Georgia Tech College of Computing

CS4641 Spring 2025

Unsupervised Learning: Gaussian Mixture Models

Bo Dai
School of CSE, Georgia Tech
bodai@cc.gatech.edu

ML Algorithm Pipeline

General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

Supervised Learning vs. Unsupervised Learning

Training Data

Supervised $\{x^i, y^i\}_{i=1}^m$

Learning Algorithm

Target Function

$f : X \rightarrow Y$

Predictor/Classifier

Supervised Learning vs. Unsupervised Learning

Training Data

Supervised $\{x^i, y^i\}_{i=1}^m$

Unsupervised $\{x_i\}_{i=1}^m$

Learning Algorithm

Target Function

$f : X \to Y$

Density Estimation $p(x)$

Clustering

$$f : X \to Y$$

$$Y \in \{0, 1, \ldots, k\}$$

Dimension Reduction

$$f : X \to Y$$

$$X \in \mathbb{R}^p, \quad Y \in \mathbb{R}^d$$

Supervised Learning vs. Unsupervised Learning

Training Data

Supervised $\{x^i, y^i\}_{i=1}^m$

Unsupervised $\{x_i\}_{i=1}^m$

Learning Algorithm

Target Function

$$f : X \to Y$$

Density Estimation $p(x)$

Clustering

$$f : X \to Y$$

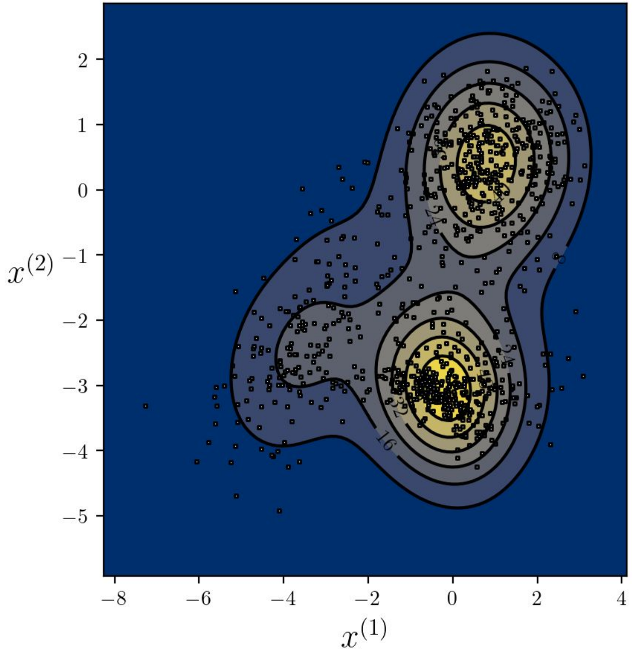
$$Y \in \{0, 1, \ldots, k\}$$

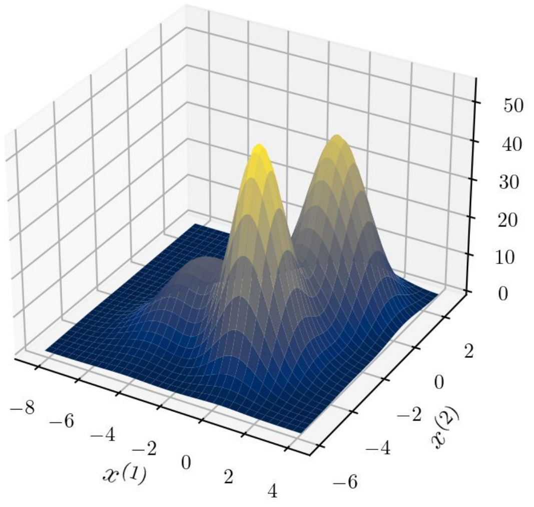
Dimension Reduction

$$f : X \to Y$$

$$X \in \mathbb{R}^p, \quad Y \in \mathbb{R}^d$$

Density Estimation

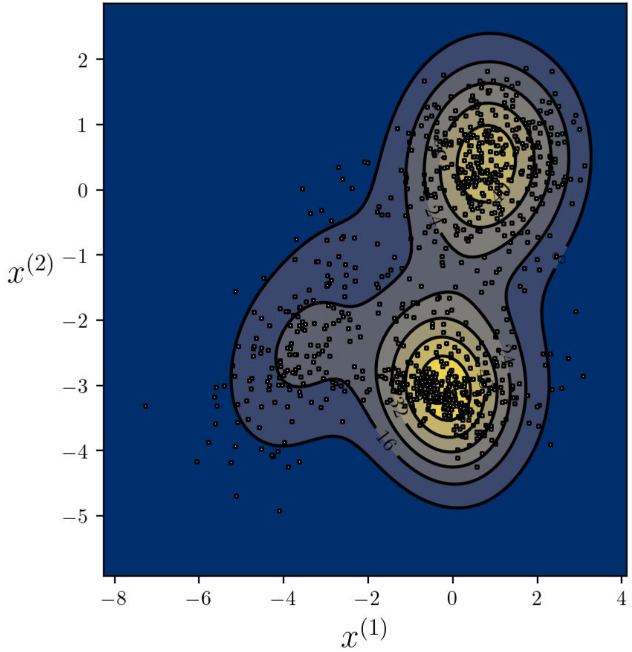




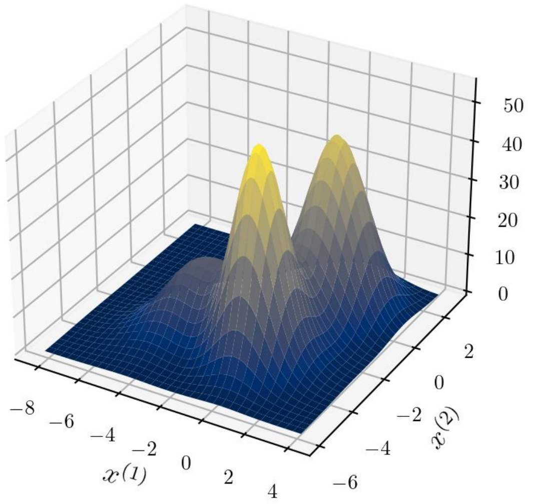
