希文 23春 模式识别与机器学习 23年6月6日

分类器评价指标

测试错误率作为真实错误率的可靠估计是有条件的. 最重要的条件是: 测试样本与训练样本独立, 测试样本与未来样本独立同分布, 测试集样本数足够大. 此时测试错误率是对真实错误率的无偏估计，但估计的置信范围取决于测试样本数目，有可能会很大.

解决训练样本集与测试样本集矛盾: n-fold 交叉验证; 留一法交叉验证 (LOOCV)(交叉验证的极端情况，样本少时被广泛应用)，自举法与 B.632 估计
若有病D^+，检测阳性T^+，则对应 TP(真阳性)，检测阴性T^-，对应 FN 假阴性.
若无病D^-，检测阳性T^+，则对应 FP(假阳性)，检测阴性T^-，对应 TN 真阴性.

混淆矩阵对应关系搞对

Sensitivity 灵敏度: $P(T^+ \mid D^+) = TP/(TP+FN)$ 为(1-犯第二类错误概率)
Specificity 特异度: $P(T^- \mid D^-) = TN/(TN+FP)$. 为(1-犯第一类错误概率)
Prevalence 患病率: 人群中患病的概率 $D^+/(D^+ + D^-)$
Discovery rate 发现率: $P(D^+ \mid T^+) = TP/(TP+FP)$
False discovery rate 误发现率: $P(D^- \mid T^+) = FP/(TP+FP)$
Accuracy 精度/准确率: $(TP+TN)/(D^+ + D^-)$
第一类错误: 假阳性; 第二类错误: 假阴性
计算采用**贝叶斯公式**, sensitivity Sn, Specificity Sp (等于 1- fp)

$$P(D^+ \mid T^-) = \frac{P(T^- \mid D^+)P(D^+)}{P(T^-)} = \frac{(1-\mathrm{Sn})P(D^+)}{(1-Sn)P(D^+) + \mathrm{Sp}\cdot(1-P(D^+))}$$

$$P(D^+ \mid T^+) = \frac{P(T^+ \mid D^+)P(D^+)}{P(T^+)} = \frac{Sn\cdot P(D^+)}{Sn\cdot P(D^+) + (1-\mathrm{Sp})(1-P(D^+))}$$

两次患病: $P(A \mid (B\cap C)) = \frac{P((B\cap C)|A)\times P(A)}{P((B\cap C)|A)\times P(A)+P((B\cap C)|A^c)\times P(A^c)}$

Other Jargons: Precision: $\frac{TP}{TP+FP}$; Recall: $\frac{TP}{TP+FN}$; Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$

ROC: Receiver Operating Characteristic. 横坐标为 False Positive Rate, 纵坐标为 True Positive Rate. AUC: Area Under Curve, higher the better, 0.5: random guess.

线性学习机器

线性判别函数: $y = \mathrm{sgn}\left(\sum_{i=1}^d w_i x_i + w_0\right)$

在 X 空间, 类均值向量: $m_i = \frac{1}{N_i}\sum_{x_j\in x_i} x_j, i=1,2$类内离散度矩阵: $\boldsymbol{S}_i = \sum_{x_j\in\mathcal{X}_i}(\boldsymbol{x}_j-\boldsymbol{m}_i)(\boldsymbol{x}_j-\boldsymbol{m}_i)^T, i=1,2$.总类内离散度矩阵$S_w = S_1 + S_2$.
类间离散度矩阵: $S_b = (m_1-m_2)(m_1-m_2)^T$
在 Y 空间, 类均值: $\widetilde{m}_i = \frac{1}{N_i}\sum_{y_j\in y_i} y_j, i=1,2$.类内离散度: $\tilde{S}_i = \sum_{y_j\in y_i}(y_j-\widetilde{m}_i)(y_j-\widetilde{m}_i)^T, i=1,2$.总类内离散度: $\tilde{S}_w = \tilde{S}_1 + \tilde{S}_2$
类间离散度矩阵: $\tilde{S}_b = (\widetilde{m}_1-\widetilde{m}_2)^2$

Fisher 准则: $\max_w J_F(w) = \frac{(\widetilde{m_1}-\widetilde{m_2})^2}{\widetilde{S_1}+\widetilde{S_2}}$, where $y_i = w^T x_i$. 代价函数: $J_F(w) = \frac{J_2}{J_1} = \frac{w^T S_b w}{w^T S_w w}$. 由于只与方向有关, 构造$w^T S_w w = 1$, 则将问题转化为$\min_w - w^T S_b w$, s.t. $w^T S_w w = 1$. 定义$L(w) = -w^T S_b w + \lambda(w^T S_w w - 1)$, 求导为 0 可得$S_b w = \lambda S_w w$, 即w为矩阵$S_w^{-1}S_b$的特征向量.则有$w = \frac{1}{\lambda}S_w^{-1}S_b w = \frac{1}{\lambda}S_w^{-1}(\mu_1-\mu_0)(\mu_1-\mu_0)^T w$, 只考虑方向, 则有$\boldsymbol{w}^* \propto \boldsymbol{S}_w^{-1}(\boldsymbol{m}_1-\boldsymbol{m}_2)$.还需确定决策的分界点, 如$w_0 = -\frac{1}{2}(\widetilde{m}_1+\widetilde{m}_2)$或$w_0 = -\widetilde{m}$

Perceptron: $y = \mathrm{sgn}\left(\sum_{i=1}^d w_i x_i + w_0\right)$.若样本线性不可分, 则可以容忍错误, 使错误尽量小(比如强制收敛, MSE), 或寻求非线性方法(比如神经网络, SVM); 若样本线性可分时多解, 可寻求最优分类器, 如 SVM. 若样本为多类, 则可使用多类方法, 或用两类分类器完成多类分类.

线性回归:

$$\min_w E(\boldsymbol{w}) = \frac{1}{N}\sum_{j=1}^N (\hat{y}_j - y_j)^2 = \frac{1}{N}\sum_{j=1}^N (f(\boldsymbol{x}_j)-y_j)^2 = \frac{1}{N}\|\boldsymbol{Xw}-\boldsymbol{y}\|^2 = \frac{1}{N}(\boldsymbol{Xw}-\boldsymbol{y})^T(\boldsymbol{Xw}-\boldsymbol{y})$$

解析解: $\nabla E(\boldsymbol{w}) = \frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{2}{N}\boldsymbol{X}^T(\boldsymbol{Xw}-\boldsymbol{y}) = 0$, 有$\boldsymbol{X}^T\boldsymbol{Xw} = \boldsymbol{X}^T\boldsymbol{y}$

若可逆, 则$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$. 可逆的条件是特征间线性独立. 若特征间不是线性独立, 仍可以计算伪逆, 但解不唯一. 解决方案: 通过特征选择或变换去除冗余, 或通过引入其他准则对解加以限制(如 SVD 或正则化)

评价指标: $R^2 = 1 - \frac{\sum_{i=1}^N(y_i-\hat{y}_i)^2}{\sum_{i=1}^N(y_i-\bar{y})^2}$. 在 OLS 模型中, $R^2 = \frac{\sum_{i=1}^N(\hat{y}_i-\bar{y})^2}{\sum_{i=1}^N(y_i-\bar{y})^2}$. $R^2 = 1$ indicates perfect regression, while $R^2 = 0$ indicates baseline model. 意义: percentage of dependent variable variations that the linear model explains.

Pearson Correlation Coefficient: $r_{XY} = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{var}(X)}\cdot\sqrt{\mathrm{var}(Y)}} = \frac{\sum(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum(X_i-\bar{X})^2}\cdot\sqrt{\sum(Y_i-\bar{Y})^2}}$

MSE 准则: 求α 使得$\alpha^T y_i > 0, i=1,\cdots,N$. 当样本集线性不可分时, 使尽可能多的样本满足不等式. 为方便求解, 为每个样本引入b_i,
并令$\alpha^T y_i = b_i > 0, j=1,\cdots,N$. 这样就可用最小二乘法解决.
优化问题: $\boldsymbol{\alpha}^*: \min_{\boldsymbol{\alpha}} J_S(\boldsymbol{\alpha})$, where $J_S(\boldsymbol{\alpha}) = \|\mathbf{Y}\boldsymbol{\alpha}-\mathbf{b}\|^2 = \sum_{i=1}^N(\boldsymbol{\alpha}^T\mathbf{y}_i - b_i)^2$.
问题: 如何给定$b = [b_1, b_2, \cdots, b_N]^T$?
可以证明, 如果b选为$b_i = \begin{cases} N/N_1, \text{if } y_i\in\omega_1 \\ N/N_2, \text{if } y_i\in\omega_2\end{cases}$, 则 MSE 解等价于$w_0 = -\hat{m}$的 Fisher Linear Discrimination 解.
如果$\boldsymbol{b}$选为$b_i = 1, i=1,\cdots,N$, 则当$N\to\infty$时, MSE 解以最小均方误差最优逼近贝叶斯判别函数$g_0(\boldsymbol{x}) = P(\omega_1\mid\boldsymbol{x}) - P(\omega_2\mid\boldsymbol{x})$,
使 $e^2 = \int[\boldsymbol{\alpha}^T\boldsymbol{y} - g_0(\boldsymbol{x})]^2 p(\boldsymbol{x})d(\boldsymbol{x})$ 最小

Logistic Function: $P(y\mid x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$. Odds 几率 $\frac{P(y|x)}{1-P(y|x)} = e^{\alpha+\beta x}$

$h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$, $\theta(s) = \frac{e^s}{1+e^s}$

