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).

$$\log P = \sum_i \left[-\frac{1}{2}\log 2\pi - \log\sigma - \frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}\right]$$

$$\frac{\partial}{\partial\mu}\log P = \sum_i \frac{x_i-\mu}{\sigma^2} = 0 \Rightarrow \sum_i [x_i-\mu] = 0$$

$$\mu = \frac{1}{N}\sum_i x_i \tag{4}$$

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).

$$P = \prod_i \frac{1}{2\pi^{\frac{d}{2}}}|S|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_i-\mu)S^{-1}(x_i-\mu)^T\right\}$$

$$\log P = \sum_i \left[-\frac{d}{2}\log 2\pi - \frac{1}{2}\log|S| - \frac{1}{2}D\right]$$

$$D = (x_i-\mu)S^{-1}(x_i-\mu)^T$$

$$\frac{\partial}{\partial S}\log P = \sum_i \left[-\frac{1}{2}S^{-1} + \frac{1}{2}(x_i-\mu)S^{-2}(x_i-\mu)^T\right] = 0$$

$$\times \sum_i \left[-S + (x_i-\mu)(x_i-\mu)^T\right] = 0$$

$$S = \frac{1}{N}\sum (x_i-\mu)(x_i-\mu)_i^T \tag{7}$$

To be able to recursively update directly the inverse matrix, we derive (8), where again t_i is the time and is updated according to (6).

$$\begin{aligned} S_{t_i}^{-1} &= \frac{1}{\frac{t_i-1}{t_i}S_{t_i-1} + \frac{1}{t_i}(x_{t_i}-\mu)(x_{t_i}-\mu)^T} \\ &= \frac{1}{\frac{t_i-1}{t_i}S_{t_i-1}\left[1 + \frac{1}{t_i-1}(x_{t_i}-\mu)S_{t_i-1}(x_{t_i}-\mu)^T\right]} \\ &= \frac{1}{\frac{t_i-1}{t_i}S_{t_i-1}} + \frac{-\frac{t}{(t-1)^2}S_{t_i-1}^{-1}(x_{t_i}-\mu)\left[S_{t_i-1}^{-1}(x_{t_i}-\mu)\right]^T}{\left[1 + \frac{1}{t_i-1}(x_{t_i}-\mu)S_{t_i-1}^{-1}(x_{t_i}-\mu)^T\right]} \\ S_{t_i}^{-1} &= \frac{t_i}{t_i-1}S_{t_i-1}^{-1} - \frac{t_i}{t_i-1}C \end{aligned} \tag{8}$$

$$C = \frac{S_{t_i-1}^{-1}(x_{t_i}-\mu)\left(S_{t_i-1}^{-1}(x_{t_i}-\mu)\right)^T}{t_i - 1 + (x_{t_i}-\mu)S_{t_i-1}^{-1}(x_{t_i}-\mu)^T}$$

3.2. Our novel Incremental Similarity measurement

In this paper we propose the use of Incremental Similarity (IS) in several models, with a satisfying trade-off between the computational cost and accuracy. The main reason for searching for such trade-off is the real-time setting of many on-line and incremental learning systems. Here, the computational cost is one of the crucial factors for the decision about the learning and recognition model. In our work we focus on real-time on-line learning with starting from scratch. Our method offers a simple updating formula that does not rely on covariance matrix and at the same time is able to deliver high accuracy. Rather that the distance from mean and covariances, it describes the distance from all points in the rule i (9).

$$\beta_i = \frac{1}{1 + \frac{1}{t_i}\sum_{j=1}^{t_i}\left(x - x_j\right)^2} \tag{9}$$

In order to accommodate incremental learning into the model, we need to bring up a recursive model for learning the membership function that can be easily derived from the original non-recursive one. This is achieved by substituting x^2, $\sum_{j=1}^{t_i} x_j$ and $\sum_{j=1}^{t_i} x_j^2$ using parameters α_i, η_i and γ_i (12), along with update formulas for these parameters in an incremental manner. Here, none of the samples will be forgotten or changed, but in these parameters, the true values of past samples are stored. Thus, every time new sample arrives to enrich the rule, updating formulas are triggered so that for next sample with unknown competency for a rule, membership β_i is derived (11).

