Jin Gu
Department of Automation, Tsinghua University
Email: jgu@tsinghua.edu.cn
Phone: (010) 62794294-866

Chapter 9

Inference as Optimization:

Structured Variational Inference

2021 Fall

Jin Gu (古槿)

Outlines

- Variational inferences
  - Basic idea
  - Evidence lower bound
- Mean field variational inferences
  - Basic methods
  - Example for Ising model
- Structured variational inferences
- Deep variational autoencoder

Chapter 9 Variational Inference & MAP Inference

Textbook2

Chapter 21 Variational Inference

Textbook1

Chapter 11.3.6 Variational Analysis of Belief Propagation

Chapter 11.5 Structured Variational Approximations

Chapter 13 MAP Inference

Other references

[1] Doersch C. **Tutorial on variational autoencoders**.
https://arxiv.org/abs/1606.05908

[2] Kingma DP, Welling M. **Auto-encoding variational Bayes**.
https://arxiv.org/abs/1312.6114

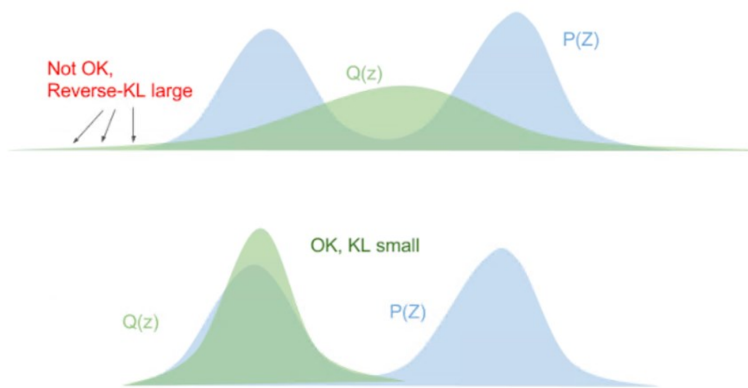
[3] Blei MN, Ng AY and Jordan MI. **Latent Dirichlet Allocation**. *Journal of Machine Learning Research* 2003, 3:993-1022.

What is Variational Inference?

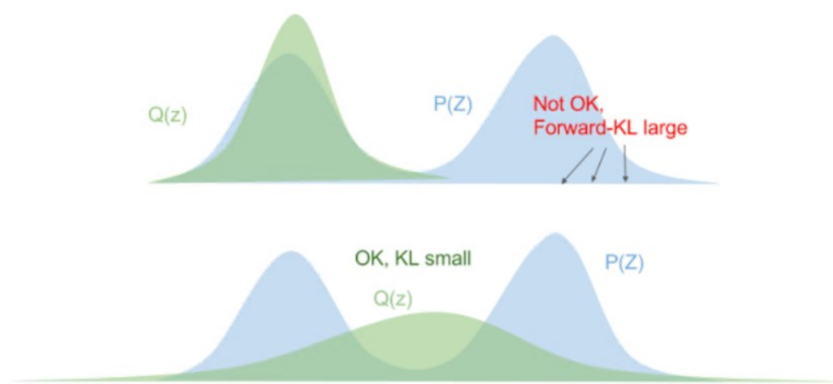
- To infer $P(X)$, find another distribution $Q(X)$ which can approximate it
  - Target distribution: $P(X)$
  - Proposal distribution: $Q(X)$
    - Q is restricted to a family of distribution with simple form
  - KL divergence: $\boldsymbol{D_{KL}(Q||P)}$ or $D_{KL}(P||Q)$
    - $D_{KL}(Q||P) = E_{X\sim Q}\left(\ln\left(\frac{Q}{P}\right)\right)$
  - Aim: $\min_{Q} D_{KL}(Q||P)$
- Structured variational inference
  - Define $Q(X)$ based on the graph with simple structure
    - E.g. A graph with no edge (*mean field algorithm*)

Comments On the Two Divergences

- Assume $Q(X)$ is restricted as Gaussians
  - Reverse KL (I-projection): $\boldsymbol{D_{KL}(Q||P)}$
  - Forward KL (M-projection): $D_{KL}(P||Q)$



Reverse KL: Q should model one of the peaks in $P > 0$



Forward KL: Q should cover $P > 0$

Q: Why reverse KL is commonly used?

Calculations in Variational Inferences

- I-projection
  - General conditional target distribution $P(X|Z{=}z)$

$$D_{KL}(Q||P) = E_{X\sim Q}\left(\ln \frac{Q(X)}{P(X|z)}\right) \qquad P(X|z) = P(X,z)/P(z)$$

$$D_{KL}(Q||P) = E_{X\sim Q}\left(\ln \frac{Q(X)}{P(X,z)}\right) + \ln P(z)$$

$$\ln P(z) - D_{KL}(Q||P) = -E_{X\sim Q}\left(\ln \frac{Q(X)}{P(X,z)}\right) = E_{X\sim Q}(\ln P(X,z)) + H(Q(X))$$

- Two key questions:
  - ***Q1***: How to choose the family of the proposal distribution?
  - ***Q2***: How to maximize the above energy functional?

Exact Inference as Optimization

- Energy functional

  $D_{KL}(Q||P_\Phi) = \ln Z - \left(H(Q(X) + \sum_{\phi\in\Phi} E_Q(\phi)\right)$

  – The second term is energy functional $F[\tilde{P}_\Phi, Q]$

- For a clique tree with clique C_i, we have a set of beliefs $\boldsymbol{Q}$ ($\beta_i, \mu_{i,j}$ - not calibrated). Its energy functional:

  $\tilde{P}_\Phi = \prod_i \psi_i$ (ψ_i is the initial factor for each clique)

$$\tilde{F}[\tilde{P}_\Phi, \boldsymbol{Q}] = \sum_i H_{\beta_i}(C_i) - \sum_{(i-j)\in\mathcal{E}} H_{\mu_{i,j}}\left(S_{i,j}\right) + \sum_i E_{\beta_i}[\ln \psi_i]$$

Exact Inference as Optimization

- If $\boldsymbol{Q}$ is calibrated as $Q*$, we can conclude that
$$\tilde{F}[\tilde{P}_\Phi, Q^*] = \max_{\boldsymbol{Q}} \tilde{F}[\tilde{P}_\Phi, \boldsymbol{Q}] = \ln Z$$
- Because the distribution is invariant for any calibrated beliefs, the relative entropy is minimized as 0
- Transform SP/BU algorithm as optimization

CTree-Optimize-KL:

