

Dynamically Subsumed-OVA SVMs for Fingerprint Classification

Jin-Hyuk Hong and Sung-Bae Cho

Dept. of Computer Science, Yonsei University
134 Sinchon-dong, Sudaemoon-ku
Seoul 120-749, Korea

`hjinhs@scslab.yonsei.ac.kr`, `sbcho@cs.yonsei.ac.kr`

Abstract. A novel method to fingerprint classification, in which the naïve Bayes classifier (NB) and OVA SVMs are integrated, is presented. In order to solve the tie problem of combing OVA SVMs, we propose a subsumption architecture dynamically organized by the probability of classes. NB calculates the probability using singularities and pseudo codes, while OVA SVMs are trained on FingerCode. The proposed method not only tolerates ambiguous fingerprint images by combining different fingerprint features, but produces a classification accuracy of 90.8% for 5-class classification on the NIST 4 database, that is higher than conventional methods.

1 Introduction

Since the Henry system categorizes fingerprints by the relative position and number of core and delta points, many researchers have tried to extract them in the flow of the ridges [1]. Karu and Jain proposed a heuristic algorithm with singularities [2], while Zhang and Yan used singularities together with pseudo ridges to classify fingerprints [3]. In order to obtain a high classification rate, various features have been also actively investigated. Jain, *et al.* proposed FingerCode that uses a Gabor filter to extract the directional ridge flow [4], Park used the orientation filtered by fast Fourier transform [5], where Min, *et al.* proposed localized models of SVMs using FingerCode [6]. There are some other attempts to integrate several features and methods to produce a robust fingerprint classifier [1, 7]. Senior used hidden Markov models and decision trees to recognize the ridge structure of the print [1], while Yao, *et al.* combined flat and structured features using the recursive neural networks and support vector machines (SVMs) [7].

This paper describes a novel fingerprint classification approach integrating the naïve Bayes classifier (NB) and SVMs. In order to accomplish highly accurate classification, SVMs with FingerCode are generated based on the one-vs-all (OVA) scheme while NB with singularities dynamically organizes them.

2 A Dynamic Fingerprint Classifier

2.1 Overall Architecture

Contrary to conventional methods that have a static classification process, we propose a dynamic fingerprint classifier that not only uses various fingerprint features

(singularity, pseudo ridges and FingerCode) but also solves the ambiguity of OVA SVMs. The proposed method consists of NB and OVA SVMs as shown in Fig. 1. NB estimates the posterior probability for fingerprint classes $prob = \{p_w, p_l, p_r, p_a, p_t\}$ by using singular points and pseudo ridges, while OVA SVMs classify fingerprints by using the FingerCode, where the margin of a sample $o\text{-svm} = \{m_w, m_r, m_l, m_a, m_t\}$ is produced. We evaluate them based on the subsumption architecture to manage the ambiguity such as ties and rejects. The subsumption architecture selects an action when there are multiple models by sequentially evaluating each model. When a model is satisfied, it suppresses the other models.

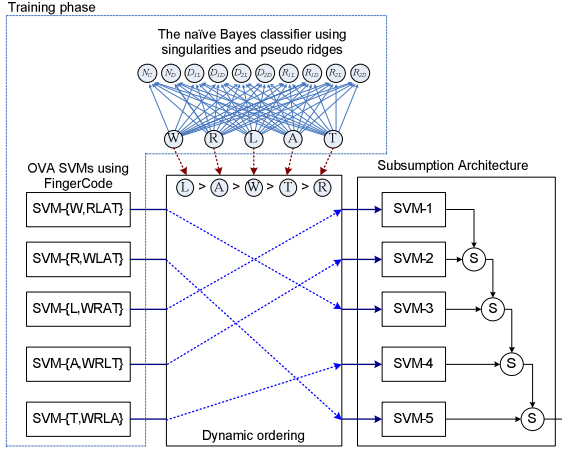


Fig. 1. Overview of the proposed method

<pre> prob[5] = {p_w, p_r, p_l, p_a, p_t} // prob[] is calculated by the naïve Bayes classifier order[5] = {0, 1, 2, 3, 4} o-svm[5] = {m_w, m_r, m_l, m_a, m_t} // o-svm[] is obtained by the OVA SVMs // determine the order of OVA SVMs to evaluate for(i=0; i<5; i++) for(j=i+1; j<5; j++) if(prob[i] < prob[j]) { int iTemp = prob[i]; prob[i] = prob[j]; prob[j] = iTemp; iTemp = order[i]; order[i] = order[j]; order[j] = iTemp; } </pre>	<pre> // classify with OVA SVMs according to the subsumption architecture if(prob[order[0]] < r₁) // r₁ is a rejection threshold return reject; for(i=0; i<5; i++) { if(o-svm[order[i]] >= a) // a is a threshold { if(o-svm[order[i]] < r₂) // r₂ is a rejection threshold return reject; return order[i]; } } return order[0]; </pre>
---	--

