Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 3 (2011) 336–342

Procedia Computer Science

www.elsevier.com/locate/procedia

WCIT-2010

Semi-supervised Persian font recognition

Maryam Bahojb Imani [a] *, Mohamad Reza Keyvanpour [a], Reza Azmi [a]

[a]*Alzahra University, Tehran, Iran*

Abstract

Font recognition is one of the fundamental tasks in document recognition, because it is an important factor in optical character recognition. Classical supervised methods need lot of labeled data to train a classifier. Since it is very costly and time consuming to label large amounts of data, it is useful to use data sets without labels. So many different semi-supervised learning methods have been studied recently. Among the semi-supervised methods, self-training is one of the important learning algorithms that classify the unlabeled samples with small amount of labeled ones and add the most confident samples to the training set. In this paper, we apply majority vote approach to classify the unlabeled data to reliable and unreliable classes. Then, we add the reliable data to training set and classify the remaining data including unreliable data in iterative process. We test this method on the extracted features of ten common Persian fonts. Experimental result indicates that proposed method improves the classification performance and it's effective.

Selection and/or peer-review under responsibility of the Guest Editor.

Keywords: Font Recognition; Persian; Semi-Supervised Learning; Self Training; Majority Vote.

1. Introduction

Font recognition is one of the fundamental tasks in document analysis and recognition, and it is a difficult and time consuming task. Font recognition can impact on the automatic document analysis in at least two ways. Firstly, font recognition is an important factor in optical character recognition (OCR) and handwriting and it is obvious that recognition of multi-fonts documents is much more difficult than those that have single-font, because characters take different shapes in different fonts. Consequently, font classification can help OCR to reduce the different number of shapes in each class. So it converts the multi-fonts character recognition problem to single-font character recognition problem that has less complexity. Secondly, ideal output for automatic document processing systems not only includes the content of each input document, but also includes an input document font to help the typesetting to be automatically performed. Font recognition system has benefits for both end-user and system developer. End-user benefits from seeing a regenerated document very similar to the original image, and developer can rely on the results of font recognition system to train a special engine for each font and improve the OCR accuracy [1] [2]. So far, so many methods have been applied in OCR, but few of them, especially in Persian language, have considered font recognition. Furthermore, these Persian font recognitions have done by supervised learning methods. These standard supervised learning methods use only labeled data to train and learn classifiers. Due to the diversity of data in the

* Maryam Bahojb Imani. Tel.: +98-912-579-1741.
E-mail address: imani@student.alzahra.ac.ir

1877-0509
doi:10.1016/j.procs.2010.12.057

mentioned filed, labeled instances are often more difficult, expensive and time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile unlabeled data may be relatively easy to collect in practice.

Recently, semi-supervised learning has been proposed to make use of unlabeled and labeled data by assuming that data with similar attributes lead to similar labels. This learning framework deals with situations where labeled data is only a few while unlabeled data is given in a large quantity. In font recognition problem, this situation is provided inherently, because there are a lot of text image that used different types of font and label assigning are difficult and time consuming.

The main algorithms of semi-supervised learning include generative models, Self-training, Co-training, S^3VM, various graph-based methods and so on. Self-training is a commonly used method for semi-supervised learning. In self-training a classifier at first trained with the small amount of labeled data. Then this classifier is used to classify the unlabeled data. Typically the most confident unlabeled points with their predicted labels, are added to the training set, but in this paper, after that a classification of unlabeled samples were partitioned into two parts, i.e. reliable and unreliable in the iterative process; then the reliable group of data are added to the training dataset. Here we utilize the concept of aggregating technique that achieves strong classifier by using multiple learners.

The rest of this paper is organized as follows. In Section 2, we described a brief overview of related researches in Persian font recognition and semi-supervised methods. Section 3 provides explanation of semi-supervised algorithm which is used. After a brief description of our dataset in Section 4, section 5 demonstrates experimental results on Persian font dataset. Finally, we conclude our study.

2. Related researches

In following sub-section, we'll talk about related researches in font recognition, especially in Persian language. After that, the semi-supervised learning with its approaches will be presented.

