

Lecture 6: Support Vector Regression

课件链接: [Hsuan-Tien Lin - support vector regression](#)

Support Vector Regression(支撑向量回归)

- Kernel Ridge Regression: 核岭回归
- Support Vector Regression Primal: SVR的原始形式
- Support Vector Regression Dual: SVR的对偶形式
- Summary of Kernel Models: 核模型的总结

1. Kernel Ridge Regression: 核岭回归

Ridge Regression, 即"岭回归", 是L2正则化线性回归。上一章我们介绍了Representer Theorem——任何L2正则化的线性模型, 其最佳解都可以被样本点线性表示; 而解可以被样本点线性表示, 则可以使用kernel trick, 例如KLR。由于Ridge Regression也是L2正则化线性模型, 因此也可以将其转化为Kernel Ridge Regression。

回忆使用平方误差的回归问题:

对于普通线性回归和岭回归来说, 都有analytic solution(封闭解)。那么, 对于kernel ridge regression来说, 有analytic solution吗?

ridge regression问题:

将 $\mathbf{w}_* = \sum_{n=1}^N \beta_n \mathbf{z}_n$ 代入即可得到kernel ridge regression问题:

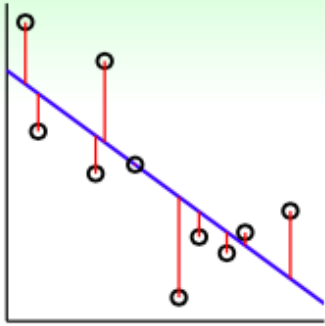
$$\begin{aligned} \min_{\beta} \quad & \underbrace{\frac{\lambda}{N} \sum_{n=1}^N \sum_{m=1}^N \beta_n \beta_m K(\mathbf{x}_n, \mathbf{x}_m)}_{\text{regularization of } \beta \text{ on } K\text{-based regularizer}} + \underbrace{\frac{1}{N} \sum_{n=1}^N \left(y_n - \sum_{m=1}^N \beta_m K(\mathbf{x}_n, \mathbf{x}_m) \right)^2}_{\text{linear regression of } \beta \text{ on } K\text{-based features}} \\ = \quad & \frac{\lambda}{N} \beta^T \mathbf{K} \beta + \frac{1}{N} \left(\beta^T \mathbf{K}^T \mathbf{K} \beta - 2 \beta^T \mathbf{K}^T \mathbf{y} + \mathbf{y}^T \mathbf{y} \right) \end{aligned}$$

上述无约束最优化问题的目标函数是 $\mathbf{\beta}$ 的二次式, 因此可以直接使用导数置零的方法得到analytic solution:

- 对于任何 $\lambda > 0$, 逆一定存在, 因为: \mathbf{K} 是半正定矩阵(Mercer's condition), 对角线加上正数, 一定得到正定矩阵, 因此可逆;
- 时间复杂度: $O(N^3)$; 并且, 该矩阵是dense的, 算逆矩阵更加困难。

最后, 将ridge regression与kernel ridge regression进行对比:

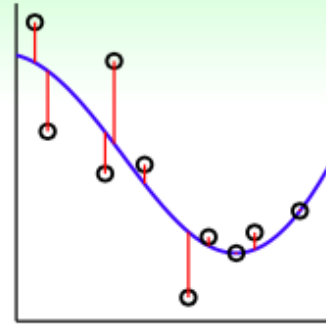
Linear versus Kernel Ridge Regression



linear ridge regression

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

- more restricted
- $O(d^3 + d^2 N)$ training;
 $O(d)$ prediction
—efficient when $N \gg d$



kernel ridge regression

$$\beta = (\lambda \mathbf{I} + \mathbf{K})^{-1} \mathbf{y}$$

- **more flexible** with K
- $O(N^3)$ training;
 $O(N)$ prediction
—hard for big data

