Combining Multiple Classifiers based on Third-Order Dependency*

Hee-Joong Kang
Division of Computer Engineering
Hansung University
389 Samsun-dong 2-ga, Sungbuk-gu, Seoul, Korea
hjkang@hansung.ac.kr

David Doermann
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742-3275, USA
doermann@umiacs.umd.edu

Abstract

Without an independence assumption, combining multiple classifiers deals with a high order probability distribution composed of classifiers and a class label. Storing and estimating the high order probability distribution is exponentially complex and unmanageable in theoretical analysis, so we rely on an approximation scheme using the dependency. In this paper, as an extension of the second-order dependency approach, the probability distribution is optimally approximated by the third-order dependency and multiple classifiers are combined. The proposed method is evaluated on the recognition of unconstrained handwritten numerals from Concordia University and the University of California, Irvine. Experimental results support the proposed method as a promising approach.

1. Introduction

Many methods for combining multiple classifiers have been proposed using three decision forms: measurement scores, ranking, and a single choice [1]. It is desirable that combination methods be developed at a single choice level, because the combination methods could be independent of the classification results, and the results easily combined. Previous combination methods at the single choice level include majority voting [1], the Behavior-Knowledge Space (BKS) method [2], the use of a Dempster-Shafer formalism used in evidential reasoning [3], and the use of a Bayesian formalism with an independence assumption [1, 4] or a dependency-based approximation [5].

Combining multiple classifiers at the single choice level can be formulated as follows, using probability theory and the Bayesian formalism. When an input x is given to K classifiers (e.g., $E_1, E_2, \cdots, E_K$) in parallel, a K-dimensional decision vector $C =< E_1(x) = M_1, E_2(x) = M_2, \cdots, E_K(x) = M_K >$ is observed, where a set of L decisions or classes is denoted by $M = \{M_1, M_2, \cdots, M_L\}$. A set of K-dimensional decision vectors and a set of class labels can be used to build a high order frequency table with the BKS method and to estimate a high order probability distribution with the use of the Bayesian formalism.

The main task of combining multiple classifiers with the Bayesian formalism is to determine a hypothesized class m which maximizes a posterior probability P^*, that is, $\underset{m}{max}\ P(m \in M | E_1(x) = M_1, E_2(x) = M_2, \cdots, E_K(x) = M_K)$. That is, a $(K+1)$st-order probability distribution should be estimated from samples at a training stage. This idea entails the computation of the $(K+1)$st-order probability distribution, $P(m, E_1(x) = M_1, E_2(x) = M_2, \cdots, E_K(x) = M_K)$.

Without an independence assumption, dealing with such a high order probability distribution composed of classifiers and a class label is exponentially complex and unmanageable in theoretical analysis, so we rely on an approximation scheme using the dependency. Chow and Liu [6] attempted to approximate an nth-order distribution with a product of $(n-1)$ second-order component distributions by the first-order dependency. To find an optimal product set of $(n-1)$ first-order dependencies among n variables, a procedure was derived to yield an approximation of a minimum difference in information. Kang et al. [5] incorporated the dependency into combining multiple classifiers. In addition to finding the optimal set by first-order dependencies, a new method to find an optimal product approximation by second-order dependency is proposed using the same measure of the minimum information.

In this paper, as an extension of the second-order dependency approach, the probability distribution is optimally approximated by the third-order dependency and multiple classifiers are combined. The proposed method is evaluated with multiple classifiers recognizing unconstrained hand-

*Dr. Kang is a visiting researcher at the LAMP laboratory, University of Maryland. This work was supported by the postdoctoral fellowships program from Korea Science & Engineering Foundation (KOSEF) and the DOD under contract MDA90402C0406 and NSF under grant EIA0130422.

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

written numerals from Concordia University and the University of California, Irvine.

The remainder of this paper is organized as follows. Section 2 explains how to combine multiple classifiers based on dependency. Background on finding the optimal product approximation by the third-order dependency is provided in Section 3. The experimental results of combining multiple classifiers based on dependency are provided in Section 4 and a discussion is presented in Section 5.