2.1. Font recognition

Despite of the obvious importance of automatic font recognition, inappreciable font recognition researches in English, Spanish, Korean and Japanese have been done [1] [3] [4] [5]. As well as, in Persian font recognition fewer investigations have been done.

In [6], 24 Gabor filters in four scales and six directions were used for classification of 20 common Persian font types. In [7], 10 font types of 7 different sizes and 4 different types were used and feature extraction technique was based on the first and second order moment. In this method, the k-nearest neighbor was used as classifier. In [8], global typographical features based on Gabor filters are used to extract the features. In classification, the advantage of two classifiers with weighted Euclidean distance and support vector machines are taken. In another recent research, features obtained from Sobel and Roberts matrix. This method attains better result than the two methods mentioned in Persian fonts recognition [2]. All of the mentioned methods in Persian font recognition were done by supervised learning methods, therefore this the first paper in this field that uses unlabeled instances beside label instances.

2.2. Semi-supervised learning

Semi-supervised learning is somewhere between supervised and unsupervised learning, so classifier learns from both labeled $|D^L|$ and unlabeled $|D^U|$ data. There is usually one assumption that there are much more unlabeled samples available than labeled ones, i.e. $|D^L| << |D^U|$. Semi-supervised learning (SSL) is applied for classification, clustering and regression. In this paper, we used it in classification tasks. It's most popular methods are defined as follows [9]:

In *Generative method*, by looking only at unlabeled data, marginal data distribution P(x) can be estimated. It assumes a model $p(x,y) = p(y)p(x$ where $p(x|$ is an identifiable mixture distribution, for example Gaussian mixture models. In *self-training* a classifier is first trained with the small amount of labeled data. The classifier is then used to classify the unlabeled data. Usually the most confident unlabeled points, together with their predicted labels, are added to the training set and the procedure repeated. These methods are wrapper methods, i.e. they allow supervised learning methods to be applied to a semi-supervised learning task. *Co-training* assumes that the features

can be split into two conditionally independent subsets. Each subset is sufficient to train a good classifier, and then uses the predictions of each classifier on unlabeled examples to argue the training set of the other. *TSVM* is an extension of standard support vector machines with unlabeled data. The goal of TSVM is to find a labeling of the unlabeled data. Intuitively, unlabeled data guides the linear boundary away from dense regions, so that the decision boundary has the smallest generalization error bound on unlabeled data. Since TSVM requires solving a combinatorial optimization problem, they are susceptible to local optima, and therefore are sensitive to the initialization of solutions. *Graph-based* method defines a graph where the vertices are labeled and unlabeled examples in the dataset, and edges (may be weighted) reflect the similarity (or distance) of examples. These methods usually assume label smoothness over the graph.

3. Proposed semi-supervised algorithm

Here we use majority vote method as a wrapper of the self training semi-supervised approach. In this algorithm, three machine learning algorithms are applied, such as support vector machine (SVM), radial bias function neural network (RBFNN), and K-nearest nabour (KNN). The brief introduction of these algorithms is in next subsections, but we first present the semi-supervised algorithm which is used in our experiment.

In this paper, majority vote approach is applied to classify the unlabeled data to reliable and unreliable parts. It means that at first unlabeled data are classified by three well-known classifiers with labeled data. The classifiers which mentioned before are used in this algorithm. If the majority of these classifiers predict the same label for each unlabeled data, this label will be assigned to them and it has a reliable label. But, if the classifiers don't agree on a label (which means that each classifier predicts a different label), then we don't assign any reliable label to them and they are still unlabeled and unreliable. Then we retrain this algorithm with new labeled data which consists of labeled data from the beginning and reliable data, and in this step our aim is predicting the labels of unreliable data. This iterative process continues until all of the unlabeled data have label or the condition of limited loop is met. Therefore, this algorithm can predict the labels of large amount of data with the strength of the three other classifiers, as well as reducing the reinforcement error of self-training method.

Finally, labeled data from the beginning and reliable data are used as training dataset and then test data will be classified by SVM and RBF.

