

The Study of Clique Tree Structure on Graph Model

Han Kaiwen*, Xu Cheng

School of Mathematics and Statistics, Qingdao University, Qingdao, Shandong, China

*Corresponding author: Han Kaiwen
 DOI: [10.21276/sjpms.2019.6.1.1](https://doi.org/10.21276/sjpms.2019.6.1.1)

| Received: 05.01.2019 | Accepted: 16.01.2019 | Published: 23.01.2019

Abstract

Original Research Article

The clique tree expresses dependencies in a high dimensional attribute space and can be used to make probabilistic inferences about data. For the high dimensional data analysis problem, the joint probability distribution is constructed by using the clique tree structure; a joint probability distribution in a high dimensional space can be decomposed into a product of lower dimensional probabilities. Within low dimensional spaces, the data is more concentrated and a probability distribution can be successfully derived.

Keywords: Clique tree, Joint probability distribution, Maximal clique, Probability graph model.

Copyright © 2019: This is an open-access article distributed under the terms of the Creative Commons Attribution license which permits unrestricted use, distribution, and reproduction in any medium for non-commercial use (NonCommercial, or CC-BY-NC) provided the original author and source are credited.

INTRODUCTION

In recent years, Probabilistic graph models are widely used in many fields such as medical diagnosis, fault diagnosis, gene and chromosome data analysis, communication and coding. In the probability graph model, the clique tree structure is an active research field. The clique tree structure has been successfully applied in transportation [1], fault diagnosis [2] and many other fields. Research about clique trees in literature has generally focused on inference algorithms as well as building of clique trees with a narrow width for fast inference. For instance, clique tree based inference is discussed in [3] and [4]. The clique tree approach is analogous to density estimation for numerical domains, but is more general as it can be used to infer probabilities of both numerical and categorical data. In high dimensional data sets, there is generally insufficient data from which to characterize probabilities; the available data points are spread too thinly over a very large space of possible attribute combinations. However, if it is known that some subsets of the variables are independent from other subsets of variables, a joint probability distribution in a high dimensional space can be decomposed into a product of lower dimensional probabilities [5,6]. In order to construct a clique tree, we need to find a maximal clique in the network. The most commonly used search algorithm in the maximal clique is the Bron-Kerbosh (BK) algorithm [7], which is recursive. Applying to sparse graphs leads to an increase in recursive trees and affects computation time. We use a maximal clique search algorithm to determine all the maximal cliques in a network, and then construct a clique tree structure through the maximal clique. The clique tree expresses dependencies in a high dimensional attribute space, a joint probability distribution in a high dimensional space can be decomposed into a product of lower dimensional probabilities.

Preliminaries

Definition 1 $G = (V, E)$ represents an undirected graph [7], where V is the set of vertices and $E \subseteq V \times V$ represents the corresponding set of edges.

Definition 2 A complete subgraph of G is called a clique, If a clique is not included by any other clique, the clique is called the maximal clique of G .

Definition 3 Let T be a clique tree over a set of factors ϕ . We denote by V_T the vertices of T and by \mathcal{E}_T its running edges. We say that T has the running intersection property if, whenever there is a variable X such intersection property that $X \in C_i$ and $X \in C_j$, then X is also in every clique in the (unique) path in T between C_i and C_j .

Definition 4 Let ϕ be a set of factors over \mathcal{X} . A clique tree over ϕ that satisfies the running intersection property clique tree is called a clique tree (sometimes also called a junction tree or a join tree). In the case of a clique tree, the cliques are also called cliques.

Definition 5 The joint probability distribution on the set of random variables $X = \{X_1, X_2, \dots, X_n\}$ is denoted by $P(X_1, X_2, \dots, X_n)$.

Constructing a Clique Tree

Generally, we construct a clique tree by finding all the maximal cliques in the network. So first we should find out the maximal clique in the network. Learning through probability graph models and graph theory, we can use a search algorithm for maximal cliques in the network. The algorithm steps are as follows:

Algorithm 1

Input: Undirected graph $G = (V, E)$, Where V is the set of vertices and E is the set of edges.

