# 机器学习

> 简要整理，复习时作提纲用

UNIkeEN, 2024.6

## 1.Intro

- 七个基本学习基础：监督学习、非监督学习、半监督（迁移）学习、强化学习、对弈学习、群体智能、集成学习

## 需要额外复习的基础知识

- 向量/矩阵求导

Matrix-cook-book

$$
\begin{aligned}
\partial \mathbf{A} &= 0 \qquad (\mathbf{A} \text{ is a constant}) \\
\partial(\alpha \mathbf{X}) &= \alpha \partial \mathbf{X} \\
\partial(\mathbf{X}+\mathbf{Y}) &= \partial \mathbf{X} + \partial \mathbf{Y} \\
\partial(\mathrm{Tr}(\mathbf{X})) &= \mathrm{Tr}(\partial \mathbf{X}) \\
\partial(\mathbf{X}\mathbf{Y}) &= (\partial \mathbf{X})\mathbf{Y} + \mathbf{X}(\partial \mathbf{Y}) \\
\partial(\mathbf{X} \circ \mathbf{Y}) &= (\partial \mathbf{X}) \circ \mathbf{Y} + \mathbf{X} \circ (\partial \mathbf{Y}) \\
\partial(\mathbf{X} \otimes \mathbf{Y}) &= (\partial \mathbf{X}) \otimes \mathbf{Y} + \mathbf{X} \otimes (\partial \mathbf{Y}) \\
\partial(\mathbf{X}^{-1}) &= -\mathbf{X}^{-1}(\partial \mathbf{X})\mathbf{X}^{-1} \\
\partial(\det(\mathbf{X})) &= \det(\mathbf{X})\mathrm{Tr}(\mathbf{X}^{-1}\partial \mathbf{X}) \\
\partial(\ln(\det(\mathbf{X}))) &= \mathrm{Tr}(\mathbf{X}^{-1}\partial \mathbf{X}) \\
\partial \mathbf{X}^{T} &= (\partial \mathbf{X})^{T} \\
\partial \mathbf{X}^{H} &= (\partial \mathbf{X})^{H}
\end{aligned}
$$

  - 1x1矩阵变Tr -> Tr里的元素可以“转圈”；Tr对称阵是特征根之和

- 概率统计（各类分布、高斯分布、高维高斯分布、似然估计）PPT3
  - 高维高斯求log

# Multivariate Gaussian

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

## 2.Cluster

### 距离度量

- 闵可夫斯基距离
- 非负性、同一性、对称性、直通性（三角不等式）
- 平方距离最小的点 -> 均值

### K-means

- 初始化中心 -> 全部点分配 -> 更新全部中心 -> 迭代
- 是否一定收敛（HW1 T1），即证明E和M两个步骤都不会增加误差函数
- 解不唯一，性能受初始化点的影响、可能陷入极值点而找不到全局最优解
- 最坏O(tkn)，t迭代次数
  - 改进思考1：在某些数据下更换距离度量（比如数据分布类似于狭长椭圆时 -> 马氏距离）
  - 改进思考2：分配点时是否引入概率（soft方法，如 HW1 T2）
    - 距离与概率分布的关系（$e^{-d}$）
  - 改进思考3：如何从数据中确定最佳的类别数K

### 层次聚类

- 初始化每个点自成聚类 -> 最近两个聚类合并
- 最小距离、最大距离、平均距离、豪斯多夫距离
- 不需要预先设置聚类数，最后根据过程确定（树状图）
- 确定距离度量和目标聚类数，解唯一吗?
- 缺点：时间复杂度高（视距离度量 O(n^2log2) 或 O(n^3)）

### 数据非一次性给出（competitive learning）

- 新数据分配到中心后，中心向新数据**部分**移动
- 可能导致"一家独大"，改进方法如下：
- FSCL，对高频winner做惩罚（如距离权重，已经有n个点的中心计算与新点的距离时乘以n）
  - 缺点：过分的平均可能会惩罚过头（如数据大致只有四类、共五个点时第五个点仍然会移动到数据之间）
- RPCL，让第一近点靠近的同时、第二近点（rival）略微远离以惩罚

- 可能的改进方向：是否对 winner 进行额外奖励？

> 问题 Q: competitive learning 和 K-means 是否等价？
>
> A: K-means更新均值是$x_{t+1}^- = \bar{x}_t + \frac{1}{t+1}(x_{t+1} - \bar{x}_t)$ 从这个角度上等价于学习率动态变化的 CL（这里的 1/(t+1)就是靠近新点的距离） 两者的比较见 HW1 T3
>
> Q: 设计 K-means 版本的 RPCL
>
> A: HW1 T3

## GMM

- 假设样本的生成过程由高斯混合分布给出（西瓜书 P206）
  - 从假定由高维高斯分布生成的样本进行最大似然估计求参数
  - 距离 -> 概率，归一化带来参数使高斯分布的协方差矩阵可解
- Expectation-Maximization（EM）
  - GMM 的 EM：计算后验概率 -> 更新各个高斯的均值向量、协方差矩阵、混合系数（高斯之间的权重）-> 直到满足条件（参数收敛/最大似然估计收敛）
  - 任意分布的 EM 算法

    The General EM Algorithm

    Given a joint distribution $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ over observed variables $\mathbf{X}$ and latent variables $\mathbf{Z}$, governed by parameters $\boldsymbol{\theta}$, the goal is to maximize the likelihood function $p(\mathbf{X}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

    1. Choose an initial setting for the parameters $\boldsymbol{\theta}^{\text{old}}$.
    2. **E step** Evaluate $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$.
    3. **M step** Evaluate $\boldsymbol{\theta}^{\text{new}}$ given by

       $$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$$

       where

       $$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}).$$

    4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

       $$\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$$

       and return to step 2.

