【公式】①$x^Ta=a^Tx=a$；②$a^TXb=ab^T$；
③$a^TX^Tb=ba^T$；④$a^TX^Ta=a^TXa=aa^T$；
⑤$b^TX^TXc=X(bc^T+cb^T)$
⑥$(Bx+b)^TC(Dx+d)=B^TC(Dx+d)+D^TC^T(Bx+b)$
⑦$x^TBx=(B+B^T)x$ ⑧$b^TX^TDXc=D^TXbc^T+DXcb^T$
⑨$(Xb+c)^TD(Xb+c) = (D+D^T)(Xb+c)b^T$

【定义】如果它在任务 T 中的性能被 P 测试,随着经验 E 的增加而提高,则被认为是计算机程序是从经验 E 中学习某些类别的任务 T 和性能度量 P【监督】利用一组已知类别的样本调整分类器的参数，使其达到所要求性能的过程(分类、回归)【非监督】根据类别未知(没有被标记)的训练样本的数据分析实现对样本分类的一种数据处理方法(聚类、密度估计)【分类】将实例数据划分到合适的分类中【回归】预测数值型数据；区别：分类模型的输出是离散的回归模型的输出是连续的。联系：分类预测连续值，但是连续值是类标签的概率形式。 回归预测离散值，但是以整数量的形式预测离散值。分类和回归可以互相转换（预测数量转换为离散桶/对类值排序映射到连续范围）【监督学习框架/机器学习三阶段】训练：使用标记过的训练样本让机器学习模型；测试：使用未经训练的测试样本评估模型表现；应用【假设-学习-决策】具有(未知)参数(或结构)的数学模型;用最大似然估计 MLE、MAP、贝叶斯估计、损失函数优化;贝叶斯决策、直接预测函数，预测函数使得期望效益最大化【核心概念三要素】模型：我们要求的可以由输入产生正确输出的函数或者概率模型；策略：即考虑用什么样的准则学习；算法：指学习模型的具体方法，如何与优化损失函数。例如梯度下降法等【准则-损失函数】感知机;最小均方误差-均方差函数；最大间隔-Hinge Loss；最小交叉熵 CE-交叉熵损失函数；最大似然估计-~~似然函数 L(θ)~~

【线性回归 LMS 求解】假设 $h_\theta(x)=\theta_0+\sum\theta_i*x_i \leftrightarrow \theta^Tx$；代价函数 $J(\theta)=1/2\sum(h_\theta(x^{(k)})-y^{(k)})^2$；目标:$\theta^*=\arg_\theta \min J(\theta)$;令 $X=[(x^{(1)})^T]$;$y=[y^{(1)}]$;则 $J(\theta)=1/2(X\theta-y)^T(X\theta-y)$;由公式⑥ $\theta^*=(X^TX)^{-1}X^Ty$ 求逆复杂度高难计算/求不出

【梯度下降】优化过程：从初始位置开始(初始参数$\theta^{(0)}$)；在当前位置$\theta^{(t)}$重复直到收敛：1.计算当前位置梯度$\nabla_\theta f(\theta)|_{\theta=\theta(t)}$；2.沿梯度反方向移动到下一个位置$\theta^{(t+1)}=\theta^{(t)}-\alpha\cdot\nabla_\theta f(\theta)|_{\theta=\theta(t)}$ α是学习率；3.t=t+1；
$dJ(\theta)/d\theta = \sum(h_\theta(x^{(k)})-y^{(k)})\cdot x^{(k)}$
GD 优化：$\theta:=\theta-\alpha\cdot(dJ(\theta)/d\theta)$