3.1. Support Vector Machine

Support vector machines are binary classifiers that separate two classes with a hyperplane boundary. In this method samples that constitute the classes' boundary are called support vectors. This means that training instances that lie on the two parallel hyperplanes are called support vectors (Fig 1). Assume that the data are constructed from two classes' model and these classes have $x_i = 1, 2, ..., L$ training point that x_i is a vector. These two classes are labelled with $y_i = \pm 1$. This separating hyperplane can be described by $w.x + b = 0$, where x are points on the decision boundary and w is a n dimensional vector which is perpendicular to the decision boundary. $b / \| w \|$ is perpendicular distance from decision boundary to the origin and $w.x$ represents the inner product of two vectors w and x. All the training data should satisfy the following constraints [10] [11] :

$$\begin{aligned} & y_i (x_i.w + b) - 1 + \xi_i \geq 0 \\ & \xi_i \geq 0, \forall i \end{aligned} \qquad (1)$$

ξ_i, i=1,2,..., L is introduced to handle data that is not fully linearly separable. Referring to Fig. 1, subject to the constraints in (1), maximizing the SVM's margin is equivalent to finding

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{l} \xi_i \\ & s.t. \quad y_i (x_i w + b) - 1 + \xi_i \geq 0, \xi_i \geq 0, \forall i \end{aligned} \qquad (2)$$

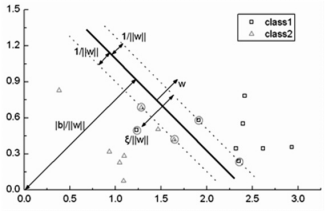

Fig. 1. Soft support vectors and margin

where C controls the trade-off between the ξ penalty and the size of the margin. By introducing Lagrange multipliers α_i , the dual form of (2) can be formed as

$$L_D \equiv \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij} \qquad (3)$$

$$s.t. \quad 0 \le \alpha_i \le C, \forall i \quad and \quad \sum_{i=1}^{l} \alpha_i y_i = 0$$

Here the used kernel function that is defined as follows:

$$k(x_i, x_j) = \phi(x_i)\phi(x_j) \qquad (4)$$

where $k(x_i, x_j)$ is a function in the vector space and is equal to the inner product of two vectors in feature space. By solving the dual optimization problem, the decision function is given by

$$f(x) = sign\left(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b \right). \qquad (5)$$

Here, we choose RBF kernel as our kernel function. The RBF kernel nonlinearly maps examples into a higher dimensional space; it can handle the situation when the relation between class labels and attributes is nonlinear:

$$k(x_i, x_j) = e^{-\|x-y\|^2 / 2\delta^2} \qquad (6)$$

We use the “one against one” approach in which classifiers are constructed and each one train data from two different classes. Then each test data was classified by all of these classifiers and a vote is given to winner class in every classification. The class which has maximum vote will be considered as a label of the test data.

3.2. Radial basis function neural networks

RBF network is as supervised learning neural networks which has two-layer architecture where each unit in the hidden layer represents a radial basis function. These units measure the degree of overlap between input vectors and a set of prototypes drawn from the training set (Fig 2) [12] [13].

A RBFN is a mapping $M: R^n \rightarrow$ such that each input vector $x_i \in$ is of dimension and vectors $C_i \in R^n$ $(j = 1..$ representing the prototypes of the input vectors. The output space of the mapping is of - dimensions (i.e., size of the output vectors). The output of each RBF unit is given as:

$$\phi_j(x_i) = \phi_j\left(\left\| x_i - C_j \right\| \right) \qquad (7)$$

where $\|.\|$ is the Euclidean norm on the input space to compute the distance between the n-dimensional input i and a hidden unit j. The function ϕ has various forms. Here, the Gaussian function is considered. Therefore, ϕ is:

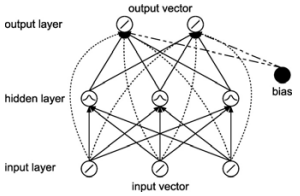

Fig. 2. Radial Basis Function Neural Network schema

$$\phi_j(x_i) = \exp\left(-\frac{\|x_i - C_j\|^2}{\sigma_j^2}\right) \quad (8)$$

where is the width of the th RBF unit. The th output $y_i($, of a RBF network according to the weighted sum option is in which $w(k,$ is the bias of the th output:

$$y_k(x_i) = \sum_{j=0}^{H} \phi_j(x_i).w(k, j) \quad (9)$$

Let denote an *N*-by-*N* matrix with elements: $\Phi=\{\varphi_{ji} \mid j,i = 1,2,...,N\}$ .The prototypes and the widths . Generally, prototypes representing the classes of the input space are found using clustering algorithms. Using the centers found, the widths, which are the radii of the Gaussian basis functions.

