Information Processing Letters 62 (1997) 67–75

Information Processing Letters

Approximating optimally discrete probability distribution with kth-order dependency for combining multiple decisions [1]

Hee-Joong Kang *, Kawon Kim, Jin H. Kim

Computer Science Department and Center for Artificial Intelligence Research, KAIST, 373-1 Kusŏng-dong, Yusŏng-gu, Taejŏn 305-701, South Korea

Received 9 February 1996; revised 7 January 1997
Communicated by K. Ikeda

Abstract

A probabilistic combination of K classifiers' decisions obtained from samples needs a $(K+1)$st-order probability distribution. Chow and Liu (1968) as well as Lewis (1959) proposed an approximation scheme of such a high-order distribution with a product of only first-order tree dependencies. However, if a classifier follows more than two classifiers, such first-order dependency does not estimate adequately a high-order distribution. Therefore, a new method is proposed to approximate optimally the $(K+1)$st-order distribution with a product set of kth-order dependencies where $1 \leqslant k \leqslant K$, which are identified by a systematic dependency-directed approach. And also, a new method is presented to combine probabilistically multiple decisions with the product set of the kth-order dependencies, using a Bayesian formalism.

Keywords: Combining multiple decisions; kth-order dependency; High-order probability distribution; Optimal approximation; Dependency-directed approximation; Probabilistic combination

1. Introduction

Combining multiple decisions in pattern recognition is a matter of combining classifiers' decisions or classification results in parallel [5,10]. When an input x is given to K classifiers (e.g., $C_1, C_2, \dots, C_K$) in parallel, a K-dimensional decision vector $D = \langle C_1(x) = M_1, C_2(x) = M_2, \dots, C_K(x) = M_K \rangle$ is observed, where a set of L decisions is denoted by $M = \{M_1, M_2, \dots, M_L\}$. The main task of combining multiple decisions is to determine a decision m which maximizes a posterior probability P^* which is $\max_P(m \in M \mid C_1(x) = M_1, C_2(x) = M_2, \dots, C_K(x) = M_K)$. However, it is well known that it is exponentially complex in storing and estimating, so for a small K, the probability distribution made by a set of the K-dimensional decision vectors becomes unmanageable. Thus, a new method has to compute such a high-order probability for combining

* Corresponding author. Currently working for Samsung Electronics Co., Ltd.; current address: Samhwa Bldg. 8F, 144-17 Samsung-Dong Kangnam-Ku, Seoul, South Korea 135-090.

[1] This work is partly supported by the Notepad Consortium and the Offline Consortium.

PII S0020-0190(97)00039-2

multiple decisions. That is, a $(K+1)$st order probability distribution should be estimated from samples. This idea can be expressed by the formulation, $P(m, C_1(x)=M_1, C_2(x)=M_2, \ldots, C_K(x)=M_K)$ where m is the decision variable.

Previous studies for estimating a high-order distribution in combining multiple decisions have assumed that classifiers' performances are conditionally independent of each other for a given input [12,8]. This is expressed by the formulation,

$$\begin{aligned} &P(m, C_1(x)=M_1, C_2(x)=M_2, \ldots, C_K(x)=M_K) \\ &\quad = P(m) P(C_1(x)=M_1, C_2(x)=M_2, \ldots, C_K(x)=M_K \mid m) \\ &\quad \approx P(m) \prod_{j=1}^{K} P(C_j(x)=M_j \mid m). \end{aligned}$$

However, Huang and Suen [3,4] assumed no independence in their proposed method, which was called Behavior-Knowledge Space (BKS). The former approach has the advantages of simple computation and small storage needs (i.e., $K \cdot L^2$), but a conditional independence assumption often does not hold for a real situation. On the contrary, the BKS method has the disadvantages of exponential computation and large storage needs (i.e., L^{K+1}) in theoretical analysis, and it also has the disadvantage of producing potentially high rejection rates due to unseen multiple decisions. But, the BKS method no longer assumes independence of classifiers.

