Notes on Variational Methods for Graphical Models by M.I. Jordan, 1999

Notes compiled by D.Gueorguiev, 10/20/24

# Introductory Notes

The problem of probabilistic inference in graphical models is the problem of computing a conditional probability distribution over the values of some of the nodes (the “hidden” or “unobserved” nodes), given the values of other nodes (the “evidence” or “observed” nodes).

Let us denote with the set of hidden nodes, and with the set of evidence nodes. We want to calculate the conditional probability related to a hidden node given as

(1)

We also may want to calculate marginal probabilities in graphical models, in particular the probability of the observed evidence . is a function of the parameters of the graphical model for fixed , it is also known as *likelihood*.

Clearly (from (1)) the evaluation of the likelihood is closely related to the calculation of .

## The need for approximation methods for computing the inference

Although there are many cases in which the exact algorithms provide a satisfactory solution to inference problems, there are cases in which the time and/or space complexity of the exact calculation is unacceptable. For example, within the context of the junction tree construction the time complexity is exponential in the size of the maximal clique in the junction tree. The junction tree algorithm will be discussed in the next section.

# Exact Inference

# References

[1] [Introduction to Variational Methods for Graphical Models, Michael I. Jordan et al, UC Berkeley, 1999](https://github.com/dimitarpg13/image_processing/blob/main/literature/articles/variational_methods/Intro_to_Variatonal_Methods_for_Graphical_Models_Jordan_1999.pdf)

[2] [Probabilistic Modeling and Reasoning: The Junction Tree Algorithm, David Barber, 2003](https://github.com/dimitarpg13/image_processing/blob/main/literature/articles/variational_methods/Probabilistic_Modeling_And_Reasoning_The_Junction_Tree_Algorithm_David_Barber_2003.pdf)

#### [3] [Bayesian Reasoning and Machine Learning, David Barber, 2007-2020](https://github.com/dimitarpg13/probabilistic_machine_learning/blob/main/books/BayesianReasoningAndMachineLearningBarber2007-2020.pdf)

[4] [Learning Bayesian Networks, Richard Neapolitan, 2006](https://github.com/dimitarpg13/learning_bayesian_networks/blob/main/book/LearningBayesianNetworksRNeapolitan.pdf)

# Appendix

## Belief Networks

Note: Belief Network is also known as Bayesian Network (Neapolitan, [4]) and it will be abbreviated as BN through this document.

**Definition**: A BN is a distribution of the form

(A1.1)

where represent the *parental variables* of variable . A BN corresponds to a DAG with the th node in the graph corresponding to the factor .

Question: Does a given BN corresponds to a specific instance the distribution (A1.1) given by its conditional probability tables or refers to *any* distribution consistent with the specified structure of (A1.1)? One can distinguish between BN containing numerical specification and BN graph which describes structure and does not have numerical specification. We will always elaborate if we are dealing with the former or the latter if it is not clear from the context.

Note that we have a choice how to recursively use Bayes’ rule. For example, in four variables we could choose factorization:

or :

Note: in general, two different BNs may represent the same independence assumptions. More on this in the next section.

A diagram of a diagram

Description automatically generated with medium confidence

Figure A1.1: The BNs for the example 4 variable distribution. Both networks represent the same distribution . They represent the same (lack of) independence assumptions.

Since any distribution can be written in cascade form the figure above gives an algorithm for constructing a BN on variables : write down the -node cascade graph, label the nodes with the variables in any order; now each successive independence statement corresponds to deleting one of the edges. More formally, this corresponds to an ordering of the variables which, without loss of generality we write as .

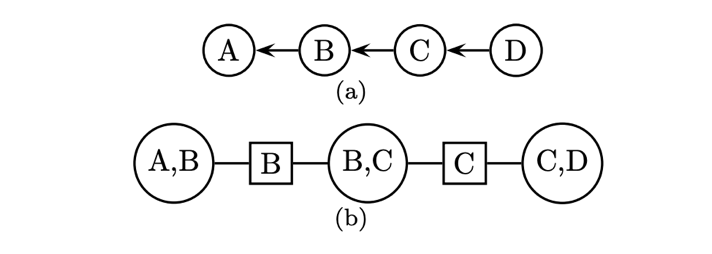
Then, from Bayes’ rule we write:

Thus, the representation of any BN is therefore a DAG.

## The Junction Tree Algorithm

### Cluster Potential Representation of a Graph

Consider a directed chain where the set (for *universe*) represents all the variables in the graph. The cluster graph distribution is defined as the product of all the cluster potentials, divided by the product of the separator potentials. In Figure A.1, for a cluster graph to represent the Belief Network we need

Figure A2.1 (a) A belief network. (b) A cluster graph representation of the network. The cluster potentials are defined on the round/oval nodes, and the separator potentials are defined on the square nodes.