Pattern Recognition Letters 74 (2016) 61–67

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Incremental Similarity for real-time on-line incremental learning systems☆

Marta Režnáková*, Lukas Tencer, Mohamed Cheriet

École de technologie supérieure (ÉTS), Synchromedia Lab, Montreal, Quebec, Canada

ARTICLE INFO

Article history:
Received 29 May 2015
Available online 4 February 2016

Keywords:
Incremental learning
Online learning
Incremental Similarity
Classification

ABSTRACT

The expectation of higher accuracy in recognition systems brings the problem of higher complexity. In this paper we introduce a novel Incremental Similarity (IS) that maintains high accuracy while preserving low complexity. We apply IS to on-line and incremental learning tasks, where the need of low complexity is of significant need. Using IS enables the system to directly compute with the samples themselves and update only few parameters in an incremental manner. We empirically prove its efficiency on several evolving models and show that by using IS they achieve competitive results and outperform the baseline models. We also consider the problem of incremental learning used to handle fast growing datasets. We present a very detailed comparison for not only evolving models, but also for the well-known batch models, showing the robustness of our proposal. We perform the evaluation on various classification problems to show the wide application of evolving models and our proposed IS.

1. Introduction

In many areas of research, the growth of the amount of data is inevitable. This is very positive for machine learning and pattern recognition, where the lack of data often results in low accuracy. However, the bigger the data is, the more time we need to process it and to train our models. Furthermore, if new data arrives, all of them need to be re-processed so that all the information is included. This may cause a delay in the usage of the system. Thus, in last few years we have been noticing more attempts for incremental learning of the models aimed at absorbing all the necessary information and at the same time lowering the burden of huge datasets.

In incremental learning, the model is incrementally built (learned) each time new data arrives. Once the information on this new data is stored, the original data is often discarded and thus the system does not have access to the original data after. Thus, the model works faster than offline batch techniques. However, besides all the benefits, there are also challenges that ask for solutions and this paper aims to solve some of them described in the following:

- Processing time vs. accuracy challenge asks how far we want to go with the complexity of our model in order to achieve better performance.
- At the beginning of the learning process, we can struggle with a lack of data, missing important information regarding the variances within classes. This occurs especially when incremental learning techniques are applied to real-time recognition and dynamic tasks, in which the recognition does not wait for the whole learning process to be finished. Using incremental techniques can make the process faster, however it should not be done so to the detriment of high recognition capabilities.

Incremental Similarity introduced in this paper allows to learn from scratch, where even a small number of samples gives reasonable estimate of the class. At the same time, with the increasing number of samples, the learning time is kept constant, learning only few non-matrix parameters and describing the samples themselves.

In the following sections we investigate through incremental learning (IL) modeling approaches (Section 2), then give a detailed description of Incremental Similarity (IS) and learning of its parameters (Section 3). We then apply IS to the baseline incremental and batch approaches to develop new models (Section 4). At the end of this paper, we evaluate the performance of these models for various applications (Sections 5 and 6).

2. Related works

In this paper we focus on a classification task, in which the aim is mainly to classify handwritten symbols. We also focus on online incremental learning techniques, where the learning

☆ This paper has been recommended for acceptance by Eckart Michaelsen.

* Corresponding author. Tel.: +1 514 396 8800.
E-mail addresses: marta.reznakova@gmail.com (M. Režnáková), lukas.tencer@gmail.com (L. Tencer), mohamed.cheriet@etsmtl.ca (M. Cheriet).

http://dx.doi.org/10.1016/j.patrec.2016.01.010
0167-8655/

and classification are done on one sample level and the model is learned by small increments.

This section briefly summarizes several groups of incremental or otherwise online techniques, out of which some focus on the clustering part only and others use clustering as one of their layers followed by other classification or regression layers.

There are many works that focus on online, incremental or adaptive learning. Usually, researchers choose an offline method and adapt it to work in an online manner. There are several attempts to transfer offline clustering methods, such as K-means or K-NN into incremental or online learning based. From these we mention [1] in which the authors propose online version of K-means, [2,3] for incremental K-NN, or [4] for incremental vector quantization (VQ).

In many applications, SVM based methods are widely used. Thus, there have also been attempts for incremental learning variations, such as in [5–7] for incremental and online SVM in regression and [8], [9] for online SVM.

There have been several attempts for incremental subspace methods, especially for images, such as in [10–12] proposing incremental PCA and [13] with online PCA.

In [14] the authors propose an online version for bagging and boosting algorithms, coming up with AdaBoost (adaptive boosting). Another versions of AdaBoost has been proposed in [15] used for vision problems, [16] and [17] for multi-class online boosting.

