
Stochastic Systems Project 1

Topic: Graphical Models

Pattern Recognition and Machine Learning, 2006

By Christopher M. Bishop

Critique By

Tharun Battula - 824000197

Texas A&M University

Faculty: Dr. P. R. Kumar

Report compiled in partial fulfillment of the requirements for the

ECEN 755 stochastic systems, spring 2016

1 Abstract

In machine learning and pattern recognition problems probabilities play central role. Most of the real life systems can be modeled using stochastic models or Bayesian reasoning. One can notice at the wide variety of applications, stochastic systems can cover either as direct mixture model or approximation or hidden models. Probability inference and learning of probability works on the sum rule and the product rule. In this context, this book develops analysis using diagrammatic representations of probability distributions, called probabilistic graphical models. These offer several useful properties and become highly advantageous in learning complex models. This document summarizes my understanding and provides critique review on the description of this chapter on graphical models.

2 Introduction

The author uses the sum rule and product rule of probability to support simplify complex calculations. Sum rule is the probability marginalization. This below equation is sum rule over event $Y = y_j$ varying over L possibilities. Explanation in the chapter 1 of the book.

$$p(X = x_i) = \sum_{j=1}^L p(X = x_i, Y = y_j)$$

Product Rule – is described as below is multiplication conditional distribution over subset class with priori of subset class equals probability of combined occurrence of events over entire sample space.

$$\begin{aligned} p(X = x_i, Y = y_j) &= \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N} \\ &= p(Y = y_j | X = x_i) p(X = x_i) \end{aligned}$$

2.1 Bayesian Networks

The author explains of simple construction of graphical models starting from simple conditional independence models. For example simple Bayesian theorem is used for expanding joint distribution as follows and directed acyclic graph is constructed where conditional distribution is marked as arrow.

$$p(a, b, c) = p(c|b, a)p(b|a)p(a)$$

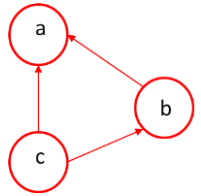
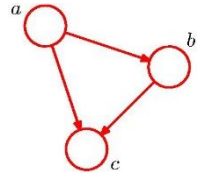
Author explains about only one configuration whereas different configurations are also possible for same joint distribution. $p(a, b, c) = p(a|b, c)p(b|c)p(c)$. There is no unique ness for a joint distribution, but we can find graphical model unique mapping for a given algebraic equation. Author extends the discussion

to joint distribution with multiple variable showing the equivalent model. These graphs form Directed Acyclic graph (DAG)

$$p(x_1, \dots, x_K) = p(x_K | x_1, \dots, x_{K-1}) \dots p(x_2 | x_1) p(x_1) \quad p(\mathbf{x}) = \prod_{k=1}^K p(x_k | \text{pa}_k)$$

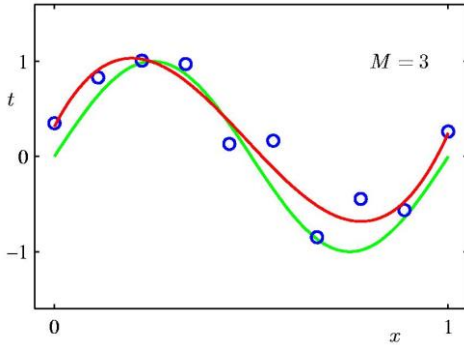
which have specific properties explored from graph theory. Author generalized

factorization for joint distribution as on left side product, where pa_k is the parent node of x_k node.



2.1.1 Bayesian Curve Fitting

The author illustrates through curve fitting of polynomial regression example. To introduce the new



representation of different model parameters and variables in graphical model. Polynomial where w_j coefficients are unknown can be estimated by bayesian curve fitting model. Where t_n are observation points,

\mathbf{w} is vector of coefficients

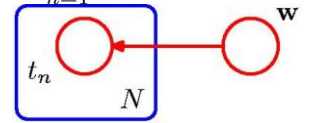
$$y(x, \mathbf{w}) = \sum_{j=0}^M w_j x^j$$

$$p(\mathbf{t}, \mathbf{w}) = p(\mathbf{w}) \prod_{n=1}^N p(t_n | y(\mathbf{w}, x_n))$$

To simplify author uses new representation by using plate to represent number of observations. Same model can be represented in more generic

way with parameters and stochastic variables as below. The author's notation, is clearly specified and introduced gradually.

