Prototype Selection Methods for On-line HWR

Jakob Sternby

Prototype Selection Methods for On-line HWR

Jakob Sternby
jakob@maths.lth.se
Centre for Mathematical Sciences
Sölvegatan 18, Box 118
S-221 00, Lund, Sweden

Abstract

Prototype matching strategies such as DP-matching are powerful methods for character recognition problems with very large number of classes such as on-line Chinese character recognition. As for many problems with a large number of classes data is normally comparatively scarce for on-line Chinese characters and therefore prototype selection methods for use under these circumstances need to be robust against over-training. The methods investigated in this paper are incremental in the sense that all prototype additions are based on the number of misclassifications caused by the current database. This may be highly beneficial as it is possible to use the methods on databases that have undergone other optimization methods. Experiments have been conducted on the HANDS Kanji data and it reveals some interesting results.

1 Introduction

The problem of on-line handwriting recognition of Chinese characters differs from the problem of recognizing alphabetic characters by its huge number of classes. For alphabetic handwriting recognition recent research has seen a lot of focus on learning methods such as Neural Networks [7] whereas methods based on a distance function (normally some kind of DP-matching) with a prototype database is still an active area of research for Chinese character recognition [4]. Methods consisting of a prototype database also require methods for selecting prototypes to include in the database. For alphabetic handwriting recognition it is conventional to use clustering techniques that aim at selecting a set of distinct representatives of shape variations for each class [1]. This is however not an optimal method with respect to recognition accuracy and it is has been shown that a method that clusters with respect to the neighborhood of other classes as perceived by the recognition engine may improve the recognition performance of distance based recognition methods [9]. Furthermore especially for Chinese characters there may be situations where it is not preferable to construct a completely new database but to improve the performance of a given database by adjusting the prototype set. One way to do this is by altering the actual feature vectors of the database by methods such as LVQ [3]. The methods in this paper instead aim at finding optimal prototypes to add to a given database and the methods presented here select prototypes from a test/training set with the overall aim of increasing recognition accuracy. The methods contain various parameters that can be altered for example to minimize the risk of over-training. The experiments have been conducted with a conventional DP-matching technique on the TUAT HANDS-kuchibue-d_97-06-10 database [6] consisting of almost twelve thousand character samples from 10 different writers.

2 Prototype selection methods

This paper describes three methods for selecting prototypes from a set to include in a database. Like most training methods for pattern recognition the selection is intricately dependent on the quality, size and variance of the sample set used for training. The methods are iterative and at most one prototype for one class is added in one iteration. To safeguard against overtraining on the training set there is a limit stating that a prototype needs to correct at least a threshold T errors to be allowed to be added. A problem when adding prototypes for a recognition task with as many classes as Chinese characters is that it may be difficult to foresee the impact a new prototype has on the total recognition accuracy. This is the issue that is in focus here and that the different methods treat in different ways. All of the methods presented here aim at not reducing the recognition rate on any character based on one such addition operation. For this reason the neighborhood of each character is also evaluated for each selected prototype according to Algorithm 1. Let $\mathbb{X} = \{x_j\}$ be the set of all character samples and let $\mathcal{C}(x_j)$ be the class label of sample x_j from the set of all class labels $\mathbb{L} = \{l_i\}^m$. Let the neighborhood $\mathcal{N}(x_j)$ of a sample be the set of the n closest classes to x_j based on the distance function $d(\cdot, \cdot)$ and the current prototype database. Denote the interpretation of a sample x_j by $I(x_j) \in \mathbb{L}$.

All the three methods for prototype selection presented in this paper are based on Algorithm 1 and they differ only in the two items marked by ① and ⑪ as described below.

2.1 The mean method

This method is based on the principle that the best representative of a new set of samples (i.e. a set of currently

Algorithm 1 Prototype Selection

for $i = 1, \ldots, m$ **do**
  $E_{l_i} := \{e_{i,j}\}_j = \{x | \mathcal{C}(x) = l_i, l_i \neq I(x)\}$
  for $j = 1, \ldots, |E_{l_i}|$ **do**
    ① Calculate addition effect of sample $e_{i,j}$
  end for
  if ② Sample e_{i,j^*} should be added according to rule **then**
    Add sample e_{i,j^*} to database prototype set
    for k s.t. $l_k \in \mathcal{N}(e_{i,j^*})$ **do**
      Recalculate E_{l_k} with
    end for
  end if
end for

misinterpreted samples) is the one that is closest to most other misinterpreted samples. This is somewhat similar to the motivation for the voting method presented in [8]. In this paper however the sample achieving the smallest mean distance value to the other methods will however be chosen instead of the sample with the highest count of closest samples. Furthermore the strategy as described in [8] computes the distances to all samples and not just the remaining misinterpretations. The specifics of Algorithm 1 of the mean method are specified as

① Calculate the distance to all other samples of E_{l_i} as well as the number of samples n_C corrected in E_{l_i} if the sample were to be added to the database.