During the training stage, for each data point $x_i, y_k($ is computed. This can be expressed in a matrix form as:

$$Y = \Phi W \quad (10)$$

The goal of the training stage is to find the weight . This can be done by computing directly, i.e. $W = \Phi^{-1}Y$ provided that is nonsingular. To avoid the singularity problem, a small value is added to the diagonal terms, i.e., if we let $\varphi = \Phi +$, then: $W = \varphi^{-1}Y$. Where is the identity matrix. Where $\Phi$is the inverse of the interpolation matrix .

4. Dataset

For experimental studies we have used ten common Persian fonts database. This database contains 5000 font images from 10 different fonts. So, the number of images for each font in our experiments is 500 [2]. Fig 3 shows some samples images of this database.

Feature extraction technique based on wavelet was used. Applying a gridding approach we divide each texture of size 128*128 into 16 sub blocks of size 32*32 and combined wavelet energy and wavelet packet energy features of each sub block to obtain a feature vector.

Fig. 3. Some samples images of this database

In the 2-D case, the wavelet transform is usually performed by applying a separable filter bank to the image. The wavelet decomposition of a 2-D image can be obtained by performing the filtering consecutively along horizontal and vertical directions.

The wavelet energy is the sum of square of detailed wavelet transform coefficients. For an Image of size $N \times N$ the related wavelet energy can be calculated in horizontal, vertical and diagonals direction at $i-$level, respectively as below:

$$E_i^h = \sum_{x=1}^{N}\sum_{y=1}^{N}(H_i(x,y))^2$$
$$E_i^v = \sum_{x=1}^{N}\sum_{y=1}^{N}(V_i(x,y))^2 \qquad (11)$$
$$E_i^d = \sum_{x=1}^{N}\sum_{y=1}^{N}(D_i(x,y))^2$$

$(E_i^h, E_i^v, E_i^d)_{i=1,2,...,K}$ is wavelet energy feature vector. Where K is the total wavelet decomposition level and H_i, V_i, D_i are the wavelet coefficient in horizontal, vertical and diagonals direction, respectively.

Wavelet packet transform recursively decomposes the high-frequency components, thus constructing a tree-structured multiband extension of the wavelet transform. An image of a square local area is decomposed and related wavelet packet coefficients are extracted, then the following averaged energy is calculated:

$$E = \frac{1}{N * N}\sum_{i=1}^{N}\sum_{j=1}^{N}[s(i,j)]^2 \qquad (12)$$

Where $s(i,j)$ denotes the wavelet coefficients of a feature sub image in the N×N window centered at pixel (i,j) [14].

5. Experimental results

As we talked before, the dataset consists of 500 images for each font. We randomly select 50 images from each category as test data. The remaining images are divided to two subsets of labelled and unlabeled data. The amount of labelled data is very few at first, and this amount increases in next experiment, i.e. the amount of unlabelled data decreases in next experiment. The amount of K in KNN classifier is considered 10.

We compare the result of supervised and proposed semi-supervised algorithms in font recognition area together. The result of experiment is shown in Fig 4.

