Pattern Recognition and Machine Learning
(模式识别与机器学习)

左旺孟

哈尔滨工业大学计算机学院
机器学习研究中心
综合楼712
cswmzuo@gmail.com
13134506692

课程内容

- ~~绪论~~
- ~~贝叶斯学习~~
  - 贝叶斯公式
  - ~~贝叶斯判别准则~~
  - NaiveBayes（朴素贝叶斯）
- 线性分类器
- 非线性分类器
- 特征提取
  - 特征选择
  - ~~PCA/LDA~~

课程内容

- 浅层->深度学习
  - ~~随机优化算法~~
  - 激活函数
  - BN
  - ~~Dropout~~
- 卷积神经网络
- 循环神经网络
- Transformer

1.1 单变量选择法

单变量选择法就是把n维特征每个分量单独使用时的可分性准则函数值都算出来，然后按准则函数值按从大到小排序，如

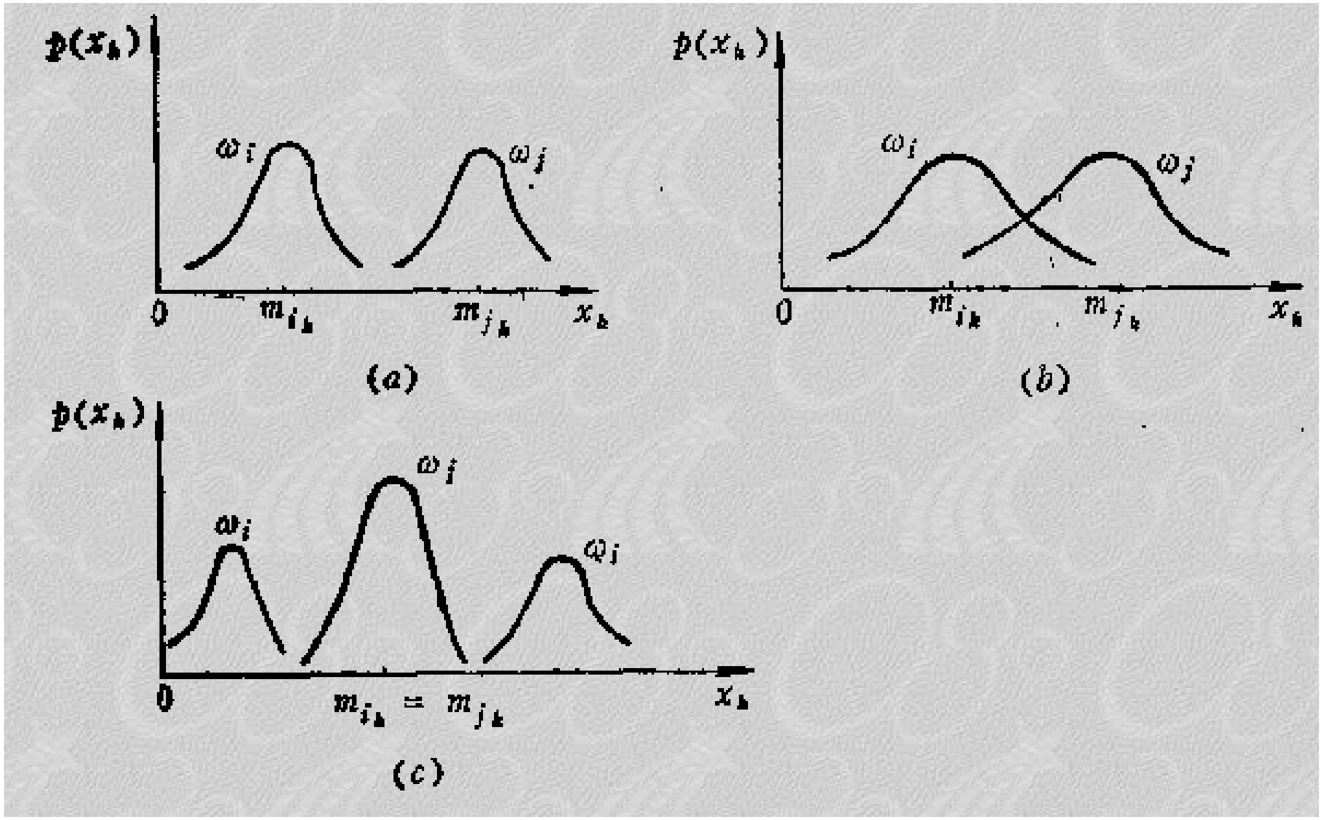
$$G(x_1) > G(x_2) > \cdots > G(x_m) > \cdots > G(x_n)$$