As mentioned before, previous studies have been conducted for better combining multiple decisions, but most of them did not focus on dependencies among classifiers [6]. A dependency means the predictive capability of a classifier's decision to the other one's. Thus, kth-order dependency can be defined by the predictive capability of k classifiers' decisions to the others. The assumption that classifiers perform independently is not invulnerable, because classifiers tend to be statistically dependent on others. Therefore, in this paper, a new method is proposed to approximate optimally a high-order distribution with a product of kth-order distributions where $k \geqslant 1$, by using a systematic dependency-directed approach without an independence assumption. This method has smaller storage needs (i.e., $(K+1-k) \cdot L^{k+1}$) than the BKS method. And then, a new method is also presented to combine probabilistically multiple decisions with an optimal product set of kth-order dependencies, using a Bayesian formalism. The usefulness of the proposed methods is shown in experimental results obtained from the recognition of totally unconstrained on-line handwritten numerals and English characters.

2. Systematic dependency-directed approximation

When a high-order probability distribution is approximated, a criterion is needed to measure how close the approximation is to the actual distribution. Such a criterion, depending on the information theory model, was developed by Lewis in [9]. This is called as the measure of closeness in [1] or the measure of divergence in [7]. 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. For notation convenience, the authors will denote $C_j(x)=M_j$ in D by C_j. Without approximation, the $(K+1)$st-order probability distribution is converted into a product of low-order distributions, using the definition of chain rule:

$$P(C_1, C_2, \ldots, C_K, C_{K+1}) = P(C_1) P(C_2 \mid C_1) \cdots P(C_{K+1} \mid C_1, C_2, \ldots, C_K).$$

Chow and Liu [1] struggled to solve the optimization problem proposed by Lewis in [9] and to approximate optimally an nth-order binary variable distribution by a product of $(n-1)$ second-order component distributions. In their work, a method was presented to approximate optimally an nth-order discrete probability distribution by a product of the distributions of first-order tree dependencies, using the Maximum Weight Spanning Tree (MWST) algorithm of Kruskal. Because they focused on only first-order dependency, however, their method was not appropriate to consider high-order dependency. For example, it is such a case that a classifier mainly follows more than two classifiers.

In this paper, the authors propose a dependency-directed approach for approximating the $(K+1)$st-order probability distribution with an optimal product set of kth-order dependencies where $1 \leqslant k \leqslant K$, and a probabilistic combination method of multiple decisions using a Bayesian formalism. This new approach can be regarded as a natural extension of the first-order dependence tree proposed by Chow and Liu in [1]. The authors assume that the dependency can be stochastically determined by only observing classifiers' decisions obtained from samples. The order of dependency can be increased up to the higher for better approximation.

When first-order (i.e., $k=1$) dependency is considered, the approximate distribution is defined in terms of second-order distributions like the following expression:

$$P_a(C_1, C_2, \ldots, C_K, C_{K+1}) = \prod_{j=1}^{K+1} P\left(C_{n_j} \mid C_{n_{i(j)}}\right), \quad \text{where } 0 \leqslant i(j) < j, \tag{1}$$

such that C_{n_j} is conditioned on $C_{n_{i(j)}}$, and where $n_1, n_2, \ldots, n_K, n_{K+1}$ is a permutation of integers $1, 2, \ldots, K, K+1$. And $P(C_{n_j} \mid C_0)$ means $P(C_{n_j})$, by definition. By applying the dependence tree method by Chow and Liu [1] to the first-order directed approximation of the $(K+1)$st-order probability distribution P composed of a true decision C_{K+1} and multiple decisions $C_1, \ldots, C_K$, the authors can determine both the permutation of $n_1, n_2, \ldots, n_K, n_{K+1}$ and their conditioned permutation of $n_{i(1)}, n_{i(2)}, \ldots, n_{i(K)}, n_{i(K+1)}$ from the chosen optimal dependence tree. The details on the algorithm of finding the optimal dependence tree are referred in [1,11].