Output: Collection of maximal cliques M .

Begin

Step 1 make $V_{\min} \rightarrow V$, Which V_{\min} represents the maximal clique with the fewest number of nodes;

Step 2 If there is no edge between the point a in the vertex set V and the point outside, we should delete the point, otherwise we should leave the point;

Step 3 If there are two endpoints a, b of one edge $e_{a,b}$ in E , and edges $e_{a,p}, e_{b,p}$ exist at a point p other than vertex set V , then edge $e_{a,b}$ is preserved, otherwise edge $e_{a,b}$ is deleted.

Step 4 If the network after the deletion becomes an empty graph, we can execute Step 5; Otherwise, we should execute Step 1 to Step 4 until the network becomes empty, and Step 5 is executed.

Step 5 Output M Algorithm 1 is described by running the procedure of Algorithm 1 through Concrete Example. The graph model structure with twelve random variables is shown in Figure 1[8].

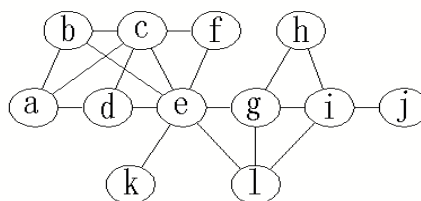


Fig-1: The graph with 12 random variables

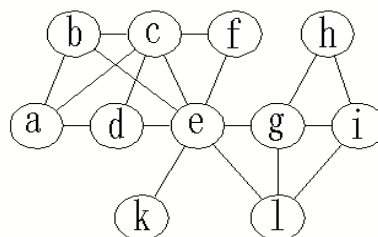


Fig-2

In Figure 1, each node represents a related variable, and the edge between nodes represents the correlation between two nodes. Because the algorithm is mainly applied to the network structure, we do not explain the meaning of the node specific representation.

The process of algorithm 1 for Figure 1, We can get a clique clique of $M_1(i, j)$ through Step 1., The edge set $E = \{e_{i,j}\}$ and the vertex set $V = \{i, j\}$ are contained in the clique M_1 ; We can execute Step 2 and get the point j , we need to delete the point j ; We can execute Step3 and get the edge $e_{i,j}$, we need to delete the edge $e_{i,j}$. Finally, We can get a new network graph as shown in Figure 2.

Looking at Figure 2, we should Step 4, because Figure 2 is not an empty graph, so we need to execute Step 1 to Step 4. We can get a clique clique $M_2(e, k)$ of through Step 1 in Figure 2., The edge set $E = \{e_{e,k}\}$ and the vertex set $V = \{e, k\}$ are contained in the clique M_2 ; We can execute Step 2 and get the point k , we need to delete the point k ; We can execute Step3 and get the edge $e_{e,k}$, we need to delete the edge $e_{e,k}$. Finally, We can get a new network graph as shown in Figure3.

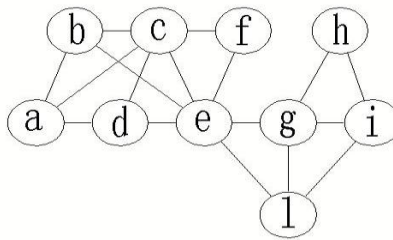


Fig-3

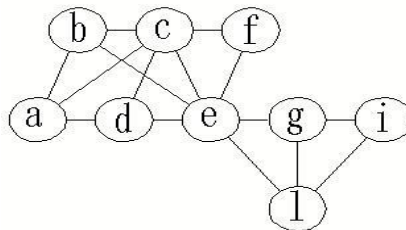


Fig-4

Looking at Figure 3, we should Step 4, because Figure 3 is not an empty graph, so we need to execute Step 1 to Step 4. We can get a clique clique $M_3(g, h, i)$ of through Step 1 in Figure 2., The edge set $E = \{e_{g,h}, e_{h,i}, e_{g,i}\}$ and the vertex set $V = \{g, h, i\}$ are contained in the clique M_3 ; We can execute Step 2 and get the point h , we need to delete the point h ; We can execute Step3 and get the edge $e_{g,h}, e_{h,i}$, we need to delete the edge $e_{g,h}, e_{h,i}$. Finally, We can get a new network graph as shown in Figure 4.