$\{x_i\}_{i=1}^{m}$

$p(x)$

Density Estimation



$\{x_i\}_{i=1}^m$



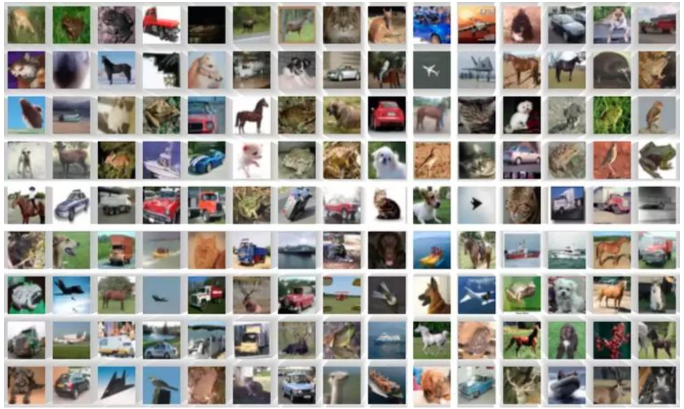
$p(x)$

Generative Models

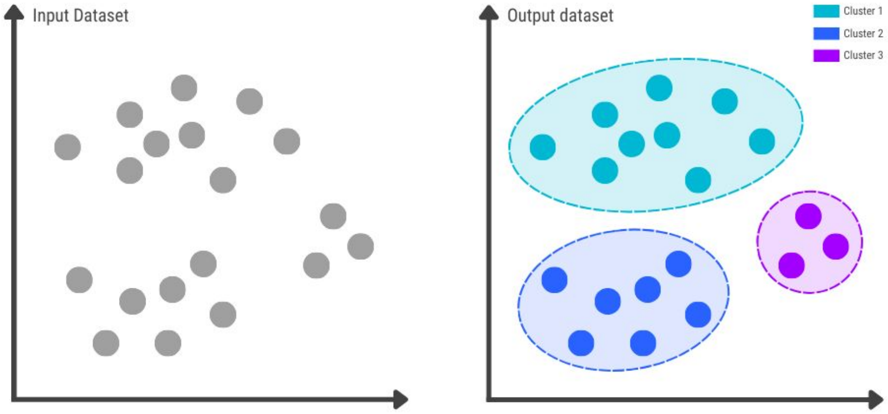
$x \sim p(x)$

Density Estimation: Generative Models

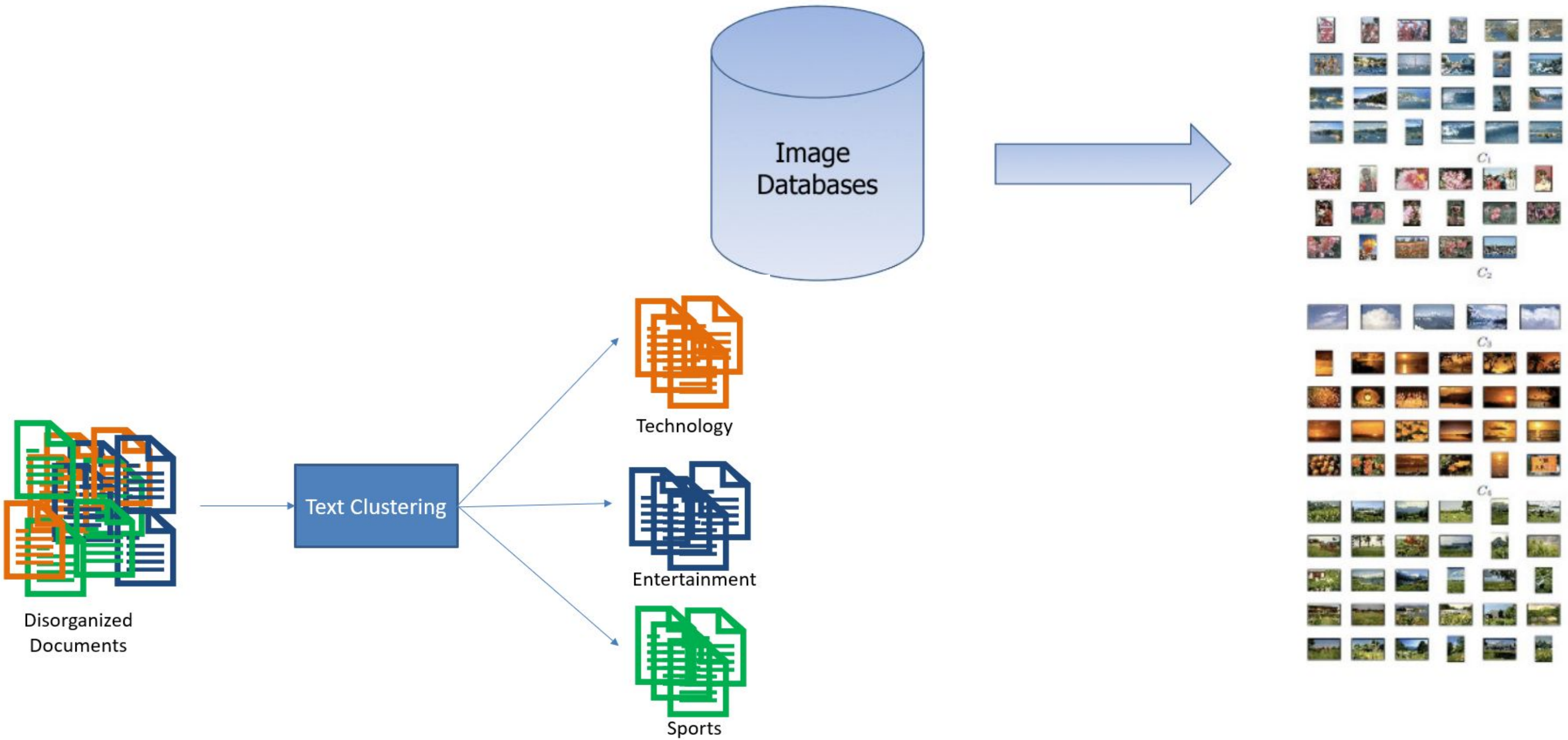
$$x \sim p(x)$$



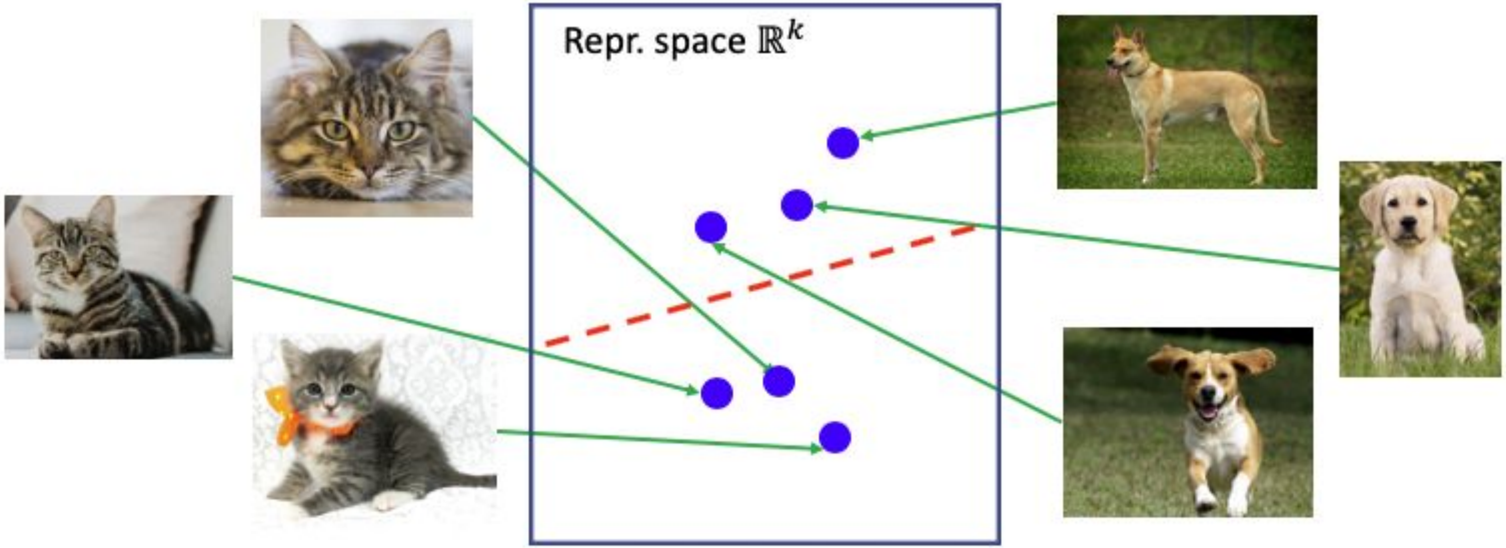
Clustering



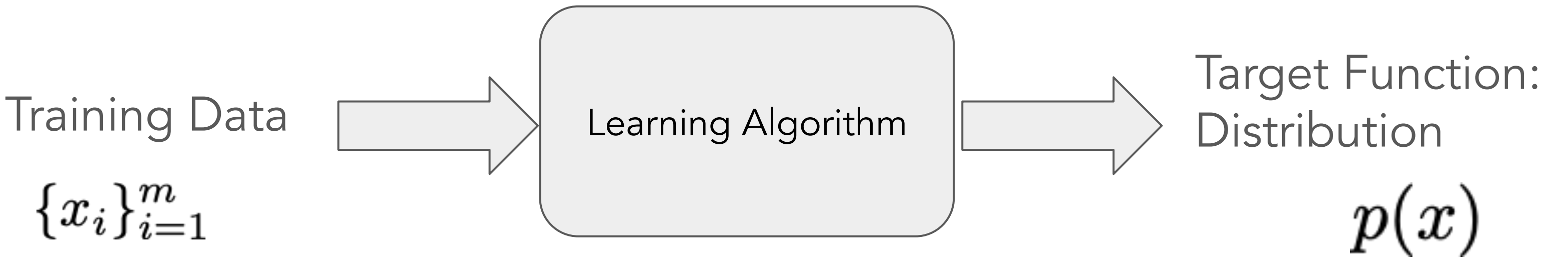
Clustering: Data Organization



Dimension Reduction/Representation Learning



Density Estimation

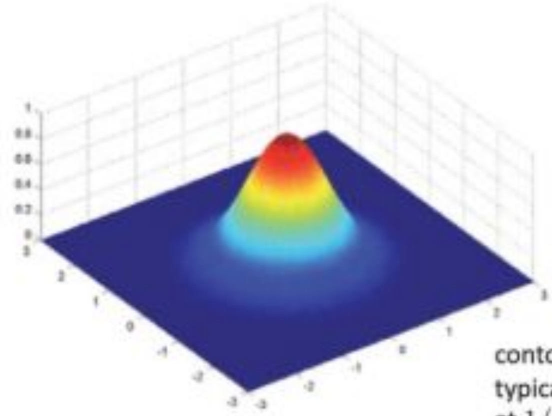


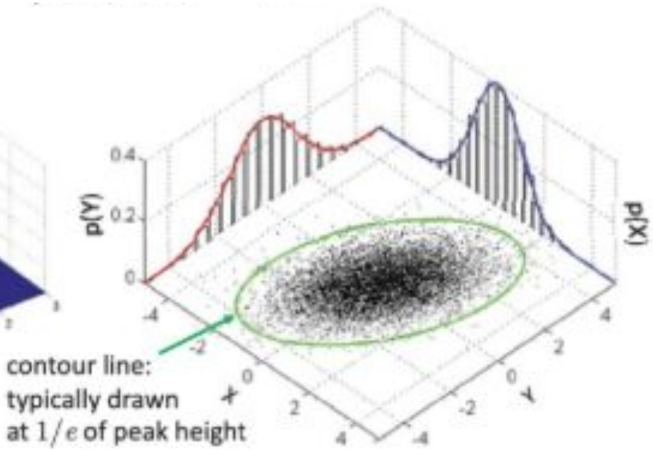
Density Estimation Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