$$\begin{aligned} \beta_i &= \frac{1}{1 + \frac{1}{t_i}\sum_{j=1}^{t_i}\left(x^2 - 2xx_j + x_j^2\right)} \\ &= \frac{1}{1 + x^2 - \frac{1}{t_i}2x\sum_{j=1}^{t_i}x_j + \frac{1}{t_i}\sum_{j=1}^{t_i}x_j^2} \\ \beta_i &= \frac{t_i}{t_i + t_i x^2 - 2x\sum_{j=1}^{t_i}x_j + \sum_{j=1}^{t_i}x_j^2} \end{aligned} \tag{10}$$

$$\beta_i = \frac{t_i}{t_i + t_i\alpha_i - 2x\eta_i + \gamma_i} \tag{11}$$

$$\begin{aligned} &(a)\ t_i = t_{i-1} + 1 \qquad c)\ \eta_i = \eta_{i-1} + x;\ \ \eta_0 = 0 \\ &(b)\ \alpha_i = x^2 \qquad\qquad d)\ \gamma_i = \gamma_{i-1} + \alpha_i;\ \ \gamma_0 = 0 \end{aligned} \tag{12}$$

Using this novel technique we tackle the challenge of the lack of data at the beginning of the learning process as well as the processing time vs. accuracy challenge. Since our proposed IS is not dependent on a high number of samples to create relevant statistics (as would be the case in Normal distributions), it can describe small amount of data better.

4. Models

In this section we describe four TS fuzzy models and K-means that we combine with ED, MD and IS. As we show in Section 5, the combination with IS leads to superior results compared to the models solely. Each Neuro-Fuzzy model is composed of a number of fuzzy rules in a form of *IF x IS a THEN b*. Here, the *x IS a* is the antecedent part of the rule based on the similarity or the firing degree of the rule, denoting how much x is similar to a, x being a single input sample, a being a set of samples within the rule already known (not necessarily of one class) and b being the consequence of the rule. In a classification part, each rule gives an opinion for every class recorded in the system in a form of value within range [0, 1]. Then, these values are weighted by the membership function and averaged. The winning class is assigned to the class for which the weighted averaged opinion was the highest, i.e. the most confident. In the case of K-means, we choose the class that is most present (according to labels) within each cluster. More detailed description is given in Section 4.5.

4.1. ETS

In [20] the authors introduce a novel evolving TS (ETS) fuzzy model. The generation and management of rules is handled by Recursive Mountain Clustering, in which each new sample is assigned a potential that is compared to updated potentials of known centers of rules (tops of the mountains). These centers are updated within the algorithm. These updates are based on the samples, which are added to the centers or those which replace the centers based on distance conditions. In addition, each time a new class is added to the system, a new rule is created for it as well. In the original paper, the antecedent part is derived as univariate normal density function, however it has been replaced by ED in current paper. For the consequent part, authors propose to use local RLS, in which the optimization is handled for each rule locally, rather than globally for whole system.

4.2. ETS+

In [21] the authors introduce the ETS+ model inspired by ETS. They also use Recursive Mountain clustering for rule management, however with some slight adjustments. Similarly to ETS also in ETS+ is the addition of new rules influenced by the arrival of new classes. First the multivariate normal along with Cauchy distributions were proposed for handling the antecedent part, but MD yielded better results and is used for comparison in this paper. Similarly to ETS, authors use local RLS for the consequent part.

4.3. Incremental fuzzy model (IFM)

In [25] the authors propose the use of IS for TS fuzzy models combined with very simple clustering based on the performance of the winning rule for each sample. Similarly to other approaches, it takes into account the arrival of new classes which influence the creation of new rules. It is based on the belief that on one hand each rule can, and does contain samples from more classes, but at the same time it should not be too confused in distinguishing them using consequent learning. Consequent learning itself is also based on local RLS.

Table 1
Datasets.