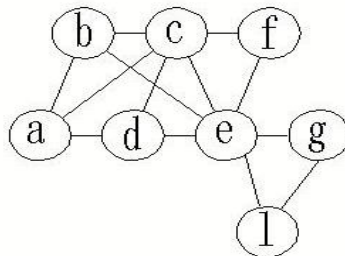


Fig-5

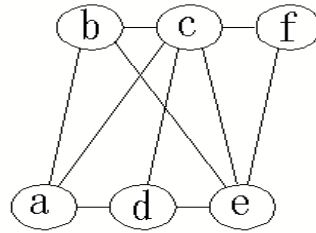


Fig-6

Looking at Figure 4, we should Step 4, because Figure 3 is not an empty graph, so we need to execute Step 1 to Step 4. We can get a clique $M_4(g, l, i)$ of through Step 1 in Figure 2. The edge set $E = \{e_{g,l}, e_{l,i}, e_{g,i}\}$ and the vertex set $V = \{g, l, i\}$ are contained in the clique M_4 ; We can execute Step 2 and get the point i , we need to delete the point i ; We can execute Step 3 and get the edge $e_{g,i}, e_{l,i}$, we need to delete the edge $e_{g,i}, e_{l,i}$. Finally, We can get a new network graph as shown in Figure 5.

Repeat the above steps, the graphic changes during the execution of the algorithm are as shown in Figure 5, Figure 6 and Figure 7. The maximal clique $M_5(g, l, e)$, $M_6(c, e, f)$ and $M_7(b, c, d, e)$ are obtained in turn, and then Figure 8 is obtained.

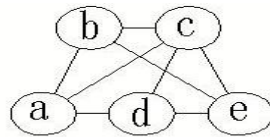


Fig-7

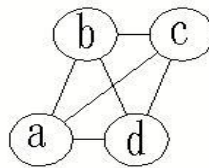


Fig-8

Perform Step 4 on Figure 8, because Figure 8 is not an empty graph, so we need to execute the algorithm, we can get a maximal clique $M_8(a, b, c, d)$. We need to delete the point a, b, c, d and the edge $e_{a,b}, e_{b,c}, e_{c,d}, e_{d,a}, e_{a,c}, e_{b,d}$. Because we get an empty figure, so execute Step 5 and get the maximal clique combination $M = \{M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8\}$ of the network in Figure 1.

In order to form a clique graph, to form a clique graph, the maximal cliques of the input graph become the nodes of the clique graph. Two clique graph nodes are linked if the cliques have at least one underlying node in common. For instance, in the graph in Figure 9(a), the node set $\{a, b, d\}$ is a maximal clique, and therefore $\{a, b, d\}$ becomes a node in the clique graph. Find all the maximal cliques in Figure 9(a) by algorithm 1, and then construct the clique tree structure according to the maximal clique. The network structure of Figure 9(b) is a clique tree of the network of Figure 9(a), in which the ellipse in Figure 9(b) represents a maximal clique and the rectangle represents the separator set.

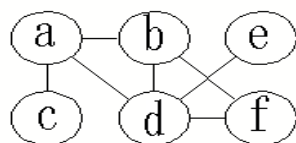


Fig-9(a)

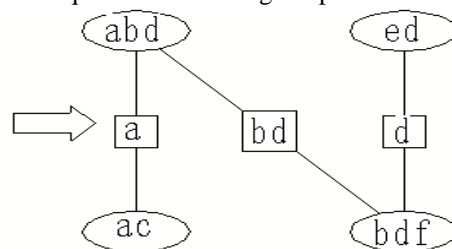


Fig-9(b)

Lastly, we can construct the clique tree of Figure 1, and the resulting clique tree is shown in Figure 10.

