CS 8501 Advanced Topics in Machine Learning

Lecture 08: Variational Inference (II)

Yangfeng Ji

Information and Language Processing Lab

Department of Computer Science

University of Virginia

https://yangfengji.net/

Variational Bayes EM

Latent Variable Models

Consider a latent variable model $p(x, z; \theta)$, where z is latent variable and θ denotes all model parameters.

The conceptual way of learning a latent variable model is

$$\hat{\theta} \leftarrow \mathrm{argmax}_{\theta'} p(x; \theta)$$

Gaussian Mixture Models

Consider a specific example of latent variable model: Gaussian mixture model

$$p(x;\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x;\mu_k,\Sigma_k)$$

With N training examples $\{x^{(n)}\}_{n=1}^{N}$, we have the log-likelihood function

$$\sum_n \log p(x^{(n)};\theta) = \sum_n \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x;\mu_k,\Sigma_k)$$

There is no closed-form solution for this problem

Gaussian Mixture Models (II)

The equivalent formulation with latent variable $z = (z_1, \ldots, z_K)$ as a categorical random vector with unknown parameter $\gamma = (\gamma_1, \ldots, \gamma_K)$

$$p(x, z; \theta, \gamma) = \prod_{k=1}^{K} p(z = k; \gamma) \cdot \mathcal{N}(x; \mu_k, \Sigma_k)$$

With N training examples,

$$\sum_n \log p(x^{(n)}, z^{(n)}; \theta, \gamma^{(1:N)}) = \sum_n \sum_k \{\log p(z^{(n)} = k; \gamma^{(n)}) + \log N(x^{(n)}; \mu_k, \Sigma_k)\}$$

Each training example has its own latent variable!

Parameters

In this latent variable, θ has two sets

- associated with each individual Gaussian component

$$\theta = \{\mu_k, \Sigma_k\}_{k=1}^{K}$$

- associated with each training example

$$\gamma^{(1:N)} = \{\gamma_1^{(n)}, \ldots, \gamma_K^{(n)}\}_{n=1}^{N}$$

  Recall that z_n is a categorical random variable with its **posterior distribution**

$$p(z^{(n)} = k; \gamma^{(n)}) = \gamma_k^{(n)}$$

EM Algorithm

The EM algorithm alternates between the two sets of parameters with the following two steps:

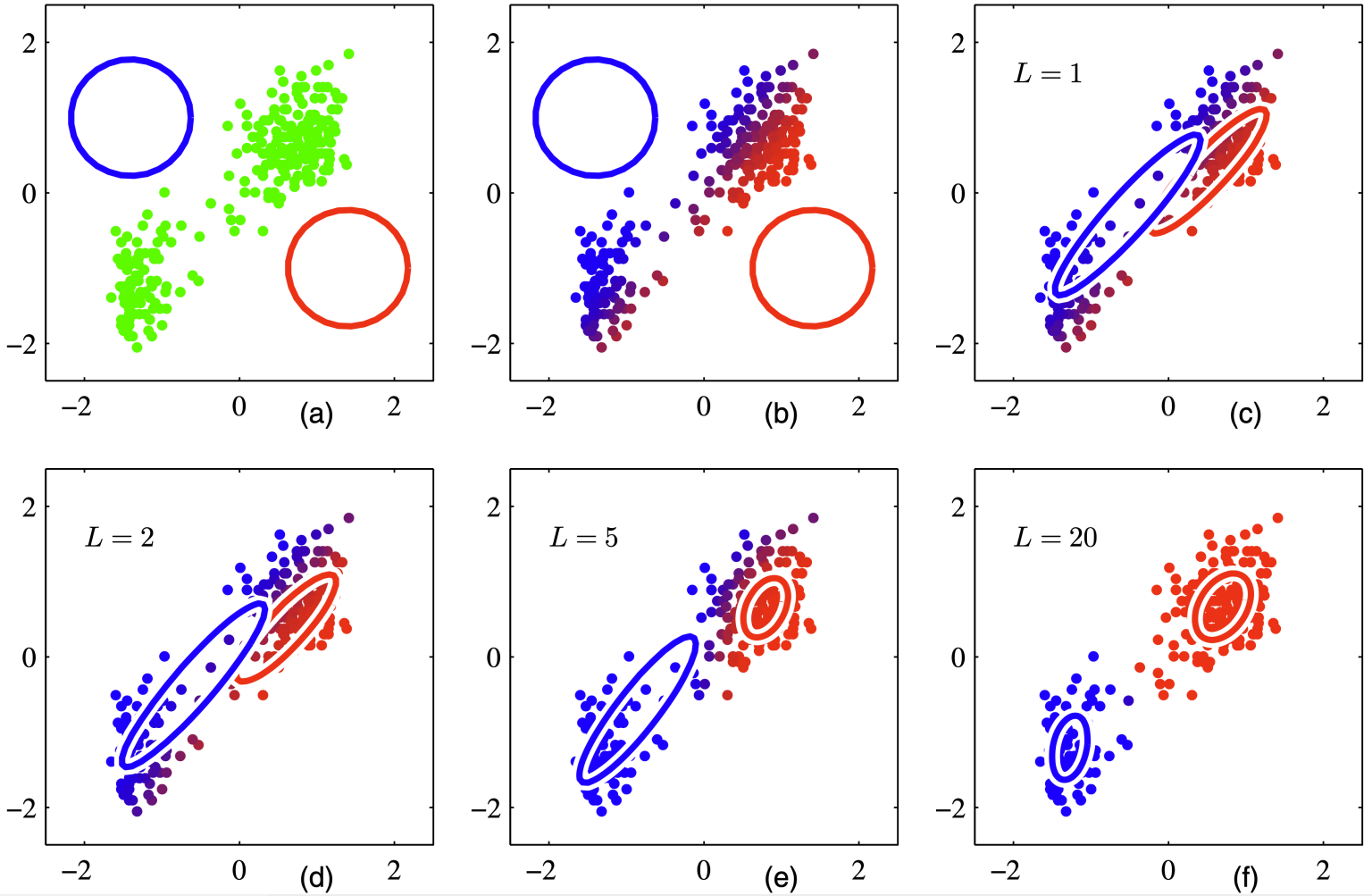
- **E-step**: Estimate $\hat{\gamma}^{(1:N)}$ with given $\hat{\theta}$