    12

  - M-step，最大化对数似然
- 是否能找到全局最优解？不能，基于迭代的方法通常难以找到全局最优解
- GMM is more general then K-means by considering mixing weights, covariance matrices, and soft assignments.

> 问题 Q: EM for GMM 退化到 K-means

> A: 取概率最大的先分配，先验概率中的权重退化到 1/k
>
> Q: 设计一种 k-mean 和 EM 之间的算法
>
> A: （HW1）方法很多，一种方法是将距离修改为马氏距离，协方差矩阵由样本动态计算得；或者保留概率最大的前两个进行等比例/按比例分配... 还有将欧式距离变成概率进行分配...

### KL散度

- 性质：大于等于0，度量两个分布的距离
- 对于离散分布：

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} = E_P\left[\log \frac{P(i)}{Q(i)}\right]$$

- 对于连续分布：

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} P(x) \log \frac{P(x)}{Q(x)}\, dx = E_P\left[\log \frac{P(x)}{Q(x)}\right]$$

## 3.Learning theory: General EM, Maximum Likelihood, Bayesian Learning

### General EM

- 分布 Y 生成数据 X -> 生成 q(x|y, $\theta$)q(y|$\theta$) = 待求 p(y|x)p(x) -> min KL(p(y|x)p(x)|q(x|y, $\theta$)q(y|$\theta$)) -> min KL(p(x)||q(x|$\theta$)) = min(-log q($x_n|\theta$))

> 问题 Q: E-step 无法求得 p(y|x) 的解析解，怎么办
>
> A: 模拟方法，采样（蒙特卡洛，吉布斯）/ 给 p(y|x) 赋予一个“退而求其次”的选择，比如高斯分布（逼近，均值比如取一个线性函数）
>
> Q: M-step，Q函数的优化无法使导数为0，如何解决（不能maximize）
>
> A: 求出导数之后使用梯度法（还是沿着最大似然的方向走）

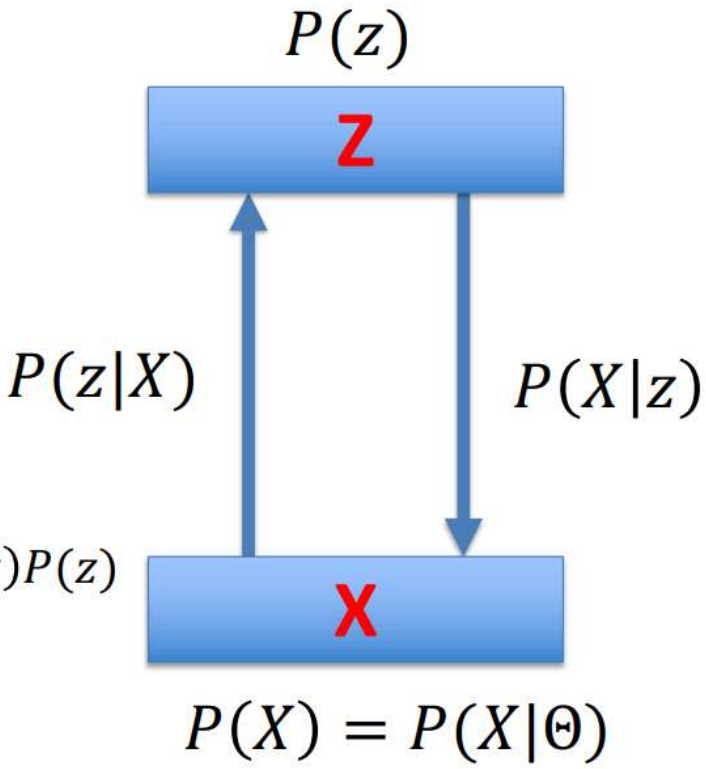
# Recall EM algorithm

**E-Step:** Compute

$$P^{old}(z|X) = \frac{P(X|z)P(z)}{P(X)}$$

**M-Step:** Update Θ by

$$\max_{\Theta} \int P^{old}(z|X) \ln P(X|z)P(z)$$




[TODO]

