支持向量机与凸优化求解

AI100学院

2017年8月

大纲

- 背景
- 线性分类
- 非线性分类
- 松弛变量
- 多元分类
- 应用
- 工具包

SVM发展历史

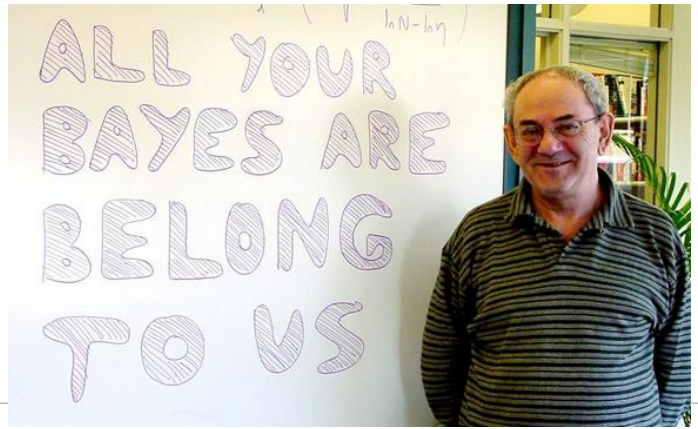
- 重要理论基础1
  - 60年代，Vapnik和Chervonenkis提出**VC维理论**
- 重要理论基础2
  - 1982年，Vapnik提出**结构风险最小化理论**
- Cortes 和Vapnik于1995年首先提出**支持向量机** (Support Vector Machine)
- 在解决**小样本、非线性及高维**模式识别中表现出许多特有的优势，能够推广到**函数拟合**等其他机器学习问题中

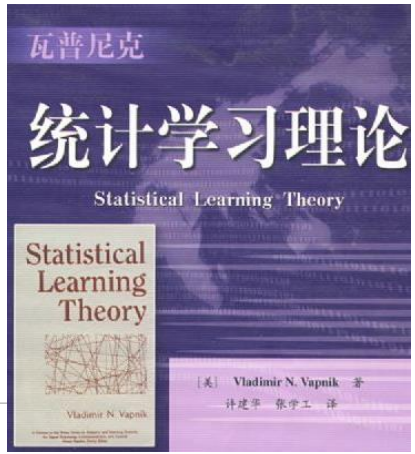
SVM特性

- 小样本学习
  - 与问题的复杂度相比，SVM算法要求的样本数相对较少
- 非线性
  - SVM擅长应对样本数据线性不可分的难题，主要通过松弛变量和核函数技术实现
- 高维数据
  - 例：文本的向量表示，几万维。反例如KNN难以处理高维数据

作者简介

- Vapnik
  - 《Statistical Learning Theory》作者
    - 书中详细论证了**统计机器学习**之所以区别于传统机器学习的本质，在于统计机器学习能够精确地给出学习效果，并解答所需样本数等一系列问题。

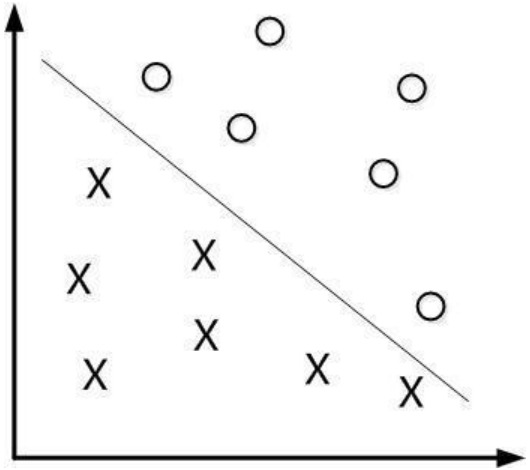




大纲

- 背景
- **线性分类**
- 非线性分类
- 松弛变量
- 多元分类
- 应用
- 工具包

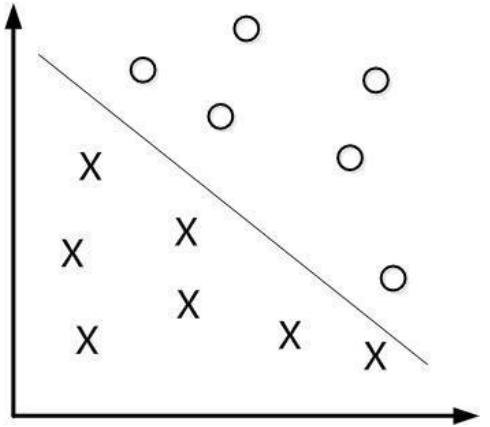
线性分类器



- 问题的引入
  - X和O是两类样本
  - 中间的直线就是一个分类函数，它可以将两类样本完全分开。
- 如果不关注空间的维数，这类线性函数即为上节课介绍的超平面