然后，取使 G 较大的前m个特征作为选择结果。

如两类样本 w_i 和 w_j，$G(x_k)$ 可定义为：

$$G(x_k) = \frac{(m_{ik} - m_{jk})^2}{\sigma_{ik}^2 + \sigma_{jk}^2}, \quad k = 1,2,\cdots,n$$

该方法简单，但适用范围与模式特征的概率分布有关。当类概率密度不能用正态分布近似时，不适用。



相关性评价

- 独立性

$$P(X, Y) = P(X)\,P(Y)$$

- 相关性：互信息

$$MI(X, Y) = \int P(X,Y) \log \frac{P(X,Y)}{P(X)P(Y)}\, dX\, dY$$

$$= KL(\, P(X,Y) \,||\, P(X)P(Y) \,)$$

Kullback-Leibler散度

- 熵

$$H(X) \equiv \sum_{x \in \mathcal{A}_X} P(x) \log \frac{1}{P(x)},$$

- 相互熵

$$H(X,Y) = \sum_{xy \in \mathcal{A}_X \mathcal{A}_Y} P(x,y) \log \frac{1}{P(x,y)}.$$

- KL散度

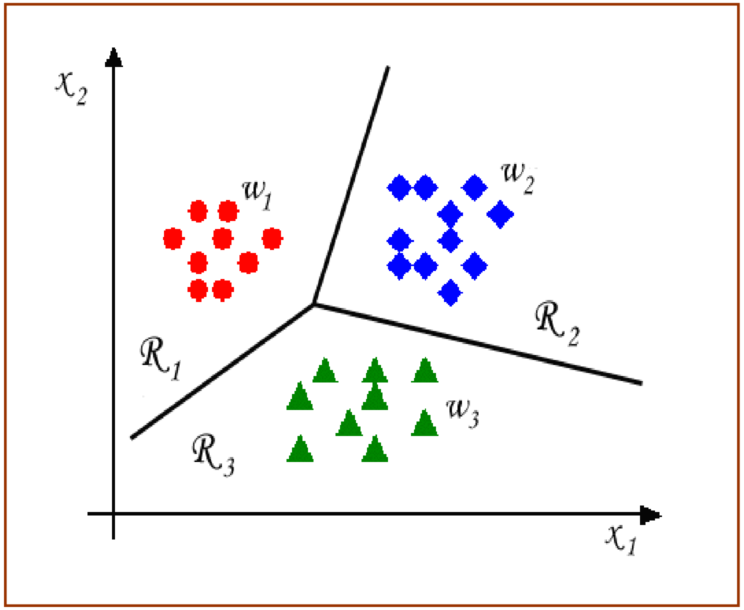
$$D_{\mathrm{KL}}(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}.$$

