Advanced Deep Learning

Approximate Inference

U Kang
Seoul National University

In This Lecture

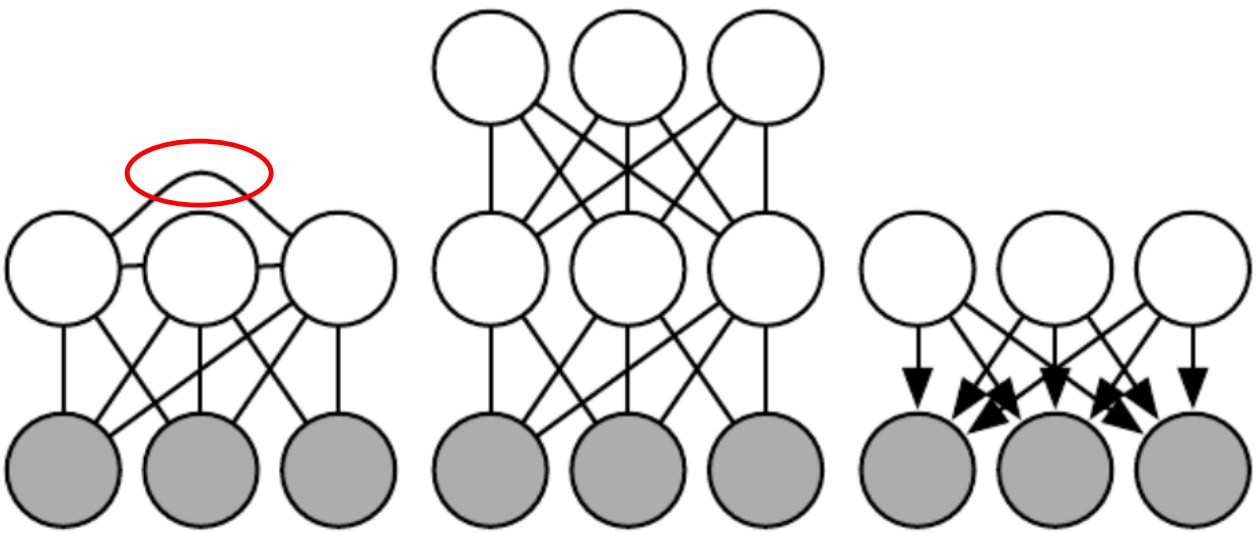
- Inference as Optimization
- Expectation Maximization
- MAP Inference and Sparse Coding
- Variational Inference and Learning

Motivation

- Intractable inference problems in deep learning are usually the result of **interactions between latent variables** in a structured graphical model.
- These interactions can be due to edges **directly connecting** one latent variable to another or longer paths that are activated when the child of a **V-structure** is observed.

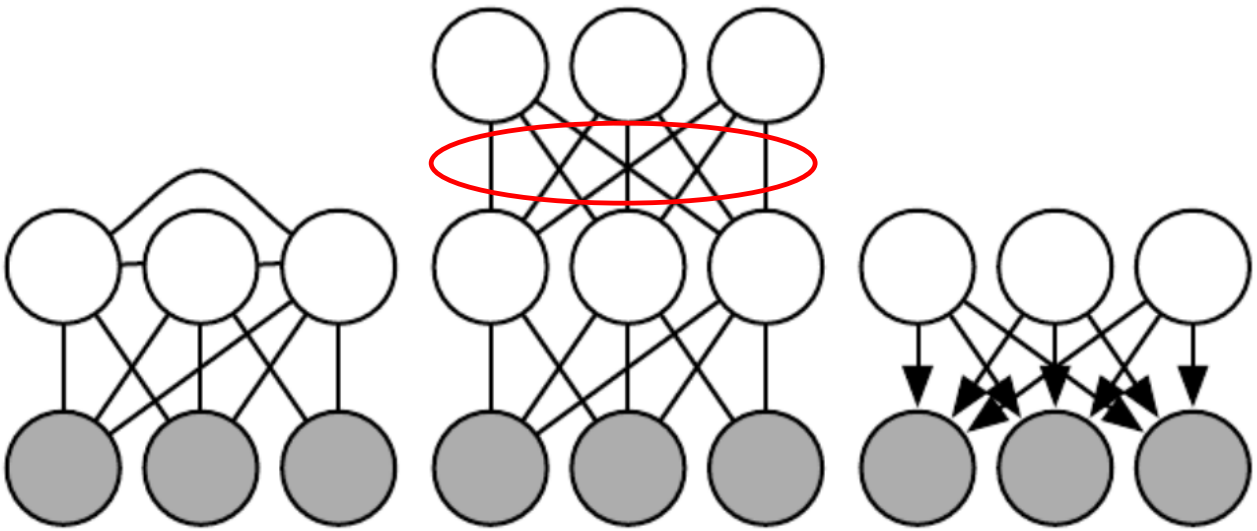
Motivation

- Left. These **direct connections** between latent variables make the posterior distribution intractable since latent variables are **dependent**.



