Discriminative Training of Bayesian Chow-Liu Multinet Classifiers

Kaizhu Huang, Irwin King and Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong SAR, China
Email: {kzhuang, king, lyu}@cse.cuhk.edu.hk

Abstract—**Discriminative classifiers such as Support Vector Machines directly learn a discriminant function or a posterior probability model to perform classification. On the other hand, generative classifiers often learn a joint probability model and then use Bayes rules to construct a posterior classifier from this model. In general, generative classifiers are not as accurate as discriminant classifiers. However generative classifiers provide a principled way to handle the missing information problems, which discriminant classifiers cannot easily deal with. To achieve good performances in various classification tasks, it is better to combine these two strategies. In this paper, we develop a novel method to iteratively train a kind of generative Bayesian classifier: Bayesian Chow-Liu Multinet classifier in a discriminative way. Different with the traditional Bayesian Multinet classifiers, our discriminative method adds into the optimization function a penalty item, which represents the divergence between classes. Iterative optimization on this optimization function tries to approximate the dataset as accurately as possible. At the same time, it also tries to make the divergence between classes as big as possible. We state the theoretical justification, outline of the algorithm and also perform a series of experiments to demonstrate the advantages of our method. The experiments results are promising and encouraging.**

I. Introduction

Generative classifiers have showed their advantages to deal with missing information problems in many classification tasks, even though their overall performances are not as good as discriminative classifiers such as Support Vector Machines [18]. As one of competitive generative models, Bayesian Multinet classifiers [8], [6], [5], [10] separately use the belief network to model the joint probability of the data as accurately as possible for each class and then apply Bayes rule to build up a posterior probability classifier. This kind of framework to construct classifiers seems to be incomplete since this construction procedure actually discards some important discriminative information for classification. With no consideration of the other data with different class labels, this method only tries to approximate the information in each subdataset. On the other hand, discriminative classifiers preserve this discriminative information well by directly constructing a classifier among all the data. Thus for Bayesian Multinet classifiers, it is not enough that the generative model can approximate the data accurately. It should be also discriminative enough to separate this class from other classes.

One of the remedy method is to directly learn a posterior probability model rather than a joint probability model. However in the framework of Bayesian Multinet, this kind of approaches are often computationally hard to perform the optimization. A typical example is the Bayesian Chow-Liu tree Multinet classifier [5][1]. When directly optimizing the posterior probability, the optimization is transformed into a conditional log likelihood, which does not decompose over the structure of the network as in the original Chow-Liu tree. The inability of decomposition make the learning difficult to perform [5].

In this paper, beginning with modelling a Bayesian Chow-Liu tree Multinet over the pre-classified datasets, we provide a discriminative way to iteratively train this generative classifier. On one hand, our novel method tries to build up a tree probabilistic model to approximate the data as accurately as possible for each class . On the other hand, our method also tries to make the divergence among the models for differen classes as far as possible, which will benefit the classification greatly. Furthermore, What's the most important is that in each iteration, our method remains a modified version of Chow-Liu tree problem, which can be easily optimized in polynomial time. Our proposed method only demonstrates a two-category classification task, further work will be done to extend the two-category classification method into multi-category one.

This paper is organized as follows. In the next section, we present a short review on the related work. Then in Section III, we introduce the background knowledge for this paper such as the Chow-Liu tree and the Bayesian Multinet classifier. In Section IV, we describe our discriminative training framework in detail. In Section V, we demonstrate the advantages of our methods in a real world dataset. Finally, in Section VI, we conclude this paper with remarks.

II. Related work

Combining generative classifiers and discriminative classifiers has been one of an active topics in machine learning. A lot of work [1], [7], [17], [2] has been done in this area. However nearly all of these methods are designed for Gaussian Mixture Model (GMM) [13] and Hidden Markov Model (HMM) [15]. Since the structures of these models are often fixed, optimization only on parameters associated with the structure will be a relatively easy job. By contrast, our discriminative approaches are developed for Bayesian Multinet tree classifiers, where the structure is non-fixed in the tree family. Copying the techniques in HMM and GMM cannot solve the problem. On the other hand, Jaakkola et. al. developed a method to explore generative models from discriminative classifiers [16]. Different with this approach,

[1]In [5], Bayesian Chow-Liu tree Multinet is called directly Bayesian Multinet.