## 模型选择

- k-means和GMM中K越大J（或似然函数的相反数）越小，所以不能自动确认K
  - k-means每个数据一类，J降到0；GMM中在分好的类的某个高斯边缘加入小高斯，似然函数上升
- 同时考虑效果和模型复杂度，trade-off曲线（当K到一定程度，优化幅度不如模型复杂度的上升程度时，拐点为最优K）
- AIC：$\ln p(x|\theta) - d_k$，$d_k$ 是模型的自由参数个数
- BIC：$\ln p(x|\theta) - 1/2 d_k \ln N$，$N$ 是样本的数量（BIC会比AIC更准确？）

## 贝叶斯学习

- 最大似然估计(MLE)，最大化 $P(X|\theta)$，独立同分布时似然函数是累乘$P(x_i|\theta)$
  - MLE广泛应用于机器学习模型的训练，例如线性回归、逻辑回归、正态分布参数估计等。
- 最大后验估计(MAP)，通过最大化后验概率估计模型参数，最大化 $P(\theta|X)$ 等价于 最大化 $P(X|\theta)P(\theta)$
  - 先验分布$P(\theta)$是观察到数据之前对参数的先验知识
  - MAP在处理小样本数据时特别有用，因为它可以利用先验知识进行更稳健的估计。常见应用包括贝叶斯网络、正则化回归等。
- 在样本量很大时，MLE和MAP的估计结果趋于一致。但在样本量较小或存在先验知识时，MAP可以提供更好的估计

- 贝叶斯模型选择：

## Two-phase method for model selection

- Assume the optimal K* is within the range $[1, K_{max}]$.
- Phase (1): For each $k = 1, \ldots, K_{max}$, compute the maximum likelihood estimator:

$$\hat{\Theta}_{ML}(k) = \underset{\Theta}{\operatorname{argmax}} \log[P(X|\Theta, k)]$$

- Phase (2): Select the optimal K* by optimizing the values of the model selection criterion J, e.g., AIC, BIC:

$$K^* = \underset{k}{\operatorname{argmax}} J(\hat{\Theta}_{ML}(k))$$

Akaike's Information Criterion (AIC)

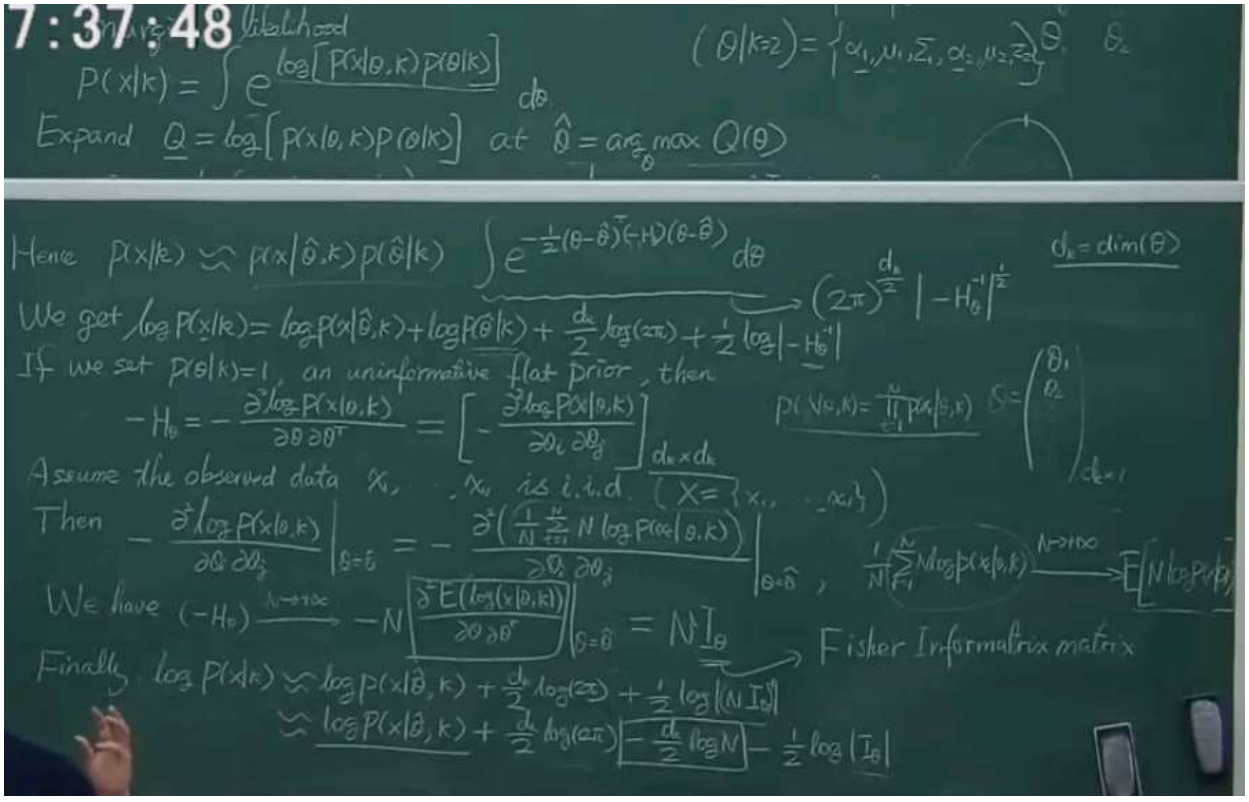
$$\ln p(X_N | \hat{\Theta}_K) - d_k$$

Bayesian Information Criterion (BIC)