【Logistic】$\delta(z)=1/(1+e^{-z})$
假设 $p(y=1|x;\theta)=h_\theta(x)=\delta(\theta^Tx)$
$=1/(1+e^{\wedge}(-\theta^Tx))$
$p(y=0|x;\theta)=1-p(y=1|x;\theta)$
化简=>$p(y|x;\theta)=(h_\theta(x))^y(1-h_\theta(x))^{(1-y)}$
似然函数 $L(\theta)=\prod(h_\theta(x^{(i)}))^{y(i)}(1-h_\theta(x^{(i)}))^{(1-y(i))}$
最大似然估计 $\max L(\theta)\Leftrightarrow\max\sum y^{(i)}\log h_\theta(x^{(i)})+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))$ 负对数似然函数与交叉熵损失函数等价=>无约束优化问题：
$l = \min\sum-y^{(i)}\log h_\theta(x^{(i)})-(1-y^{(i)})\log(1-h_\theta(x^{(i)}))$
$dl/d\theta = \sum-y^{(i)}\cdot(\nabla_\theta h_\theta(x^{(i)})/h_\theta(x^{(i)}))+(1-y^{(i)})\cdot(\nabla_\theta h_\theta(x^{(i)})/1-h_\theta(x^{(i)}))=\sum(-y^{(i)}\cdot(1-h_\theta(x^{(i)}))+(1-y^{(i)})h_\theta(x^{(i)}))\cdot x^{(i)}=\sum(h_\theta(x^{(i)})-y^{(i)})\cdot x^{(i)}$
(误差 * 输入)
其中$\nabla_\theta h_\theta(x^{(i)})=h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))\nabla_\theta\theta^T$
$=h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))\cdot x^{(i)}$
GD 梯度优化$\theta:=\theta-\alpha\cdot\sum(h_\theta(x^{(i)})-y^{(i)})\cdot x^{(i)}$
SGD 随机梯度下降：随意选训练样本(x,y)计算梯度成分$(h_\theta(x)-y)x$;更新$\theta:=\theta-\alpha\cdot(h_\theta(x)-y)x$

【Softmax】假设：

$$p(y=j|x;\theta)=h_j(x)=\frac{e^{\theta_j{}^Tx}}{\sum_{j'=1}^{C}e^{\theta_{j'}{}^Tx}},j=1,2,\dots,C,\text{where }\theta_C=\vec{0}$$

似然函数：

$$l(\theta)=\prod_{i=1}^{N}p(y^{(i)}|x^{(i)};\theta)$$
$$=\prod_{i=1}^{N}\prod_{j=1}^{C}\left(p(y^{(i)}|x^{(i)};\theta)\right)^{1\{y^{(i)}=j\}}$$

对数似然函数：

$$l(\theta)=\sum_{i=1}^{N}\log\prod_{j=1}^{C}\left(\frac{e^{\theta_j{}^Tx}}{\sum_{j'=1}^{C}e^{\theta_{j'}{}^Tx}}\right)^{1\{y^{(i)}=j\}}$$
$$=\sum_{i=1}^{N}\sum_{j=1}^{C}1\{y^{(i)}=j\}\log\left(\frac{e^{\theta_j{}^Tx}}{\sum_{j'=1}^{C}e^{\theta_{j'}{}^Tx}}\right)$$
$$=\sum_{i=1}^{N}\sum_{j=1}^{C}1\{y^{(i)}=j\}\log h_j(x^{(i)})$$

梯度下降优化

$$\frac{\partial\log h_j(x)}{\partial\theta_k}=\begin{cases}(1-h_k(x))x, & j=k\\ -h_k(x)x, & j\neq k\end{cases}$$

$$\frac{\partial\sum_{j=1}^{C}1\{y=j\}\log h_j(x)}{\partial\theta_k}=\begin{cases}(1-h_k(x))x, & y=j=k\\ -h_k(x)x, & y=j\neq k\end{cases}$$
$$=(1\{y=k\}-h_k(x))x$$

$$\frac{\partial l(\theta)}{\partial\theta_k}=\sum_{i=1}^{N}\left(h_k(x^{(i)})-1\{y^{(i)}=k\}\right)x^{(i)}$$

更新：GD

$$\theta_k:=\theta_k-\alpha\sum_{i=1}^{N}\left(h_k(x^{(i)})-1\{y^{(i)}=k\}\right)x^{(i)}$$

$$\text{where }h_k(x)=\frac{e^{\theta_k{}^Tx}}{\sum_{k'=1}^{C}e^{\theta_{k'}{}^Tx}},k=1,2,\dots,C$$

SGD:

$$\theta_k:=\theta_k-\alpha(h_k(x)-1\{y=k\})x$$

【感知机】模型：$h_w(x)=\{1 \text{ if } w^Tx\geqslant0\}$
准则:感知机准则 $\{0 \text{ if } w^Tx<0\}$
损失函数：

$$J_P(x)=\sum_{x^{(i)}\epsilon M_0}\omega^Tx^{(i)}-\sum_{x^{(j)}\epsilon M_1}\omega^Tx^{(j)}$$
$$=\sum_{i=1}^{N}\left((1-y^{(i)})h_w(x^{(i)})-y^{(i)}\left(1-h_w(x^{(i)})\right)\right)\omega^Tx^{(i)}$$
$$=\sum_{i=1}^{N}(h_w(x^{(i)})-y^{(i)})\omega^Tx^{(i)}$$