Find $Q = \{\beta_i : i \in \mathcal{V}_T\} \cup \{\mu_{i,j} : (i\text{–}j) \in \mathcal{E}_T\}$
maximizing $-\boldsymbol{D}(Q \| P_\Phi)$
subject to

$$\mu_{i,j}[s_{i,j}] = \sum_{C_i - S_{i,j}} \beta_i(c_i) \quad \forall (i\text{–}j) \in \mathcal{E}_T, \forall s_{i,j} \in Val(S_{i,j})$$

$$\sum_{c_i} \beta_i(c_i) = 1 \qquad \forall i \in \mathcal{V}_T.$$

Energy Functional

- Key is to *find a Q* to *minimize the KL distance* that is equal to *maximize the energy functional*
  $L(Q) = E_{X\sim Q}(\ln P(X,z)) + H(Q)$
  - Or minimize $J(Q) = -L(Q)$

- Normally, Q is restricted to a distribution family
- The first term: $E_{X\sim Q}(\ln P(X,z))$
  - Q should mimic P
- The second term: $H(Q)$
  - Q should diffuse (reduce overfitting)

$$\boxed{D_{KL}(Q||P_\Phi) = \ln Z - \left(H(Q(X) + \sum_{\phi\in\Phi} E_Q(\phi)\right)}$$

Evidence Lower Bound (ELBO)

- Because KL divergence is non-negative, the energy functional is bound by $\ln P(z)$

$$L(Q) = E_{X\sim Q}(\ln P(X,z)) + H(Q)$$
$$= \ln P(z) - D_{KL}(Q||P) \leq \ln P(z)$$

- So, we also call $L(Q)$ as ELBO
- Find a distribution Q to maximize the ELBO

What to **LEARN** in this lecture?

- Simplest case: treat all variables independently (no edge) in the proposal distribution Q

Mean field inference (平均场)

- Extended to general structures of proposal distribution Q (structured VI)
- Use deep NNs to do variational inferences

Mean Field Variational Inference

- Now, we need to calculate (infer)
  - $p(x|z)$ [*Note: p or P are treated as the same here*]
  - But, it is usually hard to calculate…
- We want to use a simplified distribution *q* to mimic the distribution *p*

- The simplest distribution family is in *fully factorized* format

$$q(x;\theta) = \prod_i q_i(x_i)$$

Mean Field Variational Inference

- Recall the energy functional
  $L(q) = E_{x \sim q}(\ln p(x,z)) + H(q)$

- For fully factorized $q = \prod_i q_i(x_i)$

$$E_{x \sim q}(\ln p(x,z)) = \sum_x \left[\prod_i q_i(x_i)\right] \ln p(x,z)$$

$$H(q) = \sum_x \left[\prod_i q_i(x_i)\right] \left[-\sum_i \ln q_i(x_i)\right]$$

- So finally, we get

$$L(q) = \sum_x \left[\prod_i q_i(x_i)\right] \left[\ln p(x,z) - \sum_k \ln q_k(x_k)\right]$$

Mean Field Variational Inference

- We can derive the energy function for each q_j, assuming the other terms q_{-j} are fixed

$$L(q_j) = \sum_x \prod_i q_i(x_i)\left[\log \tilde{p}(x) - \sum_k \log q_k(x_k)\right]$$

$$= \sum_{x_j} \sum_{x_{-j}} q_j(x_j) \prod_{i \neq j} q_i(x_i)\left[\log \tilde{p}(x) - \sum_k \log q_k(x_k)\right]$$

$$= \sum_{x_j} q_j(x_j) \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \log \tilde{p}(x)$$

$$- \sum_{x_j} q_j(x_j) \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i)\left[\sum_{k \neq j} \log q_k(x_k) + \log q_j(x_j)\right]$$

$$= \sum_{x_j} q_j(x_j) \log f_j(x_j) - \sum_{x_j} q_j(x_j) \log q_j(x_j) + \text{const}$$

$$\boxed{\log f_j(x_j) \triangleq \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \log \tilde{p}(x) = \mathbb{E}_{-q_j}[\log \tilde{p}(x)]}$$

Mean Field Variational Inference

- Maximize the energy $L(q_j)$

$$L(q_j) = \sum_{x_j} q_j(x_j) \log \left[\frac{\exp\left\{E_{-q_j}[\ln \tilde{p}(x)]\right\}}{q_j(x_j)} \right] + c$$

- Recall the KL divergence

$$D_{KL}(q||\tilde{p}) = -\sum q \ln \frac{\tilde{p}}{q}$$

$q^* \propto \tilde{p}$ will minimize the divergence

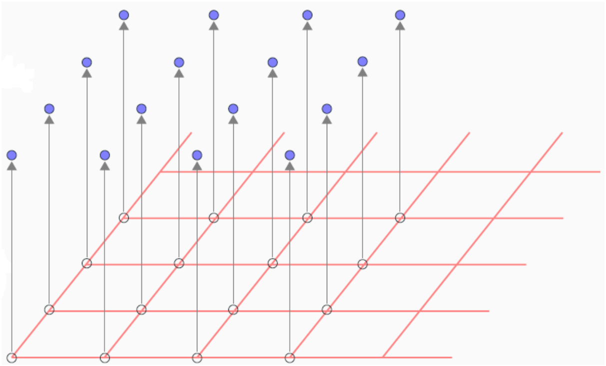
KEY: $L(q_j)$ follows a form of KL divergence

- **We can easily see the solution:**

$$\boldsymbol{q_j(x_j) \propto \exp\left\{E_{-q_j}[\ln \tilde{p}(x)]\right\}}$$

An Example for Ising Model

- The hidden variables x_i are binary and z_i are always observed. (Please note *the edges are directed* from x to z)



- Aim: calculate $E_{-q_j}[\ln\tilde{p}(x)]$
- The target distribution P

  $p(x|z) \propto p(z|x)p(x)$

  - $p(x) = \frac{1}{Z} e^{\sum_{i,j} w_{ij} x_i x_j} = \frac{1}{Z} e^{-E_0(x)}$
  - $p(z|x) = e^{\sum_i \ln p(z_i|x_i)} = e^{\sum_i L_i(x_i)}$

$$\boxed{q_j(x_j) \propto \exp\left\{E_{-q_j}[\ln\tilde{p}(x)]\right\}}$$

- We can rewrite as

$$\ln \tilde{p}(x) = \sum_{i,j} w_{ij} x_i x_j + \sum_i L_i(x_i) + const$$

Mean Field for Ising Model

- Set the family of the variational distribution $q = \prod_i q(x_i;\theta_i)$.
  - Each $q(x_i;\theta_i)$ is a *binary* distribution
    - $q_i(x_i = 1) = \theta_i$ and $q_i(x_i = -1) = 1 - \theta_i$
- NOTE: the parameter θ_i is not determined
- KEY: variational inference here is to find the optimal parameter θ_i^* to maximize the energy functional (then we get q^*)
- AIM: inference in q^* rather than in p

$$\boxed{q_j(x_j) \propto \exp\left\{E_{-q_j}[\ln\tilde{p}(x)]\right\}}$$

Mean Field for Ising Model

- Recall the previous result

  $\ln q_j^*(x_j) = E_{-q_j}[\ln\tilde{p}(x)] + const$

Please remember the above optimal solution for mean-field inference

- For Ising model

$$\ln \tilde{p}(x) = \sum_{i,j} w_{ij} x_i x_j + \sum_i L_i(x_i) + const$$

$$E_{-q_j}[\ln\tilde{p}(x)] = E_{-q_j}\left(\sum_{i,j} w_{ij} x_i x_j + \sum_i L_i(x_i)\right) + c$$

$$= x_j \sum\nolimits_{i \in Neighbor(j)} \left[w_{ij}(2\theta_i - 1)\right] + L_j(x_j) + c$$