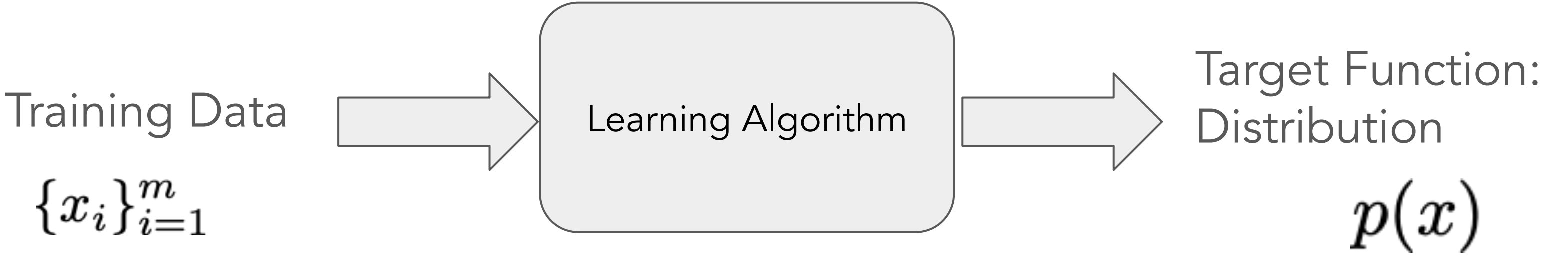
Gaussian Distribution

$$X_i \sim \mathcal{N}(\mu, \Sigma) \quad p(x_i; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2}(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)\right)$$





Density Estimation



Density Estimation Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP....)
3. Select optimizer

MLE of Gaussian Model

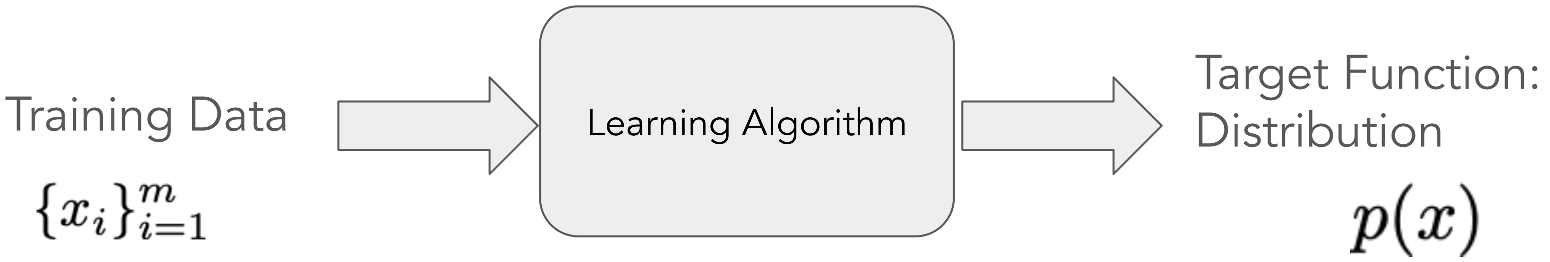
The log-likelihood is given by

$$\log L(\mu, \Sigma) = \sum_{i=1}^{m} \log p(x_i|\mu, \Sigma) = -\frac{mn}{2}\log(2\pi) - \frac{m}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{m}(x_i - \mu)^T\Sigma^{-1}(x_i - \mu)$$

The maximum likelihood estimator is

$$(\hat{\mu}, \hat{\Sigma}) = \arg\max_{\mu,\Sigma} \log p(\mathbf{X} \mid \mu, \Sigma)$$

Density Estimation



Density Estimation Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP....)
3. Select optimizer

Gradient Calculation of MLE

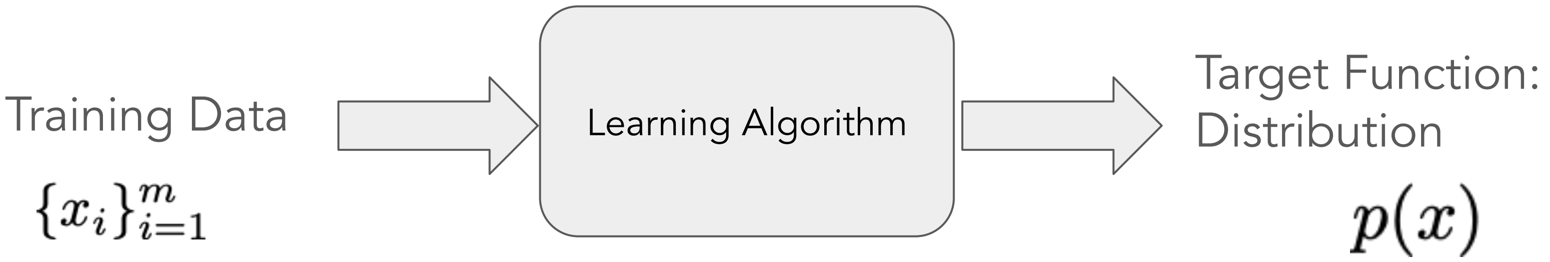
$$\nabla_\mu \log p(\mathbf{X} \mid \mu, \Sigma) = \Sigma^{-1} \sum_{i=1}^{m} (\mathbf{x}_i - \mu)$$

$$\nabla_\Sigma \log p(\mathbf{X} \mid \mu, \Sigma) = -\frac{m}{2} \Sigma^{-1} + \frac{1}{2} \sum_{i=1}^{m} \Sigma^{-1} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T \Sigma^{-1}$$

The closed-form solution is

$$\hat{\mu}_{\mathrm{MLE}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \qquad \hat{\Sigma}_{\mathrm{MLE}} = \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{x}_i - \hat{\mu}_{\mathrm{MLE}}\right)\left(\mathbf{x}_i - \hat{\mu}_{\mathrm{MLE}}\right)^T$$