On the other hand, if $C_{n_{i(j)}}$ in the expression (1) is identical to all the C_{n_j}, that is, the C_j are assumed to be conditionally independent of each other for a given C_{K+1}, i.e., decision m, then the approximate distribution is defined in terms of second-order distributions like the following expression:

$$P_a(C_1, C_2, \ldots, C_K, C_{K+1}) = \prod_{j=1}^{K} P\left(C_j \mid C_{K+1}\right).$$

Such an approximation is regarded as a well known conditional independence assumption which can be defined as a particular case of first-order dependency approximations.

When second-order (i.e., $k=2$) dependency is considered, the approximate distribution is defined in terms of third-order distributions like the following expression:

$$P_a(C_1, C_2, \ldots, C_K, C_{K+1}) = \prod_{j=1}^{K+1} P\left(C_{n_j} \mid C_{n_{i2(j)}}, C_{n_{i1(j)}}\right), \quad \text{where } 0 \leqslant i2(j), i1(j) < j, \tag{2}$$

such that C_{n_j} is conditioned on both $C_{n_{i2(j)}}$ and $C_{n_{i1(j)}}$, and where $n_1, n_2, \ldots, n_K, n_{K+1}$ is a permutation of integers $1, 2, \ldots, K, K+1$. And $P(C_{n_j} \mid C_0, C_{n_{i1(j)}})$ means $P(C_{n_j}, C_{n_{i1(j)}})$, by definition. For notation convenience, the authors will drop the subscript n and denote, for example, C_{n_j} by C_j in subsequent discussions. The authors can also use the measure of closeness for approximating optimally distribution with a product of second-order dependencies like the following expressions:

$$\begin{aligned} I(P(C), P_a(C)) &= \sum_{C} P(C) \log \frac{P(C)}{P_a(C)} \\ &= \sum_{C} P(C) \log P(C) - \sum_{j=1}^{K+1} \sum_{C} P(C) \log P\left(C_j \mid C_{i2(j)}, C_{i1(j)}\right) \end{aligned}$$

$$= -H(C) - \sum_{j=1}^{K+1} \sum_{C} P(C) \log P(C_j)$$

$$- \sum_{\substack{j=1 \\ i2(j)\neq 0, i1(j)\neq 0}}^{K+1} \sum_{C} P(C) \log \frac{P(C_j \mid C_{i2(j)}, C_{i1(j)})}{P(C_j)}$$

$$= - \sum_{\substack{j=1 \\ i2(j)\neq 0, i1(j)\neq 0}}^{K+1} M(C_j;\, C_{i2(j)}, C_{i1(j)}) + \sum_{j=1}^{K+1} H(C_j) - H(C), \tag{3}$$

$$H(C) = -\sum_{C} P(C) \log P(C),$$

$$M(C_j;\, C_{i2(j)}, C_{i1(j)}) = \sum_{C_j, C_{i2(j)}, C_{i1(j)}} P(C_j, C_{i2(j)}, C_{i1(j)}) \log \frac{P(C_j \mid C_{i2(j)}, C_{i1(j)})}{P(C_j)}.$$

From the above expression (3), minimizing $I(P(C), P_a(C))$ is to maximize $\Sigma_{j-1}^{K+1} M(C_j;\, C_{i2(j)}, C_{i1(j)})$ which is the total sum of average mutual information (see [2]) satisfied with a given constraint, since the remaining terms (i.e., $\Pi_{j=1}^{K+1} H(C_j)$ and $H(C)$) are constant. Then, the next step is in how to identify an optimal set of second-order dependencies from all the permissible product sets. The process of identifying the optimal set of second-order dependencies can be algorithmically described as follows. This algorithm begins with one of the first-order dependencies as a constraint of the probability property, and ends with a set of optimal $(K-1)$ second-order dependencies which has maximum $\Sigma_{j=1}^{K+1} M(C_j;\, C_{i2(j)}, C_{i1(j)})$, using an exhaustive search.

Algorithm for second-order dependency

Input:

The set of s samples $S^1, S^2, \ldots, S^s$.

Output:

The optimal set of second-order dependencies identified as per the measure of closeness.

