Pattern Recognition Letters 73 (2016) 76–82

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Active graph based semi-supervised learning using image matching: Application to handwritten digit recognition☆

Hubert Cecotti*

Faculty of Computing and Engineering, Ulster University, Magee campus, Londonderry, BT48 7JL, Northern Ireland, UK

ARTICLE INFO

Article history:
Received 20 July 2015
Available online 3 February 2016

Keywords:
Active learning
Semi-supervised learning
Character recognition
Image matching

ABSTRACT

With the availability of large amounts of documents and multimedia content to be classified, the creation of new databases with labeled examples is an expensive task. Efficient supervised classifiers often require large training databases that are not always immediately available. Active learning approaches solve this issue by querying an expert to set a label to particular instances. In this paper, we present a novel active learning strategy for the classification of handwritten digits. The proposed method is based on a k-nearest neighbor graph obtained with an image deformation model, which takes into account local deformations. During the active learning procedure, the user is first asked to label the vertices with the highest number of neighbors. Thus, the expert sets the label to the examples that are more likely to propagate their labels to a high number of close neighbors. Then, a label propagation function is performed to automatically label the examples. The procedure is repeated until all the images are labeled. We evaluate the performance of the method on four databases corresponding to different scripts (Latin, Bangla, Devnagari, and Oriya). We show that it is possible to label only 332 images in the MNIST training database to obtain an accuracy of 98.54% on this same database (60000 images). The robustness of the method is highlighted by the performance of handwritten digit recognition in different scripts.

1. Introduction

The fast increase of new documents and multimedia content to be classified is a source of new challenges in pattern recognition and machine learning [13]. Documents and multimedia data are exponentially growing thanks to the internet and the development of portable devices that can acquire images. In addition, cultural heritage collections are being digitized, and made available through online tools. New automatic methods must be provided to both automatically index and search through the documents because not enough manpower is available to provide useful annotations on the large volume of digitized documents [32]. With the emergence of the Big data paradigm and the creation of new classification problems, pure supervised techniques may not be able to cope with the fast increase of classification tasks, which possess only few labeled examples. Because the requirement of a classification task can evolve rapidly over time, it is essential to propose a fast evaluation of the potential performance. This diagnostic may infer a different type of approach in a later stage. In multiclass classification tasks, several approaches are possible depending on the type of data. First, the training data is well identified, and a ground truth is available, therefore classical supervised classification techniques can be used. In such a passive supervised learning, the goal of a learner is to infer an accurate predictor from labeled training data. The expert is passive as the system uses only existing labeled training examples. The labeled training data are input-output pairs (x, y): the feature set x describing the example, and its corresponding label y. Second, image retrieval methods can be used when the number of examples is too small, and when there is a large variability across examples [1,16]. Third, the images belong to a new type of classification problem, and an efficient technique has to be provided to facilitate data labeling, i.e., the creation of the ground truth. The creation of a ground truth is in fact an important aspect because providing accurate labels can be a challenging task that requires the full attention of the users. This task can require several users to validate the results. In the case of medical images, the ground truth can only be created by an expert. Therefore, the ground truth estimation, as a major component of a pattern recognition system, can be time consuming and costly.

In active learning, each example of the training database is initially unlabeled. However, the active learner is allowed to request the ground truth, the label y, of any particular example x in the training database [35]. The requests can be made after a

☆ This paper has been recommended for acceptance by Ajay Kumar.

* Tel.: +44 2871675276.
E-mail address: hub20xx@hotmail.com, h.cecotti@ulster.ac.uk

http://dx.doi.org/10.1016/j.patrec.2016.01.016
0167-8655/

non-supervised learning technique (e.g. examples are clustered, and the centroids are then labeled by an expert), or online (sequentially) in order to adapt the classifier to previous label requests. The objective of these methods is to discover the labels of examples while minimizing the number of manually labeled examples [43]. This solution is particularly adapted in large data collections where there exists a strong disparity between the availability of labeled and unlabeled data. In such a case, a challenging task is to provide semi-automatic, high accuracy labeling mechanisms. To some extent, active learning is an easier task than semi-supervised learning (SSL), because in SSL the labeled examples are predefined. The existing labeled examples may be outliers, and/or they may not provide good seeds for label propagation. In active learning, we can distinguish two strategies that require the use of an expert. First, the expert is needed because a potential confusion between two examples is detected, and this ambiguity should be raised by an expert. Second, the expert is needed to label the examples that have the highest potential to be beneficial in the learning procedure. The latter approach is considered in the proposed method: the expert sets the best seeds that can reliably propagate their labels to other unlabeled examples.

In this paper, we propose a new active learning method that combines an efficient distance measure based on Image Distortion Model Distance (IDMD) [20], a greedy SSL approach, and active learning. The efficient distance measure allows us to obtain a robust graph that respects the manifold assumption: if two examples are similar then their corresponding vertices in the graph are connected. The SSL part is dedicated to the label propagation to local neighborhoods. Finally, the active learning step guides the method to the most relevant examples to label. To show the relevance of the method, we use four databases of single handwritten digits. For the Latin script, the performance of single handwritten character recognition is typically sufficiently high when the number of images to train a classifier is high. The supervised learning methods include deep learning architecture such as convolutional neural networks [10], Support Vector Machines (SVM) [12], and their combination [22,29]. However, the accuracy of single handwritten character recognition remains below 100% in some scripts because documents are not properly conserved, and are therefore noisy once they are digitized. Furthermore, a large variability across writers, with several styles and different glyphs for a same digit or character, can become an obstacle. For all these reasons, it is essential to propose new methods to maximize the accuracy while minimizing the number of labeled training examples. Moreover, the recognition of some characters can be impossible without any contextual information and may only be achieved by a person. The remainder of the paper is organized as follows: First, we give an overview of image matching techniques in character recognition and active learning methods in Section 2. Then, we describe the new method in Section 3. In Section 4, we present the four handwritten databases. The active learning strategy is then evaluated in Section 5. Finally, the performance of the proposed approach is discussed in Section 6.