2. Combining Multiple Classifiers based on Dependency

The assumption that classifiers perform independent of each other is not invulnerable, because the classifiers tend to be statistically dependent on others. Therefore, it is desirable to take into account dependencies among the classifiers. Combining multiple classifiers based on dependency consists of two sequential steps: a dependency-directed approximation and a combination using the approximation by the Bayesian formalism. The dependency-directed approximation finds the optimal product set of the $(K+1)$st-order probability distribution by the dth-order dependency, where $(1 \leq d \leq K)$. The probabilistic combination applies the optimal product set found in the dependency-directed approximation to the Bayesian decision rules for combining multiple classifiers.

In this paper, it is assumed that the dependency is statistically measured by computing the average mutual information. The mutual information is defined as a quantitative measure of how much the occurrence of a particular event tells us about the possibility of some alternative [6]. Mathematical background on mutual information has been derived from the measure of closeness by Lewis [7]. Therefore, the dependency provides a theoretical basis for approximating the $(K+1)$st-order probability distributions with a product of low order component distributions.

The dependency-based approximation scheme plays an intermediate role between the Bayesian method based on independence assumptions and the BKS method in several respects. Considering dth-order dependency makes the storage needs of the framework $(K+1-d) \cdot L^{d+1}$ and makes a potentially high rejection rate like in the BKS method lowered. In other words, the complexity of storage needs $O(L^{d+1})$ is in the range of $O(L^2)$ in the case of independence assumption to $O(L^{K+1})$ in the case of BKS method. The order of dependency, d, can be adjusted under permissible resources for better approximation or performance. But, it still remains an open issue as to how one should select an appropriate value of d to obtain the best performance.

For a combination of K classifiers, an optimal product by the dth-order dependency should be found in the first approximation step. Let V be a $(K+1)$-dimensional variable composed of a class label and K decisions. When the $(K+1)$st-order probability is approximated by a dth-order dependency, a criterion is needed to measure how close the approximate distribution $P_a(V)$ is to the actual distribution $P(V)$. Such a criterion, depending on an information theory model, was developed by Lewis and is called the measure of closeness [6, 7]. The measure of closeness I is mathematically defined as follows:

$$I(P(V), P_a(V)) = \sum_V P(V) \log \frac{P(V)}{P_a(V)}. \qquad (1)$$

The closeness of approximation is defined as the difference between the information contained in the actual distribution and the information contained in the approximate distribution. Therefore, the optimal approximate product can be obtained by minimizing the difference of information.

3. Combining Multiple Classifiers based on Third-Order Dependency

In this section, dependency-directed approximation for the optimal product of the $(K+1)$st-order probability distribution by the third-order dependency is described in detail. For notational convenience, we will denote $E_j(x) = M_j$ by C_j and $m \in \mathbf{M}$ by C_{K+1} respectively in the $(K+1)$st-order probability distribution. That is, the $(K+1)$st-order probability distribution $P(V)$ is also represented by $P(C_1, \cdots, C_{K+1})$. We propose a new method for finding the optimal product of the $(K+1)$st-order probability distribution $P(V)$ by the third-order dependency. Considering the first- and second-order dependencies in approximating the high order probability distribution can be found in [5].

When third-order dependency is considered, the approximate distribution is defined in terms of fourth-order component distributions as follows:

$$P_a(C_1, \cdots, C_{K+1}) = \prod_{j=1}^{K+1} P(C_{n_j} | C_{n_{i3(j)}}, C_{n_{i2(j)}}, C_{n_{i1(j)}}), \qquad (2)$$
$$(0 \leq i3(j), i2(j), i1(j) < j),$$