② The sample $e_{i,j^*} = \text{argmin}_{e \in E_{l_i}} \sum_{j=1}^{|E_{l_i}|} d(e, e_{i,j})$ is added to the prototype database if the number of samples that this corrects n_C is larger than the threshold T.

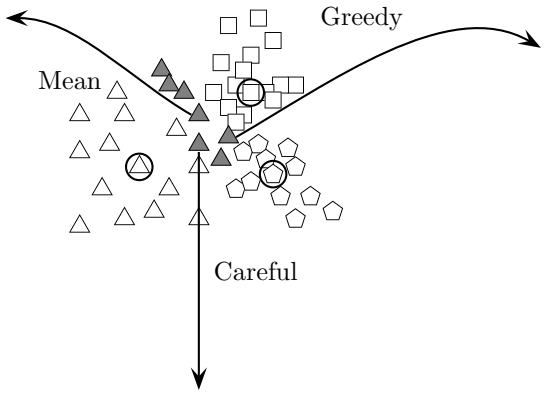

Figure 1: A graphic example of possible selections of prototypes made from the class of triangles. The gray samples mark the misinterpreted triangles from a prototype database consisting of the three encircled samples.

2.2 The careful method

The effect on the recognition accuracy of other classes when adding a prototype is very hard to foresee. What is here labeled as the *careful* method aims at evaluating not only the effect of addition of a prototype on the same class, but also the effect on the recognition of other classes. With this careful approach a sample is only added to the database if its net effect on the recognition result is positive. This can be seen as a cautious way of adding samples and it is unlikely that this will lead to some classes having extremely low recognition accuracies. However, some classes may be so interconnected that it is impossible to find samples that bring a net positive result to recognition accuracy.

The modifications to Algorithm 1 with the careful method are listed below.

① Calculate the possible number of corrections n_C that each sample $e_{i,j}$ performs on E_{l_i} like in the *mean* method, but here also calculate the number of caused misinterpretations n_D in the neighboring classes $\mathcal{N}(e_{i,j})$

② The sample e_{i,j^*} with the largest possible difference $n_C - n_D$ is added to the prototype database if $n_C - n_D > T$.

As seen in Figure 1 it is probable that the *careful* method selects a prototype that lies close to the other misinterpretations but as far away as possible from the other classes.

2.3 The greedy method

This is perhaps the most interesting of the three methods presented in this paper. The principle of the *greedy* method presented here is that most classes in a problem like on-line Chinese character recognition with a high number of classes will be considerably disjoint. This means that if the addition of a valid sample of one class (i.e. a sample that visually clearly belongs to its class) causes misinterpretations of samples of another class, there should exist a sample of the other class that balances the new prototype definition space so that both classes achieve a totally higher recognition rate. At the addition stage, unlike the *careful* method, the number of misinterpretations of other classes is not considered. Instead for each class and iteration the sample that corrects the most number of misinterpretations is added. This is actually the only difference with the *mean* method. Where the *mean* method tries to choose its representative by adding the sample that best represents the mislabellings in according to the geometry of the distance function (which is in most DP cases does not abide the triangle inequality [1]), the *greedy* method chooses the optimal sample in view of current recognition results.

The stages of Algorithm 1 for the *greedy* method are listed below.

① Calculate the possible number of corrections n_C that each sample $e_{i,j}$ performs on E_{l_i} like in the *mean* method and the *careful* methods.

② Add the sample e_{i,j^*} with the largest number of corrected samples n_C if this number exceeds the threshold T.

