Research on Support Vector Regression

目录

1 数据集性质

1.1 统计量

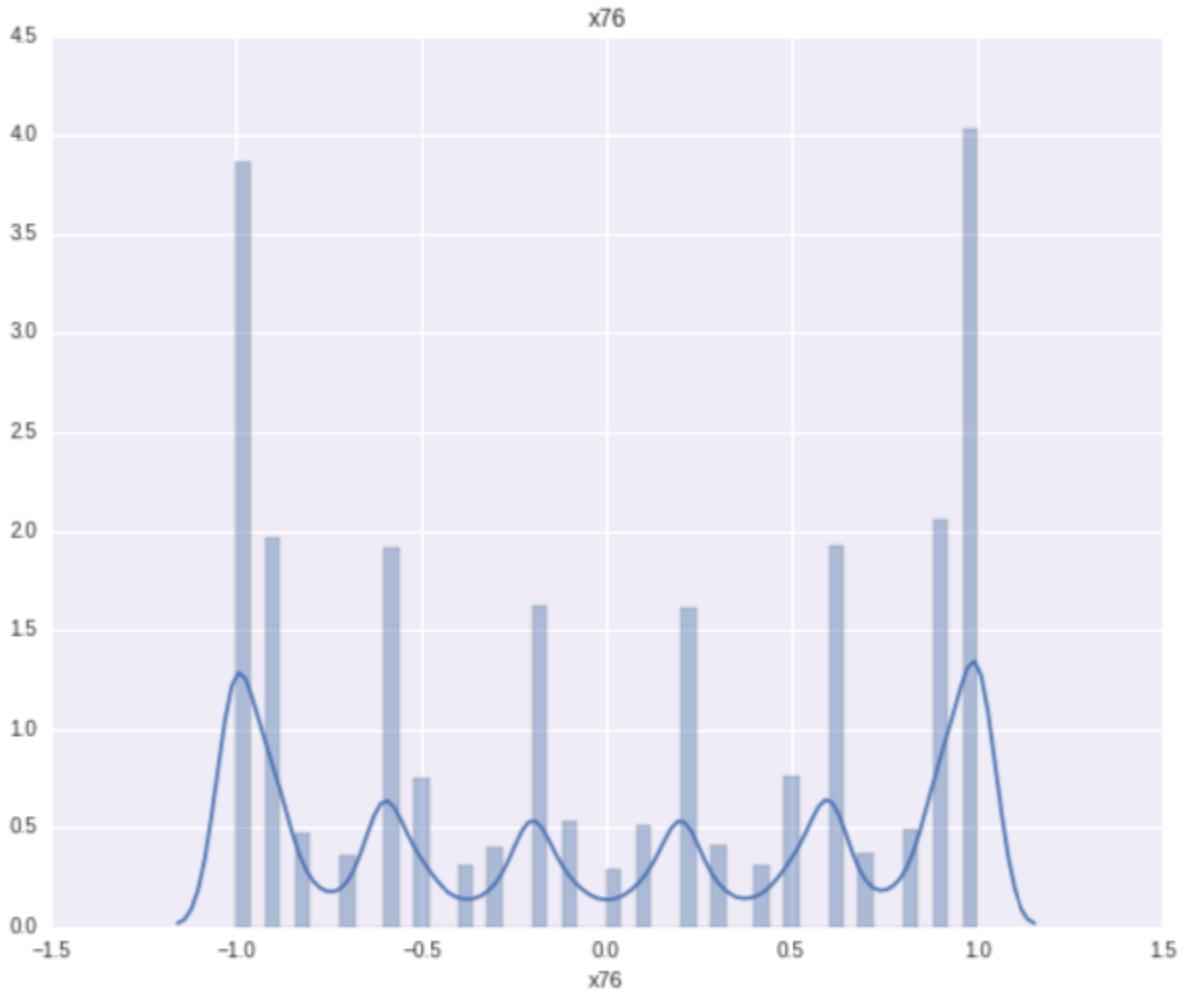
范围：除了 6 个 X，都在[-3，3]之间，范围最大的也在[-8，8]之间。

均值：从绝对值上看，abs(min)都小于 0.015；从相对大小来看，abs(min)/max 都小于 0.02

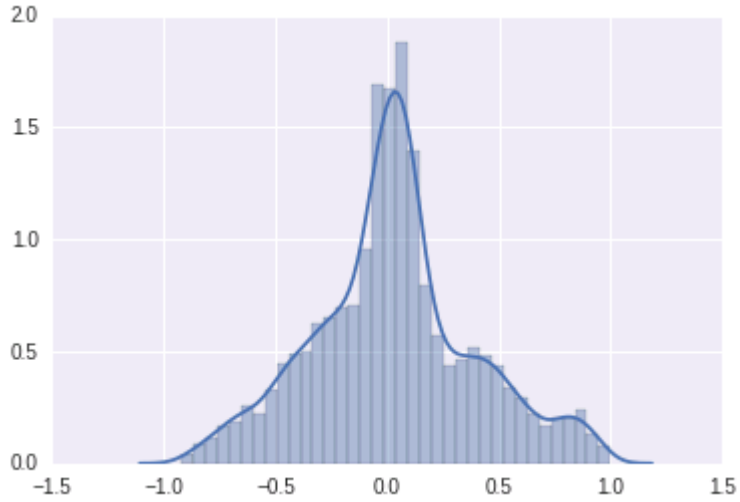
1.2 分布

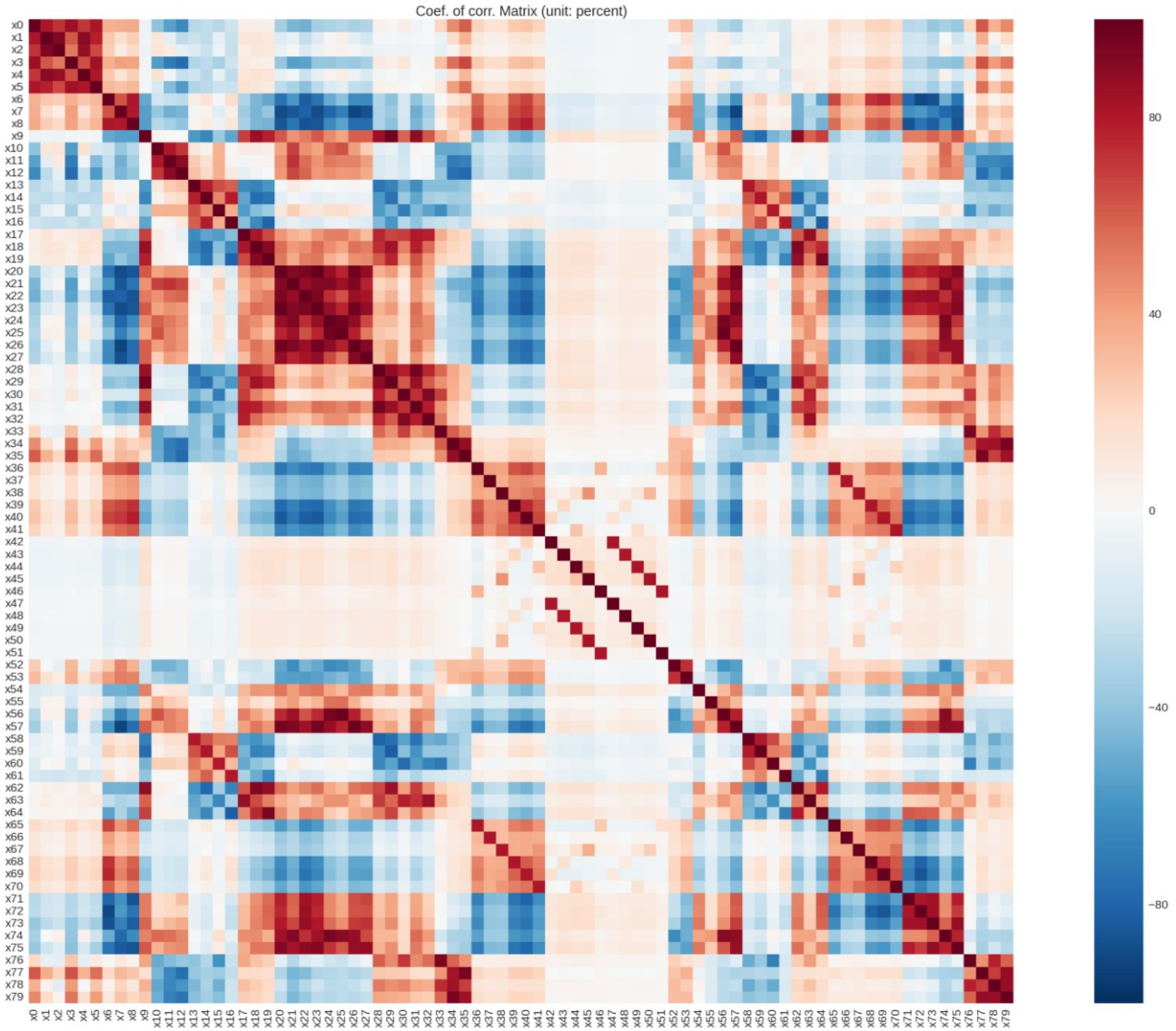
连续性：有连续的，也有离散的

分布：大部分为类正态分布，少部分怪异

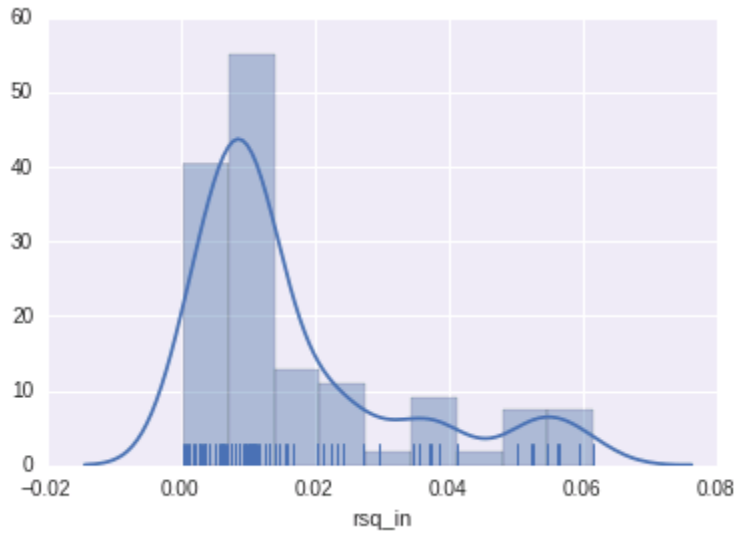


1.3 相关系数


2.0
1.5
1.0
0.5
0.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Coef. of corr. Matrix (unit: percent)
80
40
0
−40
−80

1.4 解释能力



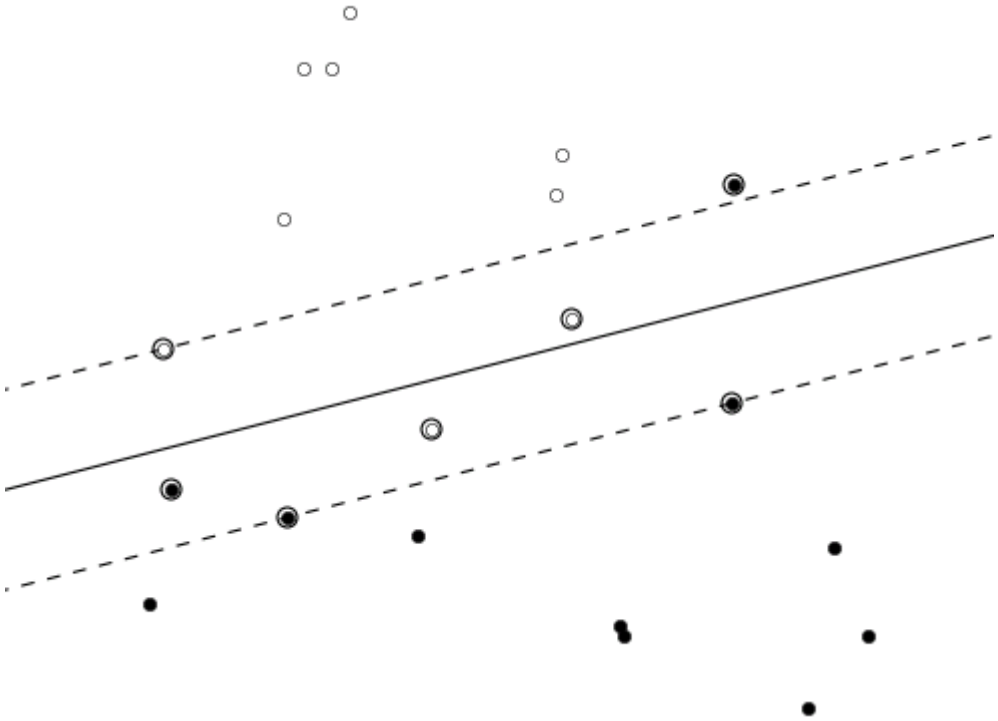
大部分在 3%以下，最大有 6%

2 支持向量机

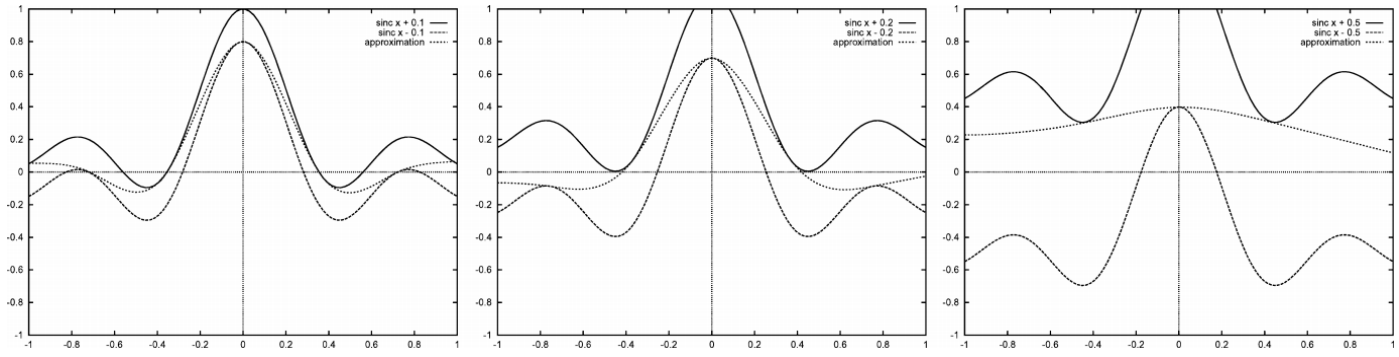
2.1 几个核心

2.1.1 Only Support Vectors are needed

Classification



Regression



Where	ξ_i	α_i
Inside the tube	= 0	= 0
On the support plane	= 0	> 0
Outside the tube	> 0	> 0

2.1.2 Kernel

Kernel -- 衡量相似度

Similarity = K(**X, Y**)

Kernel trick -- 解决计算效率问题

直接在原 space 计算 feature space 的内积

2.2 背后的数学

2.2.1 整体思路框架

Optimization problem (Objective func. + Constrains)

-->

Standard form (Minimize + Leq) (非内积形式)

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p. \end{aligned}$$

-->

Lagrangian

-->

Min(x)Max(theta) (不好解)

-->

Dual problem: Max(theta)Min(x)

-->

Solve

2.2.2 Duality

Duality gap

= Max(theta)Min(x) - Min(x)Max(theta)

Max(theta)Min(x)是Min(x)Max(theta)的一个下界

Strong / Weak duality:

如果gap = 0则称为strong duality. SVM由于其primal problem就是convex的，可推出strong duality.

2.2.3 From Loss Function to Slack Variables

Hinge loss:

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\max\left(0, |\mathbf{w}^T\mathbf{z}_n + b - y_n| - \epsilon\right)$$

把max转化为slack variable + constrain:

$$\begin{aligned}\min_{b,\mathbf{w},\xi}\ & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n \\ s.t.\ & |\mathbf{w}^T\mathbf{z}_n + b - y_n| \le \epsilon + \xi_n \\ & \xi_n \ge 0\end{aligned}$$

通过"分情况讨论"把不可微的绝对值运算变为普通的线性运算：

$$\begin{aligned}&\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\left(\xi_n^{\vee} + \xi_n^{\wedge}\right) \\ &-\epsilon - \xi_n^{\vee} \le y_n - \mathbf{w}^T\mathbf{z}_n - b \le \epsilon + \xi_n^{\wedge} \\ &\xi_n^{\vee} \ge 0, \xi_n^{\wedge} \ge 0\end{aligned}$$

2.2.4 Sklearn Implemention

Primal

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$
$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i,$$
$$\zeta_i \geq 0, i = 1, ..., n$$

Dual

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$
$$\text{subject to } y^T \alpha = 0$$
$$0 \leq \alpha_i \leq C, i = 1, ..., n$$

2.3 计算复杂度

2.3.1 时间复杂度

大致正比于 max{n**3, n**2*m}

对于 sklearn 的实现，主要时间耗费在 Fit predict 上，测试正比 O(N^2)

单独 predict 则正比与 x 略小于 1 次(10k 数据平均每个 predict 0.6ms)

2.3.2 空间复杂度

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$
$$\text{subject to } y^T \alpha = 0$$
$$0 \leq \alpha_i \leq C, i = 1, ..., n$$

N * N 的 kernel matrix

3 数据集划分与预处理

3.1 数据集划分

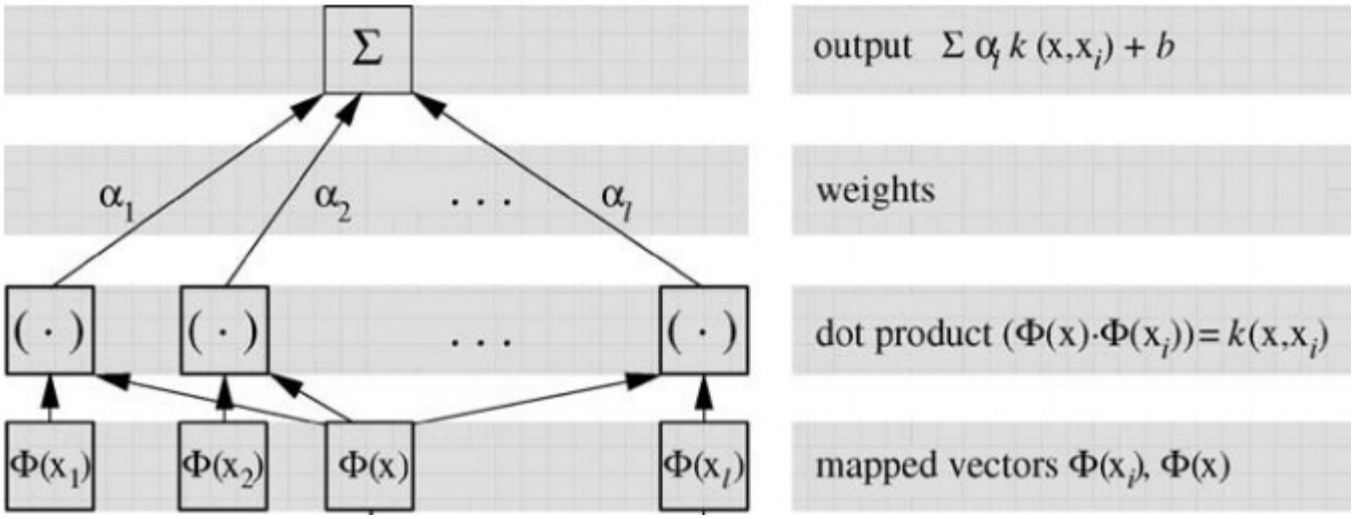
全长 870k，in: validation: test = 6: 2: 2

3.2 Standardization

对于 SVM，计算 dot product 或 similarity 的时候，如果不同 feature 的 scale 不同，则大 scale 的会 dominate 结果，所以要 re-scale.

常用 normalize 或者 standardize， 由于数据分布比较厚尾，所以采用 standardize，测试也的确表现更好。

4 调参的指导



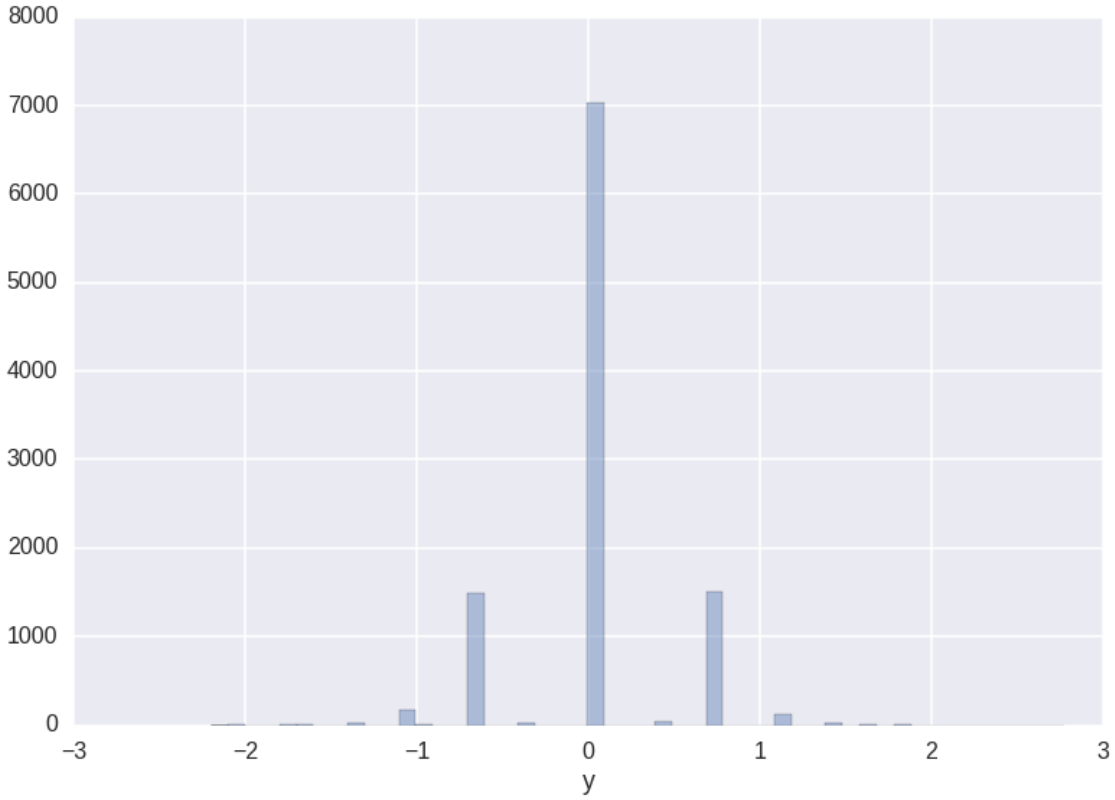
4.1 Epsilon

效果是，最终 fit 出的超平面的正负 ε 范围内的数据点都会被忽略．因而可以用来消去小噪音的影响．

该参数只由 Y 决定，估计方法是根据 Y 的信、噪幅度．一般的估计方法是，首先根据先验知识和统计学方法估计噪音(方差)，接着根据噪音大小和样本数量等估计ε的合适取值．

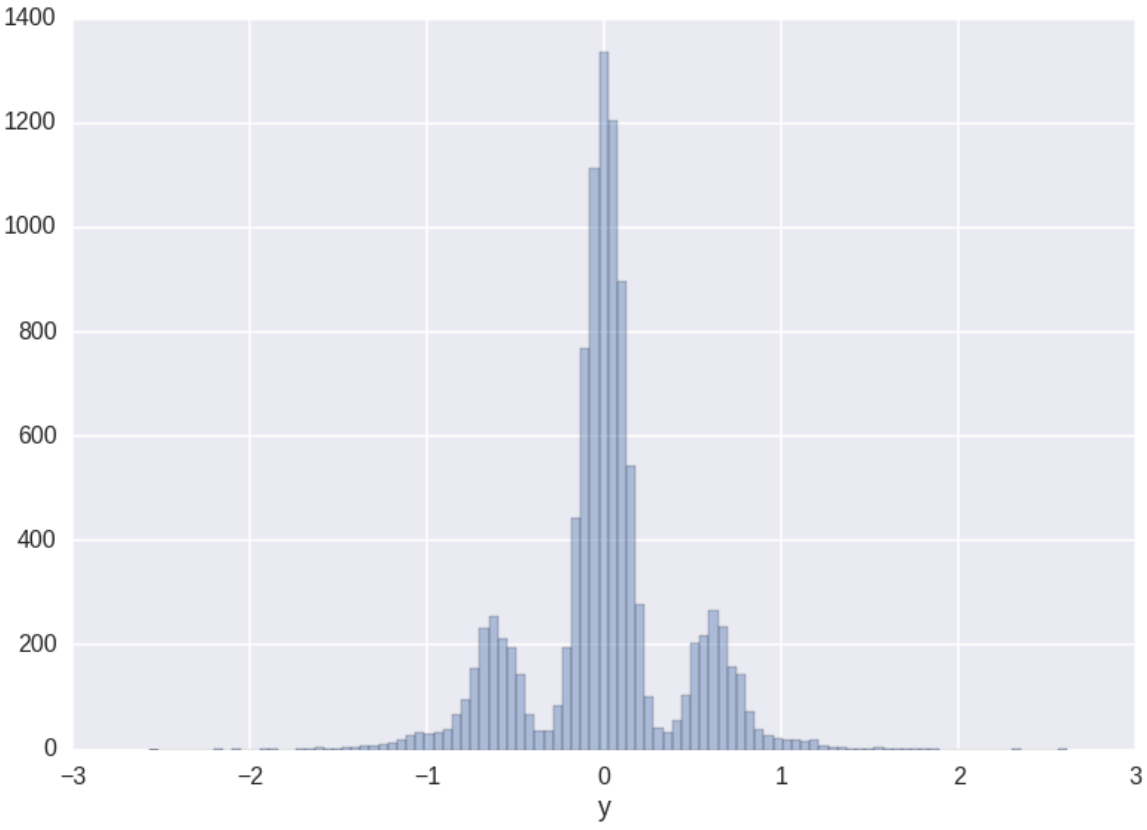
如果没有先验知识，或者无法对噪音的分布做出一些可靠的假设，则需要根据信号特点估计ε：

例 1



信号离散,最小信号是 0.38 左右,可选择ε=0.2

例 2



这是线性回归 residual 的分布，难以估计，于是可以在 0-0.5 区间 grid search.

4.2 C

是 insample 拟合强度，也是各个 S.V.的系数值上限.

有三个角度来看它:

1. primal perspective

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$
$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i,$$
$$\zeta_i \geq 0, i = 1, ..., n$$

Bias, variance Trade-off. 因而小了导致欠拟合，大了则过拟合.

2. decision func. perspective

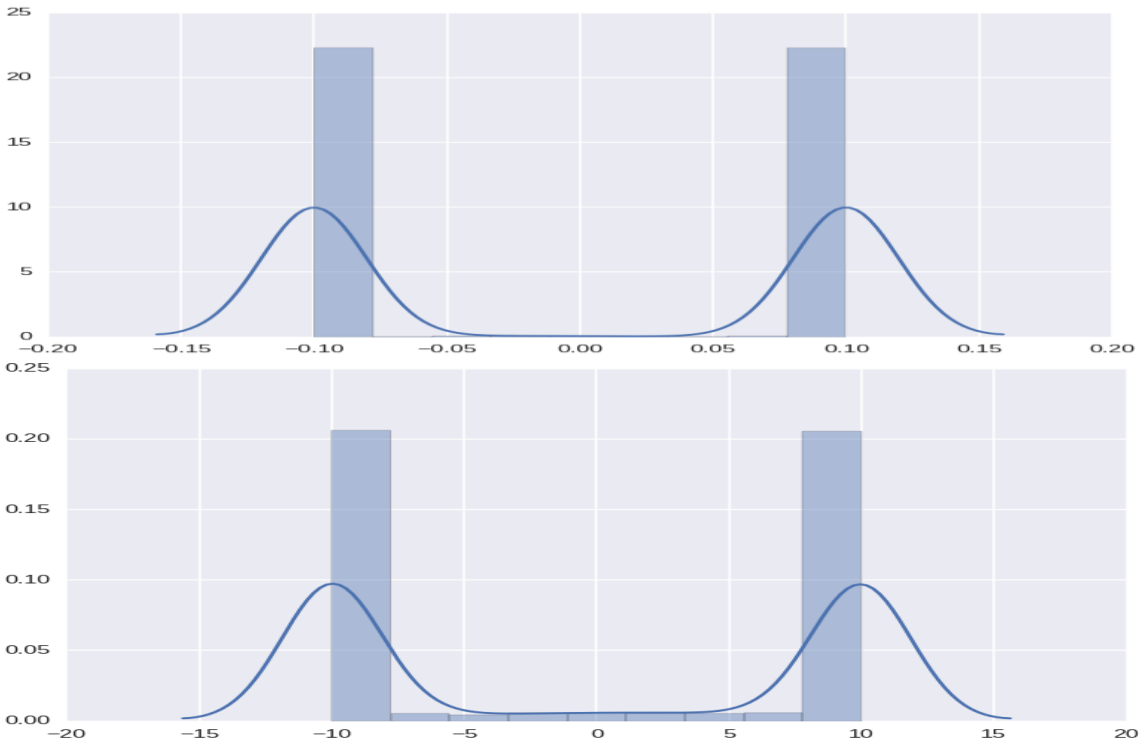
$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$
$$\text{subject to } y^T \alpha = 0$$
$$0 \leq \alpha_i \leq C, i = 1, ..., n$$

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) k(x_i, x) + b$$

每个 support vector 的系数值的上限.

综上，一个合理的取值是 max(abs(Y))，更 robust 的取值是(mean + 3σ). 这样取值使得每个 support vector 都有足够的"话语权"(变化范围足够大以涵盖 Y 的大部分取值).

但对于我们所用的这个数据集，该方法并不合适，因为 fit 得太差(Rsquare 很小). 例如上下图分别是 C = 0.1 和 C = 10.的回归结果中，support vectors 系数的分布:



两次回归结果的 support vector 比例都接近 50%. 也即一半多的数据点都不在ε-tube 内，且很多数据点离得相当远. 这导致即使 C 设置得比较大，大部分α_i 还是会达到上限，进而导致过拟合.

以上给我们提供了一个对 C 进行 grid search 的上下界. 根据 Y 的范围和 kernel 的输出范围，可以算出 C 的一个下界(不太小以至于丧失话语权)；根据 Y 的数量级可以确定 C 的上界(达到这个数量级就足够了). 而 grid search 时的标准则是欠拟合与过拟合的一个平衡.

4.3 Gamma

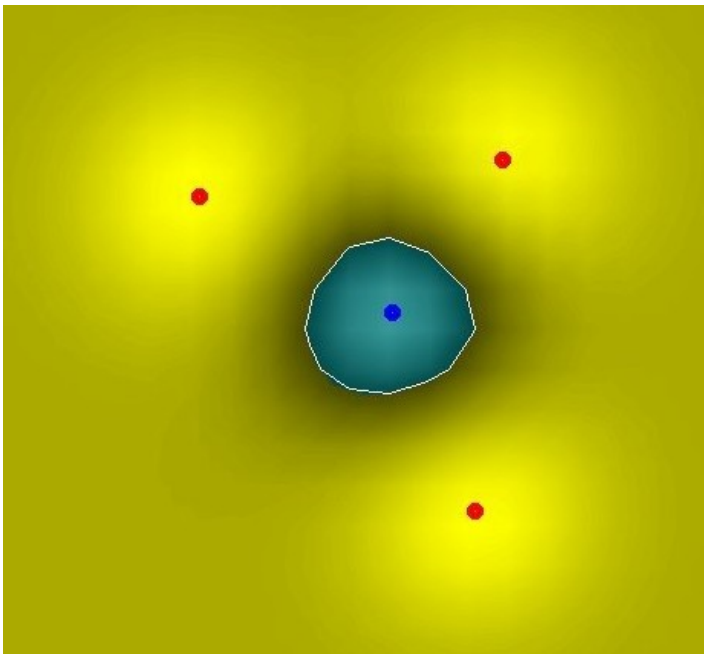
gamma 指 RBF kernel 中的参数

$$k(x,y) = e^{-\gamma||x-y||_2^2}$$

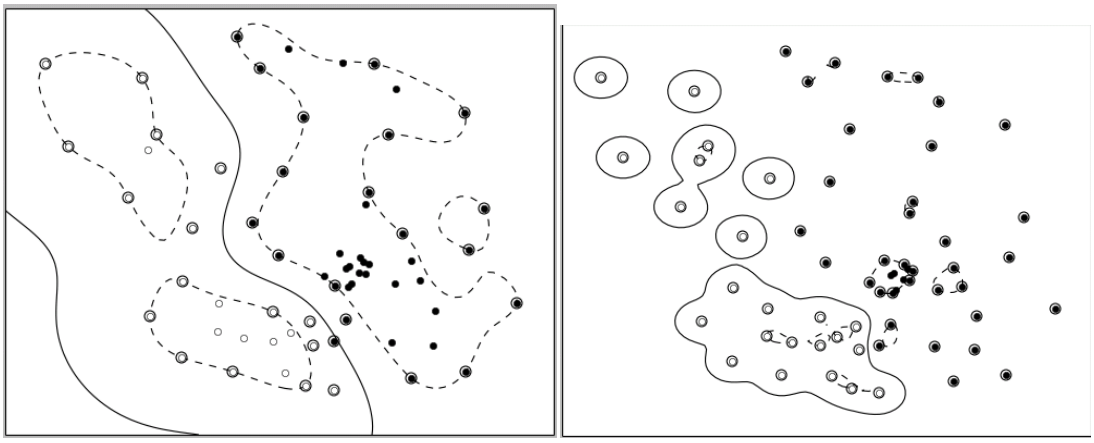
Gamma 定量表示相似度衰减的快慢. 与半衰期有关系

$$T_{1/2} = \frac{\ln 2}{\gamma}$$

以分类为例，二维情况下，下图是 4 个数据点所决定的相似度:



对比，左边 gamma 较小，半衰期较长.



可以看到，左边是比较好的分类，而右边过拟合了.

综上，一个比较好的 gamma 取值，应使得半衰期的大小约等于数据点间距离的中位数(用中位数是因为它比均值稳定)．需要注意的是，这里的距离指与 gamma 相乘的那个距离：

$$e^{-\gamma * distance}$$

确定了数量级后，可以用 grid search 更精细地 tune.

5 Kernel 的研究

5.1 线性与非线性

多元线性回归：对每个 X，确定一个 Y 随之变化的系数(变化率恒定)

非线性的核方法：基于距离，对于不同的距离，可以给出不同的变化率.

5.2 分两步选择 kernel

比如 RBF kernel 可分解为 L2 距离$||x-y||_2$，和$f(x)=e^{-\gamma x^2}$

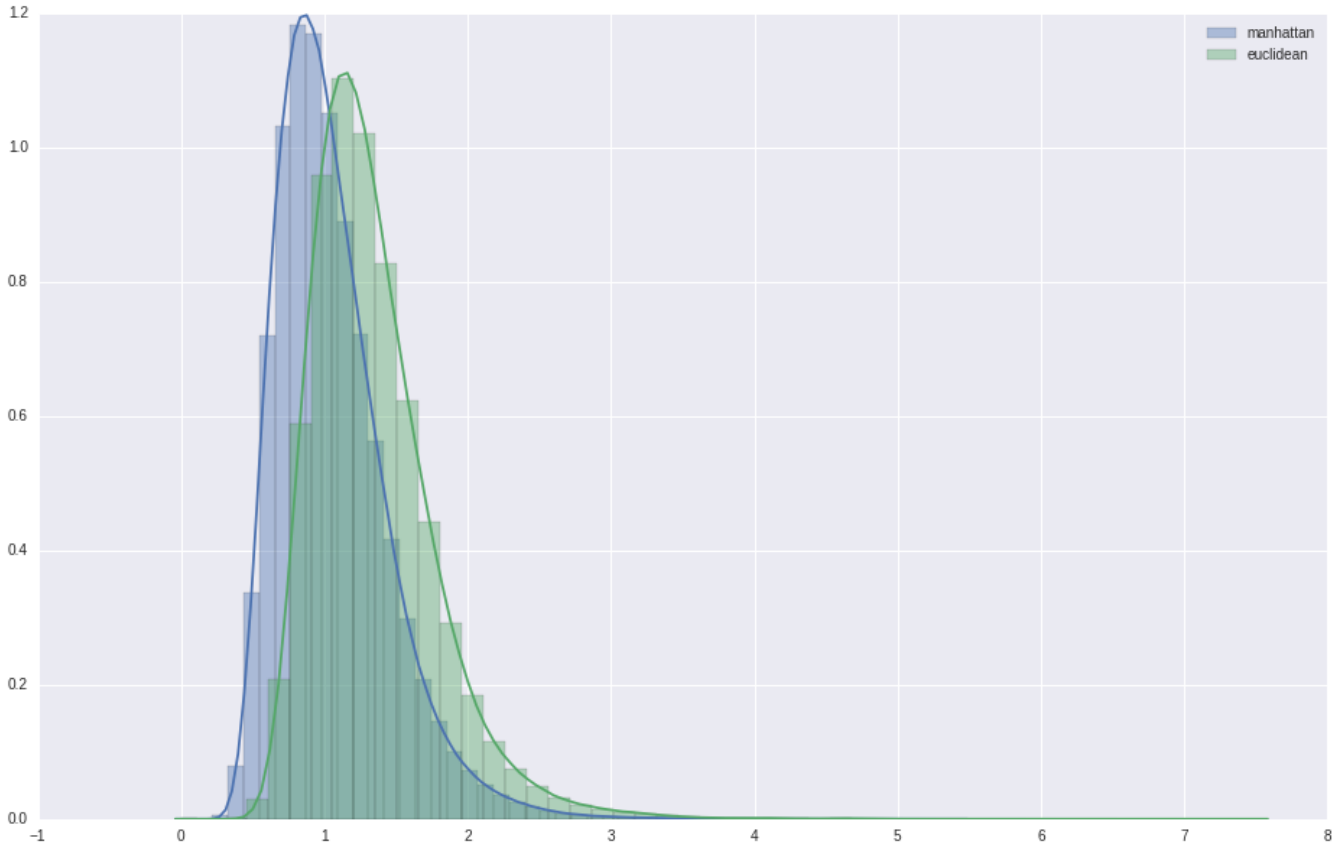
5.2.1 选择 Metric

可选 L1，L2 距离(一般的 Lp 距离计算过慢)

下图为 normalize 过的样本点间 L1，L2 距离的分布：

$$\|x-y\|_2=\sqrt{\frac{\sum(x_i-y_i)^2}{n}}$$

$$\|x-y\|_1=\frac{\sum|x_i-y_i|}{n}$$



一般认为由于计算时所用的平方而导致 L2 距离对 outlier 不够 robust，但是这只会导致 L2 距离的分布相对 L1 更加厚尾一些，由于最外层的根号，距离增速还是近似 1 次的.

测试发现对于同样的函数，使用 L2 距离效果会比 L1 有显著提高(5.05% vs 5.50%)，猜想可能是由于 L1 距离不是光滑的，进而在高维情况下不合适.

还可以考虑加入权重.

直接计算 Lp 距离，各个分量的差是直接相加的，不能体现出不同 feature 间的差异和关系.

测试了相关系数、距离相关系数、互信息系数等表征单个 X 与 Y 之间关系的 rank 方法，都无法使 performance 有明显的提高，若使用 ridge

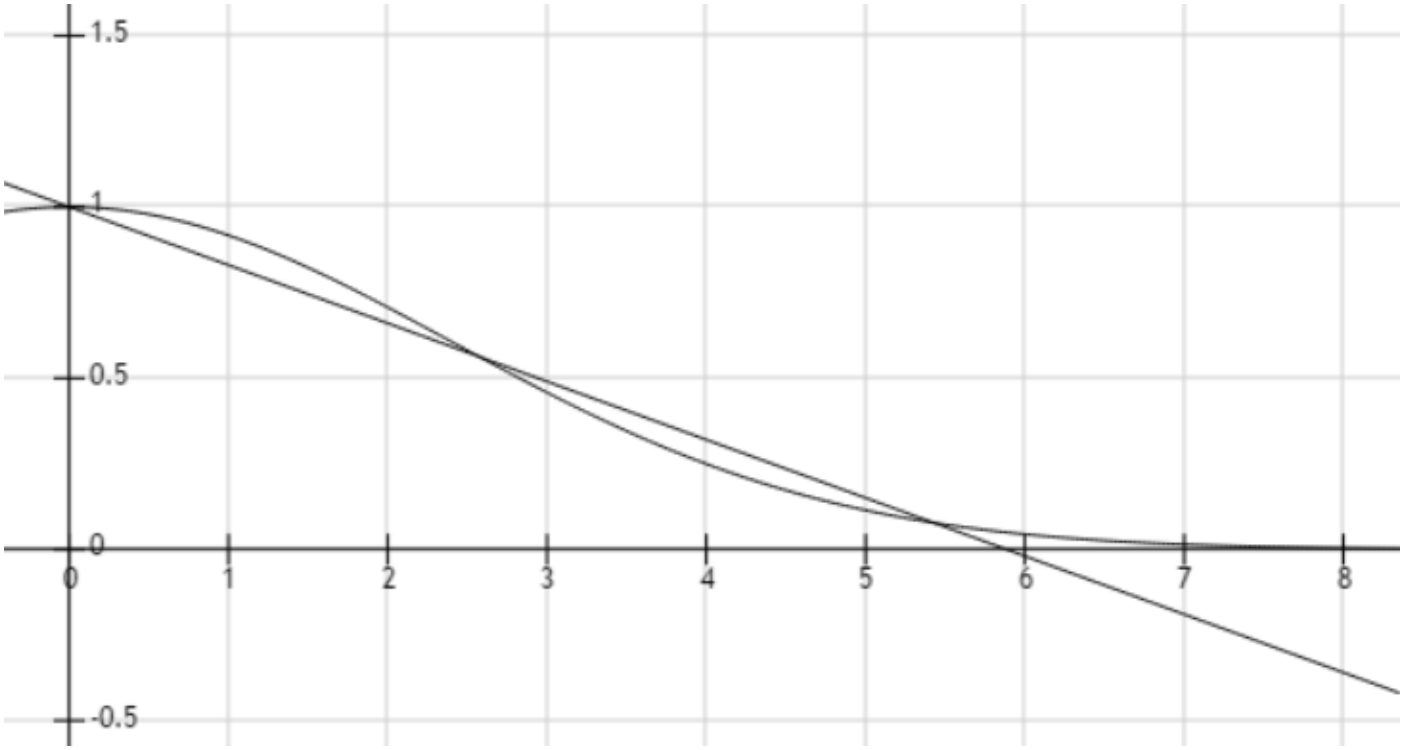
考虑可能是由于这些 rank 方法只能表征单个 X 的效果，无法体现 X 间的关系，

5.2.2 选择函数

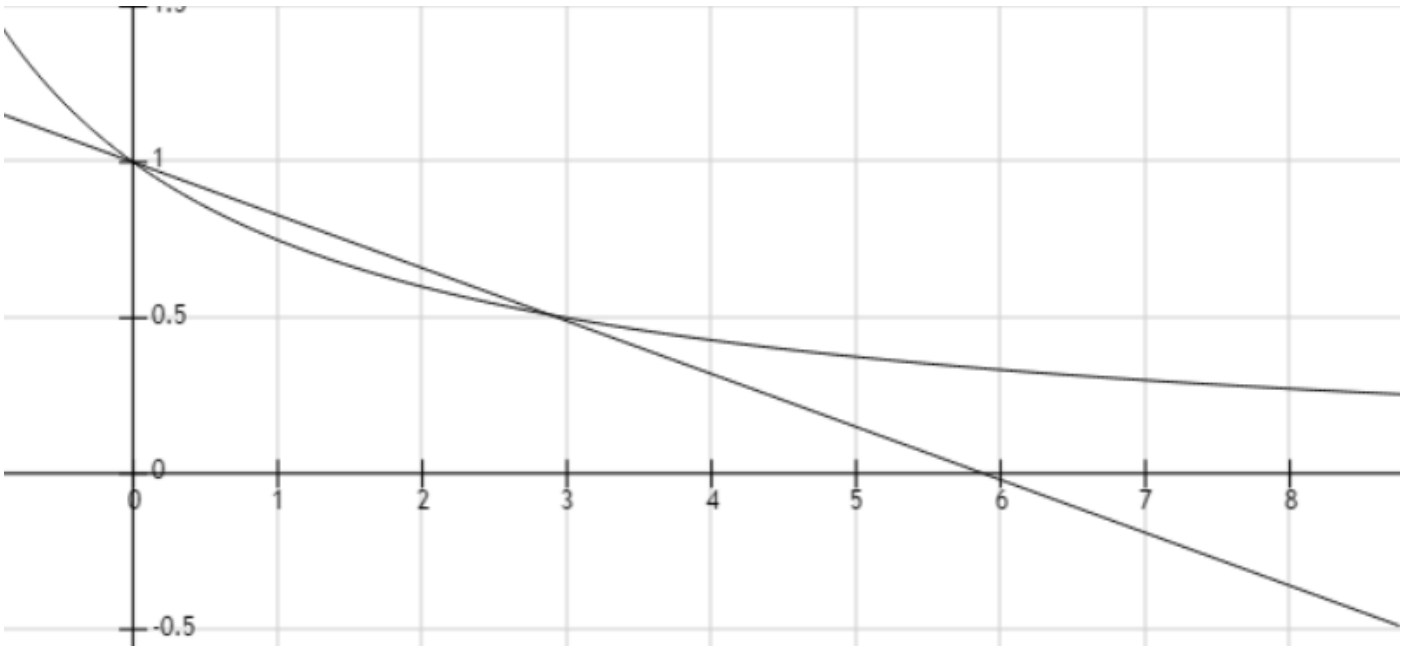
函数的选择，体现了对于上一步算出的距离的不同值，输出的相似度如何变化. 即相似度对距离的一阶和二阶导数的符号. 测试了一下几种经典 kernel 中的函数:

RBF	Laplacian	Quadratic	Spherical
Exp(-x^2)	Exp(-x)	c/(x + c)	1 - 3x/2 + x^3/2

下面三张图分别是 RBF, Quadratic, Shperical 与一次函数的对比(已 normalize)



exp(-gamma * x^2)



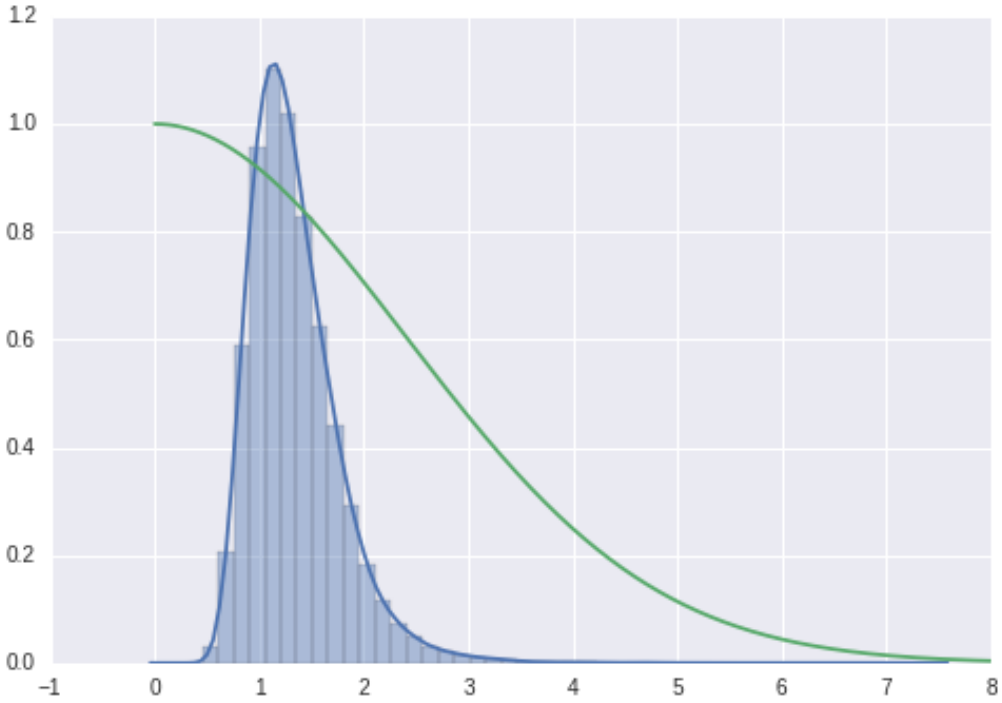
C / (c + x)

1 - 3x/2 + x^3/2

使用相同的距离，测试这些不同的函数，发现：

- Exp(-x^2)比其他明显好(5.7% vs 5.1%)
- 其他三个的函数形态很相似，performance 也都相似
- 有支集的函数(spherical)performance 不如没有的(较大的 Y 的影响被抹去了)
- 所有这四个函数 performance 都比线性(y = 1 – kx)的好.

分析 exp(-x^2)函数



蓝色是 L2 距离的 distribution，绿线是 exp(-gamma * x^2)的函数图像．四个函数与线性的区别在于有 bound，exp(-x^2)与其他三个的区别在于，在大部分范围内二阶导数为负(变化率越来越大)．

因而一个使用 distance 的 kernel，应满足以上两点要求．

6 SVR 结果

数据量	Ridge best	SVR
10k	5.83	5.71
26k	5.79	5.92
52k	6.03	6.10

7 Ensemble learning

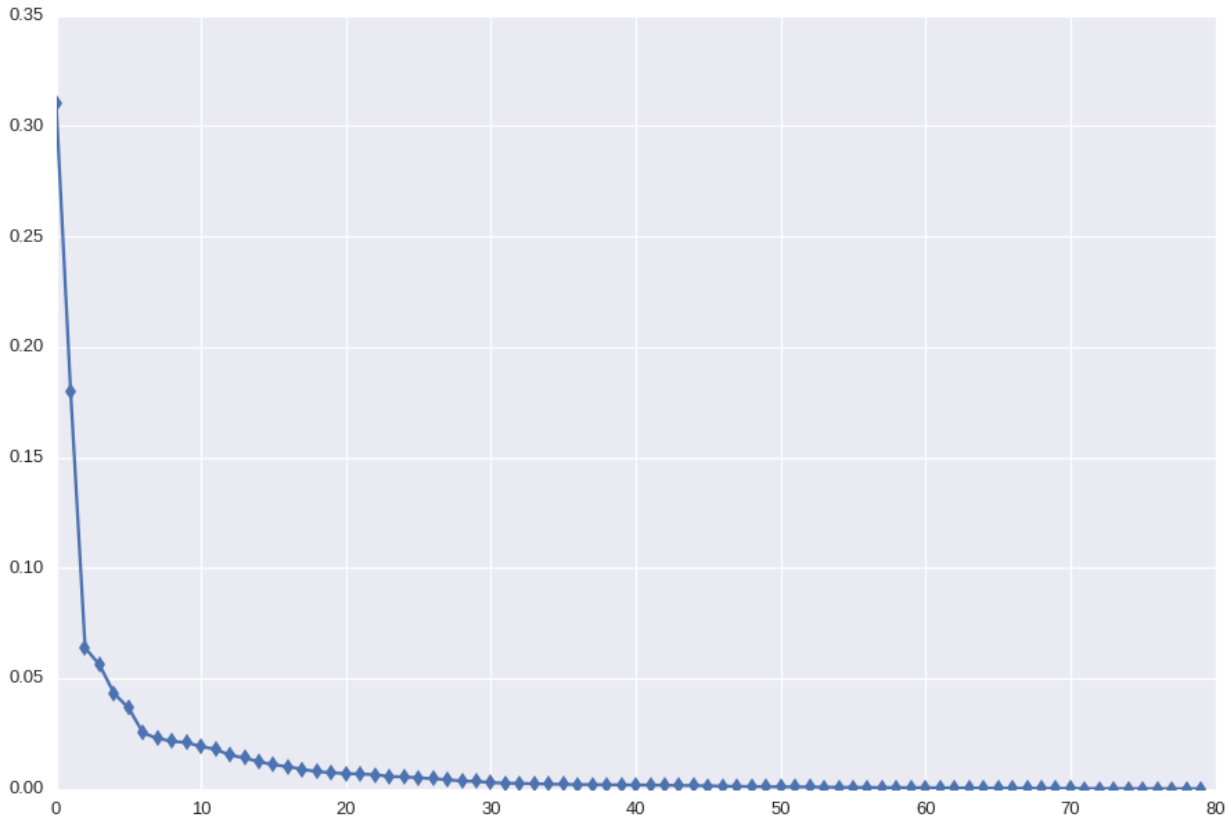
7.1 Boosting

为什么要 boosting

之前测试各种参数，各种 kernel 的时候，可能不同的 kernel 能捕捉到数据的不同侧面，但是终究只有 RBF kernel 回归效果最好，那么直接使用 SVR 的选择就只有 RBF．

如果使用 boosting：先线性回归，得到 residual 之后，再对 residual 尝试各种 SVR 的方法，就可以不用考虑已经回归好的部分.

对 residual 回归时，使用 **PCA**. 一是减小输入数据的噪音，二是防止 curse of dimension



PCA explained variance ratio

测试了使用 PCA 的前 5，6，7，10，20，40，80 维，发现在 6 左右 performance 最好，这也验证了之前的想法(太多引入噪音).

对 residual 回归的结果(单位%):

Kernel	Insample	Validation	Boosting
Ridge best			5.82
RBF	0.102	0.083	5.89
Sperical	0.038	0.029	5.87
linear	0.033	0.110	5.92

7.2 Bagging

单独测试 bagging

RBF	0.102	0.083	5.89
RBF 50 bagging	0.111	0.097	5.91

与 boosting 结合

大多数情况下，不能 beat 最优 performance，即使相关系数均小于 70%