Fig. 2. Pseudo code for probabilistically ordering OVA SVMs

As shown in Fig. 1, the order of OVA SVMs to evaluate is determined by the posterior probability of each class which NB produces. The corresponding OVA SVM of a more probable class takes precedence in the subsumption architecture against the other OVA SVMs. A sample is evaluated sequentially until an OVA SVM is satisfied. When an OVA SVM satisfies, the sample is classified into the corresponding class of the OVA SVM, while it is classified into the class of the highest probability when none of OVA SVMs are satisfied. Fig. 2 shows the pseudo code for the proposed method. Ordering OVA SVMs properly for an input sample provides dynamic classification.

2.2 Fingerprint Features: FingerCode, Singularities and Pseudo Codes

In this paper, we use FingerCode to train OVA SVMs, and singularities and pseudo codes as the feature of the naïve Bayes classifier. FingerCode, proposed by Jain in 1999, is a representative fingerprint feature extracted from the NIST database 4 [4]. The algorithm sets a registration point in a fingerprint image and tessellates it into 48 sectors. The Gabor filter is applied with 4 directions (0° , 45° , 90° , and 135°) so as to accentuate the ridge parallel to each direction. Ridges that are not parallel to the direction will be blurred. Standard deviations are computed on 48 sectors for each of the 4 filtered images to generate the 192-dimensional feature vector called FingerCode.

The number and location of core and delta points are used to classify fingerprints, where the poincare index is the representative algorithm detect singular points [3, 8]. The number of cores and deltas are denoted as NC and ND , respectively. The nearest core to the center is denoted as C (if there is no core, C represents the center of the image). Pseudo ridges proposed by Zhang, *et al.* consist of a predefined number (100 in this paper) of points [3]. A pseudo ridge is composed of 200 points based on orientation in two opposite directions from the starting point C . The naïve Bayes classifier is learned with 5 mutually exclusive and exhaustive classes (W , R , L , A , T) and 10 features (NC , ND , $D1L$, $D1D$, $D2L$, $D2D$, $R1L$, $R1D$, $R2L$, $R2D$), where each class is linked with all features as shown in Table 1.

Table 1. Features of the naïve Bayes classifier

Feature	Definition	State
N_C, N_D	Number of core points and delta points	0, 1, 2
D_{1L}, D_{2L}	Location of delta point	0, 1, 2, 3, 4, Absent
R_{1L}, R_{2L}	Location of the end point of the pseudo ridge	0, 1, 2, 3, 4, <i>turn</i>
D_{1D}, D_{2D}	Distance between C and delta points	1, 2, 3, Absent
R_{1D}, R_{2D}	Distance between C and the end point of the pseudo ridge	1, 2, 3, <i>turn</i>

3 Experimental Results

3.1 Dataset

The NIST Database 4 is used to verify the proposed method [9]. It consists of 4,000 scanned images of 512×512 resolution obtained from two impressions (F and S) of 2,000 fingerprints. Fingerprints are equally distributed into 5 classes, whorl (W), right

loop (R), left loop (L), arch (A) and tented arch (T). Due to the ambiguity in fingerprints, 350 fingerprints (17.5%) are cross-referenced with two classes. The first label is only considered in training SVMs and NB while both labels are used in the test. In the experiment, the fingerprints of the first impression are used as the training set (F0001~F2000), and the others construct the test set. FingerCode proposed by Jain is used after normalization from -1 to 1, where some rejected images are included in the training set (1.4%) and the test set (1.85) [4]. The LIBSVM package (available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>) provides SVMs using the Gaussian kernel with $\sigma^2=0.0625$.