3 Experiments

Experiments with the different prototype selection methods presented above were conducted on a conventional DP-based matching algorithm. As is customary in Chinese (Kanji) character recognition preprocessing on each character was performed by a variant of line density equalization [2]. Feature points were then chosen recursively by picking the most distant point between the start and endpoint of a segment on a curve [5]. The prototype database consisted of the feature points of the samples chosen for inclusion. Since the *greedy* and *careful* method need to compare recognition of a character to an already existent database all methods were initialized by the running one iteration of the *mean* method as described in Section 2.1. Prototype distance was calculated by conventional Dynamic Time Warping on the 2-d feature points [1].

In these experiments data was not divided into a separate test data set and training data set since the methods presented here have intrinsic properties that restrain over-training. However, since all these methods employ voting-related strategies when selecting prototypes, an upper limit for the number of samples being targeted for prototype selection was set to 35. The number of samples from each class ranged from 10 to about 90 in the TUAT HANDS-kuchibue-d_97-06-10 database used and setting an upper limit for the number of samples to influence the prototype selection therefore acts as a frequency normalization having an effect that it may be up to 3.5 as important to add samples of one class as that of another. For some characters there are therefore also independent test sets not used during the training.

All of the methods presented here are iterative and prototypes are added incrementally to the initial database constructed by the *mean* method above. From the results presented in Figure 2 it is clear that the *greedy* method provides the highest recognition accuracy but at the cost of a larger number of prototypes. The *careful* method also provides higher recognition results than the geometrically defined *mean* method and although it does not give as high results as the *greedy* method, it may be a safer way to increment the database since it is more unlikely to decrease the recognition rate of any character.

4 Conclusions

This paper presents three different methods for prototypes selection especially intended for character recognition with a high number of classes such as Chinese characters. Clustering or feature vector adjustment with LVQ are other methods that aim at improving the content of a prototype database. The methods presented in this paper are in some respects a more pragmatic approach to the problem and could be used for extending coverage of given databases to new writing styles. The methods are constructed in such a way that over-training is unlikely and should simply result in the algorithms refusing to increment the prototype database. This property may be of particular interest to problems with very high number of classes as data may be scarce.

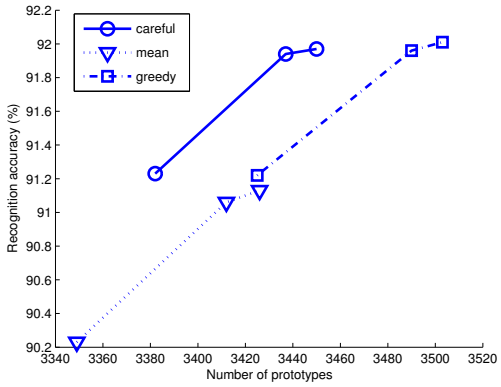

Figure 2: A plot of the recognition accuracy as a function of the number of prototypes in the database for the various prototype selection methods presented here.

References

[1] C. Bahlmann and H. Burkhardt. The writer independent online handwriting recognition system *frog on hand* and cluster generative statistical dynamic time warping. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(3):299–310, March 2004.

[2] S. Jäger, C-L. Liu, and M. Nakagawa. The state of the art in japanese online handwriting recognition compared to techniques in western handwriting recognition. *IJDAR*, 6(2):75–88, 2003.

[3] T. Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, 1990.

[4] C-L. Liu, S. Jäger, and M. Nakagawa. Online recognition of chinese characters: The state-of-the-art. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):198–213, 2004.

[5] M. Nakagawa, K. Akiyama, L. V. Tu, A. Homma, and T. Kigashiyama. Robust and highly customizable recognition of on-line handwritten japanese characters. In *Proc. 13th International Conference on Pattern Recognition*, pages 269–273, Washington, DC, USA, 1996. IEEE Computer Society.

[6] M. Nakagawa, T. Higashiyama, Y. Yamanaka, S. Sawada, L. Higashigawa, and K. Akiyama. Online handwritten character pattern database sampled in a sequence of sentences without any writing instructions. In *Proc. of the 4th International Conference on Document Analysis and Recognition*, pages 376–381, 1997.

[7] R. Plamondon and S. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1):63–84, January 2000.

[8] H. Rowley, M. Goyal, and J. Bennett. The effect of large training set sizes on online japanese kanji and english cursive recognizers. In *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition*, page 36, Washington, DC, USA, 2002. IEEE Computer Society.

[9] J. Sternby. Class dependent cluster refinement. In *Proc. 18th International Conference on Pattern Recognition*, 2006. Accepted.