In this paper we focus mostly on evolving or adaptive neuro-fuzzy models usually based on Takagi–Sugeno fuzzy model [18], or less on Mamdani model [19]. They vary in the clustering part that is essential for online purposes, as in offline, where the division of the known space is easier and more straight-forward. In [20] the authors propose to solve the clustering part using Recursive Mountain Clustering for space division and the creation of rules, which they combined with univariate normal distribution (antecedent part) and Recursive Least Squares (consequent part). This research was updated in [21] using Mahalanobis distance for antecedent part (similarity to rules – clusters). In [22,23] the authors utilize genetic algorithms for the rule control and generation. In [24] the authors solve the problem of evolution of fuzzy rules by using connectionist systems. In [25] we proposed to use incremental distance and clustering, followed by [26] where we proposed to use ART-2A clustering [27] for the rule generation and handling. In [28] authors propose to use incremental VQ for the space division.

To our knowledge, the works cited in [20,21,25,26] are closest to the proposition presented in our study. However, the extensive comparison and empirical proofing was not performed in any of them.

3. Incremental Similarity

To address some of the challenges of on-line learning, i.e. the complexity vs. precision, learning from scratch and learning on the fly, we introduce the Incremental Similarity (IS). We apply this to a number of baseline models to figure as a similarity measure. The more similar is sample x to a group of samples in a set a, the higher value the similarity measure takes. Thus, all the similarity measures in this work will be adjusted, if necessary, accordingly (1) with d being the distance. This results in the range of similarities to be [0, 1], where 1 is the lowest distance and highest similarity of sample x to the set a.

$$\frac{1}{1+d} \tag{1}$$

In this work we show the effectiveness of IS by a comparison to the Euclidean (ED) and the Mahalanobis (MD) distances that we both explain later in this section. The baseline models we apply all these similarity measures to, along with the detailed description of the structures within these models that the similarities are used at, are described in Section 4, namely Takagi–Sugeno based (similarity measures are used for antecedent learning) and K-means (similarity measures are used for distance from K-means).

3.1. Euclidean and Mahalanobis distances

In this section we describe two basic distance measurements ED and MD, taking into account the nature of the membership function (the firing degree) – the more similar the sample is to the baseline of the rule, the higher firing degree this rule has. Thus, we calculate the reversed squared ED (2) and the reversed MD (3) as an opposite to the univariate and multivariate distributions. Here μ_{t_i} is the mean at time t_i for rule i, i.e. the number of samples already introduced to the rule i and S_{t_i} is the covariance matrix at time t_i for rule i.

$$\beta_i = \frac{1}{1+(\mu_{t_i}-x)^T(\mu_{t_i}-x)} \tag{2}$$

$$\beta_i = \frac{1}{1+(\mu_{t_i}-x)^T S_{t_i}^{-1}(\mu_{t_i}-x)} \tag{3}$$

The update of the parameters μ_{t_i} and $S_{t_i}^{-1}$ can be derived from the univariate and multivariate normal distributions. For univariate normal distribution we have the probability of samples $x_1 \dots x_N$, $P(x_1 \dots x_N)$ further referred as P.

$$P = \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}\right\}$$

To find the parameter μ that minimizes the error we set the derivation of the logarithm of the function to zero. This will lead to an updating formula for μ (4).

$$\begin{aligned}\log P &= \sum_i\left[-\frac{1}{2}\log 2\pi - \log\sigma - \frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}\right]\\ \frac{\partial}{\partial\mu}\log P &= \sum_i \frac{x_i-\mu}{\sigma^2} = 0 \Rightarrow \sum_i [x_i-\mu] = 0\\ \mu &= \frac{1}{N}\sum_i x_i\end{aligned} \tag{4}$$

Then the recursive formula can be derived as in (5), where t_i is the time and it updates according to (6).

$$\mu_{t_i} = \frac{1}{t_i}\left((t_i-1)\mu_{t_i-1} + x_{t_i}\right);\ \mu_1 = x_1 \tag{5}$$

$$t_i = t_i + 1 \tag{6}$$

For multivariate normal distribution, the derivation of μ_{t_i} is similar to univariate distribution. The covariance matrix update is derived as follows resulting into (7).

$$\begin{aligned}P &= \prod_i \frac{1}{2\pi^{\frac{d}{2}}}|S|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(x_i-\mu)S^{-1}(x_i-\mu)^T\right\}\\ \log P &= \sum_i\left[-\frac{d}{2}\log 2\pi - \frac{1}{2}\log|S| - \frac{1}{2}D\right]\\ D &= (x_i-\mu)S^{-1}(x_i-\mu)^T\\ \frac{\partial}{\partial S}\log P &= \sum_i\left[-\frac{1}{2}S^{-1} + \frac{1}{2}(x_i-\mu)S^{-2}(x_i-\mu)^T\right] = 0\\ &\times \sum_i\left[-S + (x_i-\mu)(x_i-\mu)^T\right] = 0\\ S &= \frac{1}{N}\sum(x_i-\mu)(x_i-\mu)_i^T\end{aligned} \tag{7}$$