  - Set $m_j = \sum_{i \in Neighbor(j)} \left[w_{ij}(2\theta_i - 1)\right]$

$$\boxed{q_j(x_j) \propto \exp\left\{E_{-q_j}[\ln\tilde{p}(x)]\right\}}$$

Mean Field for Ising Model

- So we can get the optimal solution in this step

$$\ln q^*(x_j;\theta_j) = x_j m_j + L_j(x_j) + const$$

- Obviously, the optimal parameter

$$\theta_j^* = \frac{\exp[m_j + L_j(x_j=1)]}{\exp[m_j + L_j(x_j=1)] + \exp[-m_j + L_j(x_j=-1)]}$$

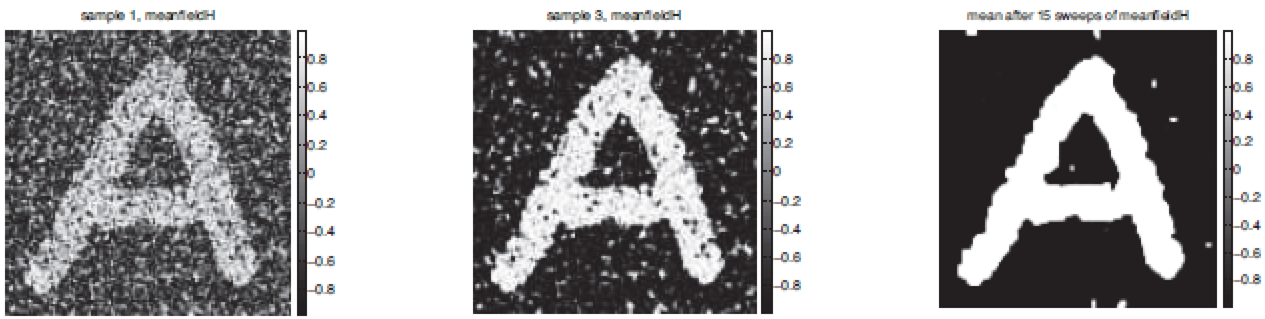
  - Here $L_j(x_j) = \ln p(z_j|x_j)$

$$\boxed{m_j = \sum_{i \in Neighbor(j)} [w_{ij}(2\theta_i - 1)]}$$

- We need to update the parameters by iteration, because we need to specify $\boldsymbol{\theta}_{-j}$

$$\boxed{q_j(x_j) \propto \exp\{E_{-q_j}[\ln\tilde{p}(x)]\}}$$

Mean Field Inference for De-noising



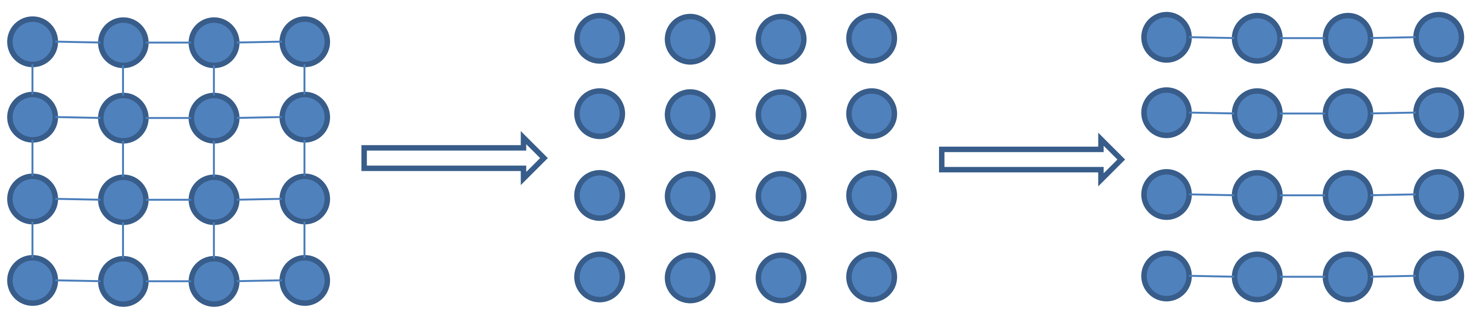
(a) (b) (c)

Figure 21.3 Example of image denoising using mean field (with parallel updates and a damping factor of 0.5). We use an Ising prior with $W_{ij} = 1$ and a Gaussian noise model with $\sigma = 2$. We show the results after 1, 3 and 15 iterations across the image. Compare to Figure 24.1. Figure generated by `isingImageDenoiseDemo`.

Infer the probability of the hidden variables according to *q* rather than *p*. Sometimes, the variational probability is "better" than the target probability: *de-noising* or *smoothing* effects.

General Structured Variational Inference

- Graph for mean field inference
  - No edge is in the variational distribution family
- General structured varational inference
  - We can use any simpler structure to do inference



$p(x|z)$ Mean field Linear chain

General Structured Variational Inference

- If we use linear chain for Ising model
- Recall: *find a Q* to *minimize the KL distance* that is equal to *maximize the energy functional*

$$L(q) = E_{x\sim q}(\ln p) + H(q)$$

- We can set $q \propto \prod_l q_l(x_l)$

$p(x|z)$ Linear chain

$$\boxed{L(q) = E_{x\sim q}(\ln p) + H(q)}$$

- $q_i(\boldsymbol{x_i}) = \frac{1}{Z_l} e^{\sum_j \beta_{j,j+1} x_j x_{j+1}}$
- $p = \frac{1}{Z} e^{\sum_{i,j} w_{i,j} x_i x_j}$
- $E_{x\sim q}(\ln p) = \sum_x \left(-\ln Z + \sum_{i,j} w_{ij} x_i x_j\right) \prod_i q_i$
- $H(q) = \sum_x \left(-\sum_i \ln q_i\right) \prod_i q_i$

Each chain has a separate variational Gibbs distribution.

Maximize the energy functional **chain by chain**!

