Some notes on Pattern Recognition and Machine Learning

Yuandong Tian

April 17, 2008

1 Graphical Models

1.1 Why Graphical Models? Part 1

1) A powerful tool for modeling dependencies among random variables.

2) *An efficient way to do "probability inference".* → Factorization

1.2 Classification of Graphical Models

There are two kinds of graphical models:

1) Directed Acyclic Graph(DAG)

2) Undirected graphical models

1.3 Directed Acyclic Graph(Bayesian Network)

1.3.1 Basic Ideas

An easy sample(See 1). The corresponding joint probability function is:

$$\begin{aligned} & p(x_1, x_2, x_3, x_4, x_5, x_6, x_7) \\ = \; & p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_6|x_3, x_4)p(x_5)p(x_7|x_1, x_4, x_5) \end{aligned} \tag{1}$$

1.3.2 Formal Definition

Generally, a Bayesian Network represents a high-dimensional joint probability function $p(\mathbf{x})$ $\mathbf{x} = [x_1, x_2, \dots x_n]$ so that

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\mathrm{pa}_i) \tag{2}$$

pa_i is the parent nodes of node i, i.e., the nodes pointing to node i.

Note: From Eqn. 2, we can have an idea what the term "factorization" is. The characteristics of Bayesian Network:

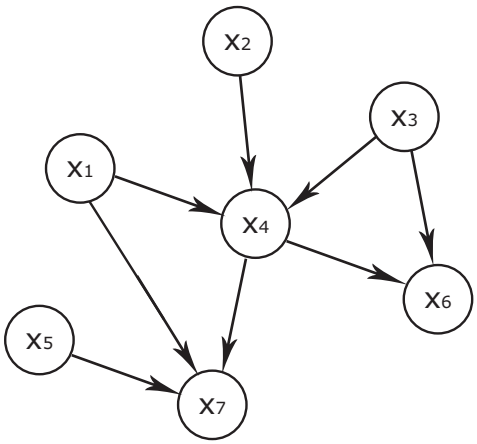

Figure 1: A simple example of Bayesian Network

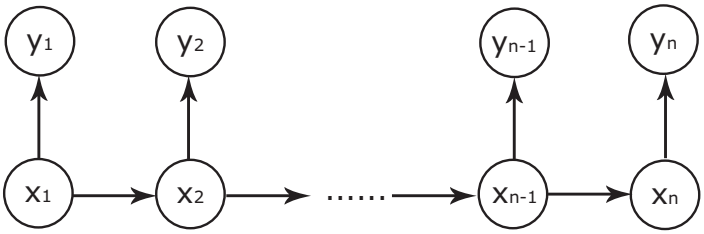

Figure 2: Hidden Markov Model

1) An intuitive way for modeling → All identities appearing in Bayesian Network have intuitive meanings.

2) A less insightful way for solution. → What does "probability inference" really means? Find Max/Min/Sum on Functions!

1.3.3 Typical Example: Hidden Markov Chain

See Fig. 2. Joint probability function: (set $p(x_1|x_0) = p(x_1)$)

$$p(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{n} p(x_i|x_{i-1})p(y_i|x_i) \tag{3}$$

1.4 Undirected graphical model

1.4.1 Basic Ideas

An easy sample (See 3).

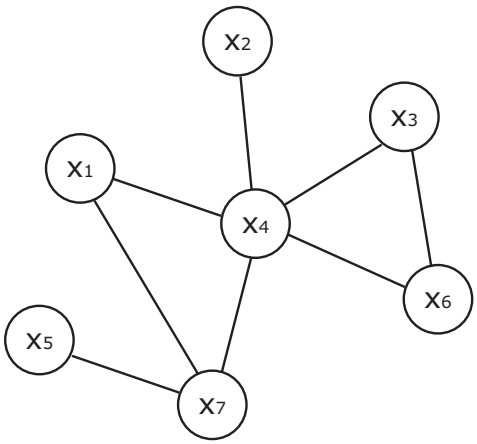

Figure 3: A simple example of Undirected graphical model. It is DIFFERENT from Bayesian Network, although they look like the same.

The corresponding joint probability function is:

$$\begin{aligned} & p(x_1, x_2, x_3, x_4, x_5, x_6, x_7) \\ = \; & p(x_1, x_4, x_7) p(x_3, x_4, x_6) p(x_2, x_4) p(x_5, x_7) \end{aligned} \tag{4}$$

Note: It is DIFFERENT from Bayesian Network, although they look like the same.

1.4.2 Formal Definition

The joint probability of an undirected graphical model is:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{\text{all maximal clique } C} \phi_C(\mathbf{x}_C) \tag{5}$$

How to get all C.(See Algorithm 1.4.2)

---

while Graph G still has edge(s) **do**
  Find one maximal *clique* C. Warning: This step potentially takes exponential time;
  Pick it out, delete all the edges of C (but not its node!)
  Add it to Eqn. 5.
end while

$\phi_C(\mathbf{x}_C)$: Potential function defined on C, $\mathbf{x}_C$ is the corresponding variables of C. Warning: $\phi_C(\mathbf{x}_C)$ is not necessarily probability function; any function works.

$Z = \sum_{\mathbf{x}} \prod_C \phi_C(\mathbf{x}_C)$: Partition function.

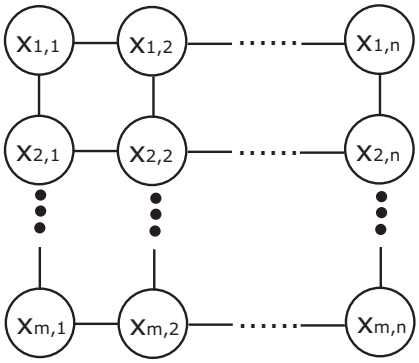

Figure 4: Markov Random Field.

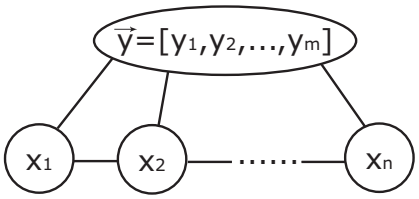

Figure 5: Conditional Random Field.

1.4.3 Typical example: MRF & CRF

See Fig. 4 for Markov Random Field and Fig. 5 for Conditional Random Field.

The joint probability of MRF is:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{i \text{ is neighbor of } j} \phi_{ij}(x_i, x_j) \tag{6}$$

It has a maximal clique of 2.

The conditional probability of CRF is:

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{Z} \prod_{i=2}^{n} \phi_i(x_i, x_{i-1}, \mathbf{y}) \tag{7}$$

It has a maximal clique of 3. Note: CRF only model the conditional probability rather than joint probability.

1.5 Why Graphical Models? Part 2

Why graphical models? Why not just have one $p(\mathbf{x})$ rather than a complicated *Factorization* of small functions?

The answer is: Factorization is efficient in computation.

We know HMM(See Eqn. 3) is efficiently solved by *Dynamic Programming*. Then ..., what is Dynamic Programming?

