Perspectives on Probabilistic Graphical Models

DONG LIU

Doctoral Thesis in Electrical Engineering
Stockholm, Sweden 2020

Devision of Information Science and Engineering
TRITA-EE XXXX KTH, School of Electrical Engineering and and Computer Science
ISSN .... SE-100 44 Stockholm
ISBN .... SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i Elektroteknik onsdag den 13 september 2017 klockan 13.15 i F3, Lindstedtsvägen 26, Stockholm.

Tryck: Universitetsservice US AB

To my beloved

Abstract

Sammanfattning

Acknowledgements

Contents

Acronyms and Notations

Notations

X	random variable
x	realization of the random variable X
$\mathcal{X}$	alphabet of the random variable X
X_i^k	random sequence $(X_i, \ldots, X_k)$
x_i^k	realization of the random sequence X_i^k
$\mathcal{X}_i^k$	alphabet of the random sequence X_i^k
X^k	random sequence $(X_1, \ldots, X_k)$
x^k	realization of the random sequence X^k
$\mathcal{X}^k$	alphabet of the random sequence X^k
$X_i^{k\setminus n}$	random sequence $(X_i, \ldots, X_{n-1}, X_{n+1}, \ldots, X_k)$
$x_i^{k\setminus n}$	realization of the random sequence $X_i^{k\setminus n}$
$\mathcal{X}_i^{k\setminus n}$	alphabet of the random sequence $X_i^{k\setminus n}$
$X^{k\setminus n}$	random sequence $(X_1, \ldots, X_{n-1}, X_{n+1}, \ldots, X_k)$
$x^{k\setminus n}$	realization of the random sequence $X^{k\setminus n}$
$\mathcal{X}^{k\setminus n}$	alphabet of the random sequence $X^{k\setminus n}$
$\|\| \cdot \|\|$	set cardinality
f_X	p.d.f. of the continuous random variable X
p_X	p.m.f. of the discrete random variable X
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with mean μ and variance σ^2

$\mathrm{D}(\cdot\|\|\cdot)$	Kullback-Leibler divergence
$\mathrm{D}_\tau(\cdot\|\|\cdot)$	τ-th order Rényi divergence
$\mathrm{C}(\cdot,\cdot)$	Chernoff information
$\mathrm{E}[\cdot]$	expectation
$\partial\cdot$	boundary of a closed set
$\hat{\partial}\cdot$	upper boundary of a two-dimensional closed set
$\check{\partial}\cdot$	lower boundary of a two-dimensional closed set
$\log(\cdot)$	natural logarithm

Chapter 1

Introduction

Motivate the research in probabilistic models.

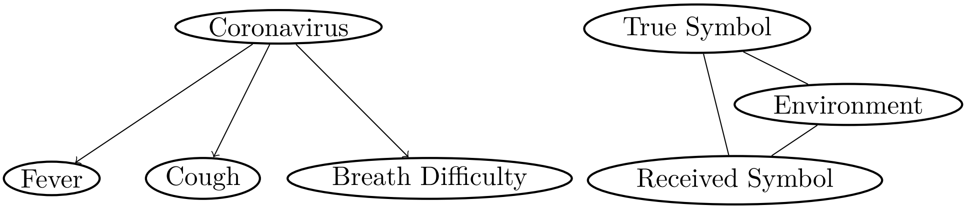
1.1 Motivations

Most tasks conducted by a person or an automated system requires a fundamental ability of *reasoning*, which is always about reaching a conclusion based on available information. At times, a conclusion is not enough and it is also required to know how reliable the conclusion is. Take the coronavirus that started from Wuhan, China at the end of 2019, as example, a doctor needs checks the information about a person to reason if the person is infected by the coronavirus. The relevant information includes symptoms such as fever, cough, breathing difficulties and probably kidney failure in severe cases. (maybe a small figure of coronavirus here.) Even after the doctor has concluded as positive or negative of ccoronavirus for the person, the natural question is why and how *confident* the diagnose is.

Two problems are inevitable to conduct the reasoning:

- How should we specify the relationship between a conclusion and the available information? In the coronavirus example, the counterpart question to answer is how the doctor should relate coronavirus infection with the symptoms. This step is called *modeling* which represents a reasoning problem abstractly by specifying the relationship between known information and unknown part, in preparation of answer query on it.
- With the model, how a conclusion should be made? This process of reaching a answer to the query is called *inference*. something about coronavirus

As times, a model is not totally fixed since one may not be sure the correctness of the assumptions about the model. A typical strategy is to leave some freedom in the configuration of the model at beginning. By using previous observations or information, the model is adjusted to be able to explain the observation in more reasonable way. This adds the following problem in reasoning:



(a) A directed graphical model for Example 1.

(b) An undirected graphical model.

Figure 1.1: Different perspectives on probabilistic graphic models. 1.1a A toy Bayesian network. 1.1b A toy Markov random field.

- Instead of having a fixed model at the first step, a set of model is given. We then need to choose one model based previous observations to do inference in order to make conclusion or answer query. This phase of choosing a model is called *learning*.

With all the discussed problems above, modelling, inference and learning, our purpose is to carry out reasoning with being aware of how confident a conclusion or answer is. These problems can be treated nicely with probabilistic models. Probabilistic models is built on the fundamental calculus of probability theory that is natural to accommodate the *uncertainty*, which is desired in reasoning. In additional, the probabilistic models offers rich space to modeling problems, where inference can be carried on either exactly or approximately. **More importantly, the modeling or modeling learning part is not necessarily coupled with inference algorithm.** This proper separation allows free that a certain family of general inference algorithms can be applies to a broad class probabilistic models. It offers the freedom of trying different models of a class without the need of replacing inference algorithm.

Example 1. *Consider the coronavirus infection problem. Using probabilistic model, we are able to model the problem in a rigid way. Additionally, we can make query more formally in probabilistic model framework. Assume each symptom among fever, cough and breathing difficulty can take value from* $\{\text{True}, \text{False}\}$. *Also the coronavirus infection is either true or false. One exemplified query can be*

$$P(\text{Infection} = \text{True}|\text{Fever} = \text{True}, \text{Cough} = \text{False}, \text{BreathingDifficulty} = \text{True}),$$

which is asking how likely the patient is infected by coronavirus if symptoms of both fever and breathing difficulty are observed but no sight of cough.