0-7803-7898-9/03/$17.00 ©2003 IEEE

our method performs a reverse way to use discriminative information in generative classifiers.

III. Background

A. Bayesian Multinet Classifier

Given a pre-classified dataset $D = \{\{x^1, C^1\}, \ldots, \{x^N, C^N\}\}$, where $x^i = (A_1^i, A_2^i, \ldots, A_n^i)$ is an n-dimension vector, $A_1, A_2, \ldots, A_n$ are called variables or attributes and C^i is the class label, either "C_1" or "C_2". The dataset is first partitioned into several sub-datasets by the class label. Then in each sub-dataset with the uniform class label C_k, $k = 1$ or 2, a Bayesian network B_k is used to model the joint probability $P_{B_k}(A_1, A_2, \ldots, A_n)$. The Bayesian network B_k is called a local network for the class label C_k. The local network structure and parameters $\{M = \{B_1, B_2\}, M_\theta = \{P_{B_1}, P_{B_2}\}\}$ are searched by minimizing a score function, often the cross-entropy or the Kullback divergence between the estimated distribution based on B_k and the empirical distribution over the sub-dataset, which is defined as:

$$KL(P_{B_k}, \hat{P}_k) = \sum_{\{a_1,\ldots,a_n\}} \hat{P}_k(a_1, \ldots, a_n) \log \frac{\hat{P}_k(a_1, \ldots, a_n)}{P_{B_k}(a_1, \ldots, a_n)}, \quad (1)$$

where $\hat{P}_k$ refers to the empirical distribution for the sub-dataset with class label C_k and $\{a_1, a_2, \ldots, a_n\}$ is a short form of $\{A_1 = a_1, A_2 = a_2, \ldots, A_n = a_n\}$.

The set of local networks combined with the prior information on the class variable C, $P(C)$ is called Bayesian Multinet. When used in classification tasks, the class label c assigned for a new data $\{a_1, a_2, \ldots, a_n\}$ is given by the following formula:

$$c = \arg\max_{C_k} P(C = C_k) P_{B_k}(a_1, a_2, \ldots, a_n) \quad (2)$$

The Bayesian Multinet classifier for the two-category classification problem can be defined as:

Definition 1: Bayesian Multinet classifier is a kind of classifier, which first finds a Bayesian model by minimizing Eq. (1) for each sub-dataset: C_k and then uses Eq. (2) to perform classification.

B. Chow-Liu Tree

Chow-Liu tree [3], [9] is a kind of Bayesian model, which assumes a tree dependence relationship exists among the attribute set $\{A_1, A_2, \ldots, A_n\}$. An optimal dependence tree structure, which minimizes the Kullback divergence defined by Eq. (1), can be obtained by the following Chow-Liu Tree algorithm:

1. Calculate $I(A_i, A_j)$ between each pair of attributes, $i \neq j$, where $I(A_i, A_j) = \sum_{\{a_i,a_j\}} \hat{P}(a_i, a_j) \log \frac{\hat{P}(a_i,a_j)}{\hat{P}(a_i)\hat{P}(a_j)}$ is the mutual information, $\hat{P}(\cdot)$ is the empirical distribution of the dataset, $\{a_i, a_j\}$ is a short form for $\{A_i = a_i, A_j = a_j\}$.
2. Construct a Maximum Weight Spanning tree among the attribute set $\{A_1, A_2, \ldots, A_n\}$, each attribute corresponds to a vertex, where the weight between two vertexes is given by the mutual information defined in the Step 1.

A remarkable characteristic of the Chow-Liu tree algorithm is that it can optimize the structure M and parameters M_θ associated with M at the same time in a polynomial cost. According to the Chow-Liu tree structure obtained from the above algorithm, the joint probability among the vertexes, i.e., the attributes can be decomposed into a product of several subitems:

$$P(A_1, A_2, \ldots, A_n) = \prod_{j=1}^{n} \hat{P}(A_j | pa(A_j)), \quad (3)$$

where $pa(A_j)$ means the parent vertex of vertex A_j. Each subitem $\hat{P}(A_j|pa(A_j))$ can be reliably estimated based on the empirical distribution.