2. Related work

2.1. Image matching

Image matching techniques in large databases are usually not used due to the high processing time that is involved, e.g. the computation of distances between the test image and the prototypes. Elastic matching techniques can be classified into two categories: parametric and non-parametric [39]. It is typically seen as an optimization problem of two-dimensional warping (2DW). This problem is directly related to point matching, which has to deal with the existence of outliers and geometric transformations that may require high dimensional non-rigid mappings [14]. Deformations in handwritten characters can be of two types: first, the global or large deformations such as rotation (with limited angles), scaling, translation; and second, the local deformations that include changes of stroke direction, curvature, and length of the lines. The local deformations that are involved by the thickness of the character depend on the pen/pencil that is used. Due to the different types of deformations that can occur within the same character, it is difficult to determine generic models of deformations. An efficient distance for image classification that takes into account local deformations was proposed by Keysers et al. [20]. They determined that more complex models (*e.g.* 2-dimensional warping) do not necessarily represent better models compared to the simple image distortion model. We define the distance L_p between two images A and B of size $N_s \times N_s$ by:

$$L_p = \left(\sum_{i=1}^{N_s} \sum_{j=1}^{N_s} |A(i,j) - B(i,j)|^p \right)^{1/p} \quad (1)$$

When $p = 2$, it corresponds to the Euclidean distance. The image distortion model distance (IDMD) takes as input two images A and B of size $N_s \times N_s$ that can have w_2 multiple channels. In this study, each channel corresponds to the graylevel image processed with a convolution filter. The distance is then computed through a range of pixels from N_{min} to N_{max}. For each pixel (i_1, j_1) of A, a square image patch, centered at the pixel, is compared to a square patch of same size at the same region in the image B (channel wise, if there are multiple channels). To cope with local variations, the position of the square patches in the image B is allowed to be slightly shifted (within distance w_0). Thus, each patch in A is compared to multiple patches in B around the same pixel location, and the minimum computed distance is taken.

$$\text{IDMD(A,B)} = \sum_{i_1=N_{min}}^{N_{max}} \sum_{j_1=N_{min}}^{N_{max}} d_1(i_1, j_1) \quad (2)$$

where

$$d_1(i_1, j_1) = \min_{(i_2,j_2)\in\{-w_0;w_0\}^2} d_2(i_1, j_1, i_2, j_2) \quad (3)$$

$$d_2(i_1, j_1, i_2, j_2) = \sum_{i_3=-w_1}^{w_1} \sum_{j_3=-w_1}^{w_1} \sum_{i_4=1}^{w_2} |v_1(i_4) - v_2(i_4)|^p \quad (4)$$

where $v_1(i_4)$ and $v_2(i_4)$ are the pixel values at the following coordinates in the i_4th channel of the image ($1 \le i_4 \le w_2$):

$$v_1(i_4) = A_{i_4}(i_1 + i_3, j_1 + j_3) \quad (5)$$

$$v_2(i_4) = B_{i_4}(i_1 + i_2 + i_3, j_1 + j_2 + j_3) \quad (6)$$

For each pixel (i_1, j_1), a displacement field of size w_0 is used in each direction (it corresponds to the elements of a square window of size $2w_0 + 1$), a square window for the consideration of the neighborhood pixels of size $2w_1 + 1$, and the sum of w_2 values, which corresponds to w_2 filtered images. It is worth noting that the *min* function aims at including a relative invariance to local deformations. The image has to be placed in a larger image with a border of the background color to include the possible shifts in the four directions, hence N_{min} and N_{max} are set to $w_0 + w_1$ and $N_{min} + N_s$, to take into account the size of the filters and the displacement fields. Finally, the Euclidean distance (L_2) between I_1 and I_2 is similar to IDMD with the following parameters: $w_0 = 0$, $w_1 = 0$, $w_2 = 1$, $p = 2$.

2.2. Semi-supervised learning and active learning

In semi-supervised learning, several approaches have been proposed [44]. Transductive SVMs optimize margins of both labeled

and unlabeled examples [18,37]. Some other approaches use the cluster assumption. In this case, the classifier takes into account decision boundaries through low-density regions in the input feature space. Most of the techniques in semi-supervised learning are graph-based techniques [2,4,8]. They rely on two main assumptions. The first one is the cluster assumption. It assumes that examples associated to the same cluster, or the same group of clusters, will share the same label. The second hypothesis is the manifold assumption, which considers that examples that are close to each other will have the same label. The label prediction of an example x will depend on both the labeled and unlabeled examples that are very close to x.

Systems based on active learning using graph matching and agglomerative clustering have been developed for mathematical and online handwritten digits recognition [15,24]. Unsupervised learning classifiers and their combinations have been efficiently used for offline character recognition [40,41]. In Vajda et al., a combination of features (raw pixels, profiles, local binary patterns, Radon transform and encoder network) were used with k-means, self-organizing maps, and growing neural gas to increase the reliability of the labeling decision. The active learning procedure was static, as there was no retraining of the classifier after labeling a set of examples, as opposed to dynamic approaches that retrain and update the model after the addition of new examples. Furthermore, an active learning is applied in conjunction with support vector machines (SVMs) to recognize underwater zooplankton [26]. SVMs have been used in a binary classification task where the active learning strategy labels examples closest to the decision boundary [38]. In [28,33], a probability model is used to label examples which could maximize the posterior entropy on the unlabeled data set.

