Pattern Recognition 48 (2015) 3714–3727

Contents lists available at ScienceDirect

Pattern Recognition

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

Decremental Sparse Modeling Representative Selection for prototype selection

F. Dornaika [a,b,*], I. Kamal Aldine [a]

[a] Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, San Sebastian, Spain
[b] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

ARTICLE INFO

Article history:
Received 28 October 2014
Received in revised form
4 May 2015
Accepted 18 May 2015
Available online 30 May 2015

Keywords:
Prototype selection
Data self-representativeness
Subspace representation
Regularization
Block sparsity
Image classification
Object recognition

ABSTRACT

Selecting few representatives or examples that can efficiently and reliably describe a set of data has always been a challenging task in computer vision and pattern recognition. Recently, *Sparse Modeling Representative Selection* (*SMRS*) was proposed as a powerful filter method for selecting the most relevant examples/instances in subspaces. The selection is achieved by ranking the examples using the L_2 norm of the associated row in a coding matrix. This coding matrix is computed using data self-representativeness (the dictionary is given by the examples themselves) adopting block sparsity regularization. In this paper, we propose a decremental Sparse Modeling Representative Selection (D-SMRS) in which the selection of the representatives is broken down into several nested processes. The key contribution is a new scheme of sparse modeling that proceeds by progressive coding and pruning. It proceeds by eliminating outlier and noisy samples in the first stages so that the final stage (coding and selection) is performed on clean data. Thus, the final instance selection will not be heavily affected by the presence of outliers and aberrant samples. The proposed method was qualitatively and quantitatively evaluated. The qualitative evaluation concerned image selection and video summarization. The quantitative evaluation was performed on six benchmark image datasets using several state-of-the art selection methods with four different classifiers: 1-Nearest Neighbor (NN), Nearest Subspace (NS), Sparse Representation Classifier (SRC), and Support Vector Machines (SVM). The outlier rejection ability of the proposed method is also studied on real images. In all cases the selection computed by our algorithm achieved or outperformed existing state-of-the-art results.

1. Introduction

1.1. Overview

Finding a subset of examples, known as representatives or exemplars, that can efficiently and reliably describe the entire dataset, is a very important issue in the analysis of scientific data, with a lot of applications in machine learning, data recovery, signal processing, image processing, etc. Due to the effectiveness of instance selection for speeding up training processes, many methods have been proposed [4,24,27,21,16,35]. The selected representatives can summarize datasets of images, videos, texts or Web documents. Finding a small number of examples which replaces the learning database have two main advantages: (i) reducing the memory space needed to store data and (ii) improving the computation time of classification algorithms. For example, the method of nearest neighbors (NN) is more efficient [18] when comparing test samples to few representatives rather than to all training samples. A reduced training dataset can also speed up the training process in the sense that the classifier learning becomes less computationally expensive. For pattern recognition tasks, it is also required that the overall performance will not be considerably affected by the data reduction.

The problem can be stated as follows: given a training set T, the goal of an instance selection method is to obtain a subset $S \subseteq T$ such that S does not contain superfluous instances and $Acc(S) \simeq Acc(T)$ where $Acc(X)$ is the classification accuracy obtained using the subset X as training set. Instance selection methods can either start with $S = \varnothing$ (incremental methods) or $S = T$ (decremental methods). The difference is that the incremental methods include instances in S during the selection process and decremental methods remove instances from S along the selection.

Like in feature selection, according to the strategy used for selecting instances, we can divide the instance selection methods into two groups: (i) wrapper methods in which the selection criterion is based on the accuracy obtained by a classifier (commonly, those instances that do not contribute with the classification accuracy are

* Corresponding author at: Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, Manuel Lardizabal, 1, 20018 San Sebastian, Spain. Tel.: +34 943018034; fax: +34 943015590.
E-mail address: fadi.dornaika@ehu.es (F. Dornaika).

http://dx.doi.org/10.1016/j.patcog.2015.05.018
0031-3203/

discarded from the training set) (e.g. [12,9]), and (ii) filter methods in which the selection criterion uses a selection function which is not based on a classifier (e.g., [29]). Most of the instance selection algorithms (e.g., [10,25,37,41]) are strongly related to the use of the k-NN classifier. One can also find instance selection algorithms that do not restrict the use of a specific classifier. Examples of this kind of algorithms are the evolutionary ones (e.g. [17,19]), which use the accuracy of a classifier as selection criterion. In these algorithms, an instance is deleted whenever it does not contribute for either maintaining or improving the classification accuracy.

A good review on wrapper and filter methods can be found in [32].

The filter algorithms can be divided into two main categories. The first category finds representatives from data contained in one or several subspaces of reduced dimensionality. The algorithm *Rank Revealing QR* (*RRQR*) [5,7] tries to select a few data points through finding a permutation of the data which gives the best conditioned submatrix. *Greedy* and *Randomized* algorithms have also been proposed in order to find a subset of columns in a reduced rank matrix [35,4,3].

