CONSTRUCTING A LARGE NODE CHOW-LIU TREE BASED ON FREQUENT ITEMSETS

Kaizhu Huang, Irwin King, and Michael R.Lyu

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong

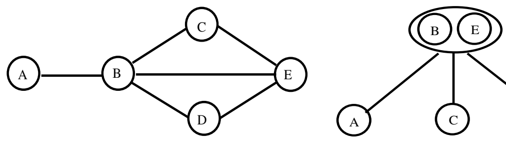
ABSTRACT

We present a novel approach to construct a kind of tree belief network, in which the "nodes" are subsets of variables of dataset. We call this Large Node Chow-Liu Tree (LNCLT). Similar to the Chow-Liu Tree, the LNCLT is also ideal for density estimation and classification applications. This technique uses the concept of "frequent itemsets" as found in the database literature to guide the construction of the LNCLT. Our LNCLT has a simpler structure while it maintains a good fitness over the dataset. We detail the theoretical formulation of our approach. Moreover, based on the MNIST hand-printed digit database, we conduct a series of digit recognition experiments to verify our approach. From the result we find that both recognition rate and density estimation accuracy are improved with the LNCLT structure.

1. INTRODUCTION

One of the interesting problems in Machine Learning is density estimation, i.e., given a training dataset, how can we estimate the data distribution. The estimated distribution can be used to perform classification or prediction tasks. Triggered by the success of Naive Bayesian network (NB), Bayesian belief network [8] has become one of the most popular techniques in solving this type of problems. By relaxing the strong assumption, i.e., the independency among the data attributes, of NB, many researchers have developed other types of Bayesian belief networks. Among them are Semi-naive Bayesian network [5], Selective naive Bayesian-network [6], and Tree Augmented Naive Bayesian network [4]. Moreover, although the Chow-Liu tree (CLT) [3] was early proposed in 1968, it can also be regarded as a type of Bayesian belief network with a looser assumption than NB.

CLT is shown to be a competitive method in distribution approximation and classification problems due to its ability to resist over-fitting problem and to have a more relaxed restriction than NB [4]. However, there are still some problems when using the CLT. The tree structure dependence assumption on the underlying structure of the training dataset will be violated in many cases. For a simple example, see Figure 1 (a). If the underlying structure of a dataset can be represented as a graph as Fig. 1 (a), the CLT method will not be able to restore this structure since Fig. 1 (a) is never a spanning tree.



(a) underlying structure (b) large node tree structure

Figure 1: A large node tree example

It is observed that if we combine some nodes as a "large node", then Fig. 1 (a) can be represented as a tree. Fig. 1 (b) is indeed such a structure which is compatible with Fig. 1 (a) since they both represent the conditional independence relationship "Given the B and E, there is an independence relationship between any two nodes in $\{A, C, D\}$".

Motivated from this observation, our approach first creates a Chow-Liu tree structure as the draft approximation over the dataset. Then we use reliable "frequent itemset" [1] together with our combination rule to refine the draft Chow-Liu tree network into the large node structure. Frequent itemsets can be considered as the subset of variables which come out together with each other frequently. Our large nodes are selected from the frequent itemsets. This selection is reasonable since the nodes in an frequent itemset are such variables which come out frequently together with each other.

We theoretically show that under our combination rule, the final refined structure will increase the data fitness over the training set. At the same time, our resulting structure can be maintained as a "tree", which is resistant to the overfitting.

In Section 2, we give the notation and definitions. In Section 3 we detail our algorithm. We demonstrate a set of

http://www.cse.cuhk.edu.hk/$\sim$$\{$kzhuang, king, lyu$\}$

experiments in Section 4. We then conclude with discussion and final remarks in Section 5. In Appendix, we show the theoretical justification of our approach.

2. NOTATION AND DEFINITIONS

Let V denote a set of n random discrete variables. Assume A is a subset of V, we denote x_A as one assignment of the variables in A. And we consider a graph $T = (V, E)$ where V is the vertex set and E is a set of undirected edges. If T is a connected acyclic graph, we call T a tree. If the number of edges $|E|$ in a tree T is equal to the number of vertex minus one: $|V| - 1$, we call T a spanning tree. Let V denote a set of random discrete variables and let V^* denote a set of the subsets of V. V^* satisfies the following condition:

$$\cup_{U_i \in V^*} U_i = V, \quad U_i \cap U_j = \phi, \quad for i \neq j, U_i, U_j \in V^*$$

A Large node tree $T^*(V^*, E^*)$ is defined as a tree where V^* is the vertex set satisfying the above conditions and E^* is the set of edges between V^*. Here we can see that each vertex of T^* is actually the subset of V and these subsets have no overlapped variables. Fig. 2 (b) is an example of a large node tree.