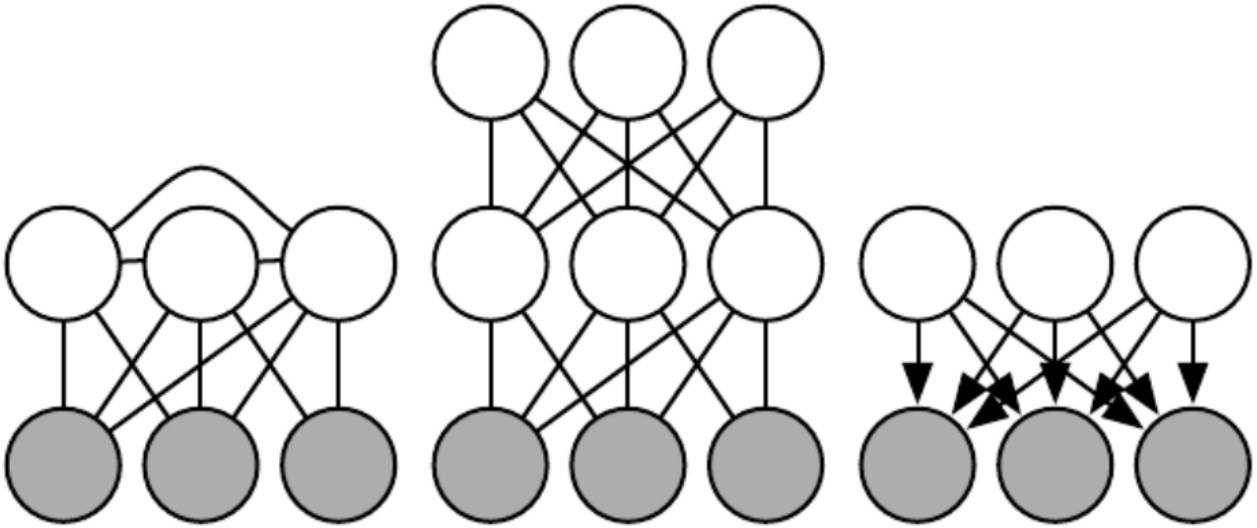
Motivation

- Center. It still has an intractable posterior distribution because of the **connections between layers**.



Motivation

- Right. This directed model has interactions between latent variables when the **visible variables** are **observed**, because every two latent variables are **coparents** (V-structure).

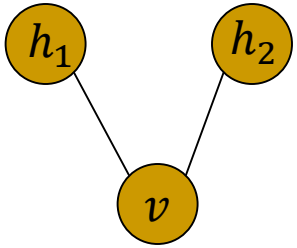


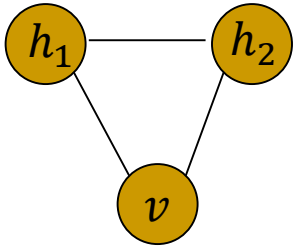
Motivation

- What do we want to do?
  - Computing $p(h|v)$
  - Taking expectations w.r.t. $p(h|v)$
- Exact inference requires an exponential amount of time in these models.
  - Computing $p(v)$ is **intractable**!
- We need some approximate inference techniques for confronting these intractable inference problems.

Example

- Consider the task of computing $p(h|v)$
- If h's are independent given v, p(v) can be efficiently computed
  - $p(v) = \sum_{h_1,h_2} p(v,h_1,h_2) = \sum_{h_1,h_2} p(v,h_1)p(v,h_2) = \sum_{h_1} p(v,h_1) \sum_{h_2} p(v,h_2)$
- Otherwise, p(v) is intractable





Outline

- **Inference as Optimization**
- Expectation Maximization
- MAP Inference
- Variational Inference and Learning

Inference as Optimization

- Exact inference can be described as an optimization problem.
- Assume: we have a probabilistic model consisting of observed variables v and latent variables h.
- Our goal: compute $p(h|v) = \frac{p(v|h)p(h)}{p(v)}$
- It is too difficult to compute $p(v;\theta)$ if it is costly to marginalize out h.