Variational Inference for LDA

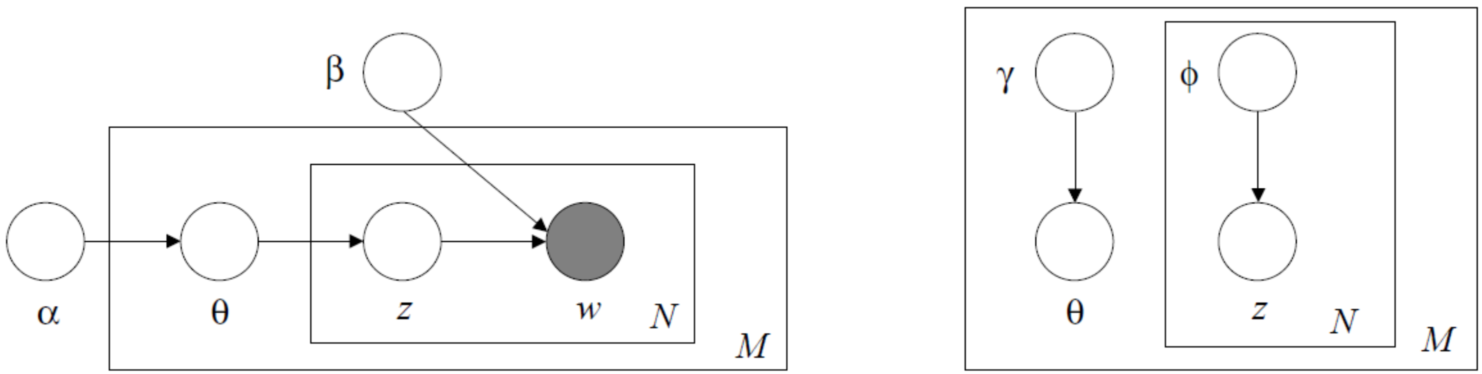


Figure 5: (Left) Graphical model representation of LDA. (Right) Graphical model representation of the variational distribution used to approximate the posterior in LDA.

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta)$$

$$q(\theta, \mathbf{z} \mid \gamma, \phi) = q(\theta \mid \gamma) \prod_{n=1}^{N} q(z_n \mid \phi_n)$$

NOTE: variables should be the same and parameters are different

Variables: θ and z
Parameters: γ and ϕ_n

General Structured Variational Inference

- For a general variational structures, we can finally derive:

$$\psi_j(\boldsymbol{c}_j) \propto \exp\left\{ \boldsymbol{E}_Q\left[\ln \tilde{P}_\Phi \mid \boldsymbol{c}_j\right] - \sum_{k \neq j} \boldsymbol{E}_Q[\ln \psi_k \mid \boldsymbol{c}_j] \right\}$$

- Comments:
  - The energy functional can be separated into maximal cliques on q
  - The update also depends on the expectation

Stochastic Gradient for Parameter Update

- In many cases, we cannot derive the close-form solution as the mean field inference
- Many heuristic optimization methods can be used to optimize the target function
  - Remember the belief propagation on cluster graph
  - Stochastic gradient is commonly used
- Stochastic gradient (for maximizing)

$$L(q) = E_{x\sim q}(\ln p) + H(q)$$

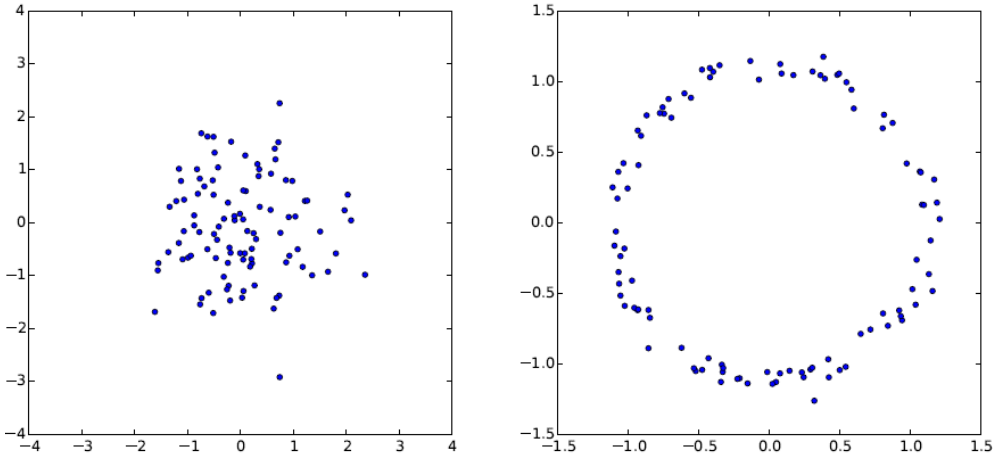
  - Calculate the partial derivate ${\partial L(q)}/{\partial \theta_i}$
  - Then, update $\theta_i^{t+1} = \theta_i^t + \delta \frac{\partial L(q)}{\partial \theta_i^t}$

Limitation of Variational Inference

- The variational inference distribution family is hard to choose
  - Simple family cannot capture the complex features in target distribution
  - Complex family is hard for computation and easy for over-fitting

NNs for Variational Inference

- Assume the observed data $X \in R^N$ are generated from a hidden subspace $Z \in R^L$ as standard Gaussian
- In most cases, the posterior $P(Z|X)$ is hard to infer



$$g(z) = z/10 + z/||z||$$

NNs for Variational Inference

- Another distribution $Q(Z|X)$ can be used to approximate $P(Z|X)$ with respect to the minimization of KL-divergence

$$\log P(\boldsymbol{X}) - \boxed{D[Q(\mathbf{z}|\boldsymbol{X})||P(\mathbf{z}|\boldsymbol{X})]}$$
$$= \mathrm{E}_{\mathbf{z}\sim Q}[\log P(\boldsymbol{X}|\mathbf{z})] - D[Q(\mathbf{z}|\boldsymbol{X})||P(\mathbf{z})]$$

First term: maximize the data generation probability
Second term: minimize the KL-divergence between standard Gaussian and the variational distribution

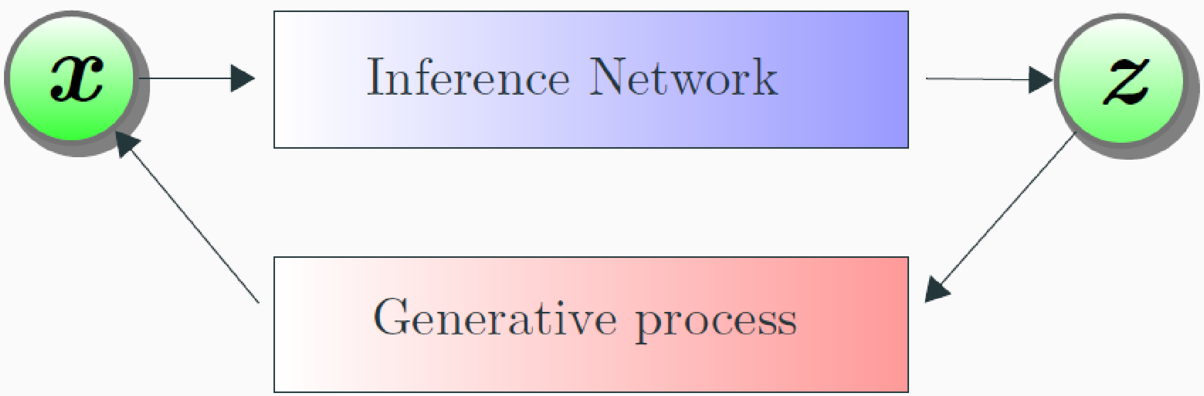
- The two terms can be treated as the "decoding" (from **z** to ***X***) and "encoding" (from ***X*** to **z**) process. Two deep neural networks can be used to approximate these terms

NNs for Variational Inference

VAE idea: Use neural networks to approximate both **variational** and **generative** parameters.