Taken HMM as an example, it is nothing but the following trick:

$$\begin{aligned} p(\mathbf{x}^*, \mathbf{y}) &= \max_{\mathbf{x}} p(\mathbf{x}, \mathbf{y}) \\ &= \max_{\mathbf{x}} \prod_{i=1}^{n} p(x_i|x_{i-1}) p(y_i|x_i) \\ &= \max_{x_n} p(y_n|x_n) \max_{x_{n-1}} p(y_{n-1}|x_{n-1}) p(x_n|x_{n-1}) \\ \dots &\quad \max_{x_2} p(y_2|x_2) p(x_3|x_2) \max_{x_1} p(x_2|x_1) p(y_1|x_1) p(x_1) \end{aligned} \tag{8}$$

If each x_i can take k possible values, then finding maximum of Eqn. 9 by brute-force takes $O(k^n)$, an impractical method. But if we define:

$$\begin{aligned} f_1(x_1) &= p(x_1) \\ f_2(x_2) &= \max_{x_1} f_1(x_1) p(y_1|x_1) p(x_2|x_1) \\ f_3(x_3) &= \max_{x_2} f_2(x_2) p(y_2|x_2) p(x_3|x_2) \\ &\dots \\ f_n(x_n) &= \max_{x_{n-1}} f_{n-1}(x_{n-1}) p(y_{n-1}|x_{n-1}) p(x_n|x_{n-1}) \\ p(\mathbf{x}^*, \mathbf{y}) &= \max_{x_n} f_n(x_n) p(y_n|x_n) \end{aligned} \tag{9}$$

it only takes $O(nk^2)$! Note that all max notation can be replaced by $\sum$ so that one can evaluate the marginalized distribution.

That's the power of factorization, that's essentially the reason why graphical model is so useful.

1.6 Factor Graph and Belief Propagation (BP)

Basic concept: "Belief Propagation" is nothing but Dynamic Programming, i.e.,change the order of min/max/sum to efficiently optimize the objective functions.

The basic principle is:

1) For directed acyclic graph (Bayesian Network), it guarantees to give global optimal solution (or "exact inference"). Yet how fast it is depends on the maximum number of arguments in functions $P(x_i|\text{pa}_i)$. The more arguments, the slower it will be.

2) For other situations, first convert the graph to *Factor Graph*.

   a) If the factor graph is loop-free, then exact inference can be performed by Belief Propagation. The time complexity of the algorithm also

depends on the maximum number of arguments in function nodes of Factor Graph.

b) In loopy case, it cannot have exact inference by Belief Propagation. Yet, one can eliminate loopy case by "merging" variables together and apply BP (See Fig. 6). This will (substantially) increase the computational complexity.

We can see why factorization is so important from highlight words.

Here (See Fig. 6)is an easy sample for conversion to factor graph. The key idea is: See the functions!.

I will not give the details of Belief Propagation. Here is a brief introduction.

If the problem visualized by the factor graph can be stated in the following:

$$f(\mathbf{x}) = \sum_{\alpha} f_{\alpha}(\mathbf{x}_{\alpha}) \tag{10}$$

And what we want is to find its maximal value $f(\mathbf{x}^*) = \max_{\mathbf{x}} f(\mathbf{x})$. Then Belief Propagation is:

$$m_{\alpha \to i}(x_i) = \max_{\mathbf{x}_{\alpha} \setminus x_i} f_{\alpha}(\mathbf{x}_{\alpha}) \sum_{j \in \mathcal{N}(\alpha) \setminus i} m_{j \to \alpha}(x_j) \tag{11}$$

$$m_{i \to \alpha}(x_i) = \sum_{\beta \in \mathcal{N}(i) \setminus \alpha} m_{\beta \to i}(x_i) \tag{12}$$

Once the maximum number of iterations is reached, the optimal $\mathbf{x}$ is given by:

$$x_i^* = \arg\max_{x_i} \sum_{\alpha} m_{\alpha \to i}(x_i) \tag{13}$$

2 Mixture Models and EM

2.1 What is mixture model?

Again, rather than giving probability background, explain it functionally!.

A mixture model looks like:

$$p(\mathbf{x}) = \prod_{i} \sum_{z_i} p_i(x_i, z_i) \tag{14}$$

Why it cannot be solved by general methods like Belief Propagation? The answer is computational infeasible!

To see that, we notice that a pure $\sum/\prod/\max$ is ready for BP. But for a *mixture* of $\sum$ and $\prod$, it gives exponential terms if simple expansion is applied.

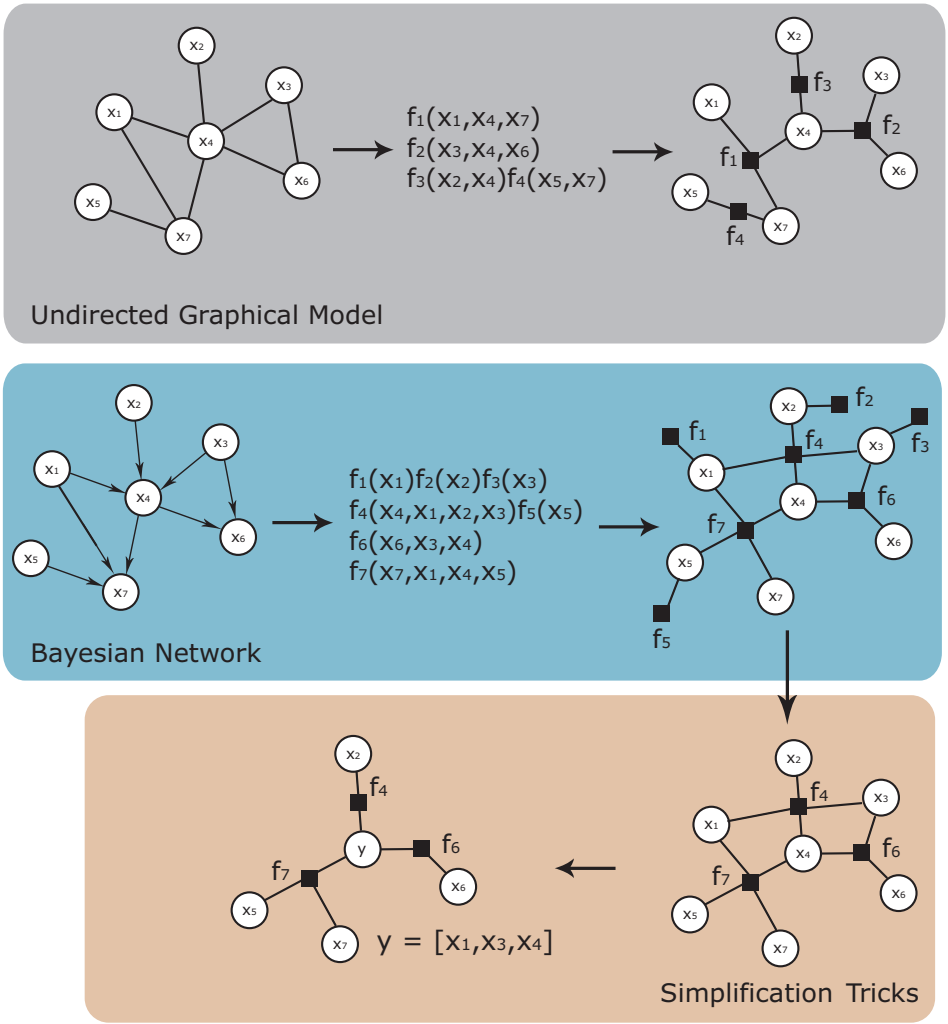

Figure 6: Some examples of conversion from Directed(Undirected) Graph to Factor Graph

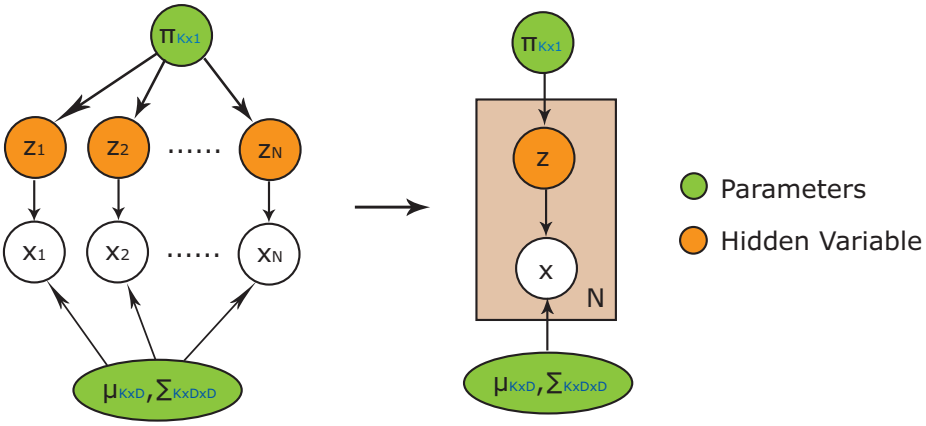

Figure 7: Gaussian Mixture Model

2.2 Examples of mixture models

Gaussian Mixture Model is a typical one. See Fig. 7 for the graphical models.

