Adaptive Sketchy Shape Recognition Based on SVM Incremental Learning

Zhengxing Sun and Lisha Zhang

State Key Lab for Novel Software Technology,
Nanjing University, 210093, China

szx@nju.edu.cn, lisa@graphics.nju.edu.cn

Abstract

This paper presents a strategy of adaptive online sketchy shape recognition. The inputting sketchy shapes are recognized online by means of a modified Support Vector Machine (SVM) incremental learning classifier. All classified results evaluated by user are collected and some important samples are selected according to their distances to the hyper-plane of the SVM-classifier. The classifier can then do incremental learning quickly on the newly added samples, and the retrained classifier can be adaptive to the user's drawing styles. Experiments show the effectiveness of the proposed method.

Keyword: Sketch-based User Interface, Online Graphics Recognition, Adaptive Sketchy Shape Recognition, Incremental Learning, Support Vector Machine (SVM)

I. Introduction

Even in this high-tech computer era, paper and pencils are still designers' preferred choices to quickly sketch bursting ideas. As computers become integrated into everyday life, pen-based user interface is considered as a primary input method. Moreover, the feature to rapidly visualize and deliver human's ideas using graphic objects, which cannot be efficiently represented by speech or text, is highly desirable. The sketchy shape in its rough state contains more information than the regularized one. But, its ambiguity and uncertainty make the deduction of intents very difficult. It will be more helpful if the sketchy shape can be online recognized and converted into the user-intended regular shape and user can realize errors or inappropriateness earlier with the online immediate feedback. This is referenced as online sketch recognition [1].

Numerous researchers have been working on this subject for many years and have produced a wide variety of techniques. However, the poor efficiency of recognition engines is always frustrating, especially for the newly added users, even in the latest experiments. This is because that the styles of sketching vary with different users, even the same user at different time. Therefore, adaptive sketchy shape recognition should be required [2], where recognition engine should incrementally be trainable and adaptable to a particular user's sketching styles. In this paper, we will present a strategy of online adaptive sketchy shape recognition based on our modified SVM (Support Vector Machine) incremental learning algorithm. The primary contribution is to make a mechanism to collect all samples evaluated online by user during his/her drawing, and to select some important samples as incremental training data of classifier. The classifier can then do incremental learning quickly on these newly obtained samples, and the new classifier is adaptive to the user's drawing style.

The remainder of this paper is organized as follows: The related works in sketch recognition are summarized in Section 2. In Section 3, our proposed strategy for online adaptive sketchy shape recognition based on SVM incremental learning classifier is described in detail. In Section 4, we present the experiment and evaluation. Conclusions and future works are given in the final Section.

II. Related Works

Numerous scientists have been working on sketch recognition for many years and have produced a wide variety of representation and recognition techniques.

Graph-based methods have been applied to hand-drawn pattern recognition problems [3][4][5]. In these methods, input patterns are first decomposed into basic geometric primitives, such as lines and curves, and then assembled into a graph structure that encodes both the intrinsic attributes of the primitives and their relationships. Pattern detection is then formulated as a graph isomorphism problem. The practical limitations of graph-based approaches are their computational complexity and their high sensitivity to the segmentation process.

As an alternative to graph-based methods, Fonseca et al [6] proposed a method of symbol recognition using fuzzy logic based on a number of rotation invariant global features. Because their classification relies on aggregate features of the pen strokes, it might be difficult to differentiate between similar shapes. Rubine [7] describes a trainable gesture recognizer for direct manipulation interfaces. A linear classifier is constructed whose weights are learned from the set of training examples. However, it is sensitive to the drawing direction and orientation because a gesture is characterized by a set of 11 geometric and 2 dynamic attributes.

Inspired by the success of machine learning in speech recognition, researchers have recently considered these techniques as another approach to hand-drawn pattern recognition. Sezgin et al [8] view the observed pattern as the result of a stochastic process that is governed by a hidden stochastic model and identified according to its probability of generating the output. However, the strong temporal organization of HMM would limit the flexibility of drawing. Furthermore, the need for large training data sets may inhibit the use of machine learning when such data is scarce. In [2], we have proposed a modified SVM incremental learning method for sketch recognition. However, only effect of the classifier is to do stroke classification for dynamic user modeling, not for adaptive sketch recognition. Our strategy for adaptive sketchy shape recognition method based on SVM incremental learning is only outlined in [9].

