

Recently there is a great interest in sparse representations of signals in image and signal processing community. Sources of data such as voice signals, images, radar images or heart signals etc. carry overwhelming amounts of data which is difficult to directly extract. Therefore, having a sparse representation plays an important role in processing signals faster and simpler with few coefficients.

We are on the search for a model for an input data as $\mathbf{X} \in R^n$. The next step is to construct $\mathbf{D = [D_1, D_2, \ldots, D_k]} \in R^{n\times k}$ as a dictionary which is a set of prototype signals i.e. a set of normalized $(\mathbf{D_j^T D_j} = 1)$ "basis vectors". $\boldsymbol{\alpha} \in R^k$ represents the sparse coefficient vector of the signal, then the problem is formulated as:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \text{ s.t. } \mathbf{X}=\mathbf{D}\boldsymbol{\alpha} \tag{1}$$

where $||\boldsymbol{\alpha}||_0$ is the l_0 norm of sparse vector $\boldsymbol{\alpha}$. In order to measure sparsity, l_p norm is used for a given p. If p is equal to 2, we do not really get what we want. Because what we want is to penalize with an equal amount every one of the entries of $\boldsymbol{\alpha}$ is non-zero.