$$\hat{\gamma}_k^{(n)} = E[p(z^{(n)} = k|x^{(n)})] = \frac{\mathcal{N}(x^{(n)}; \hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{k'} \mathcal{N}(x^{(n)}; \hat{\mu}_{k'}, \hat{\Sigma}_{k'})}$$

- **M-step**: Maximize the likelihood for $\hat{\theta}$ with $\hat{\gamma}^{(1:N)}$ given

$$\hat{\mu}_k = \frac{1}{N} \sum_{n=1}^{N} \hat{\gamma}_k^{(n)} x^{(n)}$$

Example



EM Algorithm: Further Comments

- EM algorithm gives θ a point estimate $\hat{\theta}$
- Each training example $x^{(n)}$ has its own latent variable

$$z^{(n)} = (z_1^{(n)}, \ldots, z_K^{(n)})$$

- Compute the posterior distribution $p(z^{(n)}|x^{(n)})$ and its expectation

$$(\gamma_1^{(n)}, \ldots, \gamma_K^{(n)})$$

  is an important step in this algorithm

Variational EM

At the E-step, instead of computing $p(z^{(n)}|x^{(n)};\theta;\gamma^{(1:N)})$ directly, variational EM uses a variational distribution $q(z^{(n)})$ and solving the problem by minimizing the following objective

$$\mathrm{KL}[q(z^{(n)};\psi^{(n)})\|p(z^{(n)}|x^{(n)};\gamma^{(n)})]$$

- When q is rich enough to make $\mathrm{KL}=0$, then this is reduced to the traditional EM algorithm

Variational Bayes EM

- When consider θ also as random variables, we need to define a joint distribution $q(\theta, z^{(1:N)})$ instead of just $q(z^{(1:N)})$
- Follow the mean field approximation, we have

$$q(\theta; z^{(1:N)}|\phi, \psi^{(1:N)}) = q(\theta; \phi) \prod_{n=1}^{N} q(z^{(n)}; \psi^{(n)})$$

- The algorithm will alternate between ϕ and $\psi^{(1:N)}$

Amortized Inference

Amortized Variational Inference

In mean field approximation, a typical way of defining variational distribution is

$$q(z^{(1:N)}; \psi^{(1:N)}) = \prod_{n=1}^{N} q(z^{(n)}; \psi^{(n)})$$

Each $z^{(n)}$ has its own parameters that will be estimated during the inference

For example

- $z^{(n)}$ is a Gaussian random variable
- $\psi^{(n)} = \{\mu_{(n)}, \sigma^2_{(n)}\}$

Amortized Variational Inference (II)

Instead of estimating $\psi^{(n)}$ directly, we can design a function $f_\xi(\cdot)$ and compute $\psi^{(n)}$ as