In addition, there has been a great deal of works concerning handwritten character and digit recognition [10]. Almost all of these methods have traditionally employed statistical learning methods in which each character or digit is learned from a large corpus of training data. However, due to their need for large training sets, these systems are not easily extensible to diverse applications and are normally useful only for the patterns they are originally trained for.

In summary, an important limitation in all above methods is that they did not consider the adaptabilities of recognition engine to users' drawing styles at all. This is why the poor efficiency of them is always causing for the newly added users. In fact, freehand-drawing for the same shape may be different from user to user and even from time to time by the same user, and benefiting from advances in online sketchy shape recognition system could not be expected before the problem of adaptive sketchy shape recognition is well solved.

III. Adaptive Sketchy Shape Recognition based on SVM Classifier

In a broad sense, adaptive sketchy shape recognition means that the recognition engine should be adjustable to fit the variations of user's sketchy shapes dynamically. Accordingly, the classifier should be able to analyze the incremental samples for user's different drawing styles and be re-trained with the newly added samples obtained. On the other hand, it is also very difficult and burdensome to collect and label artificially all incremental samples for retraining classifier because of varieties of sketching. In this section, we will introduce our strategy for adaptive sketchy shape recognition based on our modified SVM incremental learning algorithm.

A. *Framework of Proposed Strategy for Adaptive Sketch Recognition*

Our framework for adaptive sketchy shape recognition is shown in Fig 1. The processes can be summarized as three aspects: raw stroke processing and feature extraction, online sketchy shape recognition/classification based on SVM classifier, newly important samples selection and incremental training of SVM classifier. The details of our framework are described in the following.

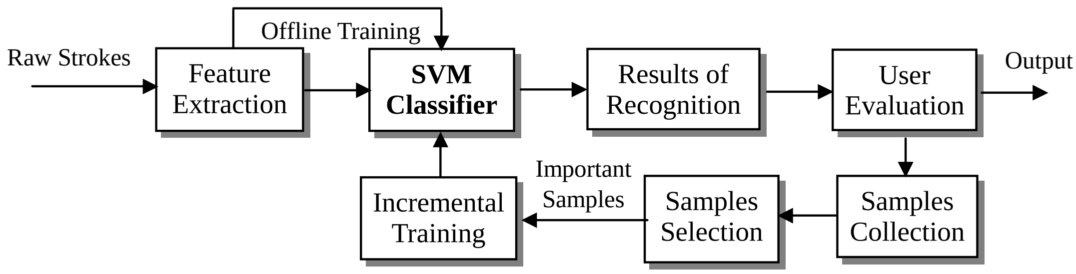


Figure 1. Framework of adaptive sketch recognition based on incremental learning

The raw strokes pre-processing is firstly used to eliminate the noise caused by input conditions or input styles and the inputting sketchy shapes are treated as the composition of some continuously connected strokes. Feature extraction is then applied to obtain the feature vectors of sketchy shapes based on an improvement of the turning function [9]. Details can be seen in [1].

In succession, sketchy shape recognition is done by means of the SVM approach. The SVM classifier is constructed using many binary SVM classifiers, where a RBF (Radial Basic Function) kernel [12] is used and the implementation of the training process is the same as in SVMTorch [13]. The "possible shapes" can then be recognized by means of the trained or re-trained SVM classifier. Additionally, the recognized results must undergo rectification so that the sketchy shapes drawn by the user look very similar to the one in his/her mind and easy to be evaluated by user. Our approach is to rectify the recognized shapes to the regular form predefined.

Although the recognition precisions of the trained SVM classifier could be very high, it still may not be suitable for a specific user's drawing styles, and user would correct some results of recognition during his/her input as shown in Fig 1. To make classifier to adapt the users' styles, we design a strategy to record and collect the samples evaluated by user, select some important samples as incremental training data of classifier until the results of classifiers are satisfactory or enough training samples are obtained, and re-train the SVM classifier with the newly added samples obtained. We name these three processes as "sample collection", "sample selection" and "incremental training" respectively. This strategy can greatly reduce the workload of manual collecting and labeling the newly added training samples, and can incrementally adapt the settings of classifier to users' accustomed styles, not only for the patterns they are originally trained for.