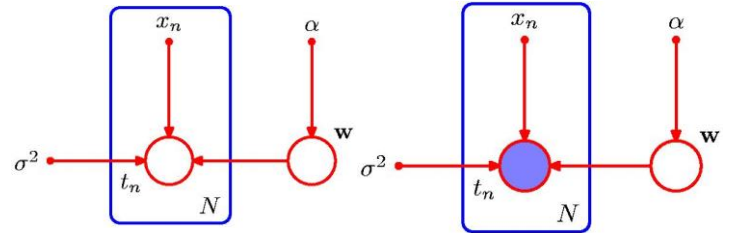
$$p(\mathbf{t}, \mathbf{w} | \mathbf{x}, \alpha, \sigma^2) = p(\mathbf{w} | \alpha) \prod_{n=1}^N p(t_n | \mathbf{w}, x_n, \sigma^2).$$



Unknown Random Variables	Open Circles
Deterministic parameters	Solid small dots
Training Data (observations)	Blue filled circle

For some training data (Blue circle) the posterior probability

can be written on priori distribution and $p(\mathbf{w} | \mathbf{t}) \propto p(\mathbf{w}) \prod_{n=1}^N p(t_n | \mathbf{w})$ conditional probabilities as this equation.



2.1.2 Model Prediction

Author provides simplistic explanation for the predictive distribution \hat{t} model becomes

$$p(\hat{t} | \hat{x}, \mathbf{x}, \mathbf{t}, \alpha, \sigma^2) \propto \int p(\hat{t}, \mathbf{t}, \mathbf{w} | \hat{x}, \mathbf{x}, \alpha, \sigma^2) d\mathbf{w}$$

immediate from the marginalizing the joint distribution over the parameter \mathbf{w}

Where the inner part is obtained through product rule for the training observation and posterior distribution.

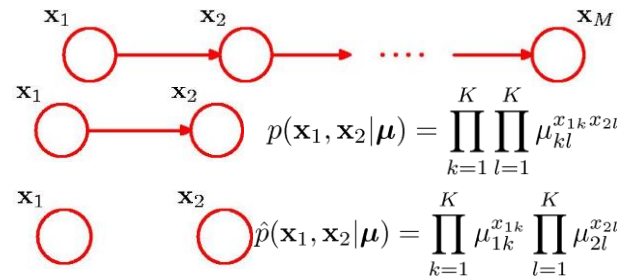
$$p(\hat{t}, \mathbf{t}, \mathbf{w} | \hat{x}, \mathbf{x}, \alpha, \sigma^2) = \left[\prod_{n=1}^N p(t_n | x_n, \mathbf{w}, \sigma^2) \right] p(\mathbf{w} | \alpha) p(\hat{t} | \hat{x}, \mathbf{w}, \sigma^2)$$

The author provides introduction to **generative models** concept and how causal process systems can be considered.

Describes the idea through image generation from object, position and orientations as the latent variables with prior distribution and object recognition is posterior distribution estimation task. The polynomial regression does not fit in this model since because there is no probability distribution associated to x variables.

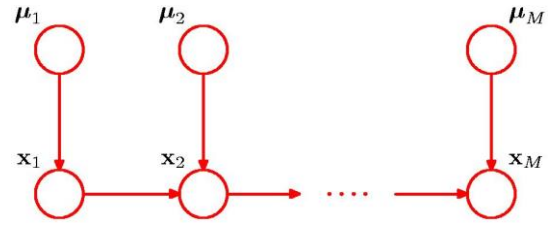
2.1.3 Discrete Variables

Author explains about joint distribution of discrete variables, with K-state random discrete variables x_1, x_2 . Explains the logic for the evaluation of K parameters. Author uses 1 of K representation to represent the calculate probabilities. General joint distribution:



requires $K^2 - 1$ parameter where as Independent joint distribution: requires $2(K - 1)$ parameters. Author provides, simplistic explanation to the advantage of evaluation of independence property in order to reduce the complexity. Author takes the description to general joint distribution over M variables requiring $KM - 1$ parameters. The chain shown in side represents M-node Markov chain where each state can take k possible options.