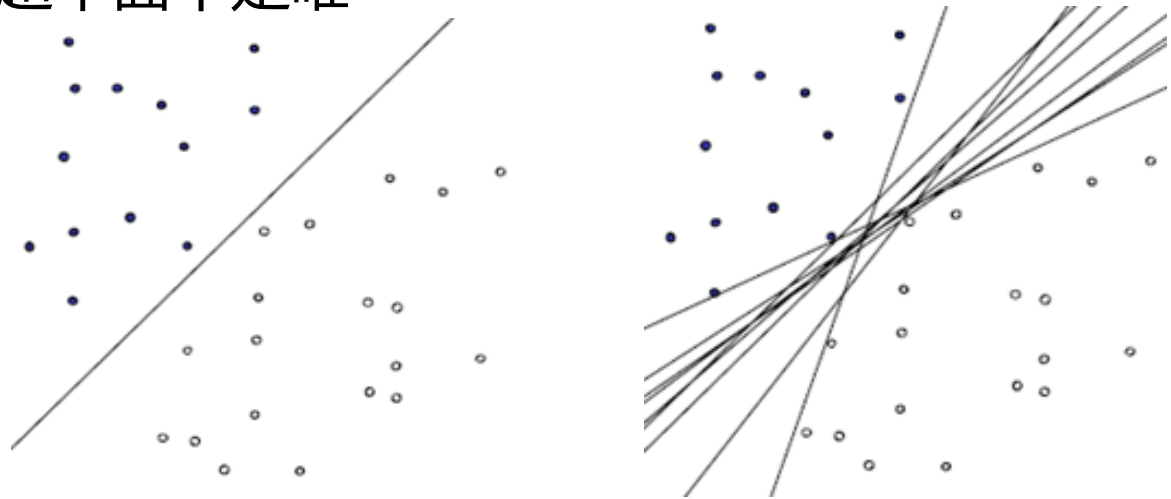
线性函数→分类问题

- 线性函数$g(x)=wx+b$，取阈值为0，有样本x_i需要判别的时候，看$g(x_i)$的值。
  - 若$g(x_i)>0$，就判别为类别O
  - 若$g(x_i)<0$，则判别为类别X
- 注意：
  - w、x、b均可以是向量
  - 中间直线的表达式为$g(x)=0$，即$wx+b=0$，称为分类面



分类面的决定

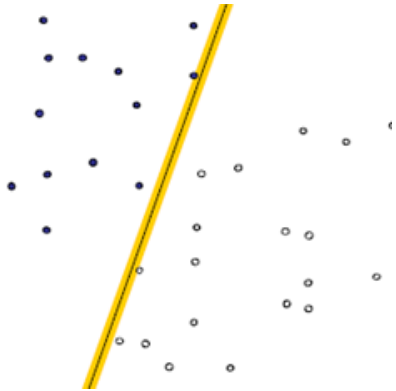
- 分离超平面不是唯一

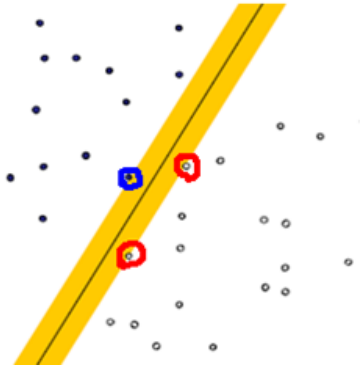


- 图上多条直线都可以完美分类数据点，存在唯一的最优分割超平面

分类面“好坏”的量化

- 一个很直观的判断是，让“离直线最近的点，距离直线尽可能地远”





- 就是分割的间隙越大越好，把两个类别的点分得越开越好

分类间隔

- $\delta_i=y_i(wx_i+b)$
  - $y_i(wx_i+b)$总大于0的，而且值恒等于$|wx_i+b|$
    - 如果某个样本属于该类别的话，$wx_i+b>0$，而y_i也大于0;反之，$wx_i+b<0$，而y_i也小于0
  - 将w和b归一化，即用w/||w||和b/||w||分别代替原来的w和b，分类间隔写成
  - 几何间隔：x_i到超平面g(x)=0的距离

