10-708: Probabilistic Graphical Models, Spring 2020

7: Variational Inference I

Lecturer: Eric P. Xing *Scribe: Yihang Shen, Yizhou He, Alex Gaudio, Amrit Singhal*

Over the last a few lectures we have learned exact inference, but when is exact inference is expensive or even impossible? This motivates methods for approximate inference.

1 Overview of Variational Methods

- 15 years ago, a third of papers at NIPS were on this topic. The field has evolved tremendously.
- Variational Inference converts inference into optimization. - **Variational** is fancy name for optimization based formulations to approximate the desired solution by relaxing/approximating the intractable optimization problem.

Example: Linear system of equations when A is too large to invert:

$$Ax = b, A \succ 0$$

The solution is $x^\star = A^{-1}b$, however this is intractable for very large system as solving A^{-1} become very expensive.

This problem can be rewritten in variational formulation,

$$x^\star = \text{argmin}_x \frac{1}{2} x^T A x - bx$$

turns to be a optimization problem where conjugate gradient can be applied to the problem.

High level idea of Variational Inference:

There is a true distribution P that we wish to perform inference on, but we can't realistically do it. Therefore, we want to approximate P with another distribution Q. Variational Inference assumes the existence of P and Q, and also requires a way to measure distance between P and Q (usually KL Divergence).

- Inference: The distribution P
- Challenge: Direct inference on P is often intractable
- Variational approach: Project P to a tractable family of distributions Q.
- Perform inference on projected Q.
- Measure of distance: Test how good the approximation is, for example KL divergence.

Next class, we will cover the mathematics of variational inference in more detail. Now, we focus on applications.

2 Applications

2.1 The generalized algorithm design problem

More generally, how do we get started modeling a task? This is a design problem with many parts. These parts can be solved one step at a time.

- What is the task? Classification? Clustering?
- Data inputs and outputs: Is it continuous, binary or counts?
- Model: We need to choose one.
- Inference: Also need to choose one. For instance: Exact, Variational, MCMC
- Evaluation: How do we measure success? Visualize the results, score them, interpret them.

We will apply these steps to a task of text processing:

3 Probabilistic Topic Models

3.1 The Task and Data: Document Embedding on Text Documents

We have a stack of text documents. We want to represent relationships between documents for the purpose of grouping or categorizing them, as well as classifying and comparing similarity. We also want to be able to explore how documents change over time.

If we could meaningfully reduce each document to a point in some high dimensional space, we may be able to find a solution that is useful to all these problems. Our task is to create a document embedding. The inputs will a set of documents, each document contains text. The outputs will be a set of points in space.

3.2 Data Representation: A review of document embedding models

We consider various models that have been used to approach the document embedding task.

Bag of words representation.

Each document is a vector in the word space. This approach has significant disadvantage that it is too high dimensional, sparse, and ignores the order of words in a document.

Topic models

Topics, instead of words, intuitively tag, label or characterize a text. And a text may contain many topics. The idea is to recover for each document a single vector containing the mixing proportion of topics. In a PGM setting, do topic discovery on unstructured collection of texts to get a structured topic network, then generate a topic simplex. We will briefly present LSI and then connect it topic models.

Latent Semantic Index (LSI)

This is an old approach from the 70s, very similar to topic models. The basic idea is to perform singular value decomposition to decompose a word-document matrix into three sub-matrices. Primarily an algebraic solution rather than probabilistic one.

$$X = W \Lambda D^T$$

A singular value decomposition. Here X is the contexts collections, the x axis are documents, and the y axis represents word vectors. W relates topics to words, Λ relates topics to topics (the diagonal has singular values), and D^T relates documents to topics.

Relation to Topic models. Topic models have a very similar form. Algebraically, the only difference is the absence of a Λ matrix:

$$X = W D^T$$

where $X = P(w)$, $W = P(w \mid z)$, and $D^T = P(z)$.

Admixture models:

Description: Objects are bags of elements, mixtures are distributions over elements, mixing vector θ represents each mixtures' contributions for objects. Objects are generated from picking mixture components from θ and picking elements from that component.

Advantage: Allowing much richer mechanism for mixture of sources.

Differences between admixture models (similar to topic models) and mixture models:

For the mixture model, we assume that each document D is generated from a latent variable Z, the graph is represented as: $Z \rightarrow D$, but for the admixture model, each word W_i in a document is generated by a latent variable Z_i, and these latent variables are parameterized by θ, the graph can be constructed as: $\theta \rightarrow Z_i \rightarrow W_i$.