Its joint probability function is:

$$\begin{aligned} p(\mathbf{x};\mu,\mathbf{\Sigma},\pi) &= \prod_{i=1}^{N} p(x_i;\mu,\mathbf{\Sigma},\pi) \\ &= \prod_{i=1}^{N}\sum_{z_i} p(x_i,z_i;\mu,\mathbf{\Sigma},\pi) \\ &= \prod_{i=1}^{N}\sum_{z_i} p(x_i|z_i;\mu,\mathbf{\Sigma})p(z_i;\pi) \end{aligned} \tag{15}$$

2.3 The Expectation-Maximization Algorithm

Still, we don't take care of its name and its background on probability, but directly attack the most important part: What does EM do?

Here is the principle:

1) The optimization problem $\max_\theta f(\theta)$ is too hard;

2) Try finding a function $g(\theta,\mathbf{w})$ so that $f(\theta)=\max_{\mathbf{w}} g(\theta,\mathbf{w})$ for each $\mathbf{x}$;

3) Then the problem becomes $\max_\theta \max_{\mathbf{w}} g(\theta,\mathbf{w})$.

4) Use coordinate-ascending algorithm to solve it.

The so-called Expectation Maximization algorithm does exactly the same by Jensen Inequality:

$$\begin{aligned}
\log p(\mathbf{x};\theta) &= \sum_i \log \sum_{z_i} p_i(x_i, z_i; \theta) \\
&= \sum_i \log \sum_{z_i} w_{i,z_i} \frac{p_i(x_i, z_i;\theta)}{w_{i,z_i}} \quad \text{s.t.} \sum_{z_i} w_{i,z_i} = 1 \\
&\geq \sum_i \sum_{z_i} w_{i,z_i} \left(\log p_i(x_i, z_i;\theta) - \log w_{i,z_i}\right)
\end{aligned} \tag{16}$$

Here $f(\theta) = \log p(\mathbf{x};\theta)$ and $g(\theta, \mathbf{w}) = \sum_i \sum_{z_i} w_{i,z_i} \left(\log p_i(x_i, z_i;\theta) - \log w_{i,z_i}\right)$.

One can verify that $g(\theta, \mathbf{w})$ does touch $f(\theta)$ when

$$w_{i,z_i} = \frac{p_i(x_i, z_i;\theta)}{\sum_{z_i} p_i(x_i, z_i;\theta)} = p_i(z_i|x_i;\theta). \tag{17}$$

3 Approximate Inference

3.1 Motivation

1) Use a partially factorized function to approximate computational intractable function.

2) Make estimation/inference by 1).

3.2 Mean field theory (MFT)

A typical way to approximate an intractable function with tractable functions. It is originated from Statistics Physics.

We have the following joint probability function $p(\mathbf{x})$:

$$p(\mathbf{x}) = \frac{1}{Z} \exp(-\frac{H(\mathbf{x})}{T}) \tag{18}$$

H is "Energy"(Hamilton), an intractable function defined in high dimension.

Define $F = -T \log Z = -T \log \sum_{\mathbf{x}} \exp(-H(\mathbf{x})/T)$ the "free energy". Apply Jensen Inequality, we easily get:

$$\begin{aligned}
F &= -T \log \sum_{\mathbf{x}} q(\mathbf{x}) \frac{\exp(-H(\mathbf{x})/T)}{q(\mathbf{x})} \quad \text{s.t.} \sum_{\mathbf{x}} q(\mathbf{x}) = 1 \\
&\leq -T \sum_{\mathbf{x}} q(\mathbf{x}) \log \frac{\exp(-H(\mathbf{x})/T)}{q(\mathbf{x})} \\
&= \sum_{\mathbf{x}} q(\mathbf{x}) H(\mathbf{x}) + T q(\mathbf{x}) \log q(\mathbf{x}) \\
&= \mathbb{E}_q[H] - T S_q \equiv F_q
\end{aligned} \tag{19}$$

MFT tries to:

1) approximate H with factorized $q(\mathbf{x}) = \prod_i q_i(x_i)$ so that its associated $H_0(\mathbf{x}) = -T \log Z_q q(\mathbf{x}) = -T \sum_i \log Z_i q_i(x_i)$ is separable. ($Z_i = \sum_{x_i} q_i(x_i)$)

2) Yet, MFT wants to properly select q so that the upper bound reaches its minimum.

Note that if q is factorized,

$$F_q = \sum_{x_i} q_i(x_i) \sum_{j \neq i} q_j(x_j) H(\mathbf{x}) + T \sum_i \sum_{x_i} q_i(x_i) \log q_i(x_i) \tag{20}$$

Using lagrange multiplier, we get the following equation:

$$\frac{\delta F_q}{\delta q_i(x_i)} = \sum_{j \neq i} q_j(x_j) H(\mathbf{x}) + T(\log q_i(x_i) + 1) + \lambda = 0 \tag{21}$$

Let $\sum_{j \neq i} q_j(x_j) H(\mathbf{x}) = \mathbb{E}_{q \setminus q_i}[H]$, we thus have the following joint equations:

$$q(x_i) = \frac{\exp(-\mathbb{E}_{q \setminus q_i}[H]/T)}{\sum_{x_i} \exp(-\mathbb{E}_{q \setminus q_i}[H]/T)} \tag{22}$$

Or more concisely:

$$\log q(x_i) = \mathbb{E}_{q \setminus q_i}[\log p(\mathbf{x})] + \text{const.} \tag{23}$$

These are called "mean field equations".

3.3 Basic ideas on MFT

3.3.1 "variational methods"

One has to optimize the whole function rather than a series of variables → *variational methods.*

3.3.2 "approximate inference"

Unlike EM, an arbitrary setting of q doesn't necessary touch the bound. → *approximate inference.*

3.3.3 What is minimized?

Minimize $F_q \rightarrow$ Minimize $\frac{1}{T}(F_q - F)$. Then what is $\frac{1}{T}(F_q - F)$?

$$\begin{aligned}
\frac{1}{T}(F_q - F) &= \sum_{\mathbf{x}} q(\mathbf{x}) H(\mathbf{x})/T + q(\mathbf{x}) \log q(\mathbf{x}) + \log Z \qquad (24) \\
&= -\sum_{\mathbf{x}} q(\mathbf{x}) \log p(\mathbf{x}) + q(\mathbf{x}) \log q(\mathbf{x}) \quad (\because -\log p = \log Z + H/T) \\
&= -\sum_{\mathbf{x}} q(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \\
&= \mathrm{KL}(q \| p)
\end{aligned}$$

Actually, we are trying to find a factorized function as "close" to original function as possible!