3.2 Results and Analysis

The proposed method, that combines NB and SVMs dynamically, is aimed at classifying fingerprints by handling more subtle discriminations. Although the major classification task is performed by SVMs, NB works as assistant by ordering them. It results an accuracy of 90.8% that is higher than the other methods including previously published ones. As shown in Table 2, the proposed method is better than SVMs with winner-takes-all as well as NB. The confusion matrix is shown in Table 3, in which fingerprints are evenly distributed comparing with NB or SVMs.

We have compared the proposed method with other methods that have previously been published. Several points are plotted in Fig. 3 along with the curve of possible

Table 2. Comparison with individual approaches

Method	Accuracy
NB	85.4%
SVMs (Winner-takes-all)	90.1%
NB+SVMs (Product)	90.2%
Proposed method	90.8%

Table 3. Confusion matrix of the proposed method

True class	Assigned class				
	W	R	L	A	T
W	373	10	10	0	0
R	4	374	0	6	15
L	5	0	377	8	9
A	0	6	4	365	40
T	1	8	15	39	295

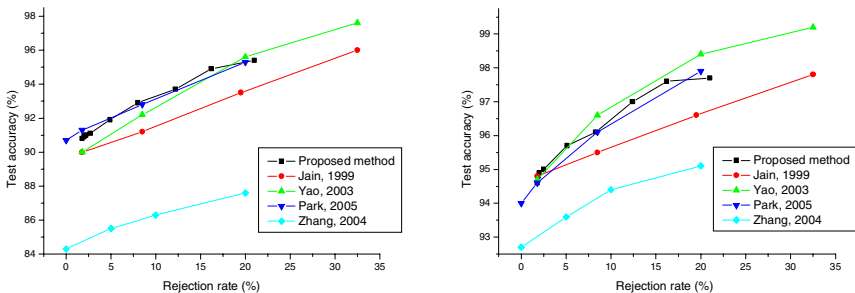


Fig. 3. Five-class and four-class accuracy-rejection plots of the NIST 4 database

points for the proposed method here and the methods of Jain, *et al.* [4], Yao, *et al.* [7], Park [5], and Zhang and Yan [3]. For each method, the accuracy of the classifier is shown with the corresponding rejection rate. As shown in Fig. 3, the proposed method yields higher performance than the others in both classification tasks.

4 Conclusion

In this paper we have proposed a novel fingerprint classification method effectively integrating NB and OVA SVMs, which gives better accuracy than previously reported in the literature on the NIST-4 database. Several popular fingerprint features such as singularities, pseudo codes and FingerCode are used in the proposed method, where the combination of methods described here produces better results (90.8% for the five-class classification problem and 94.9% for the four-class classification problem with 1.8% rejection during the feature extraction phase of FingerCode) than any of the component classifiers. The propose method also solves tie problems which often occurs in multi-class classification by ordering OVA SVMs based on the probability of classes.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei Univeristy. The authors also would like to thank Prof. Anil Jain and Dr. Salil Prabhakar for providing FingerCode data.

References

- [1] A. Senior, "A combination fingerprint classifier," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1165-1174, 2001.
- [2] K. Karu and A. Jain, "Fingerprint classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389-404, 1996.
- [3] Q. Zhang and H. Yan, "Fingerprint classification based on extraction and analysis of singularities and pseudo ridges," *Pattern Recognition*, vol. 37, no. 11, pp. 2233-2243, 2004.
- [4] A. Jain, S Prabhakar and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348-359, 1999.
- [5] C. Park and H. Park, "Fingerprint classification using fast Fourier transform and nonlinear discriminant analysis," *Pattern Recognition*, vol. 38, no. 4, pp. 495-503, 2005.
- [6] J.-K. Min, J.-H. Hong and S.-B. Cho, "Effective fingerprint classification by localized models of support vector machines," *Lecture Notes in Computer Science*, vol. 3832, pp. 287-293, 2006.
- [7] Y. Yao, G. Marcialis, M. Pontil, P. Frasconi and F. Roli, "Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines," *Pattern Recognition*, vol. 36, no. 2, pp. 397-406, 2003.
- [8] N. Yager and A. Amin, "Fingerprint classification: A review," *Pattern Analysis and Applications*, vol. 7, no. 1, pp. 77-93, 2004.
- [9] C. Watson and C. Wilson, Fingerprint Database, National Institute of Standard and Technology, Special Database 4, *FPDB*, April 1992.