Method:

1. Estimate the second- and third-order marginals from the various samples.
2. Compute the weights $M(C_j;\, C_{i(j)})$ and $M(C_j;\, C_{i2(j)}, C_{i1(j)})$ for all pairs and triplets of classifiers from the samples.
3. Compute the maximum weight of first- and second-order dependencies and its optimally associated set.

```
for $n = 1$ to number of first-order dependencies do
  choose one of first-order dependencies as a constraint;
  compute the weight of permissible second-order dependencies according to the chosen first-order one;
  find the maximum weight and identify the set of its associated dependencies;
end
```

4. Determine a finally identified set as the optimal, according to the final maximum weight.

End of Algorithm

Example. The following are the values of the average mutual information computed for the second- and third-order dependencies from a fourth-order probability distribution composed of three decisions C_1, C_2, C_3 of three classifiers E_1, E_2, E_3 in experiments and a hypothesized decision variable C_4.

$$\begin{array}{lll}
M(C_4;\, C_1) = 2.217871; & M(C_4;\, C_2) = 2.227739; & M(C_4;\, C_3) = 2.272387; \\
M(C_1;\, C_2) = 2.185223; & M(C_1;\, C_3) = 2.207577; & M(C_2;\, C_3) = 2.238237; \\
M(C_4;\, C_1, C_2) = 2.263796; & M(C_1;\, C_4, C_2) = 2.221280; & M(C_2;\, C_4, C_1) = 2.231146; \\
M(C_4;\, C_1, C_3) = 2.284759; & M(C_1;\, C_4, C_3) = 2.219949; & M(C_3;\, C_4, C_1) = 2.274464; \\
M(C_4;\, C_2, C_3) = 2.276317; & M(C_2;\, C_4, C_3) = 2.242167; & M(C_3;\, C_4, C_2) = 2.286815; \\
M(C_1;\, C_2, C_3) = 2.211508; & M(C_2;\, C_1, C_3) = 2.242168; & M(C_3;\, C_1, C_2) = 2.264522;
\end{array}$$

The authors applied the conditional independence assumption to the fourth-order distribution and obtained the following first line result, and by applying the dependence tree method to the same distribution, the authors also obtained the following second line result. In the following results, * means optimal result.

$$\begin{array}{ll}
P_a(C_1, C_2, C_3, C_4) = P(C_1, C_4)P(C_2 | C_4)P(C_3 | C_4): & \sum M = 6.717997, \\
P_a(C_1, C_2, C_3, C_4) = P(C_1, C_4)P(C_3 | C_4)P(C_2 | C_3): & \sum M = 6.728495^*.
\end{array}$$

For the purpose of considering more available information, the authors applied the proposed algorithm for the second-order dependency to the same actual distribution and obtained the following results.

$$\begin{array}{ll}
P_a(C_1, C_2, C_3, C_4) = P(C_4, C_1)P(C_2 | C_4, C_1)P(C_3 | C_4, C_2): & \sum M = 6.735832, \\
P_a(C_1, C_2, C_3, C_4) = P(C_4, C_1)P(C_3 | C_4, C_1)P(C_2 | C_1, C_3): & \sum M = 6.734503, \\
P_a(C_1, C_2, C_3, C_4) = P(C_4, C_2)P(C_1 | C_4, C_2)P(C_3 | C_4, C_2): & \sum M = 6.735834^*, \\
P_a(C_1, C_2, C_3, C_4) = P(C_4, C_2)P(C_3 | C_4, C_2)P(C_1 | C_4, C_2): & \sum M = 6.735834^*, \\
P_a(C_1, C_2, C_3, C_4) = P(C_4, C_3)P(C_1 | C_4, C_3)P(C_2 | C_1, C_3): & \sum M = 6.734504, \\
P_a(C_1, C_2, C_3, C_4) = P(C_4, C_3)P(C_2 | C_4, C_3)P(C_1 | C_4, C_2): & \sum M = 6.735834^*, \\
P_a(C_1, C_2, C_3, C_4) = P(C_1, C_2)P(C_4 | C_1, C_2)P(C_3 | C_4, C_2): & \sum M = 6.735834^*, \\
P_a(C_1, C_2, C_3, C_4) = P(C_1, C_2)P(C_3 | C_1, C_2)P(C_4 | C_1, C_3): & \sum M = 6.734504, \\
P_a(C_1, C_2, C_3, C_4) = P(C_1, C_3)P(C_4 | C_1, C_3)P(C_2 | C_1, C_3): & \sum M = 6.734504, \\
P_a(C_1, C_2, C_3, C_4) = P(C_1, C_3)P(C_2 | C_1, C_3)P(C_4 | C_1, C_3): & \sum M = 6.734504, \\
P_a(C_1, C_2, C_3, C_4) = P(C_2, C_3)P(C_4 | C_2, C_3)P(C_1 | C_4, C_2): & \sum M = 6.735834^*, \\
P_a(C_1, C_2, C_3, C_4) = P(C_2, C_3)P(C_1 | C_2, C_3)P(C_4 | C_1, C_3): & \sum M = 6.734504.
\end{array}$$