Figure 1 shows the graphical representation of topic models. The parameter θ is drawn from a prior distribution (in LDA it is Dirichlet distribution), then z_i is drawn from multinomial distribution with parameter θ, after that $w_i \mid z_i$ is drawn from multinomial distribution with parameter β, and β represents possible topics of the document.

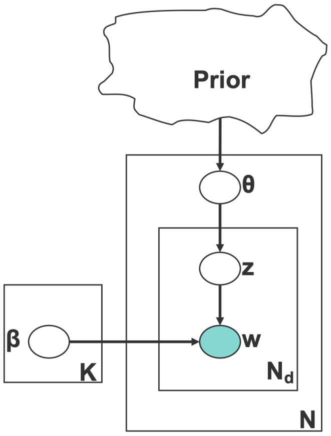


Figure 1: The graphical representation of a general topic model

LoNTAM

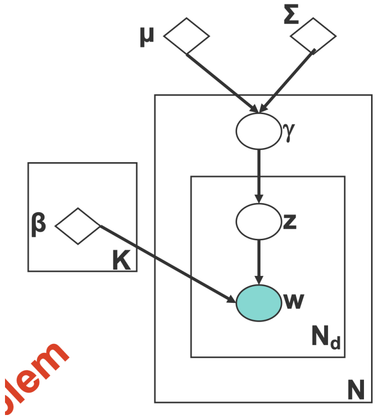


Figure 2: LoNTAM

There is another similar model called LoNTAM (Figure 2), the differences between Figure 2 and Figure 1 is that in LoNTAM, the prior used for sampling θ (γ in Figure 2) is multivariate Gaussian distribution.

Pros and cons of LDA and LoNTAM:

LDA:

Pros: Conjugate prior means more efficient inference.

Cons: Can only capture variations in each topic's intensity independently.

LoNTAM:

Pros: Capture the intuition that some topics are highly correlated and can rise up in intensity together.

Cons: Not a conjugate prior implies more difficult inference.

The joint distribution of Figure 1 is:

$$p(\beta, \theta, z, w) = \prod_{k=1}^{K} p(\beta_k \mid \eta) \prod_{d=1}^{D} p(\theta_d \mid \alpha) \prod_{n=1}^{N} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta)$$

However, it is very hard to do the exact inference according to that distribution, therefore we need methods for approximate inference.

3.3 Variational Inference

During class, slide 28 was missing many details. We fill them in here.

We have that:

$$\begin{aligned}
\log p(x) &= \log \frac{p(x,z,\theta)}{p(z,\theta \mid x)} = \log \frac{p(x,z,\theta)q(z,\phi \mid x)}{q(z,\phi \mid x)p(z,\theta \mid x)} \\
&= \int_z q(z,\phi \mid x) \log \frac{p(x,z,\theta)q(z,\phi \mid x)}{q(z,\phi \mid x)p(z,\theta \mid x)} dz \\
&= \int_z q(z,\phi \mid x) \log \frac{q(z,\phi \mid x)}{p(z,\theta \mid x)} dz + \int_z q(z,\phi \mid x) \log \frac{p(x,z,\theta)}{q(z,\phi \mid x)} dz \\
&= KL(q(z,\phi \mid x) \| p(z,\theta \mid x)) + \int_z q(z,\phi \mid x) \log \frac{p(x,z,\theta)}{q(z,\phi \mid x)} dz \\
&\geq \int_z q(z,\phi \mid x) \log \frac{p(x,z,\theta)}{q(z,\phi \mid x)} dz := L(\theta,\phi \mid x)
\end{aligned}$$

Instead of maximizing $\log p(x)$ we can maximize the lower bound $L(\theta,\phi \mid x)$, or equivalently, minimize free energy:

$$F(\theta,\phi \mid x) = -\log p(x) + KL(q(z,\phi \mid x) \| p(z,\theta \mid x))$$

We can use EM algorithm to maximize $L(\theta,\phi \mid x)$, for the E-step, fix θ, maximize L with respect to ϕ, if closed solution exists, we have

$$q^\star(z,\phi|x) \propto \exp(\log p(x,z,\theta))$$

For the M-step, fix ϕ, maximize L with respect to θ.