such that C_{n_j} is conditioned on all $C_{n_{i3(j)}}$, $C_{n_{i2(j)}}$ and $C_{n_{i1(j)}}$, and where $(n_1, \cdots, n_K, n_{K+1})$ is an unknown permutation of integers $(1, \cdots, K, K+1)$ and C_0 is a null component. $P(C_{n_j}|C_0, C_0, C_0)$ is $P(C_{n_j})$, $P(C_{n_j}|C_0, C_0, C_{n_{i\cdot(j)}})$ is $P(C_{n_j}|C_{n_{i\cdot(j)}})$, and $P(C_{n_j}|C_0, C_{n_{i\cdot(j)}}, C_{n_{i\cdot(j)}})$ is $P(C_{n_j}|C_{n_{i\cdot(j)}}, C_{n_{i\cdot(j)}})$, by definition, where $C_{n_{i\cdot(j)}}$ is $C_{n_{i3(j)}}$, $C_{n_{i2(j)}}$, or $C_{n_{i1(j)}}$. By applying the following algorithm proposed for a third-order dependency approximation to the $(K+1)$st-order probability distribution P, we can determine the unknown permutation $(n_1, \cdots, n_{K+1})$ and their unknown conditioned permutations $(n_{i3(1)}, \cdots, n_{i3(K+1)})$, $(n_{i2(1)}, \cdots, n_{i2(K+1)})$,

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

and $(n_{i1(1)}, \cdots, n_{i1(K+1)})$ from the obtained optimal third-order dependencies.

On the other hand, if $C_{n_{i1(j)}}$ in the equation (2) is identical to all C_{n_j}, that is, C_j is assumed to be conditionally dependent on $C_{n_{i3(j)}}$ and $C_{n_{i2(j)}}$ for the given C_{K+1}, then the approximate distribution is defined in terms of fourth-order distributions as follows:

$$P_a(C_1, \cdots, C_{K+1}) = \prod_{j=1}^{K} P(C_{n_j} | C_{n_{i3(j)}}, C_{n_{i2(j)}}, C_{K+1}), \quad (3)$$
$$(0 \leq i3(j), i2(j) < j).$$

Such an approximation is regarded as the conditional second-order dependency approximation which can be defined as a specific case of the third-order dependency approximations.

For notational convenience, we will drop the subscript n and denote, for example, C_{n_j} by C_j in subsequent discussions. We can apply the $(K+1)$st-order probability distribution P and the third-order dependency approximation P_a to the measure of closeness (i.e. equation (1)) for an optimal product set as in the following expressions:

$$\begin{aligned} I(P(V), P_a(V)) &= \sum_V P(V) \log \frac{P(V)}{P_a(V)} \\ &= \sum_V P(V) \log P(V) - \\ &\quad \textstyle\sum_{j=1}^{K+1} \sum_V P(V) \log P(C_j | C_{i3(j)}, C_{i2(j)}, C_{i1(j)}) \\ &= -\sum_{j=1}^{K+1} U(C_j; C_{i3(j)}, C_{i2(j)}, C_{i1(j)}) + \sum_{j=1}^{K+1} H(C_j) - H(V) \end{aligned} \quad (4)$$

$$H(V) = -\sum_V P(V) \log P(V) \quad (5)$$

$$H(C_j) = -\sum_V P(V) \log P(C_j) \quad (6)$$

$$U(C_j; C_{i3(j)}, C_{i2(j)}, C_{i1(j)}) = \sum_V P(C_j, C_{i3(j)}, C_{i2(j)}, C_{i1(j)}) \log \frac{P(C_j | C_{i3(j)}, C_{i2(j)}, C_{i1(j)})}{P(C_j)}. \quad (7)$$

From the derived equation (4), minimizing $I(P(V), P_a(V))$ is to maximize $\sum_{j=1}^{K+1} U(C_j; C_{i3(j)}, C_{i2(j)}, C_{i1(j)})$ which is the sum of average third-order mutual information, since remaining entropy terms (i.e. $\prod_{j=1}^{K+1} H(C_j)$ and $H(V)$) are constant. Then, the next step is how to find the optimal product set of third-order dependencies which satisfies the permutation constraints in the formulation of the approximate distributions, from all the permissible product sets. Finding the optimal product set is described in the following algorithm.

Algorithm for third-order dependency by the measure of closeness

Input:
The set of w samples $S^1, S^2, \cdots, S^w$.

Output:
The optimal product set of third-order dependencies as per the average mutual information measure.