	# classes	# samples	# dimensions
Digits	10	10992	16
OHS-I	17	13600	21
OHS-II	20	1923	50
Segment	7	2310	19
Mushrooms	2	8124	22

4.4. ARTIST: ART-2A driven rule management in TS model

In [26], the authors propose to use the ART-2A neural network [27] for cluster management. The generation of new rules is left solely on the incremental clustering without the influence of the arrival new classes. Similarities within the antecedent part are understood as possibilities rather as probabilities. This means, that one sample can be fully possible for all (giving 1 for every rule) or none of the rules (giving 0 for all of them). Then, for the consequent part, Competitive Consequent Learning (CCL) is used to avoid the class forgetting in incremental learning.

4.5. K-means

K-means clustering algorithm is based on k clusters (means) where the distance to each of them is measured by the Euclidean formula. At each step all points are assigned to the closest mean and all means are recalculated. Then, the distance between the old and the new means is used as a stop rule for the whole process. This process iterates until the stop criterion is met.

Since we focus on incremental learning techniques, we do not update the centroids at the end of the assigning, but during this process. For this purpose we use the same updating formulas as in Section 3 (ED, MD and IS). Thus, for the K-means combined with MD, on the top of mean values μ_i we add the update of covariance matrices S_i and parameters α_i, $\eta_i$and γ_i for K-means combined with IS. In our evaluation we put the main focus on determining the distance and learning the distance parameters, not the latter phases of the method.

Since the original K-means is unsupervised clustering and our problems are supervised, we need to change it in this manner. Thus, during the assignment of samples to clusters and the recalculation of parameters, we also assign labels to each cluster. Then, we use a majority vote to determine the winning label in the cluster. In results we will notice that in order to make this method work precisely, many clusters are needed, as they must divide the space very carefully. We also need to point out, that we use K-means only to demonstrate the differences between distance measurements in this paper.

5. Results

In this section we show a detailed comparison of the distance measurements described in Section 3 to show the potential of Incremental Similarity (IS) for incremental learning purposes and as an interesting trade-off between using only mean and possibly variance and using covariance matrix. For the evaluation of our IS in comparison to different baseline metrics, we use five datasets described in Table 1[1]. In this section we show the results on only three of all datasets, with the rest displayed online[2].

[1] Digits, Mushrooms and Segments can be found at UCI Machine Learning Repository; Online Handwritten Symbols I (OHS-I) can be found at http://www.synchromedia.ca/web/ets/gesturedataset ; Online Handwritten Symbols II (OHS-II) can be found at http://www.irisa.fr/ .

[2] Complete results are published at http://www.synchromedia.ca/membres/marta/IncrementalSimilarityResults.html.

Table 2
Digits A – complete dataset results, evolving fuzzy models (%/s/s).

	ED	MD	IS
ETS	85.4/**45**/4.9	**87.1**/55.2/5.4	85.2/49.5/**4.8**
ETS+	84.9/**49.2**/4.8	**87.1**/56.4/5.4	86.0/50.4/**4.1**
IFM	91.0/**768**/**9.6**	**93.9**/1044/11.4	89.7/810/10.2
ARTIST	94.2/2094/14.4	**95.9**/2676/19.2	89.9/**1956**/**13.4**

Table 3
Digits B – last 100 samples results, evolving fuzzy models (ms/ms).

	ED	MD	IS
ETS	11.2/4.4	25.4/2.2	**6.2**/**2.1**
ETS+	8.1/3.6	21.7/6.5	**7.4**/**2.5**
IFM	37.1/13.5	188.2/23.1	**24.2**/**8.3**
ARTIST	**6.8**/17.2	126.1/**1.5**	15.1/18.2

For each dataset we display the results in four Tables (A–D, e.g. Digits A - Digits B), first two for evolving fuzzy models and last two for K-means. All the results are then composed of the following entities: Table A – accuracy for the complete dataset/time required to learn and recognize the complete dataset/time per one rule required to process the complete dataset; Table B – average time required to learn parameters for the last 100 instances/time required to derive distance for the last 100 instances; Table C – accuracy for the complete dataset/time required to learn and recognize the complete dataset; Table D – average time required to learn parameters for one instance/time required to derive distance for one instance during assignment to clusters/time required to derive distance for one instance during classification.