回归目标: 最大化似然函数. 参数为$\boldsymbol{w}$ 的模型在数据$\{(x_1,y_1),\ldots,(x_N,y_N)\}, x_j\in R^{d+1}, y_j\in\{-1,1\}$上的似然函数为$L(\boldsymbol{w}) = \prod_{j=1}^N P(y_j\mid\boldsymbol{x}_j) = \prod_{j=1}^N \theta(y_j\boldsymbol{w}^T\boldsymbol{x}_j)$

最大化似然函数, 等价于最小化$\frac{1}{N}\sum_{j=1}^N \ln(1+e^{-y_j w^T x_j})$, 也即:

$\min E(\boldsymbol{w}) = -\frac{1}{N}\ln(L(\boldsymbol{w})) = -\frac{1}{N}\ln\left(\prod_{j=1}^N \theta(y_j\boldsymbol{w}^T\boldsymbol{x}_j)\right)$

梯度下降求解: $\nabla E = -\frac{1}{N}\sum_{j=1}^N \frac{y_j x_j}{1+e^{y_j w(k)^T x_j}}$

人工神经网络

Softmax 函数: $P(y=j\mid\boldsymbol{x}) = \frac{e^{w_j\cdot x}}{\sum_{k=1}^K e^{w_k\cdot x}}$. 交叉熵: $l(\mathbf{y},\hat{\mathbf{y}}) = -\sum_{j=1}^q y_j\log\hat{y}_j$

有 $l(\mathbf{y},\hat{\mathbf{y}}) = -\sum_{j=1}^q y_j\log\frac{\exp(o_j)}{\sum_{k=1}^q\exp(o_k)} = \log\sum_{k=1}^q\exp(o_k) - \sum_{j=1}^q y_j o_j$

QDA: $g_i(\boldsymbol{x}) = k_i^2 - (\boldsymbol{x}-\widehat{\boldsymbol{m}}_i)^T\widehat{\Sigma}_i^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{m}}_i), i=1,\cdots,C$,
decision: $\mathrm{class}(\boldsymbol{x}) = \mathrm{argmax}_i\, g_i(\boldsymbol{x})$

BP 算法训练中可能遇到的问题: 收敛慢, 学习曲线震荡, 过拟合:推广能力差.
可能的原因: 网络结构或激活函数问题, 训练样本不充分或不合适, 初值不合适, 学习率不合适, 缺乏恰当的预处理 (比如归一化), 需要更好的训练策略, 运气问题. **策略**: 激活函数: Sigmoid, tanh, ReLU; 特征尺度调整或归一化 (有效值范围, 相对重要性等); 目标值调整; 引入"伪样本"增大训练样本集, 如加噪声或者数据增强; 隐层节点数, 层数, 网络剪枝.
引入动量(Momentum), Weight Decay, 引入提示输出(Hints), 终止条件.

Hopfield Network: An array of binary threshold units fully connected with symmetric weights. Each binary "configuration" of the network has an energy.
The dynamic procedure converges to a "memory". (Associative memory)
How does Hopfield Net capture relations? Hebb-rule training: finding associations among pixels.

Boltzmann Machines consist of Visible Nodes and Hidden Nodes, without the output layer. 设 Visible Nodes x 中的 Offsets 为 c, Hidden Nodes h 中的 Offsets 为 b, x 之间的权重为 U, h 之间的权重为 V, x 与 h 之间的权重为 W, 则 Energy Function is defined as:

$E(\boldsymbol{x},\boldsymbol{h}) = -\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{h} - \boldsymbol{h}^T\boldsymbol{W}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{U}\boldsymbol{x} - \boldsymbol{h}^T\boldsymbol{V}\boldsymbol{h}$

A fuly-connected Boltzmann machine can represent complex probabilistic relations, but parameter estimation is hard.

Restricted Boltzmann Machines (RBM): Restricted to connections between visible and hidden nodes.Energy function: $E(\boldsymbol{x},\boldsymbol{h}) = -\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{h} - \boldsymbol{h}^T\boldsymbol{W}\boldsymbol{x}$

Joint probability: $p(\boldsymbol{x},\boldsymbol{h}) = e^{-E(\boldsymbol{x},\boldsymbol{h})}/Z$ 其中$Z = \sum_{\boldsymbol{x},\boldsymbol{h}} e^{-E(\boldsymbol{x},\boldsymbol{h})}$
MLE: $P(\boldsymbol{x};\theta) = \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{x},\boldsymbol{h})}/Z$,
则$\frac{\partial(-\log P(x))}{\partial\boldsymbol{\theta}} = \mathbb{E}_{P(\boldsymbol{h}|x)}\frac{\partial E(\boldsymbol{x},\boldsymbol{h})}{\partial\boldsymbol{\theta}} - \mathbb{E}_{P(\boldsymbol{x},\boldsymbol{h})}\frac{\partial E(\boldsymbol{x},\boldsymbol{h})}{\partial\boldsymbol{\theta}}$.其中$\mathbb{E}_{P(\boldsymbol{x},\boldsymbol{h})}$未知, 需要估计.

$p(\boldsymbol{h}\mid\boldsymbol{x}) = \prod_j \mathrm{sigm}\left((2h_j-1)(b_j+\boldsymbol{W}_j\cdot\boldsymbol{x})\right)$

$p(\boldsymbol{x}\mid\boldsymbol{h}) = \prod_k \mathrm{sigm}\left((2x_k-1)(c_k+\boldsymbol{hW}_{\cdot k})\right)$

1980 年代三种主要类型的神经网络. 前馈型神经网络 Feedforward NN (代表性方法: 多层感知器); 反馈型神经网络 Feedback NN (代表性方法 Hopfield NN); 竞争学习神经网络 Competitive Learning NN (代表性方法: 自组织映射 Self-organizing map)

近邻法

分段线性判别函数: 把各类划分为若干子类, 以子类中心作为类别代表点, 考查新样本到各代表点的距离. 极端情况:近邻法.
K 近邻法:找出 x 的前 k 个近邻,看其中多数属于哪一类,就把 x 分到哪一类. K 越大, 分界线越平滑.
KNN 常见问题: 存储量和计算量; 票数接近时风险较大, 有噪声时风险加大; 样本无穷多时性能优异, 有限样本下性能如何? 改进: 减少计算量和存储量; 引入拒绝机制; 根据实际问题修正投票方式.
近邻法在计算上的问题: 需存储所有训练样本; 新样本需与每个样本做比较.快速算法基本思想: 把样本集分级分成多个子集 (树状结构), 每个子集 (节点)可用较少几个样本代表; 通过将新样本与各节点比较排除大量候选样本; 只有最后的节点 (子集)中逐个样本比较,找出近邻.
分枝定界算法 (Branch-Bound Algorithm): 把已知样本划分成多个子集, 形成一个树状结构, 每个节点是一个子集, 每个子集用较少的几个量代表; 新样本按顺序与各个节点进行比较来排除不可能包含近邻的子集, 最后只在少数节点上与每个样本进行比较
剪辑近邻法:处在两类交界处或分布重合区的样本可能误导决策。应将它们从样本集中去掉。基本思路:考查样本是否为可能的误导样本,若是则从样本集中去掉(剪辑).考查方法是通过试分类, 认为错分样本为误导样本.
压缩近邻法: 主要用以减少存储量. 将训练样本分为X_S和X_G两个集合, 开始时X_S中只有一个样本, X_G中为其余样本. 考查X_G中每个样本, 若用X_S可正确分类则保留, 否则移入X_S. 最后用X_S作分类的样本集.
原型近邻法: 用若干个原型 (prototype)样本来代表训练数据. 通常原型的数量远远小于训练样本数 (如等于样本数则为 1-近邻法).
定义原型点: 利用 k-means 聚类. 对于第 i 类样本, 进行聚类分析, 获得 R 个原型 (该类样本密度高的点). 定义这 R 个原型的类别为 i; 总共获得 CR 个原型样本 (C 为类别数).
对于待分类的样本 x, 我们将其类别定为距离它最近的原型的标签.
原型近邻法的优点有: 模型测试时计算开销小; 节省内存. 没有: 模型训练时计算开销小; 可视化效果好; 能更好捕捉数据分布的细节信息.
可做拒绝决策的近邻法: 从简单多数变为绝对多数. 拒绝决策同样可引入改进的近邻法中, 比如剪辑近邻法.
高维情况下需要更多的训练样本.当特征维度较高时, "最近邻"可能相差很远.使用欧氏距离通常会对每一维特征做标准化.
按距离加权的 KNN(LOESS): 距离近的邻居加权大, 距离远的邻居加权小. 一种常用的权值函数$w^{(i)} = \exp\left[-\frac{(x^{(i)}-x)^2}{2\tau^2}\right]$. τ 为带宽.训练样本少, 带宽应该取得大一些; 模型回归系数θ^T与x 取值有关.

决策树

节点的分裂需要定义一个衡量指标.