For example, it is quite often that a user draws a triangle very quickly such that the angle is not very obvious and the system may recognize it as a quadrangle or ellipse as shown in Fig 2(a)(b)(c), and quadrangle and pentagon as ellipse by mistake, as shown in Fig 2(d). By our strategy, user can correct these errors immediately and the system will collect all these samples. After several samples are corrected, the system will do incremental learning on these new samples, and the new classifier is adaptive to recognize them correctly, as is shown in Fig 2(e)(f)(g)(h).

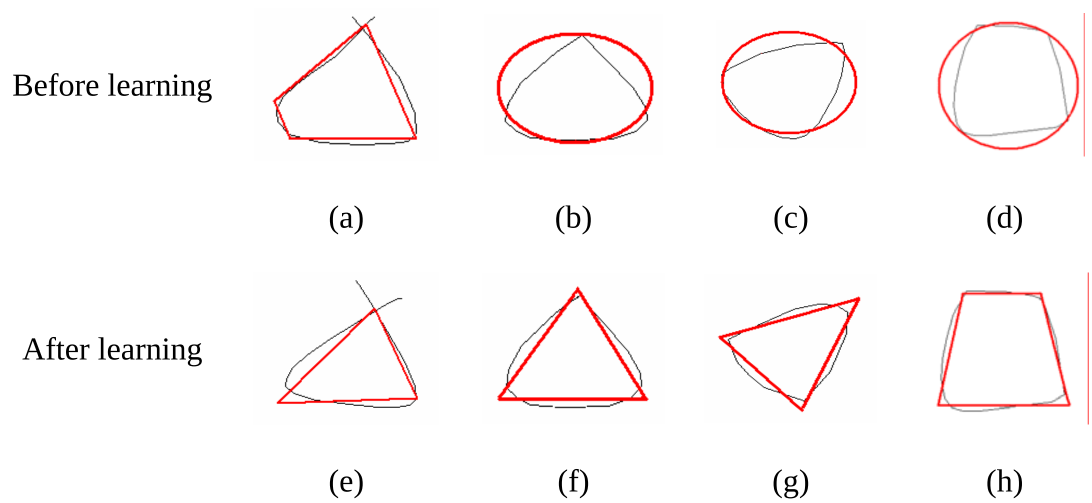


Figure 2. Illustration of the adaptive sketchy shape recognition

B. *Principle of SVM Incremental Learning Algorithm*

The main idea of SVM is to construct a nonlinear kernel function [12] to map the data from the input space into a possible high-dimensional feature space and then generalize the optimal hyper-plane with maximum margin between the two classes. Hence, it is basically used for binary (positive or negative) classification.

The equation of the optimal hyper-plane separating the two classes can be expressed by $wx + b = 0$, where w denotes the normal of the hyper-plane, b denotes the offset. Given a set of n training samples: $\{(x_1, y_1), \cdots, (x_n, y_n)\}$, where $x_i \in R^d$ denotes a vector (samples) in a d-dimensional space and $y_i \in \{+1, -1\}$ represents the class to which x_i belongs. The normal of the optimal hyper-plane w proved by Vapnik [13] can be expressed by a linear combination of the training samples as in Eq. (1).

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \, , \; 0 < \alpha_i \le C \tag{1}$$

The objective of training is to obtain each sample's α value. Classification is done based on the test sample's distance to the hyper-plane. The decision function used to classify the test sample x can be illustrated in Eq. (2).

$$f(x) = \operatorname{sgn}\left[\sum_{i=1}^{n} y_i \alpha_i (x \cdot x_i) + b\right] \quad \textbf{(2)}$$

If $f(x) = +1$, x is classified as positive; if $f(x) = -1$, x is classified as negative.

In most cases, the sample space is not linearly separable. A mapping Φ is usually used to transform non-linearly the input samples into a high dimensional feature space so as to make these samples linearly separable. Therefore, the format of the decision function has been changed to:

$$f(x) = \operatorname{sgn}\left[\sum_{i=1}^{n} y_i \alpha_i (\Phi(x) \cdot \Phi(x_i)) + b\right] = \operatorname{sgn}\left[\sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + b\right] \quad \textbf{(3)}$$

Where, $K(x, y) = \Phi(x)\Phi(y)$ is known as the kernel function.

