机器学习（本科生公选课）GEC6531

第10节 损失函数和正则化 Loss Function and Regularization

计算机科学与技术学院

张瑞 教授

邮箱：ruizhang6@hust.edu.cn

签到 & 思考

■ 微助教签到（学校要求）

1. 加入课堂：微信扫码或者通过微助教公众号

2. 微信扫码签到

回顾SVM的损失函数

今天的目录

- **经验风险最小化**
  - 背景和定义
- **二分类损失函数**
- **回归损失函数**
- **不适定问题（ILL-posed Problems）**
  - 无关参数的问题
  - 不适定问题
- **正则化**
  - 定义和案例
  - 正则项
- **常见例子**

今天的目录

- **经验风险最小化**
  - 背景和定义
- **二分类损失函数**
- **回归损失函数**
- **不适定问题（ILL-posed Problems）**
  - 无关参数的问题
  - 不适定问题
- **正则化**
  - 定义和案例
  - 正则项
- **常见例子**

SVM的损失函数

对于无约束SVM 公式：

$$\min_{\mathbf{w}} \underbrace{C\sum_{i=1}^{n} \max[1 - y_i\underbrace{(w^{\top}\mathbf{x}_i + b)}_{h(\mathbf{x}_i)}, 0]}_{Hinge-Loss} + \underbrace{\|w\|_z^2}_{l_2-Regularizer}$$

SVM 的损失函数用的是 Hinge-Loss形式，而ℓ_2-正则项反映了求解的复杂性，并惩罚复杂的解。这是一个经验风险最小化的例子，它有一个损失函数 ℓ 和一个正则项 r：

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \underbrace{l(h_{\mathbf{w}}(\mathbf{x}_i), y_i)}_{Loss} + \underbrace{\lambda r(w)}_{Regularizer}$$

其中损失函数是惩罚训练误差的连续函数，正则项是惩罚分类器复杂度的连续函数。

这里，将 λ 定义为上一讲中的 $\frac{1}{C}$

监督学习的设置

- 监督学习问题的一般设置：
  有两个对象空间 X 和 Y，希望学习一个函数 $h: X \to Y$，对给定的 $x \in X$，输出 $y \in Y$。
- 有一个训练集，包含 n 个样例 $(x1, y_1), \ldots, (x_n, y_n)$。其中 $x_i \in X$ 是输入且 $y_i \in Y$ 是希望从 $h(x_i)$ 得到的输出。
- 形式化地描述，即：假设有一个在 X 和 Y 上的**联合概率分布** $p(x, y)$，训练集包含 n 个从 $p(x, y)$ 中 i.i.d. 采样得到的实例 $(x_1, y_1), \ldots, (x_n, y_n)$。
- 联合概率分布的假设允许我们对预测中的不确定性建模。因为对于固定的 x，y 不是 x 的确定函数，而是一个具有条件分布的随机变量 $p(x|y)$。

经验风险最小化（Empirical Risk Minimization，ERM）

假设我们得到了一个非负实值的损失函数 $L(\hat{y}, y)$，用于比较假设的预测 $\hat{y}$ 与真实结果 y 的差异。与假设 $h(x)$ 相关的风险定义为损失函数的期望:

$$R(h) = E[L(h(x), y)] = \int L(h(x), y) \, \mathrm{d} \, P(x, y)$$

学习算法的目标是在一组固定的函数 h 中找到一个假设 h^*，使 $R(h)$ 的风险最小:

$$h^* = \underset{h \in H}{argmin} R(h).$$

一般来说，风险 $R(h)$ 无法计算，因为 $P(x, y)$ 的分布对于学习算法来说是未知的 (这种情况被称为不可知论学习)。不过，我们可以通过平均训练集上的损失函数计算一个近似值，称为经验风险:

$$R_{emp}(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), y_i).$$

经验风险最小化 (ERM) 是统计学习理论的一个原则。它定义了一系列学习算法，用于给出其性能的理论上界。

今天的目录

- **经验风险最小化**
  - 背景和定义
- **二分类损失函数**
- **回归损失函数**
- **不适定问题（ILL-posed Problems）**
  - 无关参数的问题
  - 不适定问题
- **正则化**
  - 定义和案例
  - 正则项
- **常见例子**

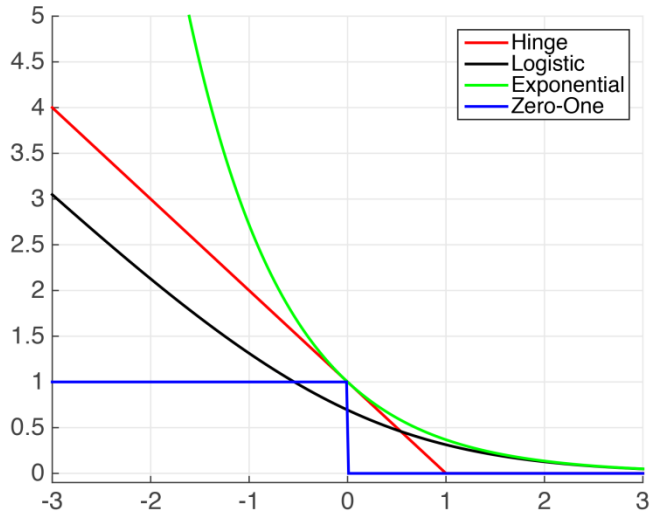
二分类损失函数

不同的机器学习算法使用不同的损失函数

Table 1: 二分类损失函数，$y \in \{-1, +1\}$

损失 $\ell(h_{\mathbf{w}}(\mathbf{x}_i, y_i))$	使用模型	评价
Hinge-Loss: $\max\left[1 - h_{\mathbf{w}}(\mathbf{x}_i)y_i, 0\right]^p$	• 标准 SVM($p = 1$) • (可微的) 均方 Hingeless SVM ($p = 2$)	当用于标准 SVM 时，损失函数表示线性分隔面与其在任何一类中的最近点之间的间隔的大小。只有当 $p = 2$ 时才可微。
Log-Loss: $\log(1 + e^{-h_{\mathbf{w}}(\mathbf{x}_i)y_i})$	逻辑回归	机器学习中最流行的损失函数之一，因为它的输出是经过良好校准的概率。