The distribution encoded in the large node tree can be written into:

$$P^*(x_{V^*}) = \frac{\Pi_{(u,v)\in E^*}, P^*(x_u, x_v)}{\Pi_{v \in V^*} P^*{}_v(x_v)^{deg(v)-1}}.$$

$deg(\cdot)$ means the degree of a vertex in the graph.

3. LARGE NODE CHOW-LIU TREE (LNCLT)

In this section, we give an upgrading algorithm we call Large node Chow-Liu tree(LNCLT) which upgrades the CL-tree to a large-node tree structure. We use frequent itemsets to guild our upgrading. In Appendix we show that combining the frequent itemsets whose attributes satisfy the *father-son relationship* or *sibling relationship* will increase the network's fitness over the dataset. In order to avoid a much more complex structure, we restricted the frequent itemset's cardinality less than K. In brief, we increase the learning accuracy by combining attributes with a certain relationship. On the other hand we avoid over-fitting problem by introducing reliable bounded frequent itemsets.

Before introducing the algorithm, we first give a definition about *combination transformation* in tree graph. A combination transformation is defined to be a transformation in a tree structure T. This transformation combines several nodes into a *large node* and keep the connection relationship of T. Fig. 2 is an illustration of combination transformation. In Fig. 2, (a) is a tree structure and (b) is the result after a combination transformation. In (b) when nodes D, B are combined, the edge $\overline{BE}$ in (a) will be kept as the edge $\overline{(BD)E}$ in (b). In Appendix we theoretically show that combining several nodes with a father-son relationship or sibling relationship based on any node as root will increase the log likelihood of the graphical structure, thus will make the result structure fit the samples more accurately.

Our algorithm consists of three phases. In the first phase we utilize Apriori on [1] to detect all the frequent itemsets. The second phase is basically the Chow-Liu tree construction algorithm. And in this phase, we check the support of frequent itemsets, which satisfy the combination condition. Here support means the frequency the frequent itemset happens in the dataset. In the last phase, we combine the attributes with higher supports iteratively and upgrade the CL-tree structure into LNCLT structure.

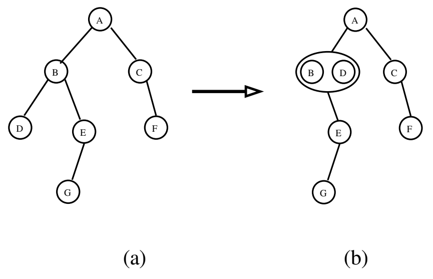


Figure 2: Illustration of combination transformation

Phase 1: Detecting the frequent itemsets

(1) Give out a good valve Minisup, which is the minimum support an frequent itemset should have. Call Apriori procedure to generate the frequent itemsets, which have the size less than k.

(2) Record all the frequent itemsets together with their supports into list L.

Phase 2: Drafting Chow-Liu tree. [3]

(3) Calculate all the mutual information between any two nodes $I(X_i, X_j)$ and insert them into Set B, initiate tree $T(V, E)$ where $V = \{$all the nodes of a data set$\}$, $E = \{\}$ and the mutual information between two variables X, Y is defined as:
$I(X, Y) = \sum_{x,y} P(x, y) \log \frac{P(x,y)}{P(x)P(y)}$.

(4) Do until E contains $n - 1$ edges (n is the number of nodes)

(5) Find the nodes pair (X_{m_1}, X_{m_2}) with maximum mutual information Im from B.
If no cycle is formed in T when the vertex X_{m_1} is

connected with X_{m_2}, add edge (X_{m_1}, X_{m_2}) in E, and delete $Im(X_{m_1}, X_{m_2})$ from B.
Else delete $Im(X_{m_1}, X_{m_2})$ from B and go to (4).