C. Bayesian Chow-Liu Tree Multinet Classifer

A Bayesian Chow-Liu tree Multinet classifier is such a Bayesian Multinet classifier, which applies Chow-Liu tree alogrithm to model the joint probability in each sub-dataset with the uniform class label, and then uses Eq. (2) to perform classification. This kind of classifier is demonstrated to perform rather well in comparison to the state-of-the-art decision tree learner C4.5 [5].

IV. Discriminative Bayesian Chow-Liu Tree Multinet Classifer

The optimization function to construct a two-category Bayesian Chow-Liu tree Multinet classifier can be written as:

$$\{B_1, B_2, P_{B_1}^*, P_{B_2}^*\} = \arg\min_{\{B_1,B_2,P_{B_1},P_{B_2}\}} \{KL(P_{B_1}, \hat{P}_1) + KL(P_{B_2}, \hat{P}_2)\}. \quad (4)$$

This optimization function only takes into account the inner divergence information inside each sub-dataset, while it throws out the class divergence information between the sub-datasets, which is very important for constructing accurate classifiers. Motivated from this point, we propose the following optimization function to preserve those discriminant information:

$$\{B_1, B_2, P_{B_1}^*, P_{B_2}^*\} = \arg\min_{\{B_1,B_2,P_{B_1},P_{B_2}\}} \{KL(P_{B_1}, \hat{P}_1) + KL(P_{B_2}, \hat{P}_2) - W \cdot Div(P_{B_1}, P_{B_2})\}, \quad (5)$$

where W is a penalty parameter, $Div(\cdot)$ refers to the measure of the divergence between P_{B_1}, P_{B_2}. $\hat{P}_1$, $\hat{P}_2$ represent the empirical distribution respectively for sub-dataset 1 and sub-dataset 2. To minimize the function (5), the inner divergence in each class, represented by the first two parts need to be as small as possible while the class divergence, represented by the last part needs to be as large as possible. However, the disadvantage caused by adding the interactive item is that we

cannot easily learn the structure and the associated parameters respectively as in Bayesian Multinet classifiers, since now these items are interactive.

To solve this problem, we propose a novel iterative scheme to perform the optimization. By this iterative process, in each step, we maintain the merit of the Bayesian Chow-Liu tree Multinet, i.e., the structure and the associated parameters can be optimized at the same time through a modified Chow-Liu tree approach. In the following, we propose our discriminant training procedure in detail.

A. Framework of Iterative Optimization on Discriminant Bayesian Chow-Liu Tree Multinet

To perform the optimization in Eq. (5), we first define the divergence metric. Different divergence metrics to distinguish two distributions can be used here. In this paper, we use Kullback divergence Eq. (6) or the similar metric Eq. (7).

$$Div_1(P_{B_1}, P_{B_2}) = \sum_{\{a_1,\dots,a_n\}} P_{B_1}(a_1,\dots,a_n) \log \frac{P_{B_1}(a_1,\dots,a_n)}{P_{B_2}(a_1,\dots,a_n)}, \quad (6)$$

$$Div_2(P_{B_1}, P_{B_2}) = \sum_{\{a_1,\dots,a_n\}} P_{B_2}(a_1,\dots,a_n) \log \frac{P_{B_2}(a_1,\dots,a_n)}{P_{B_1}(a_1,\dots,a_n)}. \quad (7)$$

We combine these two similar divergence metrics with the optimization function. Thus we obtain two optimization functions named DKL_1 and DKL_2.

$$DKL_{1(2)} = KL(P_{B_1}, \hat{P}_1) + KL(P_{B_2}, \hat{P}_2) - W \cdot Div_{1(2)}(P_{B_1}, P_{B_2}) \quad (8)$$