Inference as Optimization

- How to describe the inference problem as the optimization problem?
  - We compute **Evidence Lower BOund (ELBO)** instead of $p(v;\ \theta)$
  - Evidence Lower Bound (ELBO)

$$\mathcal{L}(\boldsymbol{v}, \boldsymbol{\theta}, q) = \log p(\boldsymbol{v}; \boldsymbol{\theta}) - D_{\mathrm{KL}}\left(q(\boldsymbol{h} \mid \boldsymbol{v}) \| p(\boldsymbol{h} \mid \boldsymbol{v}; \boldsymbol{\theta})\right)$$

    - L always has **at most** the same value as the desired log-probability since the KL divergence is always **nonnegative**.
    - If the KL divergence is 0, q is the same as $p(h|v)$

Inference as Optimization

- L can be considerably easier to compute for some distributions q.
  - L is **tractable** to compute if we choose appropriate q.

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{v}, \boldsymbol{\theta}, q) &= \log p(\boldsymbol{v}; \boldsymbol{\theta}) - D_{\mathrm{KL}}(q(\boldsymbol{h} \mid \boldsymbol{v}) \| p(\boldsymbol{h} \mid \boldsymbol{v}; \boldsymbol{\theta})) \\
&= \log p(\boldsymbol{v}; \boldsymbol{\theta}) - \mathbb{E}_{\mathbf{h} \sim q} \log \frac{q(\boldsymbol{h} \mid \boldsymbol{v})}{p(\boldsymbol{h} \mid \boldsymbol{v})} \\
&= \log p(\boldsymbol{v}; \boldsymbol{\theta}) - \mathbb{E}_{\mathbf{h} \sim q} \log \frac{q(\boldsymbol{h} \mid \boldsymbol{v})}{\frac{p(\boldsymbol{h}, \boldsymbol{v}; \boldsymbol{\theta})}{p(\boldsymbol{v}; \boldsymbol{\theta})}} \\
&= \log p(\boldsymbol{v}; \boldsymbol{\theta}) - \mathbb{E}_{\mathbf{h} \sim q} \left[\log q(\boldsymbol{h} \mid \boldsymbol{v}) - \log p(\boldsymbol{h}, \boldsymbol{v}; \boldsymbol{\theta}) + \log p(\boldsymbol{v}; \boldsymbol{\theta})\right] \\
&= -\mathbb{E}_{\mathbf{h} \sim q} \left[\log q(\boldsymbol{h} \mid \boldsymbol{v}) - \log p(\boldsymbol{h}, \boldsymbol{v}; \boldsymbol{\theta})\right]. \\
&= E_{h \sim q}[\log p(h, v)] + H(q)
\end{aligned}
$$

- For any choice of q, L provides a lower bound on the likelihood.

Inference as Optimization

- For $q(h|v)$ that are **better approximations** of $p(h|v)$, the lower bound L will be **tighter**.
- We can think of inference as the procedure for finding the q that **maximizes** L.
- Exact inference maximizes L perfectly by searching over a family of functions q that includes $p(h \mid v)$.

Outline

- ☑ Inference as Optimization
- ☐ **Expectation Maximization**
- ☐ MAP Inference
- ☐ Variational Inference and Learning

Expectation Maximization

- Now we will maximize a lower bound L by using expectation maximization(EM) algorithm.
- What is EM algorithm?
  - EM algorithm is an **iterative optimization technique** which is operated **locally**

Expectation Maximization

- EM algorithm finds **maximum likelihood parameter** estimates in problems where some variables were unobserved.
- The EM algorithm consists of alternating between two steps until convergence:
  - Expectation step
    - For given parameter values we can compute the expected values of the latent variable.
  - Maximization step
    - Updates the parameters of our model based on the latent variable calculated using ML method.

Expectation Maximization

- EM can be viewed as a coordinate ascent algorithm to maximize L

$$\mathcal{L}(\boldsymbol{v}, \boldsymbol{\theta}, q) = \log p(\boldsymbol{v}; \boldsymbol{\theta}) - D_{\mathrm{KL}}\left(q(\boldsymbol{h} \mid \boldsymbol{v}) \| p(\boldsymbol{h} \mid \boldsymbol{v}; \boldsymbol{\theta})\right)$$

- E-step: maximize L wrt. q
- M-step: maximize L wrt. θ

Expectation Maximization

- E-step: maximize L wrt. q
  - Set $q^{(t)}\left(h^{(i)}|v\right) = p(h^{(i)}|v^{(i)};\theta^{(t-1)})$ for all indices i of the training examples $v^{(i)}$ we want to train on.

$$\mathcal{L}(\boldsymbol{v}, \boldsymbol{\theta}, q) = \log p(\boldsymbol{v}; \boldsymbol{\theta}) - D_{\mathrm{KL}}\left(q(\boldsymbol{h} \mid \boldsymbol{v}) \| p(\boldsymbol{h} \mid \boldsymbol{v}; \boldsymbol{\theta})\right)$$