$$\ln p(X_N | \hat{\Theta}_K) - \frac{1}{2} d_k \ln N$$

11

- marginal likelihood（BIC背后原理）
  - 边际似然是模型的贝叶斯证据、最大则最优




## 4. Linear Model

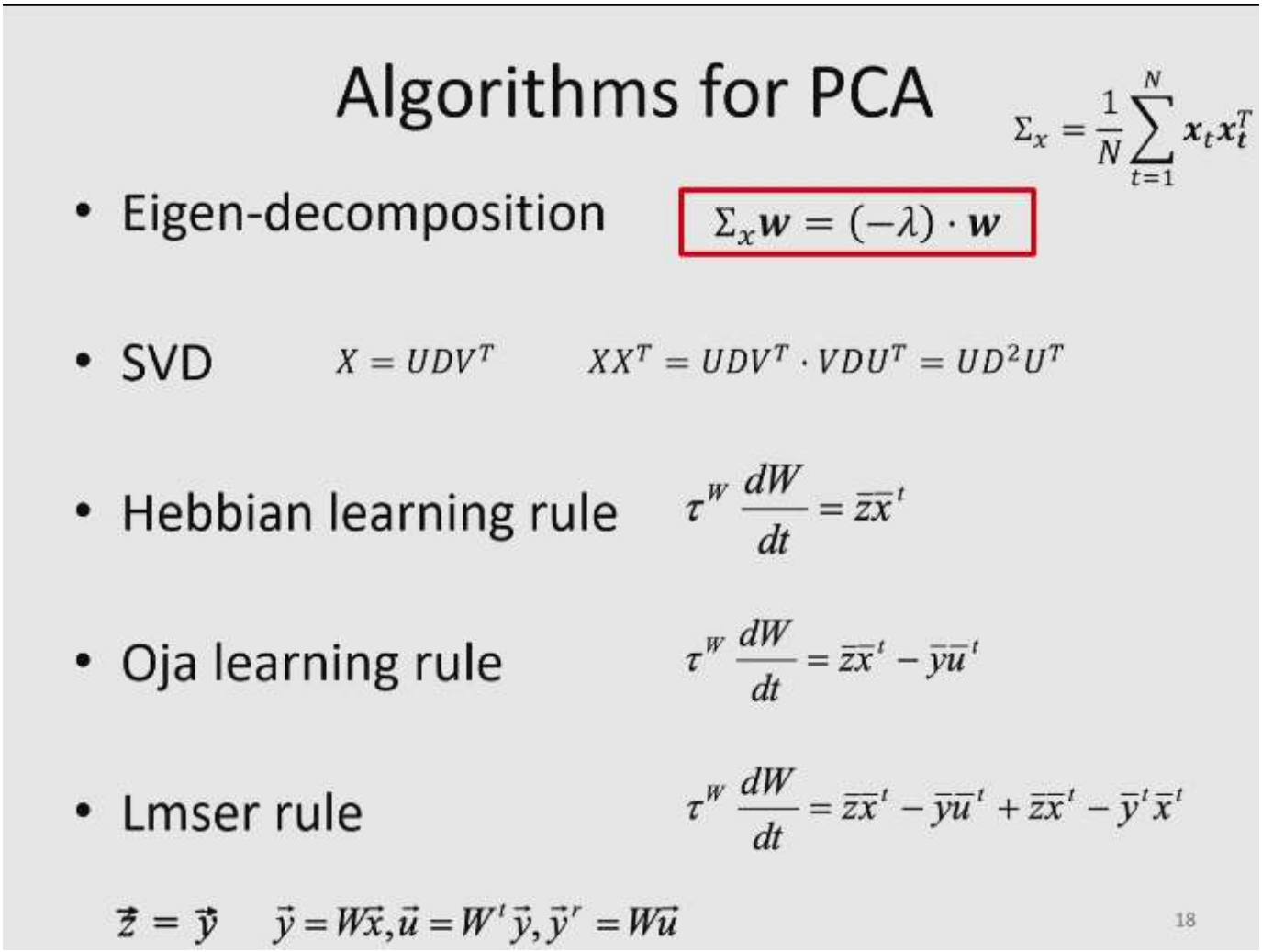
### 数学知识

- $L_p$ norm, 高维 Inner Product
- 泰勒展开
- 特征分解 Av=λv -> det(A-λI)=0 -> 取max λ -> 带回(A-λI)v=0求v
- 拉格朗日乘数法（最优解点，目标函数和约束函数梯度呈比例关系）

## PCA

- 投影后方差最大化 <-> 重构误差最小化 -> 特征分解（v是协方差矩阵 XX^T 的特征向量）
- 应用：去除数据相关性、去除冗余、去除噪声、降维、特征提取（可视化）
- Hebbian Learning: 神经元权重$\omega_i$的变化量正比于$yx_i$
  - Hebbian 学习会使神经元权重向量对输入数据中方差最大的方向进行对齐。类似于 PCA 中的第一主成分的提取
- LMSER for PCA

