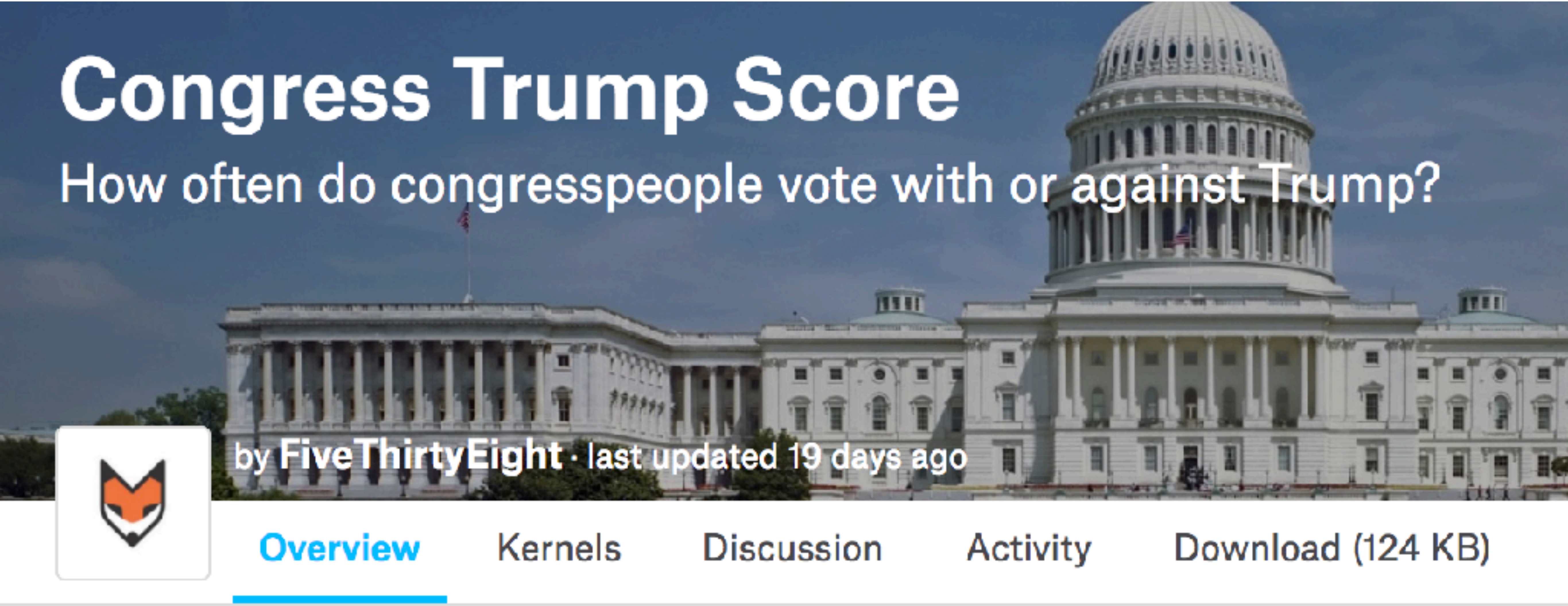
SVR

Support Vector Regression

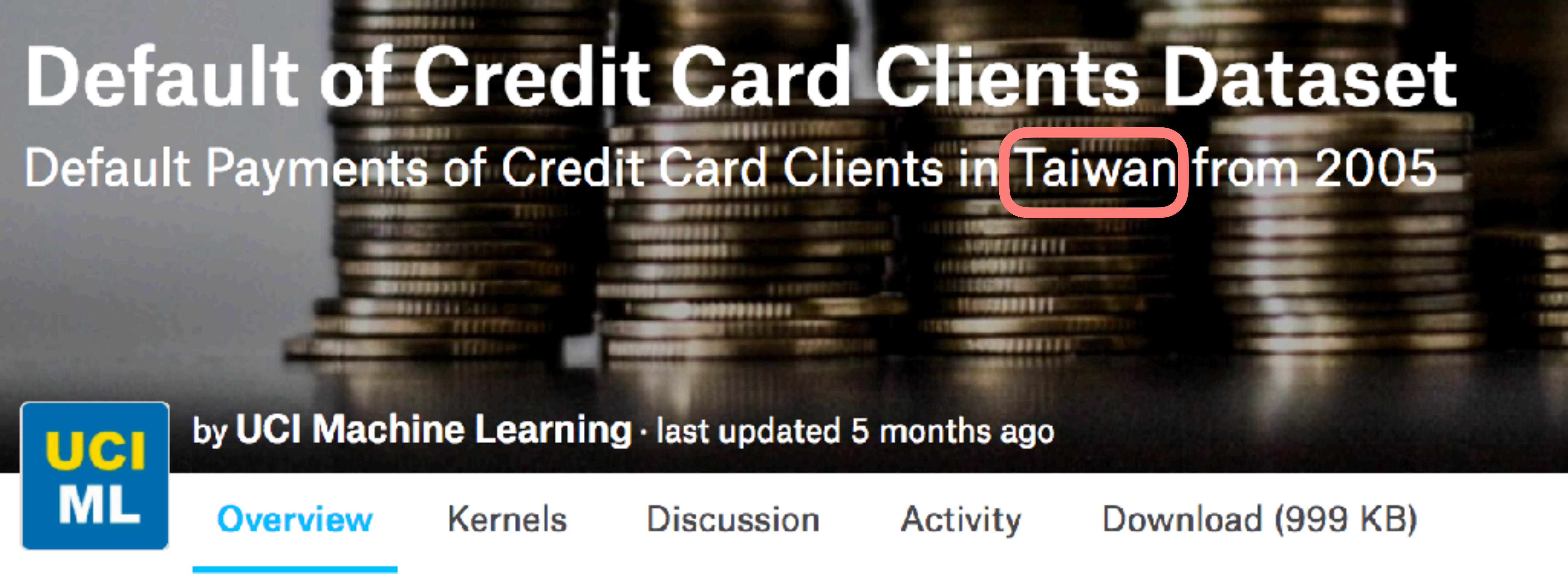
Part 1

Some Interesting Dataset in kaggle



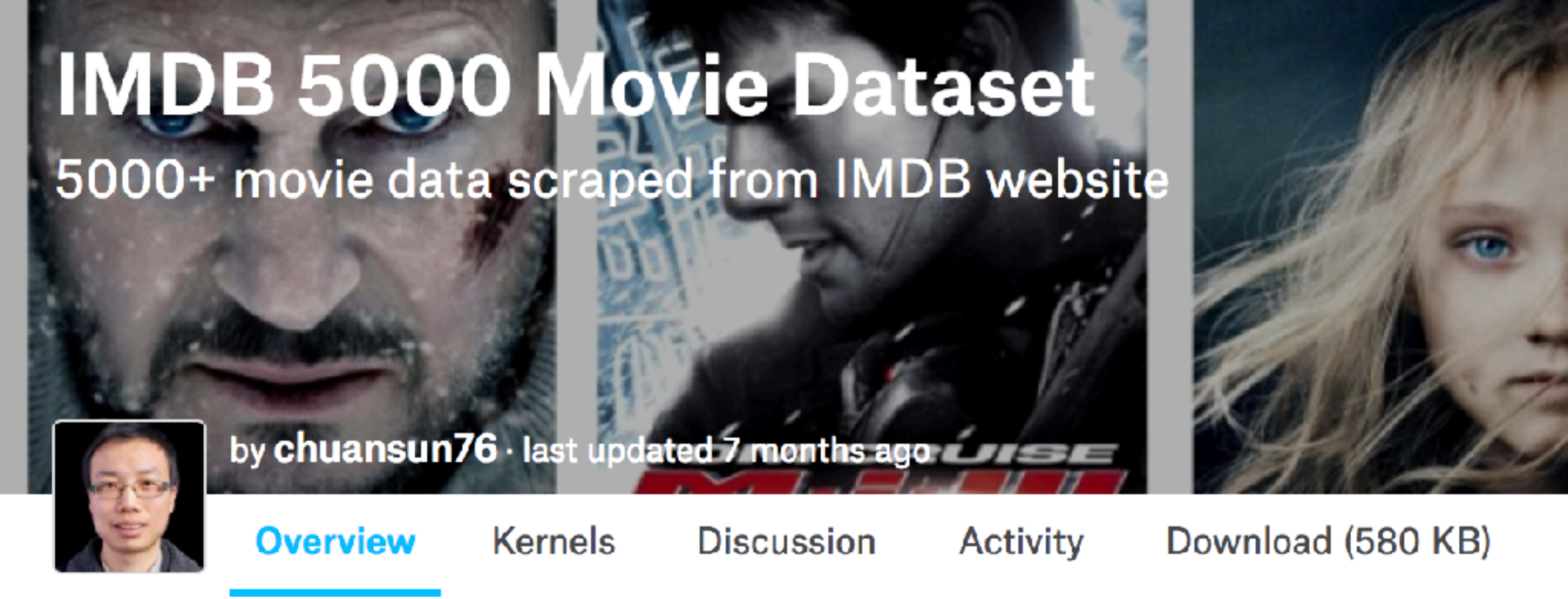
Joe Donnelly	D	IN	50.0%
Joe Manchin III	D	WV	57.6%
Henry Cuellar	D	TX-28	72.0%
Collin C. Peterson	D	MN-7	65.2%

抓到了！
黨鞭如何事前運作？
House of Cards


Default of Credit Card Clients Dataset
Default Payments of Credit Card Clients in Taiwan from 2005
UCI
ML
by UCI Machine Learning · last updated 5 months ago
Overview
Kernels
Discussion
Activity
Download (999 KB)

台灣

銀行策略

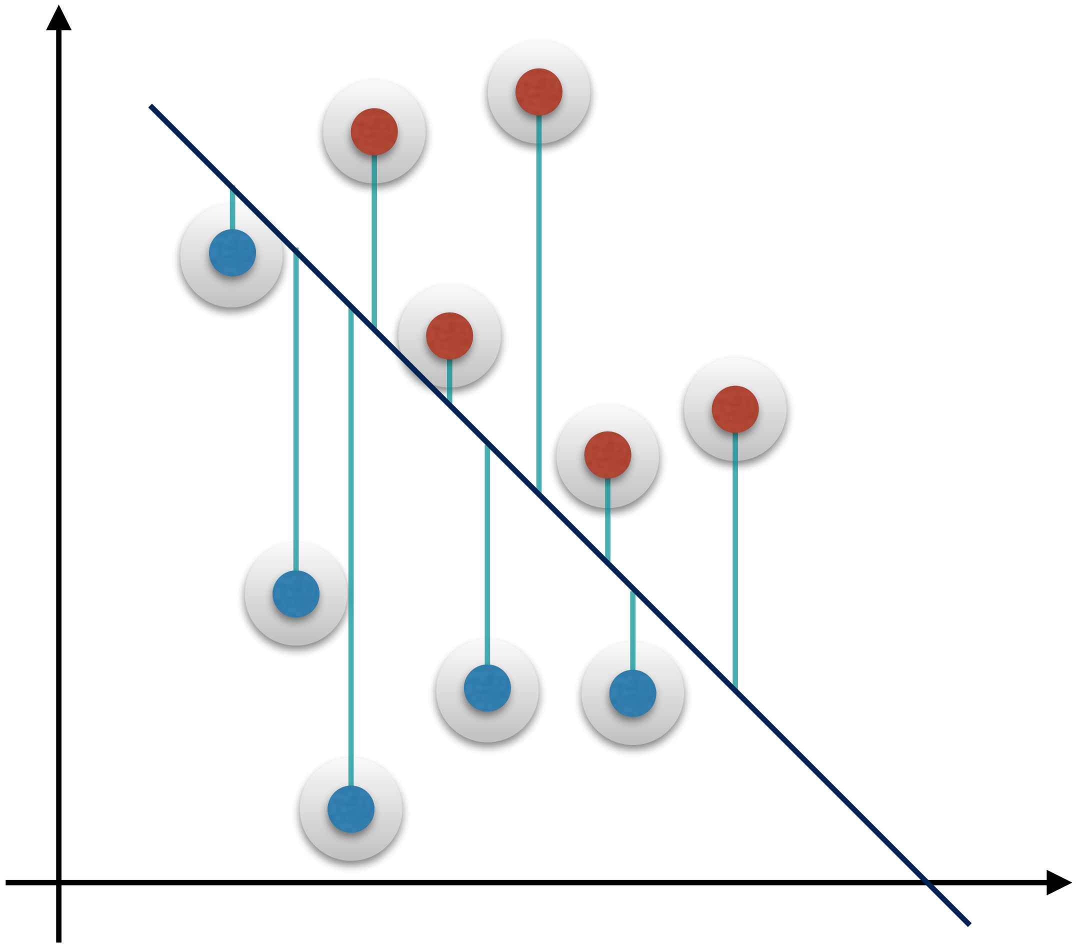

IMDB 5000 Movie Dataset
5000+ movie data scraped from IMDB website
by chuansun76 · last updated 7 months ago
Overview
Kernels
Discussion
Activity
Download (580 KB)

預測電影分數

Part 2

What is SVR

Regression

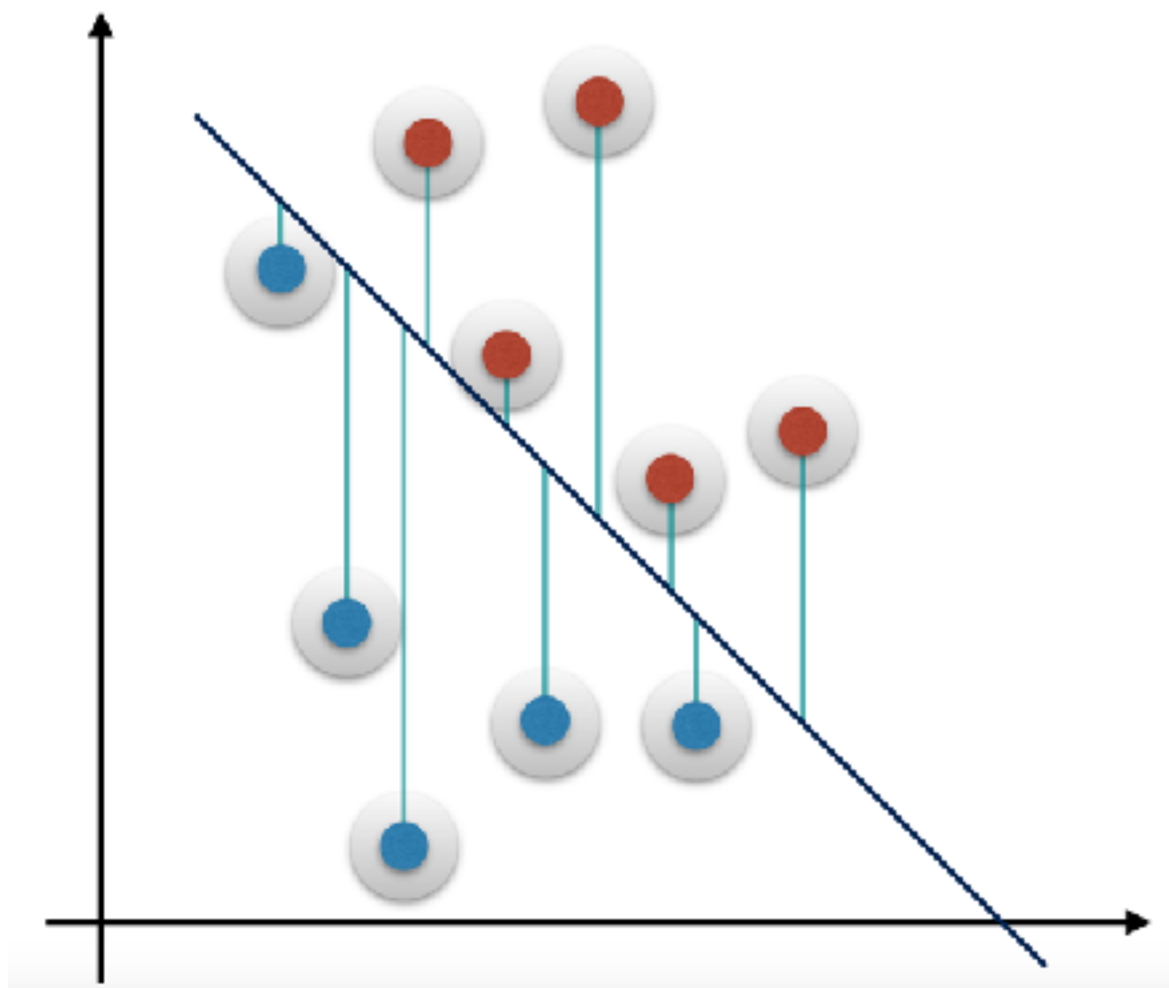


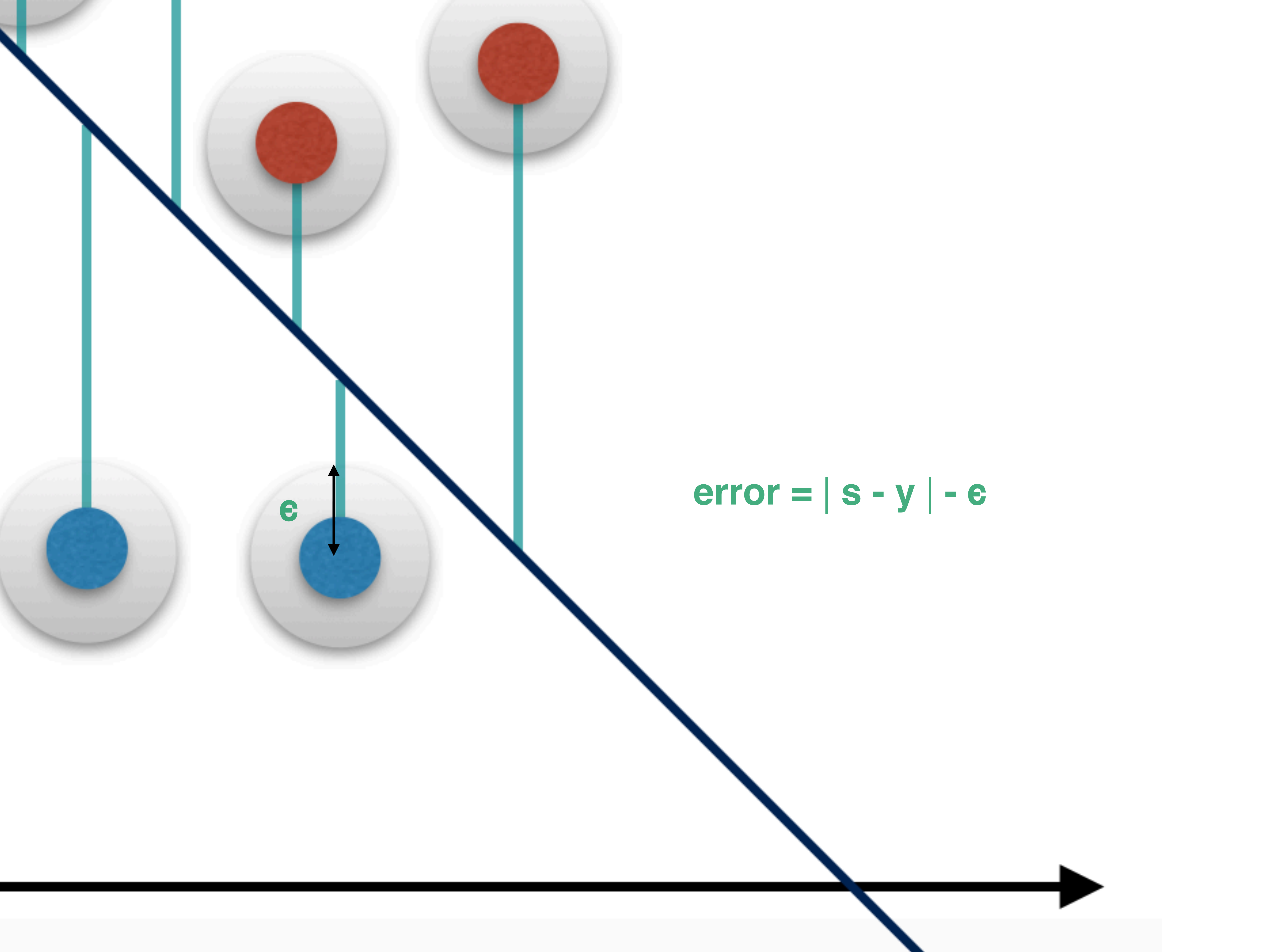
error = | s - y |

But your data may have some noise…

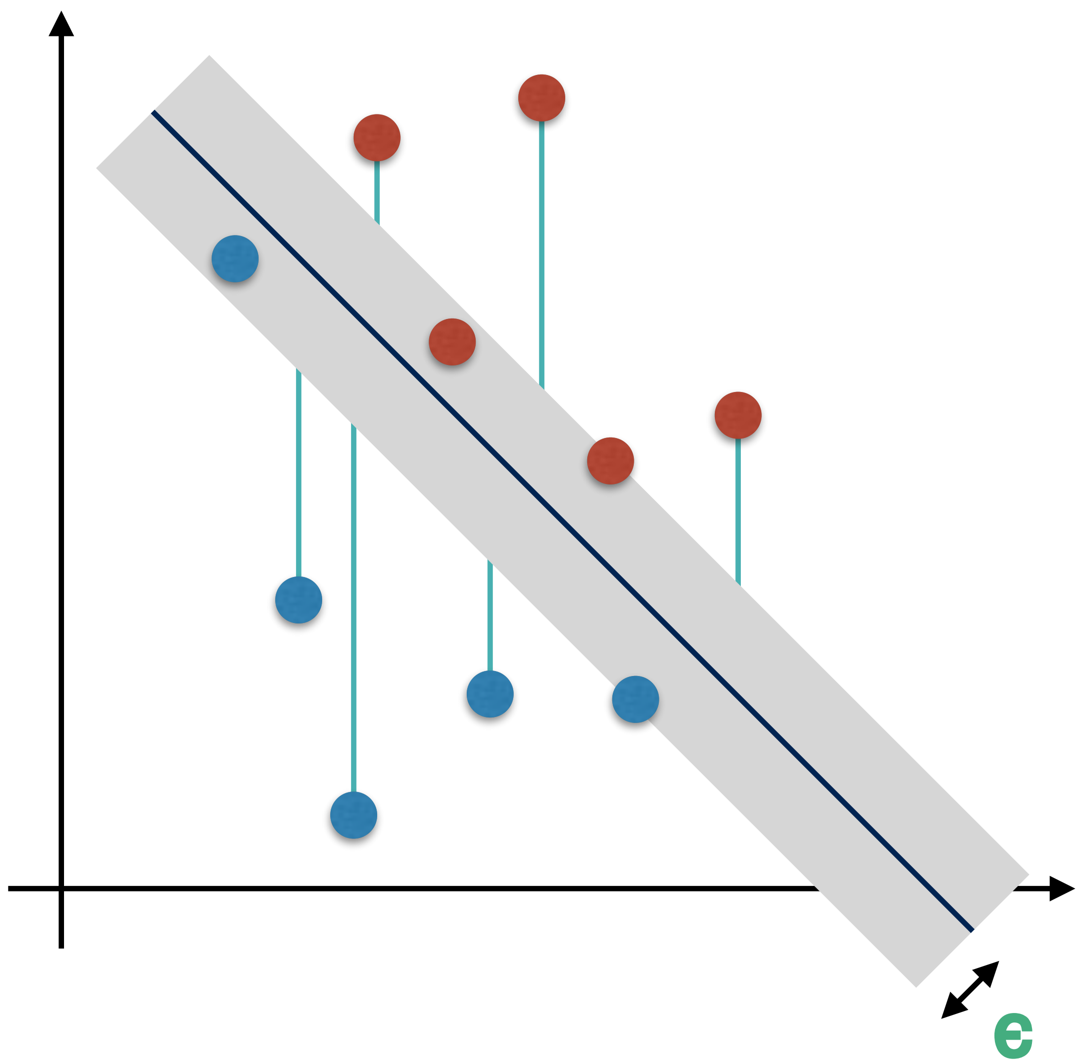
SVR Main Concept

1. Large margin
2. Regularization
3. Kernel



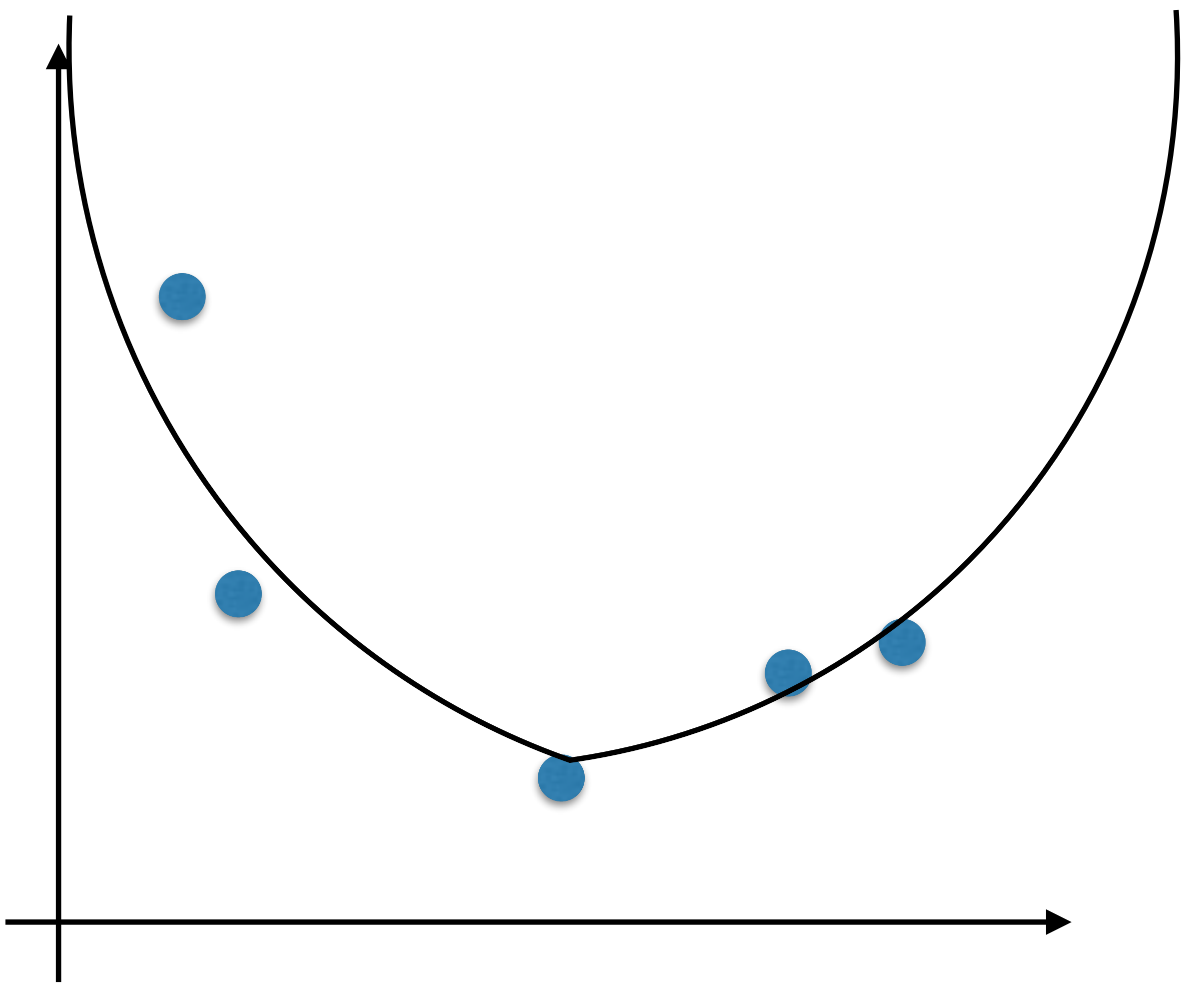

ϵ
error = | s - y | - ϵ

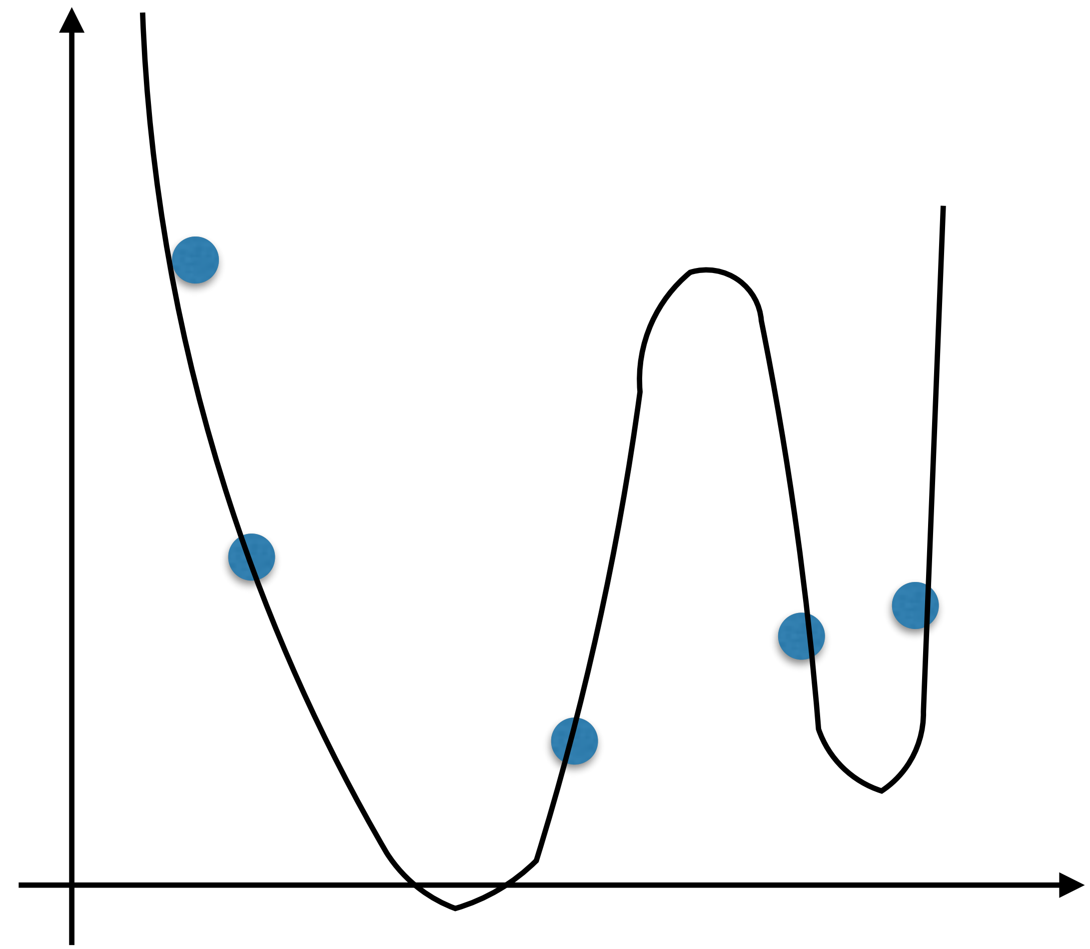
1.Tube Regression



error = 0 , inside the tube
= | s - y | - ϵ , outside the tube

2.Regularization : Avoid Overfitting



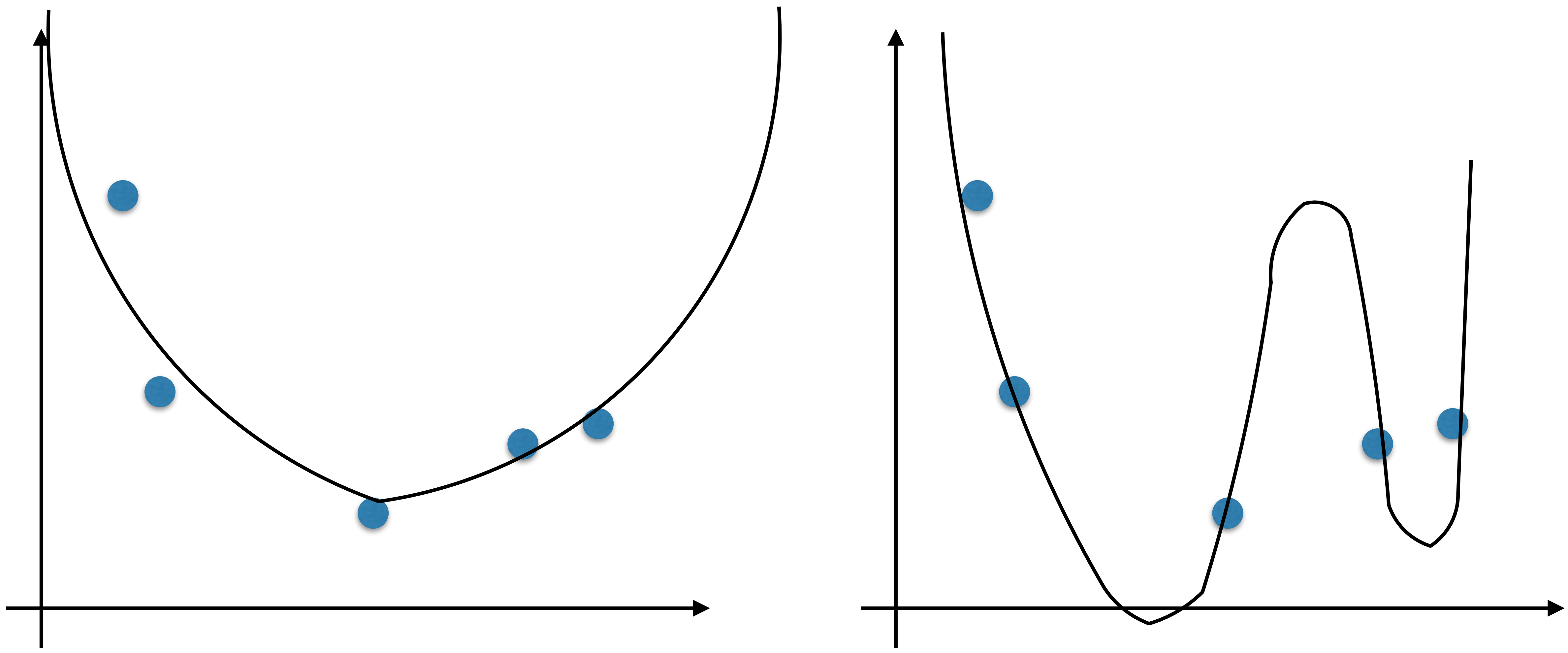


$\boldsymbol{W_a X} = W_0 + W_1 X + W_2 X^2$

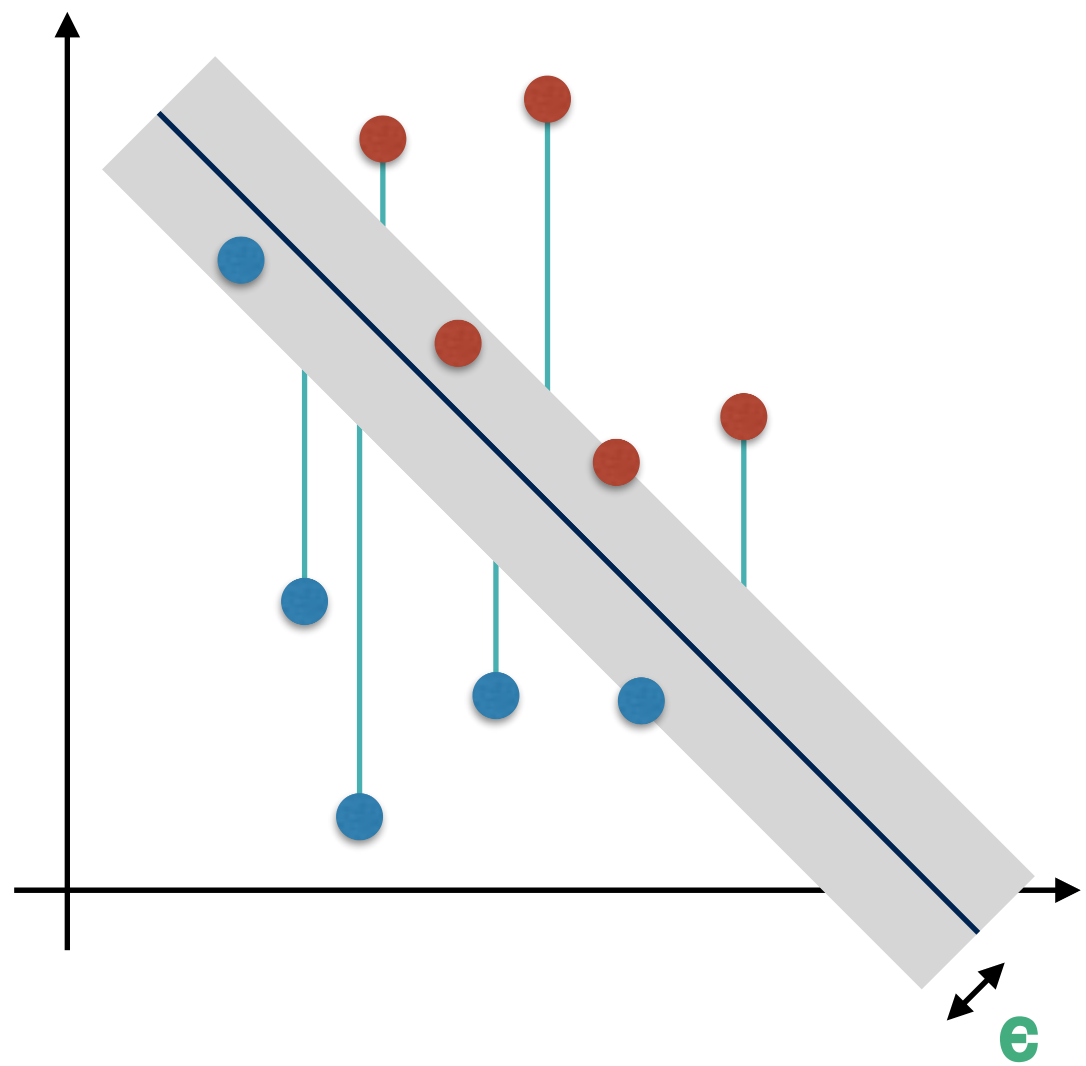
$\boldsymbol{W_b X} = W_0 + W_1 X + W_2 X^2 + W_3 X^3 + W_4 X^4$

Add constraints: $w_3 = w_4 = 0$

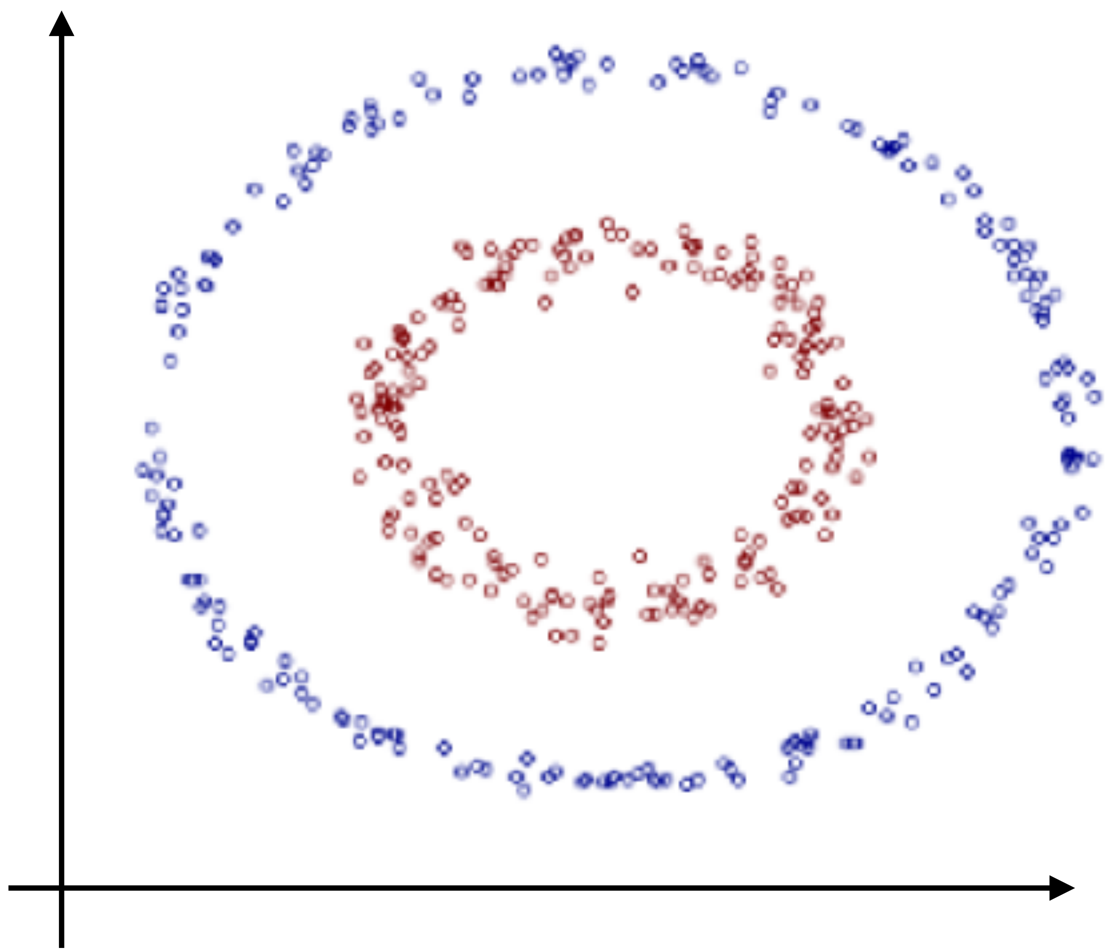
Regularization : Avoid Overfitting



$$\boldsymbol{w_a}^T \boldsymbol{w_a} < \boldsymbol{w_b}^T \boldsymbol{w_b}$$

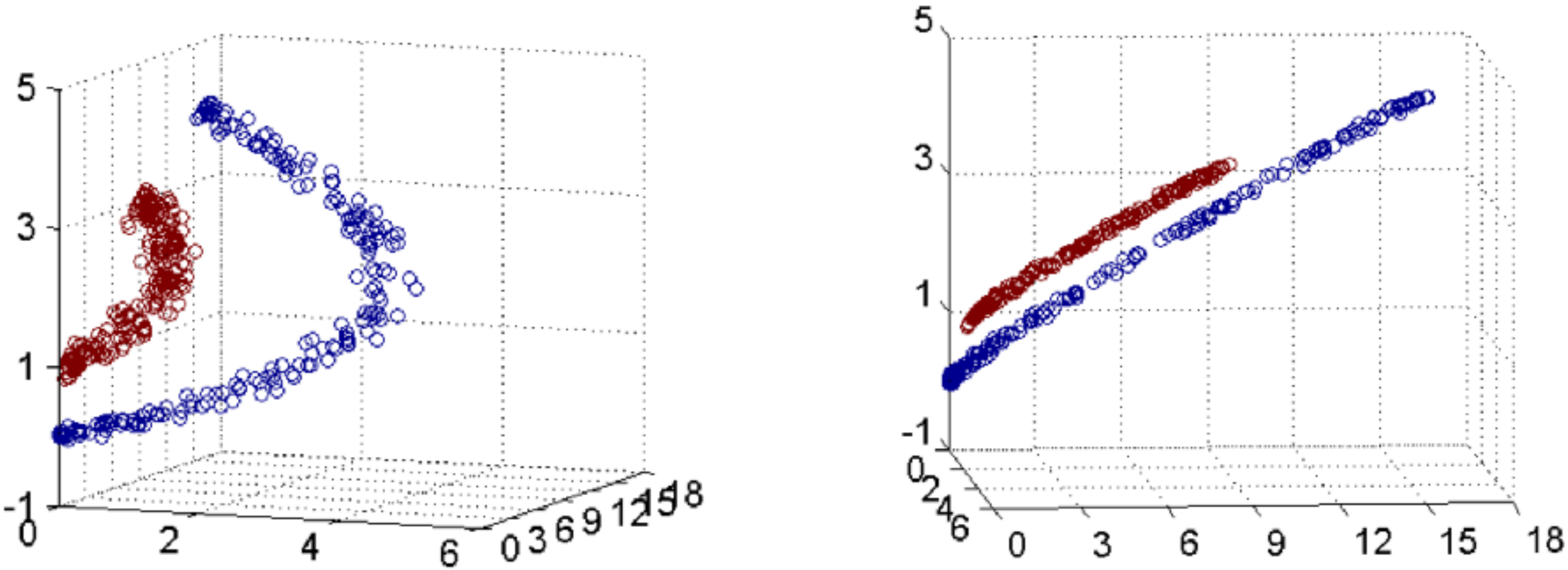


$$\min_{w,b,\xi,\xi^*} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i + C \sum_{i=1}^{l} \xi_i^*$$



What if your data is too complicated to describe?

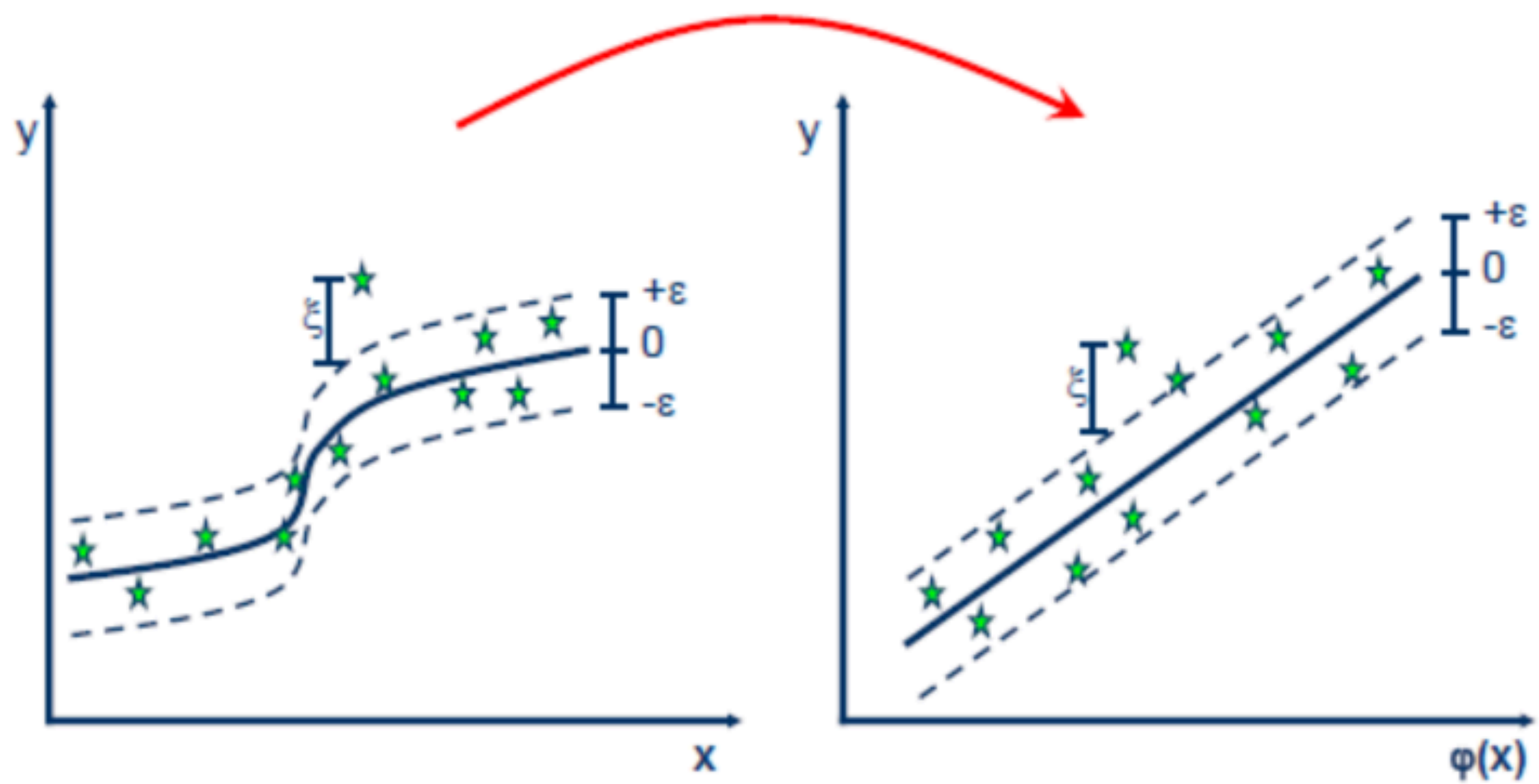
3.Kernel



Transform the data into a higher dimensional feature space to make it possible to perform the linear separation.

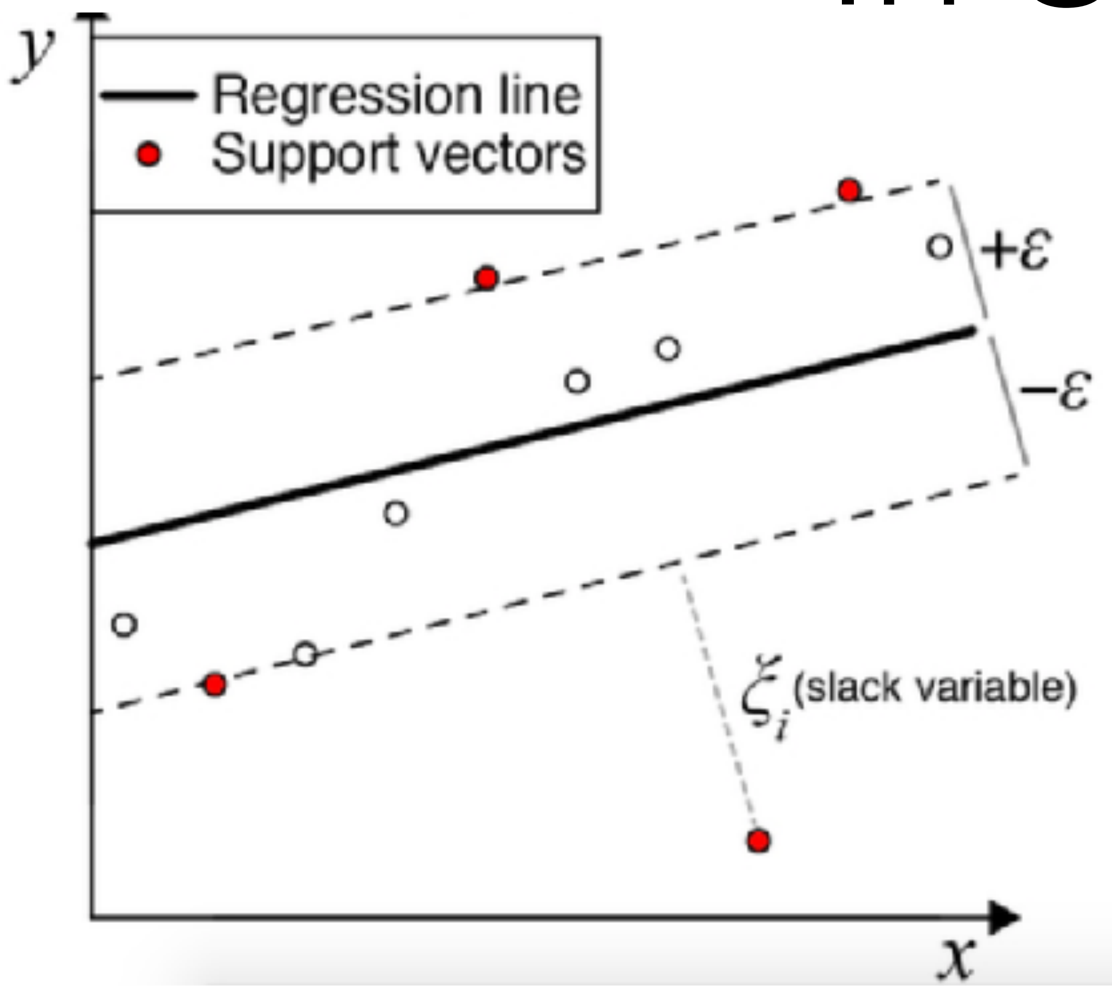
http://blog.pluskid.org/?p=685

Kernel



Transform the data into a higher dimensional feature space to make it possible to perform the linear separation.

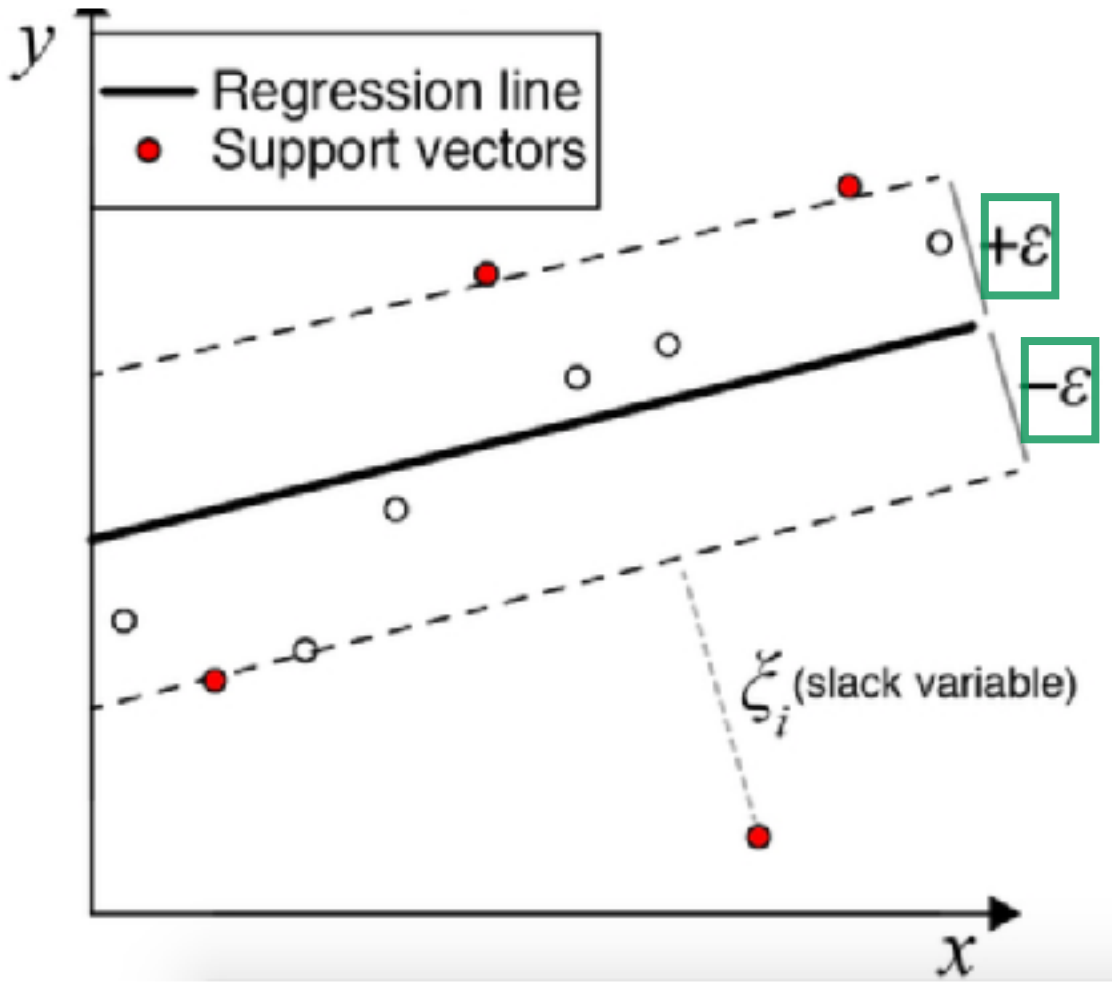
In Sum: SVR



$$\begin{aligned}
\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \quad & \frac{1}{2}\boxed{\boldsymbol{w}^T\boldsymbol{w}} + C\boxed{\sum_{i=1}^{l}\xi_i} + C\boxed{\sum_{i=1}^{l}\xi_i^*} \\
\text{subject to} \quad & \boldsymbol{w}^T\boxed{\phi(\boldsymbol{x}_i)} + b - z_i \leq \epsilon + \xi_i, \\
& z_i - \boldsymbol{w}^T\boxed{\phi(\boldsymbol{x}_i)} - b \leq \epsilon + \xi_i^*, \\
& \xi_i, \xi_i^* \geq 0, i = 1, \ldots, l.
\end{aligned}$$

- Choose a proper "kernel" to transform data.

- Minimize loss function to reduce total error and limit the model complexity.



$$
\begin{aligned}
\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \boxed{C}\sum_{i=1}^{l}\xi_i + \boxed{C}\sum_{i=1}^{l}\xi_i^* \\
\text{subject to} \quad & \boldsymbol{w}^T\boxed{\phi(\boldsymbol{x}_i)} + b - z_i \leq \boxed{\epsilon} + \xi_i, \\
& z_i - \boldsymbol{w}^T\boxed{\phi(\boldsymbol{x}_i)} - b \leq \boxed{\epsilon} + \xi_i^*, \\
& \xi_i, \xi_i^* \geq 0, \; i = 1, \ldots, l.
\end{aligned}
$$

sklearn.svm.SVR

```
class sklearn.svm.SVR(kernel='rbf', degree=3, gamma='auto', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1,
```

Trade-off

How to Choose Kernel?

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^*$$

$$\begin{aligned}\text{subject to}\quad & w^T\boxed{\phi(x_i)} + b - z_i \le \epsilon + \xi_i,\\ & z_i - w^T\boxed{\phi(x_i)} - b \le \epsilon + \xi_i^*,\\ & \xi_i, \xi_i^* \ge 0, i = 1, \ldots, l.\end{aligned}$$

Dual Problem

$$\min_{\alpha,\alpha^*} \frac{1}{2}(\alpha - \alpha^*)^T \boxed{Q(\alpha - \alpha^*)} + \epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} z_i(\alpha_i - \alpha_i^*)$$

$$\begin{aligned}\text{subject to}\quad & e^T(\alpha - \alpha^*) = 0,\\ & 0 \le \alpha_i, \alpha_i^* \le C, i = 1, \ldots, l,\end{aligned}$$

$$\text{where } \boxed{Q_{ij} = K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j).}$$

The advantage of Kernel function:

Easily compute the high dimensional inner product in low dimensional space

For example, $\mathbf{x} = (1, x_1, x_2, x_3)$

After 2nd order polynomial transform, $\Phi_2(\mathbf{x}) = (1, x_1, x_2, x_3, x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2)$

$$\begin{aligned}\Phi_2(\mathbf{x})^T\Phi_2(\mathbf{x}') &= 1 + \sum_{i=1}^{d} x_i x_i' + \sum_{i=1}^{d}\sum_{j=1}^{d} x_i x_j x_i' x_j' = 1 + \sum_{i=1}^{d} x_i x_i' + \sum_{i=1}^{d} x_i x_i' \sum_{j=1}^{d} x_j x_j'\\ &= 1 + \mathbf{x}^T\mathbf{x}' + (\mathbf{x}^T\mathbf{x}')(\mathbf{x}^T\mathbf{x}')\end{aligned}$$

How to Choose Kernel?

Good Transform Φ(x) + Inner product:
the choose of kernel function

e.g. $$\Phi_2(\mathbf{x})^T \Phi_2(\mathbf{x}') = 1 + \mathbf{x}^T\mathbf{x}' + (\mathbf{x}^T\mathbf{x}')(\mathbf{x}^T\mathbf{x}')$$

sklearn.svm.SVR

class `sklearn.svm.` **SVR** (*kernel='rbf', degree=3, gamma='auto', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1,*

kernel : string, optional (default='rbf')

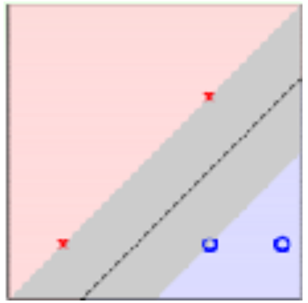
Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to precompute the kernel matrix.

How to Choose Kernel?

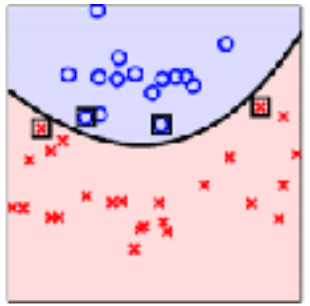
Good Transform Φ(x) + Inner product:
the choose of kernel function

Linear	Polynomial	Gaussian
$K(\mathbf{x},\mathbf{x}') = \mathbf{x}^T\mathbf{x}'$	$K(\mathbf{x},\mathbf{x}') = (\zeta + \gamma\mathbf{x}^T\mathbf{x}')^Q$	$K(\mathbf{x},\mathbf{x}') = \exp(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2)$
Fast! Try this first!	3 parameters to choose	1 parameter to choose
Restricted to linear separable	powerful than linear	Very powerful, but slow…

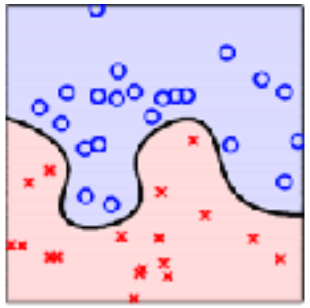
'linear'



'poly'



'rbf'



Part 3

Demo Code