It requires $(K - 1) + (M - 1) K(K - 1)$ parameters. The Markov chain

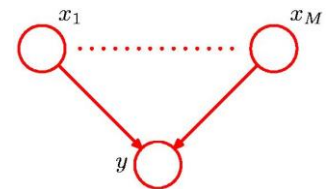


independence is used Extending the model M.C Model to include Dirichlet priors over the parameters governing the discrete distributions $p(\{x_m, \mu_m\}) = p(x_1 | \mu_1) p(\mu_1) \prod_{m=2}^M p(x_m | x_{m-1}, \mu_m) p(\mu_m)$ where $p(\mu_m) = \text{Dir}(\mu_m | \alpha_m)$

Author has been continuing the discussion in terms of controlling exponential growth of parameters, either by establishing the independence relation or conditional independence or establishing the single prior distribution. Here by sharing single μ prior for all the x random variables reduces complexity further. Another way for reducing parameters of models is to use

parameterized model for the conditional distributions. Logistic Sigmoid is provided as illustration where it uses parameters linear in size. Author beautifully connects the explanation as how this model is restrictive with less parameters where covariance matrix is considered only diagonal matrix if treated for Multivariate Gaussian

Distribution. $p(y = 1 | x_1, \dots, x_M) = \sigma \left(w_0 + \sum_{i=1}^M w_i x_i \right) = \sigma(\mathbf{w}^T \mathbf{x})$



2.1.4 Linear Gaussian Models

Author provides how a multivariate Gaussian can be expressed as DAG. Since each node is Gaussian, the mean is a linear function of the parents. Hence the expectation values of x_i can be established in linear equation. Using graphical representation based on parents this is further simplified. Similarly covariance matrix is simplified. Author extends to the vector values Gaussian nodes case with the equation as.

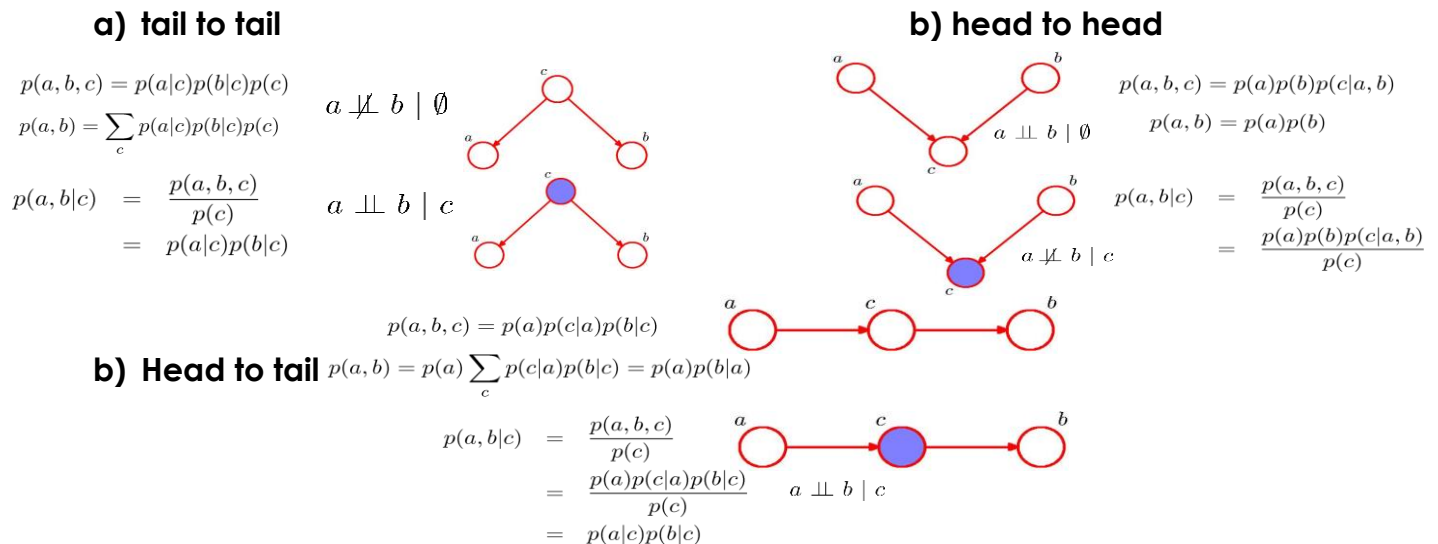
$$p(\mathbf{x}_i | \text{pa}_i) = \mathcal{N} \left(\mathbf{x}_i \left| \sum_{j \in \text{pa}_i} \mathbf{W}_{ij} \mathbf{x}_j + \mathbf{b}_i, \Sigma_i \right. \right)$$