---

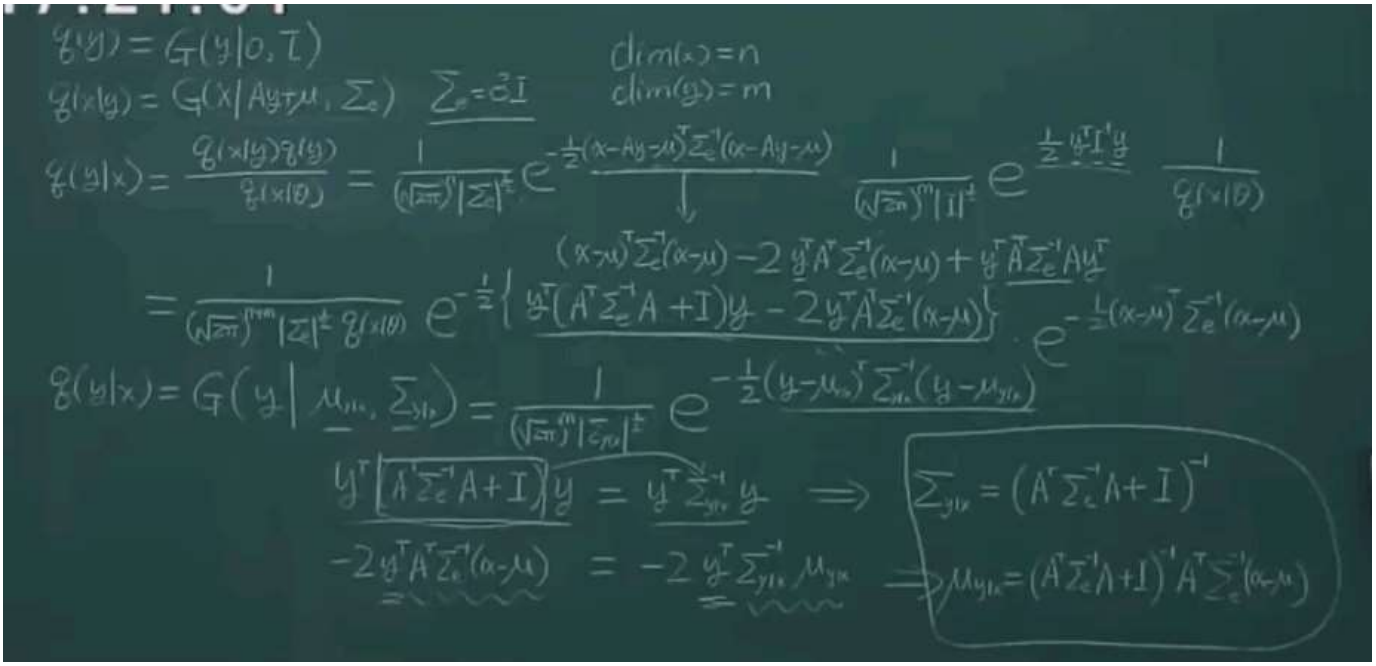



> Q: 比较各个方法
>
> A: 特征分解和 SVD 能算出所有特征向量，特征分解需要计算协方差矩阵，维度大时开销大
> SVD可以处理非方阵的矩阵（特征分解只适用于方阵）

- 选择降维类数，重建误差<=0.01 等价于 选择特征根和占总和>=0.99

$$\frac{\frac{1}{N}\sum_{t=1}^{N}||x_t-\hat{x}_t||^2}{\frac{1}{N}\sum_{t=1}^{N}||x_t||^2}\leq 0.01 \quad \Longleftrightarrow \quad \frac{\sum_{i=1}^{m}\lambda_i}{\sum_{j=1}^{n}\lambda_j}\geq 0.99$$

## FA



## ICA

- 互信息，衡量相关性，实为 p(x,y)和p(x)p(y)的KL
  - entropy
  - Where $H(\mathbf{y})$ is entropy defined by

$$H(Y)=-\sum_i P(Y=a_i)\log P(Y=a_i) \quad \text{discrete}$$

$$H(\mathbf{y})=-\int f(\mathbf{y})\log f(\mathbf{y})\mathrm{d}\mathbf{y} \quad \text{continuous}$$

The more "random", i.e. unpredictable and unstructured the variable is, the larger its entropy.

- BSS 问题，盲源分离 -> 寻找
  - x=As，源信号分量独立；问题的目标是寻找 W，得到 s=Wx

> PCA 和 ICA 的差别与联系 ...