Phase 3: Adapting the tree structure based on combination transformation

(6) According to tree T, filter out frequent itemsets which do not satisfied combination conditions from L, we get the new $L^{'}$.

(7) Sort $L^{'}$ in descending order based on the supports of the frequent itemsets.

(8) Do until $L^{'}$ is $NULL$.

(a) Do the combination transformation tr based on the first itemset l_1 of $L^{'}$.

(b) Delete l_1 and any other itemset l_i in $L^{'}$ which satisfies the following condition:

$$l_1 \cap l_i \neq \phi$$

(c) Keep the new itemsets which has the size less than k from the combination transformation tr which are large itemsets and satisfy combination conditions.

(d) Finding the new itemsets from the combination. If they are frequent itemsets , insert them into $L^{'}$ and update $L^{'}$, go to (a).

4. EXPERIMENTAL RESULTS

In this section, we conduct classification experiments on MNIST database [7] of hand-printed digits to evaluate our LNCLT method. We built 10 LNCLTs for 10 digits. When a new test digit is input, we calculate the 10 probabilities based on 10 LNCLTs and output the digit whose LNCLT has the maximum probability. We show that the data have the better fitness on the LNCLT structure than Chow-Liu tree structure according to log likelihood examination. Moreover, the recognition rate is also improved with the LNCLT structure. The MNIST datasets consist of a 60000-digit training dataset and a 10000-digit test dataset. Both the training dataset and the test dataset consist of 28×28 gray-level pixels digits. We simply use a global valve to do binarization on these two datasets. This may be the partial reason why the recognition rate in our experiments is somewhat low. We use the same method in [2] to extract 96-dimension binary features from the digits. To avoid overfitting, in the experiments, we restrict the cardinality of the frequent itemset no greater than 3.

To verify that our model is superior to Chow-Liu tree model, we conduct experiments on both log likelihood and recognition rate based on MNIST in Section 4.1 and Section 4.2.

4.1. Log likelihood

We test both the training dataset and testing dataset of MNIST. The results are showed in Table 1.

Table 1: Minus Log likelihood

Digit	Training (bits/digit)		Testing (bits/digit)	
	LNCLT	CLT	LNCLT	CLT
0	30.14	30.87	30.05	31.00
1	13.08	13.75	12.12	12.86
2	33.78	34.68	33.03	34.05
3	34.49	35.51	33.87	34.95
4	27.98	28.70	27.58	28.34
5	32.45	33.17	32.31	33.18
6	26.96	27.63	26.60	27.26
7	25.01	25.83	24.84	25.79
8	34.15	34.94	33.75	34.58
9	26.90	27.52	26.12	26.63

From Table 1, we can see that the likelihood of LNCLT is greater than the one of CLT both in training dataset and testing dataset. This result is consistent with our theoretical analysis in Appendix.

4.2. Recognition rate

We first use the 60000-digit training dataset to train our LNCLT and Chow-Liu tree. To test the performance of LNCLT and Chow-Liu tree, we extract 1000 digits from the 10000-digit testing dataset randomly as our test dataset. We do the 1000-digit test for 10 times to evaluate the performance difference between the LNCLT and Chow-Liu tree. In Table 2 is the result. From Table 2, it is clear that the LNCLT performs better than Chow-Liu tree in all of 10 testing datasets.

Table 2: Recognition Rate

Dataset	1	2	3	4	5
CLT(%)	83.20	84.70	84.10	83.50	83.70
LNCLT(%)	83.70	85.90	84.70	84.20	84.90
Dataset	6	7	8	9	10
CLT(%)	85.10	84.30	83.30	83.50	83.80
LNCLT(%)	86.00	85.40	83.50	83.90	85.70

5. CONCLUSION

In this paper, we have described a method for constructing a kind of "tree" belief network: Large node Chow-Liu tree (LNCLT). This method can be seen as the extension of Chow-liu tree algorithm. With the combination of frequent itemsets, we can maintain the performance of our LNCLT no worse than Chow-Liu tree. In both theory and experiments, we verified that the LNCLT is superior to Chow-Liu tree method more or less.