linear vs kernel: 实质是efficiency和flexibility之间的trade-off。

附：对于kernel ridge regression，得到最佳的 $\mathbf{\beta}$ 后，回传的hypothesis是：

2. Support Vector Regression Primal: SVR的原始形式

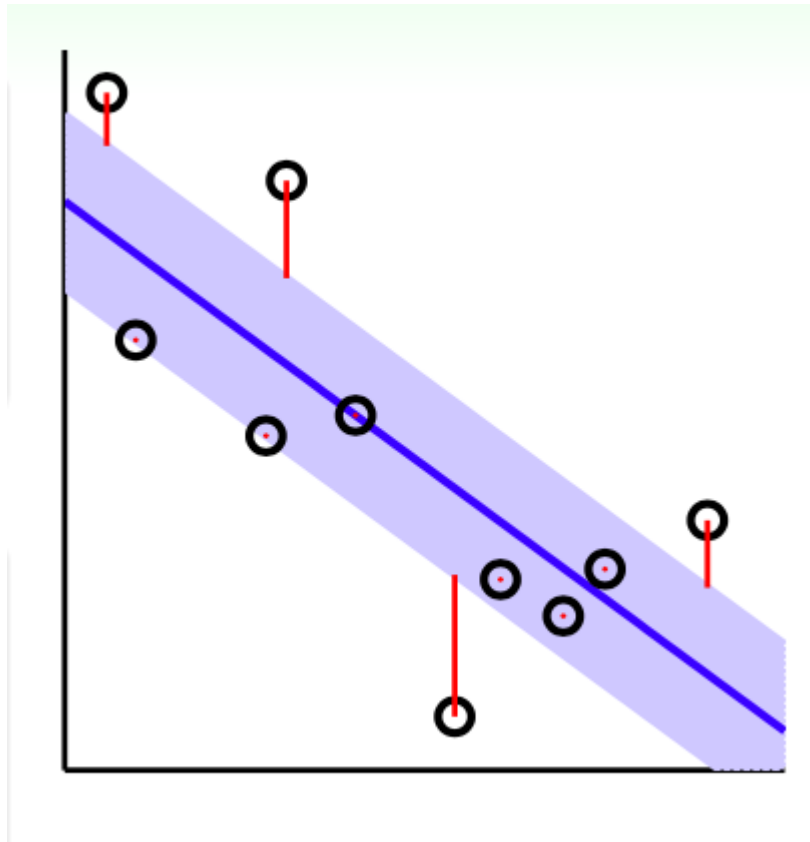
因为平方误差是0-1误差的上界，因此有regression for classification。同样的，也有kernel ridge regression for classification——这样的模型又被称为**least-squares SVM**，最小二乘法SVM，简称**LSSVM**。

Motivation

LSSVM与Soft-Margin SVM的边界形状相差不大，但会有更多的SVs——这是因为LSSVM的 $\mathbf{\beta}$ 是Dense的，而SVM的 $\mathbf{\alpha}$ 是Sparse的——Dense就会导致更慢的prediction。我们希望 $\mathbf{\beta}$ 也能是sparse的。

Tube Regression

在tube内的样本点，error不计；在tube外的样本点，error是到tube边界的距离。如下图红线所示(蓝色区域为tube)：

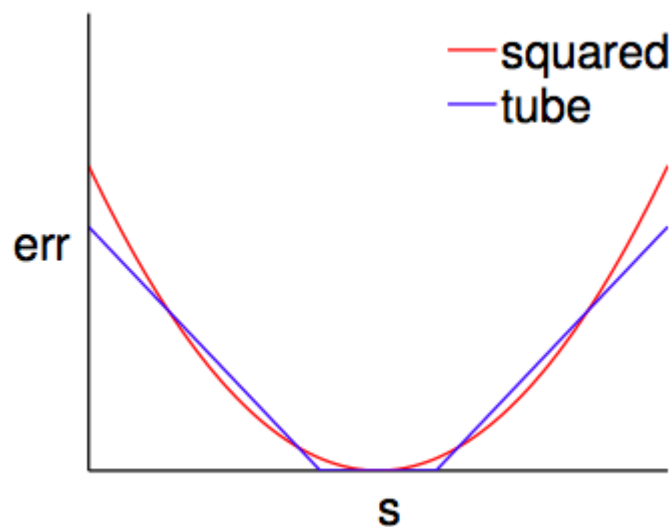


数学化一些，这里的error measure可以写作：

- 如果 $|s-y| \leq \epsilon$ ，则误差记作0；
- 如果 $|s-y| > \epsilon$ ，则误差记作 $|s-y| - \epsilon$ ；

这种误差函数被称为 **epsilon-insensitive error**，其中 $\epsilon > 0$ 。

我们将该误差函数与平方误差函数进行比较：



可见，在 s 与 y 很接近的时候，tube 的误差函数值与平方误差函数值很接近；随着 s 与 y 的偏离的增加，平方误差给予更多的惩罚(二次函数递增)，但 tube 误差函数则线性递增——因此，**tube 误差函数受极端值的影响较小**。

L2-Regularized Tube Regression

加上L2正则化后，Tube Regression的最优化问题如下：

直接解该最优化问题当然可以，但是：

1. max函数是不可微分的——不好解；
2. 可以使用kernel技巧——但得到的解不是sparse的。

因此，我们希望将上面的最优化问题，转换成SVM的形式，这样就可以利用KKT条件保证kernelize的解是sparse的。回忆Soft-Margin SVM primal的无约束条件形式，同这里的最优化目标函数十分类似。因此，我们模仿Soft-Margin SVM primal，将这里的目标函数进行微调：

将 $\max(\dots)$ 记做 ξ_n ，上述无条件最优化问题可以反推为等价的有条件最优化问题(完全模仿Soft-Margin SVM primal)：

因为存在绝对值符号，约束条件还不是线性的——打开绝对值：

上述问题即为**Support Vector Regression (SVR) primal**问题。这是一个QP问题，有 $\tilde{d}+1+2N$ 个变量， $2N+2N$ 个约束条件。

3. Support Vector Regression Dual: SVR的对偶形式

将约束条件1系列的拉格朗日乘子设为 $\alpha_n^{\{\land\}}$ ，将约束条件2系列的拉格朗日乘子设为 $\alpha_n^{\{\lor\}}$ 。

注意，我们不必关注 ξ_n 的拉格朗日乘子，因为根据之前Soft-Margin Dual的推导过程， ξ_n 的拉格朗日乘子能够被 α_n 表示，且最终 ξ_n 可以被消去；需要添加的条件仅仅为：

根据KKT条件，对 w 偏导至零得到：

根据KKT条件，对 b 偏导至零得到：

根据KKT条件中的complementary slackness，有：

SVR Dual的完整形式如下图右下角所示(左侧一列是SVM的primal与dual，右侧上面是SVR的primal)：

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n \\ \text{s.t.} \quad & y_n(\mathbf{w}^T \mathbf{z}_n + b) \geq 1 - \xi_n \\ & \xi_n \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N (\xi_n^{\wedge} + \xi_n^{\vee}) \\ \text{s.t.} \quad & 1(y_n - \mathbf{w}^T \mathbf{z}_n - b) \leq \epsilon + \xi_n^{\wedge} \\ & 1(\mathbf{w}^T \mathbf{z}_n + b - y_n) \leq \epsilon + \xi_n^{\vee} \\ & \xi_n^{\wedge} \geq 0, \xi_n^{\vee} \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N \alpha_n \alpha_m y_n y_m K(\mathbf{x}_n, \mathbf{x}_m) \\ & - \sum_{n=1}^N 1 \cdot \alpha_n \\ \text{s.t.} \quad & \sum_{n=1}^N y_n \alpha_n = 0 \\ & 0 \leq \alpha_n \leq C \end{aligned}$$

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N (\alpha_n^{\wedge} - \alpha_n^{\vee})(\alpha_m^{\wedge} - \alpha_m^{\vee}) k_{n,m} \\ & + \sum_{n=1}^N ((\epsilon - y_n) \cdot \alpha_n^{\wedge} + (\epsilon + y_n) \cdot \alpha_n^{\vee}) \\ \text{s.t.} \quad & \sum_{n=1}^N 1 \cdot (\alpha_n^{\wedge} - \alpha_n^{\vee}) = 0 \\ & 0 \leq \alpha_n^{\wedge} \leq C, 0 \leq \alpha_n^{\vee} \leq C \end{aligned}$$

最后，讨论SVR解的Sparsity。

我们知道：

对于在tube内部的样本点，即：

因为没有任何的违反，所以：

因此：

根据complementary slackness，有：

因此：

即，在tube内部的样本点，对于 \mathbf{w} 没有一点贡献。因此，SVs，即 $\beta_n \neq 0$ 的样本点，应该在边界上或在tube外面。

4. Summary of Kernel Models: 核模型的总结

PLA/pocket	linear SVR	
linear soft-margin SVM	linear ridge regression	regularized logistic regression
	kernel ridge regression	kernel logistic regression
SVM	SVR	probabilistic SVM

前两行是线性模型：

PLA/pocket minimize $\text{err}_{0/1}$ specially	linear SVR minimize regularized err_{TUBE} by QP	
linear soft-margin SVM minimize regularized $\widehat{\text{err}}_{\text{SVM}}$ by QP	linear ridge regression minimize regularized err_{SQR} analytically	regularized logistic regression minimize regularized err_{CE} by GD/SGD

- 第一行很少用，因为worse performance；
- 第二行的模型被集成在liblinear中。

后两行是kernel模型，即非线性模型：

	kernel ridge regression kernelized linear ridge regression	kernel logistic regression kernelized regularized logistic regression
SVM minimize SVM dual by QP	SVR minimize SVR dual by QP	probabilistic SVM run SVM-transformed logistic regression

- 第三行很少用，因为dense解；

- 第四行的模型被集成在**libsvm**中。