$dJ/dw=1/N(d/dw\sum(h_w(x^{(i)})-y^{(i)})\cdot w^Tx^{(i)})$
$=1/N\sum(h_w(x^{(i)})-y^{(i)})x^{(i)}$

GD $\omega:=\omega-\alpha\cdot(dJ/dw)$
SGD $\omega:=\omega-\alpha\cdot(h_w(x)-y)x$
$=\{\omega+\alpha x$ if y=1 and $h_w(x)=0\}$
$\{\omega-\alpha x$ if y=0 and $h_w(x)=1\}$
$\{\omega$ otherwise$\}$

【多分类感知机】
模型 $C^*=\arg\max_{j=1,\dots,C}\omega_j^Tx$

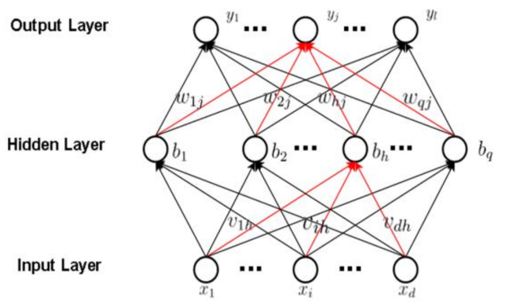
损失函数 $J_p(w)=\sum_{k=1}^{N}\left(\max_{j=1,\dots,C}\omega_j^Tx^{(k)}-\omega_{y^{(k)}}^Tx^{(k)}\right)$

更新规则

$$\omega_j:=\omega_j-\alpha\left(1\{j=c^{(k)}\}-1\{j=y^{(k)}\}\right)x^{(k)}$$
$$=\begin{cases}\omega_j-\alpha x^{(k)}, & \text{if } j=c^{(k)}\neq y^{(k)}\\ \omega_j+\alpha x^{(k)}, & \text{if } j=y^{(k)}\neq c^{(k)}\\ \omega_j, & \text{others}\end{cases}$$
$$\text{where }c^{(k)}=\arg\max_{j=1,\dots,C}\omega_j^Tx^{(k)}$$

感知机缺点：无法学习线性不可分函数

【ANN 三层前向神经网络】
模型假设：



$\hat{y}_j=\delta(\beta_j+\theta_j)$ $\beta_j=\sum w_{hj}b_h$
$b_h=\delta(\alpha_h+\gamma_h)$ $\alpha_h=\sum v_{ih}x_i$
训练集

$$D=\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),\dots,(x^{(m)},y^{(m)})\},\ x^{(i)}\in\mathbb{R}^d,y^{(i)}\in\mathbb{R}^l$$

损失函数 $E^{(k)}=\frac{1}{2}\sum_{j=1}^{l}\left(\hat{y}_j^{(k)}-y_j^{(k)}\right)^2$

参数：$v\in\mathbb{R}^{d*q},\gamma\in\mathbb{R}^q,\omega\in\mathbb{R}^{q*l},\theta\in\mathbb{R}^l$

核心思想：输入信号前向传播，误差信号反向传播，误差累计对应每个输出单元

梯度计算：$\frac{\partial E^{(k)}}{\partial v_{ih}},\frac{\partial E^{(k)}}{\partial\gamma_h},\frac{\partial E^{(k)}}{\partial\omega_{hj}},\frac{\partial E^{(k)}}{\partial\theta_j}$

$$\frac{\partial E^{(k)}}{\partial\omega_{hj}}=\frac{\partial E^{(k)}}{\partial\hat{y}_j^{(k)}}\cdot\frac{\partial\hat{y}_j^{(k)}}{\partial(\beta_j+\theta_j)}\cdot\frac{\partial(\beta_j+\theta_j)}{\partial\omega_{hj}}$$