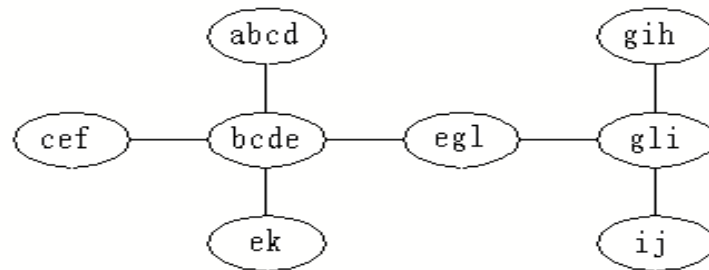


Fig-10: Clique structure constructed according to the network of Figure1

Joint probability distribution

By constructing a clique tree, the derived probability model is easily read from the clique tree. For any data point (a, b, c, d, e, f) in Figure 9(a), its probability is given by the product of the probabilities within subspaces corresponding to maximal cliques, divided by the probabilities of the subspaces given by the separator sets:

$$P(a, b, c, d, e, f) = \frac{P(a, c)P(a, b, d)P(b, d, f)P(e, d)}{P(a)P(b, d)P(d)} \quad (1)$$

Similarly, we can also derive the probability of any data point $(a, b, c, d, e, f, g, h, i, j, k, l)$ in Figure 1 as:

$$P(a, b, c, d, e, f, g, h, i, j, k, l) = \frac{P(M_1)P(M_2)P(M_3)P(M_4)P(M_5)P(M_6)P(M_7)P(M_8)}{P(b, c, d)P(c, e)P(g, l)P(g, i)P(i)P(e)P(e)}$$

Some of the benefits of this probability decomposition are immediately clear from its structure. First, the right side of equation (1) contains only lower dimensionality probabilities. The lower dimensional probabilities can be more reasonably inferred from available data, which addresses the problem of data sparsity and the curse of dimensionality.

CONCLUSION

The clique tree method decomposes the joint probability distribution into a joint probability distribution in the lower dimension. By using the low-dimensional subspace, the probability model overcomes the problem of data sparseness. With the in-depth study of the clique tree problem, the application of the clique tree will be more extensive.

REFERENCES

1. Shao Haipeng, Li Jingchen, Shei, etc. Study on the interpolation behavior of two-wheeled vehicles in bus stop based on Bayesian network [J]. Journal of Wuhan University of Sciences: Transportation Science and Engineering edition, 2017, 41 (6): 969-973., 2017, 41(6):969-973.
2. Marcunbao, Zhao Meng, Zhou Fangwang. Application de l'algorithme probabiliste causalité réseau connexion au diagnostic de panne [J]. Automatismes électriques, 2014, 36(4):16-18.
3. Pearl J. "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," (2nd ed.). San Francisco, CA: Morgan Kaufmann, 1988.
4. Shahaf D, Chechotka A, and Guestrin C. "Learning thinjunction trees," Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, FL, USA. Vol. 5 of JMLR: W&CP 5, 2009.
5. Chow C, Liu C. Approximating discrete probability distributions with dependence trees. IEEE transactions on Information Theory. 1968 May;14(3):462-7.
6. Robertson N and Seymour PS, "Graph minors," J. of Combinatorial Theory, Series B 36: 49-64, 1984.
7. Pablo San Segundo, Jorge Artieda, Darren Strash. Efficiently enumerating all maximal cliques with bit-parallelism [J]. Computers and Operations Research. 2018;92:37-46.
8. Bondy J A, Murty USR. Graph theory with application [M]. Great Britain: Macmillan. 1976:1-13.
9. Ting Yu, Mengchi Liu. A linear time algorithm for maximal clique enumeration in large sparse graphs [J]. Information Processing Letters. 2017, 125:35-40.