3.4 Approximate Inference

Two great characteristics:

1) Making no difference between *hidden variables* and *parameters*.

2) After equations are solved, posteriors of parameters are automatically found.

Suppose we have $p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z}, \mathbf{x})$. $\mathbf{z}$ are all the parameters/hidden variables. Then by our old trick of Jensen Inequality:

$$\begin{aligned} \log p(\mathbf{x}) &= \log \sum_{\mathbf{z}} q(\mathbf{z}) \frac{p(\mathbf{z}, \mathbf{x})}{q(\mathbf{z})} \quad \text{s.t.} \sum_{\mathbf{z}} q(\mathbf{z}) = 1 \\ &\geq \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{z}, \mathbf{x})}{q(\mathbf{z})} \equiv \mathcal{L}(q) \end{aligned} \tag{25}$$

As usual, one maximizes the lower bound $\mathcal{L}(q)$ with respect to $q(\mathbf{z})$ to get the best approximation. Yet, let us have a look at what the other part is:

$$\begin{aligned} \log p(\mathbf{x}) - \mathcal{L}(q) &= \sum_{\mathbf{z}} q(\mathbf{z}) \left[\log p(\mathbf{x}) - \log p(\mathbf{z}, \mathbf{x}) + \log q(\mathbf{z})\right] \\ &= -\sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} \equiv \mathrm{KL}\Big(q(\mathbf{z}) \Big\| p(\mathbf{z}|\mathbf{x})\Big) \end{aligned} \tag{26}$$

So actually we are finding "closest" function to posteriors $p(\mathbf{z}|\mathbf{x})$, this is just what we want in probabilistic inference. → To infer something, first add them up.

Suppose $q(\mathbf{z}) = \prod_i q_i(\mathbf{z}_i)$ with $\mathbf{z}_i$ a disjoint partition of $\mathbf{z}$, then we can find its optimal: (in PRML, pp. 465)

$$\log q_i(\mathbf{z}_i) = \mathbb{E}_{q \setminus q_i} \left[\log p(\mathbf{x}, \mathbf{z})\right] + \text{const.} \tag{27}$$

One can make a comparison between 27 and 23. They are much alike.

3.5 Local variational models

Function tricks: $f(x)$ complicated → $f(x) = \max_\xi g_\xi(x)$, g_ξ is good, for example, exponential family.

Then $\mathcal{L}(f) \geq \mathcal{L}(g_\xi)$ have tractable forms. Solve it, then take maximum w.r.t. ξ to minimize the error (not necessary touch $\mathcal{L}(f)$ since in general $\xi^*(x) \equiv \arg\max_\xi g_\xi(x) \neq \arg\max_\xi \mathcal{L}(g_\xi)$)

3.6 Expectation Propagation (EP)

3.6.1 Overview

Motivation:$\mathrm{KL}(q\|p) \to \mathrm{KL}(p\|q)$.

Expectation propagation: We have a probability function p to evaluate.

$$p(\mathbf{x}) = \prod_{i=1}^{m} f_i(\mathbf{x}) \tag{28}$$

And we have an approximation $q(\mathbf{x};\theta) = \prod_{i=1}^{m} \tilde{f}_i(\mathbf{x};\theta)$, with each $\tilde{f}_i$ has special forms, i.e., belonging to exponential family. θ are the parameters.

Our goal is to optimize θ so that $\mathrm{KL}(p\|q)$ is minimized.

3.6.2 The algorithm

Expectation Propagation (EP) adopts an iterative schemes to optimize $\tilde{f}_i$ (coordinate-descending):

Initialize $\tilde{f}_i$, i.e., initialize θ;
repeat
  while $i = 1$ to m **do**
    Let $q^{\setminus i}(\mathbf{x};\theta^{(k)}) = q(\mathbf{x};\theta^{(k)})/\tilde{f}_i(\mathbf{x};\theta^{(k)})$
    $\theta^{(k+1)} \leftarrow \arg\min_\theta \mathrm{KL}\left(\frac{f_i(\mathbf{x})q^{\setminus i}(\mathbf{x};\theta)}{Z_i}\|q(\mathbf{x};\theta^{(k)})\right)$
  end while
until Convergence

How to minimize KL-divergence deserves consideration!

In general, $\mathrm{KL}(p\|q)$ is not factorable if p is too complicated(no exact inference). $\to$ it works only in the following conditions.

1) p has a special analytic form, i.e., a product of mixture model. $\to$ we can have each q component corresponding to one mixture model.

2) $\min \mathrm{KL}(p\|q) \to \mathbb{E}_p[\mathbf{u}(\mathbf{x})] = \mathbb{E}_q[\mathbf{u}(\mathbf{x})]$ if $q(\mathbf{x}) = h(\mathbf{x})g(\eta)\exp\{\eta^T\mathbf{u}(\mathbf{x})\}$ belongs to exponential family.

3) $\min \mathrm{KL}(p\|q) \to q_i^*(\mathbf{x}_i) = p(\mathbf{x}_i) \equiv \int p(\mathbf{x}) \prod_{x_j \notin \mathbf{x}_i} \mathrm{d}x_j$ if q can be factorized to be $q(\mathbf{x}) = \prod_i q_i(\mathbf{x}_i)$

2) is feasible if p can be partially factorized. $\to$ Belief Propagation is its special case.

3.6.3 BP as a special case of EP

Assuming the joint probability $p(\mathbf{x})$ can be expressed as the following:

$$p(\mathbf{x}) = \prod_i f_i(\mathbf{x}_i), \tag{29}$$

with each f_i is function of only a few variables $\mathbf{x}_i$. One can readily use graphical model to express the function p. Of course, although p can be partially factorized, we still cannot find global optimal solution for some quantities, likemax$_{\mathbf{x}}\, p(\mathbf{x})$.

We set q to be an approximation of p as the following:

$$q(\mathbf{x}) \propto \prod_i \prod_k \tilde{f}_{ik}(x_k) \tag{30}$$

For example: if $p(x_1, x_2, x_3) = f_a(x_1, x_2) f_b(x_1, x_3)$, we set $\tilde{f}_a(x_1, x_2) = \tilde{f}_{a1}(x_1)\tilde{f}_{a2}(x_2)$ and $f_b(x_1, x_3) = \tilde{f}_{b1}(x_1)\tilde{f}_{b2}(x_3)$.

We can see that q is fully factorized.

If we use Algorithm 7 to find best q, we can see the iteration process is the same as in BP. (See PRML pp. 515-516)

Appendix A: Some issues on training models with partition function

Generally speaking, any undirected graphical models have a normalization "constant" called partition function. It is a constant in inference stage, yet it does change with parameters.

Fortunately, probability taking the exponential form enjoys partition function of good characteristics. Training of such models thus becomes practical.