2.2 Conditional Independence

The main property of probability distributions is independence. In this section conditional independence for graphical model is explained as the basis for further proofs and algorithms in further sections. Notation is $a \perp\!\!\!\perp b \mid c$

2.2.1 Examples

Author provides simple explanation through three examples for each. It is provided with intuitive explanation added with algebraic manipulations supporting. A path is said to be blocked if it creates the independence condition.

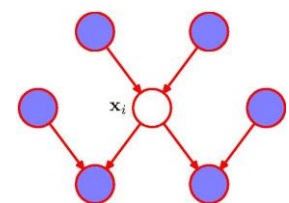


Author provides an intuitive example for head to head example, where battery and fuel are parents for Gauge node. Author calls the effect of conditioning children on parent s as “explain away”.

2.3 D-Separation

Author introduces this concept from (Pearl, 1988) for directed graph. This is to study the paths and observe that two nodes are conditionally independent if they are blocked with given conditioning on another node. It is explained in detail with illustration in the book. A, B, and C are non-intersecting subsets of nodes in a directed graph. A path from A to B is blocked if it contains a node such that either a) the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set C, or b) the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, are in the set C. If all paths from A to B are blocked, A is said to be d-separated from B by C. If A is d-separated from B by C, the joint distribution over all variables in the graph satisfies. Author illustrates the D-separation concept on an example for finding the posterior distribution for mean of a univariate Gaussian. Author explains of D separation significance in other models mentioned in other chapters of the book. The **D Separation theorem** says that the nodes, which are D-separated, form the independence and the joint probability will factorize to product of independent terms. Author visualizes this as a directed factorization where all the probabilities are filtered by the Graphical model D-separated property.

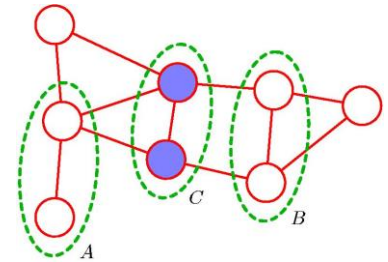
Markov Blanket is the minimal set of nodes that isolates any node x_i from rest of the graph. The author builds an argument of explaining away at head to head for any node and says that with



children of node conditioned it becomes dependent on co-parent and to isolate completely co-parent should be conditioned as well. The side diagram shows such markov blanket. Through this concept one can say that probability of a node and condition over all other nodes in the graph depends only on the markov blanket nodes conditioning where as other conditioning values get cancelled out.

3 Markov Random Fields

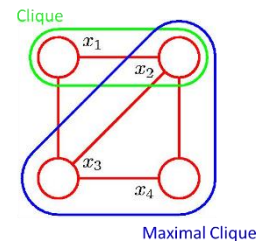
So far the directed graph has been discussed as it is more realizable in bayesian perspective. Now from the works of (Kindermann and Snell, 1980) author extend some of



the concepts to undirected graphs, though it is difficult to intuitively realize. For undirected graphs, it is difficult to build graph from algebraic model but it is possible to work from graph to model in some aspects. For this purpose **conditional independence** is introduced with an intuition of blocking. If some nodes conditioned disconnect the some paths it clearly forms independence. Author tries to address, the caveats from directed graphs to undirected graph migration in understanding D- separation to blocking explanation. No significance of explaining away property as there are no head tails in undirected graphs. Markov blanket for an undirected graph is just the set of neighbor. The author does not address with the case for loops/cycles in undirected graphs or the intuition/inference.