- M-step: maximize L wrt. θ
  - Completely or partially maximize $\sum_i \mathcal{L}(\boldsymbol{v}^{(i)}, \boldsymbol{\theta}, q)$

with respect to θ using your optimization algorithm of choice.

$$L(v, \theta, q) = E_{h\sim q}[\log p(h, v)] + H(q)$$

Another Viewpoint of EM

- Iterate the following E-step and M-step
- E-step: evaluate $p(h|v;\ \theta^{(t-1)})$
- M-step: evaluate $\theta^{(t)} = argmax_{\theta} Q\left(\theta, \theta^{(t-1)}\right)$
  - where $Q\left(\theta, \theta^{(t-1)}\right) = E_{h \sim p(h|v;\ \theta^{(t-1)})}[\log p(h, v; \theta)]$

Example of EM: Gaussian Mixture

- Consider mixtures of Gaussian model

$$\mathbf{p(x)} = \sum_{\mathbf{k=1}}^{\mathbf{K}} \boldsymbol{\pi}_{\mathbf{k}} \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{\mathbf{k}}, \Sigma_{\mathbf{k}})$$

Number of Gaussians

Mixing coefficient: weight for each Gaussian dist.

- $0 \leq \pi_k \leq 1, \sum_k \pi_k = 1$

Gaussian Mixture

- log likelihood

$$\ln p(X \mid \mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln p(x_n) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k) \right\}$$

- MLE does not work here as there is no closed form solution
- Parameters can be calculated using EM algorithm.

Gaussian Mixture

- We can think of the mixing coefficients as prior probabilities for the components.
- For a given value of ‘x’, we can evaluate the corresponding posterior probabilities, called responsibilities.
- From Bayes rule

$$\gamma_k(x) = p(k \mid x) = \frac{p(k)p(x \mid k)}{p(x)}$$

Latent Variable

$$= \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x \mid \mu_j, \Sigma_j)} \quad \text{where, } \pi_k = \frac{N_k}{N}$$

Gaussian Mixture

- Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters comprising the means and covariances of the components and the mixing coefficients.

Gaussian Mixture

1. Initialize the means μ, covariances $\sum$ and mixing coefficients π, and evaluate the initial value of the log likelihood.
2. E step. Evaluate the responsibilities using the current parameter values.

$$\gamma_k(\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x} \mid \mu_j, \Sigma_j)}$$

Gaussian Mixture

3. M step. Re-estimate the parameters using the current responsibilities.

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)(x_n - \mu_j)(x_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

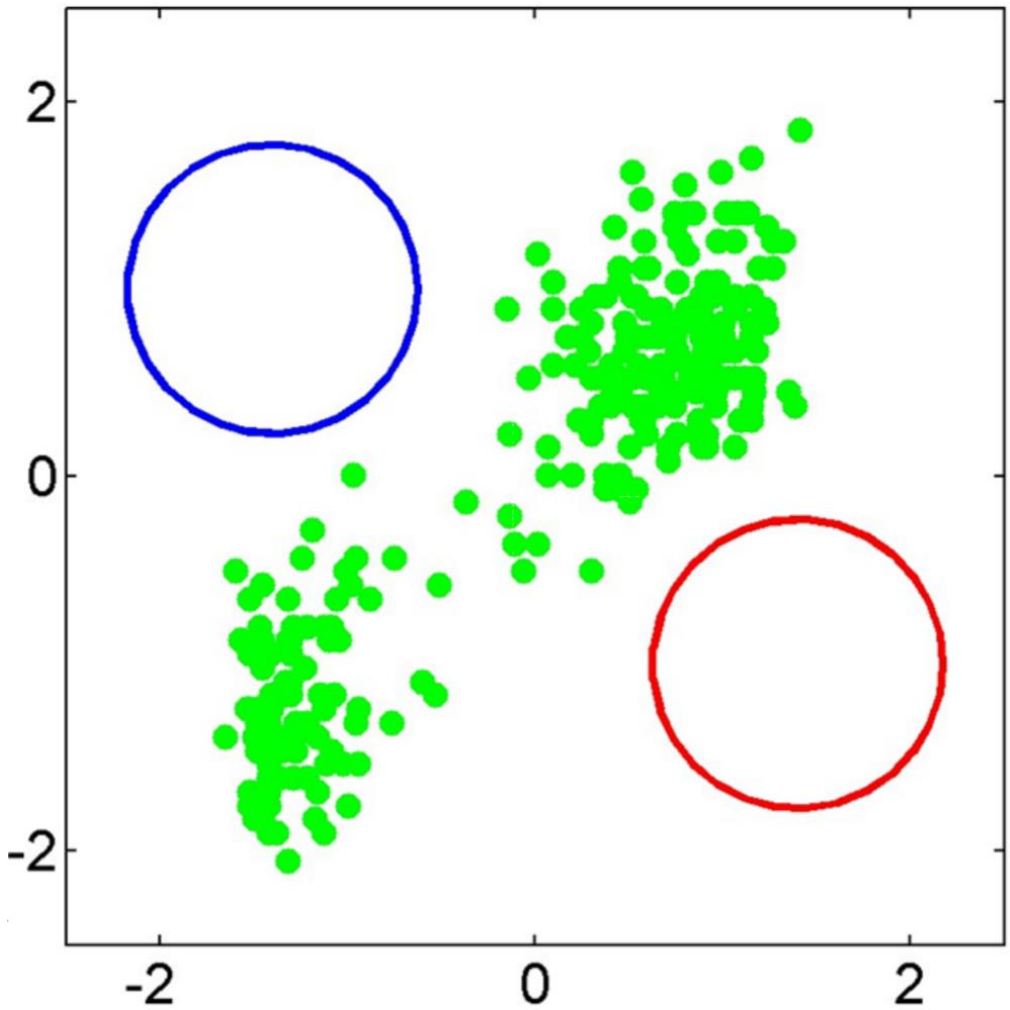
$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(x_n)$$

