



中国科学技术大学
University of Science and Technology of China

数据挖掘与数据仓库

线性分类

November 15, 2017



- ① 从例子开始
- ② 分类的几何意义
- ③ 线性 SVM



电子邮件地址识别：问题描述

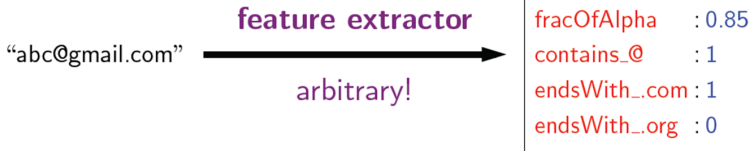
- 输入 x : 表示给定的一个字符串,
- 输出: 判断 x 是不是电子邮件地址。
 - 输入是字符串 x
 - 输出是 -1 或 +1, +1 表示是电子邮件地址, -1 表示否

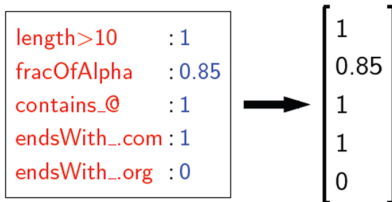
例子

- $x = \text{"zhangsan@gmail.com"} \rightarrow y = +1$ 是电子邮件地址
- $x = \text{"abcd.efgh.ijk"} \rightarrow y = -1$ 不是电子邮件地址

电子邮件地址识别：问题分析

- 输入字符串 x 的“预处理”：特征提取函数
- x 变化范围太大，非结构化的，处理不方便，先预处理，提取描述 x 的特征向量，即 $x \implies \phi(x) = (\phi_1(x), \phi_2(x), \dots, \phi_d(x))$
- 特征提取函数输出一个长度为 d 的向量，描述输入 x 的特征
- 在图像和语音识别研究中常用，且是其基本研究内容之一
- 例子如图所示， $x \implies \phi(x) = (\text{length} > 10, \text{fracOfAlpha}, \text{contains_@}, \text{endsWith_}.com, \text{endsWith_}.org)$





电子邮件地址识别：特征提取函数

- 如上图所示，任意一个字符串 x ，经过特征提取函数的处理获得一个长度为 5 的实数向量
- 特征向量可以视为 5-维空间中的一个点

问题：存在很多特征提取函数，特征提取函数有优劣之分

- 评价标准？
- 评价方法？



手写体数字识别问题分析 1: f 是什么?

- 人眼看到输入, 人脑对输出做出判决;
- 函数 f 涉及复杂的知识背景, 心理活动等; 难以用数学函数完美定义。

手写体数字识别问题分析 2: h 是什么?

- 用最简的线性函数 $h(x) = \mathbf{W} \cdot \phi(x)$ 来近似 f
- 注意到 \mathbf{W} 是一个一维向量, x 变成了特征提取函数 $\phi(x)$, 同时线性表达式的常数项 b 丢掉了 (为什么可以丢掉常数项?)



```
length>10      :-1.2  
fracOfAlpha    :0.6  
contains_@     :3  
endsWith_.com :2.2  
endsWith_.org :1.4  
...
```

权向量 W 的解释

- 权向量描述每一个特征的在判决是否是电子邮件地址时的贡献
- 如图所示例子，字符串长度大于 10 时，对是电子邮件地址的判决作用是负面的，权值为-1.2
- 字符串中包含 @ 字符，对判决是电子邮件的作用是正面的，权值为 3
- 权值绝对值越大，表明对应特征对最终判决的影响越大

Weight vector $\mathbf{w} \in \mathbb{R}^d$

length>10	:-1.2
fracOfAlpha	:0.6
contains_@	:3
endsWith_.com	:2.2
endsWith_.org	:1.4