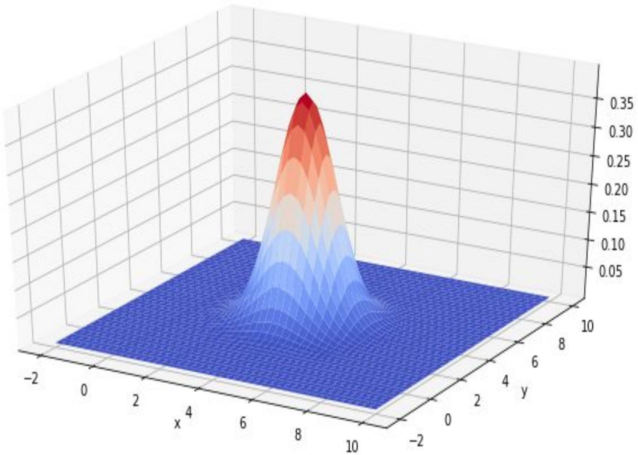
Density Estimation: Gaussian Model



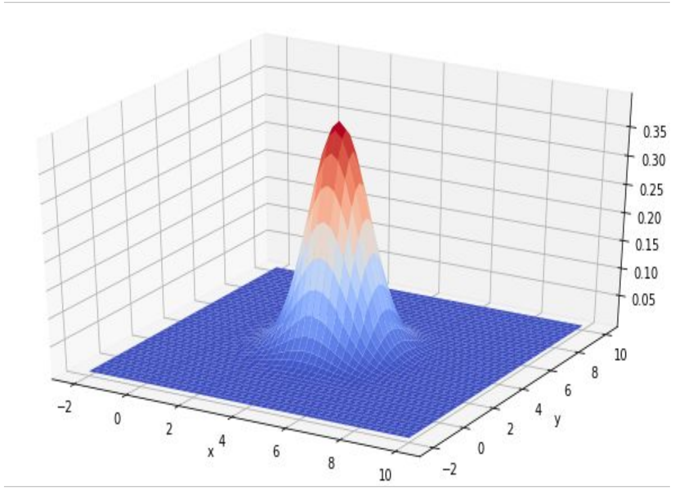
Density Estimation Pipeline

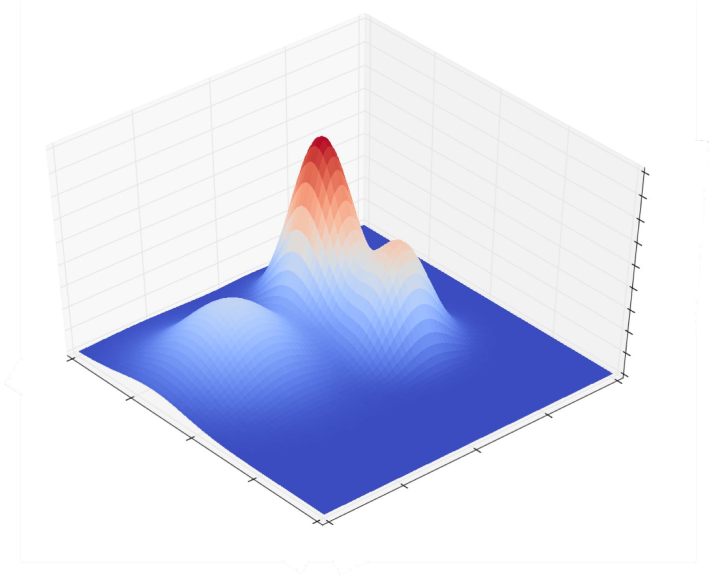
1. Build probabilistic models
   Gaussian Distribution
2. Derive loss function (by MLE or MAP….)
   MLE
3. Select optimizer
   Necessary Condition

Mixture of Gaussians

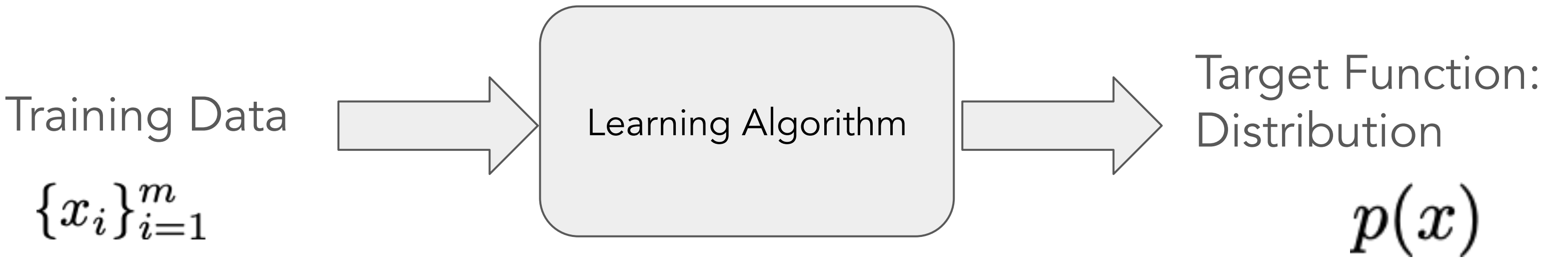


Mixture of Gaussians





Density Estimation



Density Estimation Pipeline

1. Build probabilistic models
   Gaussian Mixture Models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

Gaussian Mixture Model

Class mixture prior: $P(y)$ $\quad \pi = (\pi_1, \pi_2, \ldots, \pi_k), \quad \sum_{i=1}^{k} \pi_i = 1, \pi_i \geq 0$

Class conditional distribution: $p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$

Marginal distribution: $P(x) = \sum_{y} P(x|y)P(y) = \sum_{i=1}^{k} \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$

Connection to Gaussian Naive Bayes

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x,y)}{\sum_z P(x,y)}$$

Prior: $P(y) \quad \pi = (\pi_1, \pi_2, \ldots, \pi_k), \quad \sum_{i=1}^{k} \pi_i = 1, \pi_i \geq 0$

Likelihood (class conditional distribution : $p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$

Posterior: $P(y|x) = \dfrac{P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}{\sum_y P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}$

Connection to Gaussian Naive Bayes

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x,y)}{\sum_z P(x,y)}$$

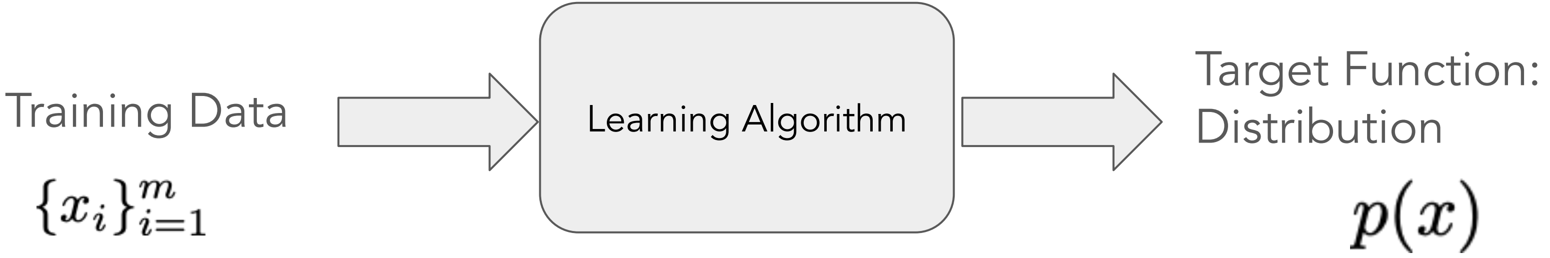
Prior: $P(y) \quad \pi = (\pi_1, \pi_2, \dots, \pi_k), \quad \sum_{i=1}^{k} \pi_i = 1, \pi_i \geq 0$

Likelihood (class conditional distribution : $p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$

Posterior: $P(y|x) = \dfrac{P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}{\sum_y P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}$

What is the difference?