For each dataset we separate the results for the Neuro-Fuzzy based models (Tables A and B, e.g. Digits A, Digits B) and K-means (Tables C and D, e.g. Digits B, Digits C). For K-means we find the best parameters k for each combination (based on accuracy) using by cross-validation within the interval $<$ c, c + 5, ..., 195, 200 $>$, where c represents the number of classes for each dataset. We select 2/3 of dataset as a learning base and 1/3 as a testing base. The complete results are shown in Figs. 1 (for accuracy) and 2 (for processing time). Note, that whereas for accuracy, the best results are the highest ones, for processing time the aim are the lowest values.

In Table 2 we can notice that most precise for the Digits dataset are models using the MD achieving 91% recognition rate, followed by the ED achieving 88.9% and the IS with 87.7% recognition rate. Out of all these models, ARTIST + MD seems to be the most accurate and ETS + ED the fastest. As for the processing time, the most successful is the IS with 716.4/8.1 s on average, followed by the ED with 739/8.4 s and the MS with 957.9/10.35 s. The best processing time is achieved by ETS + ED (total processing time) and ETS + + IS (time per rule).

We need to note, that these times, especially the time for processing the complete dataset, are influenced by many factors, such as the number of rules with exponential influence on the results, algorithms used for derivation of vector and matrix multiplications, etc.

In Table 3 we compare the processing time of parameters learning and distance derivation. We can notice that processing times of derivation of the distances are similar, as the formulas are comparably complex, especially for the ED and IS. However, for the updating formulas, the best results are achieved with the IS. This is no surprise and we will encounter similar result for K-means also. It is caused by the updating formulas themselves, where for the IS we simply perform a summation while for the ED we multiply and divide. This situation can be improved by making the updates simpler.

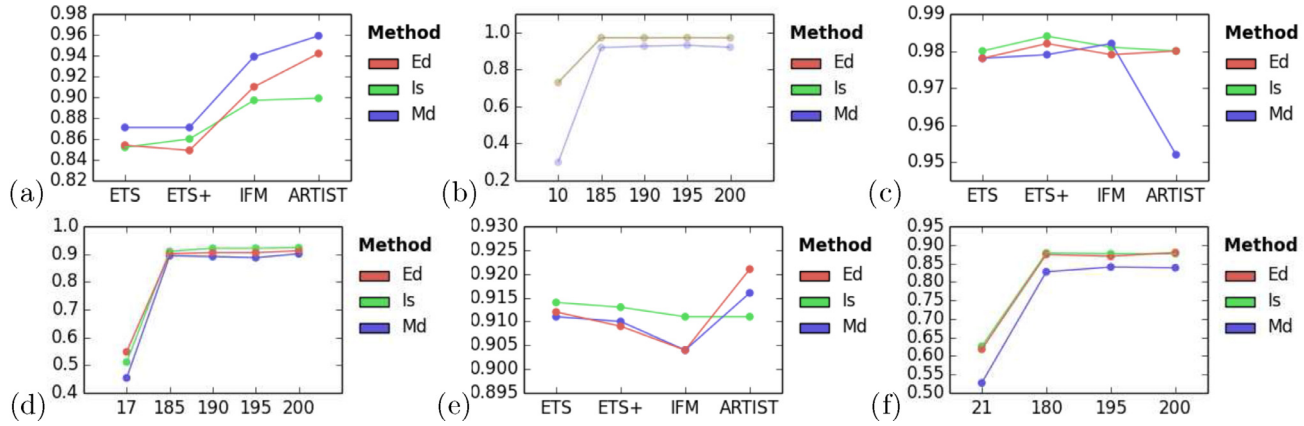

Fig. 1. Recognition rate (a), (c) and (e) for various Neuro-Fuzzy models, (b), (d) and (f) for K-means and various k, on Digits ((a), (b)), OHS-I ((c), (d)) and OHS-II ((e), (f)) datasets.