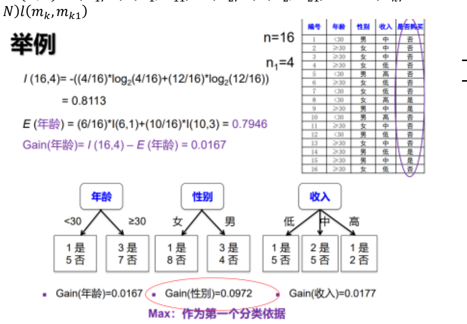
Entropy (used in ID3 and C4.5):$H(\mathcal{D}) = -\sum_{k=1}^K\frac{|\mathcal{C}_k|}{|\mathcal{D}|}\log\frac{|\mathcal{C}_k|}{|\mathcal{D}|}$

Gini index (used in CART): $Gini(\mathcal{D}) = 1 - \sum_{k=1}^K\left(\frac{|\mathcal{C}_k|}{|\mathcal{D}|}\right)^2$

当 Entropy 最大为 1 的时候, 是分类效果最差的状态, 当它最小为 0 的时候, 是完全分类的状态。因为熵等于零是理想状态, 一般实际情况下, 熵介于 0 和 1 之间.在分裂时, 我们衡量父亲的熵与两个儿子的熵的差, 即衡量分裂后是否能够减少熵$H(\mathcal{D}) - (H(\mathcal{D}_1)+H(\mathcal{D}_2))/2$
或者定义**信息增益**: $\mathrm{Gain}(N,a) = I(N,N^{(1)}) - E(N,a)$
上层节点样本数 N, 第一类样本数$N^{(1)}$, 信息熵$I(N,N^{(1)})$. 若按属性 a 分类, 则下层节点, 在属性 a 上有 k 个取值, 取值 i 下有m_i个样本, 其中m_{i1}个属于第一类, 熵$E(N,a)$

$I(N,N^{(1)}) = -\left((N^{(1)}/N)\log_2(N^{(1)}/N) + (1-N^{(1)}/N)\log_2(1-N^{(1)}/N)\right)$

$E(N,a) = (m_1/N)I(m_1,m_{11}) + (m_2/N)I(m_2,m_{21}) + \cdots + (m_k/N)I(m_k,m_{k1})$



编号	年龄	性别	收入	是否购买
1	<30	男	中	否
2	≥30	女	中	否
3	≥30	女	中	否
4	≥30	女	低	否
5	<30	男	高	否
6	≥30	女	低	否
7	<30	女	低	否
8	<30	女	高	是
9	≥30	男	中	是
10	<30	男	高	否
11	≥30	女	中	否
12	<30	男	低	否
13	≥30	女	中	否
14	≥30	男	低	是
15	≥30	男	中	是
16	≥30	女	低	否

ID3 存在一个问题, 那就是越细小的分割分类错误率越小. 所以为了避免分割太细, **C4.5** 进行了改进, 优化项要除以分割太细的代价, 比值叫做信息增益率, 分割太细分母增加, 信息增益率会降低。除此之外, 其他的原理和 ID3 相同.

$\mathrm{Gain\_ratio}(N,a) = \frac{\mathrm{Gain}(N,a)}{-\sum_{v=1}^k P_v\log P_v}$. 其中$P_v = \frac{N_v}{N}; v=1,\ldots,k$

CART 与 C4.5 算法的最大相异之处是其在每一个节点上都 是采用二分法, 也就是一次只能够有两个子节点, C4.5 则 在每一个节点上可以产生不同数量的分枝. CART 和 ID3 一样, 存在偏向细小分割, 即过度学习 (过度拟合的问题), 为了解决, 对特别长的树进行剪枝处理, 直接剪掉。
如果正负两类样本从参数完全相同的多维正态分布中抽样产生, 把其中一部分用作训练集, 一部分用作测试集, 利用决策树来构建分类器, 则在训练集上可以得到高正确率, 在测试数据上不能得到高正确率.因为过拟合.
剪枝 (Pruning): 避免过学习. 先剪枝 (prepruning): 数据划分法: 利用训练集决定节点划分, 根据测试集的分类错误率决定是否停止; 阈值法: 预先设定信息增益阈值; 信息增益的统计显著性分析. 后剪枝(Postpruning): 减少分类错误修剪法: 用训练集生成完整的树, 在独立剪枝集上减少分类错误的修剪法; 最小代价与复杂性折中: 合并分枝后错误率增加和复杂度减少折中考虑; 最小描述长度准则: 最简单的树就是最好的树.
集成学习:个体学习器之间不存在强依赖关系,可以并行生成学习器:Bagging, Random Forest;个体学习器之间存在强依赖关系, 串行生成学习器:Boosting.
Bagging (Bootstrap Aggregating): 对训练数据 D **有放回抽样**生成 m 个数据集D_i, 在每一个数据集D_i上训练分类器h_i, 对各个分类器的结果集成投票. 防止过拟合, 减小模型方差.
Random Forests: 利用 bootstrapping 的办法 (有放回抽样), 训练多棵树 (每次从 N 个训练样本中有放回地随机抽取 N 个样本作为当前训练集). 每次随机抽取 k (k<p)个特征作为当前节点下决策的备选特征, 从中选择特征进行划分 (split).最后对每一颗树的预测结果进行汇总投票.
随机选取特征的好处: 用越少的特征去构造决策树, 决策树就越不容易过拟合; 在 bootstrap samples 中, 如果一个特征非常重要, 则它会在不同的 sample 中都出现 (或者说出现的频率比较高), 则这样就会带来 Correlation.而随机森林的原则是避免相关性,因此随机选取 feature..
为什么效果好:每决策树的决策面都是平行坐标轴的立方体, 多颗决策树组合的结果可形成复杂的决策面.随机森林获得的每棵树之间结构不同, 参数不同.
KNN 中需要对各维度特征进行 Normalization, 使用决策树时不需要.
Boosting: 通过迭代对学习器的输入输出进行加权组合, additive training, 训练新学习器拟合残差.开始时所有训练样本同样权重, 每一轮学习一个"弱"学习器, 根据当前"弱"学习器的组合预测结果进行调整, 对预测偏差给予更多关注. 直到"弱"学习器数目达到预先制定的数目 k,最终将 k 个学习器的结果加权组合.
AdaBoost: Adaptive Boosting. 计算当前轮次对应的错误率与正确率, 调整正确样本与错误样本的权重, 使得下一轮分类器的训练样本一半来自分对的, 一半来自分错的. 训练过程是串行的. 计算错误率e_m, 权重$\alpha_t = \log((1-\varepsilon_t)/\varepsilon_t)$, 更新$D(x_i) = D(x_i)\cdot\exp(\alpha_t\cdot I[y_i\neq h_t(x_i)])$
Gradient Boosting: Gradient Descent + Boosting. 输入: $\{(x_i,y_i)\}_{i=1}^N$; 损失函数$l(y,F(x))$; 迭代次数k_{max}.
算法流程: 初始化$F_0(x) = \arg\min_\eta\sum_{i=1}^N l(y_i,\eta)$. 循环 $k=1$ _ k_{max}: 计算梯度$r_{ik} = -\left[\frac{\partial l(y_i,F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{k-1}(x)}$; 用数据集$\{(x_i,r_{ik})\}_{i=1}^N$训练新学习器$f_k(x)$; 计算更新权重: $\eta_k = \arg\min_\eta\sum_{i=1}^N L(y_i,F_{k-1}(x_i)+\eta f_k(x_i))$; 更新模型: $F_k(x) = F_{k-1}(x) + \eta_k f_k(x)$
XGBoost: Extreme Gradient Boosting. 改进 1: 目标函数引入正则项 (防止过拟合)); 改进 2: 引入导数项简化目标函数.随机森林, Gradient Boosting 和 XGBoost 都是由多学习器集成学习得到最终结果, 适用于分类和回归. 随机森林对样本和特征降采样, 不易过拟合, 多模型表决/均值, 对异常值不敏感, 未设计稀疏数据处理策略; Gradient Boosting 使用全部样本, 容易过拟合, 模型加权求和, 对异常值敏感, 未设计稀疏数据处理策略; XGBoost 对样本和特征降采样, 正则项防止过拟合, 模型加权求和, 对异常值敏感, 适用于稀疏数据.**Boosting 方法的优点**: 快速; 简单; 只有一个参数; 灵活: 可以与任何分类器使用, 只需要比随机好的"弱"分类器. 缺点: "弱"分类器不能太复杂 (容易过拟合); 对噪声敏感, 例如会给标记错误的样本过大的权重.**神经网络防止过拟合**: Dropout. 每次训练时随机把部分节点参数置零, 相当于利用很多不同结构网络的结果的集合.

支持向量机

若向量集合被超平面$f(\boldsymbol{x}) = \mathrm{sgn}((\boldsymbol{w}\cdot\boldsymbol{x})+b)$没有错误地分开, 且离超平面最近的向量与超平面之间的距离 (称作间隔 margin) 是最大的, 则我们说这个向量集合被这个**最优超平面** (或最大间隔超平面) 分开
要留下一定的"裕度", 故要把尺度固定下来: $y_i(\boldsymbol{w}\cdot\boldsymbol{x_i}+b)\geq 1$
点到平面距离$\frac{|\mathbf{w}\cdot\mathbf{x_0}+b|}{|\mathbf{w}|}$. 故 margin 为$2/\|\mathbf{w}\|$
优化问题$\min_{w,b}\frac{1}{2}\|\boldsymbol{w}\|_2^2$, s.t.$y_i(w\cdot x_i+b)\geq 1, 1\leq i\leq n$.
拉格朗日函数 $L(W,b,\alpha) = \frac{1}{2}|W|^2 - \sum_{i=1}^n\alpha_i\left[y_i(X_i^T W+b)-1\right]$. 也即 $\min_{W,b}\max_\alpha L(W,b,\alpha)$. 对偶问题: $\max_\alpha\min_{W,b} L(W,b,\alpha)$
求导: $\nabla_W L(W,b,\alpha) = W - \sum_{i=1}^n\alpha_i y_i X_i = 0 \Longrightarrow W = \sum_{i=1}^n\alpha_i y_i X_i$
$\nabla_b L(W,b,\alpha) = -\sum_{i=1}^n\alpha_i y_i = 0 \Longrightarrow \sum_{i=1}^n\alpha_i y_i = 0$.
代入后结果为 $\sum_{i=1}^n\alpha_i - \frac{1}{2}\sum_{i,j=1}^n y_i y_j\alpha_i\alpha_j\mathbf{x}_i^T\mathbf{x}_j$.
取极大对应取负后极小: $\min_\alpha\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\alpha_i\alpha_j y_i y_j X_i^T X_j - \sum_{i=1}^n\alpha_i$
解得$\widehat{W} = \sum_{i=1}^n\hat{\alpha}_i y_i X_i$, $\hat{b} = y_j - \sum_{i=1}^n\hat{\alpha}_i y_i X_j^T X_i$.
判别函数$f(X) = \mathrm{sign}\left(\sum_{i=1}^n\hat{\alpha}_i y_i X^T X_i + \hat{b}\right)$
软间隔: 引入惩罚参数 $\min_{w,b,\xi}\frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n\xi_i$, s.t. $y_i(w\cdot x_i+b)\geq 1-\xi_i$, $\xi_i\geq 0, 1\leq i\leq n$与最优超平面类似, 广义最优超平面满足下列 KKT 条件: $\alpha_i^0\{[(\boldsymbol{x}_i\cdot\boldsymbol{w}_0)+b_0]y_i - 1 + \xi_i\} = 0, i=1,\cdots,l$. 只有部分样本的$\alpha_i$不为 0, 是不等式$y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)\geq 1-\xi_i$
线性不可分情况下的支持向量包括两部分: 正确分类且离分类面最近的样本, 它们的$0<\alpha_i<C$; 错误分类的样本, 它们的$\alpha_i = C$

推广到非线性

挑战 1: 能否对付在特征空间中的运算 (如升到高维是否会出现维度灾难). 实际上, 变换空间中的内积运算仍然对应着原空间中某种运算,. 核函数必须是某个空间中的内积. 在核函数给定的情况下, 可以利用解线性问题的方法求解非线性问题的支持向量机, 此过程是隐式地在特征空间中进行的.
RBF 核: $K(\mathbf{x},\mathbf{x}_i) = \exp\left\{-\frac{|\mathbf{x}-\mathbf{x}_i|^2}{\sigma^2}\right\}$线性核函数的 SVM 一定会得到线性分类面.
支持向量回归(SVR).原问题:用函数$f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x}+b$拟合样本.如果所有样本均能在精度下$\varepsilon$被拟合,即$\begin{cases} y_i - \mathbf{w}\cdot\mathbf{x}_i - b\leq\varepsilon \\ \mathbf{w}\cdot\mathbf{x}_i + b - y_i\leq\varepsilon\end{cases}$,最小化$\frac{1}{2}\|\mathbf{w}\|^2$以获得最优回归.
引入松弛变量$\xi_i\geq 0, \xi_i^*\geq 0$, 原问题变为$\min\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^n(\xi_i+\xi_i^*)$,
s.t. $\begin{cases} y_i - \mathbf{w}\cdot\mathbf{x}_i - b\leq\varepsilon+\xi_i^* \\ \mathbf{w}\cdot\mathbf{x}_i + b - y_i\leq\varepsilon+\xi_i\end{cases}$. 则有$w = \sum_{i=1}^l(\alpha_i^*-\alpha_i)x_i$.
回归函数$f(\mathbf{x}) = (\mathbf{w}\cdot\mathbf{x}) + b = \sum_{i=1}^l(\alpha_i^*-\alpha_i)(\mathbf{x}_i\cdot\mathbf{x}) + b^*$.
支持向量: 只有$\alpha_i^{(*)} = C$的样本落在"$\mathcal{E}$不敏感管道"外 (类比 SVM 分类中的错分样本). 满足$\alpha_i^{(*)}\in(0,C)$的样本有$\xi_i^{(*)} = 0$且$|y_i - f(x_i)| = \varepsilon$ (类比 SVM 分类中的边界样本)

统计学习理论

为什么最大间隔是最优? 大间隔->低 VC 维->低复杂度->高推广能力.
经验风险 $R_{emp}(\alpha) = \frac{1}{l}\sum_{i=1}^l Q(z_i,\alpha)$. 经验风险最小化 Empirical Risk Minimization 原则: 用最小化$Q(z,\alpha_l)$的解近似$Q(z,\alpha_0)$
初心$\min R(\alpha) = \int Q(z,\alpha)dF(z)$ 实际行动$\min R_{emp}(\alpha) = \frac{1}{l}\sum_{i=1}^l Q(z_i,\alpha)$
一个指示函数集$Q(z,\alpha), \alpha\in\Lambda$ 的 VC 维, 是能被集合中的函数以所有可能的2^h种方式分成两类的向量$Z_1,\cdots,Z_h$ 的最大数目h. 也就是能被函数集打散 (shattered)的向量的最大数目.如果对任意的 l, 总存在一个 l个向量的集合可以被函数集打散,那么函数集的 VC 维就是无穷大. 例如: n 维坐标空间中线性函数集合的 VC 维是 n+1. 函数集合$f(z,\alpha) = \theta\sin(\alpha z)$, $\alpha\in R^1$的 VC 维是无穷大. 对指示函数集, 以下不等式以概率$1-\eta$成立.

$R(\alpha)\leq R_{emp}(\alpha) + \sqrt{\left(\frac{h(\ln(2l/h)+1)-\ln(\eta/4)}{l}\right)}$ 即$R(\alpha)\leq R_{emp}(\alpha) + \Phi(h/l)$. 第一部分是经验风险 (取决于函数集中特定的函数, 取决于训练方法), 第二部分是置信范围 (取决于整个函数集的 VC 维, 取决于算法设计). 我们的目标是最小化二者之和.
结构风险最小化. 将函数集分解为适当的结构$S_1\subset S_2\subset\cdots\subset S_n\cdots\subset S$, 选择结构中的一个适当的元素 (子集)$S_k$和这个元素中一个适当的函数$Q(z,\alpha_l^k)\in S_k$, 使得风险的界$R(w)\leq R_{emp}(w) + \Phi(h/l)$最小
举例: 正则化 .理解 SRM 的另一个视角: ERM + 正则化.,
SVM 的 VC 维:在 N 维空间中, 满足条件$\|w\|\leq A$的标准超平面构成的指示函数集的 VC 维满足下面的界: $h\leq\min([R^2A^2],N)+1$. 其中 R 为包含所有样本的最小超球的半径.在控制分类间隔后, VC 维可以大大低于特征空间维数. 定理: 如果包含 l 个样本的训练集被最大间隔超平面分开, 那么超平面测试错误率的期望有如下的界$EP_{\mathrm{error}}\leq E\min\left(\frac{m}{l},\frac{[R^2|w|^2]}{l},\frac{n}{l}\right)$. 其中, m是支持向量个数, R是包含数据的超球的半径, $|w|^{-2}$是分类间隔, n是输入空间的维数.
有限样本下, 要最优超平面有推广能力, 需三个原因满足其一: 数据压缩的期望大, 分类间隔的期望大, 输入空间维数低.SVM 特征空间中的最优超平面是否仍然最优? 是的, 因为控制了特征空间中的分类间隔!

贝叶斯决策

贝叶斯分类器是"最优分类器": 使平均错误率最小.
最小错误率贝叶斯决策: 比较$P(\omega_1\mid x)$和$P(\omega_2\mid x)$.条件期望损失: $R(\alpha_i\mid x) = E[\lambda(\alpha_i,\omega_j)\mid x]$. 期望风险: $R(\alpha) = \int R(\alpha(\boldsymbol{x})\mid\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$
贝叶斯公式 $P(\pi_1\mid\mathbf{x}_0) = \frac{p_1 f_1(\mathbf{x}_0)}{p_1 f_1(\mathbf{x}_0)+p_2 f_2(\mathbf{x}_0)}$
最小风险决策: $\min R(\alpha)$. 期望风险最小化.
ECM : $c(2\mid 1)P(2\mid 1)p_1 + c(1\mid 2)P(1\mid 2)p_2$. $R_1: \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})}\geq\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)$
一般形式: 若$\frac{P(x|w_1)}{P(x|w_2)}\gtrless\frac{L(\alpha_1|w_2)-L(\alpha_2|w_2)}{L(\alpha_2|w_1)-L(\alpha_1|w_1)}\frac{P(w_2)}{P(w_1)}$, 则$x\in\frac{w_1}{w_2}$
若损失函数表为$\lambda_{11},\lambda_{12},\lambda_{21},\lambda_{22}$, 则划为第一类的损失: $\lambda_{11}P(\omega_1\mid x) + \lambda_{12}P(\omega_2\mid x)$; 二: $\lambda_{21}P(\omega_1\mid x) + \lambda_{22}P(\omega_2\mid x)$.
当$\lambda_{11} = \lambda_{22} = 0, \lambda_{12} = \lambda_{21} = 1$时, 最小风险决策等于最小错误率决策.
这里$c(1\mid 2)$代表来自第二类, 分成第一类.
第一步: 计算后验概率$P(\omega_j\mid x) = \frac{p(x|\omega_j)P(\omega_j)}{\sum_{i=1}^c p(x|\omega_i)P(\omega_i)}$. 第二步: 计算风险 $R(\alpha_i\mid x) = \sum_{j=1}^c\lambda(\alpha_i\mid\omega_j)P(\omega_j\mid x), i=1,\ldots,k$. 第三步: 决策.

Neyman-Pearson 决策: 二分类问题. 第一类错误概率$P_1(e)=\int_{R_2} p(x\mid w_1)dx$, 第二类错误概率$P_2(e)=\int_{R_1} p(x\mid w_2)dx$, 固定一类错误率并使另一类错误率最小，即寻找两类样本的最优分界面满足$\min_t P_1(e)$, s.t. $P_2(e)=\epsilon_0$.
有:$P_1(e)=\int_{R_2} p(x\mid w_1)dx=1-\int_{R_1} p(x\mid w_1)dx$, 定义拉格朗日函数
$$L(t,\lambda)=P_1(e)+\lambda(P_2(e)-\epsilon_0)=(1-\lambda\epsilon_0)+\int_{-\infty}^{t}\lambda p(x\mid w_2)-p(x\mid w_1)dx$$
求导可得$\lambda=\frac{p(t|W_1)}{p(t|W_2)}$, $\int_{-\infty}^{t}p(x\mid w_2)dx=\epsilon_0$
正态分布条件下: $f(\mathbf{x})=\frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}}e^{-(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})/2}$判别函数(对数后验概率) $g_i(x)$. 决策面方程$g_i(x)=g_j(x)$, 对应$-\frac{1}{2}\left[(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)-(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\right]-\frac{1}{2}\ln\frac{|\Sigma_i|}{|\Sigma_j|}+\ln\frac{P(\omega_i)}{P(\omega_j)}=0$
若$\Sigma_i=\sigma^2\boldsymbol{I}, i=1,\dots,c$, 且$P(\omega_i)$相等. 则决策面方程$g_i(x)=-\frac{1}{2\sigma^2}\sum_{j=1}^{d}(x_j-\mu_{ij})^2$符合球状分布，各类先验概率相等，则分类只取决于样本到各类中心的距离(最小距离分类器). 若$P(\omega_i)$不等，则$g_i(x)=-\frac{1}{2\sigma^2}(x-\mu_i)^T(x-\mu_i)+\ln P(\omega_i)$. 则线性判别函数$g_i(\boldsymbol{x})=\boldsymbol{w}_i^T\boldsymbol{x}+b_i$. 其中$\boldsymbol{w}_i=\frac{1}{\sigma^2}\mu_i$, $b_i=-\frac{1}{2\sigma^2}\mu_i^T\mu_i+\ln P(\omega_i)$. 决策面向先验概率小的方向偏移.
若$\boldsymbol{\Sigma}_1=\boldsymbol{\Sigma}_2=\boldsymbol{\Sigma}$. $(\mu_1-\mu_2)'\Sigma^{-1}\mathbf{x}_0-\frac{1}{2}(\mu_1-\mu_2)'\Sigma^{-1}(\mu_1+\mu_2)\ge\ln\left[\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)\right]$
Sample: $(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)'\mathbf{S}_{\text{pooled}}^{-1}\mathbf{x}_0-\frac{1}{2}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)'\mathbf{S}_{\text{pooled}}^{-1}(\bar{\mathbf{x}}_1+\bar{\mathbf{x}}_2)\ge\ln\left[\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)\right]$
$g_i(x)=-\frac{1}{2}\left(x^T\Sigma^{-1}x-\mu_i^T\Sigma^{-1}x-x^T\Sigma^{-1}\mu_i+\mu_i^T\Sigma^{-1}\mu_i\right)+\ln P(\omega_i)$, 可得**线性**判别函数$g_i(x)=\boldsymbol{w}_i^T\boldsymbol{x}_i+b$, 其中$w_i=\Sigma^{-1}\mu_i$, $b_i=-\frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i+\ln P(\omega_i)$, 决策面$g_i(\boldsymbol{x})=g_j(\boldsymbol{x})$, 可写为$\boldsymbol{w}^T(\boldsymbol{x}-\boldsymbol{x}_0)=0$, 其中$\boldsymbol{w}=\Sigma^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)$,
$x_0=\frac{1}{2}(\mu_i+\mu_j)-\frac{\ln(P(\omega_i)/P(\omega_j))}{(\mu_i-\mu_j)^T\Sigma^{-1}(\mu_i-\mu_j)}(\mu_i-\mu_j)$, 当$P(\omega_i)=P(\omega_j)$时, $x_0=\frac{1}{2}(\mu_i+\mu_j)$
一般情况，各类协方差不同 分为第一类: $-\frac{1}{2}\mathbf{x}_0'(\mathbf{S}_1^{-1}-\mathbf{S}_2^{-1})\mathbf{x}_0+(\bar{\mathbf{x}}_1'\mathbf{S}_1^{-1}-\bar{\mathbf{x}}_2'\mathbf{S}_2^{-1})\mathbf{x}_0-k\ge\ln\left[\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)\right]$
$g_i(x)=-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)-\frac{1}{2}\ln|\Sigma_i|+\ln P(\omega_i)=x^TW_ix+w_i^Tx+w_{i0}$. 其中$W_i=-\frac{1}{2}\Sigma_i^{-1}$, $w_i=\Sigma_i^{-1}\mu_i$, $w_{i0}=-\frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i-\frac{1}{2}\ln|\Sigma_i|+\ln P(\omega_i)$
决策面$x^T(W_i-W_j)x+(w_i-w_j)^Tx+w_{i0}-w_{j0}=0$ 为超二次曲面.

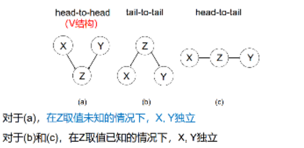
概率密度函数估计

似然函数: $l(\theta)=p(D\mid\theta)=p(\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_N\mid\theta)=\prod_{i=1}^{N}p(\boldsymbol{x}_i\mid\theta)$
最大似然估计需要假设特定的样本分布，不一定在极值点处取得，假设各个样本是独立抽取的
$p(\theta\mid D)=\frac{p(D|\theta)p(\theta)}{\int p(D|\theta)p(\theta)d\theta}=\frac{p(D|\theta)p(\theta)}{p(D)}$. 损失函数，把$\theta$估计为$\hat\theta$所造成的损失，记为$\lambda(\widehat{\theta,\theta})$. 期望风险$R=\int_{E^d}\int_\Theta\lambda(\hat\theta,\theta)p(\boldsymbol{x},\theta)d\theta d\boldsymbol{x}=\int_{E^d}R(\hat\theta\mid\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$
其中条件风险$R(\hat\theta\mid\boldsymbol{x})=\int_\Theta\lambda(\hat\theta,\theta)p(\theta\mid\boldsymbol{x})d\theta$, 最小化期望风险用最小化条件风险. 常用的损失函数$\lambda(\hat\theta,\theta)=(\theta-\hat\theta)^2$. 可以证明，如果采用平方误差损失函数，则$\theta$的贝叶斯估计量$\hat\theta$是在给定$x$时$\theta$的条件期望，即
$\hat\theta=E[\theta\mid x]=\int_\Theta\theta p(\theta\mid\boldsymbol{x})d\theta$
贝叶斯学习: 我们的目标是希望通过对已有观测数据的学习，估计样本的真实分布$p(\boldsymbol{x})$. $p(\boldsymbol{x}\mid D)=\int_\Theta p(\boldsymbol{x}\mid\theta,D)p(\theta\mid D)d\theta=\int_\Theta p(\boldsymbol{x}\mid\theta)p(\theta\mid D)d\theta$
其中$p(\theta\mid D)=\frac{p(D|\theta)p(\theta)}{\int p(D|\theta)p(\theta)d\theta}$. $p(\boldsymbol{x}\mid\theta)$容易算，关键是求得$p(\theta\mid D)$. 实际计算中，使用参数估计的递推贝叶斯方法.
正态分布下的贝叶斯估计. 对一维正态分布$p(x\mid\mu)\sim N(\mu,\sigma^2)$, σ^2已知，估计μ. 假设先验分布$p(\mu)\sim N(\mu_0,\sigma_0^2)$, 设$m_N=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$,
结论$\sigma_N^2=\frac{\sigma_0^2\sigma^2}{N\sigma_0^2+\sigma^2}$, $\hat\mu=\frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}m_N+\frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0$.$N=0$时$\hat\mu=\mu_0$,$N\to\infty$时$\hat\mu\to m_N$
对参数的先验分布没有任何了解时可以使用贝叶斯估计.
共轭先验: 对于一个似然函数，如果该先验分布与其相乘后得到的后验分布与先验分布有相同的函数形式，则称这个先验函数为似然函数的共轭先验.
二项分布与 Beta 分布即为共轭先验的例子.
非参估计: 已知样本集$D=\{x_1,\dots,x_N\}$, 其中样本均从服从$p(\boldsymbol{x})$ 的总体中独立抽取，求估计$\hat p(\boldsymbol{x})$, 近似$p(\boldsymbol{x})$.
考虑随机变量$\boldsymbol{x}$ 落入区域$\Re$ 的概率，$P_\Re=\int_\Re p(x)dx$. 则 D 中有 k 个样本落入区域$\Re$ 的概率$P_k=C_N^kP_\Sigma^k(1-P_\Sigma)^{N-k}$. k 的期望值$E[k]=NP_\Re$, 因此取$P_\Re$ 的估计$\hat P_R=\frac{k}{N}$. (k :实际落到$\Re$ 中的样本数). 由$P_R=\int_R p(x)dx=p(x)V$, 因此$\hat p(x)=\frac{k}{NV}$
关于 V(包含 x 的一个小区域的体积)的选择: 过大，估计粗糙；过小，可能某些区域中无样本. 两种选择策略: 选择 V(Parzen 窗法); 选择 k ，V 为正好包含 x 的k_N 个近邻(k_N 近邻估计)
k_N 近邻估计: $\hat p_n(x)=\frac{k_n/N}{V_n}$.
Parzen 窗法: $\hat p(x)=\frac{1}{N}\sum_{i=1}^{N}k(\boldsymbol{x},\boldsymbol{x}_i)$. 窗函数(核函数)$k(x,x_i)$: 反映$x_i$对$p(x)$的贡献，实现小区域选择. 满足条件: $k(x,x_i)\ge0$, $\int k(x,x_i)dx=1$
常用窗函数: 超立方体窗;正态窗;超球窗. 窗宽的选择: 样本数少则选大些，样本数多则选小些.
在满足一定的条件下，估计量$\hat p_N(x)$ 是渐近无偏和平方误差一致的. 条件是: 总体密度$p(x)$ 在x 点连续. 窗函数满足以下条件: $\varphi(u)\ge0,\int\varphi(u)du=1$ (窗函数具有密度函数的性质); $sup\varphi(u)<\infty$ (窗函数有界).
$\lim_{\|u\|\to\infty}\varphi(u)\prod_{i=1}^{d}u_i=0$(窗函数随着距离的增大趋于零). 窗宽受以下条件约束: $\lim_{N\to\infty}V_N=0$(窗体积随着 N 的增大而趋于零); $\lim_{N\to\infty}NV_N=\infty$(但体积减小的速度要低于 1/N)

概率图模型

生成式模型上的概率推断: 已知部分随机变量的取值$\mathbf{x}_\text{v}$, 希望推断未知变量$\mathbf{X}_\text{h}$取值的分布, $p(\mathbf{x}_\text{h}\mid\mathbf{x}_\text{v},\boldsymbol{\theta})=\frac{p(\mathbf{x}_\text{h},\mathbf{x}_\text{v}|\boldsymbol{\theta})}{p(\mathbf{x}_\text{v}|\boldsymbol{\theta})}=\frac{p(\mathbf{x}_\text{h},\mathbf{x}_\text{v}|\boldsymbol{\theta})}{\sum_{\mathbf{x}_\text{h}'}p(\mathbf{x}_\text{h}',\mathbf{x}_\text{v}|\boldsymbol{\theta})}$.生成式模型进行推断的关键: 计算联合概率分布..利用随机变量之间的独立性可以降低联合分布计算的复杂性.找出合适的排列方式，利用链式法则和条件独立性质能够大大降低联合概率计算的复杂度.利用图来表示随机变量之间的依赖关系，节点表示随机变量，边表示随机变量间的**直接**依赖关系. 在概率依赖关系推断中，引入概率图的目的是: 表示更直观，便于理解; 减少模型参数; 减少计算量.
对时间序列建模: 马尔可夫链. 分别统计训练样本中 CpG 岛和非 CpG 岛的转移矩阵$a_{st}^+=\frac{c_{st}^+}{\sum_{t'}c_{st'}^+}$, $a_{st}^-=\frac{c_{st}^-}{\sum_{t'}c_{st'}^-}$. 对于一个特定样本$\mathbf{x}$, 根据类条件概率密度分别计算它属于该类别的似然值. 计算二者的似然比作为评价函数: $l(\mathbf{x})=P(\mathbf{x}\mid\omega_1)/P(\mathbf{x}\mid\omega_2)$. 为便于计算，转化为对数似然函数
$$S(x)=\log\frac{P(x|+)}{P(x|-)}=\sum_{i=1}^{L}\log\frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}}=\sum_{i=1}^{L}\beta_{x_{i-1}x_i}$$
计算β_{st}是a_{st}^+和a_{st}^- 的对数似然比. 决策函数: If $S(\mathbf{x})>\lambda$, then $x\in\begin{cases}\omega_1\\ \omega_2\end{cases}$. 其中$\lambda$是划定的阈值(例如跟两类的先验概率有关)
在 HMM 模型中，样本的观测值是依据一定概率由一些不可观测 的系统内部状态(隐变量)决定的.这些内部状态之间的转化服从马尔可夫模型.
HMM 模型符号约定, 模型含有 n 个隐状态$Q=\{q_1,\dots,q_n\}$, 观测值的取值范围$V=\{v_1,v_2,\dots,v_V\}$, 状态转移概率矩阵$A=[a_{ij}]_{n\times n}$, 观测序列$O=o_1o_2\dots o_L$, 隐状态序列$\mathbf{x}=x_1x_2\dots x_L$, 发射概率矩阵$E=[e_{ij}]_{n\times V}$, 其中$e_{ij}=p(o=v_j\mid x=q_i)$ 表示模型处在隐状态q_i 时观测到v_j 的概率.初始概率分布$\pi=[\pi_1,\pi_2,\dots,\pi_n]$.
HMM 模型计算中的常见问题: 模型评估问题 (Evluation): 给定模型 M 和观测序列 O，求在该模型观测到该序列的概率$P(O\mid M)$, 即计算 likelihood. 隐状态推断问题 (Decoding): 给定模型 M 和观测序列 O ，求最有可能的隐状态序列 x. 模型训练问题 (Learning). (非监督：给定模型结构、观测序列 O，求解模型参数; 监督：给定模型结构、观测序列 O、隐状态序列 x，求解模型参数)

计算观测序列的似然值: 假如隐状态序列 x 已知时，很容易求得似然函数值$p(O\mid\mathbf{x})=\prod_{i=1}^{L}p(o_i\mid x_i)$, 并得到观测序列和隐状态序列的联合概率
$p(O,\mathbf{x})=p(O\mid\mathbf{x})p(\mathbf{x})=\prod_{i=1}^{L}p(o_i\mid x_i)\times\left(\pi_{x_1}\times\prod_{i=2}^{L}p(x_i\mid x_{i-1})\right)$.
观测序列 O 的概率可分解为所有可能隐状态序列下的概率和
$p(O)=\sum_\mathbf{x}p(O,\mathbf{x})=\sum_\mathbf{x}p(O\mid\mathbf{x})p(\mathbf{x})$
Forward algorithm: 一种动态规划算法. 计算算复杂度$O(n^2L)$. 定义初值$\alpha_1(j)=e_j(o_1)\times\pi_j$, 迭代求解$\alpha_t(j)=e_j(o_t)\times\sum_{i=1}^{n}\alpha_{t-1}(i)a_{ij}$, 终止结果$p(O)=\sum_{i=1}^{n}\alpha_L(i)$. 估计观测序列在当前模型下出现的概率
Backward Algorithm. $\beta_t(j)=p(o_{t+1}o_{t+2}\dots o_T\mid x_t=q_j)$. 定义初值$\beta_L(j)=1,j=1{:}n$, 迭代求解$\beta_t(j)=\sum_{i=1}^{n}a_{ji}e_i(o_{t+1})\beta_{t+1}(i),t=1{:}L,j=1{:}n$, 终止结果$p(O)=\sum_{i=1}^{n}\pi_ie_i(o_1)\beta_1(i)$.条件于当前状态下，观测到特定的后续观测序列的概率
解码问题(Decoding): 给定模型和观测序列,求解最有可能的隐状态序列. Viterbi 算法(动态规划算法). 定义初值$v_1(j)=e_j(o_1)\times\pi_j,pa_1(j)=0,j=1{:}n$, 迭代求解$v_t(j)=\max_{i=1:n}v_{t-1}(i)a_{ij}e_j(o_t)$, $pa_t(j)=\operatorname{argmax}_{i=1:n}v_{t-1}(i)a_{ij}e_j(o_t)$. 终止结果$p^*=\max_{i=1:n}v_L(i)$, $\mathbf{x}_L^*=\operatorname{argmax}_{i=1:n}v_L(i)$. 计算复杂度$O(n^2L)$
HMM 模型训练: Baum-Welch 算法. Forward-Backward Algorithm. EM（expectation-maximization）算法. E-step: 利用现有参数求期望; M-step: 对模型参数进行最大似然估计，重新估计分布参数. 算法迭代: 利用新获得的参数$\hat A,\hat E$ 重新用前向和后向 算法计算概率，并反复迭代运行，直到模型参数收敛， 或者运行达到指定的迭代次数
下列方法中使用动态规划算法的包括:前向算法，后向算法, Viterbi 算法, Baum-Welch 算法
贝叶斯网络: BN(G, Θ) 表示一个贝叶斯网络. G 是一个有向无环图，Θ 是网络参数，由一系列条件概率组成. 对于网络中的节点$\mathbf{x}$, 其参数表示为$p(\mathbf{x}\mid \text{parents}(\mathbf{x}))$. 整体的联合概率: $p(\mathbf{x})=\prod_{i=1}^{n}p(x_i\mid\boldsymbol{pa}_i)$
朴素贝叶斯分类器假设特征之间条件于类别独立. Laplace Smoothing.
$p(Y=\omega_i)=\frac{\sum_{i=1}^{N}I(y_i=\omega_i)+1}{N+C}$, $p(x_j=v_i\mid Y=\omega_i)=\frac{\sum_{i=1}^{N}I\left(x_j^{(i)}=v_i,y_i=\omega_i\right)+1}{\sum_{i=1}^{N}I(y_i=\omega_i)+S}$
在 DAG 中任意三个节点之间的连接关系可总结为 head-to-head, (V 结构), tail-to-tail, head-to-tail.



贝叶斯网络的学习: 参数学习(知道模型的网络结构，但不知道具体的概率 分布，根据已知样本数据估计模型参数); 结构学习(网络的结构未知时，需要从数据中提取信息 来寻找最能恰当描述数据的贝叶斯网络结构。从所有可 能的结构中搜寻最优的网络结构是一个 NP 难问题).
参数学习: 给定数据的情况下，求解最大后验概率（MAP）. 参数$\boldsymbol{\theta}$ 的后验概率表示为: $p(\boldsymbol{\theta}\mid D)=p(D\mid\boldsymbol{\theta})p(\boldsymbol{\theta})/p(D)$.$p(D)$ 与参数无关，只需要最大化$p(D\mid\boldsymbol{\theta})p(\boldsymbol{\theta})$, 取对数,即
$\hat{\boldsymbol{\theta}}=\operatorname{argmax}_{\boldsymbol{\theta}}\sum_{i=1}^{N}\log p(\mathbf{x}_i\mid\boldsymbol{\theta})+\log p(\boldsymbol{\theta})$. 如果模型参数的先验概率相同，则简化为最大似然估计(MLE)问题.
多项分布时的参数学习: 根据多项分布的特性，似然函数可以表示为
$p(\boldsymbol{D}_t\mid\boldsymbol{\theta}_t)=\prod_{c=1}^{q_t}\prod_{k=1}^{K_t}\theta_{tck}^{N_{tck}}=\prod_{c=1}^{q_t}p(\boldsymbol{D}_{tc}\mid\boldsymbol{\theta}_{tc})$. 对于多项式分布，通常选择 Dirichlet 分布作为先验分布. 后验概率依然是 Dirichlet.
结构学习: 寻找最优结构是 NP-hard 问题.
Markov Random Field, MRF: 利用无向图来表示随机变量之间的关系

特征选择

模型特征过多可能会导致: 模型过于复杂; 过拟合; 测试错误率高; 样本数目有限时带来病态矩阵等问题
对 MSE 的数学期望可以分解为
$$E\left(y_0-\hat f(x_0)\right)^2=\operatorname{Var}\left(\hat f(x_0)\right)+\left[\operatorname{Bias}\left(\hat f(x_0)\right)\right]^2+\operatorname{Var}(\epsilon)$$
$\operatorname{Var}(\epsilon)$ 部分从理论上不可消减, $\operatorname{Bias}(f(x_0))$ 反映了模型对数据的拟合能力. Variance $\left(\hat f(x_0)\right)$ 反应在训练数据上拟合结果的稳定性.
特征选择的目的: 是否输入变量的各个分量都对输出变量的预测有贡献? 特征太多影响模型参数估计的稳定性.
相关系数: $r=\frac{\sum[(x_i-\bar x)(y_i-\bar y)]}{\sqrt{\sum(x_i-\bar x)^2*\sum(y_i-\bar y)^2}}$
互信息:$I(X,Y)=\sum_X\sum_Y p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$
特征选择可能带来的好处: 有利于数据的可视化与数据理解,降低数据采集和数据存储的开销,降低模型使用时的计算开销，避免维数灾难，提高预测性能.
最优子集: 计算量大，当维度很高时不行;过滤法(filtering): 先选再学. 先选特征，再做学习;包裹法(wrapper): 边学边选. 以最终分类器的性能作为指标来选特征，一般要求分类器能够处理高维特征，分类器能够在特征维度高但样本数有限时仍能得到较好的效果;嵌入法: 目标函数引入正则项.
过滤法: 设计某种统计量来度量特征 X 与和响应值 Y 之间的 “关联程度”。选出其中“关联程度”最高的 k 个特征，或者高于一定阈值 t 的所有的特征. 只能用训练集来筛选特征，选特征时不要用测试集.基于相关系数的判据，基于条件熵(特征取值对分类的影响), 基于 KL 散度. 如果我们把数据分为训练集+测试集，在使用过滤法筛选特征时，应使用训练数据进行筛选.
包裹法: 把特征选择与分类器设计集成在一起，利用分类器进行特征选择. 选特征时不要用测试集，在训练集上利用交叉验证选择特征. 有 Forward Stepwise selection, Backward stepwise selection. 混合算法: 在前向算法的基础上允许特征的剔除.完全相关的变量是冗余的，加入它不会带来信息增益，但高度相关的变量不见得没有用.
Akaike information criterion (AIC): $AIC=2d-2ln(L)$
Bayesian information criterion (BIC): $BIC=\ln(n)d-2\ln(L)$
正则化: 最小化残差平方和$\text{RSS}=\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_jx_{ij}\right)^2$, Ridge regression, 最小化$\text{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$, LASSO: 最小化$\text{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$. 当 V<1 时，非凸问题，求解困难. 做特征选择时L_0最好，等价于最优子集法，但计算上求解困难.
随机置换法: 随机打乱样本的标记，通过同样的学习过程，统计出没有分类信息情况下的识别性能分布。将真实值与其进行比较，得到分类器性能的随机置换 P 值。P 值小(例如<0.05)则推断这种关系较可能是存在的.

特征提取与可视化

PCA: 转化成主成分后方差不变. $\sum_{i=1}^{p}\operatorname{var}(\xi_i)=\sum_{i=1}^{p}\lambda_i$.等于原始方差. PCA 需要 0 均值化，不需要调整到相同方差.进行 PCA 的过程中没有使用样本类别信息.
PCA 的算法流程: 输入数据$\mathbf{X}=\left[\mathbf{x}^{(1)},\mathbf{x}^{(2)},\dots,\mathbf{x}^{(n)}\right]_{p\times n}$, n 个 p 维的数据样本点. 1) 计算样本均值$\mu=\sum_{i=1}^{n}\mathbf{x}^{(i)}$; 2)对数据进行中心化$\mathbf{x}^{(i)}=\mathbf{x}^{(i)}-\mu$; 3) 计算协方差矩阵$\Sigma=\frac{1}{n}\mathbf{X}\mathbf{X}^T$; 4) 对协方差矩阵进行特征值分解; 5)取最大的 k 个特征值对应的特征向量，构成投影矩阵$\mathbf{W}=[\alpha_1,\alpha_2,\dots,\alpha_k]_{p\times k}$; 6) 投影到.$\boldsymbol{X}'=\mathbf{W}^\top\boldsymbol{X}$. **注意协方差矩阵维度**
使用 SVD 进行 PCA: 对中心化后的样本矩阵进行 SVD, $\mathbf{X}=\mathbf{U\Sigma V}^\top$. 则$\mathbf{X}^\top\mathbf{X}=\mathbf{V\Sigma}^\top\mathbf{\Sigma V}^\top$, 取$\text{V}^\top$ 中对应于 k 个最大特征值的 k 行.
K-L 变换: 把随机向量用一组正交基向量展开.
非监督的 KL 变换中用总体协方差矩阵作为 K-L 产生矩阵，等价于 PCA. 监督的 KL 变换中，首先计算总类内离散度矩阵$S_w=\sum_{i=1}^{c}P_i\Sigma_i$. 作谱分解，得本征值为$\lambda_i$ 的特征向量u_i. 计算$J(y_i)=\frac{u_j^TS_bu_j}{\lambda_i}$ 作为评价新特征的各个分量的分类性能的指标. 其中$S_b=\sum_{i=1}^{c}P(\omega_i)(\mu_i-\mu)(\mu_i-\mu)^T$ 为类间离散度矩阵. 用$J(y_i)$ 排序$J(y_1)\ge J(y_2)\ge\cdots\ge J(y_d)\ge\cdots\ge J(y_D)$. 选择前 d 个分量组成新的特征向量,相应的$u_j$ 组成变换阵U
PCA 的结果倾向于全局表示，而 NMF 的结果倾向于局部表示
Manifold Learning. LLE：Locally Linear Embedding; Isomap (Isometric feature mapping):当样本分布较密集的区域，假定样本空间结构可在该局部用欧式 距离度量，对于两个相距较远的点，寻找一系列两两相邻的样本点构成连接二者的路径，用最短路径上局部距离之和度量二者距离，得到重新定义样本“距离矩阵”后，用 MDS 映射到低维空间.
Multidimensional Scaling, MDS: 根据**样本之间的距离**关系或不相似度关系在低维空间里生成对样本的表示. 把样本之间的距离关系或不相似关系在二维或三维空间里表示出来.
t-SNE: 利用概率分布来表示样本间距离. 当高维和低维有相同的相似度时，在高维空间中很近的点在低维空间中更近了，而高维空间中很远的点在低维空间中更远了，这样在低维空间中更有区分度.在高维空间中用正态分布,在低维空间中用 t 分布，损失函数: 两个分布之间的距离--KL 散度
$C=KL(P\parallel Q)=\sum_i\sum_j p_{ij}\log\frac{p_{ij}}{q_{ij}}$. C 恒大于 0, 梯度下降让其尽量小.
在 t-SNE 方法中嵌入概率 P 的取值受到方差σ_i的影响. 对于每个样本点x_i其取值都不相同，对于样本点比较密集的区域，使用较小的σ参数；反之，对于稀疏的样本区域，可以使用较大一些的取值. σ_i的大小会影响分布的信息熵，的熵值随着的增大而增大. 我们把控制σ取值的参数称为困惑度 (Perplexity)，其定义为$\text{Perp}(P_i)=2^{H(P_i)}$. 困惑度大致等价于在匹配每个点的原始和拟合分布时考虑的最近邻数。较小的困惑度取值意味着我们在计算嵌入分布时候只考虑最近的几个近邻之间结构关系，而较大的困惑度取值意味着周围更多的样本被考虑进来。注意: 经过 tSNE 投影后在低维度空间中的样本分布不一定能保持在原空间中样本间的距离关系。因此，通常不使用 t-SNE 投影后的坐标来直接度量样本之间的距离
可以用来做样本低维可视化的方法包括: PCA, Isomap, LLE, t-SNE, MDS.

聚类分析

K-means 方法: $\text{SSE}=J(c,\mu)=\sum_{i=1}^{n}\left|x_i-\mu_{C_i}\right|^2$. k 均值聚类的局限性: 要求类别数已知;是最小方差划分，并不一定能反映内在分布;对类别的硬划分;与初始划分有关，不保证全局最优;与距离的度量方式有关.
Fuzzy k-means: 元素以一定的程序属于某集合. 基本思想: 根据一定的模糊化规则，把原来的一个 (或几个)特征变量分成多个模糊变量，使每个模糊特征表达某一局部特性，利用这些新特征来进行模式识别.
GMM 模型: 利用 EM 算法求解. E-step: 利用现有参数求类别标签的期望.
M-step: 对模型参数进行最大似然估计，重新估计分布参数
K 均值算法的结果可能与初值选取有关，分级聚类的结果是确定性的.
一致聚类 (Consensus Clustering). 解决思路：多次聚类，再合并结果. 原始数据集表示为D, 进行S次采样，表示为$D^{(I)},\dots,D^{(S)}$, 在每个数据集上分别进行聚类. 定义第S个集合上聚类数设置为K时的连接矩阵$M_{(s)}^{(K)}$:$M_{(s)}^{(K)}(i,j)=1$代表样本 i 和样本 j 在同一个类里.整体的一致性矩阵M, $\mathcal{M}^{(K)}(i,j)=\frac{\sum_s M_{(s)}^{(K)}(i,j)}{\sum_s I_{(s)}(i,j)}$. 其中$I_{(s)}$ 为指示矩阵.$\mathcal{M}^{(K)}$ 中的元素取值为 0-1，1 表示每次聚类都聚在一起，0 则反之.如果数据聚类结果的一致性很高，则$\mathcal{M}^K$中元素的取值不是靠近 1,就是靠近 0，中间取值的点很少。反之，若多次聚类结果的一致性差，则$\mathcal{M}^K$中有很多元素的取值在 (0, 1)之间. 距离度量$\text{Dist}^{(K)}=\left(\mathbf{1}-\mathcal{M}^{(K)}\right)$.定义 CDF, 以及曲线下面积$\text{A}(K)$, 定义衡量变化的指标$\Delta(K)$. 可以取$\Delta(K)$趋近稳定前一时刻的 K 为可能的聚类数目
对于一致聚类算法，对于内层聚类: 无放回样本抽样; 特征子集抽样; k 均值算法采用不同的初始条件运行多次;可以混合使用不同的聚类算法.
神经网络方法: 自组织映射网络 (SOM). 两层: 输入层和输出层. 输入层: 节点数量等于输入向量维度. 输出层: 一维或二维 (方形网格或蜂窝状)，更高维不常见. 竞争层的每一个神经元都连接到输入层的所有节点，但彼此之间没有明确连接，却会互相影响. 自组织 (学习)过程: 初始化：对所有连接权重进行初始化; 竞争：对于某一个输入，计算竞争层中每一个神经元对该输入的响应强度，为竞争提供基础。响应强度最大的神经元被判定为胜者 (winner); 合作：获胜的神经元决定了兴奋神经元拓扑邻域的空间位置，为相邻神经元的合作提供基础; 适应：邻域内受激神经元适当调整相关的连接权重，使得获胜神经元对相似输入的后续响应增强. 当达到事先确定的迭代次数，或邻域缩小到只包含该获胜节点 本身时，停止迭代。SOM 算法的目标函数为$E=\sum\sum h_{kj}\|X_i-\boldsymbol{m}_j\|^2$. 与 K-means 聚类的目标函数非常相似.当取消邻域范围时，即只更新获胜神经元权重，SOM 算法退化为 K-means 算法的随机迭代，神经元的数量 N 为聚类数目 k.
聚类结果评估: 有监督方法(有类别基准，判断类别标签与聚类结果的吻合程度): 基于互信息（Mutual Information）的分数，调整后的兰德系数（Adjusted Rand Index）.无监督方法(无类别基准，判断类内和类间的离散程度): 轮廓系数（Silhouette Coefficient）.
U 与 V 之间的互信息为: $MI(U,V)=\sum_{i=1}^{|U|}\sum_{j=1}^{|V|}P(i,j)\log\left(\frac{P(i,j)}{P(i)P'(j)}\right)$, $P(i,j)=|U_i\cap V_j|/N$. 当预测结果和真实标签完全相同时，互信息等于信息熵。因此，在同一数据集和相同类别数前提下，MI 越高，聚类结果越好
对于单个样本 i，其轮廓系数为$s(i)=\frac{b(i)-a(i)}{max(a(i),b(i))}$. 对于整个样本集合，定义其轮廓系数是所有样本轮廓系数的平均值.轮廓系数本质上是在衡量聚类的紧致性，轮廓系数介于[-1,1]之间，取值越大表示类内距离小，类间距离大.

深度学习

Adam: 同时考虑动量和自适应学习率因子. $\boldsymbol{g}_t=\nabla_{\boldsymbol{\Theta}}\mathcal{L}(f(\boldsymbol{x}_i;\boldsymbol{\theta}),y_i)$. 动量: $\boldsymbol{m}_t=\beta_1\boldsymbol{m}_{t-1}+(1-\beta_1)\boldsymbol{g}_t$. 自适应学习率因子: $\boldsymbol{v}_t=\beta_2\boldsymbol{v}_{t-1}+(1-\beta_1)\boldsymbol{g}_t\odot\boldsymbol{g}_t$. 修正动量和学习率因子: $\widetilde{\boldsymbol{m}_t}=\frac{\boldsymbol{m}_t}{1-\beta_1}$, $\widetilde{\boldsymbol{v}_t}=\frac{\boldsymbol{v}_t}{1-\beta_2}$. $\boldsymbol{\Theta}=\boldsymbol{\Theta}-\alpha\frac{1}{(\sqrt{\widetilde{v_t}+\varepsilon})}\odot\widetilde{\boldsymbol{m}_t}$
BatchNorm: 用 Batch 内均值方差替代.
CNN 特点一：稀疏连接; 特点二：参数共享; 特点三：等变表示.
多通道卷积: 图像有多个颜色通道,各通道分别进行卷积,再把卷积的结果相加. 一个卷积核提取一个特征,多个卷积核产生多个特征,每个特征对应于一个通道. 原始图$N\times N$, 卷积核$m\times m$, 卷积核$(N-m+1)\times(N-m+1)$.
池化: 最大池化，平均池化，随机池化.
令 O=输出图像的尺寸, I=输入图像的尺寸, K=卷积层的核尺寸, S=移动步长, P =填充数,则输出图像尺寸$O=\frac{I-K+2P}{S}+1$. 严格来说这个尺寸是要**向下**取整的
参数个数: (核长 x 核宽 x **输入通道数**+1)x 输出通道数
RNN: $\boldsymbol{h}_t=\sigma_h(\boldsymbol{b_h}+\boldsymbol{W}\boldsymbol{h}_{t-1}+\boldsymbol{U}\boldsymbol{x}_t)$; $\boldsymbol{y}_t=\sigma_y(\boldsymbol{b}_y+\boldsymbol{V}\boldsymbol{h}_t)$
$\frac{\partial L_t}{\partial U}=\sum_{s=0}^{t}\frac{\partial L_t}{\partial y_t}\frac{\partial y_t}{\partial h_t}\frac{\partial h_t}{\partial h_s}\frac{\partial h_s}{\partial U}$, 其中$\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial h_{t-2}}\dots\frac{\partial h_{s+1}}{\partial h_s}$
LSTM: 时间步上的 Residual connection/更好地处理长程记忆. 遗忘门（Forget gate）: 决定对原有的 cell state 的记忆程度. GRU: 与 LSTM 思想类似，但计算开销更小.
Attention. Query: $f_q(h_t^d)$; Key: $f_k(h_i^e),i=1,\dots,n$; Value: $h_i^e,i=1,\dots,n$. 而有
$a_{t,i}=\frac{\exp\left(f_q(h_t^d)\cdot f_k(h_i^e)\right)}{\sum_{j=1}^{n}\exp\left(f_q(h_t^d)\cdot f_k(h_j^e)\right)}$, $c_t=\sum_{i=1}^{n}a_{t,i}h_i^e$
Transformer: 适合多模态：任何数据都在做 tokenize 后才进行输入，且 attention 本身就很方便用于模态间对齐;方便提取长程依赖;便于并行化，有很高的数据容量. 缺点: 缺少类似平移不变性与卷积的这样的对问题的特征洞悉和相应的结构假设,需要大量有监督训练数据或设计特殊的预训练任务来弥补.
$X_o=\text{Attention}(Q,K,V)=\text{softmax}\left(\frac{M+QK^T}{\sqrt{d_k}}\right)V$. 参数量为$n^2L^2+n^2Ln$
生成模型: 生成模型通常是把样本看作是从某种未知概率分布中的采样，因 此生成模型的任务就是估计或模拟样本的概率分布
自回归类模型的总结: 直接：利用待拟合分布为离散分布的特点，将网络输出直接定为分布列; 自回归：把待拟合分布拆分成一系列条件概率乘积，对应 地，进行采样生成时，会一步一步地生成最终结果的一小部分;参数共享：通常用同一个模型，为它输入条件概率中的任 何条件，输出下一部分的分布列
VAE: 限制样本在隐空间中的分布，利用变分推断来求解参数. 利用自编码器形状的神经网络构造的有向概率图模型; 模型的推断采用了变分推断(Variational Inference)的方法; 除了生成新样本，变分自编码器还在很多领域中被用 来从高维数据中提取嵌入的低维流形
GAN: 优势：模型训练时不需要对隐变量做推断; 理论上,只要是可微分函数都可以用于构建判别器 D 和生成器 G; 生成器 G 的参数更新不是直接来自数据样本,而是使用来自 D 的反向传播. 劣势: 可解释性不足,生成模型的分布没有显式的表达，训练 GAN 需要达到纳什均衡，GAN 的训练中同步 G 与 D 往往困 难，模型的收敛不稳定，易出现“model collapse”的情况
求导: $\nabla_\mathbf{x}\mathbf{A}\mathbf{x}=\mathbf{A}^\top$, $\nabla_\mathbf{x}\mathbf{x}^\top\mathbf{A}=\mathbf{A}$, $\nabla_\mathbf{x}\mathbf{x}^\top\mathbf{A}\mathbf{x}=(\mathbf{A}+\mathbf{A}^\top)\mathbf{x}$
若$y=\boldsymbol{A}X$, 则$\frac{\partial y}{\partial\boldsymbol{A}}=x^\top$. $\tanh'(x)=1-\tanh^2(x)$. $\text{sigmoid}(x)=\text{sigmoid}(x)\cdot(1-\text{sigmoid}(x))$
$f(x)=\frac{1}{1+e^{-x}}$, $f'(x)=\frac{e^{-x}}{(1+e^{-x})^2}$; $g(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$; $g'(x)=\frac{4}{(e^x+e^{-x})^2}$