3.1 Factorization Properties

Author defines locality of a node with neighbors in undirected network and uses it for factorization of joint distribution using conditional independence properties. Author introduces graphical concept called **Clique**, as subset of nodes in a graph such that all pairs of nodes are in subset. i.e., fully connected subset. Maximal clique is clique where no other node from graph can be added clique subset. The joint distribution can be written as product of potential functions $\psi_C(x_C)$ over maximal cliques of graph. This factorization can be described through independence property for nodes in cliques with nodes with other cliques.



Here Z normalized constant which can be evaluated from $p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$ $Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$

It is exponentially complex to evaluate this normalization constant. However for local distributions we can work with normalized values. Author gives the intuition and relates to other literature in terms of viewing the model as filter, Cites Hammersley Clifford theorem where this factorization and maximal cliques filter are identical. Author provides a convenient example of potential function energies and the Boltzmann distribution. This restricts the representation to positive probabilities and pertaining to energy of system. However no account is given on description of this energy or intuition.

3.2 Illustration – Image Denoising

This is an excellent illustration where user can understand the application and realizes the importance of markov random field concepts and factorization could be used. Constructs Noisy Model first from original image by adding random noise. Author beautifully explains construction

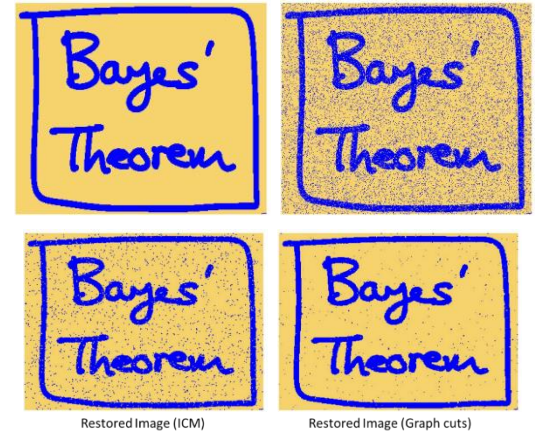
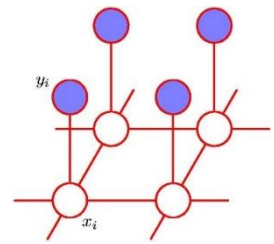
of the Markov random field Model for each pixel as a network as white circle as hidden original state of pixel value. Whereas blue circle for observed value of noisy output of the image. I could relate to the **State Estimation Problems** examples explained in the class for this.

Author mathematically intuitively explains the guess for energy function, in generic simple form where the current node bias, neighbor nodes influence/similarity and state/output similarity are considered.

$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i$$

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\{-E(\mathbf{x}, \mathbf{y})\}$$

The promising result images gives a good intuition and makes realize significance of the graphical model. The iterative conditional models results (ICM) result is explained and it being local maximum. Author provides citation to grap cut algorithm where it can develop global maximum and the discussion in other chapters.



4 Relation to Directed Graphs

Author draws analogy between the graphical frameworks developed. He explains the building of undirected graph frameworks by converting the directed graphs by simply making removing arrows in the edges. For a simple market chain converting to directed graph would result this below analogy in equations

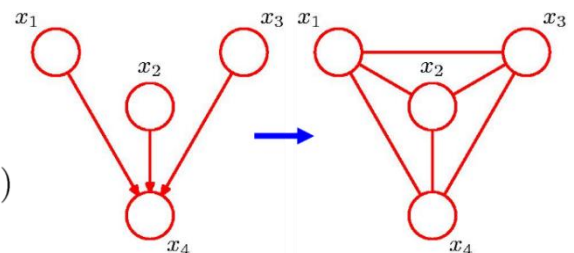
$$p(\mathbf{x}) = p(x_1)p(x_2|x_1) p(x_3|x_2) \cdots p(x_N|x_{N-1})$$

$$p(\mathbf{x}) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N)$$

Using these equation one can establish the correspondence in factorization

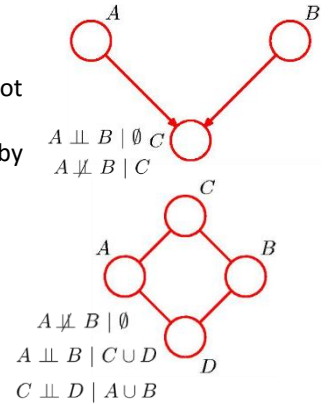
by grouping elements and potential functions in terms of probabilities and conditioned probabilities with variables.