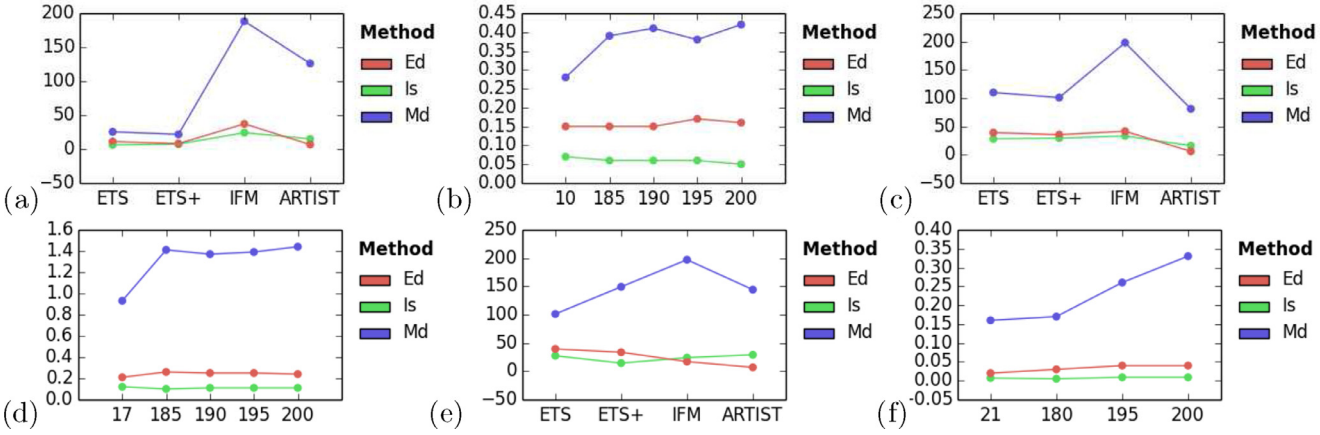

Fig. 2. Computational complexity (a), (c) and (e) for various Neuro-Fuzzy models, (b), (d) and (f) for K-means and various k, on Digits ((a), (b)), OHS-I ((c), (d)) and OHS-II ((e), (f)) datasets.

Table 4
Digits C – complete dataset results, K-means (%/s).

k	ED	MD	IS
10 (lowest)	72.9/**9.6**	29.9/49.2	**73.1**/13.4
185 (best ED)	**97.2**/**86.4**	91.8/294	**97.2**/314
190 (best IS)	97.0/**93.6**	92.6/309.6	**97.3**/312.1
195 (best MD)	97.1/**97.8**	93.1/348.6	**97.3**/302.4
200 (highest)	97.0/**90.6**	92.0/413.4	**97.3**/334.8

Table 5
Digits D – per one step results, K-means (ms/ms/ms).

k	ED	MD	IS
10	0.15/**0.003**/**0.003**	0.28/0.009/0.009	**0.07**/0.007/0.008
185	0.16/**0.003**/**0.003**	0.39/0.009/0.009	**0.06**/0.007/0.006
190	0.15/**0.003**/**0.003**	0.41/0.009/0.009	**0.06**/0.007/0.006
195	0.17/**0.003**/**0.003**	0.38/0.009/0.008	**0.06**/0.006/0.007
200	0.16/**0.003**/**0.003**	0.42/0.009/0.01	**0.05**/0.006/0.007

Table 4 shows the results for lowest, highest and best K-means for each distance for a fair comparison. We can notice that in all cases, IS with K-means outperforms both other distances, although its accuracy is comparable with the results for the K-means + ED. As for the processing time, the lowest in general is achieved by the ED, followed by the IS and the MD.

In Table 5 it is noticeable that for the learning of parameters, K-means + IS achieves the best processing time. However, we need to mention again, that although we used the well-known recursive formula for updating mean for ED, it is not needed due to the usage of means at the end of this learning process. If we use the same strategy as for the IS and at the end divide the summed samples by their total amount, the learning time would be smaller than for the IS, as that executes one more operation. Such a difference in updating will also change the formula for deriving the distance itself (as we need to derive the mean vector as well). For the formula updating used in this work, the ED achieves the best processing time, followed by the IS and the MD.

Table 6
Online handwritten symbols I A – complete dataset results, evolving fuzzy models (%/s/s).