The optimal structure B_1, B_2 and the associated optimal parameter P_{B_1}, P_{B_2} can be obtained by minimizing one of these two functions, for example:

$$\{B_1, B_2, P^*_{B_1}, P^*_{B_2}\} = \arg\min_{\{B_1,B_2,P_{B_1},P_{B_2}\}} DKL_1$$

$$= \arg\min_{\{B_1,B_2,P_{B_1},P_{B_2}\}} \Big(\sum_{\{a_1,\dots,a_n\}} \hat{P}_2 \log \frac{\hat{P}_2}{P_{B_2}} - W \sum_{\{a_1,\dots,a_n\}} P_{B_1} \log \frac{P_{B_1}}{P_{B_2}} + KL(P_{B_1}, \hat{P}_1) \Big) \quad (9)$$

In the above equation, $\hat{P}_1$, $\hat{P}_2$ are constant for each instance $\{a_1,\dots,a_n\}$, thus, it can be further written as:

$$\{B_1, B_2, P^*_{B_1}, P^*_{B_2}\} = \arg\min_{\{P_{B_1},P_{B_2}\}} DKL_1$$

$$= \arg\min_{\{B_1,B_2,P_{B_1},P_{B_2}\}} \Big(- \sum_{\{a_1,\dots,a_n\}} ((\hat{P}_2 - W \cdot P_{B_1}) \log P_{B_2}) - W \sum_{\{a_1,\dots,a_n\}} P_{B_1} \log P_{B_1} + KL(P_{B_1}, \hat{P}_1) \Big) \quad (10)$$

It will be hard to directly perform optimization on the structures and the parameters B_1, P_{B_1} and B_2,P_{B_2} in Eq. (10). However if we fix B_1, P_{B_1}, Eq. (10) will be changed into the following optimization on B_2, P_{B_2}.

$$\{B_2, P^*_{B_2}\} = \arg\min_{\{B_2,P_{B_2}\}} DKL_1$$

$$= \arg\min_{\{P_{B_2}\}} \Big(- \sum_{\{a_1,\dots,a_n\}} ((\hat{P}_2 - W \cdot P_{B_1}) \log P_{B_2}) \Big) \quad (11)$$

When applied the tree dependence assumption on the variables set, Eq. (11) can be written into the following decomposed form:

$$\max_t \Big\{ \sum_{i=1}^{n} \max_{P|t} \sum_{\{a_i, a_{pa(i)}\}} [(\hat{P}_2(a_i, a_{pa(i)}) - W \cdot P_{B_1}(a_i, a_{pa(i)})) \log P_{B_2}(a_i|a_{pa(i)})] \Big\}, \quad (12)$$

where $\{a_i, a_{pa(i)}\}$ is the short form of $\{A_i = a_i, Pa(A_i) = a_{pa(i)}\}$. This formula can be optimized by a modified Chow-Liu tree algorithm under the constraint that parameters $P_{B_2}(a_i, a_{pa(i)}) \geq \xi_2$ and $\sum_{\{a_i,a_{pa(i)}\}} P_{B_2}(a_i, a_{pa(i)}) = 1$. ξ_2 is a positive constant close to zero. [2] The detailed optimization process can be seen in the Appendix. We just directly give the optimization result. The structure can be obtained by a Maximum Weight Spanning Tree algorithm and the associated parameters are obtained as follows:

$$P^*_{B_2}(a_i, a_j) = \xi_2, \quad if \quad \hat{P}_2(a_i, a_j) - W \cdot P_{B_1}(a_i, a_j) \leq 0, i \neq j; \quad (13)$$

$$P^*_{B_2}(a_i, a_j) = (\hat{P}_2(a_i, a_j) - W \cdot P_{B_1}(a_i, a_j))/Z, \quad if \quad \hat{P}_2(a_i, a_j) - W \cdot P_{B_1}(a_i, a_j) > 0, i \neq j; \quad (14)$$

where Z is a normalization factor, which is used to guarantee $\sum_{\{a_i,a_j\}} P^*_{B_2}(a_i, a_j) = 1$; $P_{B_1}(a_i, a_j)$ and $P_{B_2}(a_i, a_j)$ are the short forms of $P_{B_1}(A_i = a_i, A_j = a_j)$ and $P_{B_2}(A_i = a_i, A_j = a_j)$ respectively.