Feature vector $\phi(x) \in \mathbb{R}^d$

length>10	:1
fracOfAlpha	:0.85
contains_@	:1
endsWith_.com	:1
endsWith_.org	:0

分数: *score*

- $h(x) = \mathbf{W} \cdot \phi(x)$
- 图中两个向量 $\mathbf{W}, \phi(x)$ 的点积/内积, 是核心计算过程:
 - 两个实数向量的内积, 输出是实数, 并不能作为真实 f 的输出 $y \in \{0, 1\}$ 的近似, 因为 y 是离散的 -1 和 +1
 - 重新定义两个向量的内积为 *score*/分数, 不是真实 f 的输出 y , 而是给 y 打出的一个“分数”, 即: $score \triangleq \mathbf{W} \cdot \phi(x)$



从分数 $score$ 到输出 y

- 将“分数”离散化为 $\{-1, +1\}$
- 如下给出一种可选方法：

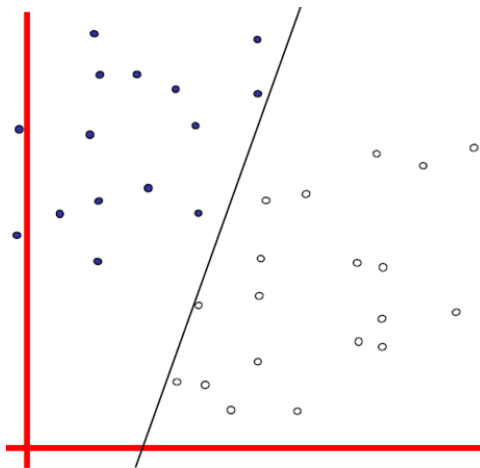
$$h(x) = \text{sign}(W \cdot \phi(x)) = \begin{cases} +1 & , \text{ if } W \cdot \phi(x) > 0 \\ -1 & , \text{ if } W \cdot \phi(x) < 0 \\ * & , \text{ if } W \cdot \phi(x) = 0 \end{cases}$$



电子邮件地址识别：总结

- 问题描述
- 问题分析：特征提取， h 选择线性函数
- 问题求解：最优 W 的确定、及其意义的说明、分数定义及其到输出响应 y 的变换等。

与回归问题的区别？



解释说明

- 考虑图示二维的情形: $\mathbf{x} = (x_1, x_2)$
 $y \in \{\text{实心点}, \text{空心点}\}$
- 直线方程 $ax_1 + bx_2 + c = 0$, 可扩展到 d 维空间 $\mathbf{W} \cdot \mathbf{x} + b = w_1x_1 + w_2x_2 + \dots + w_dx_d + b$
- 三维时, 是平面, $d(>0)$ 维时称为“超平面”
- 直线上的点满足 $ax_1 + bx_2 + c = 0$, 实心点满足 $ax_1 + bx_2 + c > 0$, 空心点满足 $ax_1 + bx_2 + c < 0$
- h 的线性表达式为 0 时, 就是该超平面, 也称“决策边界/分类器”:
$$h(x) = \text{sign}(W \cdot \phi(x)) = \begin{cases} +1 & , \text{ if } W \cdot \phi(x) > 0 \\ -1 & , \text{ if } W \cdot \phi(x) < 0 \\ * & , \text{ if } W \cdot \phi(x) = 0 \end{cases}$$

以二分类问题为例: $y = \{+1, -1\}$

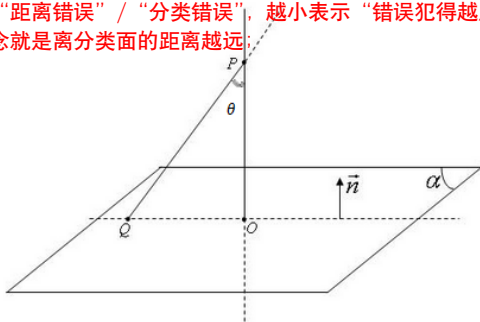
- 分数 $score \triangleq W \cdot \phi(x)$, 描述了对 f 的输出 y 进行预测时, 对预测结果有多大程度的“信任”, 是个打分, 正得越大表示越相信 y 是 $+1$, 负得越多表示越相信 y 是 -1
- 间隔 $margin \triangleq W \cdot \phi(x)y$, 表示对 f 的输出 y 进行预测时, 对预测结果的正确性有多大把握, 数值越大, 把握越大;
 - 间隔为负, 表示预测错误
 - 间隔的定义和符号函数 $sign(\cdot)$ 定义的离散化方法相比, 保留了更多的信息。
 - “将简单粗暴的 $\{+1, -1\}$ 的硬判决搞得温柔一些”, 用间隔来实现, 间隔的大小具有一定的“物理意义/几何意义”