Density Estimation



Density Estimation Pipeline

1. Build probabilistic models
   Gaussian Mixture Models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

MLE of GMM

$$\max_{\{\pi_i,\mu_i,\Sigma_i\}_{i=1}^k} \sum_{j=1}^m \log p(x_j) = \sum_{j=1}^m \log \Big(\sum_{i=1}^k \pi_i \mathcal{N}(x_j|\mu_i,\Sigma_i) \Big)$$

Want $\arg\max_\theta \log L(\theta)$ subject to $\sum_{j=1}^k \pi_j = 1$

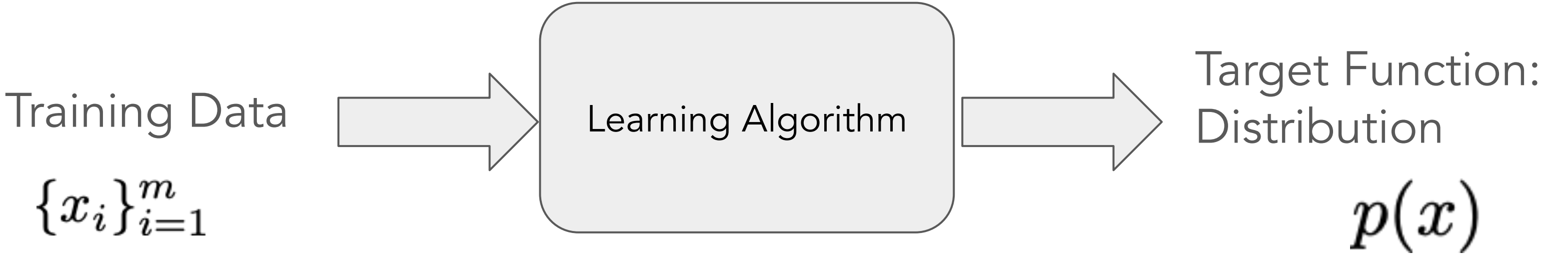
MLE of GMM vs. MLE of Gaussian Naive Bayes

$$\max_{\{\pi_i,\mu_i,\Sigma_i\}_{i=1}^k} \sum_{j=1}^m \log p(x_j) = \sum_{j=1}^m \log\Big(\sum_{i=1}^k \pi_i \mathcal{N}(x_j|\mu_i,\Sigma_i)\Big)$$

$$\max_{\{\pi_i,\mu_i,\Sigma_i\}_{i=1}^k} \sum_{j=1}^m \log p(x_j, y_j) = \sum_{j=1}^m \log \pi_{y_j} + \log \mathcal{N}(x_j|\mu_{y_j},\Sigma_{y_j}))$$

Want $\arg\max_\theta \log L(\theta)$ subject to $\sum_{j=1}^k \pi_j = 1$

Density Estimation



Density Estimation Pipeline

1. Build probabilistic models
   Gaussian Mixture Models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

Select Optimizer

Stochastic Gradient?

$$\max_{\{\pi_i,\mu_i,\Sigma_i\}_{i=1}^k} \sum_{j=1}^m \log p(x_j) = \sum_{j=1}^m \log\Big(\sum_{i=1}^k \pi_i \mathcal{N}(x_j|\mu_i,\Sigma_i)\Big)$$

Plausible but tedious

If label is given….Gaussian Naive Bayes!

$$\max_{\{\pi_i,\mu_i,\Sigma_i\}_{i=1}^k} \sum_{j=1}^m \log p(x_j) = \sum_{j=1}^m \log \Big(\sum_{i=1}^k \pi_i \mathcal{N}(x_j|\mu_i,\Sigma_i)\Big)$$

Easy to solve

$$\max_{\{\pi_i,\mu_i,\Sigma_i\}_{i=1}^k} \sum_{j=1}^m \log p(x_j, y_j) = \sum_{j=1}^m \log \pi_{y_j} + \log \mathcal{N}(x_j|\mu_{y_j},\Sigma_{y_j}))$$

$$\arg\max_\theta \log L(\theta) \qquad \sum_{j=1}^k \pi_j = 1$$

If label is given….Gaussian Naive Bayes!

$$\log L(\theta) = \sum_{i=1}^{m}\sum_{j=1}^{k} y_j^i \log \pi_j - \sum_{i=1}^{m} \log Z - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{k} y_j^i (x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j)$$

Easy to solve

$$\max_{\{\pi_i,\mu_i,\Sigma_i\}_{i=1}^k} \sum_{j=1}^{m} \log p(x_j, y_j) = \sum_{j=1}^{m} \log \pi_{y_j} + \log \mathcal{N}(x_j|\mu_{y_j}, \Sigma_{y_j}))$$

Want $\arg\max_{\theta} \log L(\theta)$ subject to $\sum_{j=1}^{k} \pi_j = 1$

If label is given….Gaussian Naive Bayes!

$$\log L(\theta) = \sum_{i=1}^{m}\sum_{j=1}^{k} y_j^i \log \pi_j - \sum_{i=1}^{m} \log Z - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{k} y_j^i (x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j)$$

$$\frac{\partial \log L}{\partial \mu_k} = -\sum_{i=1}^{m} y_k^i \Sigma_k^{-1}\left(x^i - \mu_k\right) = 0$$

$$\boxed{\mu_k = \frac{\sum_{i=1}^{m} y_k^i x^i}{\sum_{i=1}^{m} y_k^i}}$$

If label is given….Gaussian Naive Bayes!