Method:
1. Estimate the second-, third-, and fourth-order marginals from the samples.
2. Compute the weights $U(C_j; C_{i(j)})$, $U(C_j; C_{i2(j)}, C_{i1(j)})$, and $U(C_j; C_{i3(j)}, C_{i2(j)}, C_{i1(j)})$ for all pairs, triplets, and quadruplets from the samples.
3. Find the maximum weight of first-, second-, and third-order dependencies and its associated optimal product set, as in the following statements.

maxTweight = 0.;
for $n1 = 1$ **to** number of first-order dependencies **do**
  T1 = weight of the chosen first-order dependency;
  for $n2 = 1$ **to** number of second-order dependencies **do**
    T2 = weight of the chosen second-order dependency according to the chosen first-order one;
    T3 = 0;
    for $n3 = 1$ **to** number of untraversed classifiers **do**
      choose one of untraversed classifiers;
      choose the largest permissible third-order dependencies associated with the chosen classifiers;
      T3 = MAX(*T3*,T1+T2+(weight of the chosen third-order dependencies));
      store *T3* and its associated first-, second-, and third-order dependencies;
    end
  end
  maxTweight = *MAX*(*maxTweight*,*T3*);
  store *maxTweight* and its associated first-, second-, and third-order dependencies;
end
obtain maximum *maxTweight* and its associated first-, second-, and third-order dependencies;

End of Algorithm

This algorithm finds the optimal product set by accumulating the weights and composing the permissible product sets by the **for** loops, as the order of dependency increases from the first to the third, and by choosing the product set having maximum *maxTweight* which is the sum of the average third-order mutual information. The computational complexity of the proposed algorithm is $O(n^3)$, so some efforts will be needed for reducing the computational complexity in the future. However, this algorithm is only run once from training samples, and thus the computational complexity is not significant to the combining multiple classifiers.

By using the systematic dependency-directed approximation, the order of dependency to be considered can be extended to the *d*th-order under permissible resource requirements. The optimal *d*th-order dependency product set consists of one first-order dependency, one second-order dependency, ..., one $(d-1)$st-order dependency, and multiple (i.e. $(K-d)$) *d*th-order dependencies as in the following approximate distribution:

$$P_a(C_1, \cdots, C_{K+1}) = \prod_{j=1}^{K+1} P(C_{n_j} | C_{n_{id(j)}}, \cdots, C_{n_{i1(j)}}), \quad (8)$$
$$(0 \leq id(j), \cdots, i1(j) < j),$$

such that C_{n_j} is conditioned on permissible maximum d components from $C_{n_{i1(j)}}$ to $C_{n_{id(j)}}$, where $(n_1, \cdots, n_K, n_{K+1})$ is an unknown permutation of integers

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

$(1, \cdots, K, K+1)$ and C_0 is a null component. The equation (8) looks like the definition of the chain rule of probability when $(d = K)$, because the chain rule is one of Kth-order dependency approximations.

4. Bayesian Decision Combination

In order to combine K classifiers, the approximate distribution found from the optimal product set of third-order dependencies is applied to the Bayesian formalism. Bayesian decision combination for K classifiers is derived from the Bayesian formalism and the approximate distribution. For the hypothesized class candidate m, a supported belief function $Bel(m)$ is defined by the following expression:

$$Bel(m) = P(m \in \mathbf{M}|C_1, \cdots, C_K). \quad (9)$$

By using the Bayesian theorem and the optimal product set of third-order dependencies, and by allowing the class candidate $m \in \mathbf{M}$ to be denoted by C_{K+1}, we have the following belief expressions from the equations (2) and (9):

$$\begin{aligned} Bel(m) &= P(m \in \mathbf{M}|C_1, \cdots, C_K) \\ &= \frac{P(C_1, \cdots, C_K, C_{K+1})}{P(C_1, \cdots, C_K)} \\ &= \frac{\prod_{j=1}^{K+1} P(C_{n_j}|C_{n_{i3(j)}}, C_{n_{i2(j)}}, C_{n_{i1(j)}})}{P(C_1, \cdots, C_K)} \\ &\approx \eta \prod_{j=1}^{K+1} P(C_{n_j}|C_{n_{i3(j)}}, C_{n_{i2(j)}}, C_{n_{i1(j)}}), \quad (10) \end{aligned}$$