符号函数:

$$h(x) = sign(W \cdot \phi(x)) = \begin{cases} +1 & , \text{ if } W \cdot \phi(x) > 0 \\ -1 & , \text{ if } W \cdot \phi(x) < 0 \\ * & , \text{ if } W \cdot \phi(x) = 0 \end{cases}$$

间隔若为负，表示“距离错误” / “分类错误”，越小表示“错误犯得越厉害”

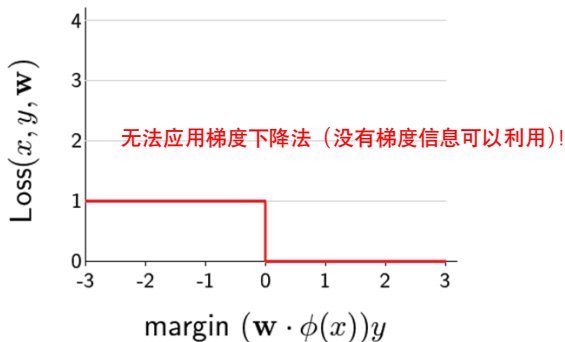
间隔越大，直观概念就是离分类面的距离越远；



间隔：平面外点 P 到平面的距离的 $\|\vec{W}\|$ 倍

- 平面 α 方程： $\vec{W} \cdot \vec{X} + b = 0$
- 平面外一点 P 到 α 的垂足为点 Q ，平面的单位法向量为 $\vec{n} = \vec{W} / \|\vec{W}\|$
- Q 为平面内任意一点；辅助计算 $dist(P, \alpha)$

$$\begin{aligned} dist(P, \alpha) &= \|\vec{PQ}\| = \|\vec{PQ}\| \cos \theta = \\ &= \vec{PQ} \cdot \vec{n} = \vec{PQ} \cdot \vec{W} / \|\vec{W}\| = \\ &= (\vec{P} - \vec{Q}) \cdot \vec{W} / \|\vec{W}\| = \vec{P} \cdot \vec{W} / \|\vec{W}\| - \\ &= \vec{Q} \cdot \vec{W} / \|\vec{W}\| = \vec{P} \cdot \vec{W} / \|\vec{W}\| + b / \|\vec{W}\| = \\ &= (\vec{P} \cdot \vec{W} + b) / \|\vec{W}\| \end{aligned}$$

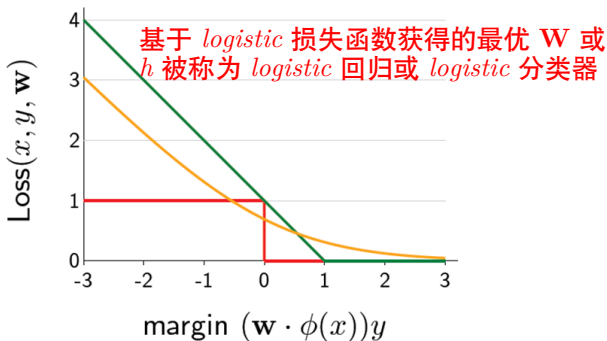


简单粗暴的方法：符号函数 $\text{sign}(\cdot)$

- $\mathbf{W} \cdot \phi(x)y < 0, \text{loss}(x, y, \mathbf{W}) = 1$
- $\mathbf{W} \cdot \phi(x)y = 0, \text{loss}(x, y, \mathbf{W}) = 1$
- $\mathbf{W} \cdot \phi(x)y > 0, \text{loss}(x, y, \mathbf{W}) = 0$

硬判决!