$$\delta_i = \frac{1}{\|w\|}|g(x_i)|$$

几何间隔的现实含义

- H是分类面，而H_1和H_2是平行于H，且过离H最近的两类样本的直线，H_1与H，H_2与H之间的距离就是几何间隔

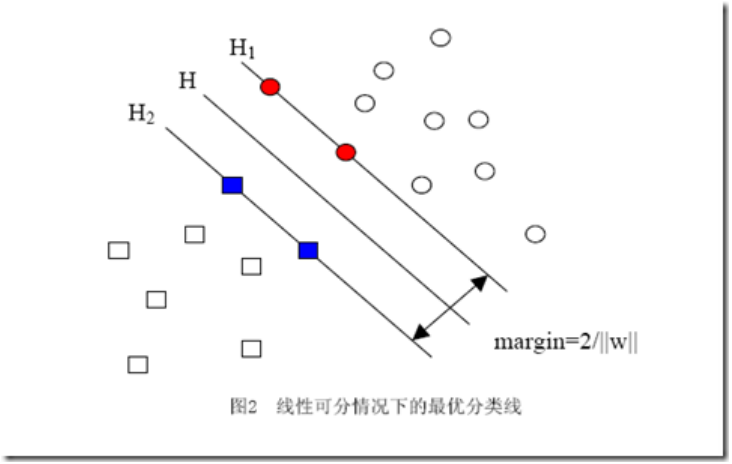
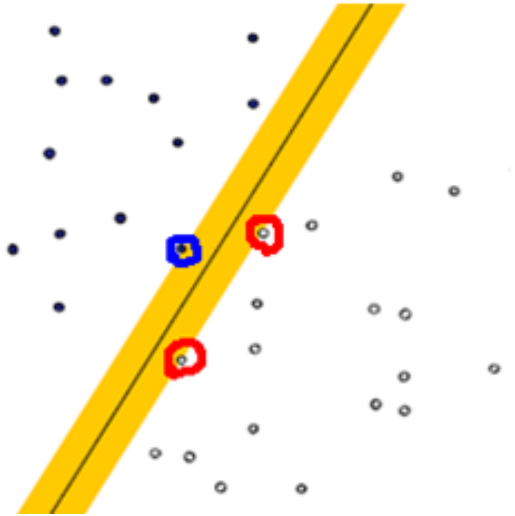
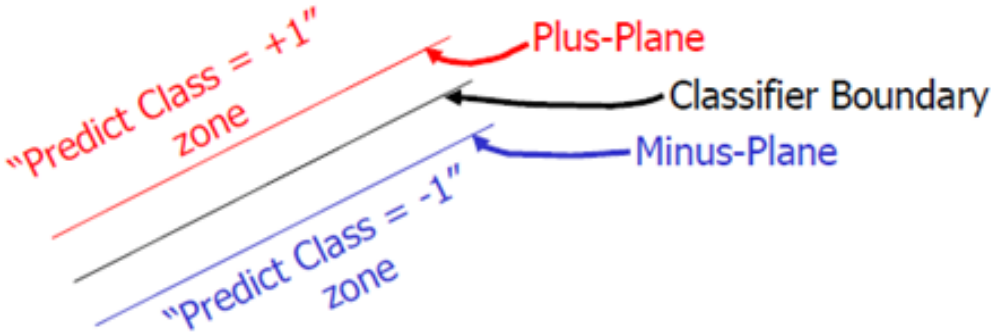


图2　线性可分情况下的最优分类线

支持向量与最大化间隔





- 红色与蓝色点即为支持向量(support vector)，两线间隔为最大化的分类间隔。
- 分类器边界为f(x) ，分类原则为最大化间隔（ Maximum Marginal ）

$$\max \frac{1}{\| w \|} \rightarrow \min \frac{1}{2} \| w \|^2$$

最大化间隔的凸优化形式

- 最大化间隔的完整形式化

$$\min \frac{1}{2} \| w \|^2 \quad s.t., y_i(w^T x_i + b) \geq 1, i = 1, \cdots, n$$

  - 带约束的二次规划(Quadratic Programming, QP)问题，属于凸优化问题
  - 凸二次规划有解，且为全局最优解

凸二次规划问题求解

- 等式约束求极值：通过拉格朗日转化等变为无约束问题
- 不等式约束问题：
  - 方法一：用现成的QP (Quadratic Programming) 优化包进行求解(缺点是效率低)
  - 方法二：求解与原问题等价的对偶问题(dual problem)得到原始问题的最优解(更易求解、可以推广到核函数)
    - 拉格朗日乘子法、拉格朗日对偶性、KKT理论支撑

求解步骤

1. 转化为对偶问题
   – 对偶转化 & KKT条件
2. 求解w、b极小化
   – 拉格朗日乘子极值
3. 求解α极大化
   – 用SMO算法求解α乘子

1、对偶问题的转化

$$min\ \frac{1}{2}\|w\|^2$$
$$subject\ to\ y_i[(wx_i)+b]-1\geq 0(i=1,2,3,\ldots,n)$$

- 给每一个约束条件加上一个拉格朗日乘子（Lagrange multiplier），定义拉格朗日函数

$$\mathcal{L}(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i\left(y_i(w^Tx_i+b)-1\right)$$

- 根据对偶算法与KKT条件约束，这个问题可以从

$$\min_{w,b}\theta(w)=\min_{w,b}\max_{\alpha_i\geq 0}\mathcal{L}(w,b,\alpha)=p^*$$

- 转化为

$$\max_{\alpha_i\geq 0}\min_{w,b}\mathcal{L}(w,b,\alpha)=d^*$$

- 其中 p*和d*等价条件就是KKT条件*

而KKT条件就是指上面最优化数学模型的标准形式中的最小点 x* 必须满足下面的条件：

1. $h_j(\mathbf{x}_*)=0,\ j=1,\ldots,p,\ g_k(\mathbf{x}_*)\leq 0,\ k=1,\ldots,q,$

2. $\nabla f(\mathbf{x}_*)+\sum_{j=1}^{p}\lambda_j\nabla h_j(\mathbf{x}_*)+\sum_{k=1}^{q}\mu_k\nabla g_k(\mathbf{x}_*)=\mathbf{0},$

$\lambda_j\neq 0,\ \mu_k\geq 0,\ \mu_k g_k(\mathbf{x}_*)=0.$

2、w、b的极小化

- 上述问题转化为

$$\max_{\alpha_i \geq 0} \min_{w,b} \mathcal{L}(w, b, \alpha)$$

- 先固定α，求w、b的最小值

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

- 将以上结果代入之前的L，

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i (w^T x_i + b) - 1 \right)$$

- 得到只含α的优化结果

$$\begin{aligned} \mathcal{L}(w, b, \alpha) &= \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - b \sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \end{aligned}$$

3、α的极大化

- 优化问题接上一步处理结果

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$
$$s.t., \alpha_i \geq 0, i = 1, \ldots, n$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