Given the fact that probabilistic theory offers a rigid foundation to model and study the problems, which is used to answer query that we concerns, it soon becomes intractable when dozens or hundreds of relevant attributes are joint considered. This can be exemplified by giving finer levels of each symptom in coronavirus

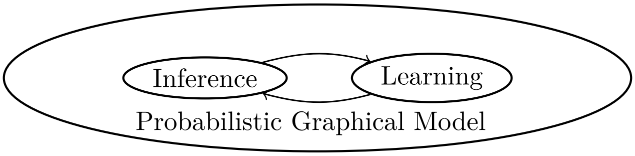


Figure 1.2: Two key aspects in practical graphical models.

infection, e.g. symptom fever is represented by the actual body temperature in integer instead of true-or-false binary state on the one hand. On the other hand, there could be more directly and indirectly relevant symptoms such as muscle pain and congestion. Together with the symptoms, the recent travel itinerary is also related. Additionally, season flu could also similarly bring up some symptoms listed above.

Probabilistic graphical model offers a general framework to encode random variable dependency of a complex probabilistic distribution into a structured graph, which is a powerful tool to compactly model relevant attributes and facts of a complex problem. As show in Figure 1.1a that represents the problem of Example 1 into a directed graphical model (or Bayesian network), the nodes (or vertexes) corresponds to the variables that represents symptoms and infection state, whereas the edges between nodes correspond how one variable may influence others. At times, it is natural to use directed graphical models (generative models) to represent that a observable variable is dependent on a latent variable that generates the observations. For instance, a noisy location sensor of a car keeps measuring the car's true location and reading noisy locations.

In contract to the directed graphical model, there are more scenarios that the interaction between related random variables is not directional, an undirected edge is used, which leads the undirected graphical model (or Markov random field, Section 2.3) representation. The undirected graphical models are popular used in computer vision [], computational biology [], digital communication [], statistical physics, etc. Figure 1.1b illustrates an exemplified undirected graphical model in digital communication context, where the receiver wants to guess what is the true symbol by joint considering the communication environment and the received symbol, whereas the symbol received by the receiver is jointly formulated by the true symbol and environment.

Probabilistic graphical model offers a 'scientific language' to do reasoning with uncertainty within framework of probabilistic theory. It is usually very nature to represent a complex system or problem by a probabilistic graphical model. The compact representation of probabilistic graphical model bridges the joint distribution of a complex system, and its graphical abstraction that captures the statistic dependency reflecting our understanding of the system. The advantage of its representation power is one of its popular application in difference disciplines.

Probabilistic graphical model coupled with its underlining distribution is also a powerful tool for effective inference, apart from its advantage of representation power. It allows to answer queries with regarding to the underlining distribution when practical inference algorithms are provided, which meets our need of reasoning with uncertainty. In addition to inference, probabilistic graphical model also supports learning from data. With certain amount of data available, a probabilistic graphical model can be learned to explain the observed data better in addition to align with our own understanding of a domain. The learned graphical model can serve to do inference with higher confidence in return. A diagram is illustrated in Figure 1.2. As would become clear in Part II, the inference may be needed to carry out model learning as well, apart from the above mutual-benefiting interaction.

1.2 Scope and Thesis Outline

We gives the intuition and motivation of probabilistic graphical model in last section, and the interaction between inference and learning in this framework. We would explore a bit further and state what topics within this framework we would cover in this thesis.

Inference in probabilistic graphical model is about the answer queries with regarding to its coupled distribution. These queries can be generally grouped into the following cases:

- Computing the likelihood of observed data or unobserved random variable.
- Computing the marginals distribution over a particular subset of nodes.
- Computing the conditional distribution a subset of nodes given the configuration of another subset of nodes.
- Computing the most likely configuration of (a subset of) nodes.

The work of this thesis would be mainly related with the first three cases in inference part.

Due to either the requirement of efficiency in solving a problem or the graphical structure of the problem's representation, it is not always that case that the above inference problem can solved exactly. Thus inference methods can be divided into

- Exact inference,
- Approximate inference.

For a limited class of graphs, exact inference such as variable elimination and sum-product algorithm can be used. Some graphs also allow efficient inference after mild modification, e.g. junction tree method. However, the above listed inference problems can only be approximately solved in general graphs. The approximation inference family can be further broken into

- Stochastic Approximation (Particle methods),

- Deterministic Approximation (Variational methods).

Stochastic approximation mainly relies on samples to answer queries. Gibbs sampling, importance sampling and Markov Chain Monte Carlo are within this family. On the other hand, deterministic approximations refer to the variational methods, such as mean field approximation, loopy belief propagation, expectation propagation etc. *From the perspectives of methodology, we related work in this thesis locates in the family of variational methods under approximate inference category.*

As for learning in probabilistic graphical models, there are two types of learning problems

- Structure learning,
- Parameter learning.

The first case refer to determine the structure of a graphical model from observation of data, which is usually reduced to the problem of whether there should be an edge between a pair of nodes in the graphical model. The parameter learning is about to determine the parameter of a probabilistic graphical model (or its coupled distribution), with its graphical structure known. Structure learning is out of the scope of this thesis. The term *learning* in thesis means the estimation of the parameters of a distribution. This problem is mainly discussed in Part II, where we would touch the learning of both undirected and directed graphical models.

As for the learning techniques, the learning principles can categorized into

- Maximal likelihood estimation
- Maximal conditional likelihood
- Bayesian estimation
- Maximal ‘Margin‘
- Maximum entropy

in general. We would touch and use techniques of the first four cases in Part II.

Publications

Todo: add code repository

- Dong Liu and Lars K Rasmussen. Region-based energy nerual network for approximate inference. Under review, 2020.
- Dong Liu, Minh Thành Vu, Li Zuxing, and Lars K Rasmussen. α belief propagation for approximate beyeisian inference. Under review, 2020.