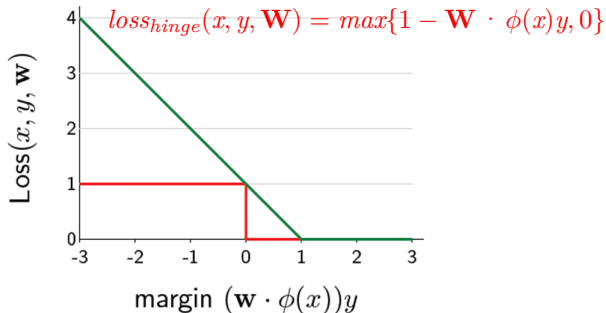
可用梯度下降
法求最优 \mathbf{W}



改进方法：软判决，使得梯度信息存在且有意义

- 如图中的黄色曲线，连续的，大于 0 的，光滑曲线。
- 基本趋势：间隔越大，损失越小。
- *logistic* 回归：用该损失函数作为计算训练损失的核心。
- 虽然名为“回归”实际干的活却是“分类”。
- 黄色曲线的方程： $loss_{logistic}(x, y, \mathbf{W}) = \log(1 + e^{-\mathbf{W} \cdot \phi(x)y})$ 。

可用梯度下降
法求最优 \mathbf{W}



改进方法：把关键部分的梯度信息变成非 0

- 如上图所示绿色曲线，称之为 *hinge* 损失函数，如同门的“转轴/*hinge*”
- 梯度信息不再是处处为 0
- 任意一个绿点定义的损失值都大于相同间隔的红点定义的损失值，也就是说相对于“简单粗暴法”的损失函数定义，我们把损失“夸大”了（上界），我们更激烈地反对“严重错误”
- 注意到间隔在 $(0,1)$ 之间的时候，我们也定义了非 0 的损失，这迫使我们优化的时候要考虑间隔要大于等于 1，即正确的分类判断要足够有说服力（间隔大于 1）
- *hinge* 损失函数，可以让我们用梯度下降法来求线性表达式的参数 \mathbf{W}



hinge 损失函数不仅仅是为了可以使用梯度下降法

- 推导出另一种求解最优 W 的方法
- 从几何结构入手，重新定义最优化问题，然后求解最优 W



代数间隔

- $a - \text{margin} \triangleq \mathbf{W} \cdot \phi(x)y$, 或
- $a - \text{margin} \triangleq (\mathbf{W} \cdot x + b)y$

几何间隔：数据点到分类平面的距离

- $g - \text{margin} \triangleq \frac{\mathbf{W} \cdot \phi(x)y}{\|\mathbf{W}\|}$
- $g - \text{margin} \triangleq \frac{(\mathbf{W} \cdot x + b)y}{\|\mathbf{W}\|}$

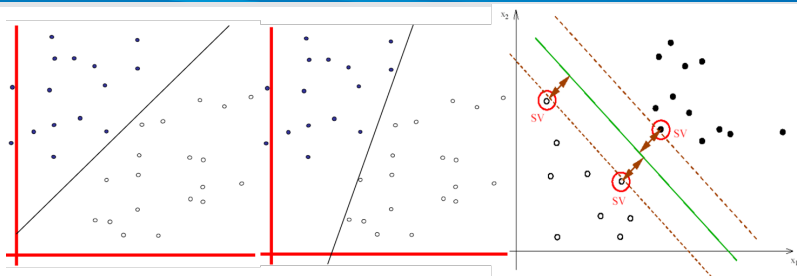
从几何结构上来看，追求代数间隔大于等于 1

- 从 *hinge* 损失函数的定义看，我们需要让代数间隔大于等于 1，使得 *hinge* 损失为最小值 0；
- 即 $loss_{hinge}(x, y, \mathbf{W}) = \max\{1 - \mathbf{W} \cdot \phi(x)y, 0\} = 0, \Rightarrow \mathbf{W} \cdot \phi(x)y \geq 1$
- 用几何间隔来描述，即 $\frac{\mathbf{W} \cdot \phi(x)y}{\|\mathbf{W}\|} \geq \frac{1}{\|\mathbf{W}\|}$ ，即任何一个数据点，它到分类面的距离都大于等 $\frac{1}{\|\mathbf{W}\|}$