3. System overview

A graph $g = (V, E)$ is defined by the nodes $V = \{1, \ldots, n\}$, which represent all the n examples of a training database $X = \{x_1, \ldots, x_n\}$, and edges E, which represent the similarities between examples. The similarities are typically represented by a weight matrix $W \in \mathbb{R}_+^{n \times n}$. A cell $W(i, j)$ corresponds to the similarity between the example x_i and x_j, *i.e.* the edge (i, j) in E. If x_i and x_j are close to each other (they belong to the same neighborhood), then $W(i, j)$ has a non-zero value. $W(i, j)$ is defined by the IDMD between the images x_i and x_j in the training database. For creating the graph, the following parameters were used for IDMD: $w_0 = 2$, $w_1 = 1$, and $p = 2$. We pre-process images with the Sobel operator in two directions (vertical and horizontal, $w_2 = 2$). IDMD has a high computational cost when dealing with large databases in algorithms such as k-nearest neighbor (k-nn) [11], i.e. when the number of prototypes is large. To reduce the computational cost to process a database, a strategy consists in limiting the number of prototypes by using a less computational expensive distance. First, we obtain the 500 closest examples with the distance L_2, which will be used as the prototypes that will be used in the next step. Then, the distances between images are obtained with IDMD, where the prototypes of IDMD are the 500 best answers obtained with distance L_2 at the previous stage. The creation of the graph is equivalent to the selection of the k-nearest neighbor of each example in the training database.

For each example, the distances to the 500 neighbors are sorted by ascending order. We select a subset of k neighbors from the 500 neighbors. Hence, the graph is represented by two matrices $W_k \in \mathbb{R}_+^{n \times k}$ and $E_k \in \mathbb{N}^{n \times k}$. W_k contains the sorted weights of the k neighbor examples for each example x_i. E_k contains for each example x_i, the associated list of the vertices corresponding to the closest neighbors. We obtain $W_k(i, 1) = 0$ and $E_k(i, 1) = i$, because each point is its closest neighbor in the graph. This representation allows us to store the data in $\mathcal{O}(kn)$, which is significantly smaller than a storage in $\mathcal{O}(n^2)$. We define the associated ground truth Y, where $y_i = m$ if x_i belongs to the class $m \in \{1 \ldots M\}$, and $y_i = -1$ if x_i is unlabeled (it is a default value); X^l and X^u represent the set of labeled, and unlabeled examples, respectively.

Algorithm 1 GetLabel.

```
M ∈ ℕ^n                                   # 0 values
for i = 1 to n do
  if (X_i, Y_i) ∈ X^u then
    for j = 2 to s + 1 do
      M(E_k(i, j)) = M(E_k(i, j)) + 1
M_sort ← sort(M)                          # sort descend
return arg(M_sort(1))                     # example to label
```

Algorithm 2 Semi-supervised learning with active learning.

```
X^u ← (X, Y) , X^l ← ∅, X^m ← ∅
while |X^m| < L_max and X^u ≠ ∅ do
  (x, y) ← GetLabel                       # Active learning
  X^l ← X^l + (x, y) , X^u ← X^u − (x, y) , X^m ← X^m + (x, y)
  repeat
    for all x_i ∈ X^u do
      a ← E_k(i, 2)                       # first neighbor
      b ← E_k(i, 3)                       # second neighbor
      if x_a ∈ X^l then                   # y_a ≠ −1
        y_i ← y_a                         # AL_1 condition
        X^l ← X^l + (x_i, y_i) , X^u ← X^u − (x_i, y_i)
      else if x_b ∈ X^l then              # y_b ≠ −1
        y_i ← y_b                         # AL_1 and AL_2 condition
        X^l ← X^l + (x_i, y_i) , X^u ← X^u − (x_i, y_i)
  until convergence
return X^l                                # labeled examples
```

The function GetLabel (Algorithm 1) is defined to return the most relevant example that should be labeled. It corresponds to the active learning step where it is possible to directly access to the ground truth of an example. First, the vertices of the graph are sorted based on the number of neighbors that they share, by considering a neighborhood of size s, $s \leq k$. The goal of this function is to find the examples that are close to as many other examples as possible that could lead to wide and safe labeled examples through a greedy label propagation approach. If an example corresponds to a noisy image, it is assumed that not many examples will share this example in their close neighborhoods. By labeling such an example, a label propagation procedure has a higher chance to label examples with a wrong label. Furthermore, if an example does not belong to any neighborhood, then it cannot be propagated, and it has no impact on the label of other examples.

By selecting the labeled example x_l with GetLabel, and querying its label y_l that may lead to a safe label propagation, we consider two conditions: AL_1 and AL_2. For AL_1, each unlabeled example that has as a first neighbor x_l will be assigned the label y_l. For AL_2, each unlabeled example that has as a first or second neighbor x_l will be assigned the label y_l. The labels are then propagated in a greedy manner to their neighborhood until convergence (i.e. there are no more examples that can be automatically labeled). The procedure is repeated until a maximum number of manually labeled examples (X^m) is reached (L_{max}), or if there are no more examples to label in the database. The method is described in pseudo-code in Algorithm 2. In the result section, we limit the maximum number of manually labeled examples per class to 100, $k = 10$, and $s = 2$. In Section 5, we assess the performance of the method in two ways. First, we determine to what extent the active learning approach allows to correctly label the training database. Second, we evaluate

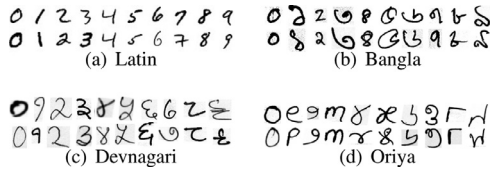

Fig. 1. Representative handwritten digits for the different databases (from zero to nine).