- 如果求出了α*，那么w和b就可以随之求解

$$w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}.$$

$$b^* = -\frac{\max_{i:y^{(i)}=-1} w^{*T} x^{(i)} + \min_{i:y^{(i)}=1} w^{*T} x^{(i)}}{2}.$$

- 最终得出分离超平面和分类决策函数。
- 利用SMO算法求解对偶问题中的拉格朗日乘子α

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$
$$s.t., \alpha_i \geq 0, i = 1, \ldots, n$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

分类函数的预测

- 将w的表达式带入分类函数后

$$f(x) = \left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T x + b$$
$$= \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle + b$$

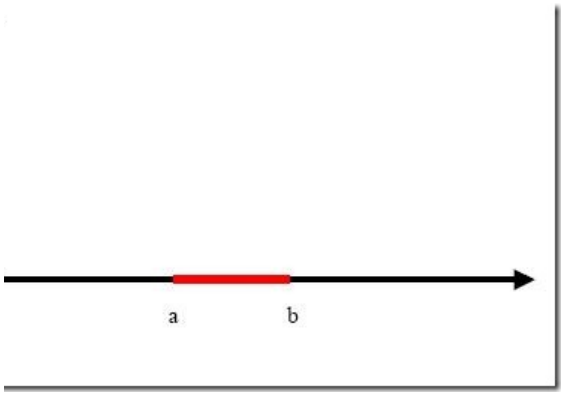
  - 对于新点 x的预测，只需要计算它与训练数据点的内积即可（表示向量内积）
  - 所有非Supporting Vector 所对应的系数α_i均等于零，因此对于新点的内积计算只要针对少量的“支持向量”，无需所有的训练数据。

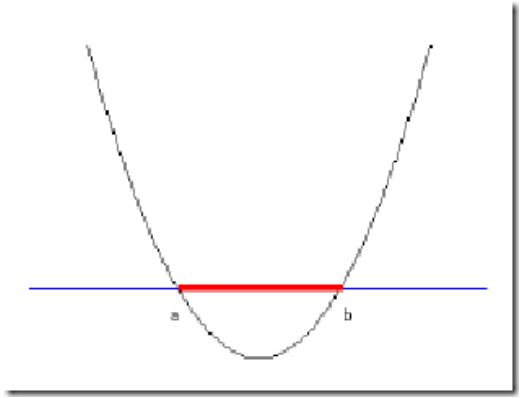
大纲

- 背景
- 线性分类
- **非线性分类**
- 松弛变量
- 多元分类
- 应用
- 工具包

非线性分类——问题的引入

- 指定横轴上端点a和b之间红色部分里的所有点定为正类，指定两边的黑色部分里的点定为负类。
- 能否找到一个线性函数将两类正确分离？答案是不能，因为二维空间里的线性函数就是指直线，显然找不到符合条件的直线。
- 二次曲线 $g(x) = c_0 + c_1 x + c_2 x^2$ 能够将两类正确分离





非线性特征映射——核函数

- 解决线性不可分问题的基本思路：通过核函数实现特征映射(feature mapping)向高维空间转化，使其变得线性可分。

- 核函数的形式化定义： $K(x,z) = \phi(x)^T\phi(z)$
  - 满足Mercer条件*的函数，都可以作为核函数。
  - 核函数的基本作用就是接受两个低维空间里的向量，能够计算出经过某个变换后在高维空间里的向量内积值。

Mercer定理：如果函数K是 $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ 上的映射（也就是从两个n维向量映射到实数域）。那么如果K是一个有效核函数（也称为Mercer核函数），那么当且仅当对于训练样例 $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$，其相应的核函数矩阵是对称半正定的。

核函数——例子1

- 假设x和z都是n维，核函数展开为：

$$K(x,z) = (x^T z)^2 = \left(\sum_{i=1}^{n} x_i z_i\right)\left(\sum_{j=1}^{n} x_j z_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j z_i z_j$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} (x_i x_j)(z_i z_j) = \phi(x)^T \phi(z)$$

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}.$$

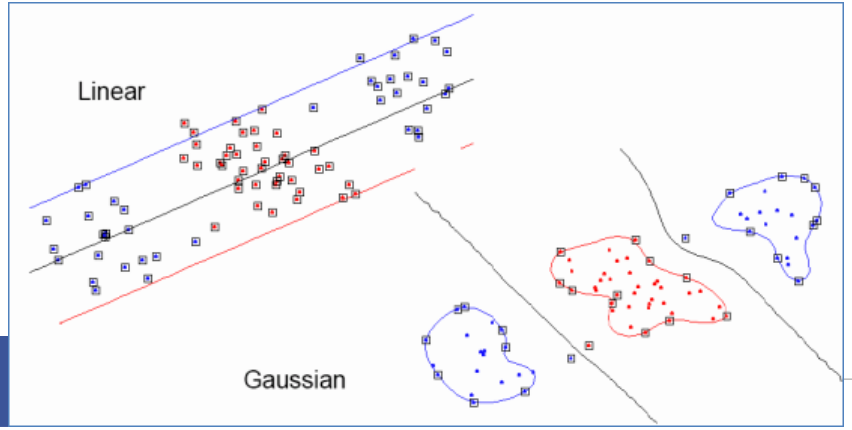
- 计算原始特征x和z内积的平方，时间复杂度是O(n) ，与计算映射后特征的内积等价。