新优化问题描述

- 约束条件：找到一个分类面 $h(x) = \mathbf{W} \cdot \phi(x)$ ，使得它能分开所有的训练数据，且使得任意一个数据点到分类界面的距离（几何间隔）大于等于 $\frac{1}{\|\mathbf{W}\|}$
- 优化目标： $\frac{1}{\|\mathbf{W}\|}$ 越大越好，因为 $\frac{1}{\|\mathbf{W}\|}$ 是训练数据到分类面的距离的下界，下界越大，说明分类器分类带来的置信程度越高。最大化 $\frac{1}{\|\mathbf{W}\|}$ 可等价于最小化 $\|\mathbf{W}\|$
- 该优化问题来自对分类问题几何意义的理解和形式化。我们称该优化问题是追求“结构风险最小化”，不同于定义在训练集上的损失最小的“经验风险最小化”问题。

如何求得最优 \mathbf{W} ?



如图所示，不同的分类面

- W 定义的超平面很多，每个数据点离不同超平面的距离会有所不同
- 这些点中有些点离决策边界/超平面很关键，即距离最短，是离超平面最近的点。超平面可以由这些点来完全确定（计算出来），这些点称为“支持向量” / sv
- 非支持向量有什么用？当你选择的 W 发生改变的时候，有些非支持向量就会变成支持向量，原来的支持向量可能就不是支持向量了
- 因此，这些非支持向量就是一种“约束”，它约束你选择的 W 确定的决策面及对应的支持向量间的距离达到“最大”

线性 SVM 优化问题：适用于线性可分的数据集/问题

$$\min \|\mathbf{W}\| \iff \min \frac{1}{2} \mathbf{W} \cdot \mathbf{W}$$

s.t.

$$1 - (\mathbf{W} \cdot \mathbf{x} + b)y \leq 0, \forall (\mathbf{x}, y) \in D_{train}$$

解释说明

- 每个样本/训练数据带来一个约束条件
- 约束条件表明所有的训练数据都被正确分开 $(\mathbf{W} \cdot \mathbf{x} + b)y \geq 1 > 0$
- 且分开得足够远，即代数间隔 > 1

如何求解？带有约束条件，梯度下降？

对约束条件的处理

- 每个样本/训练数据，对应一个非负乘法因子 $\alpha_i \geq 0$ ，并带入对应的约束条件中，使得

$$[1 - (\mathbf{W} \cdot \mathbf{x}_i + b)y_i]\alpha_i = 0, \forall (\mathbf{x}_i, y_i) \in D_{train}$$

- 意义：离分类界面最近的训练数据，因为 $1 - (\mathbf{W} \cdot \mathbf{x}_i + b)y_i = 0$ ，所以 α_i 取值可以随意变化；而非最近的数据，必须要让 $\alpha_i = 0$ ，来确保 $[1 - (\mathbf{W} \cdot \mathbf{x}_i + b)y_i]\alpha_i = 0$ 支持向量的 不为0，非支持向量的 为0

- 然后，把所有的约束条件加起来：

$$\sum_{(\mathbf{x}_i, y_i) \in D_{train}} [1 - (\mathbf{W} \cdot \mathbf{x}_i + b)y_i] \leq \sum_{(\mathbf{x}_i, y_i) \in D_{train}} [1 - (\mathbf{W} \cdot \mathbf{x}_i + b)y_i]\alpha_i = 0$$