4. Evaluate log likelihood

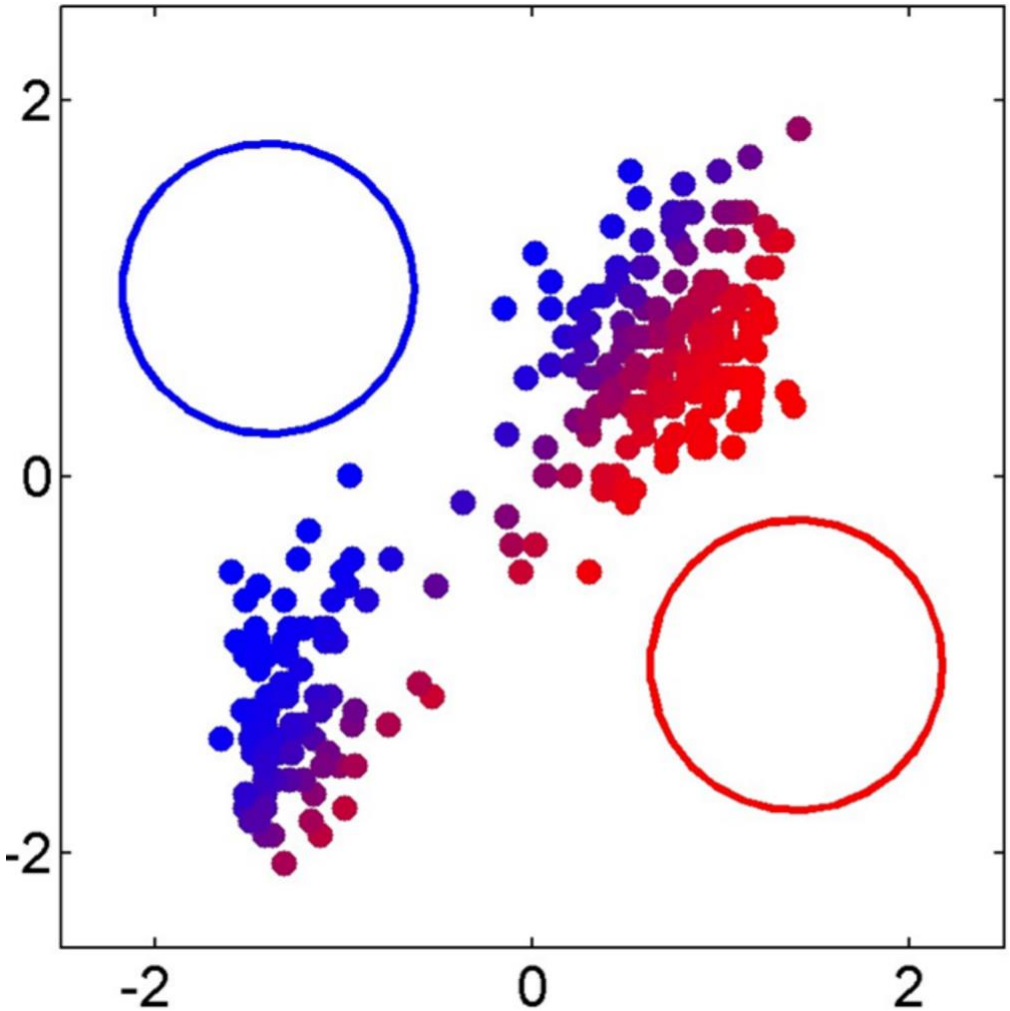
$$\ln p(X \mid \mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k) \right\}$$

If there is no convergence, return to step 2.