Table 1
Properties of the handwritten digit databases.

Database	MNIST	Bangla	Devnagari	Oriya
	Training			
# samples	60000	19392	18783	4970
# per class	6000 ± 339	360	1878 ± 15	497 ± 3
Size (x)	28	58 ± 16	65 ± 16	73 ± 25
Size (y)	28	54 ± 16	62 ± 19	73 ± 26
	Test			
# samples	1000	4000	3763	1000
# per class	1000 ± 62	400	376 ± 3	100
Size (x)	28	59 ± 17	66 ± 17	75 ± 25
Size (y)	28	54 ± 18	62 ± 20	74 ± 26

the performance drop involved by the wrongly labeled examples in the training database on the test database.

4. Databases

Four databases of handwritten digits (10 classes) have been chosen for the performance analysis. The databases contain images of digits in four scripts: Latin, Bangla, Devnagari, and Oriya. Samples of digits are presented in Fig. 1. All the images were normalized with the same procedure. Because some databases have very noisy images, images were first binarized with the Otsu method at their original size [30], then they were size normalized to fit in a 20 × 20 pixel box while preserving their aspect ratio. The resulting images contain 8 bit gray levels due to the bicubic interpolation for resizing the images. All the images were centered in a 28 × 28 pixel box field by computing the center of mass of the pixels. Finally, the gravity center of the image was translated to the center of the 28 × 28 field. Table 1 presents for each database the number of classes, the total number of images in the database, and the number of images per class, for both training and the test.

The first database is MNIST. It contains Latin (Arabic) digits [23], with a training and test database of 60000 and 10000 images, respectively. The error rate reaches quasi human performance level of 0.23% with a combination of 35 convolutional neural networks [10]. With k-nn, the best error rate is 0.52% by using a Pseudo 2D Hidden Markov Models [20], followed by Image Deformation Model with an error rate of 0.54%, and 0.63% with shape matching using shape contexts [3], without the addition of artificial examples in the training database. India is a multilingual country, with twenty-two official languages and twelve scripts. In Indian language scripts, the concept of upper case and lower-case characters is not present. The databases of Indian digits were created at the Indian Statistical Institute, Kolkata, India [5,9,31]. The second database contains Bangla digits, which is the fourth most popular script in the world, used by more than 200 million people [42]. The third database has Devnagari digits, which is part of the Brahmic family of scripts of India, Nepal, Tibet, and South-East Asia [34]. The fourth database contains Oriya (Utkala Lipi) digits [7]. An accuracy of 90.50% was obtained by using Hidden Markov Models [7]. A two-stage framework that combines modified quadratic discriminant function (MQDF) [21] and MLPs was used for the recognition of Bangla characters [6]. Other databases of the Bangla characters (50 classes), Mandal et al. use features based on the combination of gradient features and Haar wavelet coefficients at different scales with a k-nn classifier to reach an accuracy of 88.95% [27].

5. Results

The classifier accuracy for the four training databases is presented in Table 2. The impact of the method on the accuracy with the test database is given in Table 3 by using a k-nn classifier with the same distance that was used to build the graph (i.e. IDMD with prototypes obtained with distance L_2). The table shows to what extent the wrongly labeled images in the training database have an impact on the classification performance in the test. By labeling only 0.55% of the training database of MNIST (332 images), it is possible to automatically label the examples with an accuracy of 98.54%, i.e. only 332 images can be labeled to obtain a ground truth of 60000 images, which is reliable at 98.54%. This estimated ground truth leads to an accuracy of 99.10% on the test database. It corresponds to a performance drop of 0.22% compared to the approach that uses the correct ground truth of the whole training database. As expected, more examples are labeled with the method AL_1 than AL_2, as AL_1 is more restrictive. An unlabeled example can obtain a label only from its direct labeled first neighbor with AL_1 while it can obtain a label if its first or second neighbor is labeled with AL_2. This increase of manually labeled examples provides a better performance (98.79% and 99.23% for training and the test, respectively). The same pattern of performance is observed for the three Indian scripts. It is possible to achieve an accuracy of 96.73, 98.65, and 96.00% with AL_2 for Bangla, Devnagari, and Oriya, respectively. In the three Indian databases, the number of required

Table 2
Accuracy (in %) on the four training databases with different number of labeled examples on the training database.

	MNIST		Bangla		Devnagari		Oriya	
Method	AL_1	AL_2	AL_1	AL_2	AL_1	AL_2	AL_1	AL_2
# label	1170	332	1281	344	1954	401	1686	388
# label/class	117 ± 12	33 ± 13	128 ± 12	34 ± 15	195 ± 61	40 ± 28	169 ± 40	39 ± 24
0	99.65	99.71	99.95	100.0	99.95	99.95	100.0	100.0
1	98.43	97.30	98.82	96.40	99.95	99.95	100.0	98.39
2	99.04	99.45	99.02	97.74	97.94	89.05	100.0	100.0
3	99.15	99.46	98.82	78.58	99.79	99.73	100.0	99.20
4	98.82	97.43	99.74	99.85	99.95	99.79	100.0	99.60
5	98.65	98.78	99.33	98.60	99.84	99.74	100.0	99.59
6	99.22	99.27	96.89	97.72	99.89	99.52	99.80	99.40
7	97.86	96.92	99.64	99.90	98.98	99.73	100.0	88.76
8	98.51	98.44	99.59	99.64	99.47	99.58	100.0	98.39
9	98.66	98.79	95.12	94.45	98.94	99.47	97.76	96.34
all	98.79	98.54	98.69	96.26	99.46	98.64	99.76	97.97

Table 3
Accuracy (in %) on the four test databases with k-nearest neighbor and IDMD.