- Dong Liu, Antoine Honoré, Saikat Chatterjee, and Lars K Rasmussen. Powering hidden markov model by neural network based generative models. In the 24th European Conference on Artificial Intelligence (ECAI), 2020.
- Dong Liu, Minh Thành Vu, Saikat Chatterjee, and Lars K Rasmussen. Neural network based explicit mixture models and expectation-maximization based learning. arXiv preprint arXiv:1907.13432, 2019.
- Antoine Honoré, Dong Liu, David Forsberg, Karen Coste, Eric Herlenius, Saikat Chatterjee, and Mikael Skoglund. Hidden markov models for sepsis detection in preterm infants. In 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),2020.
- D. Liu, N. N. Moghadam, L. K. Rasmussen, J. Huang, and S. Chatterjee. α belief propagation as fully factorized approximation. In 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 1-5, 2019.
- Dong Liu, Minh Thành Vu, Saikat Chatterjee, and Lars K Rasmussen. Entropy-regularized optimal transport generative models. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3532-3536, 2019.
- Dong Liu, Baptiste Cavarec, Lars K Rasmussen, and Jing Yue. On dominant interference in random networks and communication reliability. In ICC 2019 IEEE International Conference on Communications (ICC), pages 1-7, 2019.
- Dong Liu, Viktoria Fodor, and Lars K Rasmussen. Will scale-free popularity develop scale- free geo-social networks? IEEE Transactions on Network Science and Engineering, 2018.
- Dong Liu, Chao Wang, and Lars K Rasmussen. Discontinuous reception for multiple-beam communication. IEEE Access, pages 46931-46946, 2019.

Outline of Thesis

Tools (code) developed:

Chapter 2

Background

In this chapter, we discuss the background knowledge and introduce notation that will be used for the thesis. We begin with intruding graphical models formally, which is followed with the inference tasks of graphical models. We also discusses the learning principles in model learning. Provide more thorough explanation here after this chapter is finished.

2.1 Graphical Models

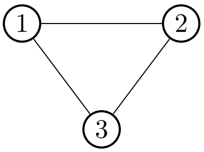
Graphical models provide a formal graph representation of statistical dependency of complex problems. The conditional independence of any random variables can be conveniently analyzed in a graphical model (I-map, d-Separation, markov blanket). More importantly, a intractable complex problem can be resolved by local interactions of small parts of a graphical model.

More formally, a graphical model is a representation of a collection of random variables (along their domains) and nonnegative functions. Let $\boldsymbol{X} = (X_1, X_2, \cdots, X_N)$ be a vector of random variables, where an element variable X_i can be either discrete or continuous and takes values from its domain $\mathcal{X}_i$. Note that domain of a random variable is not necessary be the same as that of another.

We use lower case letters, e.g. $x_i \in \mathcal{X}_i$, to indicate a value assignment of X_i. Similarly, $\boldsymbol{x}$ denotes the assignment of $\boldsymbol{X}$, and $p(x_1, x_2, \cdots, x_N) = p(X_1 = x_1, X_2 = x_2, \cdots, X_N = x_N)$, which sometimes is simplified as $p(\boldsymbol{x}) = p(\boldsymbol{X} = \boldsymbol{x})$. We denote $\boldsymbol{\mathcal{X}} = \prod_{x=1}^{N} \mathcal{X}_i$ and then $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$.

As mentioned in Chapter 1.1, a graphical model can be directed or undirected. Directed graphical model also is also know as Bayesian network or generative model in literature [1, Chapter 8]. We might use the names alternatively. A graphical model over random variable $\boldsymbol{X}$ consists of a set of local nonnegation functions.

In directed graphical models, i.e. Bayesian networks, the local functions are conditional probability function. The joint probability distribution is represented



X_1	X_2	$\varphi(X_1, X_2)$
0	0	10
0	1	1
1	0	1
1	1	10

$\cdots$

X_2	X_3	$\varphi(X_1, X_2)$
0	0	5
0	1	3
1	0	3
1	1	5

Figure 2.1: A Markov random field with three binary nodes. Potential factors are represented by tables.

as product of these conditional probability functions,

$$p(\boldsymbol{x}) = \prod_{n=1}^{N} p(x_n|\mathcal{P}(x_n)), \tag{2.1}$$

where $\mathcal{P}(\cdot)$ denotes the set of parent nodes in the directed graph. In an directed graphical model, the local functions are actually conditional probability distributions, e.g. $\{p(X_n|\mathcal{P}(X_n))\}$, which are properly normalized and sum to one. Additionally, sampling from a underlining distribution $p(\boldsymbol{X})$ of a directed graphical model is efficient. Due to acyclic property of directed graph, by the well know *ancestral sampling*, a sample $\{x_1, x_2, \cdots, x_N\}$ can be drawn sequentially via following the directed edges. In another word, x_n is always sampled after $\mathcal{P}(x_n)$. This process might be viewed as the 'generative' process of signal $\boldsymbol{x}$.

A Bayesian network is usually easier to be interpreted due to the fact that the its local functions are conditional probabilities. But Bayesian networks can only to applied to limited cases where influence between variables is directional. In many practical cases, interaction between variables can not be naturally described by impact with directionality. Problems of this kind can be represented by undirected graphical models, i.e. Markov random field (MRF). Under certain condition, a Bayesian network can be perfectly represented by a Markov random field without loss of independence information by moralizing edges [2, Chapter 4.5]. Instead of conditional probability, the local functions of MRF represents the compatibility of states of different variables, which are known as potential factors in literature. Different from conditional probabilities the Bayesian network, the potential factors in MRFs are not necessary summed to one. We provide a toy example of MRF with three variable nodes as follows.

Example 2. *As shown in Figure 2.1, the MRF encodes dependency of three random variables X_1, X_2, and X_3, where node i is associated with variable X_i and each has a binary domain, i.e. $\mathcal{X}_i = 0, 1$ for $i = 1, 2, 3$. Three potential factors of the MRF together define the joint distribution*

$$p(\boldsymbol{x}) = \frac{1}{Z}\varphi_{1,2}(x_1, x_2)\varphi_{2,3}(x_2, x_3)\varphi_{1,3}(x_1, x_3)$$

where $Z = \sum_{x_1,x_2,x_3} \varphi_{1,2}(x_1, x_2)\varphi_{2,3}(x_2, x_3)\varphi_{1,3}(x_1, x_3)$ normalizes the potential factors such that $p(\boldsymbol{x})$ sums to one. The exemplified potential factors in Figure 2.1 *demonstrate that it is more compatible or likely when X_1, X_2 and X_3 are in the same state (either* 0 *or* 1*) than they are configured into different states.*

From the above example to a formal statement, a MRF over random vector $\boldsymbol{X}$ can be represented by a undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with each node $i \in \mathcal{V}$ is associated with a random variable X_i and undirected edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. This MRF encodes a collection of distributions that factorize as

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{\alpha \in \mathcal{I}} \varphi_\alpha(\boldsymbol{x}_{\boldsymbol{\alpha}}; \boldsymbol{\theta}), \tag{2.2}$$

where $\mathcal{I}$ is the set of indexes of potential factors, and each factor φ_α for $\alpha \in \mathcal{I}$ is defined on subset of $\boldsymbol{X}$, i.e. $\varphi_\alpha : \mathcal{X}_\alpha \to \mathbb{R}^+ \cup \{0\}$, where $\mathcal{X}_\alpha = \prod_{i \in \alpha} \mathcal{X}_i$ is the domain of potential factor φ_α. The scope of factor α is $\boldsymbol{X}_\alpha = \{X_i | i \in \alpha\}$. In (2.2),

$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_{\alpha \in \mathcal{I}} \varphi_\alpha(\boldsymbol{x}_{\boldsymbol{\alpha}}; \boldsymbol{\theta}) \tag{2.3}$$

is the *partition function.* Apparently, the partition function normalizes the potential factors such that $p(\boldsymbol{x}; \boldsymbol{\theta})$ is a proper probability.