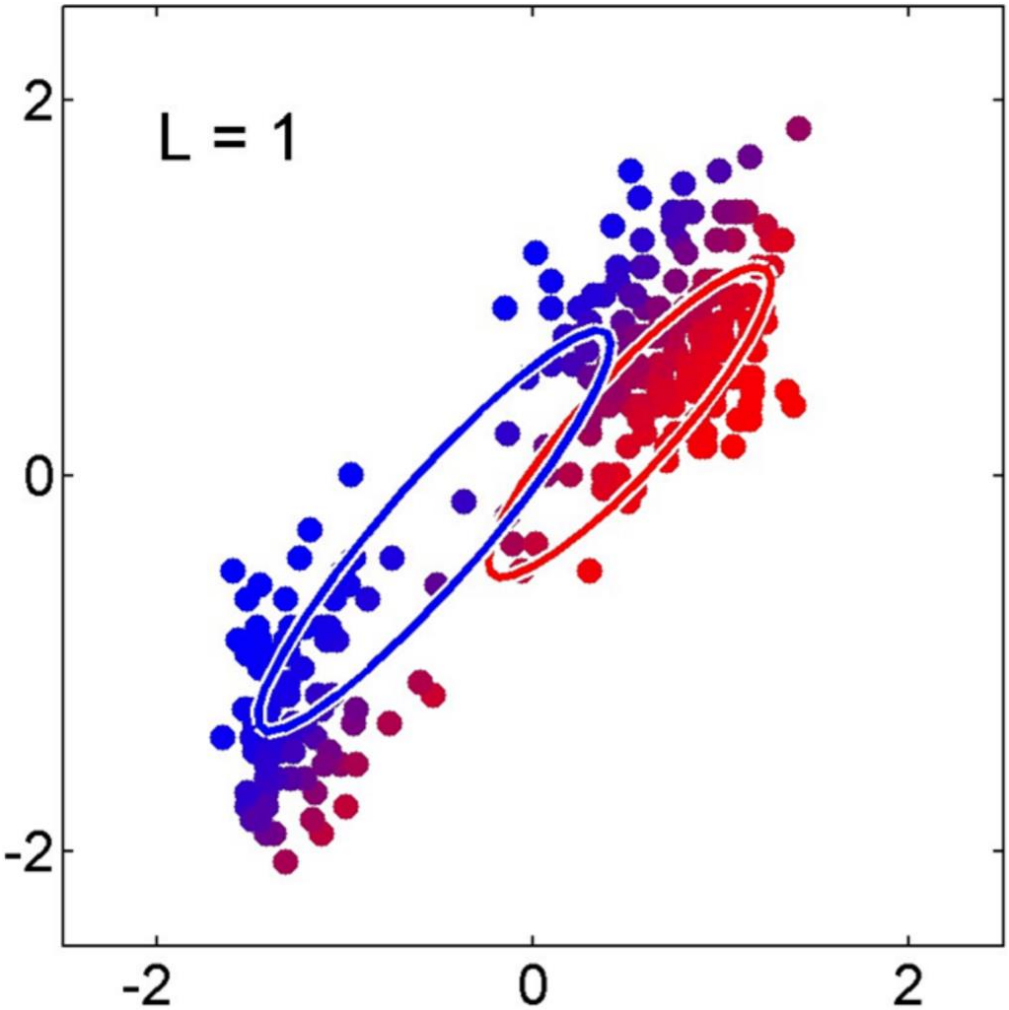
Gaussian Mixture



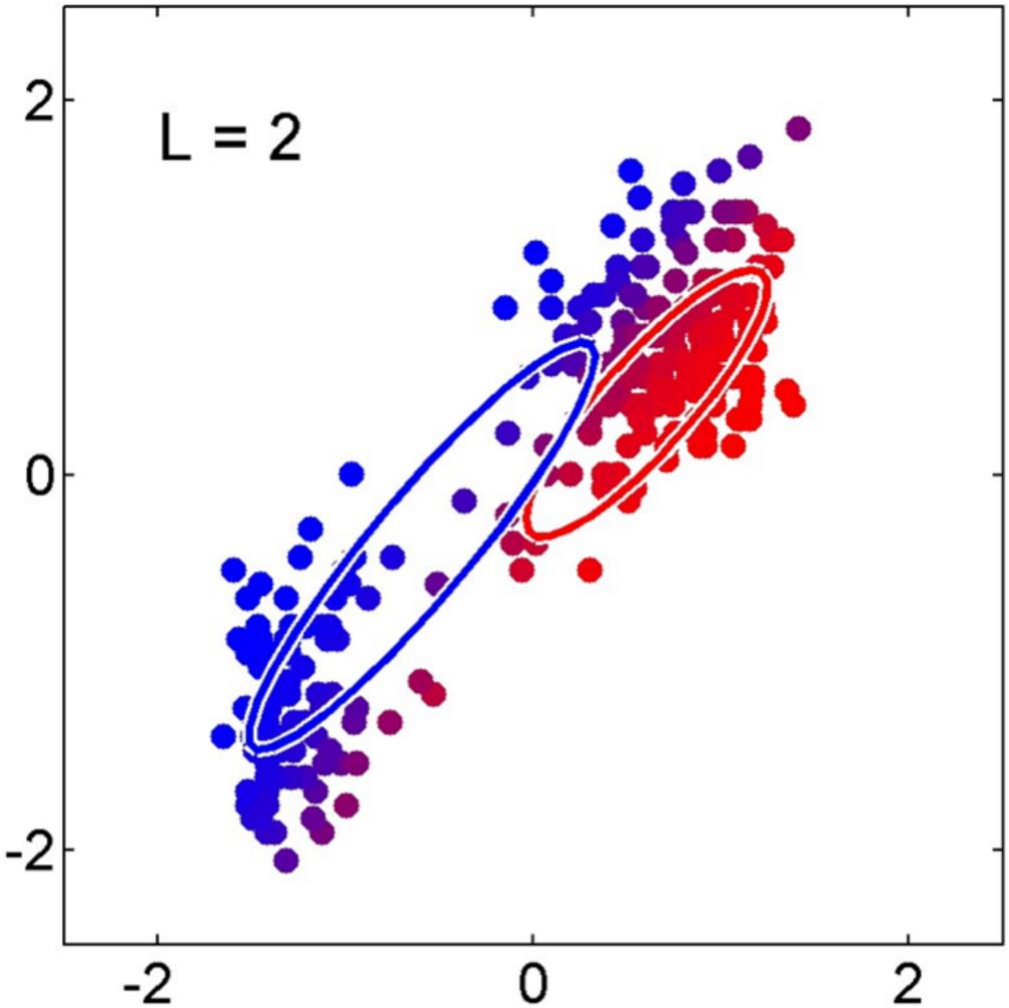
