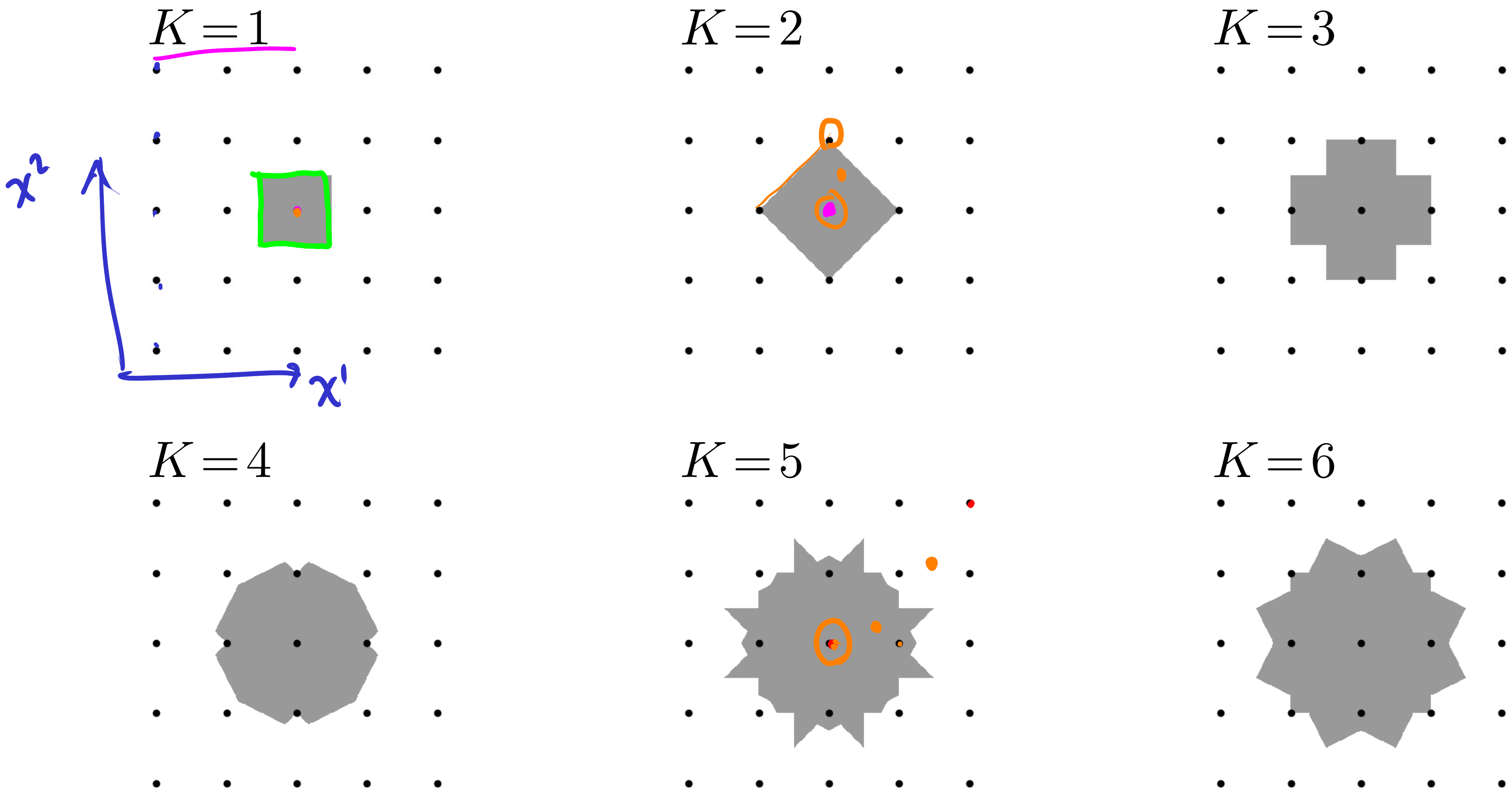
Statistics and Data Science for Engineers
E178 / ME276DS

Kernel functions, Support vector machines

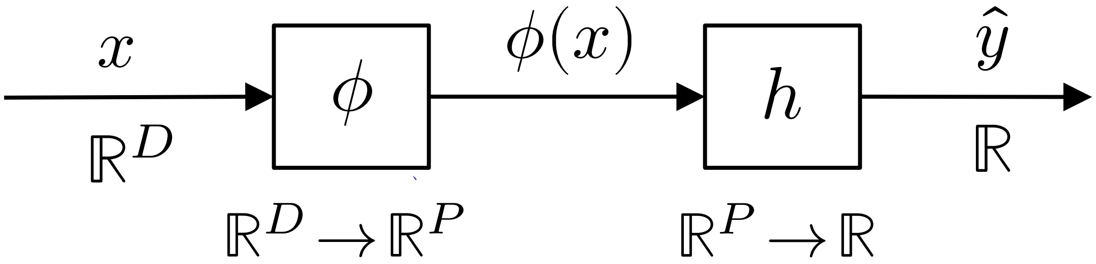
- Neural Networks, Decision trees. ... non-convex. ⇒ local minimum.
-