	MNIST			Bangla			Devnagari			Oriya		
Method	all	AL_1	AL_2	all	AL_1	AL_2	all	AL_1	AL_2	all	AL_1	AL_2
Training (in %)	100	1.95	0.55	100	6.61	1.77	100	10.40	2.13	100	33.92	7.81
$k = 1$	99.27	98.86	98.81	98.55	97.45	95.25	99.12	98.67	98.06	98.00	98.00	96.90
$k = 3$	99.36	99.24	99.05	98.28	97.68	96.20	99.20	98.94	98.46	97.90	97.90	96.60
$k = 5$	99.32	99.23	99.10	98.23	97.80	96.73	99.20	99.04	98.65	97.50	97.50	96.70
$k = 10$	99.19	99.10	99.08	97.23	97.48	96.73	98.99	98.88	98.65	96.60	96.60	96.00

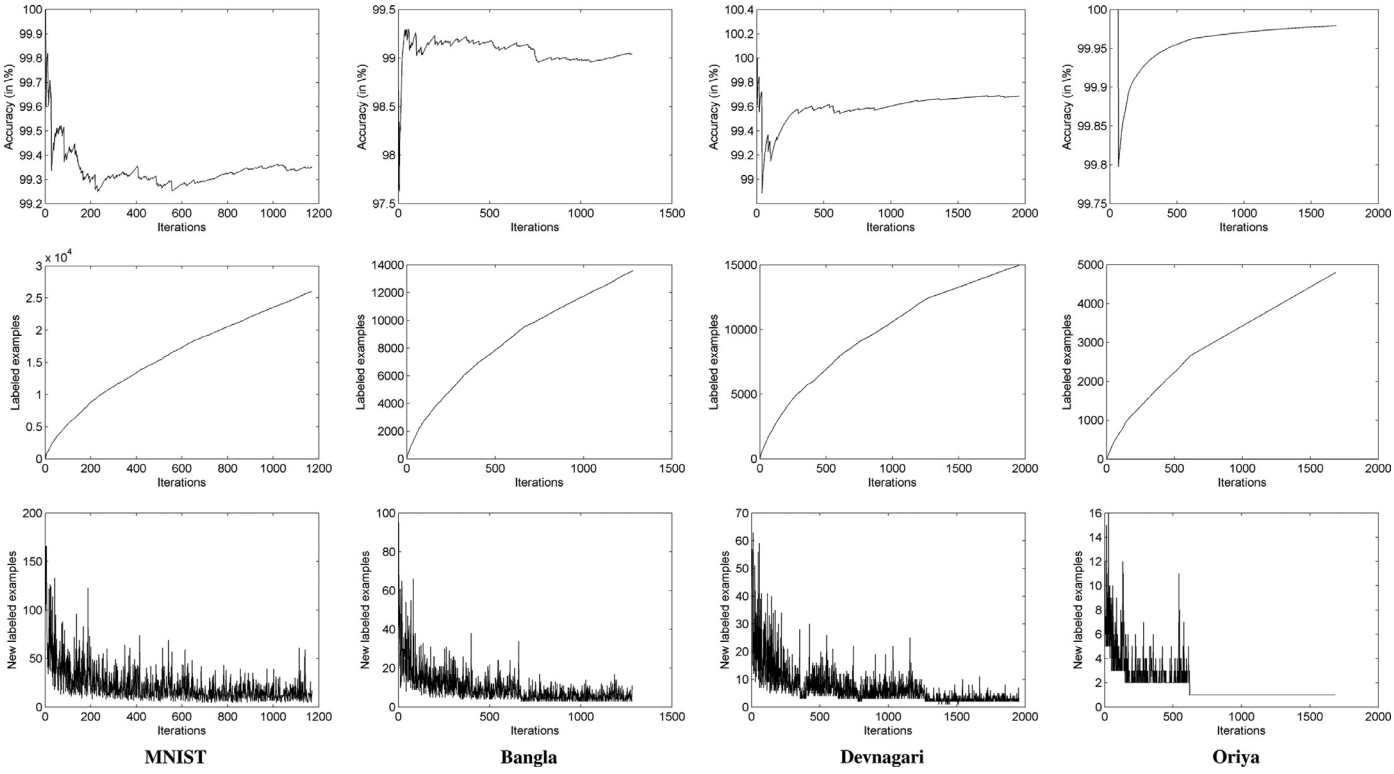

Fig. 2. AL_1: **First row:** Accuracy (in %) in relation to the number of iterations (manually labeled images). **Second row:** Total number of labeled examples in relation to the number of iterations. **Third row:** Number of new labeled examples at each iteration (each time a manually labeled image is added in the database).

labeled examples is significantly reduced: 1.77, 2.13, and 7.81% for Bangla, Devnagari, and Oriya, respectively. It is worth noting that the number of manually labeled examples is about 350 for the four databases, suggesting that only 35 example-prototypes should be used per class. The remaining examples can be labeled with a transductive approach through label propagation. To evaluate the impact of the active learning procedure with examples selected on their number of neighbors, we compare the condition AL_2 with the case where a random unlabeled example is selected to be labeled by the expert. Monte Carlo simulations are used to estimate the performance of this condition (50 repetitions). We observe a drop of performance in the accuracy of the labeling procedure on the training database (97.54 ± 2.23, 96.03 ± 1.03, 96.86 ± 0.97, and 98.22 ± 0.14 for MNIST, Bangla, Devnagari, and Oriya). This drop in the accuracy is also accompanied by an increase of the number of examples to label manually (682 ± 58, 668 ± 36, 819 ± 29, and 577 ± 19 for MNIST, Bangla, Devnagari, and Oriya). These results suggest that the selection of the examples to label during active learning plays a crucial role.