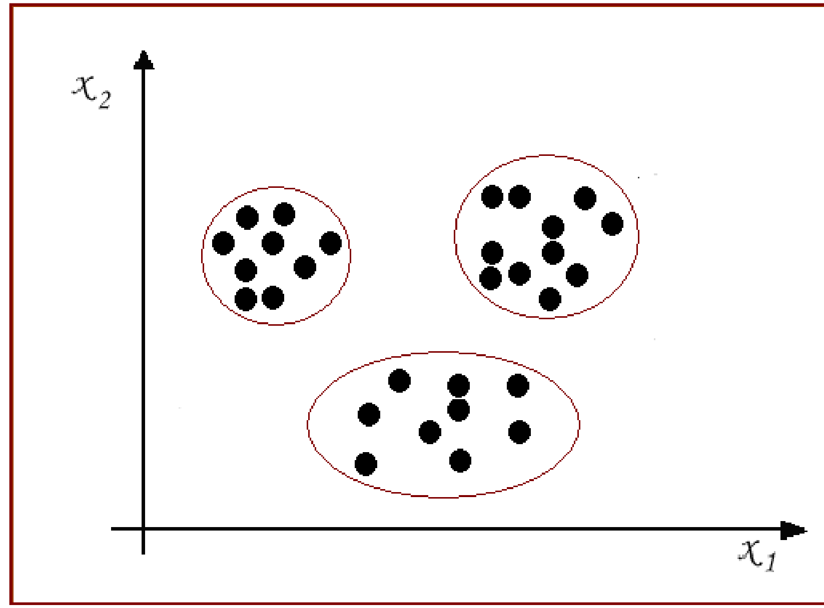
$$I(f_1, f_0) \neq I(f_0, f_1)$$

- Jenson-Shannon熵

$$I_J(f_1, f_0) = I(f_1, f_0) + I(f_0, f_1)$$

分类和聚类





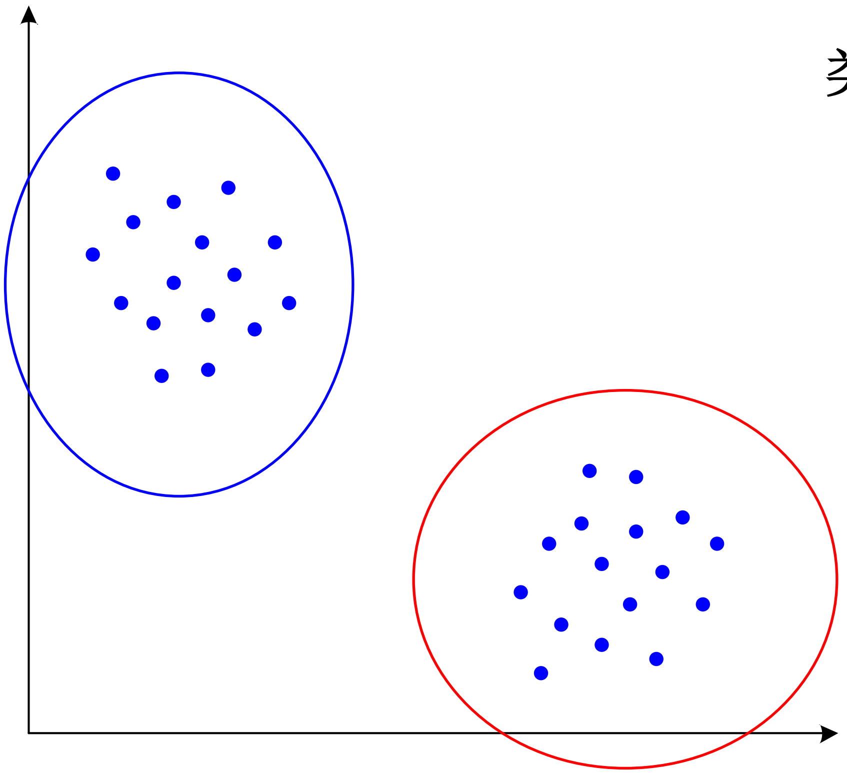
给定样本和类别标签，确定决策函数

给定样本，根据样本的邻近关系等分析其分布的内在结构

聚类分析

- 聚类准则
- 高斯混合模型
  - EM算法
- K均值聚类
  - 模糊K均值聚类
- 层次聚类 (简单介绍)

1. 聚类准则函数



类别数 **c = 2**

误差平方和准则

- 将样本分成c个子集$D_1, \ldots, D_c$，n_i为第i个子集的样本数，m_i为样本均值：

$$\mathbf{m}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in D_i}\mathbf{x}$$

- **误差平方和准则：**

$$J_e = \sum_{i=1}^{c}\sum_{\mathbf{x}\in D_i}\left\|\mathbf{x}-\mathbf{m}_i\right\|^2$$

散布矩阵

- 类内散布矩阵：

$$\mathbf{S}_w = \sum_{i=1}^{c} \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t$$

- 类间散布矩阵：

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t$$

- 总体散布矩阵：

$$\mathbf{S}_T = \sum_{\mathbf{x} \in D} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^t = \mathbf{S}_w + \mathbf{S}_B$$