Similarly, when fixing the P_{B_2}, we can easily minimize the DKL_2 to find the parameter P_{B_1}. Motivated from these findings, we develop an iterative algorithm to perform optimization between two functions: DKL_1 and DKL_2. As Fig. 1 in each iteration i, we have two steps.

1. Fix B_1, P_{B_1} to $B_1^{i-1}, P_{B_1}^{i-1}$, find $B_2^i, P_{B_2}^i$ to minimize DKL_1
2. Fix B_2, P_{B_2} to $B_2^i, P_{B_2}^i$, find $B_1^i, P_{B_1}^i$ to minimize DKL_2

We call this iterative optimization algorithm as Iterative Discriminative Bayesian Multinet algorithm (IDBM). We outline this algorithm in Algorithm 1 and illustrate the algorithm.

In our experiments, we empirically demonstrate our iterative method converges rapidly in just several steps. The theoretic analysis on the convergence performance will be conducted as one of our future work.

[2] None of the decomposed probabilities can be zero. Otherwise, the restored joint probability as Eq. (3)will be always zero whatever the other decomposed probabilities are.

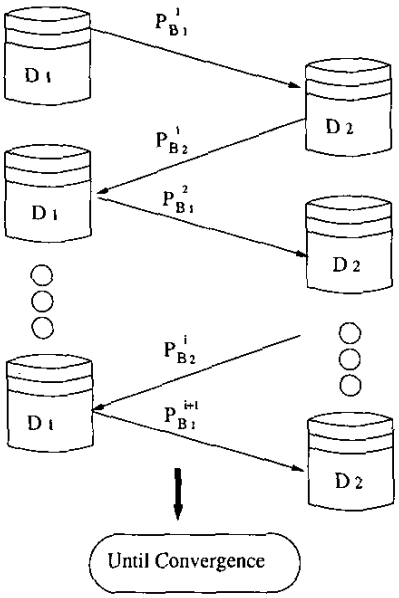

Fig. 1. Iterative Optimization Procedure. D_1 and D_2 are the sub-dataset with class label C_0 and C_1 respectively.

Algorithm IDBM(D)

input(*pre-classified Dataset* $D = \{x^1, x^2, \ldots, x^N\}$,$W$)

Initialization: $\{B_1, P_{B_1}\} = \{B_1^0, P_{B_1}^0\}$, $i = 1$. Partition D into two sub-datasets D_1 and D_2 by class;

repeat

  if (i *mod* $2 == 1$) **then**

    $\{B_1^i, P_{B_1}^i\} \leftarrow \{B_1^{i-1}, P_{B_1}^{i-1}\}$;

    Find $B_2^i, P_{B_2}^i$ by minimizing DKL_2;

  else

    $\{B_2^i, P_{B_2}^i\} \leftarrow \{B_2^{i-1}, P_{B_2}^{i-1}\}$;

    Find $B_1^i, P_{B_1}^i$ by minimizing DKL_1;

  end

  $i = i + 1$;

until *convergence*;

output($B_1, B_2, P_{B_1}, P_{B_2}$)

Algorithm 1: Iterative Bayesian Multinet Optimization Algorithm

V. Experiments

To evaluate the performance of our discriminative Chow-Liu tree (DCLT) Bayesian Multinet Classifier, we implement our algorithm on the Hepatitis database from UCI Machine learning Repository [14]. This dataset consists of 155 instances, 19 attributes and two classes "die" or "live". For continuously-value attributes, we use an equal-interval method to quantize them into discrete values. We compare our methods with Chow-Liu tree (CLT) Bayesian Multinet classifier and the Naive Bayesian (NB) Classifier [4], [12], which is a kind of competitive classifier as well. We set the penalty parameter W to 0.2 and we use the five-fold cross validation (CV5) [11] method to test the performance of these methods. The recognition results are described in Table I. It can be observed that DCLT performs significantly better than CLT and NB in this dataset, which shows that incorporating discriminative information will greatly benefit the classifier's performance. In our experiments, it is also interesting that the iterative process rapidly converges in several steps . In our five-fold cross validation experiments, all of the five times experiments converge well. We plot out one of the convergence curve in Figure 2. This phenomena deserves our further theoretic exploration.