其中①$=\hat{y}_j^{(k)}-y_j^{(k)}$
②$=\delta(\beta_j+\theta_j)(1-\delta(\beta_j+\theta_j))=\hat{y}_j^{(k)}(1-\hat{y}_j^{(k)})$
③$=b_h$
定义 $error_j^{outputLayer}$=①·②
则 w_{hj} 梯度$=error_j^{outputLayer}\cdot b_h$; 再对$\theta_j$:

$$\frac{\partial E^{(k)}}{\partial\theta_j}=\frac{\partial E^{(k)}}{\partial\hat{y}_j^{(k)}}\cdot\frac{\partial\hat{y}_j^{(k)}}{\partial(\beta_j+\theta_j)}\cdot\frac{\partial(\beta_j+\theta_j)}{\partial\theta_j}$$
$$=error_j^{OutputLayer}\cdot1$$

$$\frac{\partial E^{(k)}}{\partial v_{ih}}=\sum_{j=1}^{l}\frac{\partial E^{(k)}}{\partial(\beta_j+\theta_j)}\cdot\frac{\partial(\beta_j+\theta_j)}{\partial b_h}\cdot\frac{\partial b_h}{\partial(\alpha_h+\gamma_h)}\cdot\frac{\partial(\alpha_h+\gamma_h)}{\partial v_{ih}}$$

其中①$=error_j^{outputLayer}$ ②$=w_{hj}$
③$=\delta(\alpha_h+\gamma_h)(1-\delta(\alpha_h+\gamma_h))=b_h(1-b_h)$ ④$=x_i^{(k)}$
令 $error_h^{HiddenLayer}=\sum$①·②·③
则 $v_{ih}=error_h^{HiddenLayer}\cdot x_i^{(k)}$

$$\frac{\partial E^{(k)}}{\partial\gamma_h}=\sum_{j=1}^{l}\frac{\partial E^{(k)}}{\partial(\beta_j+\theta_j)}\cdot\frac{\partial(\beta_j+\theta_j)}{\partial b_h}\cdot\frac{\partial b_h}{\partial(\alpha_h+\gamma_h)}\cdot\frac{\partial(\alpha_h+\gamma_h)}{\partial\gamma_h}$$
$$=error_h^{HiddenLayer}\cdot1$$

更新$\omega hj:=\omega hj-\eta\cdot\partial E/\partial\omega hj;\theta j:=\theta j-\eta\cdot\partial E/\partial\theta j$
$vih=vih-\eta\cdot\partial E/\partial vih;\gamma h=\gamma h-\eta\cdot\partial E/\partial\gamma h$
算法流程图
Input:training set:$D=\{(x^{(k)},y^{(k)})\}^m_{k=1}$ 学习率η
Steps:1 initialize all parameters within(0,1)
2repeat 3for all$(x^{(k)},y^{(k)})\in D$do: 4calculate $y^{\wedge}(k)$
5cal erroroutputL 6 cal errorHidden 7update ω θ v γ 8end for 9until reach stop cond
梯度 = 前序误差 * 当前单元梯度,
误差从顶层向底层不断累积，反向传播

【支持向量机 SVM】超平面$\omega^Tx+b=0$
线性模型：$y(x)=\omega^Tx+b$
距离(正样本)$r=\omega^Tx+b/||\omega||$
距离(负样本)$r=-(\omega^Tx+b/||\omega||)$
则几何间隔 $r^{(k)}=y^{(k)}(\omega^Tx^{(k)}+b/||\omega||)$
$y^{(i)}=\{-1,+1\}$表 $x^{(k)}$的类别标签
函数距离 $r^{\wedge(k)}=y^{(k)}(\omega^Tx^{(k)}+b)=||\omega||r^{(k)}$
几何距离独立于比例因子函数距离成正比

【最大间隔准则】**形式 1：**

$$\max_{\omega,b}\gamma\ s.t.\quad y^{(k)}\left(\frac{\omega^Tx^{(k)}+b}{||\omega||}\right)\geq\gamma,k=1,\dots,N$$

$$I.e.\quad\gamma^{\wedge}=\min_{k=1\dots N}\gamma^{\wedge(k)}$$

形式 2：

$$\max_{\gamma^\wedge,\omega,b}\frac{\gamma^\wedge}{\|\omega\|}\ s.t.\quad y^{(k)}(\omega^T x^{(k)}+b)\ge\gamma^\wedge,k=1,\dots,N$$