- 其实就是把 SVM 优化问题的原始约束条件右端的 0，设计了一个等于 0 的计算公式（利用引入的参数 α_i ）



对目标函数的处理

- $\mathcal{L}(\mathbf{W}, b, \alpha) \triangleq \frac{1}{2} \mathbf{W} \cdot \mathbf{W} + \sum_{(\mathbf{x}_i, y_i) \in D_{train}} [1 - (\mathbf{W} \cdot \mathbf{x}_i + b)y_i] \alpha_i$
- 把原问题的目标函数加上等于 0 的约束条件，并写成自由参数 \mathbf{W}, b, α 的函数式 $\mathcal{L}(\mathbf{W}, b, \alpha)$
- 原问题（带约束）的最小值，就等于 $\mathcal{L}(\mathbf{W}, b, \alpha)$ （无约束）的最小值。

拉格朗日乘子法的核心在于把约束条件去掉（加到目标函数中去）

有没有更好一点的方法来求 $\mathcal{L}(\mathbf{W}, b, \alpha)$ 的最小值？

- 由问题满足 KKT 条件，以及对偶性等，可以将对 $\mathcal{L}(\mathbf{W}, b, \alpha)$ 最小化问题转化为：

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (1)$$

s.t.

$$\sum_i \alpha_i y_i = 0, \forall \alpha_i \geq 0 \quad (2)$$

以上的 只涉及支持向量

解释说明

- 因为非支持向量的 α_i 为 0，所以目标函数公式 (1) 中仅仅包含非零 α_i 的支持向量间的内积运算，并求最大值；
- 也就是说，若知道支持向量是哪些训练样本，那么直接带入公式 (1)，然后求最大值对应的 α_i 值即可。然而我们并不知道谁是“支持向量”！
- 另有两组隐含的等式： $\mathbf{W} = \sum_i \alpha_i y_i \mathbf{x}_i$, $b = y_i - \mathbf{W} \cdot \mathbf{x}_i, \forall (\mathbf{x}_i, y_i)$ is a SV

根据 的定义，W 由支持向量得到的

有兴趣看推导过程的请去查阅相关材料。(极值点导数为 0)

SMO: 序贯最小化算法/Sequential Minimal Optimization 算法思想: (假设数据集有 m 个样本/训练数据) m 个样本, 每个样本有一个

- 每次固定 $m - 2$ 个乘法因子 α_i , 让其余两个任意变化, 寻找这两个任意变化乘法因子的最佳值; 循环迭代调整固定不同的 $m - 2$ 个乘法因子 α , 直到收敛。
- 局部搜索的思想, 寻找最佳的自由变量 α_i , 也就是对应的支持向量及其非零的 α_i
- 训练过程中 (解优化问题时), 尽量避免每次循环迭代都扫描一遍训练数据 (这类算法不适用于海量数据/样本的学习问题)

SMO: 关键点之一

- 每次固定 $m - 2$ 个乘法因子 α_i , 让其余两个任意变化, 寻找这两个任意变化乘法因子的最佳值; 此时为含等式约束的二元二次函数的优化;
- 可以通过约束等式, 把可变参数变成一个, 即获得一元二次函数, 其最大值计算有简单的公式和方法, 可快速计算;

SMO: 关键点之二

- 如何确定可变的两个乘法因子 α_i, α_j , 使得整个循环迭代的次数最少, 实现最快收敛?
 - 随机次序
 - 固定次序
 - *heuristics*/启发式函数确定次序



时空复杂性 SVM时间复杂度较高，适合小样本学习

- 线性空间需求，参照数据集大小 m
- 时间代价：数据集大小的若干倍，视情况而定，看哪个大，如：数据特征维数， sv 的个数的平方等等，有 *heuristics*，很难获得精确估计。

进一步思考

- 计算过程要多次扫描数据集，大数据量时，如何优化？可能大部分数据都非支持向量，可能没太多用处，能否缩小数据集大小？
- 非线性可分的情况如何处理？