Usually, only a small portion of samples has non-zero α_i coefficients, whose corresponding x_i (which is referred to as support vectors, or in short, SVs) and y_i fully define the decision function. Therefore, the SV set can fully describe the classification characteristics of the entire training set. In other words, the generalization property of an SVM does not depend on all the training data but only Support Vectors. The optimal hyper-plane can then be expressed as:

$$\sum_{x_i \in SV} \alpha_i y_i K(x, x_i) + b = 0, \quad 0 < \alpha_i \le C \quad \textbf{(4)}$$

That is, $w_0 \Phi(x) + b = 0$, where $w_0 = \sum_{x_i \in SV} y_i \alpha_i \Phi(x_i)$

In the training process, the training set $\boldsymbol{\mu}$ is divided into two sub-sets: the SV set and the non-SV set. If most SVs are not on the classification boundary, the computational complexity of the training process can be specified in Eq. (5)[14].

$$O(\|\mu_{sv}\|^3 + \|\mu\| \times \|\mu_{sv}\|^2 + d\|\mu\| \times \|\mu_{sv}\|) \quad \textbf{(5)}$$

Where, ***d*** is a constant value.

Generally, $\|\mu_{sv}\| << \|\mu\|$ and they are correlated. Therefore, the training complexity can be specified in Eq. (6).

$$O(\|\mu\| \times \|\mu_{sv}\|^2) \quad \textbf{(6)}$$

Because the training process of SVM involves solving a quadratic programming problem, the computational complexity of the training process is very high with respect to the number of training samples. Hence, if we train SVM classifiers on the SV set instead of the whole training set, the training time can be reduced greatly with the same classification precision or a little loss. This is the main idea of certain incremental learning algorithms.

Accordingly, we have proposed a modified SVM incremental learning algorithm by inserting an evaluating process in the training process of SVM incremental learning. The evaluating process is used to evaluate and select some samples so that the training would only preserve the SVs at each incremental step and then add them instead of all historic data to the training sets collected in the evaluating step. Denoting the training process of SVM as ***Train*(*μ*)**, the process of evaluating all newly added samples as ***Evaluate*(*μ*)**, the initial and incremental training samples as ***IS*** and ***INS*** respectively, the temporary training samples as ***NS*** and working data as ***WS***, the process of our algorithm can be briefly described as:

(1). ***Γ*=*Train***(***IS***), ***WS*=*IS*$_{SV}$**.

(2). ***Γ*=*Train***(***WS***), ***WS*=*WS*$_{SV}$**.

(3). ***NS***=***Evaluate***(***INS***), ***WS***=***WS*∪*NS***.

(4). Retraining. That is, repeat (2) and (3) until the results of SVM classifiers are satisfactory.

For a SVM classifier, we define two learning strategy: one is repetitive learning, whose training process can be specified as ***Train***(***IS*∪*INS***); the other is incremental learning, whose training process can be specified as ***Train***(***Sub***(***IS***)∪***INS***), where ***Sub***(***IS***)⊂***IS***. A basic assumption for incremental learning is that the incremental training set does not change the classification boundary very much. Otherwise, repetitive training may be a better choice. If this pre-condition is met, we obtain:

$$\|(IS \cup INS)_{sv}\| \approx \|(IS_{sv} \cup INS)_{sv}\| \approx \|IS_{sv}\| \quad \textbf{(7)}$$

Because in most cases, $\|INS\| << \|IS\|$, the computational complexity of the repetitive learning is:

$$O(\|IS \cup INS\| \times \|IS_{sv}\|^2) = O(\|IS\| \|IS_{sv}\|^2) \quad \textbf{(8)}$$

Meanwhile, the computing complexity of the evaluation process is:

$$O(\|INS\| \times \|IS_{sv}\|) \quad \textbf{(9)}$$

That is, the complexity of the evaluation process is much less than the training complexity. So, the computational complexity of the training process in our proposed incremental learning is:

$$O(\|IS_{sv} \cup NS\| \times \|IS_{sv}\|^2) \tag{10}$$

Because our incremental learning method only retrains the classifier with the SV set and those important samples and $\|NS\| << \|INS\|$, our proposed incremental learning is faster than the repetitive learning method. The detailed descriptions of our algorithm can be found in [2]. Our experiments have also shown that: our proposed method can reduce the classifier's training time without loss of classification precision, and the one-against-one (binary-class) structure classifiers outperform the one-against-all (multi-class) structure classifiers in both training time and classification precision for incremental learning [2].

C. *Online Selection of Retraining Samples for SVM Incremental Learning*

For online adaptive sketchy shape recognition based on SVM incremental learning, the core is how to select some important samples as incremental training data of SVM classifier. That is, SVM classifier must analyze the collected samples evaluated by user, pick out the most important instances as training samples, and distinguish them from the rests in the sample collections before training.