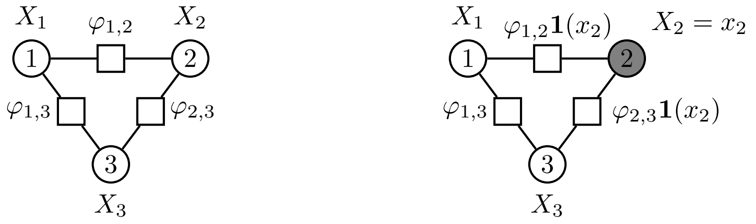
Remark 1. *We can compare directed and undirected graphical models with regarding the following aspects.*

- *Representation: The structure and the parameters directed graphical models provide a natural representation for many types of real-world domains. MRF representation is not usually as intuitive as directed graphical models. But the acyclic property of directed graphical models also limits their representation power. On the other hand, MRFs can be either cyclic or acyclic, which offers the flexibility of graph structure and can simplify the graphical representation. Due to the requirement is weaker for potential factors than that for conditional distribution, the representation of MRFs are richer.*

- *Local nonnegative functions: The local functions are conditional probability functions in directed graphical models, but are potential factors (nonnegative) in undirected cases.*

- *Sampling: Sampling is more straightforward within generative models (directed) than that in MRFs.*

- *Normalization: Since each local function is a conditional probability function in directed graphical models, partition function for normalization is not needed. A MRF in general comes with partition functions, since the requirement on potential factors are weaker than a conditional probability distribution.*



(a) Factor graph representation.

(b) Conditioning on $X_2 = x_2$

Figure 2.2: A Markov random field is represented by a factor graph, i.e. 2.2a, and conditioning of the Markov random field 2.2b.

Alternative Representation of MRF

The representation of a MRF by $\mathcal{G}(\mathcal{V}, \mathcal{E})$ as explained above is compact, but the potential factors are missing. An alternative representation of MRF is *factor graph* [3], which is a bipartite graph topology. In a factor graph, a potential factor is explicitly represented as a factor node, as counterpart of variable node associated with a random variable.

Definition 1. *A factor graph $\mathcal{G}_F$, is a bipartite graph that represents the factorization structure of* (2.2)*. A factor graph has two types of nodes: i) a variable node for each variable x_i; ii) a factor node for each potential function φ_α. An edge between a variable node i and factor node α if and only if x_i is argument of φ_α. We would denote a factor graph by $\mathcal{G}_F(\mathcal{V} \cup \mathcal{F}, \mathcal{E}_F)$ with $\mathcal{V}$ as the set of variable nodes, $\mathcal{F}$ as the set of factor nodes, and $\mathcal{E}_F$ the set of undirected edges.*

Example 3. *Let us represent the Example 2.2a by a factor graph, which is shown in Figure 2.2a. Different from the representation by $\mathcal{G}(\mathcal{V}, \mathcal{E})$ in Figure 2.1, factor nodes are explicitly represented by square nodes.*

Conditioning on Observations in MRFs

It is not rare that a graphical model may contain observed variable. The node set of a MRF can be separated into a subset $\mathcal{V}_O$ of nodes, that are associated with observed variable $\boldsymbol{X}_O$, and a subset $\mathcal{V}_U$ of nodes associated with unobserved variable $\boldsymbol{X}_U$. For an observation $\boldsymbol{X}_O = \boldsymbol{x}_O$,

$$p(\boldsymbol{x}_U | \boldsymbol{x}_O; \boldsymbol{\theta}) = \frac{p(\boldsymbol{x}; \boldsymbol{\theta})}{p(\boldsymbol{x}_O; \boldsymbol{\theta})} = \frac{Z(\boldsymbol{x}_O, \boldsymbol{\theta})}{Z(\boldsymbol{\theta})}, \tag{2.4}$$

where

$$\begin{aligned}\tilde{p}(\boldsymbol{x};\boldsymbol{\theta}) &= \prod_{\alpha\in\mathcal{I}} \varphi_\alpha(\boldsymbol{x}_\alpha;\boldsymbol{\theta}),\\ Z(\boldsymbol{x}_O,\boldsymbol{\theta}) &= \sum_{\boldsymbol{x}_U} \tilde{p}(\boldsymbol{x};\boldsymbol{\theta}),\\ Z(\boldsymbol{\theta}) &= \sum_{\boldsymbol{x}_O}\sum_{\boldsymbol{x}_U} \tilde{p}(\boldsymbol{x};\boldsymbol{\theta}).\end{aligned} \tag{2.5}$$

This means that a condition probability can be computed by partition function and sub-partition functions. Alternatively, the conditional probability can be written as

$$p(\boldsymbol{x}_U|\boldsymbol{x}_O;\boldsymbol{\theta}) = \frac{\tilde{p}(\boldsymbol{x};\boldsymbol{\theta})\mathbf{1}(\boldsymbol{x}_O)}{\sum_{\boldsymbol{x}_U}\tilde{p}(\boldsymbol{x};\boldsymbol{\theta})\mathbf{1}(\boldsymbol{x}_O)} \tag{2.6}$$

where $\mathbf{1}(\boldsymbol{x}_O)$ is an indicator function and equals to one if and only if the states of nodes in $\mathcal{V}_O$ are jointly $\boldsymbol{x}_O$. This can be understood as clamping nodes in $\mathcal{V}_O$ of the MRF to configuration $\boldsymbol{x}_O$, i.e. the domain of $\boldsymbol{X}_O$ becomes a set containing only $\boldsymbol{x}_O$. (For instance, an example of conditioning on a variable node is shown in Figure 2.2b.) Then any inference applicable to a MRF applies to the MRF with nodes clamped as well.