散布准则

- 基于行列式的散布准则：

$$J_d = \left|\mathbf{S}_w\right|$$

- 基于不变量的散布准则：

$$J_f = tr\left[\mathbf{S}_T^{-1}\mathbf{S}_W\right]$$

2. 高斯混合模型

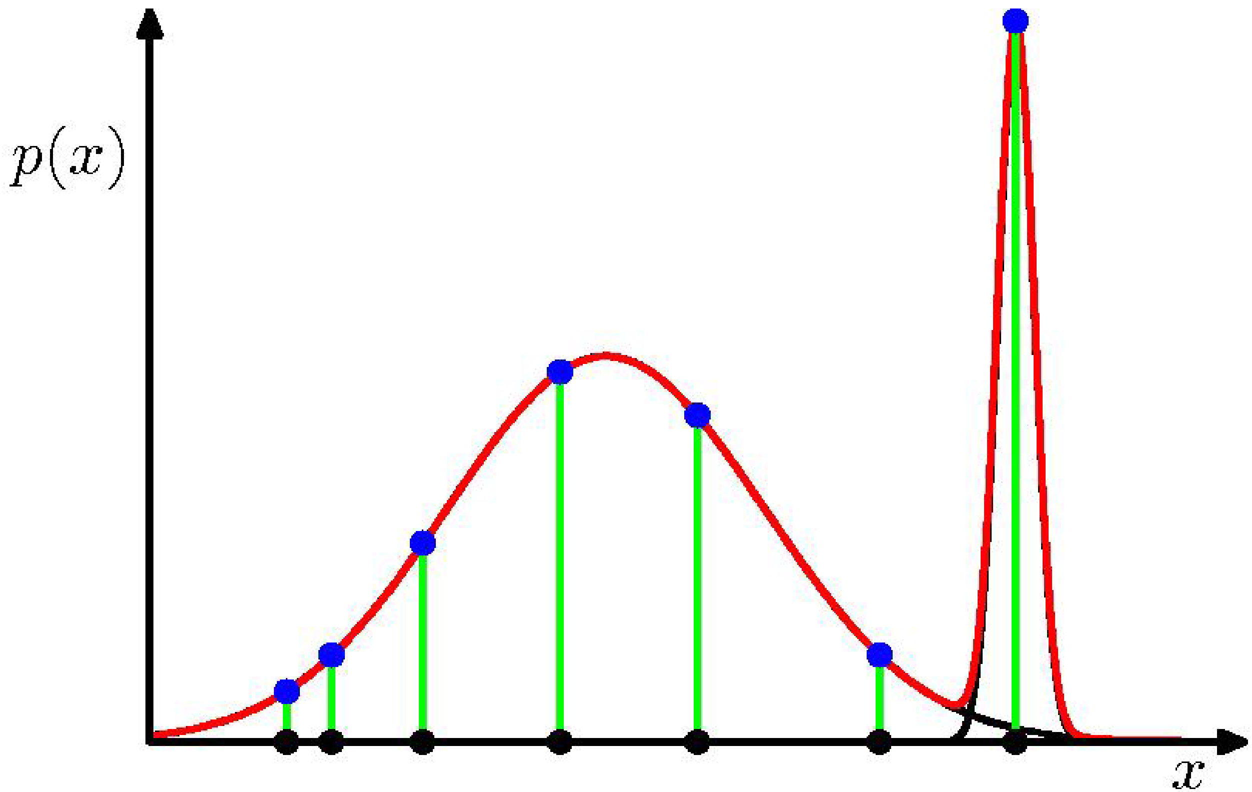
- 一个复杂的概率密度分布函数可以由多个简单的密度函数混合构成：

$$p(\mathbf{x}|\boldsymbol{\theta})=\sum_{i=1}^{M}a_i p_i(\mathbf{x}|\boldsymbol{\theta}_i),\qquad \sum_{i=1}^{M}a_i=1$$

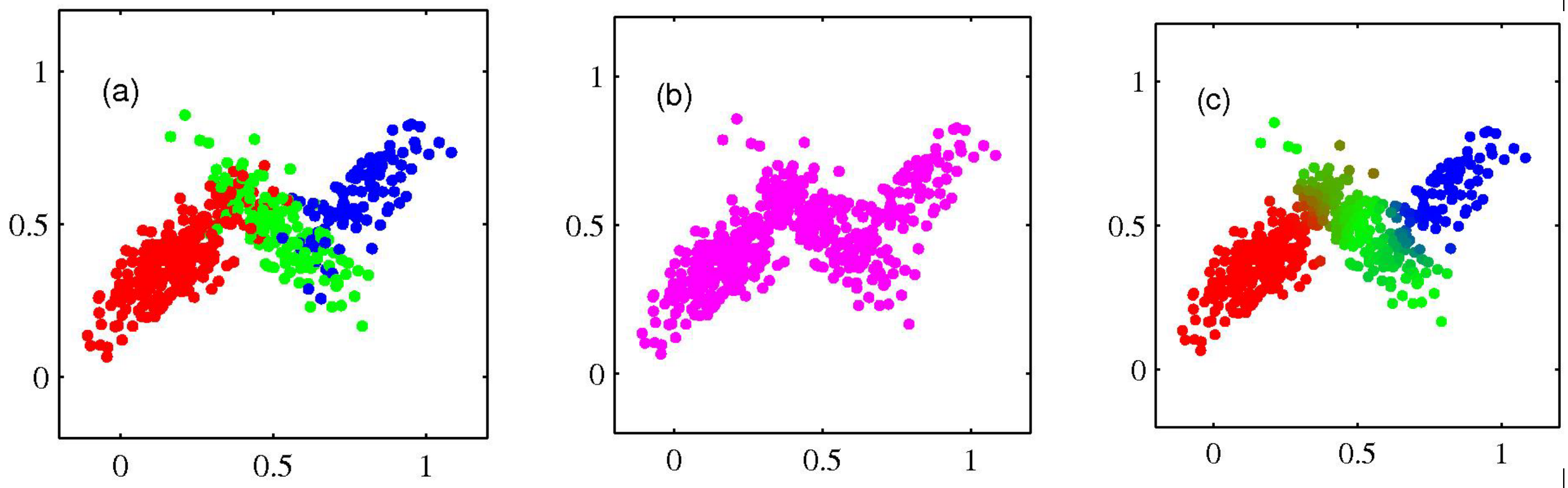
- 最常用的是高斯混合模型(GMM，Gauss Mixture Model)：

$$p(\mathbf{x})=\sum_{i=1}^{M}a_i N(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)\qquad \sum_{i=1}^{M}a_i=1$$