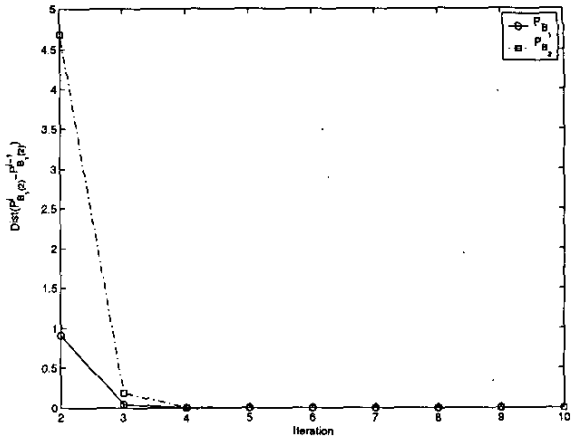

Fig. 2. Convergence curve for hepatitis dataset. The dash line is the convergence curve for P_{B_1}. The solid line is the convergence curve for P_{B_2}. The x-axis represents the iteration. And the y-axis represents the Euclidean distance between the current value and previous value for each parameter vector.

TABLE I

Recognition Rate

Database	NB	CLT	DCLT
Hepatitis(%)	84.95	86.03	89.25

VI. Conclusion

In this paper, beginning with generative modelling over the pre-classified datasets, we provide a discriminative way to iteratively train the Bayesian Chow-Liu tree Multinet classifiers. On one hand our novel approach approximates the dataset as accurately as possible. On the other hand, our approach also tries to discriminate the divergence between different classes. This discriminative characteristic will greatly benefit the classification. Even though, we cannot prove the convergence of our methods in theory for the time being, our experiments show that this iterative approach will converge rapidly. As one of our main research direction, we will focus on the theoretic exploration on the convergence property of our method in the near future.

ACKNOWLEDGMENT

This research is supported fully by grants from the Hong Kong's Research Grants Council (RGC) under CUHK 4407/99E and CUHK CUHK4360/02E.

APPENDIX

Assume $l = \sum_{\{a_1,\ldots,a_n\}}(\hat{P}_2 - W \cdot P_{B_1}) \log P_{B_2}$, under a tree dependence structure t among the variables, this formula can be decomposed as:

$$\sum_{i=1}^{n} \sum_{\{a_i, a_{pa(i)}\}} [\hat{P}_2(a_i, a_{pa(i)}) - W \cdot P_{B_1}(a_i, a_{pa(i)})] \log(P_{B_2}(a_i|a_{pa(i)}) \quad (15)$$

Let:

$$P^d(a_i, a_{pa(i)}) = \hat{P}_2(a_i, a_{pa(i)}) - W \cdot P_{B_1}(a_i, a_{pa(i)}) \quad (16)$$

Thus, the optimization is changed into:

$$\max_t l_t = \max_t \{\sum_{i=1}^{n} \max_{P|t} \sum_{\{a_i, a_{pa(i)}\}} [P^d(a_i, a_{pa(i)}) \log P_{B_2}(a_i|a_{pa(i)})]\}$$

When given a specific tree t, the inner maximization can be obtained by applying Kuhn-Tucker conditions and Lagrange multiplier method under the constraint:$P_{B_2}(a_i, a_{pa(i)}) \geq \xi_2$ and $\sum_{\{a_i, a_{pa(i)}\}} P_{B_2}(a_i, a_{pa(i)}) = 1$. ξ_2 is a positive constant close to zero. The solution for the associated probabilities P_{B_2} can be written as:

$$P^t_{B_2}(a_i, a_j) = \xi_2, \quad if \quad P^d(a_i, a_j) \leq 0, i \neq j; \quad (17)$$

$$P^t_{B_2}(a_i, a_j) = P^d(a_i, a_j)/Z, \quad if \quad P^d(a_i, a_j) > 0, i \neq j; \quad (18)$$

where Z is a normalization factor, which is used to guarantee $\sum_{\{a_i,a_j\}} P^t_{B_2}(a_i, a_j) = 1$;