The classification accuracy with the current number of labeled images is depicted in the first row of Figs. 2 (AL_1) and 3 (AL_2). These figures highlight the robustness of the method with the addition of new examples. In each database, the accuracy stays steady with the addition of new manually labeled examples and their propagated labels. The total number of examples that are labeled is presented in the second row of Figs. 2 and 3 as a function of the number of manually labeled images. The evolution of the number of examples that are automatically labeled is not linear over time (i.e. across iterations) for AL_2, contrary to AL_1. For MNIST, the method reaches a plateau after 50 manually labeled images by using AL_2. The same phenomenon is observed for Bangla and Devnagari where there exists a plateau after 100 iterations. These results indicate the presence of clusters with few examples, which cannot be reached through label propagation with a graph based on small neighborhoods. In the third row of Figs. 2 and 3, the number of new images that are labeled thanks to the addition of a new manually labeled example is given for each iteration, confirming the previous observations: Most of the propagated labels are set during the first iterations, where examples belong to large clusters. With Oriya digits, we observe several high peaks after 100 iterations, suggesting the presence of isolated clusters of images in the database.

For the computation of the distances between examples to obtain the graph and for the k-nn implementation for the test, a parallel implementation has been used for IDMD (high performance cluster), and for the distance L_2 (GPUs). The system was implemented with Matlab R2013a, using three GPU cards (NVIDIA Tesla C1060), it takes about 450, 85, 83, and 10 s to obtain the distances

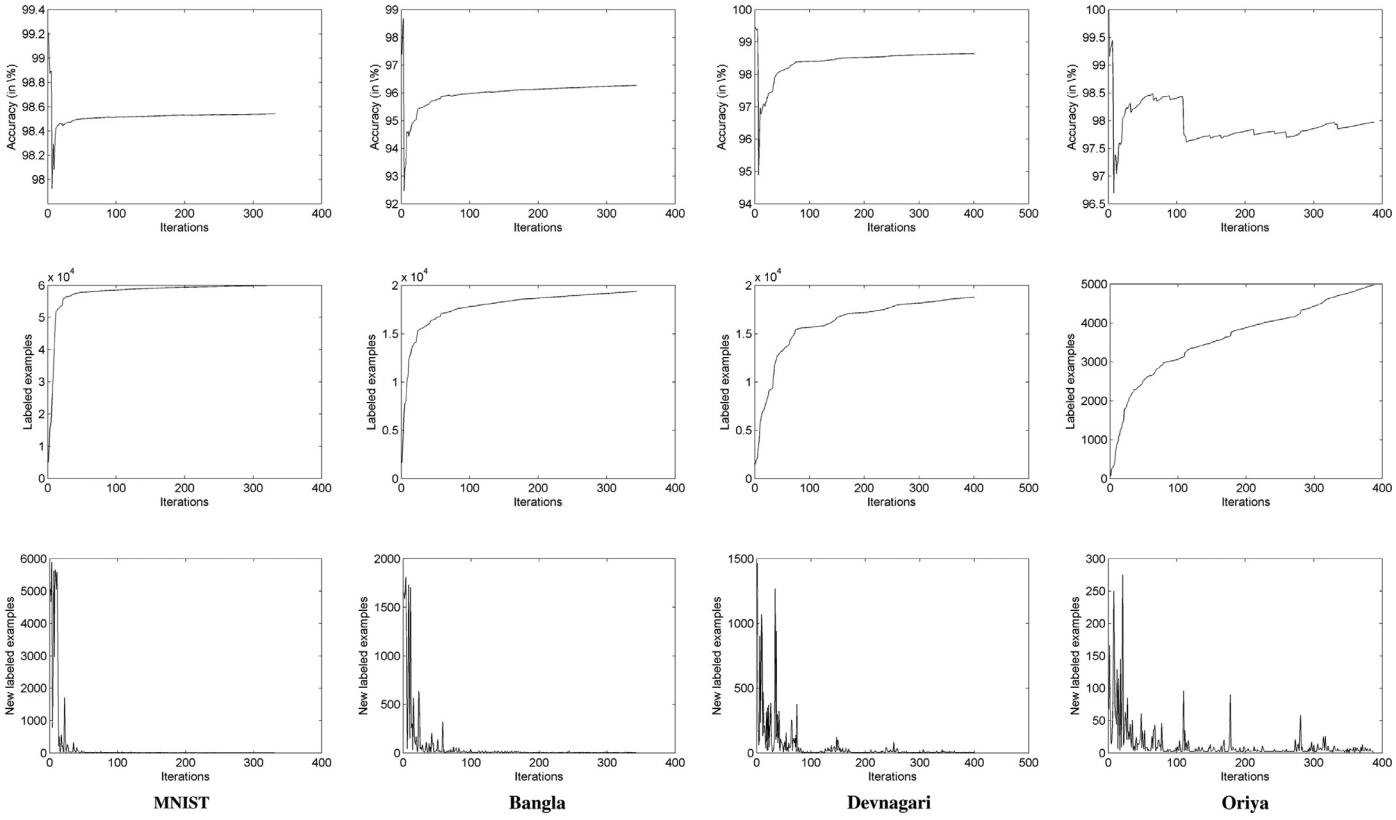

Fig. 3. AL_2: **First row:** Accuracy (in %) in relation to the number of iterations (manually labeled images). **Second row:** Total number of labeled examples in relation to the number of iterations. **Third row:** Number of new labeled examples at each iteration (each time a manually labeled image is added in the database).

with the Euclidean distance and $k = 500$, for processing the MNIST, Bangla, Devnagari, and Oriya training databases. With a parallel implementation on a cluster using 50 cores (Intel Xeon X5650 2.66 Ghz), it takes about 4200, 750, 1350, and 345 s to obtain the complete graph for the MNIST, Bangla, Devnagari, and Oriya training databases with IDMD. Finally, the processing time for the active learning part is about 7, 3, 3, and 1 s for the MNIST, Bangla, Devnagari, and Oriya, showing the relevance of the method once it has to dynamically query the user for the best examples to label.