	ED	MD	IS
ETS	97.8/**342**/**20.4**	97.8/543/31.8	**98.0**/468/25.6
ETS+	98.2/**448.8**/26.4	97.9/555.6/32.4	**98.4**/464.4/**25.2**
IFM	97.9/**1758**/**12.1**	**98.2**/1986/16.2	98.1/2101/15.1
ARTIST	**98.0**/702/26.4	95.2/1074/38.4	**98.0**/**678**/**25.2**

For the Online handwritten symbols I dataset (in Table 6), the most precise method proved to be the IS, achieving 98% rate, followed by the ED achieving 97.98% rate and the MD achieving 97.3% rate. The lowest processing time is achieved with models using the ED with 812.7/21.3 s, the IS with 927.85/22.8 s and the MD with 1039.65/29.7 s. From all the above, the combination judged the most accurate is the ETS + IS and the fastest are ETS + ED (for total processing time) and IFM + ED also (for time per rule). Using

Table 7
Online handwritten symbols I B – the last 100 samples results, evolving fuzzy models (ms/ms).

	ED	MD	IS
ETS	39.1/15.2	110/11.3	**28.1**/**9.2**
ETS+	35.5/14.7	101/25.1	**29.4**/**9.1**
IFM	41.5/14.6	198/25.2	**33.2**/**9.2**
ARTIST	**6.1**/25.2	81.1/17.4	16.1/**16.1**

Table 8
Online handwritten symbols I C – complete dataset results, K-means (%/s).

k	ED	MD	IS
17 (lowest)	**54.8**/**13.8**	45.4/97.8	51.1/52.8
185 (2nd MD)	90.1/**138.6**	89.4/889.2	**91.0**/576.6
190 (2nd ED)	90.5/**133.8**	89.1/906	**92.1**/547.2
195 (2nd IS)	90.5/**141.6**	88.7/824.4	**92.1**/552
200 (best)	91.2/**132.6**	90.1/707.4	**92.3**/487.8

Table 9
Online handwritten symbols I D – per one step results, K-means (ms/ms/ms).

k	ED	MD	IS
17	0.21/**0.005**/**0.005**	0.93/0.014/0.012	**0.12**/0.009/0.009
185	0.26/**0.004**/**0.003**	1.41/0.015/0.015	**0.1**/0.008/0.008
190	0.25/**0.004**/**0.004**	1.37/0.014/0.015	**0.11**/0.008/0.008
195	0.25/**0.004**/**0.004**	1.39/0.015/0.015	**0.11**/0.008/0.008
200	0.24/**0.004**/**0.003**	1.44/0.015/0.014	**0.11**/0.008/0.009

Table 10
Online handwritten symbols II A – complete dataset results, evolving fuzzy models (%/s/s).

	ED	MD	IS
ETS	91.2/**69.6**/**3.6**	91.1/123.6/6.1	**91.4**/81.2/4.2
ETS+	90.9/**86.4**/**4.2**	91.0/138.6/6.6	**91.3**/87/**4.2**
IFM	90.4/**228**/**1.8**	90.4/355.2/3.2	**91.1**/**228**/**1.8**
ARTIST	**92.1**/**447**/**3.1**	91.6/750.3/5.4	91.1/474/3.6

our proposed setup we are able to increase the precision and at the same time reduce the processing time.

In Table 7 we notice that the learning time of both the ED and the IS are comparable, with the IS being slightly better, followed by the MD.

In Table 8 we notice, that similarly to the Digits dataset, also for Online symbols I, using the IS leads to the best accuracy, closely followed by the ED and then by the MD. As for the processing time, using ED achieves the best performance.

In Table 9 we can notice that in terms of parameter learning, the IS leads, followed by the ED and the MD. As for the derivation of the distance itself, ED outperforms the others, followed by the IS and then the MD.

For Online Handwritten symbols II (Table 10), the most accurate models are those using the IS achieving 91.2% rate, followed by the ED with 91.15% rate and the MD with 91.0% rate. From all of the above models, ARTIST + ED seems to be the most accurate. The best performance time is achieved by the models using the ED with 207.75/3.2 s followed by the IS with 217.55/3.45 s and the MD with 341.9/5.3 s. From the results, the best processing time from all of the combinations is achieved by ETS + ED (for total processing time) and both IFM + ED and IFM + IS (for time per rule).