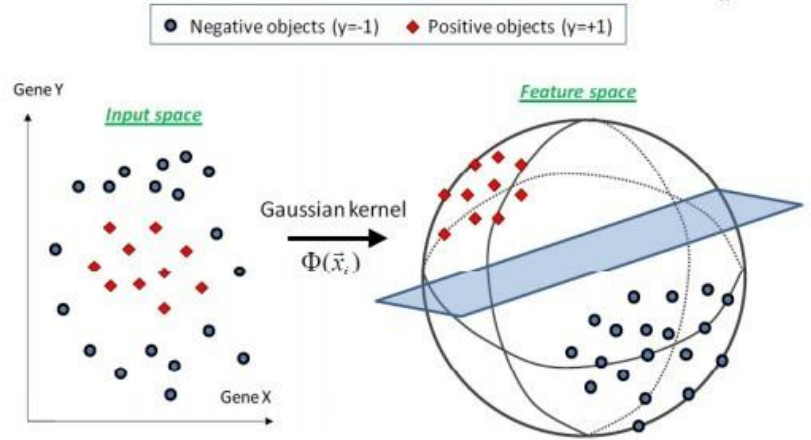
核函数——高斯核

$$K(x,z)=\exp\left(-\frac{||x-z||^2}{2\sigma^2}\right).$$

如果x和z很相近核函数值为1，如果x和z相差很大核函数值约等于0。

类似于高斯分布，称为高斯核函数，也叫做径向基函数，能够把原始特征映射到无穷维。

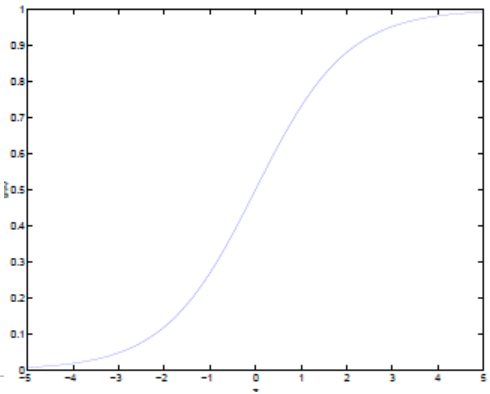




核函数——sigmoid核

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i{}^T \mathbf{x}_j + r).$$

高斯核函数比较x和z的相似度，并映射到0到1，sigmoid函数具有类似功能，因此可以使用sigmoid核函数。

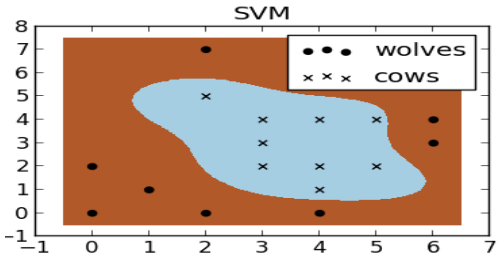


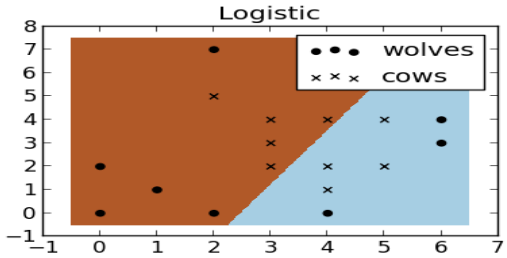
$$g(z) = \frac{1}{1 + e^{-z}}$$

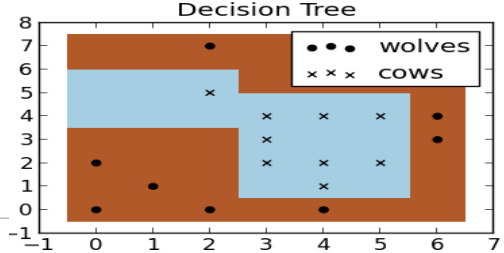
核函数小结与分类效果

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle + b \qquad \sum_{i=1}^{n} \alpha_i y_i \kappa(x_i, x) + b$$

- 经常遇到线性不可分的样例，常用做法是把样例特征映射到高维空间
- 核函数将特征从低维到高维进行转换，但核函数可以事先在低维上进行计算，将实质上的分类效果表现在了高维上，避免高维空间中的直接计算的时间复杂度





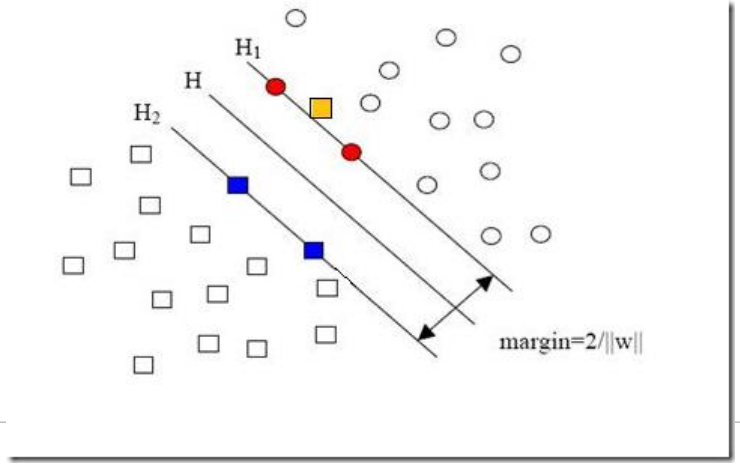


大纲

- 背景
- 线性分类
- 非线性分类
- **松弛变量**
- 多元分类
- 应用
- 工具包

异常点与近似线性可分

- 图中黄色方形点为负类样本，单独一个样本使得原本线性可分的问题变成了线性不可分的。仅有少数点线性不可分的类似的问题称为“近似线性可分”
- 原因：“硬间隔”分类法硬性要求所有样本点满足和分类平面间的距离必须大于某个值。
- 解决方案：允许一些点到分类平面的距离不满足硬性要求