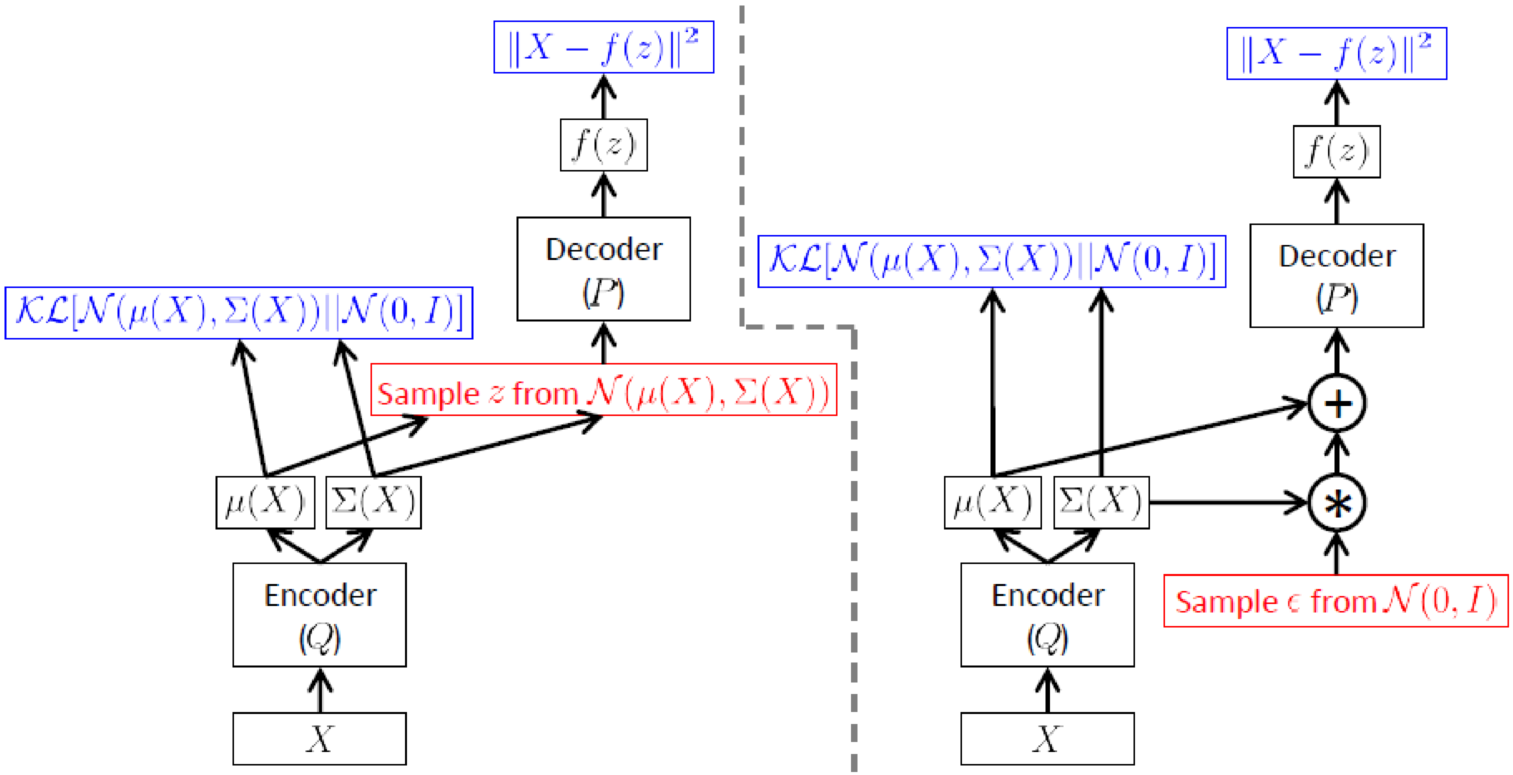
$$\begin{aligned}\mathcal{L}(\theta; \boldsymbol{x}) &= \mathbb{E}_{z \sim q(z|x;\theta)} \log p(\boldsymbol{x}, \boldsymbol{z}) + \mathbb{H}(q(z|x;\theta)) \\ &= \mathbb{E}_{z \sim q(z|x;\theta)} \log p(\boldsymbol{x}|\boldsymbol{z}) - \mathbb{KL}(q(\boldsymbol{z}|\boldsymbol{x};\theta) \,||\, p(\boldsymbol{z}))\end{aligned}$$

$q(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(f_\mu(\boldsymbol{x}), f_\sigma(\boldsymbol{x}))$