In addition to the above intuitions, conditioning can also be understood as a process of reducing the graph of a MRF. When a MRF is conditioned on $\boldsymbol{x}_O$, the variables nodes of set $\mathcal{V}_O$ are removed from $\mathcal{G}$, along with their edges. The potential factors with regarding to $\mathcal{V}_U$ are modified accordingly [2, Chapter 4.2.3].

It can be seen that MRF framework is capable to handle conditioning as well. Therefore, in the following part of the thesis, it might or might not have bee based on conditioning observed variables when a MRF is mentioned.

2.2 Divergence

Before we get into more discussion about inference and learning topics, we firstly introduce the concept of *divergence* measures since principles of both learning and inference are closed related with divergence measure. A divergence measure play a fundamental role when we try to use a probability density (or mass for discrete cases) function q to approximate another probability density function p. A divergence measure is used to formal quantity how much information is lost when p is represented by q. Denote $\mathbb{P}$ as the space of measures p and q, i.e. $p, q \in \mathbb{P}$.

Definition 2. *Given the space of probability density (or mass) function $\mathbb{P}$ for a random variable $\boldsymbol{x}$, a divergence on this space is defined as a function $D(p\|q) : \mathbb{P}\times\mathbb{P}\to\mathbb{R}$ such that $D(p\|q) \geqslant 0$ for all $p, q \in \mathbb{P}$ and $D(p\|q) = 0$ if and only if $p = q$.*

Here we introduce the classic *Kullback-Leibler divergence* [4, 5], KL divergence for short, which is one of the most widely used divergence measures in machine learning, statistics and information theory.

Definition 3. *The Kullback-Leibler (KL) divergence on* $\mathbb{P}$ *is defined as a function* $KL(\cdot\|\cdot) : \mathbb{P} \times \mathbb{P} \to \mathbb{R}^+ \cup 0$ *with the following form*

$$\mathrm{KL}(p\|q) = \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}, \tag{2.7}$$

where $\log$ *is the natural loggarithm. Note the sum in 2.7 should be replaced by intergral when* p *and* q *are probability density functions.*

KL divergence is not symmetric. In another word, there is no equivalence between $\mathrm{KL}(p\|q)$ and $\mathrm{KL}(q\|p)$ in general.

2.3 Inference Tasks

Given a probability distribution $p(\boldsymbol{x})$ as the underline distribution of a graphical model, inference in general can be generally divided into four kinds of tasks, as brought up in Chapter 1.2. Our work in this thesis would closed involved with the problems

- computing the likelihood of observed data or unobserved random variable;
- computing the marginals distribution over a particular subset of nodes, i.e. $p(x_A)$ for $A \in \mathcal{V}$. Note that single-node marginal distribution $p(x_i)$ also belongs to this case;
- computing the conditional distribution a subset of nodes given the configuration of another subset of nodes, i.e. $p(\boldsymbol{x}_A|\boldsymbol{x}_B)$ for $A, B \in \mathcal{V}$ and $A \cap B = \emptyset$;

in MRFs. The above tasks are also close related with the inference of partition function

- Computation of $Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_{\alpha \in \mathcal{I}} \varphi_\alpha(\boldsymbol{x}_{\boldsymbol{\alpha}}; \boldsymbol{\theta})$, or sub-partition functions.

2.4 Variational inference

fix notation and consistency in this section

In solving inference tasks, one important technique is based on a variational approach. With $p(\boldsymbol{x})$ as the underlining probability distribution of a graphical model, directly inference with $p(\boldsymbol{x})$ is often unfeasible due to the system represented by the graphical model is too large or complex (Even we know the form of $p(\boldsymbol{x})$, the computation in inference tasks can be prohibitive). In variational approach, a 'trial' probability distribution $b(\boldsymbol{x})$ is introduced to approximate $p(\boldsymbol{x})$. The trial distribution should be intuitively simpler than $p(\boldsymbol{x})$. *Variational free energy* [8] is used to find such a approximation, which uses a probability distribution $b(\boldsymbol{x})$ to approximate $p(\boldsymbol{x};\boldsymbol{\theta})$. The variational free energy is defined by

$$\begin{aligned} F_V(b) &= KL(b(\boldsymbol{x})||p(\boldsymbol{x};\boldsymbol{\theta})) - \log Z(\boldsymbol{\theta}) \\ &= \sum_{\boldsymbol{x}} b(\boldsymbol{x}) \ln \frac{b(\boldsymbol{x})}{p(\boldsymbol{x};\boldsymbol{\theta})} - \ln Z(\boldsymbol{\theta}) \\ &= \sum_{\boldsymbol{x}} b(\boldsymbol{x}) \ln \frac{b(\boldsymbol{x})}{\tilde{p}(\boldsymbol{x};\boldsymbol{\theta})} \end{aligned} \tag{2.8}$$

where $\tilde{p}(\boldsymbol{x};\boldsymbol{\theta}) = \prod_{\alpha\in\mathcal{I}} \psi_\alpha(\boldsymbol{x}_\alpha;\boldsymbol{\theta}_\alpha)$. Since $\mathrm{KL}(b(\boldsymbol{x})\|p(\boldsymbol{x};\boldsymbol{\theta}))$ is always non-negative and is zero if and only if $b(\boldsymbol{x}) = p(\boldsymbol{x};\boldsymbol{\theta})$, we have $F_V(b) \geqslant -\log Z(\boldsymbol{\theta})$, with equality when $b(\boldsymbol{x}) = p(\boldsymbol{x};\boldsymbol{\theta})$.