Gaussian Mixture



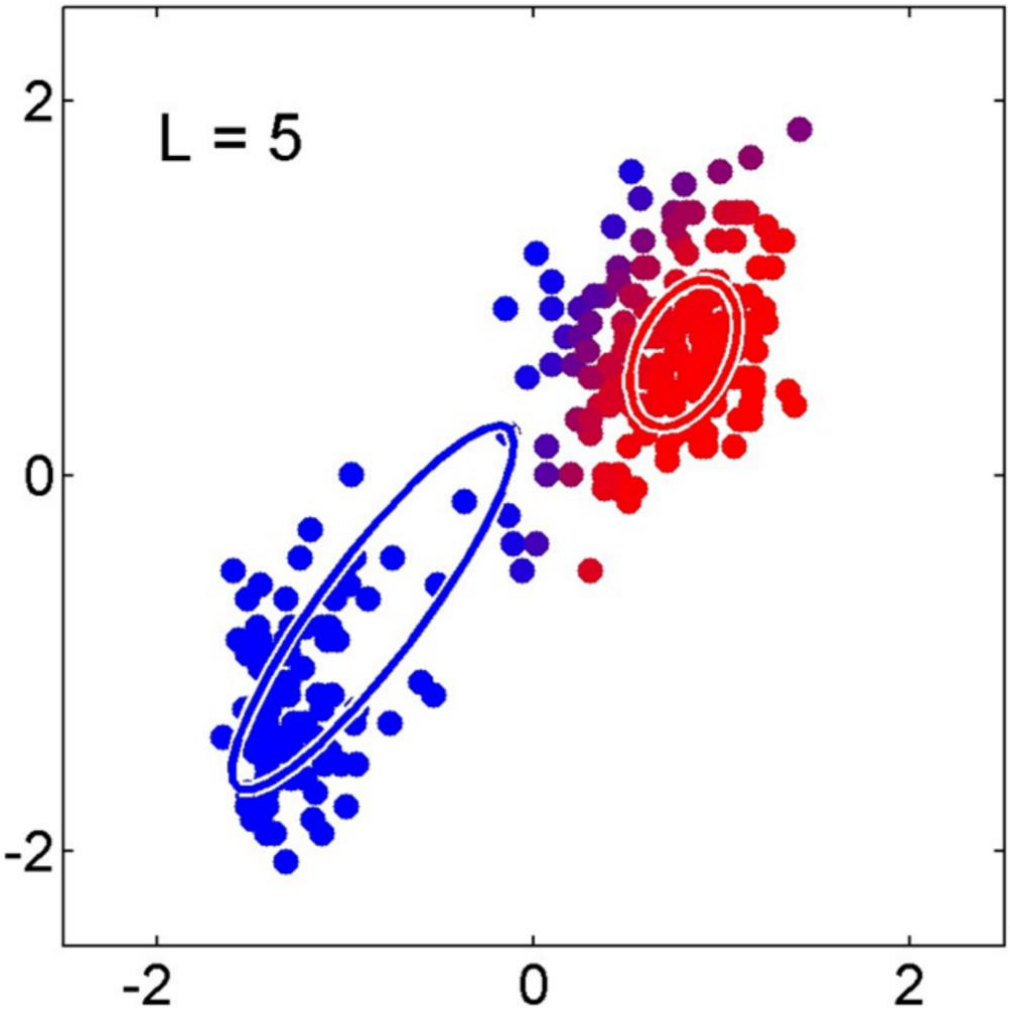
Gaussian Mixture