高斯混合模型



GMM模型产生的2维样本数据



混合密度模型的参数估计

- 混合密度模型的参数可以表示为：

$$\boldsymbol{\theta} = \left(a_1, a_2, \cdots, a_M, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_M\right)$$

- 参数的估计方法：
  1. 利用最优化方法直接对似然函数进行优化，如梯度下降法；
  2. 引入未知**隐变量Y**对问题进行简化，将Y看作丢失的数据，使用EM算法进行优化。

GMM模型的参数估计

- 首先引入隐含数据集合: $Y=\{y_1,y_2,\cdots,y_n\}$

其中：$y_i\in\{1,\cdots,M\}$ 代表第i个训练样本是由第 y_i 个高斯函数产生的，将Y作为丢失数据集合，采用EM算法进行迭代估计。

期望最大化(E-M)算法

- E-M算法并不是一种独立的参数估计方法，而是一种针对**数据缺失**情形下进行最大似然估计时使用的一种经典算法。

- 缺失数据：实际应用中得到的观测数据X会不完整，假设完全数据为$Y=(X, Z)$，而我们只能基于观测数据X进行参数估计，此时称Z为缺失数据。

- 由于其简单性、有效性和收敛性，E-M算法已被广泛应用于各种数据缺失情形下的参数估计问题。

缺失数据

- 存在的必然性
  - 诊断数据、微阵列数据、显示器屏幕
  - 隐变量

- 解决方法
  - 解决的必要性
  - 解决的可能性
  - EM算法

期望最大化算法(EM算法)

- EM算法的应用可以分为两个方面：

  1. 训练样本中某些**特征丢失**情况下，分布参数的最大似然估计；

  2. 对某些复杂分布模型假设，最大似然估计很难得到解析解时的迭代算法。
     1. 引入**隐变量**并利用EM算法求解

基本EM算法

- 令**X**是观察到的样本数据集合，Y为丢失的数据集合，完整的样本集合D=X∪Y。

$$p(D|\boldsymbol{\theta}) = p(\mathbf{X},\mathbf{Y}|\boldsymbol{\theta})$$

- 由于Y未知， 在给定参数θ时， **似然函数**可以看作Y的函数：

$$l(\boldsymbol{\theta}) = l(\boldsymbol{\theta}|D) = l(\boldsymbol{\theta}|\mathbf{X},\mathbf{Y}) = \ln p(\mathbf{X},\mathbf{Y}|\boldsymbol{\theta})$$

基本EM算法

- 由于Y未知，因此我们需要根据当前的参数估计结果，寻找到一个在Y的所有可能情况下，平均意义下的似然函数最大值，即似然函数对Y的期望的最大值：

$$Q\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{i-1}\right)=E_{\mathbf{Y}}\left(l\left(\boldsymbol{\theta}\middle|\mathbf{X},\mathbf{Y}\right)\middle|\mathbf{X},\boldsymbol{\theta}^{i-1}\right)$$

$$=E_{\mathbf{Y}}\left(\ln p\left(\mathbf{X},\mathbf{Y}\middle|\boldsymbol{\theta}\right)\middle|\mathbf{X},\boldsymbol{\theta}^{i-1}\right)$$

$$\boldsymbol{\theta}^{i}=\arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{i-1}\right)$$

基本EM算法

1. **begin initialize** $\boldsymbol{\theta}^0$ ，T，i←0;
2. **do** i←i+1
3. E步：计算 $Q\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{i-1}\right)$；
4. **M步**：$\boldsymbol{\theta}^i = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{i-1}\right)$
5. **until** $Q\left(\boldsymbol{\theta}^{i+1}\middle|\boldsymbol{\theta}^i\right) - Q\left(\boldsymbol{\theta}^i\middle|\boldsymbol{\theta}^{i-1}\right) \le T$
6. **return**

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{i+1}$$

EM算法的性质

- EM算法具有收敛性；
- EM算法往往只能保证收敛于似然函数的局部最大值点（极值点），而不能保证收敛于全局最优点。

GMM参数的EM估计算法

1. 设定混合模型数M，初始化模型参数 $\boldsymbol{\theta}^0$ ，阈值T，i←0；
2. 用下列公式迭代计算模型参数，直到似然函数变化小于T为止：

$$p\left(m \middle| \mathbf{x}_t, \boldsymbol{\theta}^i\right) = \frac{a_m^i p_m\left(\mathbf{x}_t \middle| \boldsymbol{\theta}_m^i\right)}{\sum_{j=1}^{M} a_j^i p_j\left(\mathbf{x}_t \middle| \boldsymbol{\theta}_j^i\right)}$$