KNN on a grid



Recall Linear Regression

Setup



Features: $\phi(x) = [\overset{x}{\phi_1(x)} \quad \overset{x^2}{\dots} \quad \overset{x^3}{\phi_P(x)}]$ $\qquad$ $\phi(x)$ ... row vector

Model: $\boxed{\hat{y} = \theta_0 + \underbrace{\phi(x)\underline{\theta}_1}_{\phi_1(x)\cdot\theta_1 + \phi_2(x)\cdot\theta_2 \dots + \phi_P(x)\cdot\theta_P}}$ $\qquad$ $\underline{\theta}_1$ ... column vector

Goal: $\underbrace{(\hat{\theta}_0, \hat{\underline{\theta}}_1)} = \underset{(\theta_0, \underline{\theta}_1) \in \mathbb{R}^{P+1}}{\operatorname{argmin}} \sum_{i=1}^{N} \underbrace{\left(\theta_0 + \phi(x_i)\,\underline{\theta}_1 - y_i\right)^2} + \overbrace{\lambda \sum_{j=1}^{P} \theta_j^2}^{\text{Ridge regul.}}$

Training data

$N >> P$

Φ_c

P

N

0	0.247746	36.0	266.0	0.247746	88.019654	57.000823
1	0.179340	37.0	365.0	0.179340	27.211935	22.471227
2	0.956807	21.0	151.0	0.956807	97.456012	56.357366
3	0.869653	43.0	437.0	0.869653	43.203221	34.6[illegible]2475
4	0.825345	47.0	160.0	0.825345	98.933930	64.426163
5	0.331114	321.0	371.0	0.331114	8.257917	14.218720
6	0.765523	17.0	364.0	0.765523	96.696783	59.123769
7	0.956807	21.0	151.0	0.956807	97.456012	52.983470

Φ $\quad$ $\mathbf{Y}$

$\phi(x_i)$ $\quad$ y_i

$\hat{\mu}_X$ $\quad$ $\hat{\mu}_Y$

Matrices:

$$\Phi = \begin{bmatrix} \phi(x_1) \\ \vdots \\ \phi(x_N) \end{bmatrix} \in \mathbb{R}^{N\times P} \qquad \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^{N\times 1}$$

Means:

$$\hat{\mu}_X = \frac{1}{N}\mathbf{1}_N^T\Phi \qquad \hat{\mu}_Y = \frac{1}{N}\mathbf{1}_N^T\mathbf{Y}$$

Centered matrices:

$$\Phi_c = \Phi - \mathbf{1}_N\hat{\mu}_X \qquad \mathbf{Y}_c = \mathbf{Y} - \mathbf{1}_N\hat{\mu}_Y$$

Solution

p

p

ridge

Stationarity condition: $$(\Phi_c^T \Phi_c + \lambda I_P)\,\underline{\hat{\theta}}_1 = \Phi_c^T \mathbf{Y}_c$$

Optimal parameters: $$\underline{\hat{\theta}}_1 = \overbrace{(\Phi_c^T \Phi_c + \lambda I_P)^{-1}}^{p \times p}\, \Phi_c^T \mathbf{Y}_c$$

normal eqn.

$$\hat{\theta}_0 = \underbrace{\hat{\mu}_Y - \hat{\mu}_X\, \underline{\hat{\theta}}_1}$$

Prediction: $$\hat{y} = \hat{\theta}_0 + \phi(x)\underline{\hat{\theta}}_1$$

h(x).

$$= \underbrace{\hat{\mu}_Y + \phi_c(x)\underline{\hat{\theta}}_1}_{\text{another prediction formula.}}$$

Solution

Stationarity condition: $\left(\Phi_c^T\Phi_c + \lambda I_P\right)\underline{\hat{\theta}}_1 = \Phi_c^T \mathbf{Y}_c$

Optimal parameters:

$$\underline{\hat{\theta}}_1 = \left(\Phi_c^T\Phi_c + \lambda I_P\right)^{-1}\Phi_c^T \mathbf{Y}_c$$

$$\hat{\theta}_0 = \hat{\mu}_Y - \hat{\mu}_X\,\underline{\hat{\theta}}_1$$

Prediction:

$$\hat{y} = \hat{\theta}_0 + \phi(x)\underline{\hat{\theta}}_1$$

$$\hat{y} = \hat{\mu}_Y - \hat{\mu}_X\,\underline{\hat{\theta}}_1 + \phi(x)\underline{\hat{\theta}}_1$$

$$\hat{y} - \hat{\mu}_Y = (\phi(x) - \hat{\mu}_X)\underline{\hat{\theta}}_1$$

$$\hat{y}_c = \phi_c(x)\underline{\hat{\theta}}_1$$

Kernel form of Ridge regression

$\Phi_c \Phi_c^T =$ N × N

Standard form

- $\hat{\underline{\theta}}_1 = \big(\underbrace{\Phi_c^T\Phi_c + \lambda I_P}_{}\big)^{-1} \Phi_c^T \mathbf{Y}_c$ — $P \times P$
- $\hat{y} = \underbrace{\hat{\mu}_Y}_{} + \phi_c(x)\hat{\underline{\theta}}_1$

$$\hat{\underline{\theta}}_1 = \Phi_c^T \alpha$$

$$\alpha = -\frac{1}{\lambda}\left(\Phi_c \hat{\underline{\theta}}_1 - \mathbf{Y}_c\right)$$