$$\log L(\theta) = \sum_{i=1}^{m} \sum_{j=1}^{k} y_j^i \log \pi_j - \sum_{i=1}^{m} \log Z - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{k} y_j^i (x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j)$$

$$\frac{\partial \log L}{\partial \mu_k} = -\sum_{i=1}^{m} y_k^i \Sigma_k^{-1} (x^i - \mu_k) = 0$$

$$\boxed{\mu_k = \frac{\sum_{i=1}^{m} y_k^i x^i}{\sum_{i=1}^{m} y_k^i}}$$

$$\frac{\partial \log L}{\partial \Sigma_k^{-1}} = -\sum_{i=1}^{m} y_k^i \left[\frac{\partial \log Z_k}{\partial \Sigma_k^{-1}} - \frac{1}{2} (x^i - \mu_k)(x^i - \mu_k)^\top \right] = 0$$

$$\boxed{\pi_k = \frac{\sum_{i=1}^{m} y_k^i}{m}}$$

$$\boxed{\Sigma_k = \frac{\sum_{i=1}^{m} y_k^i (x^i - \mu_k)(x^i - \mu_k)^\top}{\sum_{i=1}^{m} y_k^i}}$$

How to get label? Guess by model

Gaussian Naive Bayes Prediction:

Compute the conditional probability by Bayes rule!

$$P(y|x) = \frac{P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}{\sum_y P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}$$

$$y_j^l = \frac{\pi_l \mathcal{N}(x_j|\mu_l, \Sigma_l)}{\sum_{l=1}^k \pi_l \mathcal{N}(x_j|\mu_l, \Sigma_l)}$$

$$l = 1, \ldots, k$$

$$j = 1, \ldots, m$$

Expectation-Maximization

For t = 1……

- E-Step: Guess sample labels based on current model

$$y_j^l = \frac{\pi_l \mathcal{N}(x_j|\mu_l, \Sigma_l)}{\sum_{l=1}^{k} \pi_l \mathcal{N}(x_j|\mu_l, \Sigma_l)}$$

- M-Step: Update the parameters with current labels

$$\mu_k = \frac{\sum_{i=1}^{m} y_k^i x^i}{\sum_{i=1}^{m} y_k^i} \qquad \pi_k = \frac{\sum_{i=1}^{m} y_k^i}{m} \qquad \Sigma_k = \frac{\sum_{i=1}^{m} y_k^i \left(x^i - \mu_k\right)\left(x^i - \mu_k\right)^\top}{\sum_{i=1}^{m} y_k^i}$$

Expectation-Maximization

For t = 1……

- E-Step: Guess sample labels based on current model

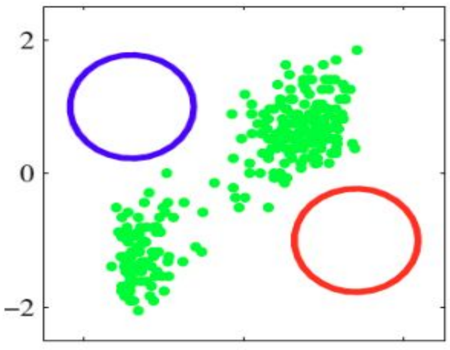
$$y_j^l = \frac{\pi_l \mathcal{N}(x_j|\mu_l, \Sigma_l)}{\sum_{l=1}^k \pi_l \mathcal{N}(x_j|\mu_l, \Sigma_l)}$$

- M-Step: Update the parameters with current labels

$$\mu_k = \frac{\sum_{i=1}^m y_k^i x^i}{\sum_{i=1}^m y_k^i} \quad \pi_k = \frac{\sum_{i=1}^m y_k^i}{m} \quad \Sigma_k = \frac{\sum_{i=1}^m y_k^i \left(x^i - \mu_k\right)\left(x^i - \mu_k\right)^\top}{\sum_{i=1}^m y_k^i}$$

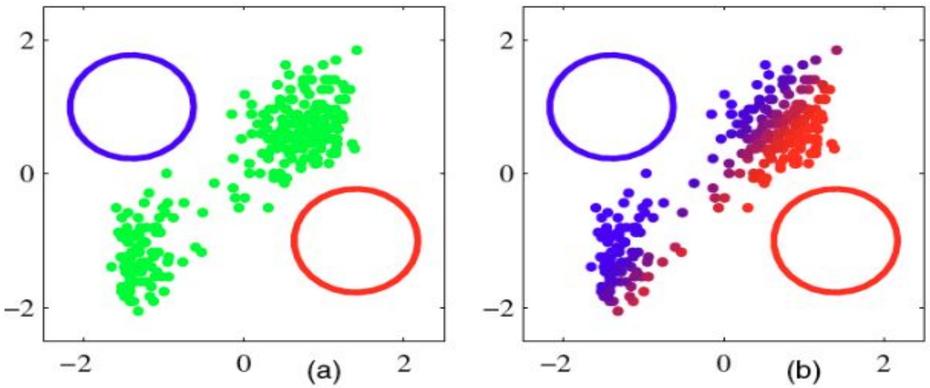
This procedure is actually optimizing an upper bound of MLE, therefore, it converges

EM for GMM

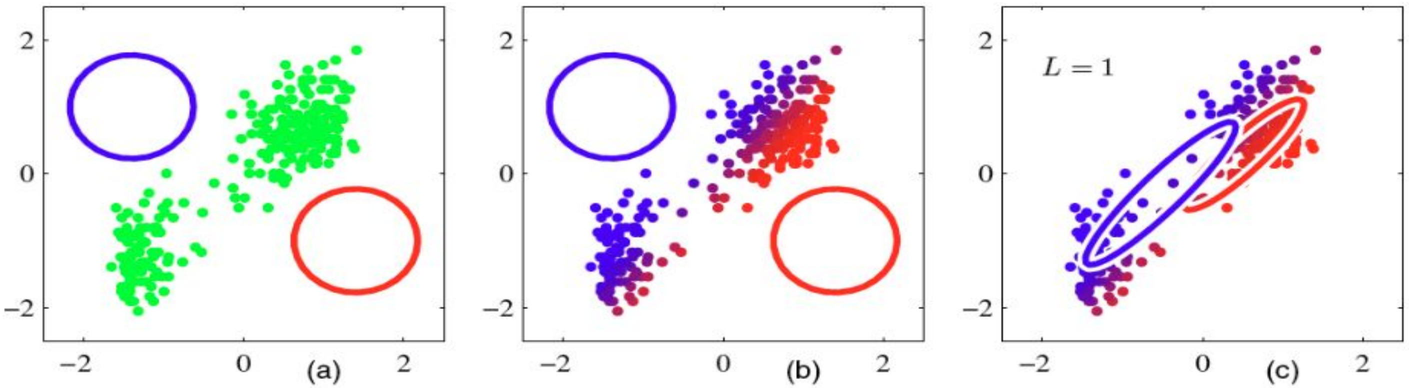


(a)