By means of our modified SVM incremental learning algorithm, the candidate incremental training samples are determined by estimating their distances from the hyper-plane. We build the hyper-plane as expressed in Eq. (4), and the coefficients α_i in it can be obtained from solving the following optimization problem:

$$\begin{cases} \text{Minimize}: w(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \Phi(x_i)\Phi(x_j) \\ \text{Subject to constraints}: \sum_{i=1}^{l} \alpha_i y_i = 0, \alpha_i \geq 0 \end{cases} \tag{11}$$

The distance between the sample x and the hyper-plane can be defined as:

$$d(x, w) = \frac{|w_0 \Phi(x) + b|}{\|w_0\|} \tag{12}$$

Since $\|w_0\| = 1$, the distance can then be changed to:

$$d(x,w) = |w_0 \cdot \Phi(x) + b| = |\sum_{x_i \in SV} \alpha_i y_i K(x, x_i) + b| \tag{13}$$

During sample selection, a threshold of distance must be selected. Intuitively, there is a conflict between the precision and the speed if a constant distance threshold is used to choose the important samples. Hence, we consider using a dynamic threshold in our strategy. In the beginning of incremental learning, due to the small size of the newly training dataset, the time to re-train SVM classifiers is relatively short and the recognition precision of SVM classifiers is also relatively low. In this case, a larger threshold value is used such that the precision can be a little bit higher while the time is still acceptable. As the number of added training data increases in later incremental learning steps, the cost (in both time and space) of training and the recognition precision will also increase. In this case, the selection process should judge the importance of every newly added sample more carefully with a smaller threshold in order to avoid substantial increase of the number samples necessary for re-training of the SVM classifiers. In another aspect, if we want to adapt to a new user, we can use a large threshold value to adapt him quickly. Using this dynamic threshold method, we can obtain higher precision with shorter training time.

IV. Experiments and Performance Evaluation

To validate the performance of our proposed method, we have done experiments and comparisons of four algorithms (the repetitive learning algorithm [14], Syed et al's algorithm [15], Xiao et al's algorithm [16] and ours). We collected two datasets, a 5-class dataset (including triangle, quadrangle, pentagon, hexagon, and ellipse) and a 14-class dataset shown in Fig 3(a) are the classes of sketchy shapes and Fig 3(b) are their regular shapes). The 5-class dataset is used to validate our strategy for adaptive sketchy shape recognition. The 14-class dataset together with the 5-class dataset are used to evaluate and compare the performance of the four algorithms. All experiments are done on an Intel PC (with a 2.4GHz CPU and 256MB memory) running on Microsoft Windows XP Professional.

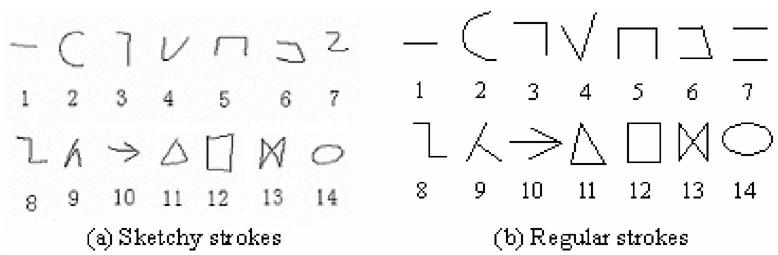
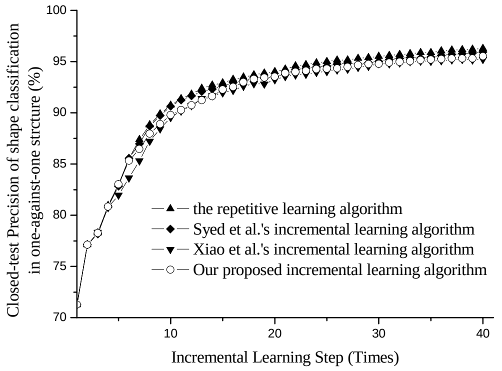
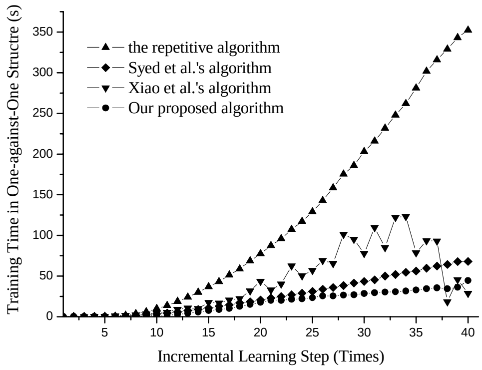