Therefore, Eq. (17) continues to be:

$$\begin{aligned} \max_t l_t &= \\ &= \{\sum_{i=1}^{n} \sum_{\{a_i, a_{pa(i)}\}} P^d(a_i, a_{pa(i)}) \log P^t_{B_2}(a_i|a_{pa(i)})\} \\ &= \max_t \{[\sum_{i=1}^{n} \sum_{\{a_i, a_{pa(i)}\}} [P^d(a_i, a_{pa(i)}) \\ &\quad \log(\frac{P^t_{B_2}(a_i, a_{pa(i)})}{P^t_{B_2}(a_i) P^t_{B_2}(a_{pa(i)})})]] - \sum_{i=1}^{n} H(A_i)\} \end{aligned} \quad (19)$$

Where, $H(A_i) = -\sum_{i=1}^{n} P^d_{B_2}(a_i) \log P^t_{B_2}(a_i)$, $1 \leq i \leq n$. $-\sum_{i=1}^{n} H(A_i)$ is a constant for any tree structure. Thus we can remove this item from Eq. (19). Further, we define the *discriminative mutual information* for a pair of variable a_i, a_j as:

$$I^d(A_i, A_j) = \sum_{\{a_i,a_j\}} P^d(a_i, a_j) \log \frac{P^t_{B_2}(a_i, a_j)}{P^t_{B_2}(a_i) P^t_{B_2}(a_j)} \quad (20)$$

Eq. (19) continues to be:

$$\begin{aligned} &\max_t l_t \\ &= \max_t \sum_{i=1}^{n} I^d(A_i, A_{pa(i)}) \end{aligned} \quad (21)$$

This is a Maximum Weight Spanning Tree problem, where the weights are given by the discriminative mutual information.

REFERENCES

[1] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. Estimating hidden markov model parameters so as to maximize speech recognition accuracy. IEEE *Transactions on Speech and Audio Processing*, 1:77–82, 1993.

[2] F. Beaufays, M. Wintraub, and Y. Konig. Discriminative mixture weight estimation for large gaussian mixture models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 337–340, 1999.

[3] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory*, 14:462–467, 1968.

[4] R. Duda and P. Hart. In *Pattern classification and scene analysis*. John Wiley & Sons, 1973.

[5] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–161, 1997.

[6] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and bayesian multinets. *Artificial Intelligence*, 82:45–74, 1996.

[7] T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society(B)*, 58:155–176, 1996.

[8] D. Heckerman. In *Probabilistic Similarity Networks*. Cambridge, MA: MIT Press., 1991.

[9] Kaizhu Huang, Irwin King, and Michael R. Lyu. Constructing a large node chow-liu tree based on frequent itemsets. In Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima, Soo-Young Lee, and Xi Yao, editors, *Proceedings of the International Conference on Neural Information Processing (ICONIP2002), Orchid Country Club, Singapore*, pages 498–502, 2002.

[10] Kaizhu Huang, Irwin King, and Michael R. Lyu. Learning maximum likelihood semi-naive bayesian network classifier. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC2002), Hammamet, Tunisia*, page TA1F3, 2002.

[11] R. Kohavi. A study of cross validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th IJCAI*, pages 338–345. San Francisco, CA:Morgan Kaufmann, 1995.

[12] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. *In Proceedings of AAAI-92*, pages 223–228, 1992.

[13] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker Inc., New York, 1988.

[14] Patrick M. Murphy. UCI repository of machine learning databases. In *ftp.ics.uci.edu: pub/machine-learning-databases*.

[15] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986.

[16] Jaakkola Tommo S. and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems*, 1998.

[17] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young. Lattice-based discriminative training for large vocabulary speech recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 605–608, 1996.

[18] V. N. Vapnik. *The nature of statistical learning theory*. Springer, New York, 2nd edition, 2000.