$$I.e.\quad \gamma^\wedge=\min_{k=1\dots N}\gamma^{\wedge(k)}$$

形式 3：$\gamma^\wedge=\min\gamma^{\wedge(k)}=1$ 等价$\gamma=\min\gamma^{(k)}=1/||w||$,即缩放 w,b,使得$||w||=1/r$

$$\max_{\omega,b}\frac{1}{\|\omega\|}\ s.t.\quad y^{(k)}(\omega^T x^{(k)}+b)\ge1,k=1,\dots,N$$

$$\min_{\omega,b}\frac{1}{2}\|\omega\|^2\ s.t.\quad y^{(k)}(\omega^T x^{(k)}+b)\ge1,k=1,\dots,N$$

等价性：对于样本集中的每个样本点选取与超平面函数间隔最小的点，使这个间隔最大化

【拉格朗日函数】

等式约束优化：min f(x) s.t. g(x)=0 =>
$L(x,\lambda)=f(x)+\lambda g(x)\quad \lambda\neq0\ \{\partial L/\partial x=0\ g(x)=0\}$

不等式约束优化：min f(x) s.t. $g(x)\le0$ =>
$L(x,\lambda)=f(x)+\lambda g(x)\quad \lambda\neq0$

激活:$\{\partial L/\partial x=0\ g(x)=0\ \lambda>0\}$

非激活:$\{\partial L/\partial x=0\ g(x)<0\ \lambda=0\}$

=>$\{\partial L/\partial x=0\ g(x)\le0\ \lambda\ge0\}$

多重约束 min f(x) s.t. $h_i(x)=0,g_j(x)\le0$

$L(x,\lambda,\mu)=f(x)+\sum\lambda_i h_i(x)+\sum\mu_j g_j(x)$

KKT 条件

$\{\partial L/\partial x=0$ 稳定性条件$\}$

$\{h_i(x)=0;\ g_j(x)\le0$ 原问题可行性$\}$

$\{u_j\ge0$ 对偶可行性$\}\{u_jg_j(x)=0$ 互补条件$\}$

【原问题与最小最大问题等价性】

$$\max_{\alpha,\beta:\alpha_i\ge0}L(\omega,\alpha,\beta)=\max_{\alpha,\beta:\alpha_i\ge0}\left(f(\omega)+\sum_{i=1}^{k}\alpha_i g_i(\omega)+\sum_{j=1}^{l}\beta_j h_j(\omega)\right)$$

$$=\begin{cases}f(\omega) & if\ g_i(\omega)\le0,h_j(\omega)=0\\ \infty & otherwise\end{cases}$$

假如 $g(\omega)>0\ h(\omega)\neq0$ 则总可以调整α,β使得原式有最大值为正无穷 满足约束时为 f(ω) 原问题求 min f(ω)转换为 min max L(ω,α,β)

原 $\min_{\omega}\max_{\alpha,\beta;\alpha_i\ge0}L(\omega,\alpha,\beta)\ \le\ \max_{\alpha,\beta;\alpha_i\ge0}\min_{\omega}L(\omega,\alpha,\beta)$ 对偶

【原问题与对偶问题等价条件】1.f,gi 是凸函数，hi 是仿射变换(affine);2.gi(严格)可行:一定存在 w 使得 gi(w)<0;3.满足 KKT 条件.=>原问题和对偶问题等价

【SVM 求解】

$$\min_{\omega,b}\frac{1}{2}\|\omega\|^2\ s.t.\quad y^{(k)}(\omega^T x^{(k)}+b)\ge1,k=1,\dots,N$$

$g_k(\omega)=1-y^{(k)}(\omega^Tx^{(k)}+b)\le0$

$L(\omega,b,\alpha)=||\omega||^2/2-\Sigma\alpha_k(y^{(k)}(\omega^Tx^{(k)}+b)-1)$

原问题 min max L 对偶 max min L 先对 w b 参数求最小，在对α参数求最大

$\partial L/\partial\omega=\omega-\Sigma\alpha_ky^{(k)}x^{(k)}=0\ =>\omega=\Sigma\alpha_ky^{(k)}x^{(k)}$

$\partial L/\partial b=\Sigma\alpha_ky^{(k)}=0$

代入 $L=\Sigma\alpha_k-1/2\Sigma_k\Sigma_ly^{(k)}y^{(l)}\alpha_k\alpha_lx^{(k)T}x^{(l)}$