Most such models are originated from Statistics Physics.

$$p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \exp(-V(\mathbf{x}; \theta)) \tag{31}$$

The following is an extreme important equation:

$$\begin{aligned} \frac{\partial \log Z(\theta)}{\partial \theta} &= \frac{1}{Z(\theta)} \frac{\partial Z(\theta)}{\partial \theta} \\ &= -\frac{1}{Z(\theta)} \sum_{\mathbf{x}} \exp(-V(\mathbf{x}; \theta)) \frac{\partial V(\mathbf{x}; \theta)}{\partial \theta} \\ &= -\mathbb{E}_{p(\mathbf{x};\theta)} \left[\frac{\partial V(\mathbf{x}; \theta)}{\partial \theta} \right] \end{aligned} \tag{32}$$

With Eqn.33, we can find that

$$\frac{\partial \log p(\mathbf{x}; \theta)}{\partial \theta} = \mathbb{E}_{p(\mathbf{x};\theta)} \left[\frac{\partial V(\mathbf{x}; \theta)}{\partial \theta} \right] - \frac{\partial V(\mathbf{x}; \theta)}{\partial \theta} \tag{33}$$

This is the basics for training such models. Particularly, if $\mathbf{x}_i$ are drawn i.i.d. with distribution Eqn. 31, the training rule will become

$$\mathbb{E}_{p(\mathbf{x};\theta)}\left[\frac{\partial V(\mathbf{x};\theta)}{\partial \theta}\right] = \mathbb{E}_{\text{sample}}\left[\frac{\partial V(\mathbf{x};\theta)}{\partial \theta}\right] \tag{34}$$

In addition, if $V(\mathbf{x};\theta)$ is convex with respect to θ, then $\log p(\mathbf{x};\theta)$ is concave, because $\log \sum_{\mathbf{x}} \exp f(\mathbf{x};\theta)$ is convex(concave) if $f(\mathbf{x};\theta)$ is convex(concave) w.r.t θ for any $\mathbf{x}$.

4 Sampling methods

4.1 Motivation

What is sampling? → Draw samples from a given probability distribution.

Why sampling? → 1)Simulation 2)Evaluation expectations involving a prohibitive probability distribution. i.e., not factorizable.

$\mathbb{E}[f] = \int f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x} \rightarrow \hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{x}^{(l)})$

4.2 Independent Sampling

4.2.1 How to sample?–Basic technique

Transformation between probability distributions:

$$p(\mathbf{y}) = \left|\frac{\partial \mathbf{x}}{\partial \mathbf{y}}\right| p(\mathbf{x}) \tag{35}$$

For example: Exponential/Gaussian/Other distributions that $F(x)$ can be easily evaluated and inverted.

In one dimension: $Y = F^{-1}(X)$ if X is drawn from U$[0,1]$ and $F(x) = \int_{-\infty}^{x} f(y)\mathrm{d}y$.

A typical example: Gaussian distribution (Box-Muller)

$$\begin{aligned} r &= \sqrt{z_1^2 + z_2^2} \\ y_1 &= z_1 \frac{\sqrt{-2\log r}}{r} \\ y_2 &= z_2 \frac{\sqrt{-2\log r}}{r} \end{aligned} \tag{36}$$

where z_1 and z_2 are uniformly distributed in the unit circle $r \leq 1$.

4.2.2 How to sample?–Advanced technique

Rejection sampling: We have $p(\mathbf{x})$, easy to evaluate the function but hard to draw sample from this distribution.

Using q to approximate p

If we can find a simple distribution q so that $kq(\mathbf{x}) \geq p(\mathbf{x})$ always, then:

1) Sampling $\mathbf{x}$ from q;

2) Draw $u \sim \mathrm{U}[0, kq(\mathbf{x})]$;

3) Accept sample $\mathbf{x}$ when $u < p(\mathbf{x})$.

We want low rejection rate→ Its efficiency depends on how big k is.

4.2.3 Important Sampling

Characteristics: 1)Evaluate expectations 2)doesn't draw samples from the corespondent distribution.

Using q to approximate p

$$\begin{aligned} \mathbb{E}[f] &= \int f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= \frac{1}{L}\sum_{l=1}^{L}\frac{p(\mathbf{x}^{(l)})}{q(\mathbf{x}^{(l)})}f(\mathbf{x}^{(l)}) \end{aligned} \tag{37}$$

Better than the two stage schemes: 1)sampling p using complicated methods. 2)Approximate expectation using $\frac{1}{L}\sum_{l=1}^{L} f(\mathbf{x}^{(l)})$.

4.3 Dependent Sampling

4.3.1 Motivation

How to sample? More advanced methods→ MCMC.

Big Idea: what we want is not independency, but properly scattered samples over the probability space.

4.3.2 Basic Knowledge on Markov Chain

Basic idea: a mapping between probability distributions. $T(\mathbf{z}'|\mathbf{z}) : p(\mathbf{z}) \mapsto p(\mathbf{z}')$ by definition of $p(\mathbf{z}') = \sum_{\mathbf{z}} T(\mathbf{z}'|\mathbf{z})p(\mathbf{z})$.

Important property: ergodic/irreducible *"The probability of traversing from every state to every other state is strictly positive."*[1].

Stationary distribution $\pi(\mathbf{z})$ (Theorem 11.10[1]): *Every ergodic Markov Chain has a unique stationary distribution* $\pi(\mathbf{z})$*, so that* $\sum_{\mathbf{z}} T(\mathbf{z}'|\mathbf{z})\pi(\mathbf{z}) = \pi(\mathbf{z}')$.

Detailed balanced (reversible): $T(\mathbf{z}'|\mathbf{z})\pi(\mathbf{z}) = \pi(\mathbf{z}')T(\mathbf{z}'|\mathbf{z})$.

Main theorem: ergodic + reversible = convergence to stationary distribution regardless of initial condition.

4.3.3 Markov Chain Monte Carlo

We have a transition function(matrix): $q(\mathbf{z}|\mathbf{z}^{(\tau)})$. Using this function we sample as following (Symmetric case, *Metropolis Algorithm* $q(\mathbf{z}_A|\mathbf{z}_B) = q(\mathbf{z}_B|\mathbf{z}_A)$):

1) Given sample $\mathbf{z}^{(\tau)}$, we sample next candidate one by $q(\mathbf{z}|\mathbf{z}^{(\tau)})$;

2) The sample is accepted with probability

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})}\right)$$

3) If accepted, $\mathbf{z}^{(\tau+1)} = \mathbf{z}^*$, otherwise $\mathbf{z}^{(\tau+1)} = \mathbf{z}^{(\tau)}$.

(More general case *Metropolis-Hastings Algorithm*)

$$A_k(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)q_k(\mathbf{z}^{(\tau)}|\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})q_k(\mathbf{z}^*|\mathbf{z}^{(\tau)})}\right)$$

Each time we use different q_k to get sampling.

4.3.4 Gibbs Sampling

Gibbs sampling is a special kind of Metropolis-Hastings Algorithm by setting $q_k(\mathbf{z}^*|\mathbf{z}) = p(z_k^*|\mathbf{z}_{\setminus k})$ with $\mathbf{z}^*_{\setminus k} = \mathbf{z}_{\setminus k}$.

4.4 Hybrid Monte Carlo Algorithm

4.4.1 Motivation

Random walk phenomena: On average, the distance of traveling by MCMC is the square root of iteration steps.

1) Step too small in each iteration → low efficiency to traverse the probability space.

2) Step too big in each iteration → high rejection rate.

Goal: move in large scale yet keep low rejection rate.

4.4.2 Physics here!

Construct a dynamical system so that the "particle" moves in a systematic way. → *The Hamilton dynamics.*

First we define the momentum $r_i = \frac{\mathrm{d}z_i}{\mathrm{d}\tau}$. Assuming the probability can be expressed by $p(\mathbf{z}) = \frac{1}{Z_p}\exp(-E(\mathbf{z}))$, we define the Hamilton function ("total energy") in the phase space $(\mathbf{z}, \mathbf{r})$:

$$H(\mathbf{z}, \mathbf{r}) = E(\mathbf{z}) + K(\mathbf{r}) \quad \text{with } K(\mathbf{r}) = \frac{1}{2}\sum_i r_i^2 \tag{38}$$

And Hamilton dynamics are defined as the following equations:

$$\begin{aligned} \frac{\mathrm{d}z_i}{\mathrm{d}\tau} &= \frac{\partial H}{\partial r_i} \\ \frac{\mathrm{d}r_i}{\mathrm{d}\tau} &= -\frac{\partial H}{\partial z_i} \end{aligned} \tag{39}$$

Naturally, the associated joint probability $p(\mathbf{x}, \mathbf{r})$ is defined as $\frac{1}{Z_H}\exp(-H(\mathbf{z}, \mathbf{r}))$. This dynamics has two important properties:

1) H is a constant of motion, i.e., $\frac{\mathrm{d}H}{\mathrm{d}\tau} = 0$ along the trajectory.