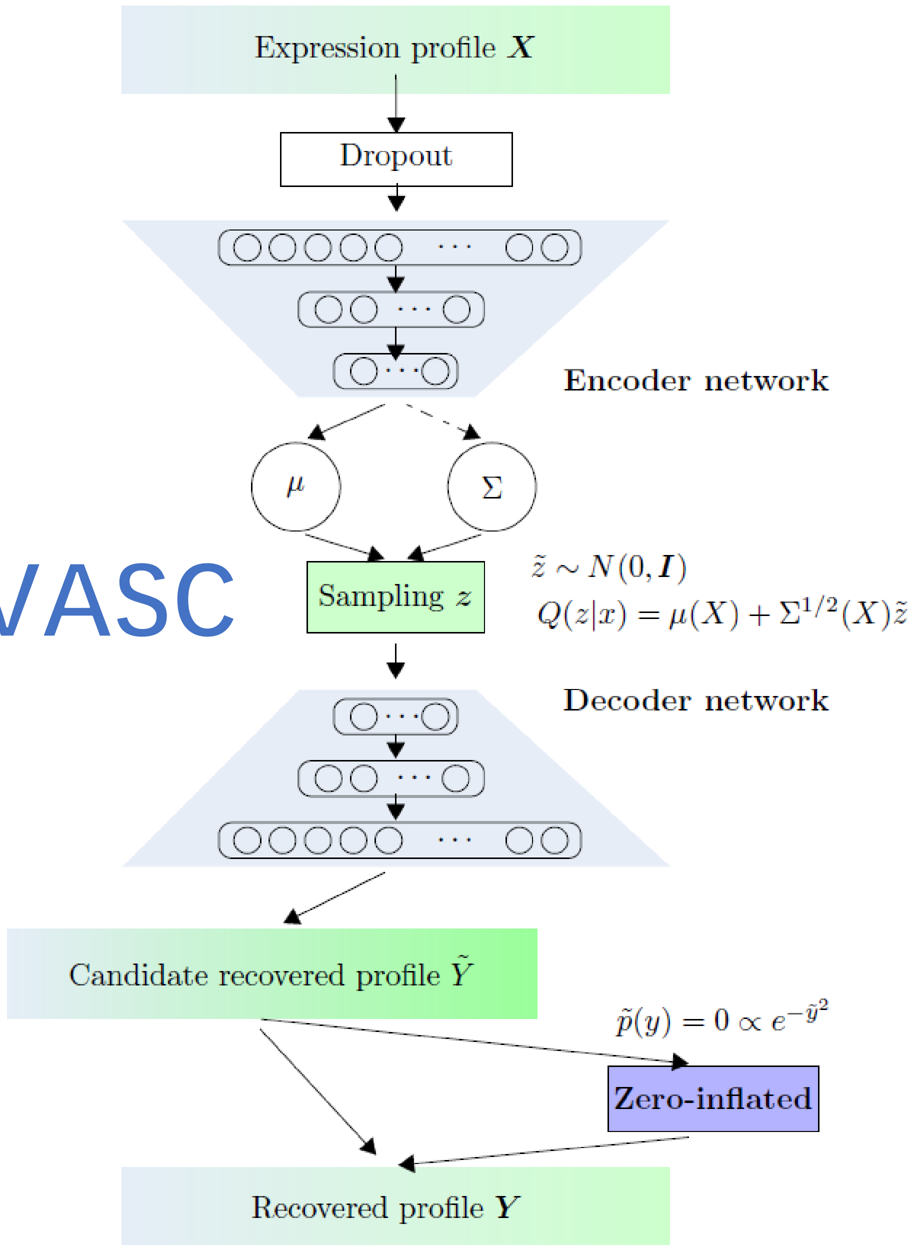
$p(\boldsymbol{x}|\boldsymbol{z})$

NNs for Variational Inference



Re-parameterization trick for learning

VASC



$\text{Loss}(\boldsymbol{X}, \boldsymbol{Y}) = \text{binary\_entropy}(\boldsymbol{X}, \boldsymbol{Y}) + \text{KL}(\ Q(z|\boldsymbol{X}) || P(z)\ )$

Input: scRNA-seq FPKM matrix

Additional random dropout!!

Three-layer encoder NNs:
- L1 norm in first layer
- ReLU activation

Target: Low-dimension representation by sampling

Three-layer decoder NNs:
ReLU activation

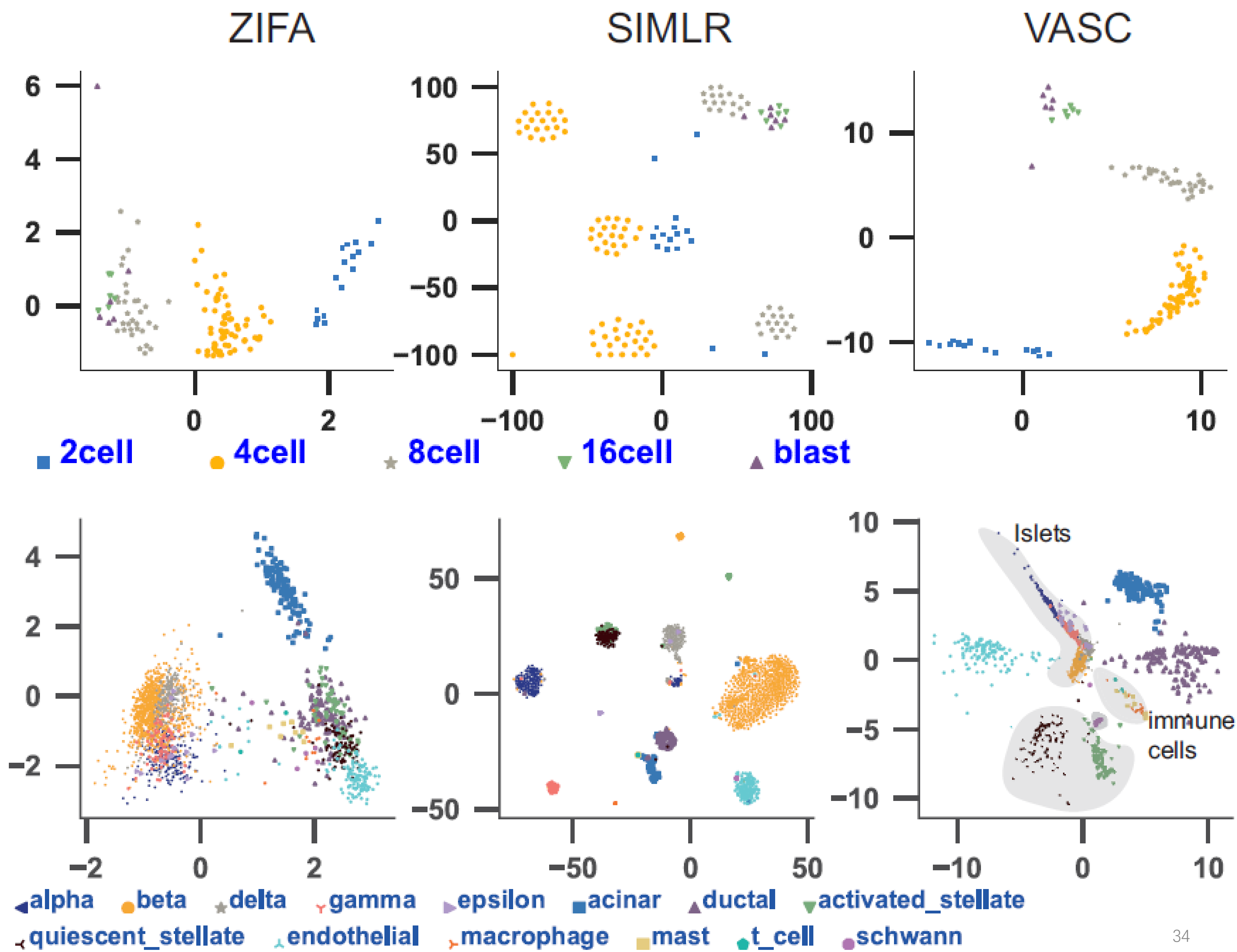
Model dropouts by Gumbel distribution instead of “hard” ZI