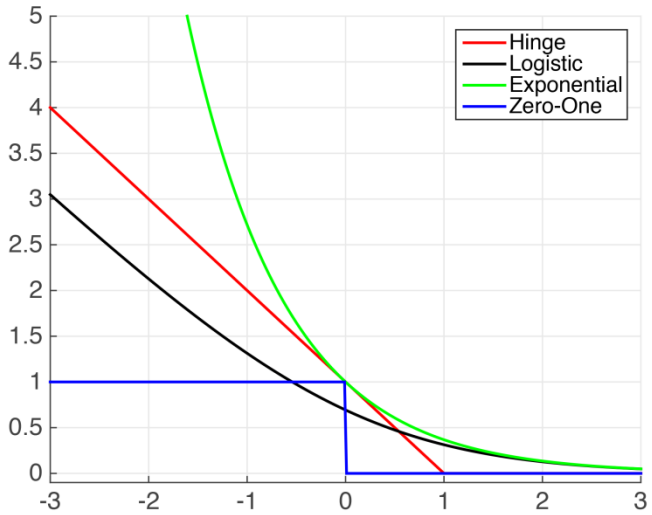
图1: 常见分类损失函数图。x 轴: *h(xi)yi （预测准确率）; y 轴: 损失值*



二分类损失函数

Table 1: 二分类损失函数，$y \in \{-1, +1\}$

损失函数 $\ell(h_{\mathbf{w}}(\mathbf{x}_i, y_i))$	使用	评价
Exponential Loss: $e^{-h_{\mathbf{w}}(\mathbf{x}_i)y_i}$	AdaBoost	这个函数非常激进。错误预测的损失随着 $-h_{\mathbf{w}}(\mathbf{x}_i)y_i$ 的值呈指数增长。这可能会使其有良好的收敛结果，例如 Adaboost，但它在处理噪声数据时会有问题。
0-1 Loss: $\delta(\text{sign}(h_{\mathbf{w}}(\mathbf{x}_i)) \neq y_i)$	实际分类损失	非连续，其优化在实际应用中非常困难



二分类损失函数

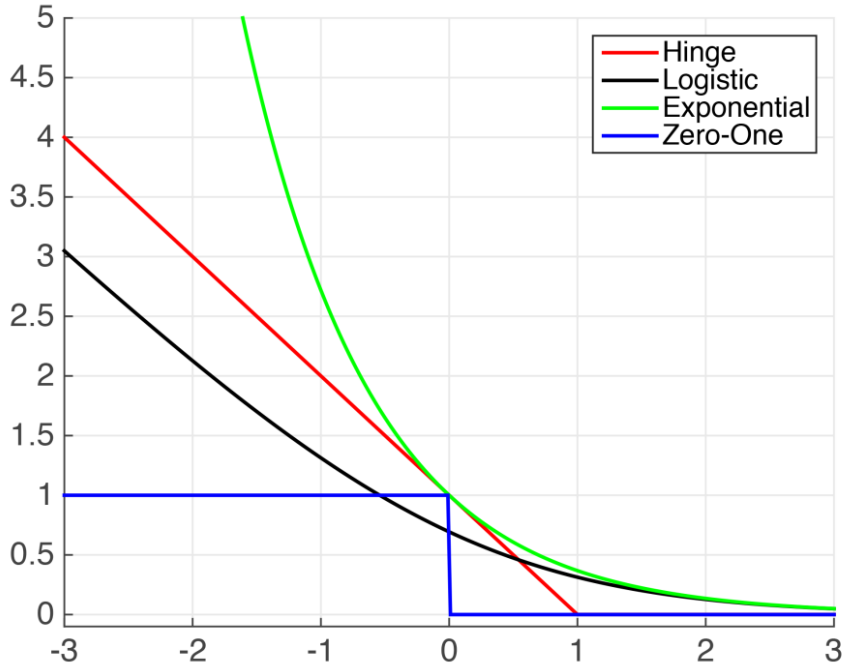


图1: 常见分类损失函数图。x 轴: $h(x_i)\, y_i$ （预测准确率）; y 轴: 损失值

思考: 哪些函数是在0-1 损失的严格上界?

思考: 对于 Hinge-loss 和 log-loss 在 $z \to -\infty$ 时，分析其性质差异?

今天的目录

- **经验风险最小化**
  - 背景和定义
- **二分类损失函数**
- **回归损失函数**
- **不适定问题（ILL-posed Problems）**
  - 无关参数的问题
  - 不适定问题
- **正则化**
  - 定义和案例
  - 正则项
- **常见例子**

回归损失函数

回归算法的损失 (预测可以为任何实数)

表 1: 回归损失函数, $y \in \mathbb{R}$

损失函数 $\ell(h_{\mathrm{w}}(\mathrm{x}_i, y_i))$	评价
均方损失: $(h(\mathrm{x}_i) - y_i)^2$	• 最通用的回归损失函数 • 估计平均 的标签 • 也被称为最小二乘法 (OLS) • 缺点: 对异常值/噪声较敏感
绝对值损失: $\lvert h(\mathrm{x}_i) - y_i \rvert$	• 这也是一个很常用的损失函数 • 估计中位数 的标签 • 优点: 对噪音不敏感 • 缺点: 在 0 处不可微

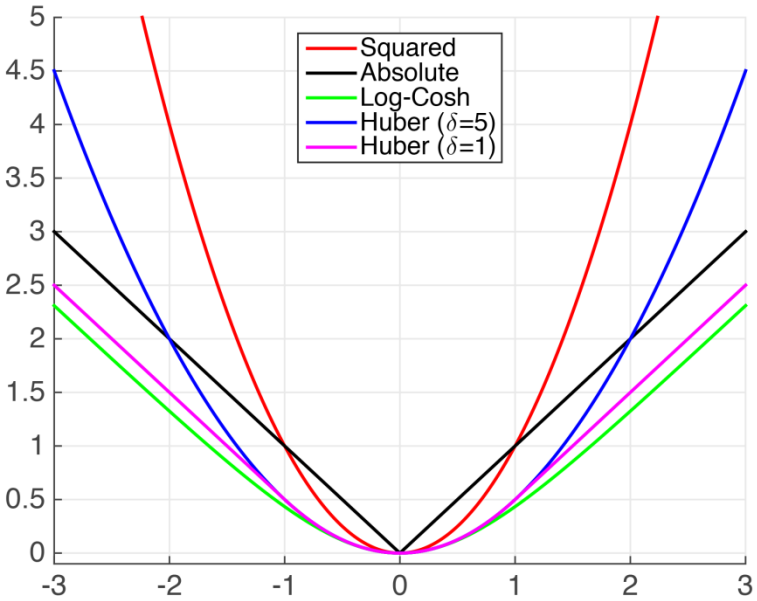


图2: 常见的回归损失函数; x 轴: $h\ (\mathrm{x}_i)\ y_i$ 或预测的“误差”; y 轴: 损失函数值

回归损失函数

回归算法的损失 (预测可以为任何实数)

表 1: 回归损失函数，$y \in \mathbb{R}$

损失函数 $\ell(h_{\mathbf{w}}(\mathbf{x}_i, y_i))$	评价
Huber 损失: • $\frac{1}{2}(h(\mathbf{x}_i) - y_i)^2$ if $\lvert h(\mathbf{x}_i) - y_i \rvert < \delta$, • 否则 $\delta(\lvert h(\mathbf{x}_i) - y_i \rvert - \frac{\delta}{2})$	• 也被称为平滑绝对值损失 • 一次可微 • 在损失较小时表现为平方损失，在损失较大时表现为绝对损失 • 优点: “两全其美”的均方 和绝对 损失
Log-Cosh Loss: $log(cosh(h(\mathbf{x}_i) - y_i))$, $cosh(x) = \frac{e^x + e^{-x}}{2}$	优点: 类似于 Huber 损失，但处处可微

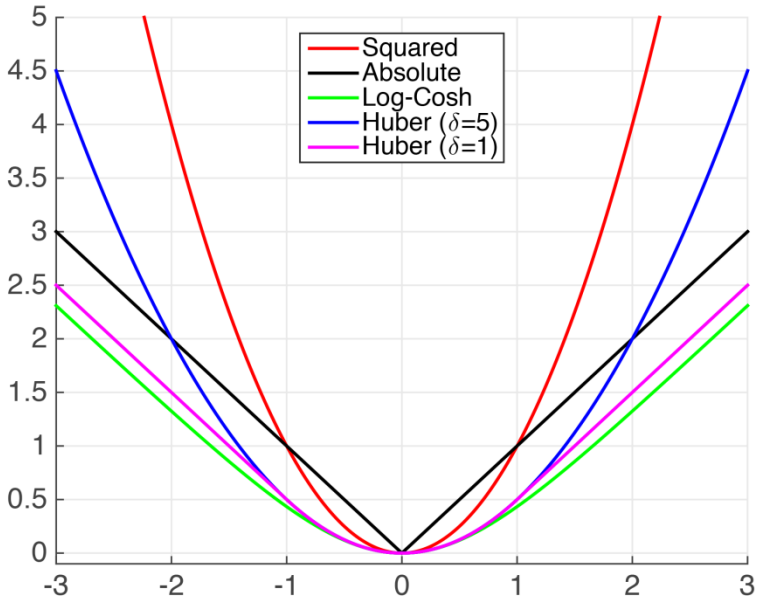


图2: 常见的回归损失函数; x 轴: $h(\mathrm{x}_i) - y_i$ 或预测的“误差”; y 轴: 损失函数值

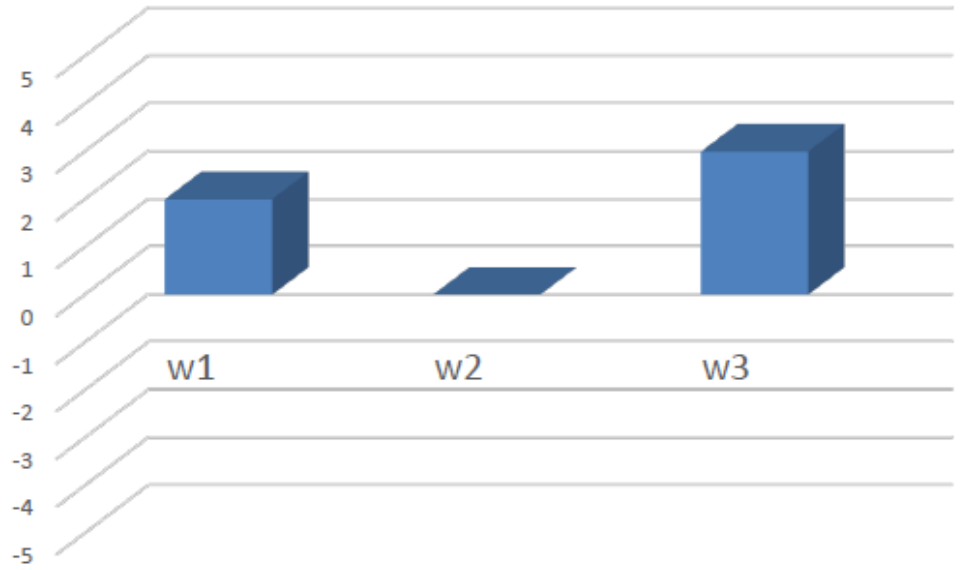
今天的目录

- **经验风险最小化**
  - 背景和定义
- **二分类损失函数**
- **回归损失函数**
- **不适定问题（ILL-posed Problems）**
  - 无关参数的问题
  - 不适定问题
- **正则化**
  - 定义和案例
  - 正则项
- **常见例子**

无关参数的问题：参数重要性

- 3个特征的线性回归模型
  - n = 4条特征数据
  - 模型：$y = w_1x_1 + w_2x_2 + w_3x_3 + w_0$

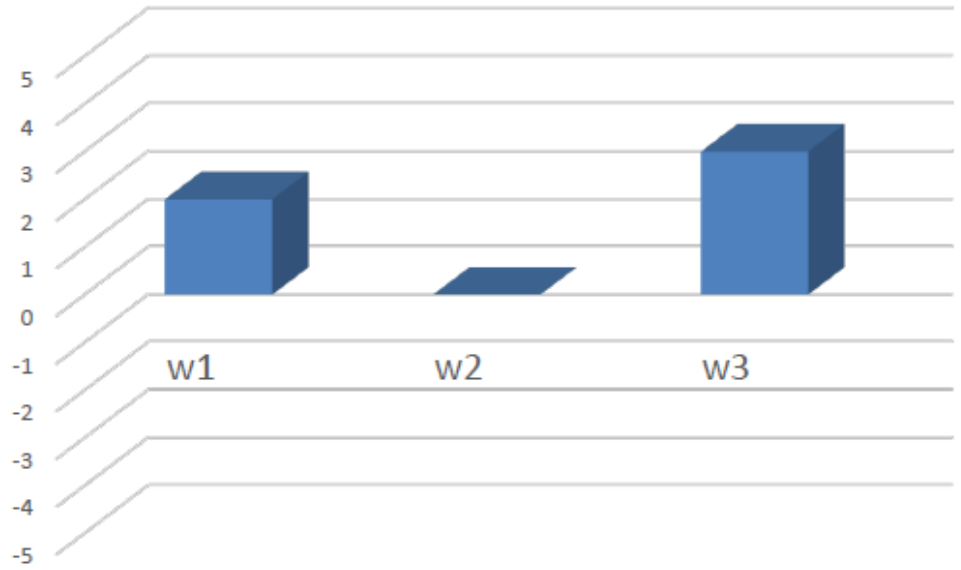
哪个特征更重要?



无关参数的问题：参数重要性

- 3个特征的线性回归模型，**前2列一模一样**
  - n = 4条特征数据
  - 模型：$y = w_1x_1 + w_2x_2 + w_3x_3 + w_0$
  - 特征1和2中有一个是无关参数

3	3	7
6	6	9
21	21	79
34	34	2



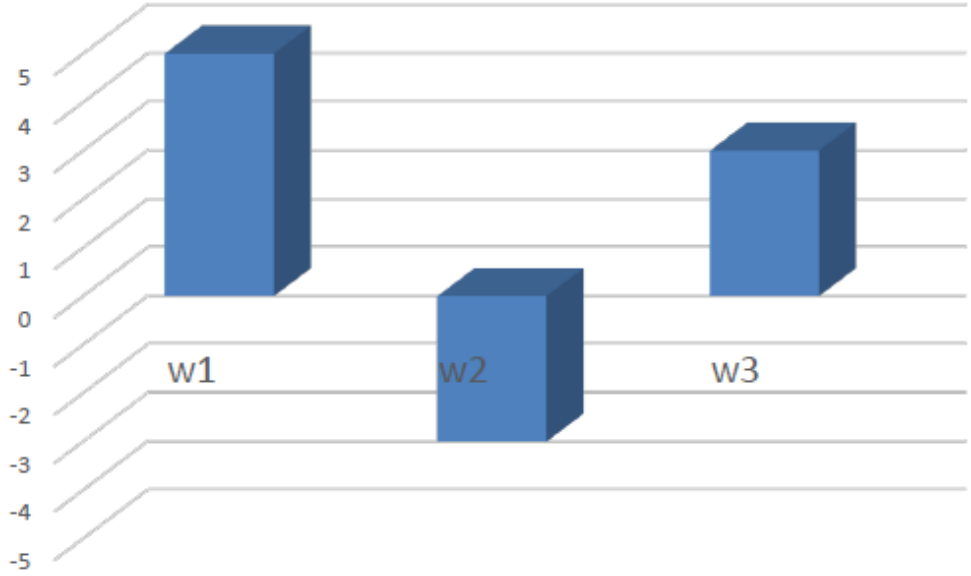
添加扰动对模型的效果如何？

- 加 δ 到 w_1 上
- 从 w_2 减去 δ
- 模型值不变

无关参数的问题：参数重要性

- 3个特征的线性回归模型，**前2列一模一样**
  - n = 4条特征数据
  - 模型：$y = w_1x_1 + w_2x_2 + w_3x_3 + w_0$
  - 特征1和2中有一个是无关参数

3	3	7
6	6	9
21	21	79
34	34	2

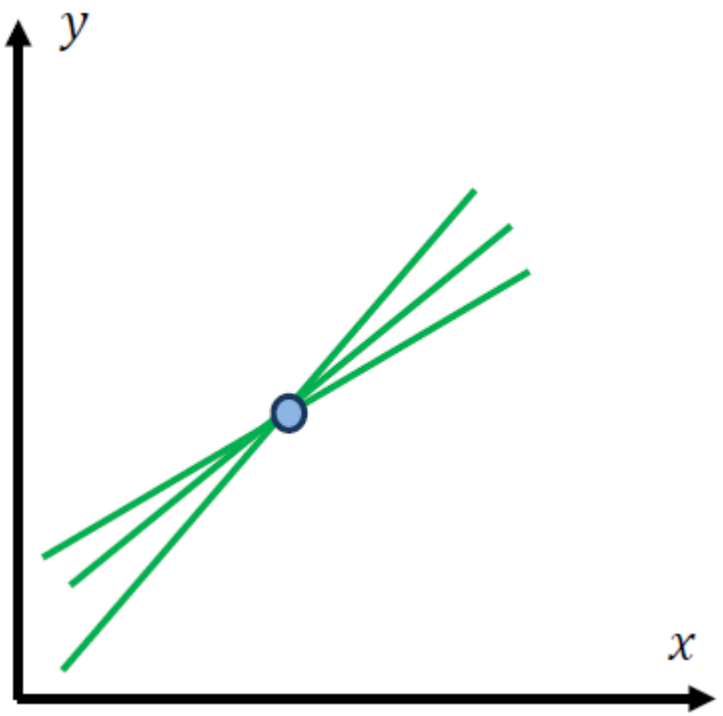


添加扰动对模型的效果如何？

- 加 δ 到 w_1 上
- 从 w_2 减去 δ
- 模型值不变

无关参数的问题：数据缺乏

- 极端例子
  - 模型有2个参数
  - 只有1个数据点
- 无法确定解
- 也是不适定问题/病态问题



无关参数的问题：参数重要性

- 假设 $[\hat{w}_0, \hat{w}_1, \hat{w}_2, \hat{w}_3]'$ 是最优解
- 对于任何 δ, 那么 $[\hat{w}_0, \hat{w}_1 + \delta, \hat{w}_2 - \delta, \hat{w}_3]'$
  - 有同样的预测值
  - 有同样的误差平方和
- 带来如下问题
  - 解不唯一
  - 没有可解释性
  - 使得优化模型参数变成一个**不适定**问题/**病态**问题（ill-posed problem）：问题定义不合理

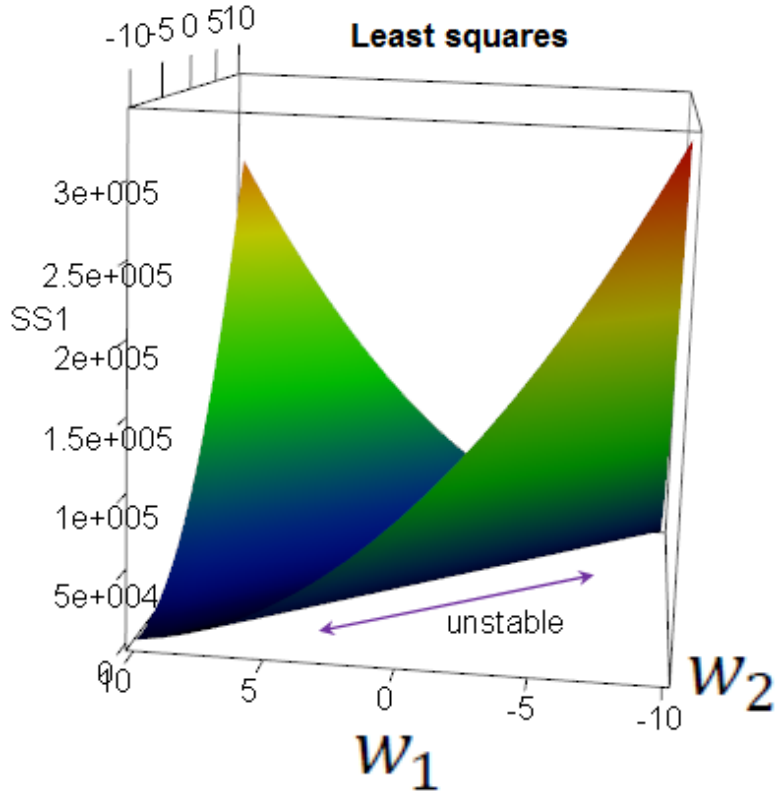
不适定问题（ILL-posed Problems）

- 问题的解无法明确定义
  - w_1和w_1无法唯一确定，有无数可能

- 线性回归的解析解：

$$\mathbf{w} = (\mathbf{X}\mathbf{X}^{\top})^{-1}\mathbf{X}\mathbf{y}$$

- 当存在无关特征时，$\mathrm{X}\ \mathrm{X}^{\mathrm{T}}$没有逆矩阵



凸函数，但不严格

今天的目录

- **经验风险最小化**
  - 背景和定义
- **二分类损失函数**
- **回归损失函数**
- **不适定问题（ILL-posed Problems）**
  - 无关参数的问题
  - 不适定问题
- **正则化**
  - 定义和案例
  - 正则项
- **常见例子**

解决方案：正则化（Regularization）

- 正则化：添加一个条件来约束参数的长度
- 原来的问题要最小化：

$$\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2$$

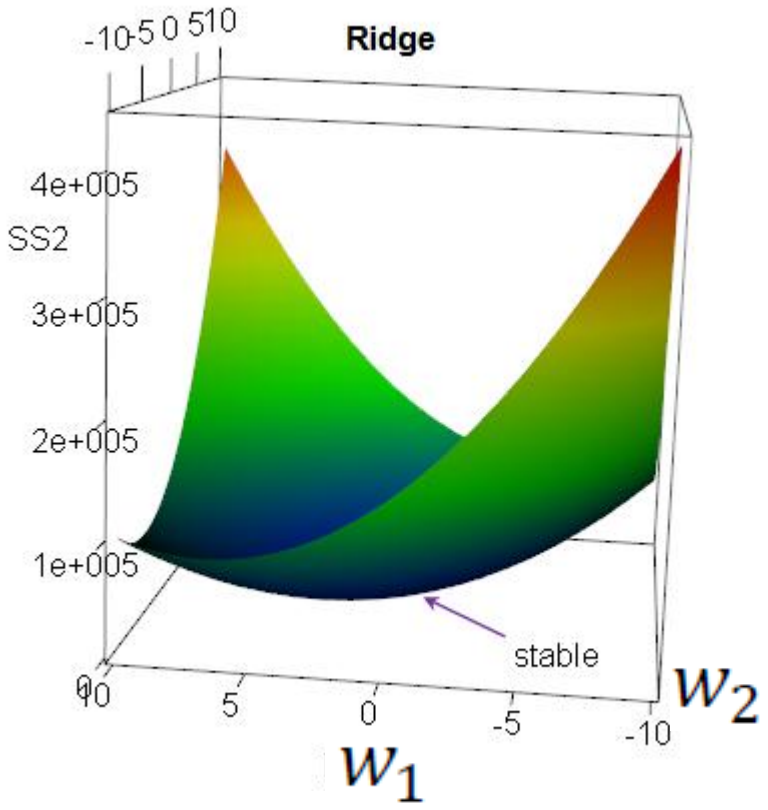
- 正则化后的问题要最小化：

$$\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 \text{ for } \lambda > 0$$

- 问题的解变成

$$\mathbf{w} = (\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{y}$$

- 这个加了正则化的解叫做脊回归（ridge regression）
  - 将线形的“屋脊”变成园拱的单个顶点
  - 将 λ 加入 X X^T的特征向量，变成可逆矩阵



严格凸函数

正则项

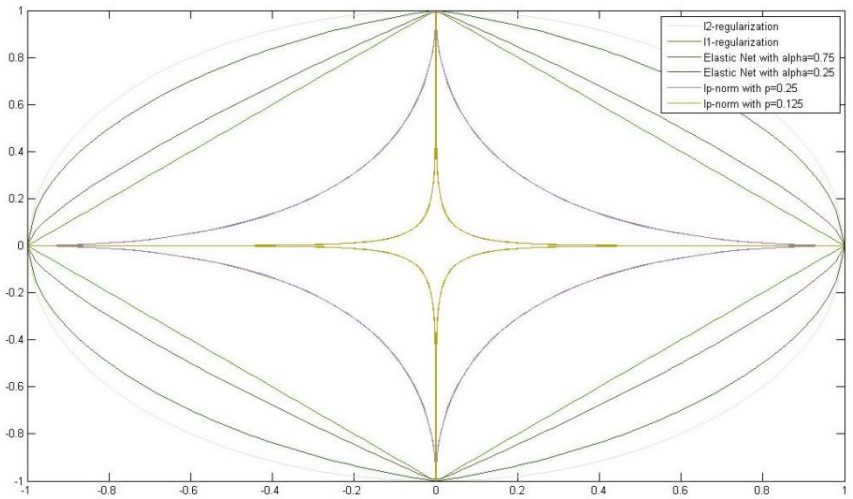
- 在数学、统计学和计算机科学中，特别是在机器学习问题中，正则项是为了解决不适定（ill-posed）问题或防止过拟合而添加的信息。
- 当我们研究正则项时，它有助于改变优化问题的公式，以获得更好的几何直观。
- 在前面的章节中，l_2 在支持向量机中引入正则项发反映解的复杂度限制。

除了 l_2 正则项，其他类型的有效正则项及其特点在下表中列出。

正则项

表 3: 正则项类型

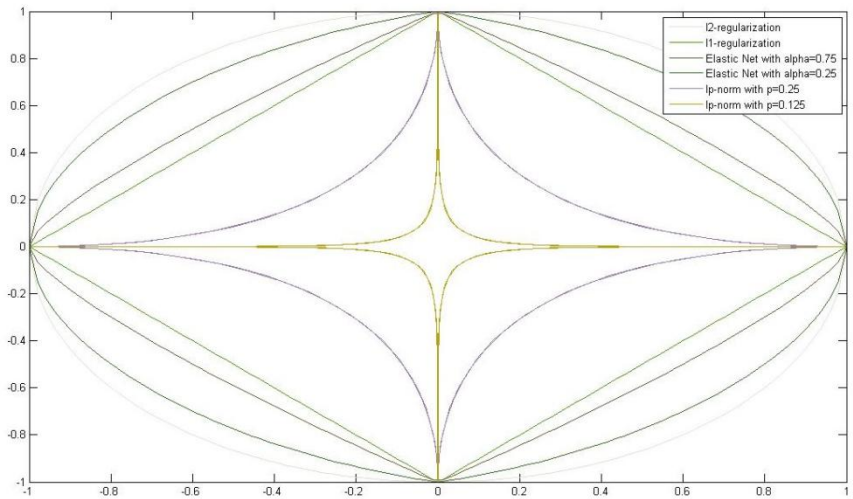
正则项 $r(\mathbf{w})$	性质
l_2-Regularization $r(\mathbf{w}) = \mathbf{w}^\top \mathbf{w} = \|\mathbf{w}\|_2^2$	• 优点: 严格凸; 可微 • 缺点: 对所有特征都有权重，即在某种程度上依赖所有特征 (理想情况下我们希望避免这种情况)–即稠密解。
l_1-Regularization $r(\mathbf{w}) = \|\mathbf{w}\|_1$	• 凸 (但不严格) • 缺点: 在 0 处不可微 (这是最小化的目的) • 效果: 稀疏 解



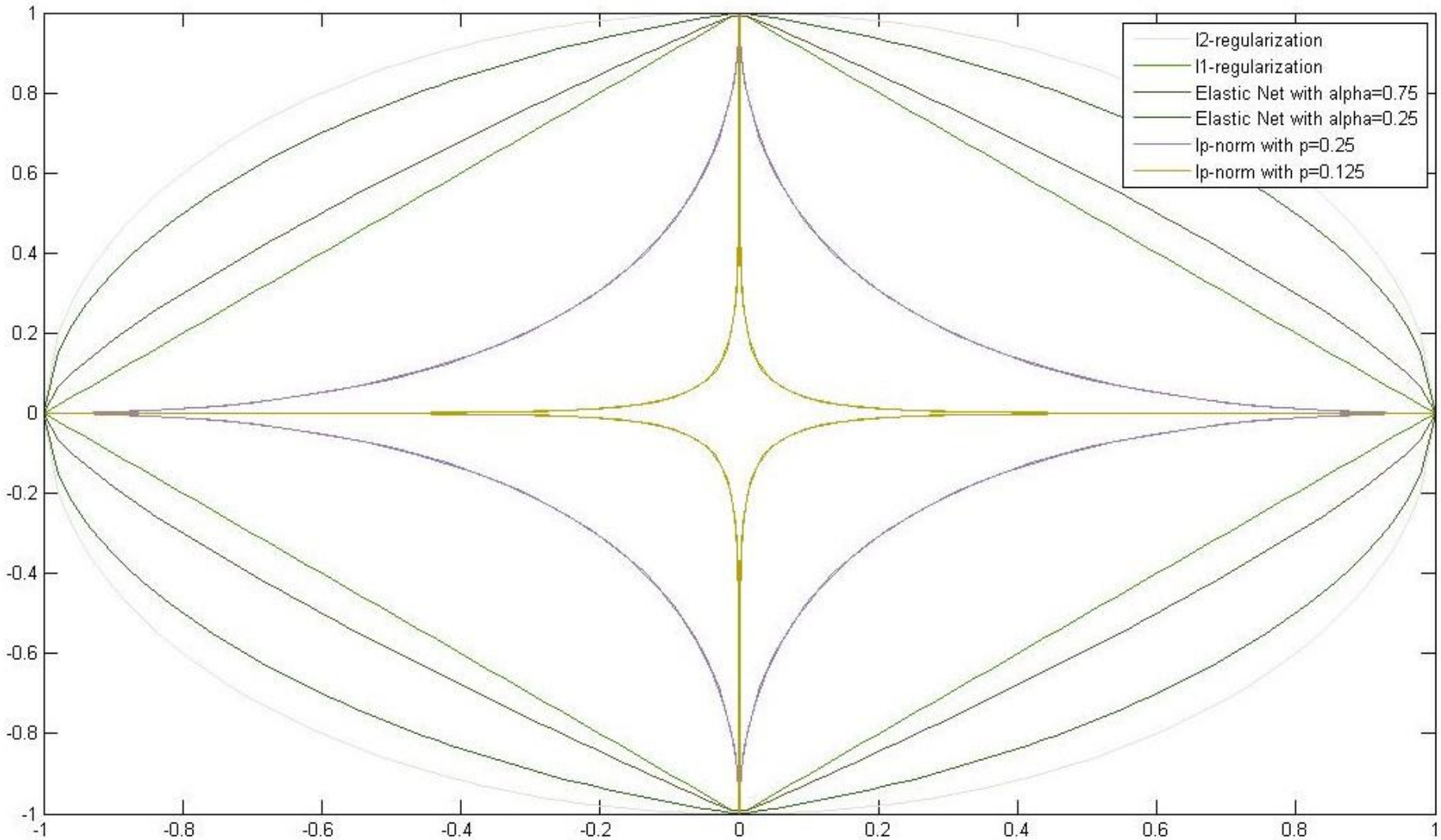
正则项

表 3: 正则项的类型

正则项 $r(\mathbf{w})$	特性
l_p-范数 $\|\mathbf{w}\|_p = (\sum_{i=1}^{d} v_i^p)^{1/p}$	• (往往 $0 < p \leq 1$) • 依赖于初始化 • 优点: 非常稀疏的解 • 缺点: 非凸; 不可微
弹性网络: $\alpha\|\mathbf{w}\|_1 + (1-\alpha)\|\mathbf{w}\|_2^2 \quad \alpha \in [0,1)$	• 优点: 严格凸 (即唯一解) • 优点: 稀疏诱导 (有利于特征选择); 均方损失支持向量机的对偶形式, 见SVEN • 缺点: 不可微

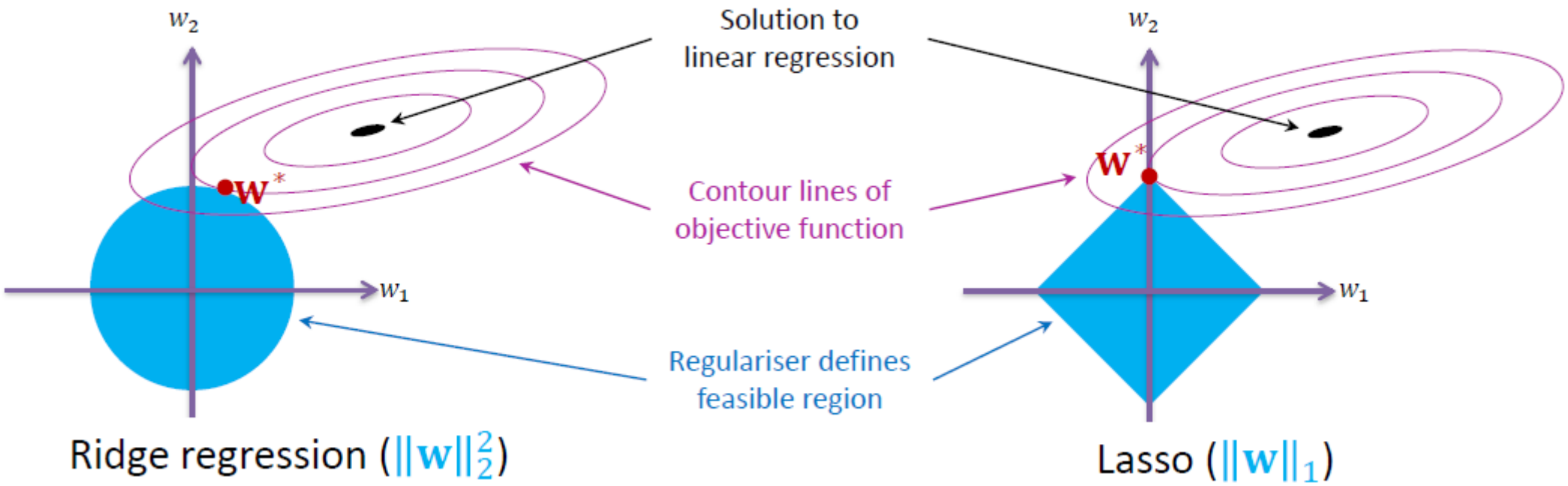


正则项


l2-regularization
l1-regularization
Elastic Net with alpha=0.75
Elastic Net with alpha=0.25
lp-norm with p=0.25
lp-norm with p=0.125
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

把正则项看成约束

$$\text{minimise } \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \textit{ subject to } \|\mathbf{w}\|_2^2 \leq \lambda \text{ for } \lambda > 0$$



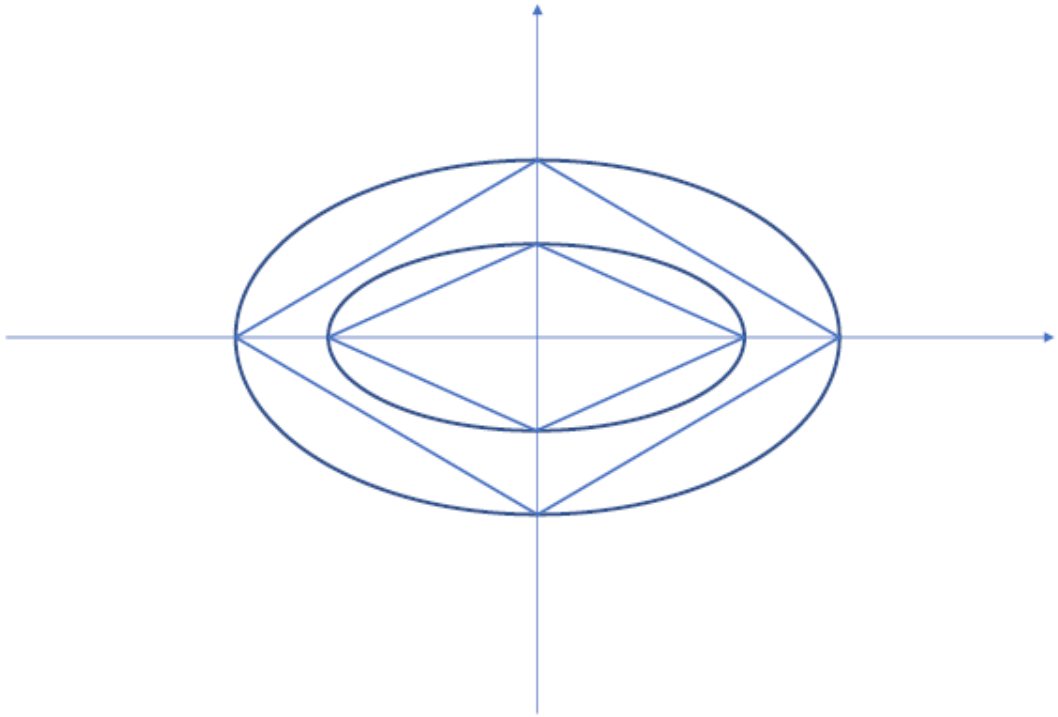
Lasso (L_1 正则) 倾向坐标轴上的解，也就是有些维度是0，解更加稀疏

正则项

思考

以下的损失函数等高线分别对应于哪个不同类型的正则项和超参数？

$(a)\ell(w)+\lambda\|w\|_1$　　$(b)\ell(w)+2\cdot\lambda\|w\|_1$　　$(c)\ell(w)+\lambda\|w\|_2$　　$(d)\ell(w)+2\cdot\lambda\|w\|_2$