For parameter learning for the Online symbols II dataset (in Table 11), the ED and the IS achieve comparable results for average update on last 100 samples. The derivation of the distance is the fastest for the IS, followed by the ED and the MD.

For Online symbols II, K-means achieves the best results using the IS, while the ED achieves similarly good results, both followed

Table 11
Online handwritten symbols II B – the last 100 samples results, evolving fuzzy models (ms/ms).

	ED	MD	IS
ETS	39.5/15.1	101.2/11.5	**27.4**/**10.1**
ETS+	33.6/14.9	149.1/26.5	**14.4**/**8.1**
IFM	**17.1**/14.8	197/23.1	24.1/**7.2**
ARTIST	**6.9**/23.7	144.1/**12.9**	29.1/21.3

Table 12
Online handwritten symbols II C – complete dataset results, K-means (%/s).

k	ED	MD	IS
21 (lowest)	61.8/**1.8**	52.7/2.4	**62.6**/2.1
180 (best IS)	87.4/**6.3**	82.7/14.4	**87.8**/16.8
195 (best MD)	86.9/**7.2**	84.0/16.2	**87.6**/18.9
200 (best ED)	**87.9**/**7.2**	83.8/14.4	87.6/17.8

Table 13
Online handwritten symbols II D – per one step results, K-means (ms/ms/ms).

k	ED	MD	IS
21	0.02/**0.004**/**0.004**	0.16/0.021/0.023	**0.007**/0.008/0.007
180	0.03/**0.003**/**0.004**	0.17/0.025/0.025	**0.005**/0.007/0.006
195	0.04/**0.003**/**0.003**	0.26/0.03/0.027	**0.009**/0.007/0.007
200	0.04/**0.003**/**0.004**	0.33/0.028/0.026	**0.009**/0.007/0.007

by the MD (Table 12). As for the performance time, the fastest proved to be the ED, followed by the MD performing comparably well as the IS.

In Table 13 we can notice that the IS leads on the learning performance time, followed by the ED and the MD. As for the derivation time, the ED achieves the best results, closely followed by the IS and then the MD.

6. Conclusion and discussion

In this paper we have presented a novel Incremental Similarity (IS) measurement for incremental learning models. We evaluated our technique using five baseline models and achieved better results compared to the baseline techniques. This evaluation was performed on five datasets that varied in the number of classes, features and sample size. In all of the applications IS achieved a good trade-off between high accuracy and low computational cost. In some rare cases when our technique did not achieve the best accuracy, this slight reduction in accuracy was exchanged for a significant reduction in processing speed.

Generally we can say that IS achieves better results than the any of the baseline techniques. This means, that not only is it highly accurate, but also it is adaptable to various distributions of data. This is very intriguing for incremental learning in which data is un-known beforehand and we cannot predict their distribution. In some specific cases for K-means method, the IS is outperformed by the MD. In our opinion, this is caused by the nature of K-means, where it iterates until it does not meet the stop criterion. Thus, the longer it takes to find the best means, the higher performance time it results in.

In our future work we want to explore applications of IS to a wider range of models that do not necessarily include fuzzy models only.

Acknowledgments

The authors thank NSERC RGPIN 138344 & 2014-04649 and SSHRC 412-2010-1007 of Canada for their financial support.