From the above results, the authors can choose one of $P(C_4, C_2)P(C_1 | C_4, C_2)P(C_3 | C_4, C_2)$, $P(C_4, C_3)P(C_2 | C_4, C_3)P(C_1 | C_4, C_2)$, $P(C_1, C_2)P(C_4 | C_1, C_2)P(C_3 | C_4, C_2)$, and $P(C_2, C_3) \cdot P(C_4 | C_2, C_3)P(C_1 | C_4, C_2)$, as an optimal product set for the second-order dependency, because they have the maximum weight of second-order average mutual information.

On the other hand, if $C_{n_{i1(j)}}$ in the expression (2) is the same for all the C_{n_i}, that is, C_{n_j} is assumed to be conditionally dependent on $C_{n_{i2(j)}}$ for the given C_{K+1}, then the approximate distribution is defined in terms of third-order distributions like the following expression:

$$P_a(C_1, C_2, \dots, C_K, C_{K+1}) = \prod_{j=1}^{K} P\left(C_{n_j} | C_{n_{i2(j)}}, C_{n_{K+1}}\right), \quad \text{where } 0 \leqslant i2(j) < j,$$

such that C_{n_j} is conditioned on both $C_{n_{i2(j)}}$ and $C_{n_{K+1}}$, and that $n_1, n_2, \dots, n_K$ is a permutation of integers $1, 2, \dots, K$. And, $P(C_{n_j} | C_0, C_{K+1})$ means $P(C_{n_j}, C_{K+1})$, by definition. Such an approximation is called a conditional first-order dependency approximation which can be defined as a particular case of second-order dependency approximations. From the previous example, the identified optimal product set by conditional first-order dependency is $P(C_4, C_2)P(C_1 | C_4, C_2)P(C_3 | C_4, C_2)$ or $P(C_4, C_3)P(C_2 | C_4, C_3)P(C_1 | C_4, C_2)$.

By using the new systematic dependency-directed approximation, the order of dependency to be considered can be easily extended to the kth-order for approximating probability distributions under permissible resource requirements. The optimal kth-order dependency product set consists of a first-order dependency, a second-order dependency, ..., a $(k-1)$st-order dependency, and some (i.e., $K-k$) kth-order dependencies which have maximum $\sum_{j=1}^{K+1} M(C_{n_j} | C_{n_{ik(j)}}, \dots, C_{n_{i2(j)}}, C_{n_{i1(j)}})$.

$$P_a(C_1, C_2, \dots, C_K, C_{K+1}) = \prod_{j=1}^{K+1} P\left(C_{n_j} | C_{n_{ik(j)}}, \dots, C_{n_{i2(j)}}, C_{n_{i1(j)}}\right),$$

where $0 \leqslant ik(j), \dots, i2(j), i1(j) < j$.