> FA+GMM，有没有别的组合方式

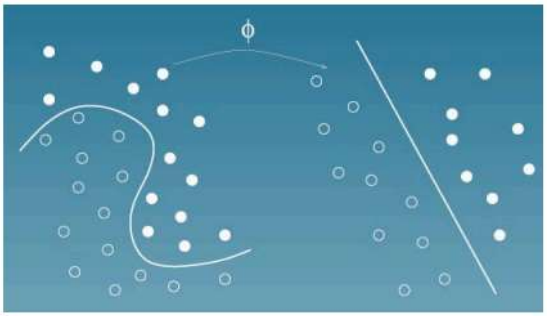
## SVM

西瓜书 P123-124

- 基本上线性可分（部分存在误差）：引入C
- 分界面完全非线性：引入核，把原数据映射到特征空间（对偶问题涉及所有训练样本对之间的相互作用，故不需要直接考虑每个样本，而只需考虑样本两两之间的关系）

# The idea of kernel representation

- Data are not represented individually anymore, but only through a set of pairwise comparisons.
- Instead of using a mapping $\phi: \mathcal{X} \rightarrow \mathcal{F}$ to represent each object $\mathbf{x} \in \mathcal{X}$ by $\phi(\mathbf{x}) \in \mathcal{F}$, a real-valued "comparison function" (called kernel) $k: \mathcal{X} \times \mathcal{X} \rightarrow \Re$ is used, and the data set $\mathcal{S}$ is represented by the $n \times n$ matrix (Gram matrix) of pairwise comparisons $k_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$.
- A main question is how to find a kernel $k$ such that, in the new space, problem solving is easier (e.g. linear).
- All kernel methods have two parts:
  (1) find such a kernel $k$, and
  (2) process such Gram matrices.




- 核SVM与线性SVM关系：...
- 核SVM与深度神经网络关系：核SVM依赖于核函数将数据映射到高维特征空间，这种映射是固定的且非参数化的。而深度神经网络通过多层非线性变换学习数据的特征表示，这种映射是自适应的且参数化的。核SVM适用于中小规模数据集、大规模时训练时间长

## 5.Supervised learning

### 线性回归

- 最小二乘法->最小均方误差，矩阵形式将b吸收进w(w->(w;b))，数据X为m*(d+1)、最后一列全1
  - 最小化(y-Xw)T(y-Xw)
- 逻辑斯蒂回归：转化为分类问题，log(y/1-y)=wTx+b，y=1/(1+e^-(wTx+b))
- 正则化惩罚可以起到模型选择的作用（过大，unde-fitting），加入约束防止模型过拟合
  - model-selection 裁去整个部门；regularization 部分降薪（特征选择）
  - 二者都旨在提高模型的泛化能力，避免过拟合
  - L1绝对值（LASSO）、L2平方项（Ridge）、弹性网（combined）

> L1正则化和L2正则化的联系和区别
>
> 共同点:

1. 防止过拟合：L1和L2正则化都通过增加正则化项来防止模型过拟合，进而提高模型的泛化能力。
2. 正则化项：两者都在损失函数中加入一个正则化项，用于惩罚过大的模型参数。
3. 超参数：都依赖于一个超参数（通常记作λ）来控制正则化项的权重，平衡模型的拟合度和复杂度。

区别： 1.结构不同，... 2.L1正则化倾向于产生稀疏模型，即许多参数会被压缩为零。这对于特征选择非常有用，因为它可以自动选择重要的特征；L2正则化倾向于缩小所有参数的值，但不会将参数压缩为零。因此，L2正则化更适合处理高维度数据，尤其是当特征之间存在共线性时。 3.L1正则化约束区域是一个高维菱形，使得解可能在坐标轴上。这导致了一些参数被压缩为零；L2正则化约束区域是一个高维球体，使得解更可能均匀分布在球体表面，参数较小但不为零 4.L1正则化由于其稀疏性，解的计算可能更复杂，尤其是在高维数据中,适用于需要自动特征选择的场景。L2正则化：计算相对简单，适用于大多数情况下的数值优化问题(适用于特征数量较少或者特征之间存在共线性的情况，能够稳定参数估计并提高模型的鲁棒性)。

## 神经网络

- M-P 神经元（相比 Hebbian 神经元只多一个 sigmoid）
- 感知机：输入层 -> M-P神经元（输入层三项：1,x1,x2，单层下可以扮演OR、AND、NOT，不能扮演XOR）
  - OR、AND、NOT都有线性超平面分开两个模式，XOR不存在线性分类器
  - 线性可分问题，感知机的学习过程一定会收敛
  - 梯度更新：$\delta w = \eta(y - y')x_i$

> Hebbian神经元
>
> - **算法基础**：基于Hebb规则，即“同时激活”的神经元之间的连接会变强。
> - **学习规则**：权重更新是通过输入和输出的乘积来调整，即 $\Delta w_i = \eta \cdot x_i \cdot y$，其中 $\eta$ 是学习率，$x_i$ 是输入，$y$ 是输出。
> - **特点**：用于解释神经元联结的强化机制，多用于生物神经网络建模。
>
> M-P神经元（感知机模型）
>
> - **算法基础**：基于阈值逻辑单元。
> - **学习规则**：使用感知机算法，通过误差校正规则更新权重，即 $\Delta w_i = \eta \cdot (d - y) \cdot x_i$，其中 $d$ 是目标输出，$y$ 是实际输出。注意：如果某特征在当前样本中值较大、其权重更新也相对大
> - **特点**：用于线性可分问题的二分类器，是早期人工神经网络的基础。