2) "Volume" in the phase space is preserved, i.e., $\mathrm{div}V \equiv \mathrm{div}\left(\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}\tau}, \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\tau}\right) = 0$.

To numerically integrate 39, we use Leapfrog method (in PRML pp. 551-552). It is not the most accurate numerical integration method, yet it has the following properties:

1) (Bad)The invariance of H along particle trajectory is not guaranteed.

2) (Good)It is time-reversible.

3) (Good)It preserves volume because it "either updates r_i or z_i by an amount that is a function only of the other variable"(See PRML pp. 552).

Key contribution→It offers a systematic way to move from one sample to another place faraway.

4.4.3 Hybrid Monte Carlo

The Hybrid Monte Carlo sampling includes the following steps:

1) Move from one sample $(\mathbf{z}, \mathbf{r})$ to $(\mathbf{z}^*, \mathbf{r}^*)$ using *Hamilton dynamics.*

2) Accept $(\mathbf{z}^*, \mathbf{r}^*)$ with probability $\min\left(1, \exp\{H(\mathbf{z}^*, \mathbf{r}^*) - H(\mathbf{z}, \mathbf{r})\}\right)$. $H(\mathbf{z}, \mathbf{r})$ should equal to $H(\mathbf{z}^*, \mathbf{r}^*)$ if the numerical integration is perfect.

3) Use Gibbs sampling to re-sample the momentum term $\mathbf{r}^*$, keeping $\mathbf{z}^*$ fixed.

It can be proved that the property of "detailed balance" holds for Step 1)+2), i.e., $p(\mathbf{x}, \mathbf{r})$ is a stationary probability for Markovian updating 1)+2) (See PRML pp. 552-553); and Gibbs sampling in Step 3) also keeps $p(\mathbf{x}, \mathbf{r})$ unchanged. As a result, it is guaranteed that the samples drawn from Hybrid Monte Carlo is distributed with $p(\mathbf{x}, \mathbf{r})$. Discarding $\mathbf{r}$, we thus get samples with distribution $p(\mathbf{x})$, the marginal distribution of $p(\mathbf{x}, \mathbf{r})$.

4.5 Application: Particle Filter

4.5.1 Motivation

Fig.2 is a typical graphical model. In many applications (like tracking or any kind of sequential data modeling), one wants to estimate the posterior $p(x_k|y_{1:k}) \equiv p(x_k|y_1, \ldots, y_k)$.

Using Bayes Theorem, we can get the following general rule:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})\mathrm{d}x_{k-1} \quad \text{Prediction} \tag{40}$$

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k})} \quad \text{Update} \tag{41}$$

For x_k discrete, the integral is replaced by summation. So the solution is trivial. Then what happens if x_k is continuous?

The problem: one cannot analytically find the posterior because of intractable integrals → Sampling methods.

4.5.2 Kalman Filter

One famous example of the graphical model Fig. 2 is the Kalman Filter:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{42}$$

$$y_k = Hx_k + v_k, \tag{43}$$

with the assumption that w_k and v_k are both Gaussian-distributed.

Given $y_1, \ldots, y_k$, Kalman filter aims to find $\hat{x}_k$, the (posterior) estimation of hidden variable x_k, so that $\mathbb{E}[\|x_k - \hat{x}_k\|^2]$ is minimized.

Note1: This can be done analytically because random variables are Gaussian.

Note2: Kalman Filter doesn't estimate the distribution, but estimate the "best" value.

4.5.3 Particle Filter

How to make intractable integral tractable? Sampling methods.

Basic idea of Particle Filter: represent the posterior distribution $p(x_k|y_{1:k})$ with "weighted particles" $(w_k^{(l)}, x_k^{(l)})$ so that for any $f(x_k)$, the intractable integral becomes simple summation:

$$\mathbb{E}[f] \equiv \int f(x_k)p(x_k|y_{1:k})\mathrm{d}x_k \approx \sum_{l=1}^{L} w_k^{(l)} f(x_k^{(l)}) \tag{44}$$

The following problem is: given "weighted particles" $(w_{k-1}^{(l)}, x_{k-1}^{(l)})$ at stage $k-1$, how to get $(w_k^{(l)}, x_k^{(l)})$?

Answer: sample $p(x_k|x_{k-1}^{(l)})$ and properly adjust the weight to give $w_k^{(l)}$ for each sample l:

$$x_k^{(l)} \sim p(x_k|x_{k-1}^{(l)}) \tag{45}$$

$$w_k^{(l)} \propto w_{k-1}^{(l)} p(y_k|x_k^{(l)}) \quad \text{s.t.} \sum_{l=1}^{L} w_k^{(l)} = 1 \tag{46}$$

if sampling $p(x_k|x_{k-1}^{(l)})$ is hard, we can adopt important sampling to yield the following updating rule:

$$x_k^{(l)} \sim q_k(x_k; x_{k-1}^{(l)}) \tag{47}$$

$$w_k^{(l)} \propto w_{k-1}^{(l)} \frac{p(y_k|x_k^{(l)})p(x_k|x_{k-1}^{(l)})}{q_k(x_k; x_{k-1}^{(l)})} \quad \text{s.t.} \sum_{l=1}^{L} w_k^{(l)} = 1 \tag{48}$$

The above update rules are derived from Eqn.44.

Note: the resulting $(w_k^{(l)}, x_k^{(l)})$ could "concentrate" on few particles. (Few particles have large weights, while the rest have negligible weights.) So re-sampling is required.

See [2] for more detailed information. In PRML pp. 645, there is also a brief introduction.

5 Basics about Gaussian Processes

5.1 Definition

GP is a random process $\{X(t), t \in \mathrm{R}\}$ so that for any finite sampling at time $t_1, \ldots, t_N$:

$$(X(t_1), X(t_2), \ldots, X(t_N)) \sim \mathcal{N}(\mu(t_1, \ldots, t_N); \Sigma(t_1, \ldots, t_N)) \tag{49}$$

For practical reason, we often choose $\mu(t_1, \ldots, t_N) = 0$ and $\{\Sigma(t_1, \ldots, t_N)\}_{ij} = K(t_i, t_j)$. K is a correlation function to model given their locations how related two data points are.

$K(\cdot, \cdot)$ has to be positive-definite. It may contain some hyper-parameters.

5.2 Train & Prediction given hyper-parameters

Use Bayesian approach → find posterior $P(t|x,\mathcal{D})$ ($\mathcal{D} = \{\mathbf{x}', \mathbf{t}'\} = \{x'_{(i)}, t'_{(i)}\}_{i=1}^{N}$ are training samples).

$$P(t|x,\mathcal{D}) \propto P(t,\mathbf{t}'|x,\mathbf{x}') \sim \mathcal{N}(x,\mathbf{x}') \tag{50}$$

which equals to finding a marginal distribution of a $N+1$-dimensional Gaussian Distribution, which is a 1-D Gaussian. Here is the result(Using Schur's complement(NOT Lemma)):

$$\begin{aligned} \mu(x) &= K^T(x)K^{-1}\mathbf{t}' && (51)\\ \sigma^2(x) &= K(x,x) - K^T(x)K^{-1}K(x) \quad \text{Schur's complement} && (52) \end{aligned}$$

where $K_{ij} = K(x'_i, x'_j)$ only depends on $\mathbf{x}'$ and $K(x) = [K(x'_1, x), \ldots, K(x'_N, x)]$.