3. Probabilistic combination of multiple decisions

Using the approximate distribution obtained from an optimal set of kth-order dependencies, the authors can apply them to a Bayesian formalism for probabilistically combining multiple decisions. For each hypothesized decision M_i, by using the Bayesian theorem and an optimal product set of first-order dependencies, and by letting a decision term $M_i \in M$ be denoted by $C_{K+1}(x) = M_{K+1}$, the authors have the following formula:

$$\begin{aligned} Bel(M_i) &= P(M_i \in M | C_1(x) = M_1, \dots, C_K(x) = M_K) \\ &= \frac{P(M_i \in M, C_1(x) = M_1, \dots, C_K(x) = M_K)}{P(C_1(x) = M_1, \dots, C_K(x) = M_K)} \\ &= \frac{\prod_{j=1}^{K+1} P\left(C_{n_j}(x) = M_{n_j} | C_{n_{i(j)}}(x) = M_{n_{i(j)}}\right)}{P(C_1(x) = M_1, \dots, C_K(x) = M_K)} \\ &\approx \eta \left(\prod_{\substack{j=1 \\ n_j = K+1 \text{ or } n_{i(j)} = K+1}}^{K+1} P\left(C_{n_j}(x) = M_{n_j} | C_{n_{i(j)}}(x) = M_{n_{i(j)}}\right) \right), \end{aligned}$$

with η as a constant that ensures that $\sum_{i=1}^{L} Bel(M_i) = 1$. And, $n_1, n_2, \dots, n_K, n_{K+1}$ is a permutation of integers $1, 2, \dots, K, K+1$. Depending on these $Bel(M_i)$ values computed by combining multiple decisions, the authors choose a maximized posterior probability, and a combined decision is determined as a decision M_i, according to the decision rule $E(D)$ given below:

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

If an optimal set of second-order dependencies is used for the higher-order dependency, then the combining formula is expressed as follows:

$$\begin{aligned} Bel(M_i) &= P(M_i \in M | C_1(x) = M_1, \dots, C_K(x) = M_K) \\ &= \frac{P(M_i \in M, C_1(x) = M_1, \dots, C_K(x) = M_K)}{P(C_1(x) = M_1, \dots, C_K(x) = M_K)} \end{aligned}$$

$$= \frac{\prod_{j=1}^{K+1} P\left(C_{n_j}(x) = M_{n_j} \mid C_{n_{i2(j)}}(x) = M_{n_{i2(j)}}, C_{n_{i1(j)}}(x) = M_{n_{i1(j)}}\right)}{P\left(C_1(x) = M_1, \ldots, C_K(x) = M_K\right)}$$

$$\approx \eta \left(\prod_{\substack{j=1 \\ n_j = K+1 \text{ or } n_{i(j)} = K+1}}^{K+1} P\left(C_{n_j}(x) = M_{n_j} \mid C_{n_{i2(j)}}(x) = M_{n_{i2(j)}}, C_{n_{i1(j)}}(x) = M_{n_{i1(j)}}\right) \right),$$

with η as a constant that ensures that $\sum_{i=1}^{L} Bel(M_i) = 1$. And, $n_1, n_2, \ldots, n_K, n_{K+1}$ is a permutation of integers $1, 2, \ldots, K, K+1$. The authors can also apply the above decision rule $E(D)$ to combining multiple decisions on the basis of second-order dependencies.

4. Experiments

The authors have three classifiers *E1*, *E2*, *E3* for recognizing totally unconstrained on-line handwritten numerals and English characters. These classifiers are the components of a *Base* system. For demonstrating the effectiveness of the dependency-directed approach, some multiple classifier systems were built by adding the highly dependent classifier created by faking one of the component classifiers to the *Base* system. For example, an *E1 faked* system consists of the component classifiers and a ditto of an *E1* classifier. In order to identify an optimal product set of kth-order dependency, 4088 numerals written by 13 subjects, and 3749 lowercases and 2464 uppercases by 19 subjects were used as a training data set. And, as a test data set, 988 numerals written by 10 subjects, and 1684 lowercases and 1169 uppercases by 9 subjects were used. The subjects of the training data were different from those of the test data in each application area. A *reject* recognition result sample of a classifier was excluded in identifying the optimal product set and in combining multiple decisions. In other words, only valid recognition results were considered. The recognition rates of the component classifiers on the test data are shown in Table 1.