- 双层神经网络（再加线性输出）可以近似任意连续函数、近似到任意精度（前提是具有足够多的隐藏unit）
  - 简化公式，把wx+b的b并入权重矩阵（则每层输入多个1）
- 神经网络：wide or deep？open question
  - deep的函数比较平滑、提高网络的表达能力，泛化性好？但是容易过拟合、梯度消失/爆炸

- BP的实际问题
  - 需要数据带标签（大部分数据没有标签）
  - 偏导数计算缓慢（对于层数多的...）
  - 误差累积、梯度消失
  - 很容易陷入局部最优值
- Deep Learning（对于带标签数据少的问题，现在的解决）
  - 训练第一层 without labels (unsupervised)，自编码器等，通过重构输入数据，第一层学会了如何表示数据的基本特征
  - 在第一层训练完毕后，冻结其参数。这意味着在接下来的训练过程中，这些参数不会再更新
  - 用第一层的输出作为输入，开始训练第二层。同样，这一层的训练可以使用无监督方法，逐层进行之
  - 将最后一层的输出作为输入、训练一个监督层（之前的仍然冻结）
  - 解冻，使用带标签数据全模型微调
- 现实中尝试"跳出"局部极小的方法（西瓜书P107）：
  - 以多组不同参数值初始化多个神经网络，最后取误差最小的解作为最终参数
  - 使用"模拟退火"，每一步以一定概率接受比当前解更差的结果；接受"次优解"的概率随训练推进而降低、保证稳定性
  - 使用随机梯度下降，标准下降法精确计算梯度、随机下降（只随机使用一个小batch、可以在新数据到达时进行参数更新，迭代次数多但计算快）即使陷入局部极小点、梯度仍可能不为0
  - 遗传算法

> 问题：多层全连接网络的问题
>
> - 最大问题在于全连接层的参数太多（按层是相乘关系），容易出现参数量远大于数据集
> - 对于图像数据，忽略了pixel之间的关系（->卷积）

## 卷积神经网络 CNN

- 为什么处理图像等不用仿射变换：
  - 图像等数据具有内在结构、存储为多维数组
  - 某些轴的顺序很重要（如图像宽高、时间轴）
  - 一个轴用于访问数据不同视图（RGB通道、左右声道）->仿射变换视为一维向量、忽略了轴顺序和局部相关性
- softmax，特殊的激活函数，a fancy normalizer，生成一个离散概率分布
- cross-entropy loss，对于分类任务效果好
- dropout：随机 ignore 一些 activations。有效减少过拟合、提升泛化性，增加训练时间、收敛可能变慢
- batch normalization：通常放在 activation layers之前，减少 bad weight initialization 的影响；减少梯度消失问题、加快训练速度；增加计算开销、依赖小批量数据

## 知名网络与其他结构

- DenseNet：每一层与其所有前面层连接、特征复用高效传递
- ResNet：跳跃连接
  - 两者都增加计算复杂度、但减少了梯度消失问题
- RNN
- KAN

> 讨论深度神经网络的优点和局限 under the local-to-global assumption（由细节到整体） 优点
>
> - 层次化特征学习、逐层从简单的局部特征到复杂的全局特征（模拟了人类视觉系统，但是也有学说认为人类是先全局再局部？），捕捉到数据的不同抽象层次
> - 层与层之间特征可以重复利用
> - 不需要手工设计特征，网络可以自动学习到数据的最优特征表示
> - 适应性强 缺点
> - 训练复杂度高、数据需求量大、可解释性差、梯度消失/爆炸问题（参考之前BP算法的问题）

### Transformer

- attention机制，为输入的每个部分分配不同的权重，允许模型专注于更重要的部分
- 首先对于给定的 query、key、value计算注意力权重（首先计算key query相似度得分，如点积） -> 归一化 -> 对值向量加权求和
- 优点：灵活性、处理长距离依赖（捕捉序列中远距离元素关系）、相比于 RNN 可以并行处理
- 正余弦位置编码 -> 补充序列信息、信息密度高