For noisy case $P(\mathbf{t}'|\mathbf{x}') \sim \mathcal{N}(0, K + \sigma^2 I)$, the expression is similar:

$$\begin{aligned} \mu(x) &= K^T(x)(K+\sigma^2 I)^{-1}\mathbf{t}' && (53)\\ \sigma^2(x) &= K(x,x) - K^T(x)(K+\sigma^2 I)^{-1}K(x) \quad \text{noise eliminated} && (54) \end{aligned}$$

5.3 Hyper-parameters

1) Maximum Likelihood to find local extremes in parameter space. Takes $O(N^3)$ for each iteration since K has to be reestimated.

2) Bayesian approach: $\mu(x) = \int_\theta \mu_\theta(x|\mathcal{D})P(\theta|\mathcal{D})\mathrm{d}\theta$, where $P(\theta|\mathcal{D}) \propto P(\mathcal{D}|\theta)P(\theta)$ are parameter posteriors.

5.4 Some idea about Gaussian Processes

How does it do regression?

$$\mu(x) = K^T(x)K^{-1}\mathbf{t}' = K^T(x)w = \sum_i w_i K(x'_i, x) \tag{55}$$

where $w = K^{-1}\mathbf{t}'$ is coefficient. GP is like kernel regression.

In the noise-free case, $\mu(x)$ should pass through precisely all the training samples → w can be determined without probabilistic framework.

The importance of GP is that it provides a probabilistic framework.

- $\sigma^2(x)$ is determined.
- One can have more general probabilistic framework incorporating GP as a part. E.g., Mixture of Expert + GP = locally GP.

6 Basics on Dirichlet Process

6.1 Definition

A Dirichlet Process is a "random distribution function" G. It has a scaling parameter α and base distribution G_0 with support A so that for any partition $A_1, A_2, \ldots, A_k$ of A:

$$(G(A_1), G(A_2), \ldots, G(A_k)) \sim \text{Dirichlet}(\alpha G_0(A_1), \alpha G_0(A_2), \ldots, \alpha G_0(A_k)) \tag{56}$$

We denote $G \sim DP(\alpha, G_0)$.

6.2 Sampling from DP

The definition is quite indirect (different from GP), how to sample it?

A theorem tells us distribution function G is like the following:

$$G(x) = \sum_i \pi_i \delta(x - x_i) \tag{57}$$

in which $x_i \sim G_0$ independently, and $\pi_i = v_i \prod_{j<i}(1-v_j)$, where $v_i \sim \text{Beta}(1, \alpha)$.

6.3 Why DP?

If $G \sim DP(\alpha, G)$, we expect to get exact the same thing from $G > 1$ times with $p > 0$, since G is a countable infinite summation of deltas. $\rightarrow$ it is possible to sample parameters of cluster.

Moreover, instead of 1) sample G from $DP(\alpha, G_0)$, 2)sample $\{y_i\}$ from G to get cluster parameters, one can directly sample y_i based on previous samples $y_1, \ldots, y_{i-1}$ by the following:

$$P(y_i|y_1, \ldots, y_{i-1}) = \begin{cases} y_i = y_j & \text{with prob } \frac{1}{i+\alpha-1} \\ y_i \sim G_0 & \text{with prob } \frac{\alpha}{i+\alpha-1} \end{cases} \tag{58}$$

This is because the joint distribution $P(y_1, \ldots, y_n) = \int_G \prod_i P(y_i|G) P(G|\alpha, G_0) \mathrm{d}G$ can be factorized to be Eqn. 58. See Fig. 8.

6.4 Infinite Mixture models

G is countable infinite summation of deltas $\rightarrow$ Infinite mixture models:

$$P(c_1, \ldots, c_N, \theta|x_1, \ldots, x_N) \propto P(c_1, \ldots, c_N) \prod_{k=1}^{\infty} P(\theta_k) \prod_{i=1}^{N} P(x_i|c_i, \theta_{c_i}) \tag{59}$$

where $P(c_1, \ldots, c_N)$ is the joint probability for Chinese Restaurant Process. It is intractable to get local maxima of both c and θ, yet one can do it by steepest descent or Gibbs Sampling:

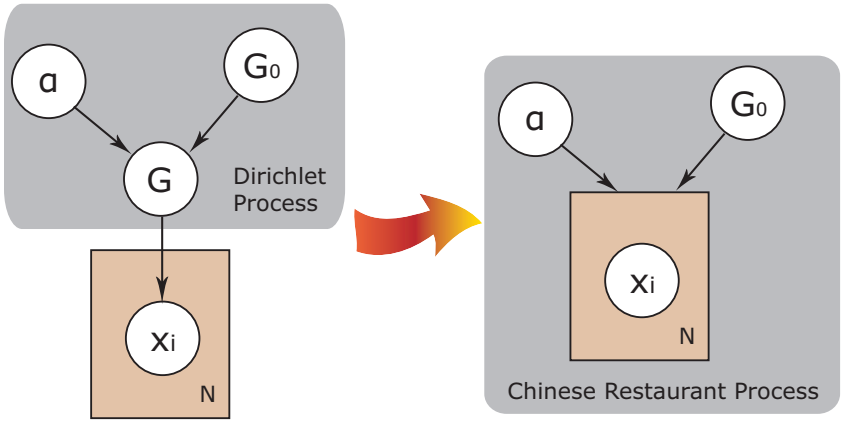

Figure 8: Dirichlet Process + independent sampling on G = Chinese Restaurant Process

$$P(c_i|c_{-i}) = \begin{cases} \frac{\#[c_{-i}=k]}{N+\alpha-1} & c_i = k \\ \frac{\alpha}{N+\alpha-1} & c_i = \text{newLabel} \end{cases} \tag{60}$$

7 Basics on Diffusion Process

7.1 Definition

A diffusion process is a random process $\{X(t)\}$ so that the following Stochastic Differential Equation (SDE) is held:

$$\mathrm{d}X_t = f(t, X_t)\mathrm{d}t + \Sigma^{1/2}\mathrm{d}W_t \quad \mathrm{d}W_t \sim \mathcal{N}(0, \mathrm{d}tI) \tag{61}$$

where $f(t, X_t)$ is the drift term and W_t is Wiener Process (random walk term).

7.2 Intuitive Ideas

It is quite sophisticated to give a formal mathematical definition, yet one can have an intuitive idea about what a diffusion process is. Dividing the time interval $[t_0, t_1]$ into K parts and using Euler-Muryama discrete approximation, we have

$$\Delta X_k = f(t_k, X_k)\Delta t + \Sigma^{1/2}\Delta W_k \quad \Delta W_k \sim \mathcal{N}(0, \Delta tI) \tag{62}$$

where $\Delta X_k = X_{k+1} - X_k$. Then we will have K random variables and their joint distribution will be the following:

$$P(X_0, \ldots, X_K) = P(X_0) \prod_{k=0}^{K-1} P(X_{k+1}|X_k) \tag{63}$$

where $P(X_{k+1}|X_k) \sim \mathcal{N}(X_k + f(t_k, X_k)\Delta t, \Sigma\Delta t)$.

$f(t_k, X_k)$ could have hyper-parameters so that they can be adjusted for better fitting for data.

Diffusion Process → Continuous Markov Chain.