From the experimental results (see Table 2) on numerals data, the kth-order dependency based combination methods (i.e., 1st-order Dep., Cond. 1st-order Dep., and 2nd-order Dep.) showed better performance than a conditional independence assumption based combination method (i.e., Cond. Indep.) and the BKS method. The best recognition rates on lowercases data (see Table 3) were obtained by a conditional first-order dependency based combination method (i.e., Cond. 1st-order Dep.) over all multiple classifier systems. The best recognition rates on uppercases data (see Table 4) were obtained by the conditional first-order dependency based combination method over all multiple classifier systems except an *E3 faked* system. In case of the *E3 faked* system, the best recognition rate was obtained by a second-order dependency based combination method (i.e., 2nd-order Dep.).

When higher-order dependency was considered, the recognition rates were increased, as shown in the tables (see Tables 2–4). However, the computational complexity in identifying the optimal product set would increase, too. In the conditional independence assumption based Bayesian method, identifying the optimal product set has computational complexity O(1) regardless of the number of classifiers, since that is unnecessary. In case of

Table 1
Recognition rates (%) of classifiers on the test data

Classifier	Numerals		Lowercases		Uppercases	
	1st	rej.	1st	rej.	1st	rej.
E1	93.09	0.61	78.92	1.25	88.37	1.80
E2	92.28	0.10	82.30	0.53	91.02	0.43
E3	94.39	1.32	86.05	2.73	87.51	4.23

Table 2
Recognition rates (%) of multiple classifier systems on numerals data

Comb. Method	*Base*		*E1 faked*		*E2 faked*		*E3 faked*	
	1st	rej.	1st	rej.	1st	rej.	1st	rej.
BKS	94.09	2.61	94.09	2.61	94.09	2.61	94.09	2.61
Cond. Indep.	94.99	0.10	94.59	0.10	95.29	0.10	94.69	0.10
1st-order Dep.	95.49	0.10	95.49	0.10	95.49	0.10	95.49	0.10
Cond. 1st-order Dep.	95.59	0.10	95.59	0.10	95.59	0.10	95.59	0.10
2nd-order Dep.	95.59	0.10	95.59	0.10	95.59	0.10	95.59	0.10

Table 3
Recognition rates (%) of multiple classifier systems on lowercases data

Comb. Method	*Base*		*E1 faked*		*E2 faked*		*E3 faked*	
	1st	rej.	1st	rej.	1st	rej.	1st	rej.
BKS	86.05	7.36	86.05	7.36	86.05	7.36	86.05	7.36
Cond. Indep.	87.00	0.42	86.22	0.42	87.29	0.42	87.23	0.42
1st-order Dept.	86.46	0.42	86.46	0.42	86.46	0.42	86.46	0.42
Cond. 1st-order Dep.	87.35	0.42	87.71	0.42	87.65	0.42	87.35	0.42
2nd-order Dep.	86.64	0.42	86.64	0.42	86.64	0.42	86.64	0.42

first-order dependency, the computational complexity needed for identifying the optimal product set is $O(N \cdot \log N)$ where N is the total number (i.e., $\frac{1}{2}(K+1)K$) of edges in the graph G. In addition, the computational complexity for identifying the optimal product set of second-order dependencies is $O(N \cdot N)$ where the former N is the same as that of first-order dependency and the latter N is the total number (i.e., $\sum_{d=2}^{K}(K+1-d)C(d,2))$ of permissible second-order dependencies according to the chosen first-order dependency, where $C(d,2)$ is a combination function in statistics. Although it is somewhat complex to consider the higher-order dependency, it is better to consider the higher-order dependency, since the recognition rates might be increased and the complex identification computation is conducted only once at training stage. But, it is not always guaranteed that considering the higher-order dependency leads to high performance.