$$a_m^{i+1} = \frac{1}{n}\sum_{t=1}^{n} p\left(m \middle| \mathbf{x}_t, \boldsymbol{\theta}^i\right)$$

$$\boldsymbol{\mu}_m^{i+1} = \frac{\sum_{t=1}^{n} \mathbf{x}_t p\left(m \middle| \mathbf{x}_t, \boldsymbol{\theta}^i\right)}{\sum_{t=1}^{n} p\left(m \middle| \mathbf{x}_t, \boldsymbol{\theta}^i\right)}$$

$$\boldsymbol{\Sigma}_m^{i+1} = \frac{\sum_{t=1}^{n} p\left(m \middle| \mathbf{x}_t, \boldsymbol{\theta}^i\right)\left(\mathbf{x}_t - \boldsymbol{\mu}_m^{i+1}\right)\left(\mathbf{x}_t - \boldsymbol{\mu}_m^{i+1}\right)^t}{\sum_{t=1}^{n} p\left(m \middle| \mathbf{x}_t, \boldsymbol{\theta}^i\right)}$$

3. k-均值聚类

1. begin initialize 样本数n，聚类数c，初始聚类中心m_1, …, m_c;
2. do 按照最近邻m_i分类n个样本;
3. 重新计算聚类中心m_1, …, m_c;
4. until m_i不再改变;
5. return m_1, …, m_c;
6. end

k-均值聚类的特点

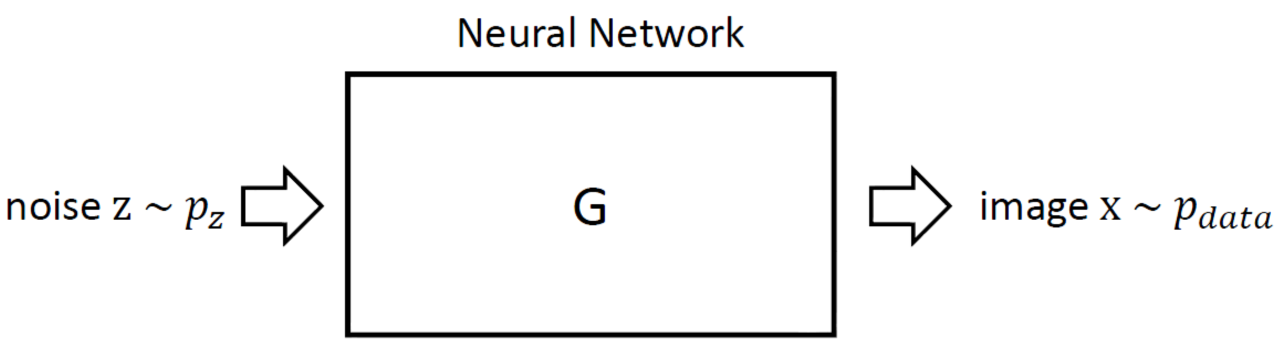
$$J_e = \sum_{i=1}^{c} \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2$$