7.3 Why Diffusion Process?

Why not just using Discrete Markov Chain? The key problem is that we don't know the location of evidences in advance, which can be placed in any $t \in [t_0, t_1]$. Discrete Markov Chain is not suitable for this situation, but Continuous Markov Chain is ok.

7.4 Posterior given data points

For marginal distribution $P(X_t)$ to be Gaussian, we set $f(t, x) = -A_t x + b_t$ (Linear SDE). Suppose $P(X_t) \sim \mathcal{N}(m_t, S_t)$ is Gaussian, we can get the corresponding ODE of m_t and S_t as follows:

$$\dot{m}_t = -A_t m_t + b_t \tag{64}$$
$$\dot{S}_t = -S_t A_t^T - A_t S_t + \Sigma \tag{65}$$

This two equations can be derived using definition of Diffusion Process (discretization → take limit) or can be formally derived by the following:

$$\mathrm{d}m_t = \mathbb{E}[X_t + \mathrm{d}X_t] - \mathbb{E}[X_t] \tag{66}$$
$$\mathrm{d}S_t = \mathbb{E}[(\bar{X}_t + \mathrm{d}\bar{X}_t)(\bar{X}_t + \mathrm{d}\bar{X}_t)^T] - \mathbb{E}[\bar{X}_t \bar{X}_t^T] \tag{67}$$

where $\bar{X}_t = X_t - m_t$.

By Eqn. 66 and Eqn. 67, one can find posterior marginal distribution $P(X_t|X_0 = x_0)$ by setting the initial condition $m_0 = x_0$ and $S_0 = 0$. And the posterior distribution of X_t is:

$$P(X_t|X_{t_1} = x_1, X_{t_2} = x_2) \qquad t_1 \le t \le t_2, \text{no extra info in } [t_1, t_2] \tag{68}$$
$$\propto P(X_t|X_{t_1} = x_1) P(X_{t_2} = x_2|X_t) \tag{69}$$

where $P(X_t|X_{t_1} = x_1)$ and $P(X_{t_2} = x_2|X_t) = P(X_t|X_{t_2} = x_2)$ (reversible) can be obtained by Eqn. 66 and Eqn. 67.

7.5 Derivation of Eqn. 66 and Eqn. 67

How to find distribution of X_t given X_0? We divided $[0, t]$ into K parts, $0 = t_0 < t_1 < \ldots < t_K = t$, each with time span Δt.

By discrete approximation ($x_i = x_{t_i}$):

$$X_1|X_0 \sim \mathcal{N}(x_0 + f(t_0, x_0)\Delta t, \Delta t\Sigma) \tag{70}$$
$$X_2|X_1 \sim \mathcal{N}(x_1 + f(t_1, x_1)\Delta t, \Delta t\Sigma) \tag{71}$$

Since $P(x_2|x_0) = \int_{x_1} P(x_2|x_1)P(x_1|x_0)\mathrm{d}x_1$ and we use linear condition that $f(t,x) = f(t)x$, which gives Gaussian distribution[1]:

$$X_2|X_0 \sim \mathcal{N}((1+f(t_0)\Delta t)(1+f(t_1)\Delta t)x_0, *) \tag{72}$$

By induction, we get:

$$X_t|X_0 \sim \mathcal{N}\left(\prod_{k=0}^{K-1}\left(1+f\left(\frac{tk}{K}\right)\frac{t}{K}\right)x_0, *\right) \tag{73}$$

Since:

$$m_t = \lim_{K\to\infty}\prod_{k=0}^{K-1}\left(1+f\left(\frac{tk}{K}\right)\frac{t}{K}\right)x_0 = x_0 e^{\int_0^t f(\tau)\mathrm{d}\tau} \tag{74}$$

We naturally have $\dot{m}_t = f_t m_t$ with initial condition $m_0 = x_0$, as indicated in Eqn.66.

7.6 An introduction on Diffusion Map

"Diffusion Map" is another concept used to interpret the mechanism behind spectral clustering.

Suppose we have N points $\{x_1, \ldots, x_N\}$ and $N \times N$ affinity matrix is defined as $a_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. Treating N points as vertices, we get a full-connected graph and associated transition matrix M:

$$M = L^{-1}A \quad L = \mathrm{diag}\left(\sum_j a_{1j}, \ldots, \sum_j a_{Nj}\right) \tag{75}$$

Given that a particle is settled in vertex i at time 0, we can thus find the probability distribution $p(j|i,m)$ over the graph at discrete time m:

$$p(j|i,m) = e_i^T M^m \tag{76}$$

where e_i contains 1 in i-th row and zero elsewhere.

Then we compare a weighted distance between $p(j|i_1,m)$ and $p(j|i_2,m)$ for similarity measure of two point x_i and x_j:

$$\mathrm{Dist}^2(i_1,i_2) = \sum_j (p(j|i_1,m) - p(j|i_2,m))^2 w(j) \tag{77}$$

This distance can be converted to Euclidean distance if we apply "Diffusion Map" for each point, and spectral clustering does exactly the same mapping. To see why, note that Eqn. 77 can be reformulated in matrix form:

$$\mathrm{Dist}^2(i_1,i_2) = (e_{i_1} - e_{i_2})^T M^m W (M^m)^T (e_{i_1} - e_{i_2}) \tag{78}$$

[1]Using the formula:

$$\frac{(a-b)^2}{n} + \frac{(kb-c)^2}{m} = \frac{m+nk^2}{mn}\left(b - \frac{ma+nkc}{m+nk^2}\right)^2 + \frac{(ka-c)^2}{m+nk^2}$$

where W is a diagonal matrix.

For any matrix M, it can be decomposed into UDV^T, where U is the column stack of its right normalized eigenvectors, V its left normalized eigenvectors and D its eigenvalues in the diagonal, with biorthogonal property that $U^TV = V^TU = I$.

Note: It is not SVD! so U and V doesn't have the property of $U^TU = V^TV = I$. Yet it is related. Suppose $M_s = L^{1/2}ML^{-1/2} = L^{-1/2}AL^{-1/2} = U'\Sigma U'^T$ (M_s symmetric), then $U = L^{-1/2}U'$, $V = L^{1/2}U'$ and $D = \Sigma$.

Then $M^m = UD^mV^T$ and $M^mW(M^m)^T = UD^m(V^TWV)D^mU^T$. If we set $W = L^{-1}$ so that $V^TWV = U'L^{1/2}L^{-1}L^{1/2}U'^T = I$, then Eqn. 77 can be written as

$$\text{Dist}^2(i_1, i_2) = (e_{i_1} - e_{i_2})^T UD^{2m}U^T(e_{i_1} - e_{i_2}) = \|y_{i_1} - y_{i_2}\|^2 \tag{79}$$

in which $y_i = D^mU^Te_i$, the "Diffusion Map".

PRML Errata

The following is the errors I found in PRML:

1) Eqn. (11.45) The leading term should be $p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z})A_k(\mathbf{z}', \mathbf{z})$. The associated derivations are thus changed accordingly.

2) In Eqn. (11.68), the factor $\min\{1, \exp(-H(\mathcal{R}) + H(\mathcal{R}'))\}$ should be $\min\{1, \exp(H(\mathcal{R}) - H(\mathcal{R}'))\}$ and In Eqn. (11.69), the corresponding factor should be altered similarly. The principle is that both expressions are exactly the same to demonstrate that the property of detailed balance holds.

3) Eqn. (11.10)-(11.11) The expression of Box-Muller Transform is not right. See Eqn. 36 for corrected version.

References

[1] http://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/Chapter11.pdf

[2] http://user.uni-frankfurt.de/~muehlich/sci/TalkBucurestiMar2003.pdf