with η as a constant that ensures that $\sum_{i=1}^{L} Bel(M_i) = 1$ and $(n_1, \cdots, n_{K+1})$ as an unknown permutation of integers $(1, \cdots, K+1)$. Depending on the belief $Bel(M_i)$ computed from the given decision vector C, we choose a maximized posterior probability $P^*(m \in \mathbf{M}|C_1, \cdots, C_K)$, and then a combined decision is determined or not, according to the decision rule $D(C)$:

$$D(C) = \begin{cases} M_i, & \text{if } Bel(M_i) = \max\limits_{M_j \in M} Bel(M_j) \\ L+1, & \text{otherwise.} \end{cases} \quad (11)$$

5. Experimental Results

Six classifiers, *E1*, *E2*, *E3*, *E4*, *E5*, *E6*, will be used in this section. These classifiers were developed by using the features in [8, 9] or by using the structural knowledge of numerals in [10], such as the bounding box, centroid, and the width of horizontal runs or strokes, developed at KAIST and Chonbuk National Universities. Their characteristics are described in Table 1. As the classifiers *E4* and *E5* were trained by the structural knowledge obtained from the numerals of Concordia University, they are not as good as the numerals of UCI. The performance of individual classifiers is shown in Figure 1 for the test data sets.

Table 1. Introduction of individual classifiers

	archi.	classifier	distance function
E1	singular	neural net.	pixel distance funct.
E2	modular	neural net.	directional distance distri.
E3	singular	neural net.	mesh feature
E4	modular	rule-based	modified structural know.
E5	modular	rule-based	structural knowledge
E6	singular	neural net.	contour feature

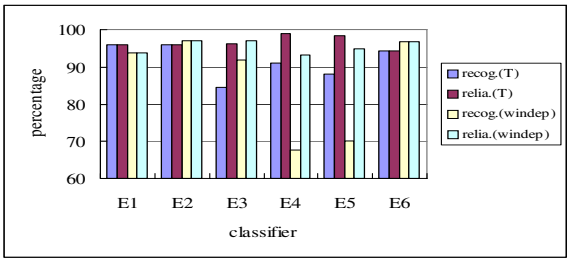

Figure 1. Results of individual classifiers on test data sets: *T, windep*

The handwritten numeral databases are as follows. The UCI data sets in [11] are used for optical recognition of handwritten digits and consist of three training data sets *tra*, *cv*, *wdep* and one test data set *windep*. The data set *tra* has about 190 digits per a class, the data sets *cv*, *wdep* have about 95 digits per a class, and the data set *windep* has about 180 digits per a class. The CENPARMI data sets consist of two training data sets *A*, *B* and one test data set *T*. Each data set has 200 digits per a class. Each neural network based classifier was trained with the training data sets *A* and *tra*. For the dependency-directed approximation, the optimal product sets were found by using the two data sets *A*, *B* and the three data sets *tra*, *cv*, *wdep*. The *reject* results of a classifier were used in finding the optimal product set.

The five classifiers, shown in Table 2, were evaluated by the Bayesian combination methods abbreviated as in Table 3 and the BKS method. From the Figure 2, the second-order dependency provides higher performance than the first-order dependency, however, the third-order dependency does not provide higher performance than the second-order dependency in all groups. On the other hand, the third-order dependency provides higher performance than the second-order dependency in 2 out of 6 groups from the Figure 3 and the second-order dependency provides higher performance than the first-order dependency in all groups. However, the Bayesian combination methods based on de-

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

pendency provide higher performance than the BKS method being without any approximations.