Figure 3. Strokes of 14 classes frequently used in sketching

For 5-class sketchy shape dataset, each user draws each shape repeatedly using the mouse and pen/tablet respectively. In total, we have collected 1367 sample shapes drawn with pen/tablet and 325 samplers drawn with mouse, and we can have 40 samples for each shape by using a 20-dimension feature vector and after transformation. We select randomly 20210 samples of the pen/tablet to form a test set, TS_1, and use 12000 samples of the mouse style to form another test set, TS_2. Then, we randomly select samples from the remained samples of the pen/tablet style to form 39 incremental training sample sets (denoted as: IS_1, IS_2… IS_{39}). The first 6 incremental training sets have 100, 100, 120, 150, 300, 700 samples, respectively, and each of the rest 33 sets has 1000 samples. The remained samples of the mouse style form a sample set with 1000 samples and are added into the incremental learning process as the last training set (denoted as IS_{40}). Specially, for Xiao et al's algorithm [16], IS_1+IS_2+IS_3 are used as the first training set because it cannot terminate if the training set is too small. Fig 4 shows the performance comparison among the four algorithms using the same one-against-one classifier structure on the 5-class dataset. For each algorithm, we use the same 40 incremental training sample sets to train them incrementally and record the incremental training time in Fig 4(b). After each incremental training step, we test it using the two test sets, TS_1 and TS_2, respectively, and record their classification precisions. Fig 4(a) shows the curves of the precisions using test set TS_1 (the precision on TS_2 is almost same with on $TS_1$and elided here).



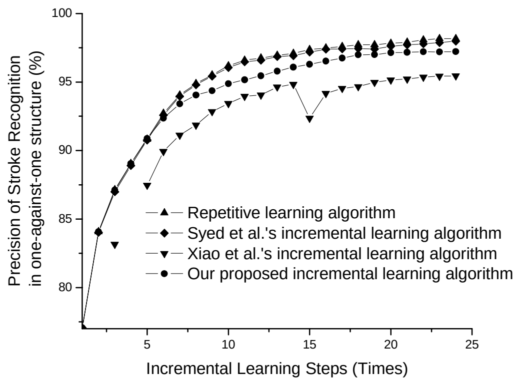
(a) Precision comparison (on TS_1)



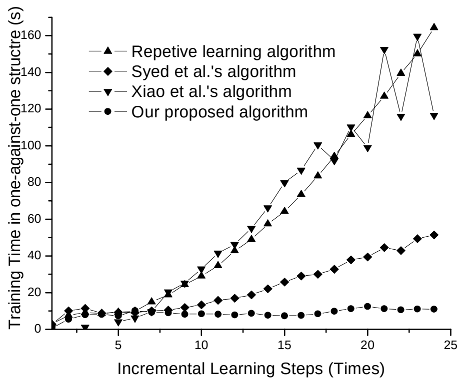
(b) Training time comparison

Figure 4. Performance evaluation using 1-1 classifier structure based on the 5-class dataset

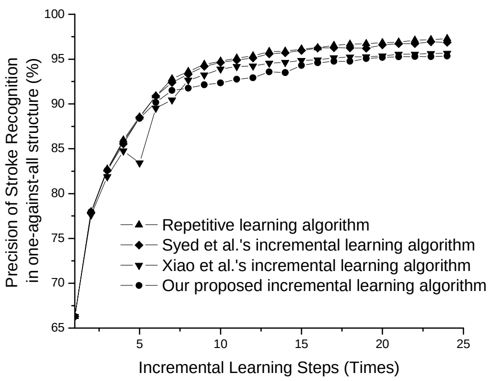
For the 14-class dataset, we collected 52802 samples of 14 classes of commonly used strokes. Each sample is represented using a 20-dimentional features vector based on the turning function. We randomly select 21932 samples to constitute the test set (TS_3) and then randomly select samples from the remaining samples to form 24 incremental training sample sets (denoted as S_1, S_2… S_{24}). The first 5 incremental training sets have 300, 300, 370, 500, and 900 samples, separately, and each of the rest 19 sets has 1500 samples, respectively. Specially, for Xiao et al's algorithm, we use $S_1+S_2+S_3$ and S_4+S_5 as the first two training sets. Fig 5 shows the performance comparison among the four algorithms using two different classifier structures on the 14-class dataset. For each structure, we use the same 24 incremental training sample sets to train them incrementally and record the incremental training time in Fig 5(b) and (d). We also test their classification precisions on the test set (TS_3) and show them in Fig 5(a) and (c).