In summary, the performances of the BKS method were unchanged even though the highly dependent classifier was added to the *Base* system. The recognition rates obtained by the higher-order dependency based combination method were higher than those by the lower-order dependency based one, in most cases. The low recognition rates by the second-order dependency based combination method on lowercases and uppercases data were caused by the lack of a large enough and well representative training data set. Incorporating the kh-order dependency into a Bayesian formalism contributed to improvement on the performance of combining multiple classifiers, especially when the highly dependent classifier was included. The difference of recognition rates

Table 4
Recognition rates (%) of multiple classifier systems on uppercases data

Comb. Method	*Base*		*E1 faked*		*E2 faked*		*E3 faked*	
	1st	rej.	1st	rej.	1st	rej.	1st	rej.
BKS	89.14	6.24	89.14	6.24	89.14	6.24	89.14	6.24
Cond. Indep.	90.59	0.43	90.59	0.43	91.10	0.43	89.22	0.43
1st-order Dep.	88.79	0.43	88.79	0.43	88.79	0.43	88.79	0.43
Cond. 1st-order Dep.	91.27	0.43	91.10	0.43	91.45	0.43	89.31	0.43
2nd-order Dep.	91.27	0.43	90.16	0.43	90.16	0.43	91.27	0.43

between the conditional independence assumption based combination method and the best kth-order dependency based combination method was usually statistically significant by t-test at significance level 0.01.

5. Summary

The first task of this paper is to present a new method which approximates optimally a high-order probability distribution by the higher-order dependency including a first-order one for probabilistically combining multiple decisions. This is regarded as the extended works of Chow and Liu. And, the second task is to provide a probabilistic combination of multiple decisions based on the kth-order dependency, using a Bayesian formalism without an independence assumption. The experimental results show that the higher-order dependency should be also considered in combining multiple decisions, and the kth-order dependency based combination method works very well if an optimal product set is constructed by a representative training data set.

References

[1] C.K. Chow and C.N. Liu, Approximating discrete probability distributions with dependence trees, *IEEE Trans. Inform. Theory* 14 (3) (1968) 462–467.

[2] R.G. Gallager, *Information Theory and Reliable Communication* (John Wiley and Sons, New York, 1968).

[3] Y.S. Huang and C.Y. Suen, An optimal method of combining multiple classifiers for unconstrained handwritten numeral recognition, in: *Proc. 3rd Internat. Workshop on Frontiers in Handwriting Recognition* (1993) 11–20.

[4] Y.S. Huang and C.Y. Suen, A method of combining multiple experts for the recognition of unconstrained handwritten numerals, *IEEE Trans. Pattern Analysis Machine Intelligence* 17 (1) (1995) 90–94.

[5] J.J. Hull, A. Commike and T.-K. Ho, Multiple algorithm for handwritten character recognition, in: *Proc. 1st Internat. Workshop on Frontiers in Handwriting Recognition* (1990) 117–129.

[6] H.-J. Kang and J.H. Kim, Dependency relationship based decision combination in multiple classifier systems, in: *Proc. 14th Internat. Joint Conf. on Artificial Intelligence*, Vol. 2 (1995) 1130–1136.

[7] H.H. Ku and S. Kullback, Approximating discrete probability distributions, *IEEE Trans. Inform. Theory* 15 (4) (1969) 444–447.

[8] D.-S. Lee and S.N. Srihari, Handprinted digit recognition: A comparison of algorithms, in: *Proc. 3rd Internat. Workshop on Frontiers in Handwriting Recognition* (1993) 153–162.

[9] P.M. Lewis, Approximating probability distributions to reduce storage requirement, *Inform. and Control* 2 (1959) 214–225.

[10] C.Y. Suen, C. Nadal, T.A. Mai, R. Legault and L. Lam, Recognition of totally unconstrained handwritten numerals based on the concept of multiple experts, in: *Proc. 1st Internat. Workshop on Frontiers in Handwriting Recognition* (1990) 131–143.

[11] R.S. Valiveti and B.J. Oommen, On using the chi-squared metric for determining stochastic dependence, *Pattern Recognition* 25 (11) (1992) 1389–1400.

[12] L. Xu, A. Krzyzak and C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, *IEEE Trans. Systems, Man, Cybernet.* 22 (3) (1992) 418–435.