6. Discussion

With the continuous development of new problems requiring pattern recognition systems, a fundamental challenge is the optimization of the labeling procedure for creating ground truths. This step is required for two reasons: first, to have a training database for classifiers based on supervised learning; and second, for the creation of benchmarks. While a system can be created with a non-supervised technique, the evaluation test will require in any case a ground truth to quantify the performance of the method. User friendly graphical user interface and semi-automatic procedure should help the creation of ground truths. With a semi-automatic approach in a multiclass problem, the expert may only confirm the accuracy of the automatic process. In such a case, the expert only provides a binary response (confirmation or not of the automatic process) [19].

The main goal of the method was to propose an efficient and comprehensive active learning strategy that can take advantage of a robust distance between single handwritten characters, and label propagation in graph-based semi-supervised learning, by including an active learning approach. Because the labeling procedure is semi-automatic, some errors may happen depending upon the quality of the graph, i.e. if the manifold and cluster assumptions are always verified, and how the vertices of the graph are connected. A major issue in SSL techniques is the robustness to the noise, i.e. outliers that are close to the decision boundaries. These outliers can be used as bridges to propagate labels from a cluster of examples belonging to a single class to another cluster corresponding to a different class. Contrary to fixed SSL approaches, where the labeled examples are determined a priori, the active learning paradigm selects interactively the best examples to label. Furthermore, the proposed method outperforms recent strategies for semi-automatic labeling through active learning. In [41], an active learning technique combining clustering techniques and a voting system provided an accuracy of 96.77% on MNIST with 750 labeled examples. With the proposed method, a significant improvement was obtained as it is possible to reach an accuracy of 99.10% with only 332 manually labeled images, which is relatively close to state-of-the art methods (e.g. 99.47% with a large convolutional neural network with unsupervised pre-training, and no artificial images in the training database [17]). Finally, the proposed approach dynamically selects the number of examples to label in relation to the difficulty of the task (i.e. the size and connectivity of the clusters).

With IDMD, only three main parameters have to be chosen: the size of the possible shifts (w_0), the size of the neighborhood of each pixel (w_1), and the number of pre-processed images (w_2). Since IDMD is computationally expensive, the recognition system must take advantage of parallel computing and computer clusters. k-nn can be easily transferred into a parallel implementation with a shared-nothing cluster on a number of commodity machines using MapReduce [25]. IDMD allows the invariance to local deformation, but it is still sensitive to large transformations, such as dilatation, and large rotations. While efficient methods of character

recognition exploit some knowledge of the domain by including deformed images [36], this strategy may be difficult to follow when the language of the script is unknown from the designer of the pattern recognition system. If prior information of the script is available, deformed patterns could be added in the databases to increase the size of the training database. The parameters of the deformations must be properly chosen based on some prior knowledge about the script. For Indian scripts, this parameter is difficult to determine because the variation across writers is important. This approach could provide more examples behaving as bridges across examples belonging to different classes, if the deformations are not carefully chosen. Finally, the databases of Indian digits are very noisy, and may require better denoising techniques.

7. Conclusion

A new active learning method using a graph based on the Image Deformation Model Distance and a greedy semi-supervised learning has been presented for handwritten character recognition. We have shown that is it possible to reliably propagate labels with a greedy procedure by using small neighborhoods. Moreover, the active learning addition to the semi-supervised part has a significant impact, as it sets a label to the most useful examples. The efficiency of the method was demonstrated by reaching state-of-the art results across four databases of different scripts. The active learning procedure should be performed online, hence further work should include an adaptive graphical user interface that allows the user to rapidly label the examples suggested by the system. An appropriate graphical representation of the graph depicting all the examples could provide a faster way to combine active learning and semi-supervised learning.

Acknowledgment

The author would like to thank Prof. Ujjwal Bhattacharya for sharing the databases of Bangla, Devnagari, and Oriya.