6. APPENDIX

We prove in the following that the log likelihood of the "tree" constructed by our algorithm is bigger than the one in the Chow-Liu tree, which means that our hypertree is superior to the Chow-Liu tree in the meaning of maximum likelihood criterion. Before the proof, some definitions are first given. We define $H(x)$ as the entropy for a random variable X, and $H(X|Y)$ as the conditional entropy of X given Y. The relationship such as father-son relationship and sibling relationship can be obtained if any node (variable) is considered as the root in a spanning tree. We denote the training dataset S as s independent observations $x^1, x^2, \ldots, x^s$, and n variables as $\{1, 2, 3, \ldots, n\}$, then each observation can be represented as $\{{x_1}^i, {x_2}^i, \ldots, {x_n}^i\}, 1 \leq i \leq s$.

Lemma 1 *Given a training dataset S and n variables defined as the above, the* log *likelihood $l_t(x^1, x^2, \ldots, x^s)$ of the observations can be written as the following when the dataset is fit as a maximum spanning tree:*

$$l_t(x^1, x^2, \ldots, x^s) = \sum_{i=1}^{n}\sum_{k=1}^{s} \log P(x_i^k |\, x_{j(i)}^k) \qquad (1)$$

where $j(i)$ is the father of variable i obtained by the ordering based on any certain variable as the root in a tree. And this log *likelihood is maximized when the spanning tree is obtained with Chow-Liu method [3].*

The proof can be seen in [3].

Proposition 1 *Given a spanning tree T, if any two nodes satisfy father-son relationship based on a certain root, then the graphical structure T^* after a combination transformation of these two nodes is superior to the original tree T based on the maximum likelihood criterion.*

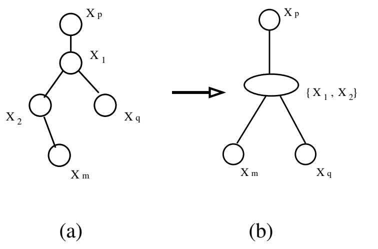


Figure 3: Father-son combination (a):The original sub-tree (b): The resulting sub-tree after the combination of X_1 and X_2

Proof: See Fig. 3, we assume that the left part is one sub-part (a) of spanning tree T (To be simple, we assume X_1 has the sons: X_2 and X_q and X_2 has only one son :X_m. We have the similar proof if X_1 and X_2 have multi-sons) and in this subpart we do the combination of two variables. As showed in the right part of Fig. 3, two nodes X_1, X_2 with father-son relationship are combined. For the spanning tree T, only the subpart (a) is changed into (b) when combining X_1 and X_2 and the other parts of T are unchanged. We rewrite the log likelihood of training dataset according to tree T into two parts:

$$\begin{aligned}
& l_t(x^1, x^1, \ldots, x^s) \\
= & \sum_{i \neq X_1, X_2, X_m, X_q} [\sum_{k=1}^{s} \log P({x_i}^k |\, x_{j(i)}^k)] + \\
+ & \sum_{k=1}^{s} [\log P(x_{X_2}^k \mid x_{X_1}^k) + \log P(x_{X_m}^k \mid x_{X_2}^k) + \\
+ & \log P(x_{X_q}^k \mid x_{X_1}^k) + \log P(x_{X_1}^k \mid x_{X_p}^k)] \qquad (2)
\end{aligned}$$

The same as (2) we can write the log likelihood of training dataset according to hypertree distribution T^* of the right part of Fig. 3 into (3):

$$\begin{aligned}
& l_{t^*}(x^1, x^1, \ldots, x^s) \\
= & \sum_{i \neq X_1, X_2, X_m, X_q} [\sum_{k=1}^{s} \log P({x_i}^k |\, x_{j(i)}^k)] + \\
+ & \sum_{k=1}^{s} [\log P(x_{X_m}^k \mid x_{X_1}^k x_{X_2}^k) + \log P(x_{X_q}^k \mid x_{X_1}^k x_{X_2}^k) + \\
+ & \log P(x_{X_1}^k x_{X_2}^k \mid x_{X_p}^k)] \qquad (3)
\end{aligned}$$