Variational Free Energy and Mean Field

In mean field approach, a fully-factorized approximation is used,

$$b_{MF}(\boldsymbol{x}) = \prod_{i=1}^{n} b_i(x_i). \tag{2.9}$$

Substituting (2.9) into the variational free energy gives

$$\begin{aligned} F_{MF} = &- \sum_{a} \sum_{\boldsymbol{x}_\alpha} \ln \psi_\alpha(\boldsymbol{x}_a;\boldsymbol{\theta}) \prod_{i\in\mathrm{ne}_\alpha} b_i(x_i) \\ &+ \sum_{i} \sum_{x_i} b_i(x_i) \ln b_i(x_i), \end{aligned} \tag{2.10}$$

where $\mathrm{ne}_\alpha = \{i \in \mathcal{V} | x_i \in \mathcal{S}(a)\}$ gives the neighboring variable nodes of factor a, and $\mathcal{S}(a)$ is the scope set (arguments) of factor node a. Solving the minimization of F_{MF} w.r.t. $b_{MF}(\boldsymbol{x})$ gives the update rule of mean field

$$\ln b_i(x_i) \sim \sum_{a\in\mathrm{ne}_i} \sum_{\boldsymbol{x}_\alpha\backslash x_i} \ln \psi_\alpha(\boldsymbol{x}_\alpha;\boldsymbol{\theta}_\alpha) \prod_{j\in\mathrm{ne}_\alpha\backslash i} b_j(x_j), \tag{2.11}$$

where $\mathrm{ne}_i = \{a | i \in \mathcal{S}(a), a \in \mathcal{F}\}$, i.e. the neighboring factors of node i.

Bethe Free Energy and (Loopy) Belief Propagation

Different from the mean field approximation, Bethe approximation also includes the non-single-node beliefs $\{b_\alpha(\boldsymbol{x}_\alpha)\}$ apart from the single-node beliefs $\{b_i(x_i)\}$ [14]. In this case, the Bethe free energy is given by

$$\begin{aligned} F_{Bethe} =& \sum_a \sum_{\boldsymbol{x}_\alpha} b_\alpha(\boldsymbol{x}_\alpha) \ln \frac{b_\alpha(\boldsymbol{x}_\alpha)}{\psi_\alpha(\boldsymbol{x}_\alpha)} \\ &- \sum_{i=1}^{N} (|\mathrm{ne}_i| - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i). \end{aligned} \tag{2.12}$$

Due to the non-single-node beliefs, there are consistency constrains $\sum_{\boldsymbol{x}_\alpha} b_\alpha(\boldsymbol{x}_\alpha) = \sum_{x_i} b_i(x_i) = 1$, $\forall\ i \in \mathcal{S}(a)$ to obey. Then, solving the Bethe free energy minimization problem

$$\begin{aligned} \min_{\{b_\alpha(\boldsymbol{x}_\alpha)\},\{b_i(x_i)\}} \quad & F_{Bethe} \\ \text{s.t.} \quad & \sum_{\boldsymbol{x}_\alpha \backslash x_i} b_\alpha(\boldsymbol{x}_\alpha) = b_i(x_i), \\ & \sum_{\boldsymbol{x}_\alpha} b_\alpha(\boldsymbol{x}_\alpha) = \sum_{x_i} b_i(x_i) = 1, \\ & 0 \leqslant b_i(x_i) \leqslant 1, \\ & b_\alpha(\boldsymbol{x}_\alpha) \in [0,1]^{|S(\boldsymbol{x}_\alpha)| \times K}, \\ & i \in \mathcal{V}, a \in \mathcal{F}, \end{aligned} \tag{2.13}$$

where $\mathcal{V}$ and $\mathcal{F}$ are the set of variable nodes and the set of factor nodes in factor graph as defined in Definition 1, gives the (loopy) BP message-passing rule

$$m_{a \to i}(x_i) \sim \sum_{\boldsymbol{x}_\alpha \backslash x_i} \psi_\alpha(\boldsymbol{x}_\alpha) \prod_{j \in \mathcal{S}(a) \backslash i} \prod_{b \in \mathrm{ne}_j \backslash a} m_{b \to j}(x_j). \tag{2.14}$$

2.5 Learning principles

We have touched the learning topic in chapter 1, which is to find the 'best' probability distribution $p(\boldsymbol{x};\boldsymbol{\theta})$ in its space $\mathbb{P}$. To make the discussion more concrete, we assume the domain is governed by a underlying distribution p^* that is induced by a (directed or undirected) graphical model, $\mathcal{M} = \{\mathcal{K}^*, \boldsymbol{\theta}^*\}$ with $\mathcal{M}^*$ representing its structure and $\boldsymbol{\theta}^*$ representing its parameter. Here we discuss about *model learning* (parameter learning only). For notation simplicity, we use $p^*(\boldsymbol{x})$ to denote this distribution. We are given a dataset $\mathcal{D} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^M\}$. Following the standard assumption, these sample instances are *independent and identically distributed (i.i.d.)*. The task is then to use the information from the dataset to learn a distribution p within its space $\mathbb{P}$, since the governing distribution $p^*(\boldsymbol{x})$ is not known.

The problem of learning a distribution in $\mathbb{P}$ to approximate p^* can be formulated as density estimation. With the concept of KL divergence in section 2.2, learning of p can be formulated as minimizing the KL divergence

$$\begin{aligned}
&\mathrm{KL}(p^*(\boldsymbol{x})\|p(\boldsymbol{x};\boldsymbol{\theta})) \\
=&\mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log\frac{p^*(\boldsymbol{x})}{p(\boldsymbol{x};\boldsymbol{\theta})}\right] \\
=&-H(p^*)-\mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p(\boldsymbol{x};\boldsymbol{\theta})\right],
\end{aligned} \tag{2.15}$$

where $H(p^*)$ is the entropy of p^*. Due to the property of divergence, the KL divergence in 2.15 is zero if and only if $p(\boldsymbol{x};\boldsymbol{\theta}) = p^*(\boldsymbol{x})$. The last line of 2.15 shows that the negative entropy term does not depends on $p(\boldsymbol{x};\boldsymbol{\theta})$. Thus we can just focus on the expectation term $\mathbb{E}_{\boldsymbol{x}\sim p^*}[\log p(\boldsymbol{x};\boldsymbol{\theta})]$, which is *expected log-likelihood.* Therefore, we can just use the expected log-likelihood to do model learning instead of minimizing the KL divergence.

Note although we can use the expected log-likelihood for model learning task and even model comparison (comparing a trained model with another one), we loss the information of how close a trained model is to p^*. This is due to the omitting of $H(p^*)$, which is not available.