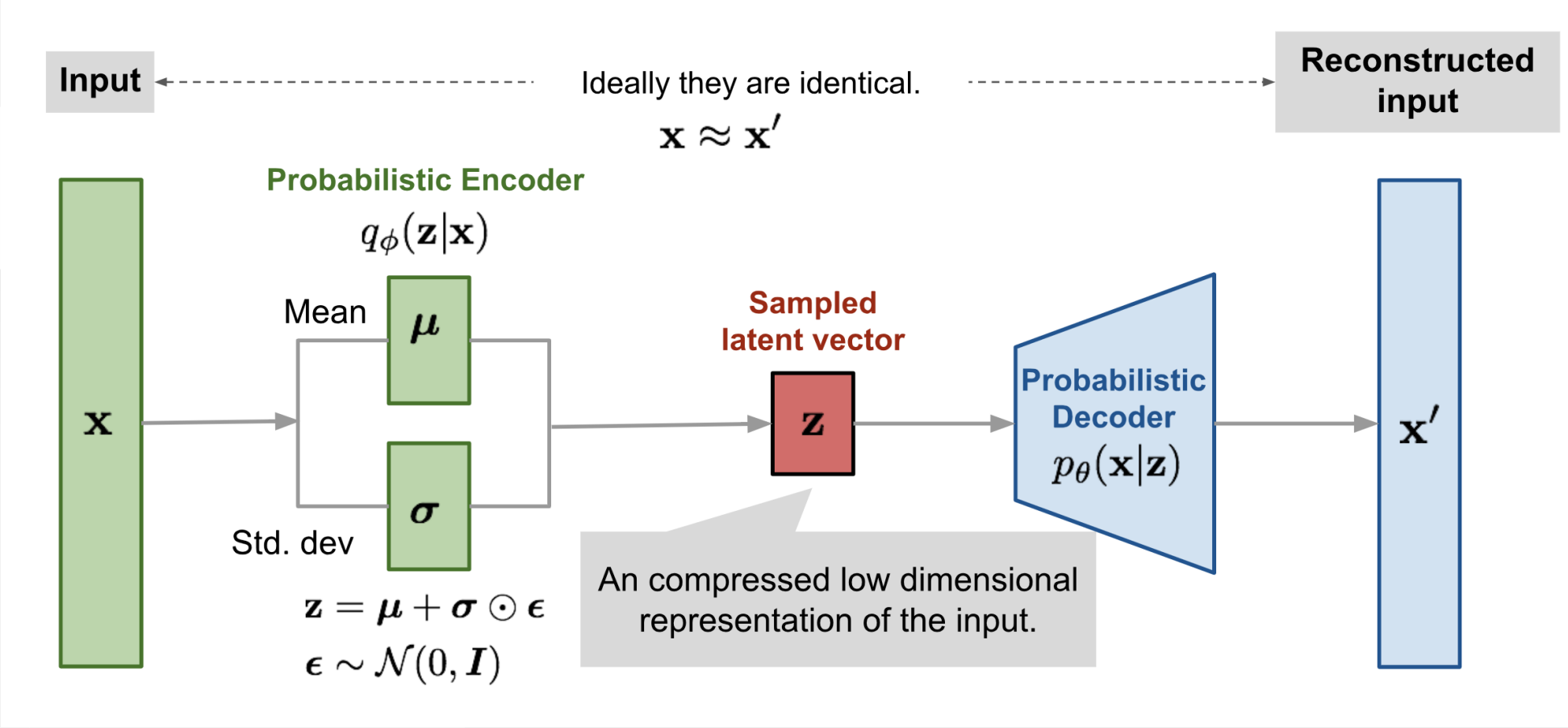
$$\psi^{(n)} = f(x^{(n)}; \xi)$$

where ξ is the parameter set for function f

For example

- $(\mu_{(n)}, \sigma_{(n)}) = f(x^{(n)}; \xi)$
- in variational auto-encoder, this function is a network and is called **inference network** or **recognition network**

Variational Auto-encoder



Amortized Variational Inference (III)

Amortized inference: reduce the cost of per-example inference on $\phi^{(n)}$ by training a model $f(x;\xi)$ that shared across all examples

$$\psi^{(n)} = f(x^{(n)};\xi)$$

- With amortized variational inference, the variational parameter set will be changed from $\{\psi^{(n)}\}$ to ξ
- With $f(\cdot;\xi)$
  - It is much easier to handle new examples, e.g., during testing phase
  - It can also reduce the number of parameters (e.g., consider 1M examples)

Issues

Consider the difference:

$$\{\psi^{(n)}\} \text{ v.s. } \{f(x^{(n)}, \xi)\}$$

- The performance of amortized inference depends on the choice of function f
- Often, $f(x^{(n)}; \xi)$ can only give sub-optimal solutions, compared with the direct estimation of $\psi^{(n)}$, which is called the **amortization gap**.

Thank You!