松弛变量的引入

$$y_i[(wx_i + b)] \geq 1(i = 1, 2, 3, \ldots, n)$$

- 约束离分类面最近的样本点函数间隔大于1，如果引入容错性，给硬性阈值加一个松弛变量，即允许

$$y_i[(wx_i + b)] \geq 1 - \zeta_i(i = 1, 2, 3, \ldots, n)$$
$$\zeta_i \geq 0$$

- 因为松弛变量是非负的，因此最终结果是要求间隔可以小于1
- 放弃对离群点的正确分类会带来精度损失，好处是不必强制分类超平面向离群点方向移动，从而得到更大的几何间隔，使得低维分类边界更平滑

松弛变量 vs 优化问题

- 原始的硬间隔分类对应的优化问题为：

$$min\ \frac{1}{2}\|w\|^2$$
$$subject\ to\ y_i[(wx_i)+b]-1\geq 0(i=1,2,3,\ldots,n)$$

- 将松弛变量加入到优化问题：

$$min\ \frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\zeta_i$$
$$subject\ to\ y_i[(wx_i)+b]\geq 1-\zeta_i(i=1,2,3,\ldots,n)$$
$$\zeta_i\geq 0$$

$\sum_{i=1}^{n}\zeta_i^2$：二阶软间隔分类器

$\sum_{i=1}^{n}\zeta_i$：一阶软间隔分类器

  - 惩罚因子 C：损失函数在目标函数中的权重，需要事先指定
  - 松弛变量只在离群点处非零，标示离群多远
  - 核函数与松弛变量的异同：
    - 相同点：解决线性不可分问题
    - 不同点：原始低维样本通过核函数映射，接近线性可分；采用松弛变量处理少数离群点

大纲

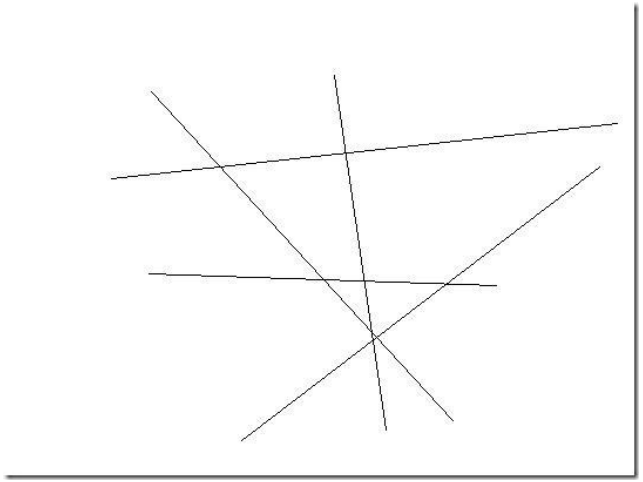
- 背景
- 线性分类
- 非线性分类
- 松弛变量
- **多元分类**
- 应用
- 工具包

多元分类

- SVM是一种典型的两类分类器，即它只回答属于正类还是负类的问题
- 而现实中要解决的问题，往往是多类的问题
- 如何由两类分类器得到多类分类器，是一个值得研究的问题

方案一：一次求解N个分类面

- 一次性考虑所有样本，并求解一个多目标函数的优化问题，一次性得到多个分类面
- 可惜这种算法还基本停留在纸面上，因为一次性求解的方法计算量实在太大，大到无法实用的地步



方案二：一类对其余

- 一类对余类法(One versus rest，OVR)
  - 构造类别数k个的二元分类器
  - 训练时第i个分类器取训练集中第i类为正类，其余类别点为负类
  - 判别时，输入信号分别经过k个分类器输出
- 优点
  - 每个优化问题的规模比较小，而且分类的时候速度很快
- 缺点
  - 分类重叠 & 不可分类 & 人为的数据偏斜

方案三：一对一

- 该方法在每两类问训练一个分类器，因此对于一个k类问题，将有k(k-1)/2个分类器
- 优点
  - 避免了数据偏斜
  - 训练阶段（也就是算出这些分类器的分类平面时）所用的总时间却比“OVR”方法少很多
  - 投票时也会有分类重叠的现象，但不会有不可分类现象
- 缺点
  - 类别数为5的时候，调用了10个分类器，类别数如果是1000，要调用的分类器数目会上升至约500,000个（但是时间上可能OVO还是比OVR少，因为考虑的样本数少）

方案四：DAG方法(有向无环图)

- DAG-SVMs是针对OVO存在误分现象提出的
- 这种方法的k(k-1)/2个分类器，构成一个有向无环图。该有向无环图中含有k(k-1)/2个内部节点和k个叶结点，每个节点对应一个二类分类器