Since it is not possible to know p^* (otherwise we do not need to learn it), the expected log-likelihood is approximated by sample instances of p^*,

$$\mathcal{L}(\mathcal{D};\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\log p(\boldsymbol{x};\boldsymbol{\theta}), \tag{2.16}$$

and

$$\mathbb{E}_{\boldsymbol{x}\sim p^*}(\log p(\boldsymbol{x};\boldsymbol{\theta})) \approx \mathcal{L}(\mathcal{D};\boldsymbol{\theta}). \tag{2.17}$$

Log-likelihood $\mathcal{L}(\mathcal{D};\boldsymbol{\theta})$ is one of the most widely used loss for model learning. However, $\mathcal{L}(\mathcal{D};\boldsymbol{\theta})$ is not always a feasible loss to compute due to:

- exact computation of $p(\boldsymbol{x})$ is not possible;
- there are some elements of $\boldsymbol{x}$ which are not observable (hidden variables).

For the first case, the typical treatment is to approximate the exact log-likelihood. This is done by approximation with employing inference methods or making simplified assumptions on dependency structure of the graphical model of $p(\boldsymbol{x})$. Then, optimization is carried out with regarding to the approximated log-likelihood. These methods include surrogate likelihood [7, 12], pseudo-likelihood [9], piecewise likelihood [6, 11], saddle-point approximation [10, 13].

For the second case that there are hidden variables, 1. EM
2. variational EM
3. a bit Bayesian ?
1. begin with KL minimization, refer to [2, chapter16]
2. talk a bit of others learning objectives, refer to Domke 2013 paper.

Learning of Full Observation

the learning diagram here

- Structural learning
- parameter learning

the learning principle:

- Maximal likelihood estimation (MLE)
- Bayesian estimation
- Maximal conditional likelihood
- Maximal " 'Margin'"
- Maximum entropy

It may be better to discuss the learning principle here.
Cited from 10-708 lecture6 note:
UNOBSERVED VARIABLES:
A variable can be unobserved or latent because it is a(n):

-Abstract or imaginary quantity meant to simplifiy the data generation process, e.g. speech recognition models, mixture models. -A real-world object that is difficult or impossible to measure, e.g. the temperature of a star, causes of disease, evolutionary ancestors. -A real-world object that was not measured due to missed samples, e.g. faulty sensors.

Discrete latent variables can used to partition or cluster data into sub-groups

Continuous latent variables (factors) can be used for dimensionality reduction (e.g. factor analysis, etc)

Dealing with latent variables

about clamping node

clamping node gives conditional distribution.

about ELBO bound

1. the bound used by EM

2. talk about ELBO/variational inference Variational Inference, which is closely related the bound used in EM.

Part I

Inference

Chapter 3

An alternative view of belief propagation

Content:

1. α Belief Propagation as Fully Factorized Approximation, GlobalSIP 2019.
2. α Belief Propagation for Approximate Bayesian Inference, under review.

3.1 α belief propagation

3.2 Convergence study

3.3 Experimental results

3.4 Summary

Chapter 4

Region-based Energy Neural Network Model

work in Region-based Energy Neural Network for Approximate Inference, under, review

4.1 Region-based graph and energy

4.2 RENN model for Approximate Inference

4.3 RENN model for markov random field training

4.4 Experimental results

4.5 Summary

Part II

Learning

Chapter 5

Learning with inference

5.1 learning Undirected graphical models/ MRF

move the MRF learning by using RENN here

I should read lecture note 7 of 10-708 again when writing this seciton.

5.2 Amortized/Neural Variational Learning and Inference of partial observed MRF

1. TRW as upper bound to partition function

2. Mean field or negative TRW as lower bound to partition function

combining above together, we can obtain two different lower bound of likelihood. Consider if worthy a paper.

- The log-likelihood of partial observed MRF is non-convex in general (log-sum-exp is convex, but the difference of two log-sum-exp functions might not be). This combination convert the original non-convex learning into convex optimization with regarding to MRF parameter? should be, but need a confirmation.
- 1. The speed of training can be improved by directly optimizing amortized beliefs.
- The bound becomes tighter by using clamping of variable, clamping can be done with or without selection of variables. No sampling is needed in training or inference.
- If need more contribution, use tree-reweighted hyper graph to obtain tighter bound.
- Not necessarily done here: the bound can also be further improved by important sampling.

Reference:

- 1. Wainwright, 2003, Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching
- 2. Weller, 2015,Clamping Improves TRW and Mean Field Approximations
- 3. Mnih, 2014, Neural Variational Inference and Learning in Belief Networks, which describes a neural variational method for belief network. The major difference is the belief network as a DAG do not have the problem of partition function difficulty as MRF or partial observed MRF.

5.3 Notation

Random variable $\boldsymbol{v} \in \mathcal{X}_v$ that can be observed. Random variable $\boldsymbol{h} \in \mathcal{X}_h$ that is hidden variable and can not be observed.

An alternative plan:

- Training RENN with marginal-likelihood instead of joint likelihood, ref to Domke13
- If the above works, use Gaussian kernals to define potential, Marvin T. T. Teichmann Convolutional CRFs for Semantic Segmentation

5.4 Model and Problem Definition

We define the conditional probabilistic model as

$$p(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \tilde{p}(\boldsymbol{v}, \boldsymbol{h} | \boldsymbol{\theta}), \tag{5.1}$$

with

$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \tilde{p}(\boldsymbol{v}, \boldsymbol{h} | \boldsymbol{\theta}) \tag{5.2}$$

$$\tilde{p}(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta}) = \exp\left\{-E(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{\theta})\right\} \tag{5.3}$$

where $E(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta})$ is the average energy: $\mathcal{X}_v \times \mathcal{X}_h \to \mathbb{R}$.

We want to maximize the marginal likelihood:

$$\max_{\boldsymbol{\theta}} \log \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \log\ Z(\boldsymbol{v}, \boldsymbol{\theta}) - \log\ Z(\boldsymbol{\theta}), \tag{5.4}$$

where $Z(\boldsymbol{v}, \boldsymbol{\theta}) = \sum_{\boldsymbol{h}} \tilde{p}(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta})$

5.5 A lower bound of the marginal likelihood

Denote $A(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta})$ and $A(\boldsymbol{v}, \boldsymbol{\theta}) = \log Z(\boldsymbol{v}, \boldsymbol{\theta})$

$$E(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{\theta}) = -\langle \boldsymbol{\theta}, \boldsymbol{\varphi}(\boldsymbol{v}, \boldsymbol{h}) \rangle \tag{5.5}$$

and

$$\boldsymbol{\mu} = \mathbb{E}_{p(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta})}[\boldsymbol{\varphi}(\boldsymbol{v}, \boldsymbol{h})]. \tag{5.6}$$

In case of overcomplete representation of $\boldsymbol{\varphi}$, $\boldsymbol{\mu}$ is the set of marginal distributions.