$x^{(k)T}x^{(k)}$写成内积$<x^{(k)},x^{(l)}>$

SVM 对偶问题 $\max_\alpha W(\alpha)\ \Sigma\alpha_k-1/2\Sigma_k\Sigma_ly^{(k)}y^{(l)}\alpha_k\alpha_l<x^{(k)},x^{(l)}>$ s.t. $\alpha_k\ge0$ $\Sigma\alpha_ky^{(k)}=0(k=1..N)$

【偏置项求解】$(\omega^*)^Tx1+b=1(\omega^*)^Tx2+b=-1$

$$b^*=-\frac{max_{i:y^{(i)}=-1}(\omega^*)^Tx_i+min_{i:y^{(i)}=1}(\omega^*)^Tx_i}{2}$$

【为什么叫支持向量】决策函数 $f(x)=(\omega^*)^Tx+b=(\sum\alpha_k^*y^{(k)}x^{(k)})^Tx+b^*=\sum\alpha_k^*y^{(k)}x^{(k)T}x+b^*$;KKT 条件$\alpha_k^*g_k(\omega^*)=0$

$gk(\omega^*)\le0,\alpha_k^*\ge0$ 其 $g_k(\omega)=-y^k(\omega^Tx^k+b)+1$

若$\alpha_k^*>0$ 则必有 $g_k(\omega^*)=0$ 即 $y^k(\omega^Tx^k+b)=1$

训练完成后，大部分的训练样本都不需要保留，最终模型仅与少量支持向量有关

问题:决策函数中α_k^*怎么计算,如何优化最大化

【SMO】把原始求解 N 个参数二次规划问题分解成很多个子二次规划问题分别求解，每个子问题只需要求解 2 个参数，将其他的变量都视为常数方法类似于坐标上升，节省时间成本和降低了内存需求。每次启发式选择两个变量进行优化，不断循环，直到达到函数最优值。

【软间隔 SVM】软间隔准则引入松弛变量ε,并使其最小,影响最小 $\min 1/2||\omega||^2+C\sum\varepsilon i$, s.t.$yi(\omega^Txi+b)\ge1-\varepsilon i$,$\varepsilon i\ge0,i=1\sim m$

$$L(\omega,b,\varepsilon,\alpha,\gamma)=\frac{1}{2}\omega^T\omega+C\sum_{i=1}^{m}\varepsilon_i-\sum_{i=1}^{m}\alpha_i(y_i(x_i^T\omega+b)-1+\varepsilon_i)-\sum_{i=1}^{m}\gamma_i\varepsilon_i$$

$\partial L/\partial\omega=\omega-\Sigma\alpha_iy_ix_i=0\quad\partial L/\partial b=\Sigma\alpha_iy_i=0$

$\partial L/\partial\varepsilon i=C-\alpha_i-\gamma_i=0\ C=\alpha_i+\gamma_i\ge\alpha_i$

代回对偶 s.t. $0\le\alpha_i\le C\ \Sigma\alpha_iy^{(i)}=0$ 多一个≤C

$$\max_{\alpha}\ J(\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)},x^{(j)}\rangle$$

KKT 互补条件:$\alpha_i(y^{(i)}(\omega^Tx^{(i)}+b)-1+\varepsilon i)=0$;

$\gamma_i\varepsilon_i=(C-\alpha_i)\varepsilon_i=0\ \gamma_i(\omega^Tx^{(i)}+b)\ge1-\varepsilon i$

结论①$\alpha_i=0 =>\varepsilon_i=0 =>\gamma_i(\omega^Tx^{(i)}+b)\ge1$

②$\alpha_i=C =>\gamma_i(\omega^Tx^{(i)}+b)\le1$ ③$0<\alpha_i<C$ r=1

【经验风险最小化框架 ERM】min1/N ∑L(yi,f(xi))经验风险是模型关于训练样本集的平均损失。经验风险最小化策略认为，经验风险最小的模型是最优的模型。当样本容量足够大时，经验风险最小化能保证有很好的学习效果，但是，当样本容量很小时会产生过拟合现象。【结构风险最小化】为了防止过拟合而提出的策略。结构风险最小化等价于正则化。结构风险在经验风险的基础上加上表示模型复杂度的正则化项 min1/N∑L(yi,f(xi))+λJ(f);其中 J(f)是模型复杂度的函数，λ系数，用来权衡经验风险和模型复杂度