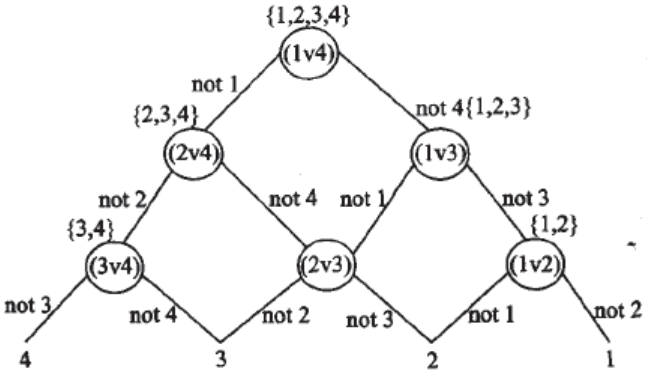


图 2　四类问题 DAGSVM 结构图

方案四：DAG方法(有向无环图)

- 优点
  - 简单易行，只需要使用k-1个决策函数即可得出结果，较“一对一"方法提高了测试速度，而且不存在误分、拒分区域
  - 由于其特殊的结构，故有一定的容错性，分类精度较一般的二叉树高
- 缺点
  - 误差积累

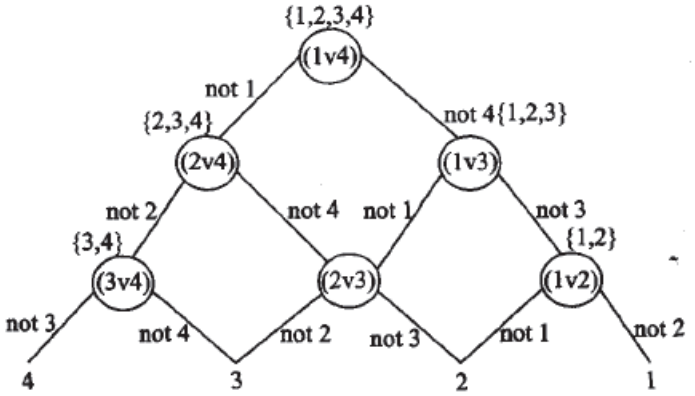


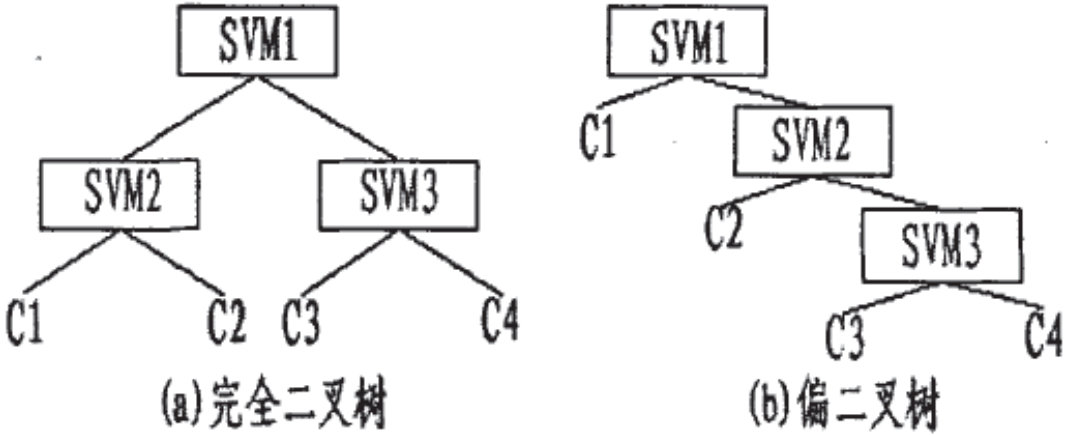
图 2　四类问题 DAGSVM 结构图

方案四：DAG方法(有向无环图)

- DAG的错误累积
  - 错误累积在一对其余和一对一方法中也都存在，DAG方法好于它们的地方就在于，累积的上限，不管是大是小，总是有定论的，有理论证明
  - 而一对其余和一对一方法中，尽管每一个两类分类器的泛化误差限是知道的，但是合起来做多类分类的时候，误差上界难以推导
- DAG方法根节点的选取
  - 取两类分类中正确率最高的那个分类器作根节点
  - 置信度最大的路径

其他方案：决策树、ECOC

- 决策树方法



- 纠错输出编码法(ECOC)
  - K*L维编码矩阵
  - 类别判定用汉明距离

大纲

- 背景
- 线性分类
- 非线性分类
- 松弛变量
- 多元分类
- **应用**
- 工具包

SVM的应用

- 文本分类
- 图像处理
  - 图像过滤、图片分类与检索
- 生物信息技术
  - 蛋白质分类
  - 语音识别
  - 人脸检测、指纹识别
- 手写字体识别
- 网络入侵检测、口令认证、网页分类
- ……