Further we can define the second part of (2) as $R(l_t)$ and write it into entropy form (4)

$$\begin{aligned}
R(l_t) &= \sum_{k=1}^{s} [\log P(x_{X_2}^k \mid x_{X_1}^k) + \log P(x_{X_m}^k \mid x_{X_2}^k) + \\
&+ \log P(x_{X_q}^k \mid x_{X_1}^k) + \log P(x_{X_1}^k \mid x_{X_p}^k)] \\
&= \sum_{k=1}^{s} \log P(x_{X_2}^k \mid x_{X_1}^k) + \sum_{k=1}^{s} \log P(x_{X_m}^k \mid x_{X_2}^k) + \\
&+ \sum_{k=1}^{s} \log P(x_{X_q}^k \mid x_{X_1}^k) - \sum_{k=1}^{s} \log P x_{X_p}^k + \\
&+ \sum_{k=1}^{s} \log P(x_{X_1}^k \mid x_{X_p}^k) + \sum_{k=1}^{s} \log P x_{X_p}^k \\
&= -H(X_2|X_1) - H(X_m|X_2) - H(X_q|X_1) - \\
&-H(X_1 X_p) + H(X_p) \qquad (4)
\end{aligned}$$

In the same way, we can write the second part of (3) into (5).

$$
\begin{aligned}
R(l_{t^*}) = \sum_{k=1}^{s} [\log P(\overset{k}{x}_{X_m} \mid \overset{k}{x}_{X_1}\overset{k}{x}_{X_2}) + \\
+ \log P(\overset{k}{x}_{X_q} \mid \overset{k}{x}_{X_1}\overset{k}{x}_{X_2}) + \log P(\overset{k}{x}_{X_1}\overset{k}{x}_{X_2} \mid \overset{k}{x}_{X_p})] \\
= -H(X_2|X_1X_p) - H(X_m|X_1X_2) - \\
-H(X_q|X_1X_2) - H(X_1X_p) + H(X_p) \qquad (5)
\end{aligned}
$$

According to entropy theory, we have:

$$
\begin{aligned}
H(X_2|X_1) \geq -H(X_2|X_1X_p), \\
H(X_m|X_2) \geq H(X_m|X_1X_2), \\
H(X_q|X_2) \geq H(X_q|X_1X_2) \qquad (6)
\end{aligned}
$$

From (4),(5) and (6) we have the following (7):

$$R(l_t) \leq R(l_{t^*}) \qquad (7)$$

From (4),(5),(7) we obtain that:

$$l_t \leq l_{t^*} \qquad (8)$$

□

Proposition 1 shows that a single combination transform will increase the log likelihood of a tree T, which means the data fitness will be increased.

Proposition 2 *Given a spanning tree T, if two nodes satisfy sibling relationship based on a certain root, then the graphical structure T^* after a combination transformation of these two nodes is superior to the original tree T based on the maximum likelihood criterion*

The proof of Proposition 2 is much similar as Proposition 1, we won't prove it here.

Based on a sequence of combination transformation, We can easily expand the Proposition 1 and Proposition 2 into the following Corollary 1 and Corollary 2. These two corollaries prove that the combination transformation of father-son relationship and siblings relationship will increase the data fitness.

Corollary 1 *Given a spanning tree T, if a subset of nodes can be sorted as a sequence based on a certain node as the root, in which each node is the father of its sequent node, then the graphical structure T^* after a combination transformation of these nodes in this subset is superior to the original tree T based on the maximum likelihood criterion.*

Corollary 2 *Given a spanning tree T, if all the nodes in a subset are sibling relationship, then the graphical structure T^* after a combination transformation of all the nodes in this subset is superior to the original tree T based on the maximum likelihood criterion.*

7. ACKNOWLEDGEMENT

This research is supported fully by grants from the Hong Kong's Research Grants Council (RGC) under CUHK 4407/99E and CUHK 4222/01E.

8. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. VLDB-94*, 1994.

[2] R. Bakis, M. Herbst, and G. Nagy. An experimental study of machine recognition of hand-printed numerals. *IEEE Transactions on systems science and cybernetics*, SSC-4(2), JULY 1968.

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

[5] I. Kononenko. Semi-naive bayesian classifier. In *Proceedings of sixth European Working Session on Learning*, pages 206–219. Springer-Verlag, 1991.

[6] P. Langley and S. Sage. Induction of selective bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, page 399C406. San Francisco, CA: Morgan Kaufmann, 1994.

[7] Y. Le Cun. http://www.research.att.com/ yann/exdb/ mnist/index.html.

[8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: networks of plausible inference*. Morgan Kaufmann, CA, 1988.