Output: recovered FPKM matrix

Advantages of VAE

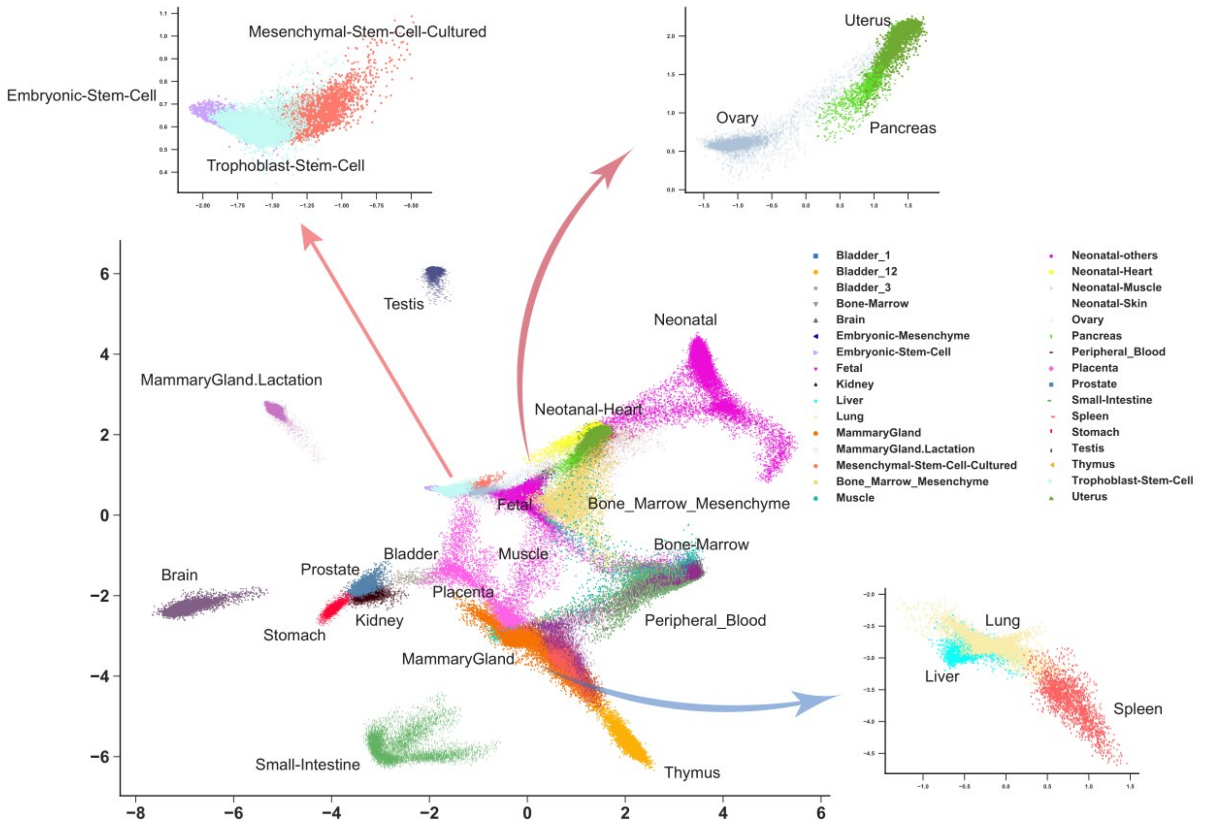
- Unequal structures in encoding & decoding networks
  - VAE is doing inference rather than autoencoding
  - Encoding/decoding NNs are used to mimic the two terms in the variational energy functional
  - So, we can freely design the topology of two NNs

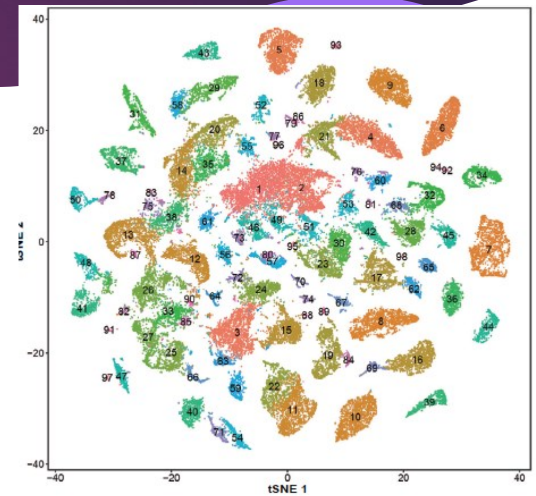
- Dimension reduction
  - Use the inferred mean value calculated by the encoding network $\mu(X)$
- A generative model
  - Fill the missing values and generate new data

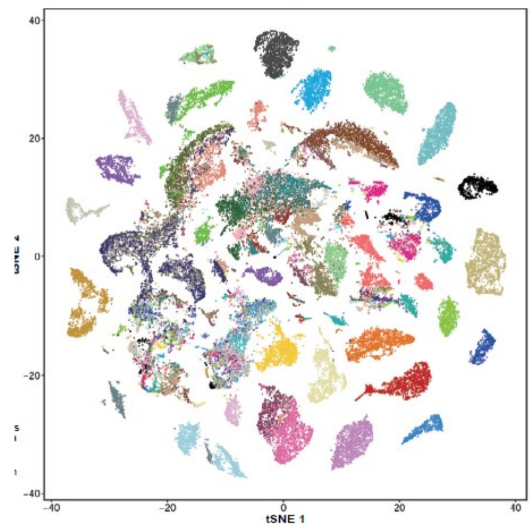


ZIFA
SIMLR
VASC
2cell
4cell
8cell
16cell
blast
Islets
immune cells
alpha
beta
delta
gamma
epsilon
acinar
ductal
activated_stellate
quiescent_stellate
endothelial
macrophage
mast
t_cell
schwann

Test on 10X & Microwell datasets







t-SNE (right) only consider the local similarity
VASC preserves the topology of cell expression patterns

Summary

- Variational inference is to find a simpler distribution q to mimic the target distribution p. Then, you can do easy inference in q.

- Variational inference is an *optimization* process to find a variational distribution q from a distribution family by *maximizing an energy functional*.

- Then variational distribution is usually simpler than the target distribution. So, it can be used for de-noising or dimension reduction.