SVM文本分类应用

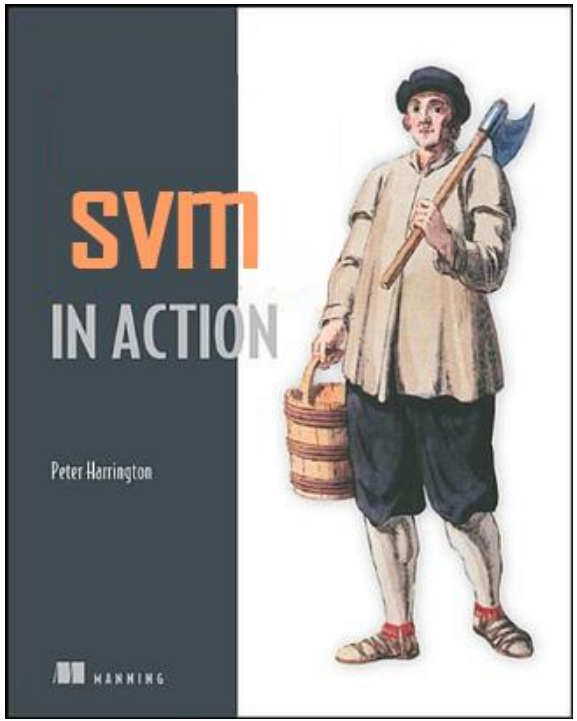
- Topic 分类
  - 14万条微信数据，33个类别。3000条测试数据，其余数据为训练数据。
- Emotion 分类
  - 8000句微博，3个类别。2000句测试数据，其余数据训练。
- 省略恢复
  - “小明买了苹果，很甜。”

大纲

- 背景
- 线性分类
- 非线性分类
- 松弛变量
- 多元分类
- 应用
- **工具包**

SVM工具包

- Libsvm
- Liblinear
- Svm_perf
- LibShortText
- ……



Libsvm简介

- LibSVM是林智仁(Chih-Jen Lin) 教授开发
- 可以很方便的对数据做分类或回归
- 程序小，运用灵活，输入参数少，并且是开源的，易于扩展，因此成为目前国内应用最多的SVM的库

Libsvm工具包

- 工具包组成
  - Java
  - Matlab
  - Python
  - svm-toy(一个可视化的工具, 用来展示训练数据和分类界面, 里面是源码, 其编译后的程序在windows文件夹下)
  - Tools(四个python文件, 用来数据集抽样(subset), 参数优选(grid), 集成测试(easy), 数据检查(checkdata))
  - Windows(包含libSVM四个exe程序包)
  - 其他.c .h源码

Libsvm工具包常用命令

- Svmtrain
  - svmtrain [options] training_set_file [model_file]
- Svmpredict
  - svmpredict [options] test_file model_file output_file
- Svmscale
  - svmscale [options] filename

Liblinear

- Liblinear
  - 线性分类器
  - 主要为大规模数据的线性模型设计
  - 由于采用线性核,所以不需要计算kernel value,速度更快
  - 缺点是内存占用较高：10G的数据需要接近50G内存

什么时候用Liblinear

- 当你面对海量的数据时，这里的海量通常是百万级别以上
  - 海量数据分为两个层次：样本数量和特征的数量。
- 使用线性和非线性映射训练模型得到相近的效果
- 对模型训练的时间效率要求较高

Liblinear高效分类的理论基础

- 信任区域方法
  - 优化的框架，与常用的线搜索互为对偶
  - x_k+1 = x_k + alpha * p_k
  - 它是先确定一个region(hyperball)，或者说先确定它的半径delta(因为球心就是x_k)，然后在此球内优化泰勒展式的局部模型(一般都是二阶)寻找方向p_k，如果优化成功则球心转移，并扩大半径；如果不成功则球心不变，缩小半径。并如此反复。（区别于line search 先确定p_k后优化alpha）

Liblinear高效分类的理论基础

- 截断牛顿方法
  - 指牛顿法中计算H * p_k + g = 0时采用数值迭代解决这个线性系统问题而不是直接高斯消元，其中g和H分别是目标函数的一阶导和二阶导。
  - 通常情况下，可以用共轭梯度的近似解来逼近。

Svm_perf

- Made by 康奈尔大学
- 对计算机硬件的性能要求比liblinear要低
- 相关文献：

NEW SVM^struct: SVM learning for multivariate and structured outputs like trees, sequences, and sets (available here).

NEW SVM^perf: New training algorithm for linear classification SVMs that can be much faster than SVM^light for large datas like F1-Score, ROC-Area, and the Precision/Recall Break-Even Point. (available here).

NEW SVM^rank: New algorithm for training Ranking SVMs that is much faster than SVM^light in '-z p' mode. (available here).

LibShortText

- LibShortText是一个开源的Python**短文本**(包括标题、短信、问题、句子等)分类工具包
- 在LibLinear的基础上针对短文本进一步优化，主要特性有：
  - 直接输入文本，无需做特征向量化的预处理
  - 二元分词（Bigram），去停顿词，做词性过滤
  - 基于线性核SVM分类器
  - 提供了完整的API，用于特征分析和Bad Case检验

总结

- 背景
  - SVM历史、宏观的理论基础与优点
- 线性分类
  - 分类面、间隔最大化、对偶问题求解参数
- 非线性分类
  - 问题的提出、核函数定义、举例
- 松弛变量
  - 离群点、软间隔、松弛变量、惩罚因子
- 多元分类
  - 多元分类方法
- 应用
  - 在NLP、图像、网络等方面的分类应用
- 工具包
  - libsvm、liblinear等

CSDN
不 止 于 代 码

THANK YOU

AI100