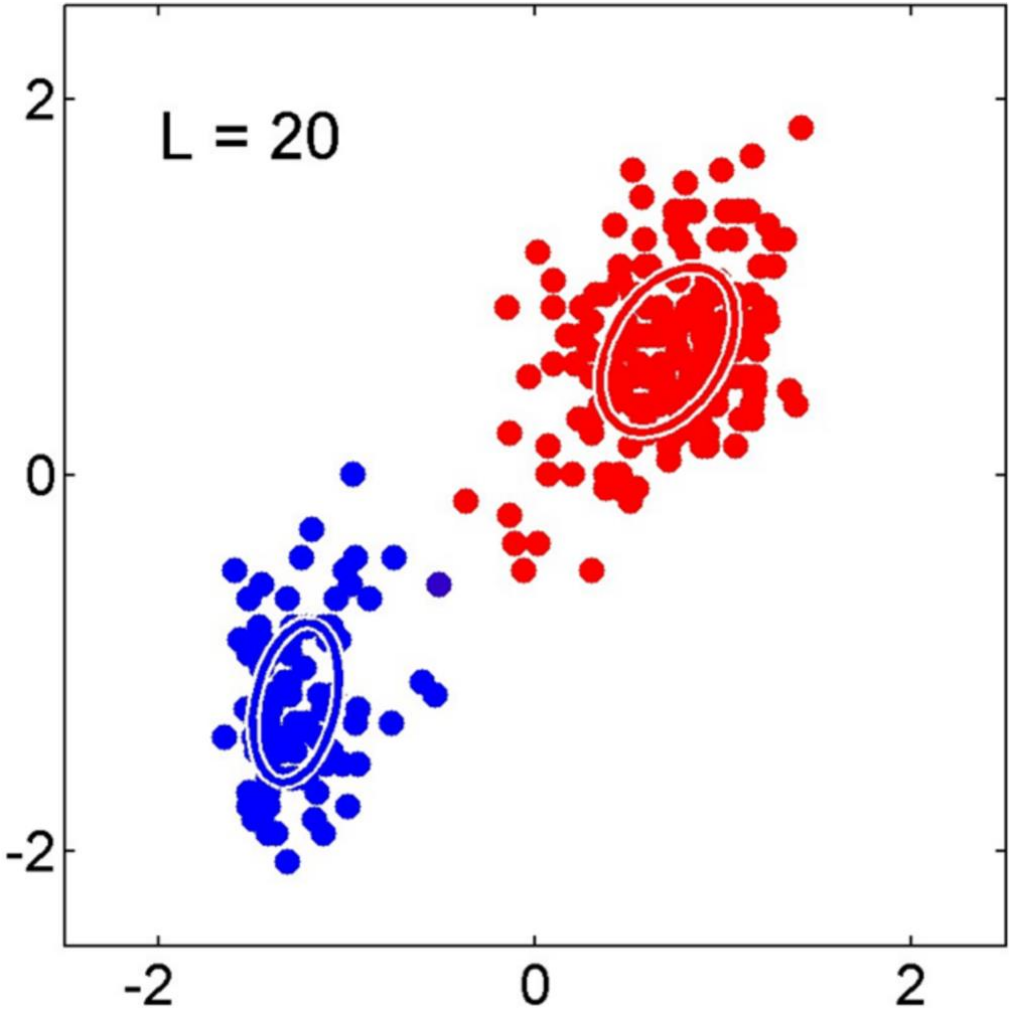
Gaussian Mixture



Gaussian Mixture



Gaussian Mixture



Questions?