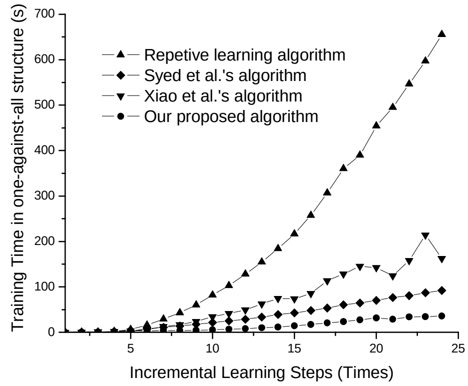
(a) Precision using 1-1 structure



(b) Training time using 1-1 structure



(c) Precision using 1-m structure



(d) Training time using 1-m structure

Figure 5. Performance comparison among the four algorithms based on the 14-class dataset

Another complete comparison experiments are also done in our works [17], which compares the performances of online adaptive sketchy shape recognition between different machine learning methods (including our modified SVM incremental learning, our modified Hidden Markov Model (HMM) and our modified Bayesian Belief Network (BN)) under various feature representations (including Rubine features [7], our modified turning function, compositional features, and so on) and multi-users based on 10-class of electric symbols (GBT4728 and GBT5465). Details can be seen from Ref. [17].

From all of our experiments, we can conclude that our method can do well in reducing the training time (this can be seen from Fig 4(b) and Fig 5(b)(d)) with very little loss of precision (this can be seen from Fig 4(a) and Fig 5(a)(c)). These prove that our strategy for online adaptive sketchy shape recognition is feasible. However, it can only suit for simple shapes drawn by some continuous strokes. This limitation is derived from that the inputting sketchy shapes are treated as the composition of some continuous connected strokes and modeled with the modified turning function.

V. Conclusion

The difficulty in online sketchy shape recognition systems comes from that sketching is usually informal, inconsistent and ambiguous both in intra-person and inter-person settings in a given situation. Sketch recognition engine should automatically adapt to a particular user’s sketching styles. Benefiting from advances in online sketchy shape recognition system could not be expected before the problem of adaptive sketch recognition is well solved.

Human cognition is usually performed incrementally, and adaptive iteratively to a particular user’s drawing styles means to retrain the recognizer/classifier with training samples obtained for this user. By utilizing the advantages of our modified SVM incremental learning algorithm, this paper presents a strategy of adaptive sketchy shape recognition, which can collect the classified results evaluated by user during his/her drawing and select some important samples as incremental training data of classifier according to their distance to the hyper-plane of the SVM-classifier. The classifier can then do incremental learning quickly on the newly added samples, and the retrained classifier can be adaptive to the user's drawing styles. Experiments show its efficiency for adaptive online sketchy shape recognition.

Nevertheless, the model of our proposed classifier is stroke-based and the sketchy shapes are composed of some continuous strokes. For some composite shapes with some disconnected multi-strokes, the virtual strokes must be introduced to link these strokes orderly and translate the multi-strokes into a single stroke. The main problem lies that the strokes are not the basic constitutive geometric primitives of a shape for human cognition and the shape representation based on a single stroke composed of physic stokes and virtual strokes is not structural and unique for user. Therefore, a novel feature-based computation model of classifier should be constructed based on geometric primitives and be not sensitive to stroke-segmentation for online adaptive sketchy composite shape recognition. All of these are the direction of our further works.

Acknowledgement: The work described in this paper was supported by a grant from National Natural Science Foundation of China (Project No. 69903006 and 60373065) and the Program for New Century Excellent Talents in University of China (2004).

References

[1] Zhengxing Sun and Jing Liu, "Informal User Interface for Graphic Computing", Lecture Notes in Computer Science, J. Tao, T. Tan and R. W. Picard Eds, Springer-Verlag, 2005, Vol. 3784, pp. 675-682.