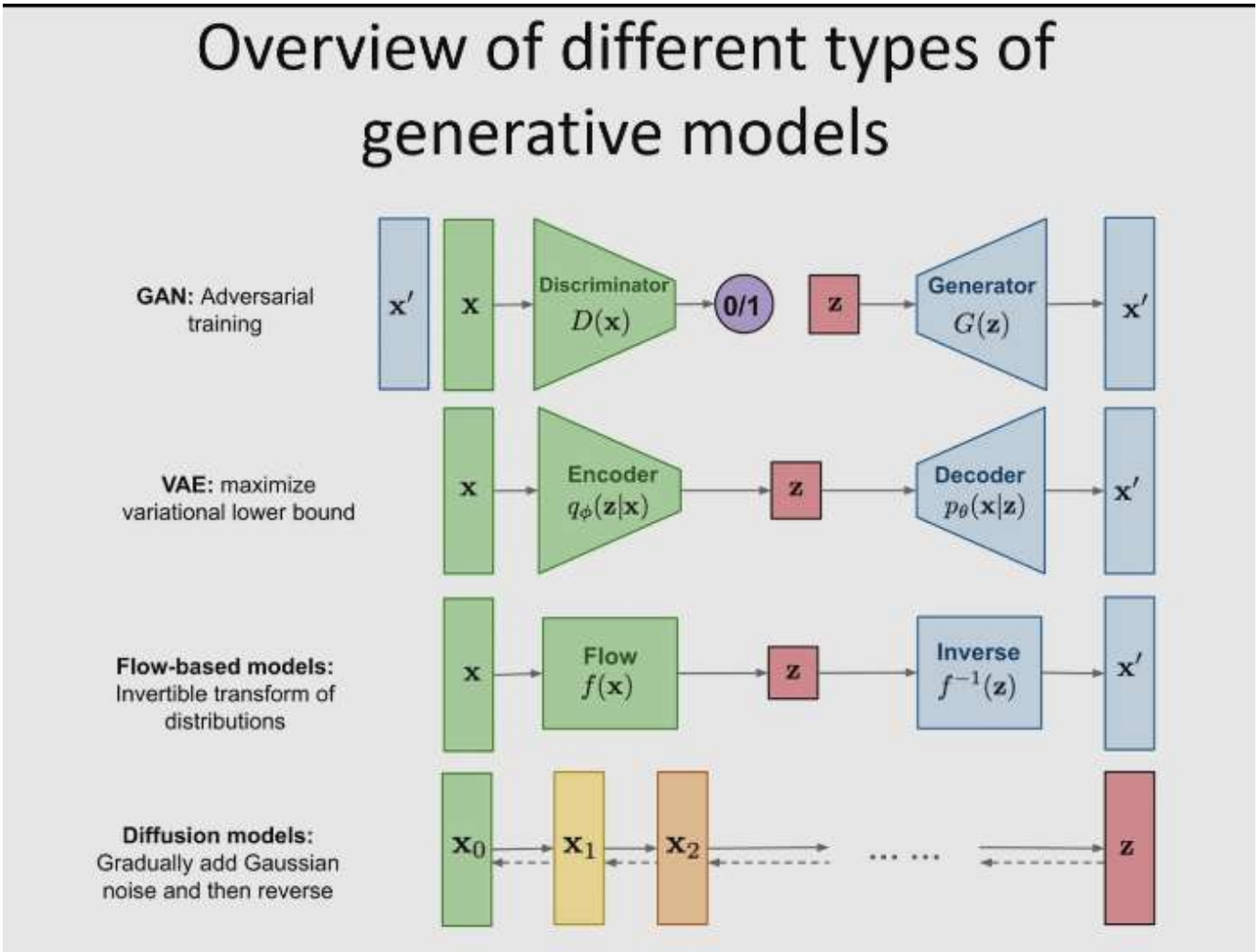
## 图神经网络

- 从 CNN 到 GCN
  - CNN：邻近信息加权求和汇聚到中心
  - GCN：邻居节点信息加权求和到点（每个节点通过聚合自身及其邻居节点的特征来更新自身特征表示）
    - 特征聚合 -> 线性变换 -> 激活函数，视为一层
  - CNN和GCN都使用卷积操作来提取特征，CNN在规则数据...GCN在不规则的...;CNN邻域固定、卷积核形状固定，GCN邻域不规则、卷积核形状不固定，两者都共享卷积核参数，从而降低模型复杂度和计算量
- GNN+Attention：根据邻居节点的重要性动态分配权重（而普通注意力在所有输入之间分配权重）
- 应用：node classification, graph classification, link prediction

## 深度生成模型

- 本质还是采样简单分布z -> 数据分布X，之前传统机器学习中间是线性函数，现在引入深度神经网络
- VAE
  - base: Auto Encoder，特征提取-恢复到原空间

    - 引入近似分布 Q(z) 近似后验分布 P(z|x)
    - 通过最小化 Q(z) 和 P(z|x) 之间的KL散度达到逼近的目的，优化目标可以表示为 logP(X)-D_KL(Q(z)||P(z|X))
    - loss函数：重构（网络输出和原始数据的差异）+KL散度（近似后验分布 Q(z|X)和先验分布P(z)之间的差异
        - z是采样的，直接对采样过程反向传播不可能，使用重新参数化技巧，将z的采样过程表示为确定性操作+噪声，然后对$\mu$、$\sigma$进行反向传播
    - E-step，计算Q
    - 测试生成结果通常依赖人工、没有好的量化措施
- Contional VAE
    - 将标签信息输入encoder-decoder中间?
    - P(x|z)和Q(z|x) replace with P(x|z,Y)和Q(z|X,Y)
- GAN
    - min-max过程，
- DM




## 6.因果发现

- association：有关（统计上不独立）
- correlation：存在线性关系

- 因果
- do x则指向x的边消失