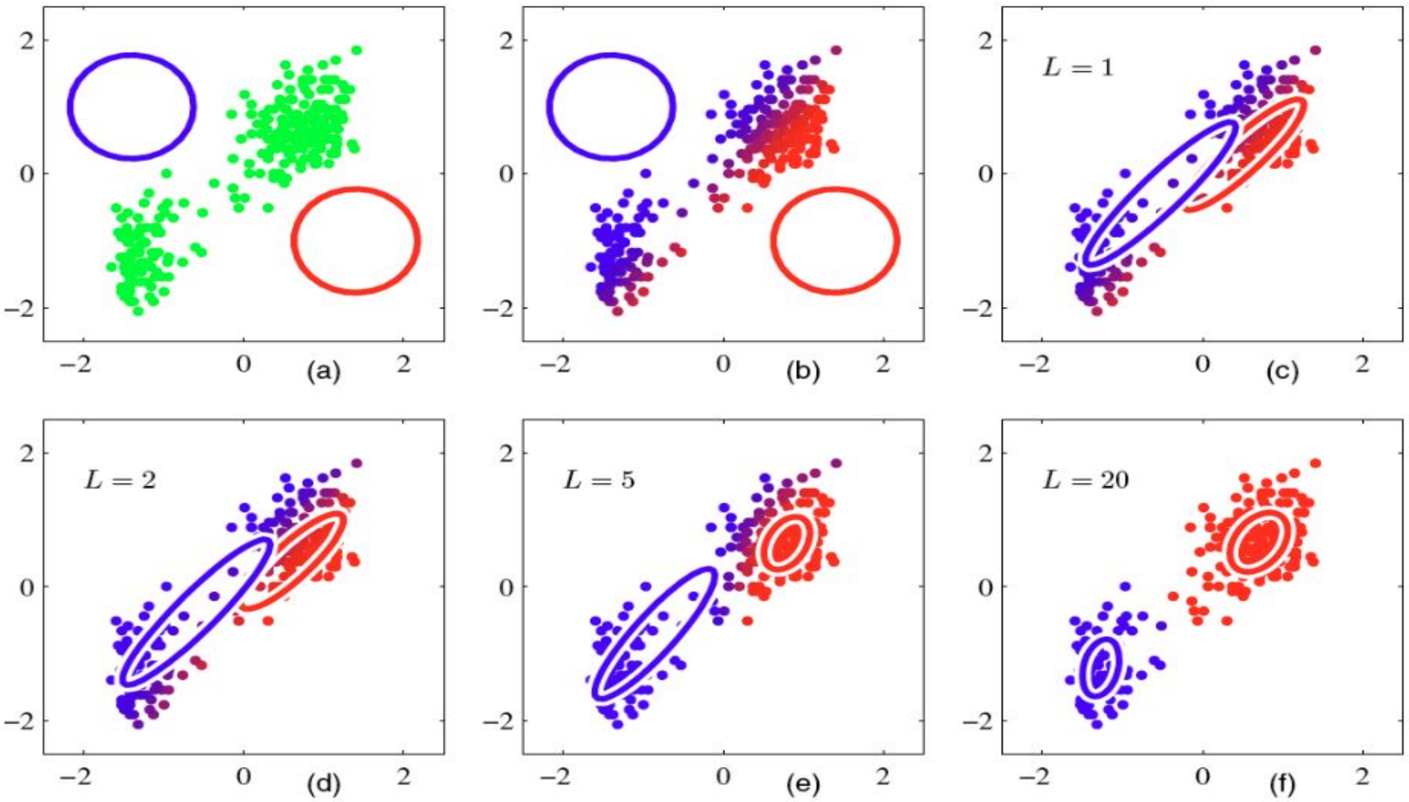
EM for GMM



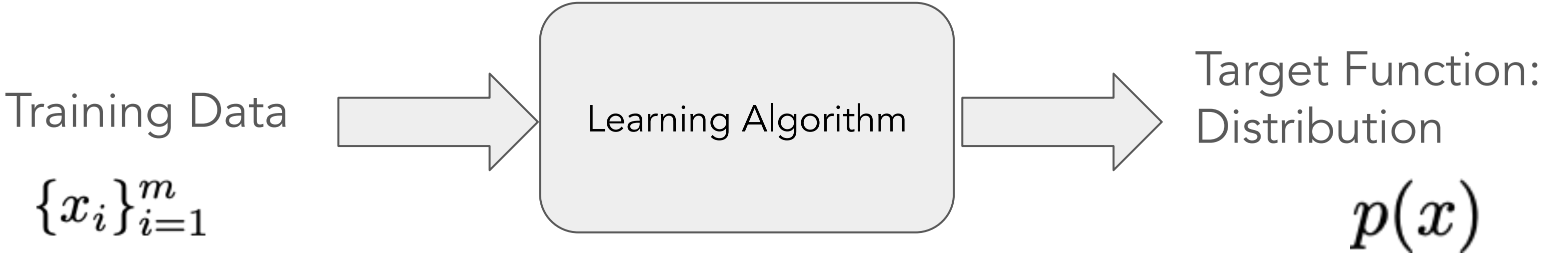
EM for GMM



EM for GMM



Density Estimation: Gaussian Mixture Model



Density Estimation Pipeline

1. Build probabilistic models
   Gaussian Mixture Model
2. Derive loss function (by MLE or MAP….)
   MLE
3. Select optimizer
   EM

Summary

GMM is a special case of hidden variable model $$P(x) = \sum_y P(x|y)P(y)$$

Summary

GMM is a special case of hidden variable model $P(x) = \sum_y P(x|y)P(y)$

A way of maximizing likelihood function for hidden variable models. It can be broken up into two (easy) pieces:

- Estimate some "missing" or "unobserved" data from observed data and current parameters.
- Using this "complete" data, find the maximum likelihood parameter estimates.

Summary

GMM is a special case of hidden variable model $P(x) = \sum_y P(x|y)P(y)$

A way of maximizing likelihood function for hidden variable models. It can be broken up into two (easy) pieces:

- Estimate some “missing” or “unobserved” data from observed data and current parameters.
- Using this “complete” data, find the maximum likelihood parameter estimates.

EM can converge, but can also get stuck in local minima.

Q&A