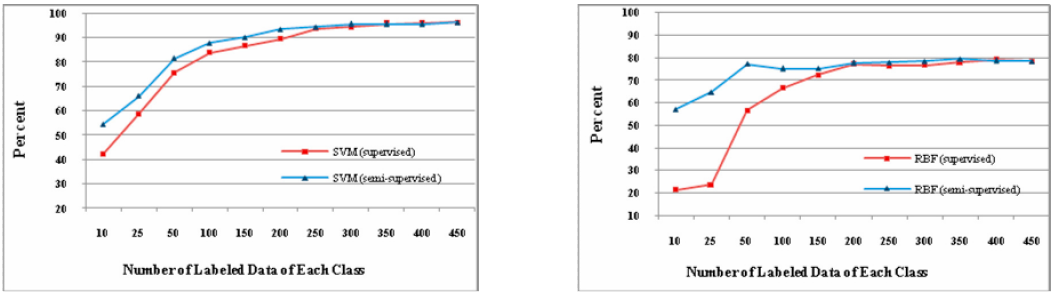

Fig. 4. Comparisons of proposed semi-supervised algorithm with supervised algorithm (a) SVM and (b) RBF

According to the Fig 4 the effect of semi-supervised algorithm is obvious when the number of labeled data is very few. However, after increasing the number of labeled data in each step, the accuracy difference of semi-supervised and supervised algorithm becomes less and less.

6. Conclusion

This paper is concerned with the problem of using both labeled and unlabeled data to classify the ten common Persian fonts for the first time. The usefulness and the contribution of unlabeled data are shown. The proposed semi-supervised algorithm is a self-training which teaches itself by labeled and unlabeled data. In this approach we utilize the majority vote technique of three well-known machine learning algorithms to teach the unlabeled data iteratively by dividing them to the reliable and unreliable data. Then reliable data which predicted labels are assigned to them are added to training data. The algorithm repeats with new labeled training dataset and unreliable ones. We evaluated the proposed semi-supervised algorithm with test data. Experimental results show that the proposed algorithm improves performance of Persian font recognition especially when few amounts of labeled data are available.

Acknowledgement

This work is supported by Education & Research Institute for ICT (ERICT) under grant number 8971/500.

References

1. Y. Zhu, T. Tan, and Y. WANG, *Font Recognition Based On Global Texture Analysis*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2001) p. 1192-1200.
2. H.Khosravi and E. Kabir, *Farsi font recognition based on Sobel-Roberts features.* Pattern Recognition Letters, 31 (2010).
3. C.A. Cruz, et al., *High order statistical texture analysis font recognition applied.* Pattern Recognition Letters, 26 (2005) p. 135-145.
4. M.C.Jung, Y.C. Shin, and S.N. Srihari, *Multifont classification using typographical attributes*, in *IEEE Computer Society Washington*, DC, USA (1999) p. 353 - 356.
5. A. Zramdini, *Study of optical font recognition based on global typographical features*, Ph.D. dissertation, University of Fribourg, 1995.
6. H. Nezam, S. Nezam Abadipour, and V. Saryazdi and Ebrahimi, *Font recognition based on Gabor filters (in Farsi)*, in *9th Iranian Computer Conference* (2003) p. 371-378.
7. E. Rashedi, H. Nezamabadi-pour, and S. Saryzadi, *Farsi font recognition using correlation coefficients (in Farsi)*, in *4th Conf. on Machine Vision and Image Processing* (2007) Iran.
8. A. Borji and M. Hamidi, *Support vector machine for Persian font recognition.* International Journal of Computer Systems Science and Engineering, vol.2, No.3, (2007).
9. X.J. Zhu, *Semi-supervised learning literature survey*, Technical Report 1530, Department of Computer Sciences, University of Wisconsin: Madison 2008.
10. C. Cortes and V. Vapnik, *Support-vector networks.* Machine Learning, Vol.20, No.3 (1995) p. 273-297.
11. C. J. C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition.* Data Mining and Knowledge Discovery, Vol.2 No.2 (1998) p. 121-167.
12. J. Moody and C. Darken, *Fast learning in networks of locally-tuned processing units.* Neural Computation,. Vol.1, No.2 (1989) p. 284–294.
13. Abdelhamid Bouchachia. *RBF Networks for Learning from Partially Labeled Data.* in *Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data*, Bonn, Germany, (2005).
14. G. V. Wouwer, S. Livens, P. Scheunders, D. V. Dyck, *Color Texture Classification by Wavelet Energy Correlation Signatur*es, Proceedings of the 9th International Conference on Image Analysis and Processing, vol. 1, pp.327 - 334, 1997.