- k-均值算法可以看作是对**平方误差准则**函数的**贪心搜索**算法；
- 聚类结果受初始聚类中心的选择影响很大，不同的初始聚类中心会导致不同的聚类结果。

图像生成

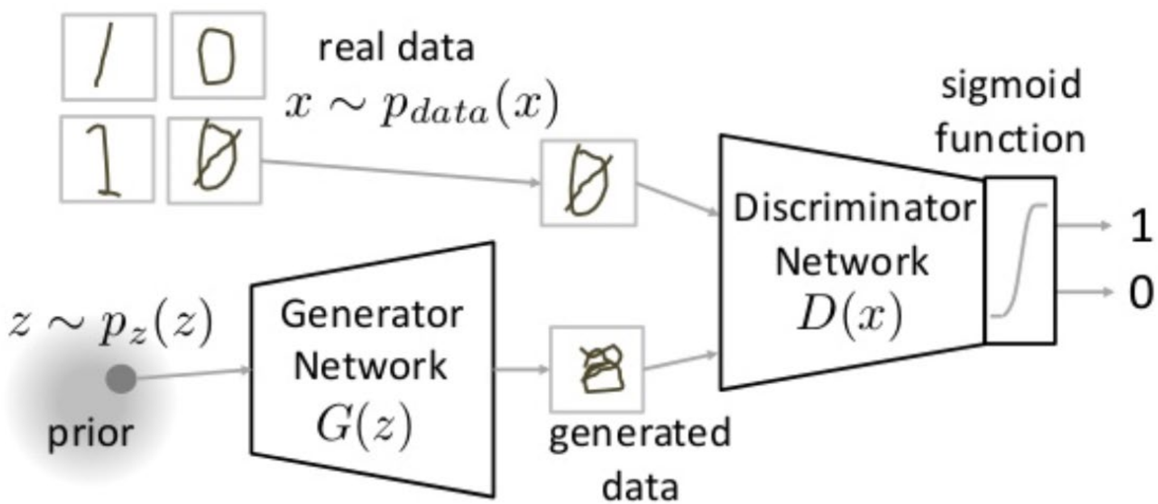
图像生成

- 目标：将任意一个随机变量（向量、矩阵）转化为一幅高质量图像
  - 输入：服从某随机分布（如：高斯分布）的向量或矩阵
  - 输出：多样性、高质量图像（符合特定分布，如自然图像或特定类别）



生成式对抗网络 (Goodfellow et al., NIPS 2014)

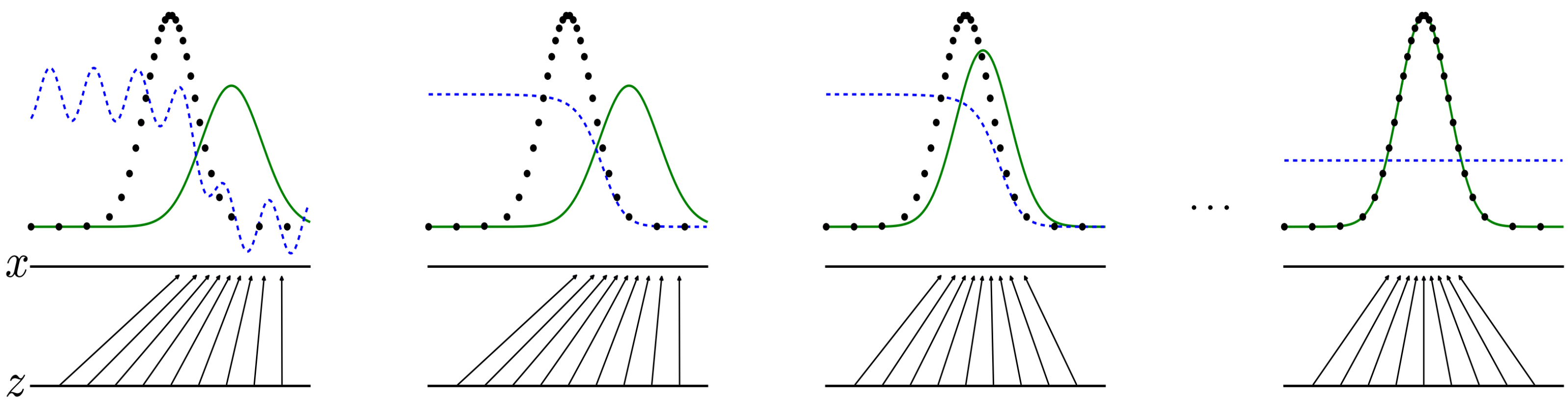
- Update the generator to generate more realistic image
- Update the discriminator to discriminate the synthetic images from real ones



Two-player minimax game

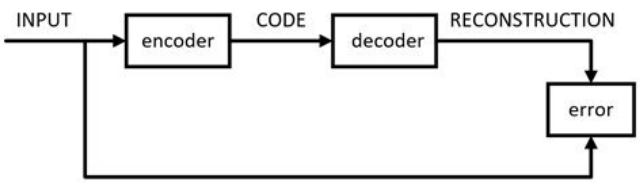
- Value function

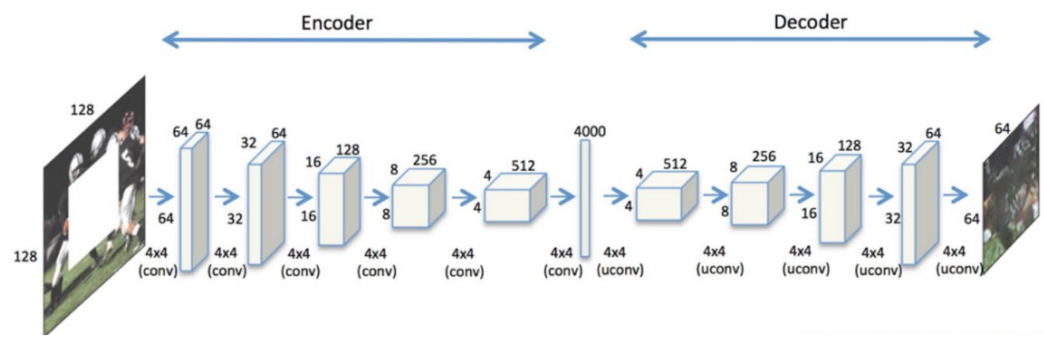
$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$