In the practical case of Figure 1, it is hard to do inference on the true posterior:

$$p(\beta,\theta,z \mid w) = \frac{p(\theta,\beta,z,w)}{p(w)}$$

Mean Field Approximation With mean field approximation, we assume q is fully factorized, that is:

$$q(\beta,\theta,z \mid \lambda,\gamma,\phi) = \prod_{k=1}^{K} q(\beta_k \mid \lambda_k) \prod_{d=1}^{D} q(\theta_d \mid \gamma_d) \prod_{n=1}^{N} q(z_{dn} \mid \phi_{dn})$$

We can have parametric form for each marginal factor:

$q(\beta_k \mid \lambda_k) = \text{Dirichlet}(\beta_k \mid \lambda_k)$

$q(\theta_d \mid \gamma_d) = \text{Dirichlet}(\theta_d \mid \gamma_d)$

$q(z_{dn} \mid \phi_{dn}) = \text{Multinomial}(z_{dn} \mid \phi_{dn})$

Therefore, we can learn parameters as the E-step:

$$\gamma^*, \lambda^*, \phi^* = argmin_{\gamma,\lambda,\phi} KL(q(\beta,\theta,z \mid \lambda,\gamma,\phi) \| p(\beta,\theta,z \mid w,\alpha,\eta))$$

Latent Dirichlet Allocation (LDA): For LDA, the optimal MF approximation can be computed in a closed form. Specifically, we define the following distributions:

- $p(\theta_d|\alpha) \propto \exp\left\{\sum_{k=1}^K (\alpha_k - 1)\log\theta_{dk}\right\}$ is the topic proportion distribution, which is Dirichlet.
- $p(z_{dn}|\theta_d) = \exp\left\{\sum_{k=1}^K \mathbf{1}[z_{dn} = k]\log\theta_{dk}\right\}$ is the word-label frequency distribution, which is Multinomial.

And plug them into the following update step, the result of which is a Dirichlet distribution:

$$\begin{aligned} q(\theta_d) &\propto \exp\left\{\mathbb{E}_{\prod_n q(z_{dn})}\left[\log p(\theta_d|\alpha) + \sum_n \log p(z_{dn}|\theta_d)\right]\right\} \\ &\propto \exp\left\{\sum_{k=1}^{K}\left(\sum_{n=1}^{N} q(z_{dn}=k) + \alpha_k - 1\right)\log\theta_{dk}\right\} \end{aligned}$$

Also update the marginals:

$$q(z_{dn}=k|\phi_{dn}) = \phi_{dn}(k) = \beta_k(w_{dn})\exp\left\{\Psi(\gamma_d(k)) - \Psi(\sum_{j=1}^{K}\gamma_d(j))\right\}$$

$$\lambda_k(j) = \eta(j) + \sum_{d=1}^{D}\sum_{n=1}^{N_d}\phi^*(k)\mathbf{1}[w_{dn}=j]$$

The $\lambda_k(j)$ term gives you a frequency (count) of the j^{th} word frequency in topic k, which is clear from the summation over every document and every word and the delta function $\mathbf{1}[...]$.

Iterating on these equations to convergence results in the MF approximation to the posterior. The algorithm formed by this iteration is coordinate ascent for LDA.

Variational Learning: the maximization step

For the M-step, usually people do not learn θ, as for the Bayesian, θ are integrated out. But one can still have MAP of θ, $MAP_{\theta,\beta}L(\theta,\phi \mid x)$, and one can even perform estimation of the hyper-parameters $MAP_{\alpha,\eta}L(\theta,\phi \mid x)$.

4 More on Mean Field Approximation

4.1 Naive Mean Field

The naive mean field algorithm approximates $p(X)$ by a fully factorized $q(X) = \prod_i q_i(X_i)$.