The second group of algorithms finds representatives assuming that there is a natural grouping of data collection based on an appropriate measure of similarity between pairs of data points [24,16,8,20]. Accordingly, these algorithms generally work on the similarity/dissimilarity between data points to be grouped. The *Kmedoids* algorithm [24], which can be considered as a variant of *Kmeans* [13], supposes that the data are located around several centers of classes, called medoids, which are selected from the data. Another algorithm based on the similarity/dissimilarity of data points is the (*Affinity propagation*) (AP) [16,20]. This algorithm tries to find representatives from the similarities between pairs of data points by using a message passing algorithm. Although AP has suboptimal properties and can find approximate solutions, it does not require any initialization (like Kmeans and Kmedoids) and has shown good performance in problems such as unsupervised image categorization [14,26].

1.2. Paper contribution and motivation

Recently a new filter method, called *Sparse Modeling Representative Selection* (*SMRS*) [15], has been proposed to find representatives and is based on setting every data sample as a linear combination of the whole dataset with a block-sparsity constraint. SMRS [15] runs a single coding based on the whole dataset. Thus, the resulting relevance scores can be inaccurate. In this paper, we propose to overcome this shortcoming by introducing a decremental Sparse Modeling Representative Selection in which the selection of the representatives is broken down into several nested processes. Thus, each pass exploits the knowledge acquired from the previous stage. We will show that a decremental selection scheme can outperform SMRS and many state-of-the art methods. The key contribution is a new scheme of sparse modeling that proceeds by progressive coding and pruning. It proceeds by iteratively invoking the coding scheme on an updated set of samples.

The paper is structured as follows: in Section 2, we describe some related works. In Section 3, we provide a brief review of the Sparse Modeling Representative Selection. In Section 4, we describe our proposed Decremental Sparse Modeling Representative Selection. The objectives of the performance evaluation are presented in Section 5. Section 6 presents a qualitative evaluation. Section 7 provides a quantitative evaluation that quantifies the classification performance based on the selected instances. Section 8 studies the robustness of the proposed method in the presence of outliers. Finally, we provide some concluding remarks in Section 9. In the sequel, capital bold letters denote matrices and small bold letters denote vectors.

2. Related work

In this section, we present some of the main instance selection methods. ENN (edited nearest neighbor) [40] is an instance selection algorithm commonly used as noise filter. ENN deletes the noisy instances as follows: it discards an instance belonging to the training set T when it does not coincide with the majority class of its k nearest neighbors, in particular, ENN uses $k=3$.

In [41], the authors proposed DROP1, DROP2, DROP3, DROP4, DROP5 (Decremental Reduction Optimization Procedure); these methods are based on the concept of associate. The associates of an instance p are those instances such that p is one of their k nearest neighbors. DROP1 discards an instance p from T if the associates of p in S are correctly classified without p; through this rule, DROP1 discards noisy instances since the associates of a noisy instance can be correctly classified without it but in DROP1, when the neighbors of a noisy instance are first eliminated, then the noisy instance will not be discarded. In order to solve this problem, DROP2 is similar to DROP1 but the associates of an instance are searched in the whole training set, that is, p is removed only if its associates in T are classified correctly without p. DROP3 and DROP4 first discard noisy instances using a filter similar to ENN and then they apply DROP2. DROP5 is based on DROP2 but it starts discarding the nearest enemies (i.e., nearest heterogeneous samples).

In [25], another method based on the *Reacheable*(p) and *Coverage*(p) sets is presented. In this method, the *Reacheable*(p) is neighbors set and the *Coverage*(p) concept only considers the associates with the same class as p in order to discard instances in the same class. Before discarding an element, this technique determines whether an instance is noisy, superfluous or critical. In this context, an instance is critical when its deletion affects the classification of other instances; in particular this method discards either noisy or superfluous (but non-critical) instances. When $|Coverage(p)| < |Reachable(p)|$ then p is considered as noisy; p is superfluous when it is correctly classified by *Reachable*(p).

The ISR (Instance Selection based on Ranking) [36] computes the relevance of an instance within the training set through the typicality concept (set to the quotient of the instances average similarity with the rest of instances in its class and the average similarity with all those instances of a different class). Once the relevance has been computed, ISR uses a wrapper process that includes in S (processing instances in a descending order according to their relevance) those instances yielding the highest number of correctly classified instances using T as test set.

Ref. [28] performed clustering on each class and searched the nearest cluster center from the opposite class to get instances near the decision boundary. The assumption of no possible class overlap, which is unpractical, is adopted.

In recent years several effective approaches for selecting boundary instances were proposed. Ref. [33] gave the Concept Boundary Detection (CBD) algorithm, which consists of two stages: concept-independent preprocessing and concept specific sampling. Neighbors of each instance are first identified. And then, the score of each instance is computed and boundary instances for each concept are determined. CBD shows better effect. However, the score of an instance may be dominated by noisy data very close to it. Several works have exploited the concept of class border in order to select representative instances (e.g., [30,33]). In [2], the authors proposed Fast Condensed Nearest Neighbor (FCNN) rule to select instances. With this rule a training-set-consistent subset is obtained and used to train SVM. This method can sharply reduce the size of a training dataset, but often decreases classification accuracy. In [31], the authors proposed Prototype Selection by Relevance (PSR) method for prototype selection. PSR selects the most relevant prototypes per class in a