References

[1] X. Bai, X. Yang, L.J. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, IEEE Trans. Pattern Anal. Mach. Intell. 32 (5) (2010) 861–874.
[2] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from examples, JMLR 7 (2006) 2399–2434.
[3] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.
[4] Y. Bengio, O. Dellalleau, N.L. Roux, Label propagation and quadratic criterion, in: O.C. B. Schölkopf, A. Zien (Eds.), Semi-Supervised Learning, MIT Press, 2006, pp. 35–58.
[5] U. Bhattacharya, B.B. Chaudhuri, Databases for research on recognition of handwritten characters of indian scripts, in: Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR'05), 2005, pp. 789–793.
[6] U. Bhattacharya, M. Shridhar, S.K. Parui, P.K. Sen, B.B. Chaudhuri, Offline recognition of handwritten Bangla characters: an efficient two-stage approach, Pattern Anal. Appl. 15 (4) (2012) 445–458.
[7] T.K. Bhowmick, S.K. Parui, U. Bhattacharya, B. Shaw, An HMM based recognition scheme for handwritten oriya numerals, in: Proceedings of the 9th International Conference on Information Technology (ICIT 2006), 2006, pp. 105–110.
[8] A. Blum, S. Chawla, Learning from labeled and unlabeled data using graph mincuts, in: Proceedings of the International Conference on Machine Learning, 2001, pp. 19–26.
[9] B.B. Chaudhuri, U. Pal, A complete printed Bangla OCR system, Pattern Recognit. 31 (1998) 531–549.
[10] D. Cireşan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3642–3649.
[11] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[12] D. DeCoste, B. Schölkopf, Training invariant support vector machines, Mach. Learn. 46 (1–3) (2002) 161–190.
[13] D. Ghosh, T. Dube, A. Shivaprasad, Script recognition: A review, IEEE Trans. Pattern Anal. Mach. Intell. 32 (12) (2010) 2142–2161.
[14] S. Gold, A. Rangarajan, C.P. Lu, S. Pappu, E. Mjolsness, New algorithms for 2-d and 3-d point matching: pose estimation and correspondence, Pattern Recognit. 31 (8) (1998) 1019–1031.
[15] N.S.T. Hirata, W.Y. Honda, Automatic labeling of handwritten mathematical symbols via expression matching, in: Proceedings of the 8th International Conference on Graph-based Representations in Pattern Recognition, 2011, pp. 295–304.
[16] R.-X. Hu, W. Jia, H. Ling, Y. Zhao, J. Gui, Angular pattern and binary angular pattern for shape retrieval, IEEE Trans. Image Process. 23 (3) (2014) 1118–1127.
[17] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, What is the best multi-stage architecture for object recognition? in: Proceedings of the 12th International Conference on Computer Vision (ICCV'09), 2009, pp. 2146–2153.
[18] T. Joachims, Transductive inference for text classification using support vector machines, in: Proceedings the International Conference on Machine Learning, 1999, pp. 200–209.
[19] A.J. Joshi, F. Porikli, N.P. Papanikolopoulos, Scalable active learning for multiclass image classification, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2259–2273.
[20] D. Keysers, T. Deselaers, C. Gollan, H. Ney, Deformation models for image recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (8) (2007) 1422–1435.
[21] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 9 (1987) 149–153.
[22] F. Lauer, C.Y. Suen, G. Bloch, A trainable feature extractor for handwritten digit recognition, Pattern Recognit. 40 (6) (2007) 1816–1824.
[23] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[24] J. Li, H. Mouchère, C. Viard-Gaudin, An annotation assistance system using an unsupervised codebook composed of handwritten graphical multi-stroke symbols, Pattern Recognit. Lett. 35 (1) (2014) 46–57.
[25] W. Lu, Y. Shen, S. Chen, B.C. Ooi, Efficient processing of k nearest neighbor joins using MapReduce, in: Proceedings of the VLDB Endowment VLDB Endowment Hompage archive, vol. 5, 2012, pp. 1016–1027.
[26] T. Luo, K. Kramer, D.B. Goldgof, L.O. Hall, S. Samson, A. Remsen, T. Hopkins, Active learning to recognize multiple types of plankton, J. Mach. Learn. Res. 6 (2005) 589–613.
[27] S. Mandal, S. Sur, A. Dan, P. Bhowmick, Handwritten Bangla character recognition in machine-printed forms using gradient information and Haar wavelet, in: Proceedings of the 2011 International Conference on Image Information Processing (ICIIP), 2011, pp. 1–6.
[28] K. Nigam, A. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Mach. Learn. 39 (2000) 103–134.
[29] X.-X. Niu, C.Y. Suen, A novel hybrid CNNSVM classifier for recognizing handwritten digits, Pattern Recognit. 45 (6) (2012) 1318–1325.
[30] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Sys. Man. Cyber. 9 (1) (1979) 62–66.
[31] U. Pal, B.B. Chaudhuri, Indian script character recognition: a survey, Pattern Recognit. 37 (9) (2004) 1887–1899.
[32] D. Picard, P.-H. Gosselin, M.-C. Gaspard, Challenges in content-based image indexing of cultural heritage collections, IEEE Signal Process. Mag. 32 (4) (2015) 95–102.
[33] N. Roy, A. McCallum, Toward optimal active learning through sampling estimation of error reduction, in: Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 441–448.
[34] I.K. Sethi, B. Chatterjee, Machine recognition of constrained hand printed devanagari, Pattern Recognit. 9 (1977) 69–75.
[35] B. Settles, Active Learning Literature Survey, Technical Report 1648, University of Wisconsin–Madison, 2009.
[36] P. Simard, D. Steinkraus, J.C. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR), 2003, pp. 958–962.
[37] V. Sindhwani, P. Niyogi, M. Belkin, Beyond the point cloud: from transductive to semi-supervised learning, in: Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 824–831.
[38] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, in: Proceedings of the 17th International Conference on Machine Learning, 2000, pp. 999–1006.
[39] S. Uchida, H. Sakoe, Survey of elastic matching techniques for handwritten character recognition, IEICE Trans. Inf. Syst. 88 (8) (2005) 1781–1790.
[40] S. Vajda, A. Junaidi, G.A. Fink, A semi-supervised ensemble learning approach for character labeling with minimal human effort, in: Proceedings of the International Conference on Document Analysis and Recognition, 2011, pp. 259–263.
[41] S. Vajda, Y. Rangoni, H. Cecotti, Semi-automatic ground truth generation using unsupervised clustering and limited manual labeling: application to handwritten character recognition, Pattern Recognit. Lett. 58 (2015) 23–28.
[42] S. Vajda, K. Roy, U. Pal, B.B. Chaudhuri, A. Belaïd, Automation OF Indian postal documents written in Bangla and English, Int. J. Pattern Recognit. Artif. Intell. 23 (8) (2009) 1599–1632.
[43] B. Zhang, Y. Wang, F. Chen, Multilabel image classification via high-order label correlation driven active learning, IEEE Trans. Image Process. 23 (3) (2014) 1430–1441.
[44] X. Zhu, Semi-Supervised Learning Literature Survey, Technical Report Computer Sciences Technical Report 1530, University of Wisconsin-Madison, 2005.