With mean field approximation,

$$A_M(\boldsymbol{v}, \boldsymbol{\theta}) = \max_{\boldsymbol{\mu}_{\boldsymbol{v}} \in \mathcal{M}_M} \langle \boldsymbol{\theta}, \boldsymbol{\mu}_{\boldsymbol{v}} \rangle + H_M(\boldsymbol{\mu}_{\boldsymbol{v}}), \tag{5.7}$$

where $\mathcal{M}_M$ is the subspace of distributions where each variable is independent. And we have

$$A_M(\boldsymbol{v}, \boldsymbol{\theta}) \leqslant A(\boldsymbol{v}, \boldsymbol{\theta}). \tag{5.8}$$

With tree-reweighted approximation, TRW,

$$A_T(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu} \in \mathcal{M}_T} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle + H(\boldsymbol{\mu}), \tag{5.9}$$

where $\mathcal{M}$ is the subspace of distributions where each variable is independent. And we have

$$A_T(\boldsymbol{\theta}) \geqslant A(\boldsymbol{\theta}). \tag{5.10}$$

We define the lower bound of marginal loglikelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = A_M(\boldsymbol{v}, \boldsymbol{\theta}) - A_T(\boldsymbol{\theta}) \leqslant \log \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta}). \tag{5.11}$$

Connection to RBM:

$$p(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\{\boldsymbol{vWh} + \boldsymbol{vb} + \boldsymbol{va}\} \tag{5.12}$$

Note $p(\boldsymbol{h}|\boldsymbol{h})$ is exactly independent, and thus the $A_M(\theta) = A(\boldsymbol{\theta})$ can be achieved, then how tight the lower bound $\mathcal{L}(\boldsymbol{theta})$ would dependent only on the TRW bound.

I should also consider how to use the trained model for prediction.

This is also closely connected to variational see Section 6.2, Wainwright, Graphical Models, Exponential Families, and Variational Inference.

5.6 Experiment

- Start with standard RBM section 4.2 in Amortized learning of MRFs
    - Try to break the conditinal independence by connecting nodes of $\boldsymbol{h}$
    - Extent to conditional RBM training for denoising and data completion
- high-order HMMs

Chapter 6

Powering the expectation maximization method by neural networks

content: Neural Network based Explicit Mixture Models and Expectation-maximization based Learning, under review

section/chapter transition text: mixture model could be obtained from clamping and condition on a discrete variable, ref to Geier, Locally conditional belief propagation. Weller, clamping variables and approximate inference

Remark 2. *Theory interpretation of EM and variational EM, see Section 6.2, Wainwright, Graphical Models, Exponential Families, and Variational Inference.*

Remark 3. *RELATIONSHIP TO K-MEANS CLUSTERING Big picture: The EM algorithm for mixtures of Gaussians is like a soft version of the K-means algorithm.*

Remark 4. *EM lower bound + entropy of posterior of latent variable if a free energy. ref to 10-708 lecture6 note. EM using posterior of latent variable is equivalent to fully observable MLE where statistics are replaced by their expectations w.r.t the posterior.*

Can be viewed as two-node graphical model learning. 10-708lecture5-note

6.1 Normalizing flow

6.2 expectation maximization of neural network based mixture models

6.3 An alternative construction method

6.4 Experiments

6.5 Summary

Chapter 7

Powering Hidden Markov Model by Neural Network based Generative Models

For permeable part and notation of this chapter, refer to [2, Chapter 6.2]. Give a figure/illustration: Dynamic Bayesian Network > 2-TBN > HMM

A bit history of HMM, see [2, Chapter 6.8]

content:

1. Powering Hidden Markov Model by Neural Network based Generative Models, ECAI 2020

2. Antoine Honore, Dong Liu, Hidden Markov Models for sepsis detection in preterm infants, ICASSP, 2020

HMM is an instance of 2-time-slice Bayesian network(2-TBN) (section 6.2.2 Koller). Also, it can be argued from CRF.

7.1 Hidden Markov Model

7.2 GenHMM

7.3 Application to phone recognition

7.4 Application to sepsis detection in preterm infants

7.5 Summary

Chapter 8

An implicit probabilistic generative model

content: Entropy-regularized Optimal Transport Generative Models, ICASSP 2019

8.1 Modeling data without explicit probabilistic distribution

8.2 Employing EOT for modeling

8.3 Experimental results

8.4 Summary

Part III

Epilogue

Chapter 9

Conclusion and Discussions

Bibliography

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[2] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193.

[3] F. R. Kschischang, B. J. Frey, and H. . Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, Feb 2001. ISSN 0018-9448.

[4] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951. URL `https://doi.org/10.1214/aoms/1177729694`.

[5] Solomon Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

[6] Guosheng Lin, Chunhua Shen, Anton van den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] You Lu, Zhiyuan Liu, and Bert Huang. Block belief propagation for parameter learning in markov random fields. *CoRR*, abs/1811.04064, 2018. URL `http://arxiv.org/abs/1811.04064`.

[8] M. Opper and D. Saad. *Advanced Mean Field Methods: Theory and Practice*. Neural information processing series. MIT Press, 2001. ISBN 9780262150545.

[9] Meng Qu, Yoshua Bengio, and Jian Tang. Gmnn: Graph markov neural networks. In *International Conference on Machine Learning*, pages 5241–5250, 2019.

[10] Vivek Srikumar, Gourab Kundu, and Dan Roth. On amortizing inference cost for structured prediction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural*

Language Learning, pages 1114–1124, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[11] Charles A. Sutton and Andrew McCallum. Piecewise training for undirected models. *CoRR*, abs/1207.1409, 2012. URL `http://arxiv.org/abs/1207.1409`.

[12] Martin J. Wainwright. Estimating the "wrong" graphical model: Benefits in the computation-limited setting. *J. Mach. Learn. Res.*, 7:1829–1859, 2006. URL `http://jmlr.org/papers/v7/wainwright06a.html`.

[13] Sam Wiseman and Yoon Kim. Amortized bethe free energy minimization for learning mrfs. In *Advances in Neural Information Processing Systems 32*, pages 15520–15531. Curran Associates, Inc., 2019.

[14] Jonathan Yedidia, William Freeman, and Yair Weiss. *Understanding belief propagation and its generalizations*, volume 8, pages 239–269. 01 2003. ISBN 1558608117.