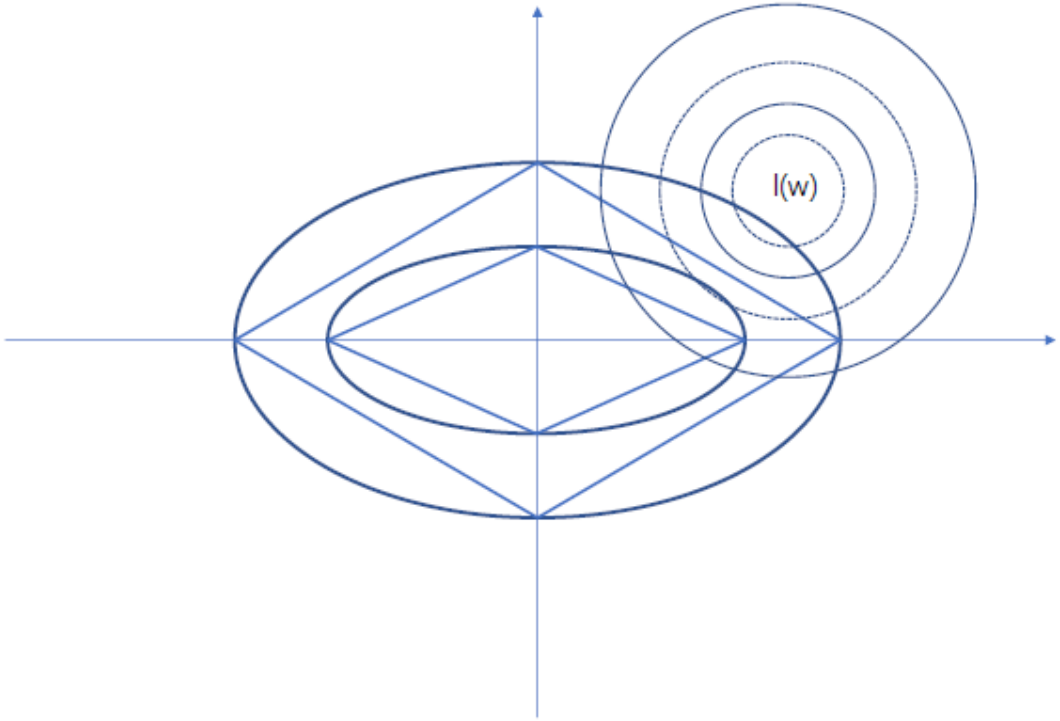


正则项

思考

以下的损失函数等高线分别对应于哪个不同类型的正则项和超参数？

$(a)\ell(w)+\lambda\|w\|_1$　　$(b)\ell(w)+2\cdot\lambda\|w\|_1$　　$(c)\ell(w)+\lambda\|w\|_2$　　$(d)\ell(w)+2\cdot\lambda\|w\|_2$



不同大小与类型正则项损失函数等高线和损失函数等高线相交结果图

今天的目录

- **经验风险最小化**
  - 背景和定义
- **二分类损失函数**
- **回归损失函数**
- **不适定问题（ILL-posed Problems）**
  - 无关参数的问题
  - 不适定问题
- **正则化**
  - 定义和案例
  - 正则项
- **常见例子**

常见例子：最小二乘法 (Ordinary Least Square, OLS)

本节包括一些处理风险最小化的特殊情况，如最小二乘法，岭回归，Lasso 和逻辑回归。下面提供了关于它们的损失函数、正则项以及解的信息。

最小二乘法

在统计学中，最小二乘法 (OLS) 是一种用于估计线性回归模型中未知参数的线性最小二乘方法

目标:

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^\top x_i - y_i)^2$$

评价:

- 平方损失
- 没有正则项
- 闭式解形式: $\mathbf{w} = (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{y}^\top, \mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n], \mathbf{y} = [y_1, ..., y_n]$
- OLS 和主成分分析 (PCA) 的第一主成分之间有一个有趣的联系。PCA 也最小化均方损失，但考虑垂直损失 (每个点与回归线之间的水平距离)。

常见例子：岭回归 (Ridge Regression)

岭回归

Tikhonov 正则项，是不适定（ill-posed）问题最常用的正则项方法。在统计学中，这种方法被称为“岭回归” **目标:**

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^\top x_i - y_i)^2 + \lambda\|w\|_2^2$$

评价:

- 平方损失
- l_2-正则项
- $\mathbf{w} = (\mathbf{X}\mathbf{X}^\top + \lambda\mathbb{I})^{-1}\mathbf{X}\mathbf{y}^\top$
- 如果数据不是非常高维的，速度很快
- 只有一行 Julia / Python 代码

常见例子：Lasso

Lasso

LASSO 是一种回归分析方法，它执行变量选择和正则项，以提高其产生的统计模型的预测准确性和可解释性 **目标:**

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_1$$

评价:

- 优点: 稀疏诱导 (有利于特征选择); 凸性
- 缺点: 非严格凸 (无唯一解); 不可微 (在 0 处)
- 用梯度下降或SVEN求解

常见例子：弹性网络 (Elastic Net)

弹性网络

目标

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \alpha\|\mathbf{w}\|_1 + (1-\alpha)\|\mathbf{w}\|_2^2$$

$$\alpha \in [0, 1)$$

评价:

- 优点: 严格凸 (即唯一解)
- $+$ 稀疏性诱导 (有利于特征选择)
- $+$ 平方损失 SVM 的对偶形式，见SVEN
- 缺点: 不可微
- 用 (次) 梯度下降法或SVEN

常见例子：逻辑回归 (Logistic Regression)

逻辑回归

在回归分析中，**逻辑回归**是估计逻辑模型的参数；这是二项回归的一种形式 **目标:**

$$\min_{\mathrm{w},b} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y_i(\mathrm{w}^\top \mathrm{x}_i + b)}\right)$$

评价:

- 通常是 l_1 或 l_2 正则项
- 用梯度下降法求解
- $\Pr(y|x) = \frac{1}{1+e^{-y(\mathrm{w}^\top x + b)}}$

常见例子：线性支持向量机 (Linear SVM)

线性支持向量机

目标:

$$\min_{\mathbf{w},b} C\sum_{i=1}^{n}\max[1-y_i(\mathbf{w}^\top\mathbf{x}_i+b),0]+\|\mathbf{w}\|_2^2$$

评价:

- 往往使用 l_2 正则项 (有时也使用 l_1)。
- 二次规划
- 可以核化，导致**稀疏**解
- 核化版本可以用专门的算法非常高效地求解 (例如SMO)

总结

- **常用损失函数**
  - 二分类：Hinge-loss, Log-loss, Exponential-loss, 0-1 loss
  - 回归：均方损失，绝对值损失，Huber loss, Log-Cosh loss
- **正则化**
  - 将参数的长度加到损失函数里
- **正则项**
  - 加入参数的 l_p 范数
- **常见例子**
  - 最小二乘法、岭回归、Lasso、弹性网络、逻辑回归、线性SVM