

第三章 线性模型

周世斌

中国矿业大学 计算机学院

May. 2022

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

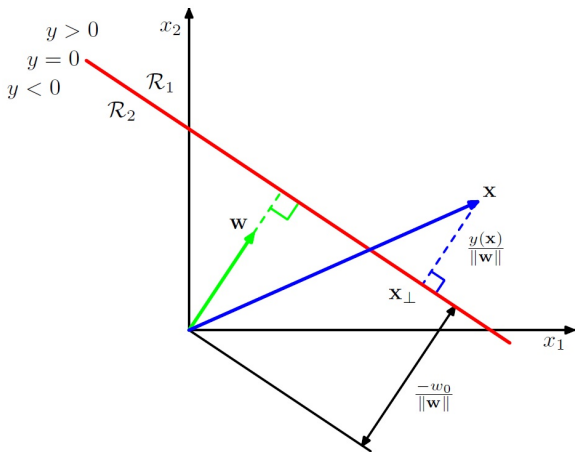
- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

点到超平面的距离

有超平面 $y(\mathbf{x})$, 方程为 $\mathbf{w}^T \mathbf{x} + w_0 = 0$, 设点 \mathbf{x} 到 $y(\mathbf{x})$ 上投影点为 \mathbf{x}_\perp



则点 \mathbf{x} 到超平面的距离为 $\frac{|y(\mathbf{x})|}{\|\mathbf{w}\|}$ 。

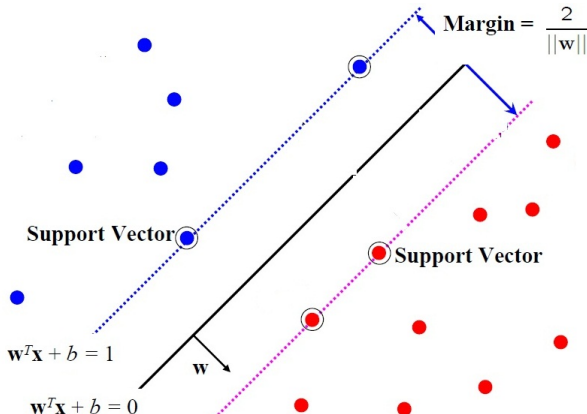


问题：如何推出？

两个分割超平面的距离

我们对最近点 \mathbf{x} 到超平面的距离为 $\frac{|y(\mathbf{x})|}{\|\mathbf{w}\|}$ 分子部分设定为 1。因此，使得所有的样本点

$$\begin{cases} y(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + w_0 \geq 1 & \mathbf{x}_i \in \text{positive class} \\ y(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1 & \mathbf{x}_i \in \text{negative class} \end{cases}$$



最大化两个超平面的距离

并且我们使上图中和决策面平行的两个超平面直接的距离最大化，即

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

这是个二次优化问题，当然还需满足下面的约束

$$\begin{cases} \mathbf{w}^\top \mathbf{x}_i + w_0 \geq 1 & \mathbf{x}_i \in \text{positive class} \\ \mathbf{w}^\top \mathbf{x}_i + w_0 \leq -1 & \mathbf{x}_i \in \text{negative class} \end{cases}$$
$$\implies y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1; \quad \begin{cases} y_i = 1 & \mathbf{x}_i \in \text{positive class} \\ y_i = -1 & \mathbf{x}_i \in \text{negative class} \end{cases}$$

这就是支持向量机 (Support Vector Machine, SVM) 原理。

支持向量机就可以表述为带约束的非线性优化问题:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1 \end{aligned}$$


- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机**
 - 最大化超平面间距离
 - **最优性条件**
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

非线性优化问题

带约束的非线性优化问题一般形式为：

$$\begin{array}{ll}\min & f(\mathbf{x}) \\ \text{s.t.} & g_i(\mathbf{x}) \geq 0 \quad i = 1, \dots, m\end{array}$$

- 满足 $g_i(\mathbf{x}) \geq 0$ 的 \mathbf{x} 取值范围称为可行域。
- 假设 \mathbf{x}^* 是最小点，如果存在 j ，使得 $g_j(\mathbf{x}^*) > 0$ ，则我们可将第 j 个约束去掉， \mathbf{x}^* 仍是去掉第 j 个约束的最小点。
- $g_i(\mathbf{x}^*) = 0$ 是积极约束。 $g_j(\mathbf{x}^*) > 0$ 是非积极约束。

 二次优化问题： $f(\mathbf{x})$ 是二次函数， $g_i(\mathbf{x})$ 都是线性函数。标准型式为：

$$\begin{array}{ll}\min & f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x} + \mathbf{g}^\top \mathbf{x} \\ \text{s.t.} & g_i(\mathbf{x}) = \mathbf{a}_i^\top \mathbf{x} + b_i \geq 0 \quad i = 1, \dots, m\end{array}$$

非线性优化之 Lagrange 函数与 KKT 条件

带约束的非线性优化

$$\begin{array}{ll}\min & f(\mathbf{x}) \\ \text{s.t.} & g_i(\mathbf{x}) \geq 0 \quad i = 1, \dots, m\end{array}$$



其 Lagrange 函数为

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{i=1}^n \lambda_i g_i(\mathbf{x}), \quad \boldsymbol{\lambda} \geq 0$$

KKT (Karush-Kuhn-Tucker) 条件

当 \mathbf{x}^* 是最优解时, 有效约束的 $\nabla g_i(\mathbf{x}^*)$ 线性无关, 则

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \nabla f(\mathbf{x}^*) - \sum_{i=1}^n \lambda_i^* \nabla g_i(\mathbf{x}^*) = \mathbf{0}$$

$$\lambda_i^* \geq 0, \quad g_i(\mathbf{x}^*) \geq 0, \quad \lambda_i^* g_i(\mathbf{x}^*) = 0, \quad i = 1, \dots, m$$

非线性优化之鞍点定理

① 因为 $\lambda_i^* g_i(\mathbf{x}^*) = 0$

$$L(\mathbf{x}^*, \boldsymbol{\lambda}) - L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = - \sum (\lambda_i - \lambda_i^*) g_i(\mathbf{x}^*) = - \sum \lambda_i g_i(\mathbf{x}^*) \leq 0 \\ \implies L(\mathbf{x}^*, \boldsymbol{\lambda}) \leq L(\mathbf{x}^*, \boldsymbol{\lambda}^*)$$

② 凸优化问题: $f(\mathbf{x})$ 是凸函数, $g_i(\mathbf{x})$ 是凹函数

$$L(\mathbf{x}, \boldsymbol{\lambda}^*) \geq L(\mathbf{x}^*, \boldsymbol{\lambda}^*) + \nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*)(\mathbf{x} - \mathbf{x}^*) = L(\mathbf{x}^*, \boldsymbol{\lambda}^*)$$

鞍点定理

Lagrange 函数 $L(\mathbf{x}, \boldsymbol{\lambda})$ 满足

$$L(\mathbf{x}^*, \boldsymbol{\lambda}) \leq L(\mathbf{x}^*, \boldsymbol{\lambda}^*) \leq L(\mathbf{x}, \boldsymbol{\lambda}^*)$$

则 $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ 为 $L(\mathbf{x}, \boldsymbol{\lambda})$ 的鞍点。

非线性优化对偶问题

由鞍点定理，得，

$$\max_{\boldsymbol{\lambda}} L(\mathbf{x}^*, \boldsymbol{\lambda}) = \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}^*)$$

所以称

$$Q(\boldsymbol{\lambda}) = L(\mathbf{x}^*, \boldsymbol{\lambda})$$

为 $f(\mathbf{x}) = L(\mathbf{x}, \boldsymbol{\lambda}^*)$ 的对偶问题，即

对偶问题

$$\max_{\boldsymbol{\lambda}} Q(\boldsymbol{\lambda}) \iff \min_{\mathbf{x}} f(\mathbf{x})$$

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

支持向量超平面的距离最大化

支持向量机就是用超平面 $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 = 0$ 将训练集 $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ 分隔开来, 由此得到带约束的非线性优化问题:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + w_0) \geq 1 \end{aligned}$$

👉 思考: y_i 的取值范围

这个优化问题一般不容易处理的, 可以求其对偶优化问题。

👉 原问题最优解为 Lagrange 函数的鞍点


$$L(\mathbf{w}, w_0) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n \alpha_i \left\{ y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 \right\}$$

$$\left\{ \begin{aligned} \frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i = 0 \\ \frac{\partial L}{\partial w_0} &= \sum_{i=1}^n y_i \alpha_i = 0 \end{aligned} \right. \implies \left\{ \begin{aligned} \mathbf{w} &= \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i \\ \sum_{i=1}^n y_i \alpha_i &= 0 \end{aligned} \right.$$

SVM 的对偶优化问题

原问题的对偶优化问题为：

$$\begin{aligned} \min \quad & W(\boldsymbol{\alpha}) = -\sum_{i=1}^n \alpha_i + \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \quad \forall \alpha_i \geq 0 \end{aligned}$$

 如何导出？

$$\begin{aligned} \Rightarrow \min \quad & W(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{y}^T \boldsymbol{\alpha} = 0 \quad \forall \alpha_i \geq 0 \end{aligned}$$

其中矩阵 \mathbf{Q} 中的元素为

$$Q_{ij} = y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

SVM 的解

可以解得 α_i^* 。同时根据 KKT 定理, α_i^* 最优解还应满足

$$\alpha_i \left\{ y_i (\mathbf{w}^T \mathbf{x}_i + w_0) - 1 \right\} = 0, \quad \forall i$$

可以看出, 只有支持向量的系数 α_i^* 不为零, 所以 \mathbf{w}^* 可以写成

$$\mathbf{w}^* = \sum_{SV} \alpha_i^* y_i \mathbf{x}_i$$

w_0 的值没有出现在对偶问题中, 利用原始约束可以找到 w_0^* :

$$\begin{aligned} \mathbf{w}^* \cdot \mathbf{x} &= \left(\sum_{SV} \alpha_i^* y_i \mathbf{x}_i \right) \cdot \mathbf{x} = \sum_{SV} \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) \\ w_0^* &= \frac{1}{2} (\mathbf{w}^* \cdot \mathbf{x}_+ + \mathbf{w}^* \cdot \mathbf{x}_-) \end{aligned}$$

最后我们得到超平面为

$$f(\mathbf{x}) = \mathbf{w}^* \cdot \mathbf{x} + w_0^*$$

$$f(\mathbf{x}) = \sum_{SV} \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) + w_0^*$$

非线性 SVM 问题的基本思想，通过非线性变换，将输入变量转化到某个高维空间中，然后再变换空间求最优分类面。

注意到，上面的对偶问题都只涉及训练样本之间的内积运算，根据相应的数学理论，只要一种核函数满足一定的条件（Mercer 条件），就对应某一变换空间的内积。

首先，通过非线性映射，

$$\phi : R^n \rightarrow H$$

将输入空间映射到高维空间 H 中。定义核函数

$$\kappa(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$$

SVM 中目前研究最多的核函数主要有三类。

- 多项式核函数

$$\kappa(\mathbf{x}_i, \mathbf{x}) = (\gamma \langle \mathbf{x}_i, \mathbf{x} \rangle + \beta)^q$$

所得到的是 q 阶多项式分类器;

- 径向基函数 (RBF)

$$\kappa(\mathbf{x}_i, \mathbf{x}) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}\|^2)$$

所得分类器与传统 RBF 方法的重要区别是：这里每个基函数中心对应一个支持向量。它们及输出权值都是由算法自动确定的。

- Sigmoid 函数

$$\kappa(\mathbf{x}_i, \mathbf{x}) = \tanh(\nu(\mathbf{x}_i, \mathbf{x}) + c)$$

LIBSVM 的介绍

LIBSVM 是台湾大学林智仁 (Chih-Jen Lin) 博士等开发设计的一个操作简单、易于使用、快速有效的通用 SVM 软件包，可以解决

- 分类问题（包括 C-SVC、 ν -SVC）
- 回归问题（包括 ϵ -SVR、 ν -SVR）
- 以及分布估计（one-class-SVM）

等问题，提供了线性、多项式、径向基和 S 形函数四种常用的核函数供选择，可以有效地解决多类问题、交叉验证选择参数、对不平衡样本加权、多类问题的概率估计等。LIBSVM 是一个开源的软件包，需要者都可以免费的从作者的个人主页 <http://www.csie.ntu.edu.tw/~cjlin/> 处获得。

Welcome to Chih-Jen Lin's Home Page



Research

- [Machine Learning:](#)

LIBSVM 在给出源代码的同时还提供了 Windows 操作系统下的可执行文件，包括：

- 进行支持向量机训练的 `svmtrain.exe`;
- 根据已获得的支持向量机模型对数据集进行预测的 `svmpredict.exe`;
- 以及对训练数据与测试数据进行简单缩放操作的 `svmscale.exe`

LIBSVM 使用的一般步骤是：

- ① 按照 LIBSVM 软件包所要求的格式准备数据集；
- ② 对数据进行简单的缩放操作；
- ③ 考虑选用 RBF 核函数

$$\kappa(\mathbf{x}_i, \mathbf{x}) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}\|^2)$$

- ④ 采用交叉验证选择最佳参数 C 与 γ ；
- ⑤ 采用最佳参数 C 与 γ 对整个训练集进行训练获取支持向量机模型；
- ⑥ 利用获取的模型进行测试与预测。

C-Support Vector Classification

Given training vectors $\mathbf{x}_i \in R^n, i = 1, \dots, l$, in two classes, and an indicator vector $\mathbf{y} \in R^l$ such that $y_i \in \{1, -1\}$, C-SVC (Boser et al., 1992; Cortes and Vapnik, 1995) solves the following primal optimization problem.

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \xi_i \\ \text{subject to} \quad & y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \dots, l, \end{aligned}$$

where $\phi(\mathbf{x}_i)$ maps \mathbf{x}_i into a higher-dimensional space and $C > 0$ is the regularization parameter. Due to the possible high dimensionality of the vector variable \mathbf{w} , usually we solve the following dual problem.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T Q \alpha - \mathbf{e}^T \alpha \\ \text{subject to} \quad & \mathbf{y}^T \alpha = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, l, \end{aligned}$$

LIBSVM 使用的训练数据和测试数据文件格式如下：

$\langle label \rangle \langle index1 \rangle \langle value1 \rangle \langle index2 \rangle \langle value2 \rangle \dots$

其中

- $\langle label \rangle$ 是训练数据集的目标值，对于分类，它是标识某类的整数 (支持多个类)；对于回归，是任意实数。
- $\langle index \rangle$ 是以 1 开始的整数，表示特征的序号；
- $\langle value \rangle$ 为实数，也就是我们常说的特征值或自变量。当特征值为 0 时，特征序号与特征值 $value$ 都可以同时省略，即 $index$ 可以是不连续的自然数。

$\langle label \rangle$ 与第一个特征序号、前一个特征值与后一个特征序号之间用空格隔开。测试数据文件中的 $label$ 只用于计算准确度或误差，如果它是未知的，只需用任意一个数填写这一栏，也可以空着不填。

例如：

+1 1 : 0.708 2 : 1 3 : 1 4 : -0.320 5 : -0.105 6 : -1 8 : 1.21

`svmtrain` 实现对训练数据集的训练，获得 SVM 模型。

用法：

```
svmtrain [options] training_set_file [model_file]
```

其中，`options`（操作参数）可用的选项即表示的涵义如下所示

- `-s svm` 类型：设置 SVM 类型，默认值为 0，可选类型有：
 - ① `C-SVC`
 - ① `ν -SVC`
 - ② `one-class-SVM`
 - ③ `ν -SVR`
 - ④ `ν -SVR`
- `-t` 核函数类型：设置核函数类型，默认值为 2，可选类型有：
 - ① 线性核
 - ① 多项式核
 - ② RBF 核
 - ③ sigmoid 核

使用实例：

```
svmtrain train3.scale train3.model
```

`svmpredict` 是根据训练获得的模型，对数据集进行预测。
用法：

svmpredict [options] test_file model_file output_file

options（操作参数）：

- `-b probability_estimates`：是否需要进行概率估计预测，可选值为 0 或者 1，默认值为 0。
- `model_file` 是由 `svmtrain` 产生的模型文件；
- `test_file` 是要进行预测的数据文件；
- `output_file` 是 `svmpredict` 的输出文件，表示预测的结果值。
`svmpredict` 没有其它的选项。

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

方差与偏差

对于线性模型

$$y = f(\mathbf{x})$$

其相应的概率版本为

$$Y = f(\mathbf{x}) + \varepsilon$$

其中 Y, ε 为随机变量, x 为确定型变量。最小均方误差 (MSE) 为:

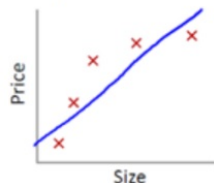
$$\begin{aligned}\text{MSE}(\mathbf{x}) &= \mathbb{E}(\hat{y} - f(\mathbf{x}))^2 \\ &= \mathbb{E}(\hat{y} - \mathbb{E}(\hat{y}))^2 + (\mathbb{E}(\hat{y}) - f(\mathbf{x}))^2 \\ &= \text{var}(\hat{y}) + \text{Bias}^2(\hat{y})\end{aligned}$$

the least squares estimates often have low bias (偏差) but large variance (方差). Prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.

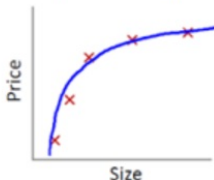
Underfitting and Overfitting

The Problem of Overfitting

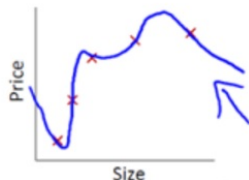
Example: Linear regression (housing prices)



$\rightarrow \theta_0 + \theta_1 x$
"Underfit" "High bias"



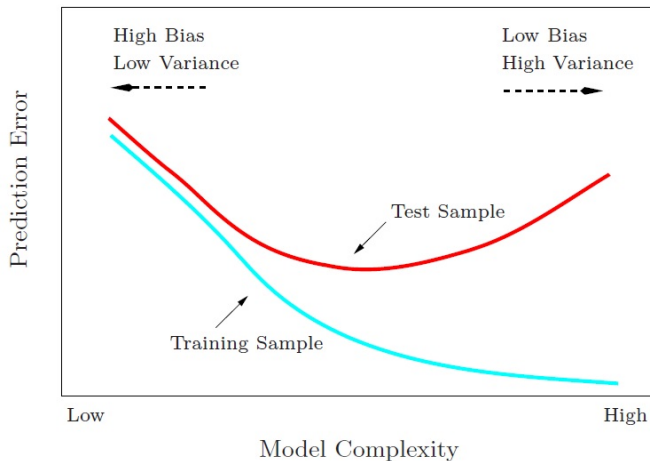
$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2$
"Just right"



$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$
"Overfit" "High variance"

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \approx 0$), but fail to generalize to new examples (predict prices on new examples).

Model Complexity



Test and training error as a function of model complexity.

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

岭回归

Ridge regression shrinks the regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum of squares,

$$\text{Min} \quad L(\mathbf{w}) = \sum_{i=1}^n (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

$$\text{s.t.} \quad \sum_{i=1}^p w_i^2 \leq s$$

$$\Rightarrow \mathbf{w}^{\text{ridge}} = \arg \min_{\mathbf{w}} \left\{ (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w} \right\}$$

$$\Rightarrow \mathbf{w}^{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrink-age: the larger the value of λ , the greater the amount of shrinkage. The coefficients are shrunk toward zero (and each other).

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

LASSO 是由 1996 年 Robert Tibshirani 首次提出, 全称 Least absolute shrinkage and selection operator. 该方法是一种压缩估计。它通过构造一个惩罚函数得到一个较为精炼的模型, 使得它压缩一些系数, 同时设定一些系数为零。因此保留了子集收缩的优点。

The lasso is a shrinkage method like ridge, with subtle but important differences. The lasso estimate is defined by

$$\text{Min} \quad L(\mathbf{w}) = \sum_{i=1}^n (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

$$\text{s.t.} \quad \sum_{i=1}^p |w_i| \leq s$$

$$\implies \mathbf{w}^{\text{lasso}} = \arg \min_{\mathbf{w}} \left\{ (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \|\mathbf{w}\|_1 \right\}$$

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - **Regularization**
 - Sparsity

早在 20 世纪初 Hadamard 观测到关于求解线性方程

$$Af = F, f \in \mathcal{F}$$

满足此类线性算子方程的函数 f 是不适定 (ill-posed) 问题, 即使方程存在唯一解, 如果方程右边有一个微小变动 (如用 $\|F - F_\delta\| < \delta$ 任意小的 F_δ 取代 F), 也会导致解有很大的变化 (即可能导致 $\|f_\delta - f\|$ 很大)。在这种情况下, 如果方程右边的 F 是不准确的 (比如等于 F_δ , 而 F_δ 与 F 相差某个 δ 水平的噪声), 那么使

$$L(f) = \|Af - F\|^2$$

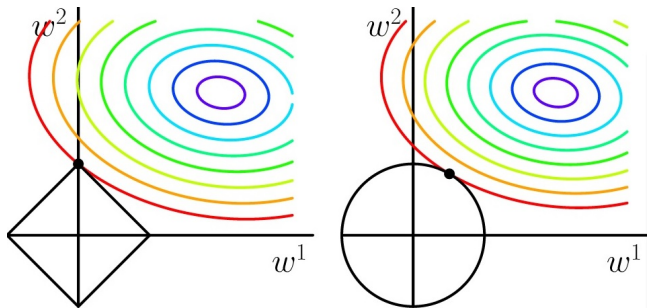
最小化的 F_δ 并不能保证在 δ 趋于 0 时是方程真实解的一个好的近似。00 年代中期, 人们发现, 如果不是最小化 $L(f)$ 而是最小化另一个称作正则化

$$L^*(f) = \|Af - F\|^2 + \lambda J(f)$$

$J(f)$ 称为罚函数。

- 1 引言
- 2 线性模型（最小二乘法求解）
- 3 主成份分析，降维方法选讲
- 4 线性判别分析
- 5 最大间隔准则与支持向量机
 - 最大化超平面间距离
 - 最优性条件
 - 支持向量机（SVM）
- 6 Shrinkage Methods and Regularization
 - Ridge Regression
 - Lasso
 - Regularization
 - Sparsity

在生物或医学领域，稀疏解除了计算上的好处外，更重要的是可解释性，比如说，一个病，如果依赖于 5 个变量的话，将会更易于医生解释、描述和总结规律。如果依赖于 5000 个变量，基本就超出了人可处理的范围了。



Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|w_1| + |w_2| \leq t$ and $w_1^2 + w_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.

压缩感知

压缩感知是由 E. J. Candes、J. Romberg、T. Tao 和 D. L. Donoho 等科学家于 2004 年提出的。

可能第一个与稀疏信号恢复有关的算法由法国数学家 Prony 提出。这个被称为的 Prony 方法的稀疏信号恢复方法可以通过解一个特征值问题，从一小部分等间隔采样的样本中估计一个稀疏三角多项式的非零幅度和对应的频率。而最早采用基于 L1 范数最小化的稀疏约束的人是 B. Logan。他发现在数据足够稀疏的情况下，通过 L1 范数最小化可以从欠采样样本中有效的恢复频率稀疏信号。D. Donoho 和 B. Logan 是信号处理领域采用 L1 范数最小化稀疏约束的先驱。但是地球物理学家早在 20 世纪七八十年代就开始利用 L1 范数最小化来分析地震反射信号了。上世纪 90 年代，核磁共振谱处理方面提出采用稀疏重建方法从欠采样非等间隔样本中恢复稀疏 Fourier 谱。同一时期，图像处理方面也开始引入稀疏信号处理方法进行图像处理。在统计学方面，使用 L1 范数的模型选择问题和相关的方法也在同期开始展开。

压缩感知在矩阵分解中的应用广泛。包括主成分分析、表示字典学习、非负矩阵分解、多维度向量估计、低秩或满秩矩阵恢复问题等。