This conversion does not simply guarantee direct analogy for all graphs. For that author tries to show simple example and how we should extra edges to make the factorization, calls it as moralization step. In a case where we add all edges to the graph in conversion, can be explained as joint distribution can be represented as potential function of all the variables. (Full



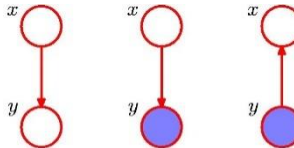
connected clique has all the variables) This increases the complexity. Not all graphs are representable in both configurations for that Venn diagram is showed below.

Author provides the limiting cases of where a conditional independence in one type of graph cannot be explained in another. As we add extra edges in directed graph we are losing information by generalizing it. Similarly for some undirected graphs where conditional independence with two variables is not cannot be obtained in directed graphs.



5 Inference on Graphical Models

Author starts explaining the Graphical Framework in intuitive manner in order to encourage more ideas through the visualization. Simple examples are given on observation conditioning and posterior distribution is converted.

$$p(y) = \sum_{x'} p(y|x')p(x')$$


$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

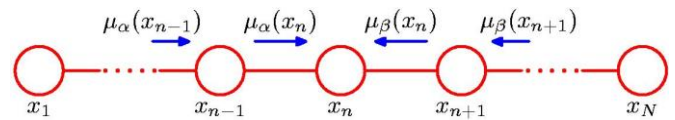
5.1 Inference on Chain

For a chain if we were to calculate marginal distribution of a node, we can start with group distribution and marginalize over all variables. This particular marginalization can be optimized based on the graphical structure and independence ideas. Author builds simple model and starts explaining step by step with mathematical notation about the marginalization with neighbours as constructs as simple recursion model.

To compute local marginal:

- Compute and store all forward messages,
- Compute and store all backward messages,
- Compute Z at any node x_m
- Compute $p(x_n)$ value at the node

$$p(\mathbf{x}) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N)$$



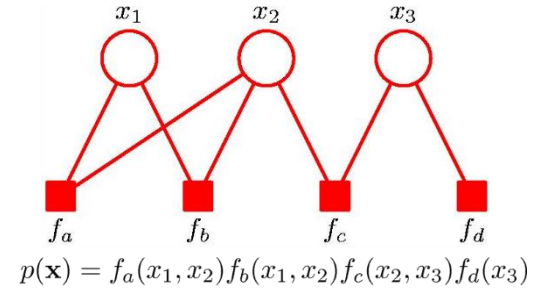
$$p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n)$$

Author extends his discussion inference on chain to trees. Here author brings discussion of directed tree and undirected trees and moralization step (adding edges) on them. In case of polytree, if added edges for moralization it would have loops.

5.2 Factor Graphs

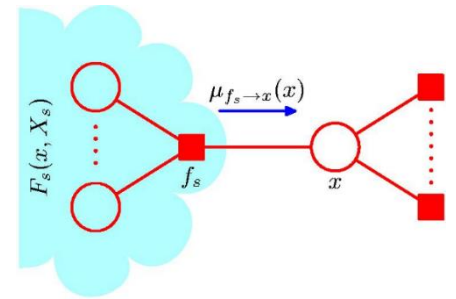
Author starts constructing factor graphs inference used for max sum algorithms in the next stages. For each factor function in the joint distribution an additional node is created in the graph and the nodes are connected to variables which functions

is variable on. A graph can have multiple types of factor graphs as different factorization is possible. Author draws analogy on the factor graphs and clique potentials. The advantage of factor graphs is it maintains tree as tree, even polytree's factor graph will have no loops avoiding problems of loops. Author explains this modeling and advantages and disadvantages including the complexity analysis. The factor graph conversion for directed and undirected is straightforward with examples.