Table 2. Groups of five classifiers

group name	classifiers
D51	*E1, E3, E4, E5, E6*
D52	*E1, E2, E4, E5, E6*
D53	*E1, E2, E3, E4, E6*
D54	*E1, E2, E3, E5, E6*
D55	*E1, E2, E3, E4, E5*
D56	*E2, E3, E4, E5, E6*

Table 3. Bayesian combination methods

method	meaning
CIAB	Condi. Independ. Assump. based Bayesian
FODB	First-Order Dependency based Bayesian
CFODB	Condi. First-Or. Depend. based Bayesian
SODB	Second-Order Depend. based Bayesian
CSODB	Condi. Second-Or. Depend. based Bayesian
TODB	Third-Order Depend. based Bayesian

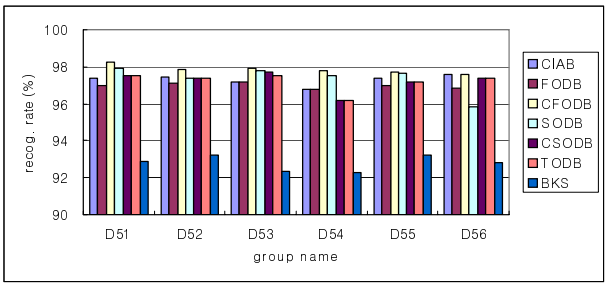

Figure 2. Results of five classifiers on Concordia data set: *T*

6. Discussion

Although the third-order dependency does not provide higher performance than the second-order dependency in all cases, combining multiple classifiers based on third-order dependency is the extended work of second-order dependency based Bayesian combination methods and is useful in a few groups as shown in the results. With a larger the order of dependency, an approximation error becomes smaller, but the computational complexity for finding the optimal product set increases and there are also the risks of over-fitting and data sparsity. So, it will be useful to deal with the selection on an appropriate order of dependency for the best performance.

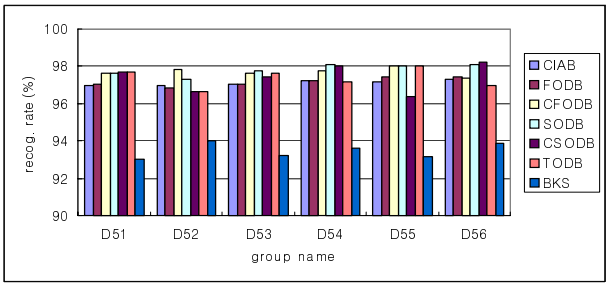

Figure 3. Results of five classifiers on UCI data set: *windep*

References

[1] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition," *IEEE TSMC*, vol. 22, no. 3, pp. 418–435, 1992.

[2] Y. S. Huang and C. Y. Suen, "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90–94, 1995.

[3] J. Franke and E. Mandler, "A Comparison of Two Approaches for Combining the Votes of Cooperating Classifiers," in *Proc. of the 11th Int. Conf. on Pattern Recognition*, vol. 2, pp. 611–614, 1992.

[4] D.-S. Lee and S. N. Srihari, "Handprinted Digit Recognition : A Comparison of Algorithms," in *Proc. of the 3rd Int. Workshop on Frontiers in Handwriting Recognition*, pp. 153–162, 1993.

[5] H.-J. Kang, K. Kim, and J. H. Kim, "Optimal Approximation of Discrete Probability Distribution with *k*th-order Dependency and Its Applications to Combining Multiple Classifiers," *PRL*, vol. 18, no. 6, pp. 515–523, 1997.

[6] C. K. Chow and C. N. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Trans. on Inf. Theo.*, vol. 14, no. 3, pp. 462–467, 1968.

[7] P. M. Lewis, "Approximating Probability Distributions to Reduce Storage Requirement," *Information and Control*, vol. 2, pp. 214–225, Sep. 1959.

[8] T. Matsui, T. Tsutsumida, and S. N. Srihari, "Combination of Stroke/Background Structure and Contour-direction Features in Handprinted Alphanumeric Recognition," in *Proc. of the 4th IWFHR*, pp. 87–96, 1994.

[9] I.-S. Oh and C. Y. Suen, "Distance features for neural network-based recognition of handwritten characters," *IJDAR*, vol. 1, no. 2, pp. 73–88, 1998.

[10] I.-S. Oh, J.-S. Lee, K.-C. Hong, and S.-M. Choi, "Class-expert approach to unconstrained handwritten numeral recognition," in *Proc. of the 5th IWFHR*, pp. 35–40, 1996.

[11] C. Blake and C. Merz, "UCI repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/mlrepository.html]. Irvine, CA, Dept. of Infor. and Comp. Sciences," 1998.

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE