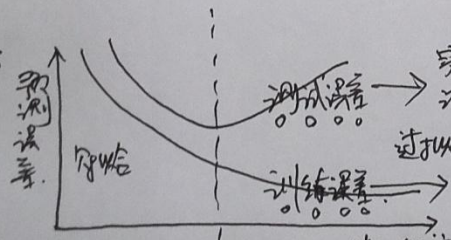
1、统计学习 { 监督学习 → 假设X和Y具有联合概率分布
非监督学习
半监督学习
强化学习

输入 → 特征 → 模型 → 输出
∈输入空间　∈特征空间　P(Y|X)或Y=f(X)　∈输出空间
∈假设空间

2、三要素：
{ 模型(假设空间)：输入空间到输出空间的映射 { P(Y|X) 条件概率模型
Y=f(X) 非概率模型
策略(评价准则) { 损失函数L(Y,f(X)) { 0-1：L(Y,f(X))={1 Y≠f(X); 0 Y=f(X)
平方损失：L(Y,f(X))=(Y-f(X))² → 非概率模型
绝对损失：L(Y,f(X))=|Y-f(X)|
对数损失：L(Y,P(Y|X))=-log P(Y|X) → 条件概率模型
风险函数 { 期望风险：R_exp(f)=E_p[L(Y,f(X))] → 学习的目的是选择期望风险最小的模型
经验风险：R_emp(f)=1/N Σ_{i=1}^{N} L(y_i,f(x_i)) 经验风险最小的模型
算法(学习算法) { 显式解 → 直接求
隐式解 → 数值计算方法 { 迭代下降法
梯度下降法
牛顿法

E_p[L(Y,f(X))]=∫_{X×Y} L(Y,f(X))P(x,y)dxdy 即期望风险最小化 → { 经验风险最小化：min 1/N Σ_{i=1}^{N} L(y_i,f(x_i))
如 极大似然估计
结构风险最小化：min 1/N Σ_{i=1}^{N} L(y_i,f(x_i))+λJ(f)
如 贝叶斯估计

3、模型选择：



实际中需要泛化误差来评价泛化能力。
泛化误差 即期望风险：R_exp(f̂)=E_p[L(Y,f̂(X))]
=∫_{X×Y} L(Y,f̂(x))P(x,y)dxdy
即经验风险：R_emp(f̂)=1/N Σ_{i=1}^{N} L(y_i,f̂(x_i))

→ 模型选择：{ 正则化：选择经验风险与模型复杂度同时较小的模型
交叉验证：{ 简单交叉验证
S折交叉验证
留一交叉验证(S=N)

模型f的泛化误差与训练误差的关系：R_exp(f̂)≤R_emp(f̂)+ε(d,N,δ)
其中，ε(d,N,δ)=√(1/2N (log d+log 1/δ))
{ N→∞ 即样本容量→∞ ⇒ 泛化误差→0
d↑即假设空间↑ ⇒ 泛化误差↑

生成方法与判别方法，分类问题，标注问题与回归问题。

- 生成方法：由 $P(X,Y)$ 间接求 $P(Y|X)$，如 NB、HMM
  - 可还原 $P(X,Y)$
  - 收敛速度快
  - 可存在隐变量
- 判别方法：直接求 $f(X)$ 或 $P(Y|X)$
  - 准确率高
  - 简化学习问题

- 分类问题：输入离散/连续，输出离散
  - $Y=\hat{f}(X)$ 或 $\hat{P}(Y|X)$ → 文本分类
- 标注问题：输入、输出均为变量序列，如 HMM、CRF
  - $Y=\hat{f}(X)$ 或 $\hat{P}(Y|X)$ → 词性标注
- 回归问题：输入、输出连续 $Y=\hat{f}(X)$，常用平方损失函数 ⇒ 最小二乘法 → 股价预测

	Relevant	NonRelevant
Retrieved	true positives (tp)	false positives (fp)
Not Retrieved	false negatives (fn)	true negatives (tn)

- 精确率 $Precision = \frac{tp}{tp+fp}$
- 召回率 $Recall = \frac{tp}{tp+fn}$
- F1 值：$\frac{2}{F1} = \frac{1}{Precision} + \frac{1}{Recall}$

LDA与PCA

1. LDA的输入带标签，监督学习，也称 Fisher ——> 使投影后的方差↓（类内）

对 $y=w^Tx$ 有

- 类别 i 的原始中心点：$m_i=\frac{1}{n_i}\sum_{x\in D_i}x$
- 类别 i 投影后的中心点：$\tilde{m}_i=w^Tm_i \Rightarrow |\tilde{m}_1-\tilde{m}_2|^2=w^T(m_1-m_2)(m_1-m_2)^Tw=w^TS_Bw$
- 投影后的方差：$\tilde{s}_i^2=\sum_{y\in Y_i}(y-\tilde{m}_i)^2=\sum_{x\in D_i}(w^Tx-w^Tm_i)^2=\sum_{x\in D_i}w^T(x-m_i)(x-m_i)^Tw=w^TS_iw$

损失函数 $J(w)=\frac{|\tilde{m}_1-\tilde{m}_2|^2}{\tilde{s}_1^2+\tilde{s}_2^2}$（分子↑，分母↓）$\Rightarrow$ 类间距离↑，类内距离↓

$\tilde{s}_1^2+\tilde{s}_2^2=w^T(S_1+S_2)w=w^TS_ww$

$=\frac{w^TS_Bw}{w^TS_ww}$

拉格朗日乘子法：$J(w)=\frac{w^TS_Bw}{w^TS_ww}$↑ 等价于 $\begin{cases}J(w)=w^TS_Bw\uparrow\\ w^TS_ww=1\end{cases}$

所以 $C(w)=w^TS_Bw-\lambda(w^TS_ww-1)\Rightarrow\frac{dC}{dw}=2S_Bw-2\lambda S_ww=0\Rightarrow S_Bw=\lambda S_ww$ ⇒求特征值.

对于多分类问题，$\begin{cases}S_B=\sum_{i=1}^{c}n_i(m_i-m)(m_i-m)^T\\ S_w=\sum_{i=1}^{c}S_i\end{cases}\Rightarrow S_Bw=\lambda S_ww$ ⇒求特征值

其中 w_i 对应于特征值中前 d 大特征向量！

2. PCA的输入不带标签，无监督学习 ——> 使投影后的方差↑（数据）

$\bar{x}=\frac{1}{N}\sum_{n=1}^{N}x_n$.

假设 u_1 为投影方向，则投影后的方差 $\frac{1}{N}\sum_{n=1}^{N}(u_1^Tx_n-u_1^T\bar{x})^2=u_1^T\cdot\frac{1}{N}\sum_{n=1}^{N}(x_n-\bar{x})(x_n-\bar{x})^T\cdot u_1=u_1^TSu_1$

拉格朗日乘子法：$\begin{cases}J(u_1)=u_1^TSu_1\uparrow\\ u_1^Tu_1=1\ (已知)\end{cases}$ 所以 $C(u_1)=u_1^TSu_1+\lambda_1(u_1^Tu_1-1)\Rightarrow\frac{dC}{du_1}=2Su_1-2\lambda_1u_1=0$

$\Rightarrow Su_1=\lambda_1u_1$ ⇒求特征值，因为 $\|u_1\|=1$ 所以 u_1 为特征向量，λ_1 为特征值。若 $J(u_1)\uparrow$ 即 $|S|\uparrow\Rightarrow\lambda_1\uparrow$

所以PCA要求当 D 维数据空间对应投影到 M 维数据空间（$M<D$），则取前 M 个特征向量构成的投影矩阵，即能够使投影后的方差最大的矩阵。

感知机与支持向量机（二类分类）

1、函数间隔与几何间隔

函数间隔：$\hat{\gamma}_i = y_i(w \cdot x_i + b)$ ⇒ 超平面函数间隔：$\hat{\gamma} = \min\limits_{i=1,2,\cdots,N} \hat{\gamma}_i$

几何间隔：$\gamma_i = y_i\left(\frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|}\right)$ ⇒ 超平面几何间隔：$\gamma = \frac{\hat{\gamma}}{\|w\|} = \min\limits_{i=1,2,\cdots,N} \frac{\hat{\gamma}_i}{\|w\|}$

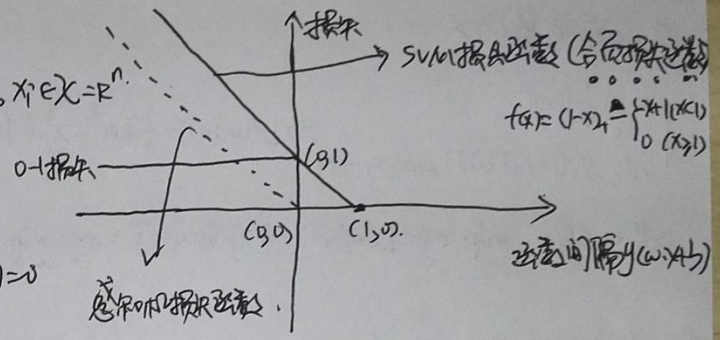
2、感知机与支持向量机的输入、输出

输入：$T = \{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，$x_i \in \mathcal{X} = R^n$

$y_i \in \mathcal{Y} = \{-1,+1\}$，$i=1,2,\cdots,N$.

输出：分离超平面：$w^* \cdot x + b^* = 0$

决策函数：$f(x) = \text{sign}(w^* \cdot x + b^*)$



$f(x) = (1-x)_+ = \begin{cases} 1-x & (x<1) \\ 0 & (x>1) \end{cases}$

3、感知机学习：误分类最小求分离超平面，解不唯一。

误分类点

$$\min_{w,b} -\sum_{x_i \in M} y_i\left(\frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|}\right) \Rightarrow \min_{w,b} L(w,b)$$，且 $L(w,b) = -\sum_{x_i \in M} y_i(w \cdot x_i + b)$ $\xrightarrow{\text{梯度下降法}}$ $\begin{cases} w^* = \sum_{i=1}^{N} \alpha_i y_i x_i \\ b^* = \sum_{i=1}^{N} \alpha_i y_i \end{cases}$

梯度下降法学习过程：

初始化 w, b

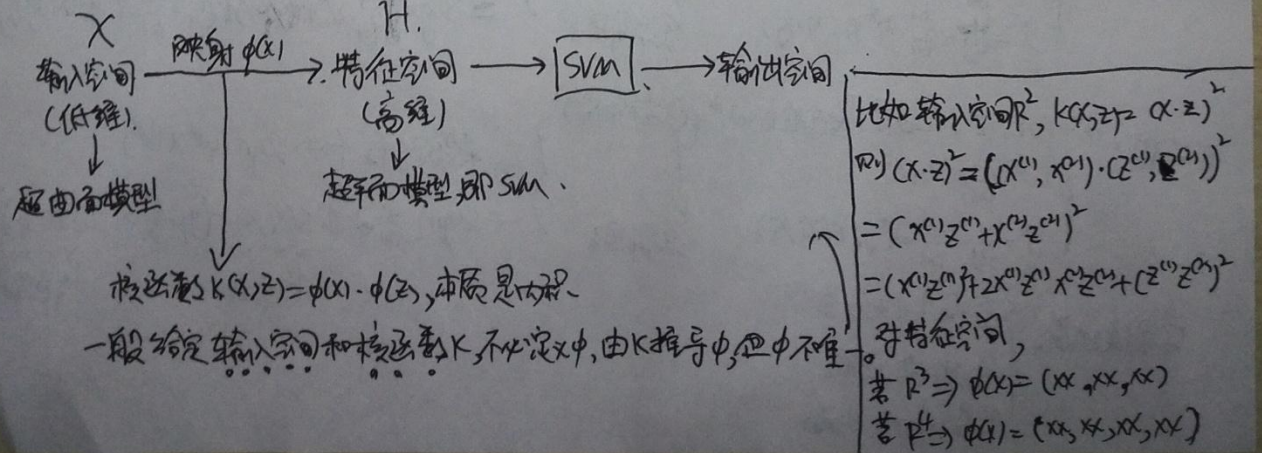
选取 (x_i, y_i) → 如果 $y_i(w \cdot x_i + b) > 0$，不更新 w, b

如果 $y_i(w \cdot x_i + b) \le 0$，则 $\begin{cases} w \leftarrow w + \eta y_i x_i \\ b \leftarrow b + \eta y_i \end{cases}$（$\because \begin{cases} \nabla_w L(w,b) = -y_i x_i \\ \nabla_b L(w,b) = -y_i \end{cases}$）

直至训练集没有误分类点

4、支持向量机：间隔最大化求分离超平面，解唯一。

超平面几何间隔



比如输入空间 R^2，$K(x,z) = (x \cdot z)^2$

则 $(x \cdot z)^2 = (x^{(1)}, x^{(2)}) \cdot (z^{(1)}, z^{(2)})^2$

$= (x^{(1)}z^{(1)} + x^{(2)}z^{(2)})^2$

$= (x^{(1)}z^{(1)})^2 + 2x^{(1)}z^{(1)}x^{(2)}z^{(2)} + (z^{(2)}z^{(2)})^2$

特征空间，

若 R^3 ⇒ $\phi(x) = (xx, xx, xx)$

若 R^4 ⇒ $\phi(x) = (xx, xx, xx, xx)$

支持向量机 {数据线性可分 ⇒ 硬间隔最大化 / 数据近似线性可分 ⇒ 软间隔最大化 / 数据线性不可分 ⇒ 核函数及软间隔最大化} {线性支持向量机

超平面及间隔

$$\max_{w,b}\min_{i=1,2,\cdots,N} y_i\left(\frac{w}{\|w\|}\cdot x_i+\frac{b}{\|w\|}\right) \Rightarrow \begin{cases}\max\limits_{w,b}\ \hat{\gamma}/\|w\| \\ \text{s.t. } y_i(w\cdot x_i+b)\geq\hat{\gamma},\ i=1,2,\cdots,N.\end{cases}$$

其中 $\hat{\gamma}=\min\limits_{i=1,2,\cdots,N} y_i(w\cdot x_i+b)$ ⟹ 分情况讨论.

① 硬间隔最大化

$$\begin{cases}\min\limits_{w,b}\ \frac{1}{2}\|w\|^2 \\ \text{s.t. } y_i(w\cdot x_i+b)\geq 1,\ i=1,2,\cdots,N\end{cases} \Rightarrow L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i y_i(w\cdot x_i+b)+\sum_{i=1}^{N}\alpha_i$$

$\alpha^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$ 且存在 $\alpha_j^*>0$ (对偶问题)

⇒ 原始问题: $\min\limits_{w,b}\max\limits_{\alpha} L(w,b,\alpha)$ ⇒ 对偶问题: $\max\limits_{\alpha}\min\limits_{w,b} L(w,b,\alpha)$ ⇒

$$\begin{cases} w^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i \\ b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i (x_i\cdot x_j)\end{cases}$$

② 软间隔最大化 ⇒ 惩罚因子

$$\begin{cases}\min\limits_{w,b,\xi}\ \frac{1}{2}\|w\|^2+C\sum\limits_{i=1}^{N}\xi_i \\ \text{s.t. } y_i(w\cdot x_i+b)\geq 1-\xi_i \\ \xi_i\geq 0,\ i=1,2,\cdots,N.\end{cases} \Rightarrow L(w,b,\xi,\alpha,u)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\alpha_i[y_i(w\cdot x_i+b)-1+\xi_i]-\sum_{i=1}^{N}u_i\xi_i$$

松弛因子

$\alpha^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$ 且存在 $0<\alpha_j^*<C$ (软间隔最大)

⇒ 原始问题: $\min\limits_{w,b,\xi}\max\limits_{\alpha,u} L(w,b,\xi,\alpha,u)$ ⇒ 对偶问题 $\max\limits_{\alpha,u}\min\limits_{w,b,\xi} L(w,b,\xi,\alpha,u)$ ⇒

$$\begin{cases} w^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i \\ b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i (x_i\cdot x_j)\end{cases}$$

③ 核函数及软间隔最大化, 核函数 $K(x_i,x_j)=\phi(x_i)\cdot\phi(x_j)$ 正定核!!!

$x_i\cdot x_j$ 映射 $\phi(x_i)\cdot\phi(x_j)=K(x_i,x_j)$, 核函数使 SVM 学习隐式地在特征空间进行。

以前解: $\alpha^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$ 且存在 $0<\alpha_j^*<C$ (支持向量)

$$\begin{cases} w^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i \\ b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i (x_i\cdot x_j)\end{cases}$$

惩罚因子

⇒ 决策函数 $f(x)=\text{sign}(w^*\cdot x+b^*)$

$$=\text{sign}\left[\sum_{i=1}^{N}\alpha_i^* y_i (x_i\cdot x)+\left(y_j-\sum_{i=1}^{N}\alpha_i^* y_i (x_i\cdot x_j)\right)\right]$$

现在解: $\alpha^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$ 且存在 $0<\alpha_j^*<C$ (支持向量)

$$\begin{cases} w^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i \\ b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i K(x_i,x_j)\end{cases}$$

惩罚因子

⇒ 决策函数 $f(x)=\text{sign}(w^*\cdot x+b^*)$

$$=\text{sign}\left[\sum_{i=1}^{N}\alpha_i^* y_i K(x_i,x)+\left(y_j-\sum_{i=1}^{N}\alpha_i^* y_i K(x_i,x_j)\right)\right]$$

核函数 核函数 核函数

常用核函数 {多项式核函数. / 高斯核函数 / 字符串核函数.}

朴素贝叶斯方法（多类分类）→ 生成模型，因为 $P(Y|X)=\frac{P(X,Y)}{P(X)}$

1. 朴素贝叶斯法的学习与分类

已知：先验概率 $P(Y)$，条件概率 $P(X|Y)$，求后验概率 $P(Y|X)$

$$P(Y|X)=\frac{P(X,Y)}{P(X)}=\frac{P(Y)P(X|Y)}{P(X)}=\frac{P(Y)\prod_j P(x^{(j)}|Y)}{P(X)}$$

（Bayes 定理；条件独立性假设：使模型中条件概率的数量少）

具体：$P(Y=c_k)$，$P(X=x|Y=c_k)$

$$P(Y=c_k|X=x)=\frac{P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k)}{P(X)}$$

因为后验概率最大化等价于在选择 0-1 损失函数时的期望风险最小化，

所以求使得后验概率最大的输出 $y=\arg\max_{c_k} P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k)$

2. 朴素贝叶斯法的参数估计

极大似然估计（等价于经验风险最小化）

1. 先验概率的极大似然估计 $P(Y=c_k)=\frac{\sum_{i=1}^N I(y_i=c_k)}{N}$

   条件概率的极大似然估计 $P(X^{(j)}=a_{jl}|Y=c_k)=\frac{\sum_{i=1}^N I(x_i^{(j)}=a_{jl},y_i=c_k)}{\sum_{i=1}^N I(y_i=c_k)}$

2. 对于给定的实例 $x=(x^{(1)},x^{(2)},\dots,x^{(n)})^T$，计算

   $P(Y=c_k)\prod_{j=1}^n P(X^{(j)}=x^{(j)}|Y=c_k)$

3. 确定实例 x 的类：$y=\arg\max_{c_k} P(Y=c_k)\prod_{j=1}^n P(X^{(j)}=x^{(j)}|Y=c_k)$

贝叶斯估计（等价于结构风险最小化）

1. 先验概率的贝叶斯估计 $P_\lambda(Y=c_k)=\frac{\sum_{i=1}^N I(y_i=c_k)+\lambda}{N+K\lambda}$（$K$ 为 Y 类样本可能取值的个数）

   条件概率的贝叶斯估计

   $$P_\lambda(X^{(j)}=a_{jl}|Y=c_k)=\frac{\sum_{i=1}^N I(x_i^{(j)}=a_{jl},y_i=c_k)+\lambda}{\sum_{i=1}^N I(y_i=c_k)+S_j\lambda}$$

   S_j 为第 j 个特征可能的取值个数

2. 同上。

最大熵模型与Logistic回归（统计学习）

1、最大熵原理认为，熵最大的模型是最好的模型。是概率模型学习或估计的一个准则！

- 变量X的熵：$H(X)=-\sum_{i=1}^{n} p_i \log p_i$ 或 $H(p)=-\sum_{i=1}^{n} p_i \log p_i$
- 给定X，Y的条件熵：$H(Y|X)=\sum_{i=1}^{n} p_i \cdot H(Y|X=x_i)=\sum_{i=1}^{n} p_i\left[-\sum_{j=1}^{m} P(y_j|x_i)\log(y_j|x_i)\right]$

$$=-\sum_{i=1}^{n}\sum_{j=1}^{m} p_i \cdot P(y_j|x_i)\log P(y_j|x_i)$$

或 $H(p)=-\sum_{i=1}^{n}\sum_{j=1}^{m} p_i \cdot P(y_j|x_i)\log P(y_j|x_i)$

- 模型$P(Y|X)$的条件熵：$H(p)=-\sum_{x,y}\tilde{P}(x)\cdot P(y|x)\log P(y|x)$

最大熵模型

$$\begin{cases}\max\limits_{P\in C} H(p)=-\sum\limits_{x,y}\tilde{P}(x)P(y|x)\log P(y|x)\\ s.t.\ E_p(f_i)=E_{\tilde{p}}(f_i),\ i=1,2,\cdots,n \text{ 即 } \sum\limits_{x,y}\tilde{P}(x)P(y|x)f_i(x,y)=\sum\limits_{x,y}\tilde{P}(x,y)f_i(x,y)\\ \sum\limits_{y}P(y|x)=1\end{cases}$$

$$\Updownarrow$$

$$\begin{cases}\min\limits_{P\in C} -H(p)=\sum\limits_{x,y}\tilde{P}(x)P(y|x)\log P(y|x)\\ s.t.\ E_p(f_i)-E_{\tilde{p}}(f_i)=0,\ i=1,2,\cdots,n\\ \sum\limits_{y}P(y|x)=1\end{cases}$$

$$L(P,w)\equiv -H(p)+w_0\left(1-\sum_{y}P(y|x)\right)+\sum_{i=1}^{n} w_i\left(E_p(f_i)-E_{\tilde{p}}(f_i)\right)$$

原始问题：$\min\limits_{P\in C}\max\limits_{w} L(P,w)$

对偶问题：$\max\limits_{w}\min\limits_{P\in C} L(P,w)$ ⇒

① 求解 $\min\limits_{P\in C} L(P,w)$ 即 $L(P,w)$ 关于 P 的最小化

即求最大熵模型公式 $P_w(y|x)$，有 $\min\limits_{P\in C} L(P,w)=L(P_w,w)$

② 求解 $\max\limits_{w} L(P_w,w)$ 即 $L(P_w,w)$ 关于 w 的最大化

即最大熵模型的学习，得到 w^*

⇒ 最终模型 $P_{w^*}(y|x)$

$$P_w(y|x)=\frac{1}{Z_w(x)}\exp\left(\sum_{i=1}^{n} w_i f_i(x,y)\right)$$

其中 $Z_w(x)=\sum_{y}\exp\left(\sum_{i=1}^{n} w_i f_i(x,y)\right)$

此步等价于 $P_w(y|x)$ 的极大似然估计 $L_{\tilde{p}}(P_w)=\log\prod_{x,y}P_w(y|x)^{\tilde{P}(x,y)}=\sum_{x,y}\tilde{P}(x,y)\log P_w(y|x)$

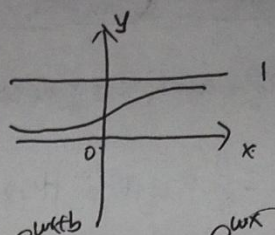
因此最大熵模型的学习，归结为极大似然估计问题！

求解极大似然估计问题，得 w^* ：

- 迭代尺度法
- 梯度下降法
- 牛顿法或拟牛顿法

2. Logistic回归

Logistic分布 $f(x)=\frac{1}{1+e^{-\theta^T x}}$ $\begin{cases} x\to+\infty \Rightarrow f(x)\to 1 \\ x\to-\infty \Rightarrow f(x)\to 0 \end{cases}$ $\Rightarrow$



假设 $P(Y|X)$ 服从Logistic分布，有 $\begin{cases} z(x)=P(Y=1|X)=\frac{e^{w\cdot x+b}}{1+e^{w\cdot x+b}}\to\frac{e^{w\cdot x}}{1+e^{w\cdot x}} \\ 1-z(x)=P(Y=0|X)=\frac{1}{1+e^{w\cdot x+b}}\to\frac{1}{1+e^{w\cdot x}} \end{cases} \Rightarrow \frac{z(x)}{1-z(x)}=e^{w\cdot x}$

学习问题采用极大似然估计，有 $L(w)=\log\prod_{i=1}^{n}[z(x_i)]^{y_i}\cdot[1-z(x_i)]^{1-y_i}$

$$=\sum_{i=1}^{n}\left(y_i\log z(x_i)+(1-y_i)\log[1-z(x_i)]\right)$$

$$=\sum_{i=1}^{n}\left(y_i\log\frac{z(x_i)}{1-z(x_i)}+\log[1-z(x_i)]\right)$$

$$=\sum_{i=1}^{n}\left[y_i(w\cdot x_i)-\log(1+e^{w\cdot x_i})\right]$$

求使 $L(w)$ 取得极大值的 w $\Rightarrow$ 梯度下降法 / 牛顿法或拟牛顿法

梯度下降法：$\frac{\partial L}{\partial w_i}=\frac{dL}{dw}\cdot\frac{\partial w}{\partial w_i}$

$$=\sum_{i=1}^{n}\left(y_i\cdot x_i-\frac{1}{1+e^{w\cdot x_i}}\cdot e^{w\cdot x_i}\cdot x_i\right)\cdot\frac{\partial w}{\partial w_i}$$

$$=\sum_{i=1}^{n}(y_i x_i-z(x_i)\cdot x_i)\cdot\frac{\partial w}{\partial w_i}=\sum_{i=1}^{n}(y_i-z(x_i))x_i\cdot\frac{\partial w}{\partial w_i}$$

$$=(y_i-z(x_i))x_i$$

$\therefore w_i\leftarrow w_i+\lambda\cdot(y_i-z(x_i))x_i$

K近邻法（分类）

1、K近邻算法 {输入：$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$；实例特征向量 x
输出：实例 x 所属的 y

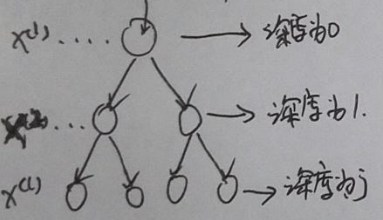
(1) 根据给定的距离度量，在训练集 T 中找出与 x 最近邻的 k 个点，记作 $N_k(x)$

(2) 在 $N_k(x)$ 中根据分类决策规则（如多数表决）决定 x 的类别 y：$y=\arg\max_{c_j}\sum_{x_i\in N_k(x)} I(y_i=c_j)$

三要素：{距离度量 $\Rightarrow L_p(x_i,x_j)=\left(\sum_{l=1}^{n}|x_i^{(l)}-x_j^{(l)}|^p\right)^{\frac{1}{p}}$，欧氏距离 $L_2(x_i,x_j)=\left(\sum_{l=1}^{n}|x_i^{(l)}-x_j^{(l)}|^2\right)^{\frac{1}{2}}$
K值的选择 $\Rightarrow$ {K小，基于最近邻 $\Rightarrow$ 模型复杂，过拟合
K个 $\Rightarrow$ 模型简单，欠拟合。 $\Rightarrow$ 交叉验证求K
分类决策规则 $\Rightarrow$ 多数表决等价于经验0-1损失下经验风险最小化。

2、K近邻的实现：{线性扫描（相当于全局搜索）
kd树（相当于局部搜索）：适用于训练实例 >> 空间维数K（与K近邻中K含义不同）

构造kd树：{输入：$T=\{x_1,x_2,\cdots,x_N\}$，其中 $x_i=(x_i^{(1)},x_i^{(2)},\cdots,x_i^{(k)})^T$
输出：kd树
（结点中保存实例）



之后 $x^{(1)},x^{(2)},\cdots,x^{(k)}$，分别对其坐标轴按坐标的中位数（平衡）进行超平面划分，将落在切分超平面上的实例点保存在对应结点。直到两个子区域都没有实例存在时停止。

其中 l,j 关系：$l=j(\mathrm{mod}\,k)+1$
kd树最深为K层（因为K维空间）

用kd树的最近邻搜索：{输入：kd树，目标点 x
输出：x 的最近邻

从根结点出发，递归向下访问kd树，直到叶结点为止，以此为"当前最近点"。
开始局部搜索：回退，重新"不断更新""当前最近点"
再看是否与另子结点相交，{相交 → 子搜索。
不相交 → 回退。
直到根结点，结束，返回"当前最近点"。

决策树（多类分类）

1. 决策树：树形结构。内部结点对应特征或属性，叶结点对应类。

特征选择：

信息增益：$g(D,A)=H(D)-H(D|A)=\left[-\sum_{k=1}^{K}\frac{|C_k|}{|D|}\log\frac{|C_k|}{|D|}\right]-\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)$

$=\left[-\sum_{k=1}^{K}\frac{|C_k|}{|D|}\log\frac{|C_k|}{|D|}\right]-\left[-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D_i|}\log\frac{|D_{ik}|}{|D_i|}\right]$

信息增益比：$g_R(D,A)=\frac{g(D,A)}{H_A(D)}=\frac{g(D,A)}{-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\log\frac{|D_i|}{|D|}}$

$|C_k|$：D中属于类C_k的样本集合

$|D_i|$：D中属于类C_k的样本集合

决策树学习

决策树的生成（局部最优）：

ID3算法：信息增益选择特征

C4.5算法：信息增益比选择特征

⇒ 从根结点开始，递归地产生决策树。

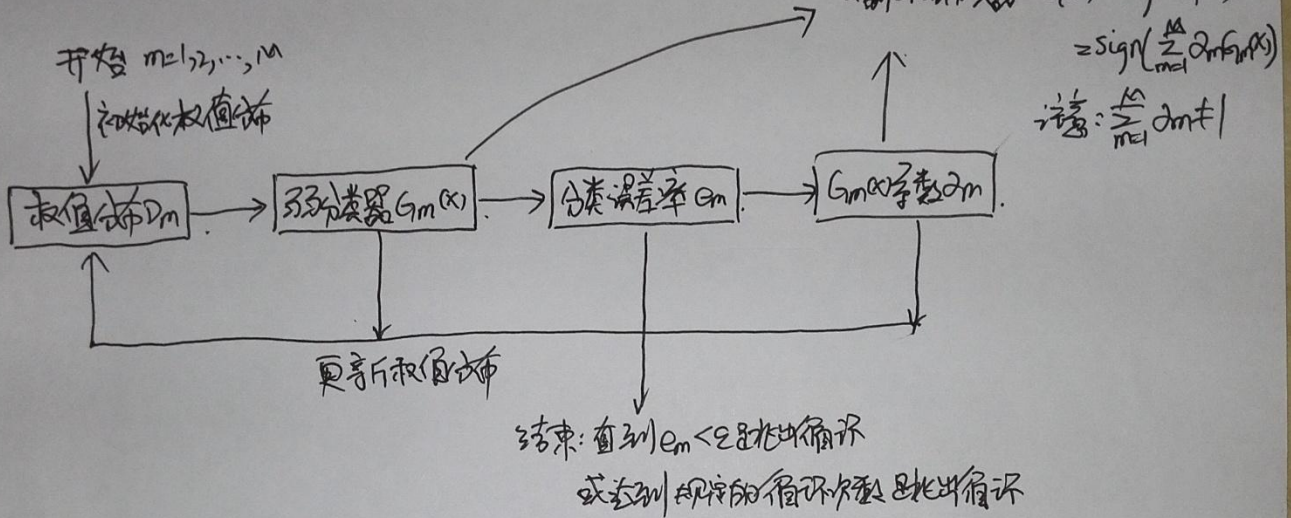
决策树的剪枝（全局最优）：解决过拟合问题。

AdaBoost算法（二类分类）

对于二分类问题：

输入：$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，$x_i\in\mathcal{X}\subseteq\mathbb{R}^n$，$y_i\in\mathcal{Y}=\{-1,+1\}$

输出：最终分类器$G(x)$



其中，$e_m=P(G_m(x_i)\neq y_i)=\sum_{i=1}^{N}w_{mi}I(G_m(x_i)\neq y_i)$，$w_{mi}$——第$m$轮中第$i$个实例的权值，$\sum_{i=1}^{N}w_{mi}=1$

$\alpha_m=\frac{1}{2}\log\frac{1-e_m}{e_m}$ $\longrightarrow$ $e_m\downarrow\Rightarrow\alpha_m\uparrow$，分类误差率↓的$G_m(x)$作用↑.

$w_{m+1,i}=\frac{w_{mi}}{Z_m}\exp(-\alpha_m y_i G_m(x_i))$，$Z_m=\sum_{i=1}^{N}w_{mi}\exp(-\alpha_m y_i G_m(x_i))$

$=\begin{cases}\frac{w_{mi}}{Z_m}e^{-\alpha_m} & (\text{当 } G_m(x_i)=y_i \text{ 时})\longrightarrow\alpha_m\uparrow\Rightarrow w_{m+1,i}\downarrow\text{，被}G_m(x)\text{正确分类的样本权重}\downarrow\\ \frac{w_{mi}}{Z_m}e^{\alpha_m} & (\text{当 } G_m(x_i)\neq y_i \text{ 时})\longrightarrow\alpha_m\uparrow\Rightarrow w_{m+1,i}\uparrow\text{，被}G_m(x)\text{错误分类的样本权重}\uparrow\end{cases}$