【核函数】思想：低维线性不可分问题转化为高维线性可分 x->ϕ(x);定义：向量映射到高特征空间后的内积;用核函数代替高维特征空间 **SVM** 的内积运算:很难准确知道低维到高维映射函数,但相对容易定义高位空间内积(核函数);SVM 中计算都内积;只需要核函数即可高位空间 SVM 分类

$$K(x,z)=\phi(x)^T\phi(z)=\langle\phi(x)\quad,\phi(z)\quad\rangle$$

$$\max_{\alpha}\ W(\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}y_iy_j\alpha_i\alpha_j\langle\phi(x_i),\phi(x_j)\rangle$$

【Merce 条件】核函数矩阵:对于任何有限集合点{x1…xn} 核函数矩阵 Kij=K(xi,xj);对称 Kij=Kji;半正定:∀z≠0 $z^TKz\ge0$【常见核函数】线性:$K(x1,x2)=x1^Tx2$

多项式=$(x1^Tx2+c)^d$ 径向基:$\exp(-|x1-x2|^2/2\delta^2)$

【生成式 vs 判别式】对观察和标签的联合概率 p(x, y)建模，并基于贝叶斯法则 p(y|x)=p(x|y)p(y)/p(x)来预测(Naïve Bayes 最大似然估计联合分布);vs 对给定观测值的标签的后验概率$p(y|x)$建模(决策函数直接建模 h=f(x)感知机 交叉熵 最大间隔损失等/对后验概率建模 h=p(y|x) logistic/softmax 最大似然估计条件分布)

假设 pbs①决策函数直接建模 h=f(x) svm 感知机②对后验概率建模 h=p(y|x) Logistic/Softmax;scs:①对联合分布建模 h=p(x,y)=p(y)p(x|y) 朴素贝叶斯 GMM

学习 pbs①决策函数直接建模θ*=argmaxJ(θ) 学习准则:感知机交叉熵最大间隔损失等②对后验概率建模 $\theta^*=\mathrm{argmax}\Sigma\log p(y^{(k)}|x^{(k)})$ 最大似然估计(条件分布);scs:①对联合分布建模$\theta^*=\mathrm{argmax}\Sigma\log(x^{(k)},y^{(k)})$ 最大似然估计(联合分布)

决策 pbs①决策函数 y=h=f(x)②后验概率 $\mathrm{argmax}_yp(y|x)$ scs:①贝叶斯 p(y|x)=p(x,y)/p(x) $\mathrm{argmax}_yp(x|y)p(y)$

【多项式分布假设】平滑 1/V

$p(y=cj)=\pi j\quad p(x|cj)=p(\omega1\dots\omega_{|x|}|cj)=\prod p(\omega_h|cj)\propto\prod_{i=1}^{V}p(t_i|c_j)^{N(t_i,x)}=\prod_{i=1}^{V}\theta_{i|j}^{N(t_i,x)}$ 联合概率:$p(x,y=cj)=p(y=cj)p(x|y=cj)=\pi j\prod\theta_{i|j}^{N(ti,x)}$

似然 $L(\pi,\theta)=\log\prod p(xk,yk)=\log\prod\Sigma I(y_k=c_j)p(y_k=c_j)p(x_k|y_k=c_j)=\Sigma\Sigma I((y_k=c_j)(\log\pi_j+\Sigma N(ti,xk)\log\theta_{i|j})$等式约束:$\max_{\pi,\theta}L(\pi,\theta)$ s.t.$\Sigma\pi j=1$; $\Sigma\theta_{i|j}=1$(C 个) 拉格朗日 $J=L+\alpha(1-\Sigma\pi j)+\Sigma\beta j(1-\Sigma\theta_{i|j})$.$\partial J/\partial\pi j=\Sigma I/\pi j\ -\alpha=0$;$\partial J/\partial\theta_{i|j}=\Sigma IN/\theta_{i|j}-\beta j=0$;解得$\alpha=\Sigma\Sigma I\quad\pi j=\Sigma I/\Sigma\Sigma I=Nj/N$(cj 类文档总数/训练集中所有文档总数) $\beta j=\Sigma\Sigma IN\quad\theta_{i|j}=\Sigma IN/\Sigma\Sigma IN$(cj 类 ti 出现次数/cj 类所有单词次数和)