[2] Zhengxing Sun, Wenyin Liu, Binbin Peng, et al, "User Adaptation for Online Sketchy Shape Recognition", Lecture Notes in Computer Science, J. Llados and Y. B. Kwon Eds, Springer-Verlag, 2004, Vol. 3088, pp. 303-314.

[3] Chris Calhoun, Thomas F Stahovich, Tolga Kurtoglu, et al, "Recognizing multi-stroke symbols", In AAAI Spring Symposium on Sketch Understanding, AAAI Technical Report SS-02-08, Stanford University, 2002, pp. 15-23.

[4] Seong-Whan Lee, "Recognizing hand-drawn electrical circuit symbols with attributed graph matching", Structured Document Image Analysis, Springer-Verlag, 1992, pp. 340-358.

[5] Xiaogang Xu, Zhengxing Sun, Binbin Peng, et al, "An online composite graphics recognition approach based on matching of spatial relation graphs", International Journal of Document Analysis and Recognition, 2004, March, 7(1), pp. 44-55.

[6] Manueal J Fonseca, Cesar Pimentel, and Jaoquim A Jorge, "Cali-an online scribble recognizer for calligraphic interfaces", In: AAAI Spring Symposium on Sketch Understanding, AAAI Technical Report SS-02-08, Stanford University, 2002, pp. 51-58.

[7] Dean Rubine, "Specifying gestures by example", Computer Graphics, 1991, Vol. 25, pp. 329-337.

[8] Tevfik Metin Sezgin and Randall Davis, "HMM-Based Efficient Sketch Recognition", Proceedings of the 10th international conference on Intelligent user interfaces, San Diego, California, USA, 2005, pp. 281-283.

[9] Zhengxing Sun, Lisha Zhang and Enyi Tang, "An Incremental Learning Method Based on SVM For Online Sketchy Shape Recognition", Lecture Notes in Computer Science, L. Wang, K. Chen and Y. S. Ong Eds, Springer-Verlag, 2004, Vol. 3610, pp. 655-659.

[10] Rejean Plamondon and Sargur N Srihari, "On-Line and off-line Handwriting Recognition: A Comprehensive Survey", IEEE Transaction on Pattern Analysis and Machine Intelligence, 2000,Vol. 22, No. 1, pp. 63-84.

[11] M A Esther, "An Efficient Computable Metric for Comparing Polygonal Shapes", IEEE Trans. on Pattern Analysis and Machine Intelligence, 1991, Vol. 13, No. 3, pp. 209-216.

[12] J Ratsaby: "Incremental Learning with Sample Queries", IEEE Trans. on Pattern Analysis and Machine Intelligence, 1998, Vol. 20, pp. 883-888.
[13] R Collobert and S Bengio, "SVMtorch: Support Vector Machines for Large-Scale Regression Problems", Journal of Machine Learning Research, 2001, pp.143-160.
[14] V Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, 1995.
[15] N Syed, H Liu and K K Sung, "A Study of Support Vectors on Model Independent Example Selection", Proceedings of International Conference on Knowledge Discovery & Data Mining (KDD'99), San Diego, CA, USA, August 15-18, 1999, pp. 272-276.
[16] Rong Xiao, Jicheng Wang, Zhengxing Sun, et al, "An Incremental SVM Learning Algorithm Alpha-ISVM", Journal of Software (In Chinese), 2001, Vol. 12, No. 12, pp. 1818-1824.
[17] Lisha Zhang, Zhengxing Sun, Bin Zhang, "Machine Learning Methods for Adaptive Online Sketch Recognition: An Experimental Comparison", Proceeding of 2005 International Conference on Intelligent Computing (ICIC05), Hefei, China, August 23-26, 2005, pp. 825-834.

Zhengxing Sun received his B.Engg degree from Southeast University, and his M.Engg and Ph.D. degrees from Nanjing University of Aeronautics and Astronautics, in China, respectively. He is now a Full Professor in Department of Computer Science of Nanjing University. His research interests include: Multimedia Computing, Perceptual Intelligence, Intelligence Human Computer Interaction and Pervasive Computing. He has authored or co-authored more than 80 publications in these areas.

Lisha Zhang received her B.S degree from Nanjing University in China. She has finished all the courses for M.S degree and is now directly a Ph.D. candidate in Department of Computer Science and Technology of Nanjing University. She does researches in Intelligent Human Computer Interaction, Adaptive Sketch Recognition and Handwritten Signature Identification. She has accomplished and coauthored more than 10 papers in these areas.