5.3 The Sum Product Algorithm

The objective of the sum produce algorithm is to to obtain an efficient, exact inference algorithm for finding marginals and to allow computations to be shared efficiently, in situations where several marginals are required. It used key idea from Distributive Law $ab + ac = a(b + c)$ over probability sum and product rules. Author spends completer illustration explaining in detail. The probabilities for a node is mainly marginalized over joint probability which has factors.



Using factor graphs, this is easy to realize the clusters of data

$$p(x) = \sum_{\mathbf{x} \setminus x} p(\mathbf{x})$$

$$p(\mathbf{x}) = \prod_{s \in \text{ne}(x)} F_s(x, X_s)$$

$$p(x) = \prod_{s \in \text{ne}(x)} \left[\sum_{X_s} F_s(x, X_s) \right]$$

$$= \prod_{s \in \text{ne}(x)} \mu_{f_s \rightarrow x}(x)$$

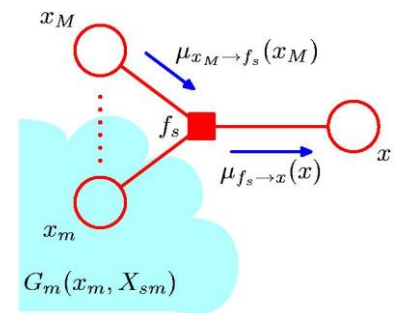
$$F_s(x, X_s) = f_s(x, x_1, \dots, x_M) G_1(x_1, X_{s1}) \dots G_M(x_M, X_{sM})$$

$$\mu_{f_s \rightarrow x}(x) \equiv \sum_{X_s} F_s(x, X_s)$$

$$\mu_{f_s \rightarrow x}(x) = \sum_{x_1} \dots \sum_{x_M} f_s(x, x_1, \dots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \left[\sum_{X_{sm}} G_m(x_m, X_{sm}) \right]$$

$$= \sum_{x_1} \dots \sum_{x_M} f_s(x, x_1, \dots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f_s}(x_m)$$

To compute local marginal, Pick an arbitrary node as root, Compute and propagate messages from the leaf nodes to the root, storing received messages at every node, Compute and propagate messages from the root to the leaf nodes, storing received messages at every node. Compute the product of received messages at each node for which the marginal is required, and normalize if necessary. An illustration for the



algorithm is provided in detail. The main advantage from the marginalization is that the normalization factor is not calculated at every step reducing the complexity. Author provides logical reasoning as normalization factors can be accumulated and would not affect the result.

$$\mu_{x_m \rightarrow f_s}(x_m) \equiv \sum_{X_{sm}} G_m(x_m, X_{sm}) = \sum_{X_{sm}} \prod_{l \in \text{ne}(x_m) \setminus f_s} F_l(x_m, X_{ml})$$

$$= \prod_{l \in \text{ne}(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

5.4 The Max Sum Algorithm

The objective is to find an efficient algorithm for finding the optimal setting \mathbf{x}^{\max} that maximizes joint distribution for entire graph. This can be treated as dynamic programming task maximization for graphical task. Here author takes the argument based on distributive property of $\max(ab, ac) = a * \max(b, c)$. Author starts explaining for simple chain and takes the discussion to factor graphs trees where we have factors product as joint distribution probability.

$$\max_{\mathbf{x}} p(\mathbf{x}) = \max_{x_n} \prod_{f_s \in \text{ne}(x_n)} \max_{X_s} f_s(x_n, X_s)$$

$$\max(a + b, a + c) = a + \max(b, c).$$

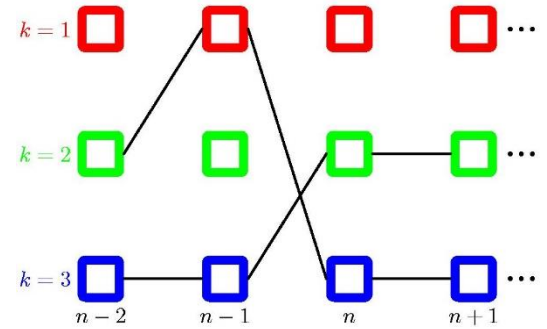
For numerical reason for small probability calculations max product calculation can be converted to max sum using logarithms and it gets the name from accordingly.