Comments: From Inference to Learning

- Variational inference can be regarded as a kind of learning: find the optimal parameters by maximizing an energy functional
  - $\max_q L(q)$ $\qquad L(q) = E_{x\sim q}(\ln p(x|z)) + H(q)$

- Learning as MLE: find the optimal parameters by maximizing likelihood function given data
  - $\max_\theta L(D;\theta)$

The End of Chapter 9

Structured variational inference is commonly used in PGMs

MAP Inference

- Probability inference
  - Compute the distribution $P(\boldsymbol{Y})$ or $P(\boldsymbol{Y}|\boldsymbol{e})$
- MAP (maximum a posterior) inference
  - Compute the optimal assignment or configuration
  - $\text{MAP}(\boldsymbol{Y}|\boldsymbol{e}) = \arg\max_{\boldsymbol{y} \in Val(\boldsymbol{Y})} P(\boldsymbol{y}, \boldsymbol{e})$
- The max-marginal of a function f
  - $\text{MaxMarg}_f(\boldsymbol{y}) = \max_{\boldsymbol{y} \in Val(\boldsymbol{Y})} f(\boldsymbol{y})$
  - $\text{MaxMarg}_F(\boldsymbol{y}) = \max_{\boldsymbol{y} \in Val(\boldsymbol{Y})} \sum_{\boldsymbol{X}=\boldsymbol{x}} F(\boldsymbol{y}, \boldsymbol{X})$

Compute $\underset{Y}{\operatorname{argmax}} \boldsymbol{P}(\boldsymbol{X}, \boldsymbol{Y} | \boldsymbol{\theta})$

Viterbi algorithm Revisited

- Known transition matrix and emission matrix
- Infer the ***maximum*** probability STATE series
- The probability at time 0 with observation x_0

$$For\ Y_1 = i: \quad \delta_{1,i} = \pi_i e_{i,x_1}$$

- The probability at time 1 with observation x_1

$$\delta_{2,i} = e_{i,x_2} \max_{y_1=1,\cdots} \left(\pi\ e\ \times t \right) = e \max_{\cdots} \left(\delta \times t \right)$$

$$\phi_2(i) = \underset{y_1=1,\cdots}{\arg\max} \left(\delta_{1,y_1} \times t_{y_1,i} \right)$$

Compute $\arg\max_{Y} \boldsymbol{P}(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{\theta})$

Viterbi algorithm Revisited

- The probability at time t with observation x_t

$$\delta_{t,i} = e_{i,x_t} \max_{y_{t-1}=1,\cdots} \left(\delta_{t-1,y_{t-1}} \times t_{y_{t-1},i} \right)$$

$$\phi_t(i) = \arg\max_{y_{t-1}=1,\cdots} \left(\delta_{t-1,y_{t-1}} \times t_{y_{t-1},i} \right)$$

- For the last observation

$$\delta_{T,i} = e_{i,x_T} \max_{y_{T-1}=1,\cdots} \left(\delta_{T-1,y_{T-1}} \times t_{y_{T-1},i} \right)$$

$$\phi_T(i) = \arg\max_{y_{T-1}=1,\cdots} \left(\delta_{T-1,y_{T-1}} \times t_{y_{T-1},i} \right) \qquad \boxed{y_T^* = \arg\max_{y_T=1,\cdots} \left(\delta_{T,y_T} \right)}$$

Compute $\arg\max_{Y} P(X, Y|\theta)$

Viterbi algorithm Revisited

- After y_t is inferred, trace back to get other y_i

$$y_T^* = \arg\max_{y_T = 1, \cdots} \left(\delta_{T, y_T} \right)$$

$$\phi_t(i) = \arg\max_{y_{t-1} = 1, \cdots} \left(\delta_{t-1, y_{t-1}} \times t_{y_{t-1}, i} \right)$$

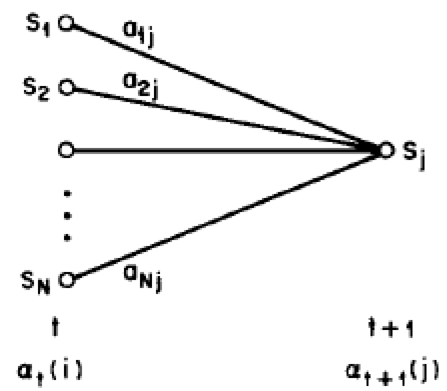
- We can infer other y_i

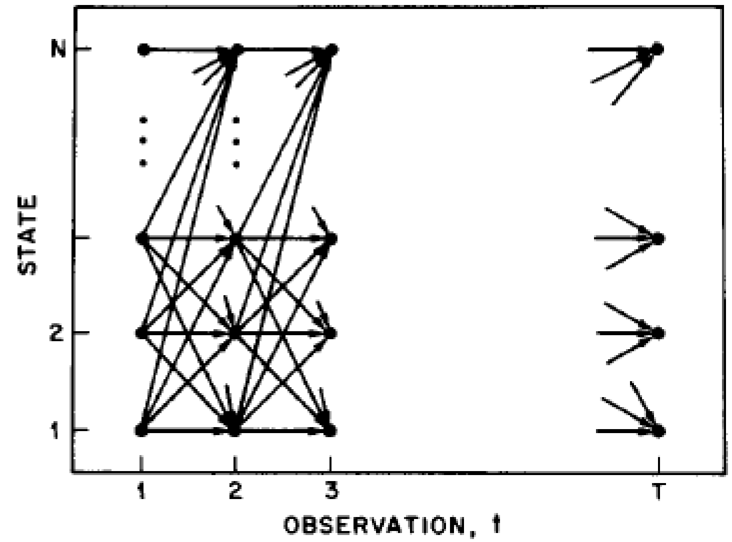
$$y_{T-1}^* = \phi_T\left(i = y_T^*\right) = \arg\max_{y_{T-1} = 1, \cdots} \left(\delta_{T-1, y_{T-1}} \times t_{y_{T-1}, i} \right)$$

$$y_{t-1}^* = \phi_t\left(i = y_t^*\right) = \arg\max_{y_{t-1} = 1, \cdots} \left(\delta_{t-1, y_{t-1}} \times t_{y_{t-1}, i} \right)$$

Viterbi Algorithm Revisited

- Do variable elimination and record the paths which maximizing the joint distribution for the first ***k*** time slices after ***k***-th run of elimination
- After all query variables are eliminated, trace back the path to get the configuration which maximizes the joint distribution





- *Could we generalize Viterbi algorithm to Bayesian Networks and Markov Networks?*

Move to ***Learning***

- Representation $P \Leftrightarrow \{P, G\}$

- Inference $P\left(\boldsymbol{Y} \mid \boldsymbol{E} = \boldsymbol{e}, \boldsymbol{\theta}\right)$

- Learning

$$\boxed{\max_{\theta} P\left(\boldsymbol{x}[1], \boldsymbol{x}[2], \cdots \qquad \mid \boldsymbol{\theta}\right)}$$

$$\boxed{P\left(\boldsymbol{\theta} \mid \boldsymbol{x}[1], \boldsymbol{x}[2], \cdots \qquad \right)}$$