References

[1] J. Beringer, E. Hüllermeier, Online clustering of parallel data streams, Data Knowl. Eng. 58 (2) (2006) 180–204.
[2] K. Forster, S. Monteleone, A. Calatroni, D. Roggen, G. Troster, Incremental kNN classifier exploiting correct-error teacher for activity recognition, in: Proceedings of the 2010 Ninth International Conference on Machine Learning and Applications, IEEE, 2010, pp. 445–450.
[3] C. Yu, R. Zhang, Y. Huang, H. Xiong, High-dimensional kNN joins with incremental updates, GeoInformatica 14 (1) (2009) 55–82.
[4] E. Lughofer, Extensions of vector quantization for incremental clustering, Pattern Recognit. 41 (3) (2008) 995–1011.
[5] M. Carozza, S. Rampone, Towards an incremental SVM for regression, in: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000), 2000, pp. 405–410.
[6] H. Wang, D. Pi, Y. Sun, Online SVM regression algorithm-based adaptive inverse control, Neurocomputing 70 (4–6) (2007) 952–959.
[7] W. Wang, C. Men, W. Lu, Online prediction model based on support vector machine, Neurocomputing 71 (4–6) (2008) 550–558.
[8] A. Bordes, The Huller: a simple and efficient online SVM, in: Proceedings of the 16th European Conference on Machine Learning (ECML 2005), Porto, Portugal, 2005, pp. 505–512.October 3–7
[9] D. Tax, Online SVM learning: from classification to data description and back, in: Proceedings of the 2003 IEEE 13th Workshop on Neural Networks for Signal Processing (NNSP'03), 2003, pp. 499–508.
[10] Y. Li, On incremental and robust subspace learning, Pattern Recognit. 37 (7) (2004) 1509–1518.
[11] D.a. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (1–3) (2007) 125–141.
[12] F. Song, H. Liu, D. Zhang, J. Yang, A highly scalable incremental facial feature extraction method, Neurocomputing 71 (10–12) (2008) 1883–1888.
[13] M. Yao, X. Qu, Q. Gu, T. Ruan, Z. Lou, Online PCA with adaptive subspace method for real-time hand gesture learning and recognition, WSEAS Trans. Comput. 9 (6) (2010) 583–592.
[14] N.C. Oza, Online bagging and boosting, in: Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 3, IEEE, 2005, pp. 2340–2345.
[15] H. Grabner, H. Bischof, On-line boosting and vision, in: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2006, pp. 260–267.
[16] A. Saffari, M. Godec, T. Pock, C. Leistner, H. Bischof, Online multi-class LPBoost, in: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3570–3577.
[17] G. Ditzler, M.D. Muhlbaier, R. Polikar, Incremental learning of new classes in unbalanced datasets: learn + + UDNC, in: Multiple Classifier Systems, 2010, pp. 33–42.
[18] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. VOL. SMC-15 (1) (1985) 116–132.
[19] E. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man Mach. Stud. 7 (1) (1975) 1–13.
[20] P.P. Angelov, D.P. Filev, An approach to online identification of Takagi–Sugeno fuzzy models, Syst. Man Cybern. Part B: IEEE Cybern. Trans. 34 (1) (2004) 484–498.
[21] A. Almaksour, E. Anquetil, S. Quiniou, M. Cheriet, Evolving fuzzy classifiers: application to incremental learning of handwritten gesture recognition systems, in: Proceedings of the 20th International Conference on Pattern Recognition, 2010, pp. 4056–4059.
[22] B. Carse, T.C. Fogarty, A. Munro, Evolving fuzzy rule based controllers using genetic algorithms, Fuzzy Sets Syst. 80 (3) (1996) 273–293.
[23] J. Gomez, D. Dasgupta, Evolving fuzzy classifiers for intrusion detection, in: Proceedings of the 2002 IEEE Workshop on Information Assurance, 2002.
[24] N. Kasabov, Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning, IEEE Trans. Syst. Man Cybern. Part B Cybern. 31 (6) (2001) 902–918.
[25] M. Režnáková, L. Tencer, M. Cheriet, Online handwritten gesture recognition based on Takagi–Sugeno fuzzy models, in: Proceedings of the 11th International Conference on Information Science, Signal Processing and Their Applications (ISSPA), 2012, pp. 1247–1252.
[26] M. Režnáková, L. Tencer, M. Cheriet, ARTIST: ART-2A driven generation of fuzzy rules for online handwritten gesture recognition, in: Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR '13), 2013.
[27] G.A. Carpenter, S. Grossberg, D.B. Rosen, ART 2-A: an adaptive resonance algorithm for rapid category learning and recognition, Neural Netw. 4 (4) (1991) 493–504.
[28] E. Lughofer, FLEXFIS: a robust incremental learning approach for evolving Takagi–Sugeno fuzzy models, IEEE Trans. Fuzzy Syst. 16 (6) (2008) 1393–1410.