This algorithm is recursive on graph exactly as similar to the previous sections product sum algorithm, with modifications on finding the max instead of marginalizing. Author also explains initialization steps for both algorithms.

The termination happens at root node and we do the back tracking tracing the path where the maximum occurred. Author gives an

example of back tracking through trellis diagram for the M.C.

Author cites the Viterbi algorithm is the application of this max-sum algorithm where the most possible path is predicted in hidden markov models.



$$\begin{aligned} \mu_{f \rightarrow x}(x) &= \max_{x_1, \dots, x_M} \left[\ln f(x, x_1, \dots, x_M) + \sum_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f}(x_m) \right] \\ \phi(x) &= \arg \max_{x_1, \dots, x_M} \left[\ln f(x, x_1, \dots, x_M) + \sum_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f}(x_m) \right] \\ \mu_{x \rightarrow f}(x) &= \sum_{l \in \text{ne}(x) \setminus f} \mu_{f_l \rightarrow x}(x) \end{aligned}$$

5.5 Other Algorithms

Author attempts to make the case for graphs with loops. Though a complete inference with the loops may not be present, The work from (Lauritzen and Spiegelhalter, 1988, Jordan, 2007) for **Junction Tree Algorithm** is briefly described. It works by turning the initial graph into a *junction tree* and then running a sum-product-like algorithm. This junction tree formation is dependent on tree width and algorithm becomes intractable on graphs with large cliques.

Loopy belief algorithm is an approximation variational method where a graph is assumed to be with no loops and simply Sum-Product algorithm is applied and assumed to work. Initial unit messages passed across all links, after which messages are passed around until convergence though convergence is not guaranteed. Though this is an approximation it works sometimes and practical for large graphs as well. Author provides the sufficient research citations and giving overview in this concept. In the end author concludes the discussion on inference, reminding that structure of graph assumed as known and

fixed which is not true for all cases. Author opens the reader to keep open mind about the research happening in learning the graph structure itself from the data.

6 Summary

Author introduces and establishes proper basics on graphical models with illustration and intuition. He relates cites the works in this area. Made interesting points with real intuitive examples captivating the attention of reader. The algorithms are made with clear objective and proper assumptions with simple illustrations. Image De-noising example is powerful illustration of Markov Random fields. Author provides good analysis in terms of complexity in marginal and conditional probability evaluation and in design perspective of model. Provides insight into how independence or factorization property could help reduce the exponential problem to simpler problem.

Major part of the section he covered works in some classes of graphs such as trees, chains for undirected and directed nature. Details on graphs with loops is only stated in case of approximation and cited the corresponding work. It is not considered while building the models from probability. Most of the details are discussed in perspective Bayesian theory in marginalizing and conditioning. Graphical modeling for other stochastic models or Hidden Markov models discussed or as an example. In directed to undirected graph conversion, author takes good care to address the matching and intuition though for the directed graphs direct probabilistic construction is difficult. For discrete variables author mentions generalizes the theory from the continuous variable theory. In the end author mentions about complex problems and inference restriction to certain classes and generalization and approximation methods.

After reading the chapter it changed my intuition in Bayesian theory and added visualization for understanding probabilistic models. The inference on graphical models helps realization of the model and solving problems by marginalizing it.

7 References

1. Pattern Recognition and Machine Learning by Christopher M Bishop, <http://research.microsoft.com/en-us/um/people/cmbishop/prml/pdf/Bishop-PRML-sample.pdf>
2. Course Slides - <http://www.cs.columbia.edu/~blei/fogm/2015F/>
3. <http://research.microsoft.com/en-us/um/people/cmbishop/downloads/Bishop-MBML-2012.pdf>
4. Koller, Friedman(2010)
5. Course Slides - <http://courses.cms.caltech.edu/cs155/>
6. Other Books: Wittaker(1990), Lauritzen(1996) Jensen(1997), Castillo(1997) Jordan(1999, 2007), Cowell(1999)