Auto-encoder

- Auto-encoder





$$\phi, \psi = argmin_{\phi,\psi} L(X, (\psi \circ \phi)X)$$

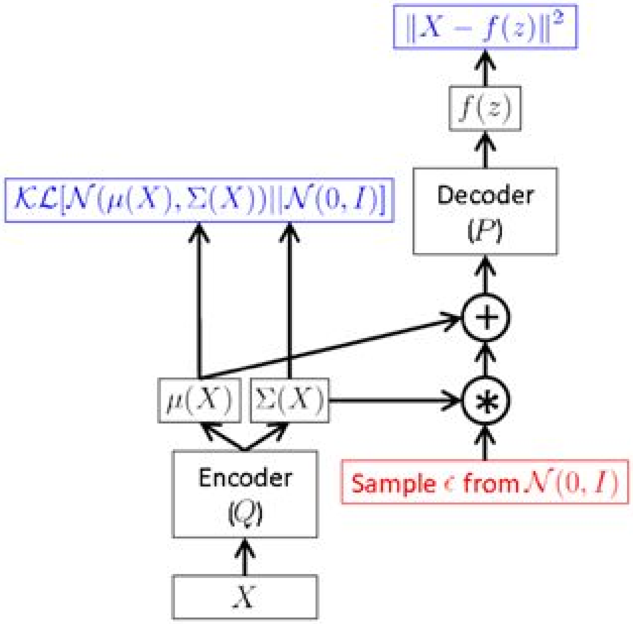
- Denoising auto-encoder

Variational AutoEncoder

- Variational AutoEncoder

$$l_i = -E_{z\sim q}[logp(x_i|z)] + KL[q(z|x_i)||p(z)]$$

  - Relaxation of discrete variable



VAE/GAN (Larsen et al., ICML 2016)

- VAE

$$\mathcal{L}_{\text{prior}} = D_{\text{KL}}(q(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))$$

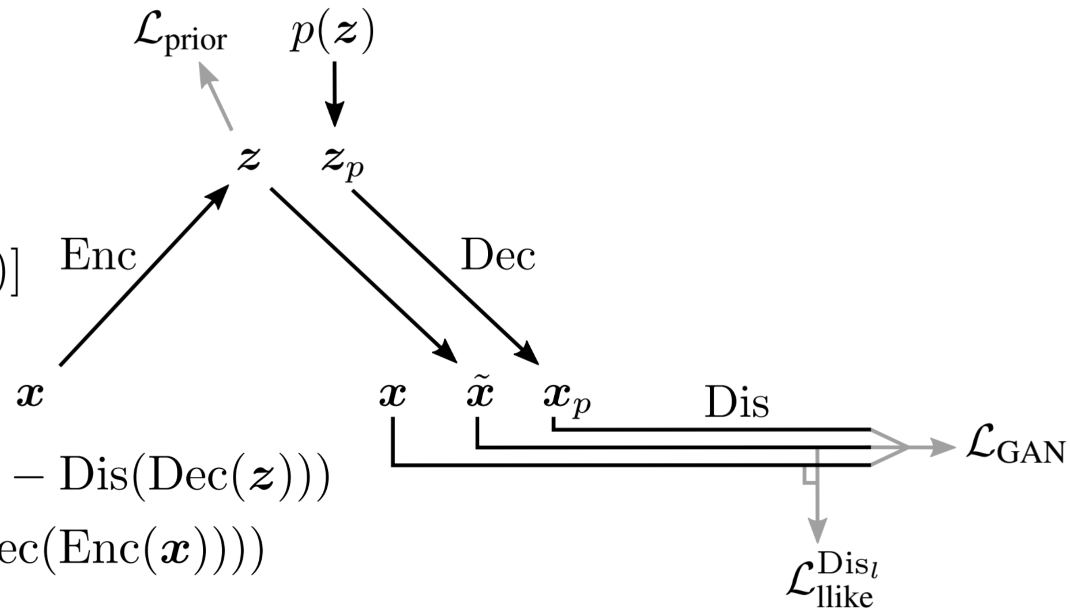
$$\mathcal{L}_{\text{llike}}^{\text{Dis}_l} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\text{Dis}_l(\boldsymbol{x})|\boldsymbol{z})\right]$$

- GAN

$$\mathcal{L}_{\text{GAN}} = \log(\text{Dis}(\boldsymbol{x})) + \log(1-\text{Dis}(\text{Dec}(\boldsymbol{z}))) + \log(1-\text{Dis}(\text{Dec}(\text{Enc}(\boldsymbol{x}))))$$

- VAE/GAN

$$\mathcal{L} = \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{llike}}^{\text{Dis}_l} + \mathcal{L}_{\text{GAN}}$$



闭卷考试，好好复习

都能取得自己满意的成绩