【伯努利】平滑 1/2 假设 p(y=cj)=πj

p(x|y=cj)=p(t1,t2…,tv|cj)

$$=\prod_{i=1}^{V}[I(t_i\epsilon x)\mu_{i|j}+I(t_i\notin x)(1-\mu_{i|j})]$$

联合概率:p(x,y=cj)=πjp(x|y=cj)

似然 L(π,μ)=同上=$\Sigma\Sigma I(yk=cj)(\log\pi j+\Sigma I(ti\in xk)\log\mu_{i|j}+I(ti\notin xk)\log(1-\mu_{i|j}))$

$\max_{\pi,\mu}L(\pi,\mu)$ s.t. $\Sigma\pi j=1;\Sigma\mu_{i|j}=1$(自然满足)

拉格朗日 $J=L+\alpha(1-\Sigma\pi j)\quad\partial J/\partial\mu_{i|j}=\Sigma I(I\in/\mu-I\notin/1-\mu)=0\ =>\Sigma I(yk=cj)[I(ti\in xk)-\mu_{i|j}]=0$

$\mu_{i|j}=\Sigma I(yk=cj)I(ti\in xk)/\Sigma I(yk=cj)$(cj 类包含词 ti 的文档数/cj 类文档总数) πj 和上面一样

【区别】伯努利模型单词从多维伯努利分布生成，多项式从多项分布中抽取单词;伯努利二元向量表单词存在与否，多项整数向量表单词出现频率；伯努利忽略多次出现单词，适合简短文档；伯努利对 the 单词每个文件均存在 P≈1，多项式基于单词在类中的相对频率 0.05

【K-means】对给定数据集$\{x^{(1)}\dots x^{(N)}\}$目标:N 个样本分 K 个聚类,最小化各聚类内平方距离和 $\mathrm{argmin}_C\Sigma_{k=1\sim K}\Sigma_{x\in Ck}||x-mk||^2$ 迭代优化 初始化:从数据集中选择K个样本作为初始均值;迭代:1.分配:将每个样本分配给最近的聚类，即平方欧氏距离最小 $Ci^{(t)}=\{x:d(x,mi^{(t)}\le d(x,mj^{(t)}),1\le j\le K\}$2.更新每个聚类的均值 $mi^{(t+1)}=1/Ci^{(t)}\Sigma_{x\in Ck}xj$ 3.重复,分配不改变,算法收敛。不保证 WCSS 全局最小值

【梯度反方向】泰勒展开 $f(\theta)\approx f(\theta^{(t)})+\nabla f(\theta^t)^T(\theta-\theta^{(t)})$ 希望 $f(\theta)<f(\theta^{(t)})$ 则$\nabla f(\theta^t)^Tv=||\nabla f(\theta^t)||\,||v||\cos\beta<0$ 则 $v=-\nabla f(\theta^t)$ β=180°

【感知机】以神经网络,感知机一个二层的神经网络.感知机实际是一种线性分类器，用于二分类问题。将每一个实例分类为正类和负类；物理意义:它是将特征空间分为正负两类的分离超平面

【越小越好?】损失函数过小产生过拟合问题，不能只看训练集，可能训练集的损失函数变小，但测试集的准确率变低；解决方案都是一个就是减少特征,另一个是正则化减弱每个特征影响

【softmax VS 贝叶斯】从模型假设来看，softmax 对其建模后验概率 p(y|x)，朴素贝叶斯对联合概率 p(x,y)建模;学习上，softmax 用条件分布极大似然估计,梯度下降进行优化, 朴素贝叶斯使用联合概率分布似然函数后,拉格朗日乘子法,梯度下降进行优化。决策上，softmax 直接用训练好的参数预测 p（y|x）,朴素贝叶斯通过贝叶斯公式来计算 p(y|x)

【交叉熵】H(p,q)=-Σp(i)logq(i) p 为真实 q 非真实;

【非线性分类】神经网络、kernel SVM

【是否唯一】logistic softmax 感知机中用梯度下降方法的都不唯一;闭式解，用拉格朗日求解啥的，SVM,朴素贝叶斯都唯一

【Softmax 求导】Sj=e^aj/Σe^ak

∂Si/∂aj=Si(1-Sj) i=j ; -SjSi i≠j