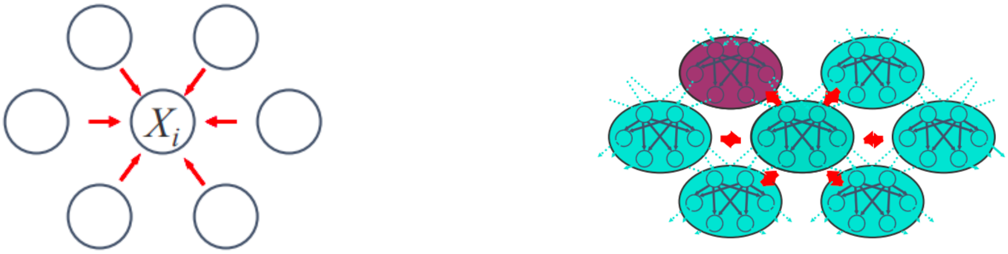
In the case of Boltzmann distribution

$$p(X) = \frac{1}{Z}\exp\left(\sum_{i<j} q_{ij}X_iX_j + q_{i0}X_i\right)$$

The mean field update equation for this is

$$\begin{aligned} q_i(X_i) &= \exp\left(\theta_{i0}X_i + \sum_{j\in N_i}\theta_{ij}X_i\langle X_j\rangle_{q_j} + A_i\right) \\ &= p(X_i|\{\langle X_j\rangle_{q_j} : j \in N_i\}) \end{aligned}$$

where $\langle X_j\rangle$ resembles a message sent from node j to node i. $\{\langle X_j\rangle_{q_j} : j \in N_i\}$ forms the "mean field" applied to X_i from its neighbourhood, as shown in Figure 3a.



(a) MF update

(b) GMF update

Figure 3: Messages passed while Mean field update

4.2 Generalised Mean Field

We can also apply more general forms of the mean field approximations, i.e. clusters of disjoint latent variables are independent, while the dependencies of latent variables in each clusters are preserved shown in Figure 3b.

4.2.1 From mean field approximation to Gibbs free energy

Assume that we are given a disjoint clustering $\{C_1, \ldots, C_K\}$ of all variables. Let

$$q(X) = \prod_{i=1}^{K} q_i(X_{C_i})$$

Then, we get the mean field energy as

$$G_{MF} = \sum_i \sum_{X_{C_i}} \prod_i q_i(X_{C_i}) E(X_{C_i}) + \sum_i \sum_{X_{C_i}} q_i(X_{C_i}) \ln q_i(X_{C_i})$$

For example, in the case of naive mean field, we get

$$G_{MF} = \sum i < j \sum_{x_i, x_j} q(x_i) q(x_j) \phi(x_i, x_j) + \sum_i \sum_{x_i} q(x_i) \phi(x_i) + \sum_i \sum_{x_i} q(x_i) \ln q(x_i)$$

Using clustering, no matter what clustering we use, we will **never** be able to be equal to the exact Gibbs free energy, but it **always** defines a lower bound of the likelihood.

So, we optimize each $q_i(X_{C_i})$, and do inference in each $q_i(x_c)$ using any tractable algorithm.

4.2.2 Generalised Mean Field Theorem

The optimum GMF approximation to the cluster marginal is isomorphic to the cluster posterior of the original distribution given the internal evidence and its generated mean fields

$$q_i^*(X_{H,C_i}) = p(X_{H,C_i} | X_{E,C_i}, \langle X_{H,MB_i} \rangle_{q_{j \neq i}})$$

The GMF algorithm is simply iterate over each q_i.

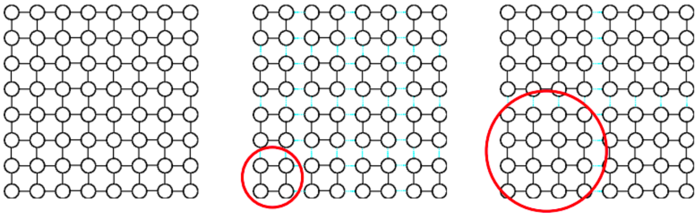


Figure 4: Generalised Mean Field for Ising models

4.2.3 Convergence Theorem

The GMF algorithm in guaranteed to converge to a local optimum, and provides a lower bound for the likelihood of the evidence(or partition function) of the model.

Example: GMF approximation to Ising models

Figure 4 shows two possible clusterings for the Ising model, one where we form 2×2 clusters, and the other where we form 4×4 clusters.

We obtain the cluster marginal of a square block C_k as

$$q(X_{C_k}) \propto \exp\left(\sum_{i,j \in C_k} \theta_{ij} X_i X_j + \sum_{i \in C_k} \theta_{i0} X_i + \sum_{i \in C_k, j \in MB_k, k' \in MB_{C_k}} \theta_{ij} X_i \langle X_j \rangle_{q(X_{C_{k'}})}\right)$$

Note that this is virtually a reparameterized Ising model of small size.

Key takeaway of class today

Know the Variational Inference principle, and in particular, the Mean Field Approximation principle. MFA is very powerful. You are effectively removing all/most of the edges resulting in a product of all singletons/clusters.