Kernel form

$$\alpha = \Big(\underbrace{\Phi_c\Phi_c^T + \lambda I_N}_{N \times N}\Big)^{-1} \mathbf{Y}_c$$ ... $\mathbb{R}^N$ ... weights on each training data point.

$$\hat{y} = \underbrace{\hat{\mu}_Y}_{} + \sum_{i=1}^{N} \alpha_i \underbrace{\phi_c(x_i)\cdot\phi_c(x)}_{K(x_i, x)}$$

Kernel form of Ridge regression

Training: $\alpha = \big(\underbrace{\Phi_c \Phi_c^T}_{N \times N} + \lambda\, I_N\big)^{-1} \mathbf{Y}_c$

$\boxed{\mathbb{K} = \Phi_c \Phi_c^T} \quad \in \mathbb{R}^{N \times N}$ ... Kernel matrix

$$
\begin{bmatrix} \phi_c(x_1) \\ \phi_c(x_2) \\ \vdots \\ \phi_c(x_N) \end{bmatrix}
\begin{bmatrix} \phi_c^T(x_1) & \cdots & \phi_c^T(x_N) \end{bmatrix}
$$

$\Phi_c \qquad \Phi_c^T$

$$
= \begin{bmatrix} \phi_c(x_1)\cdot\phi_c(x_1) & \dots & \phi_c(x_1)\cdot\phi_c(x_N) \\ \vdots & & \vdots \\ \phi_c(x_N)\cdot\phi_c(x_1) & \dots & \phi_c(x_N)\cdot\phi_c(x_N) \end{bmatrix}
$$

$$
= \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & & \vdots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix}
$$

symmetric

... Kernel function

$k(x, z) = \phi_c(x) \cdot \phi_c(z)$

symmetric

Prediction: $h(x) = \hat{y} = \hat{\mu}_Y + \sum_{i=1}^{N} \alpha_i\, k(x_i, x)$

Example

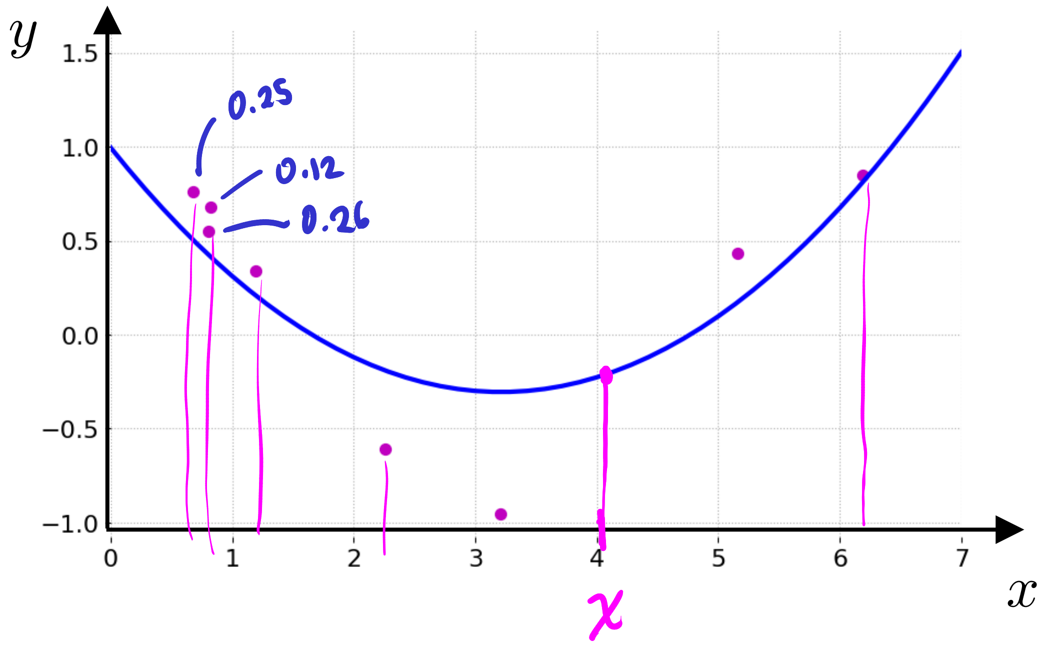
$N = 8$

$D = 1$

$P = 2$

$$\phi(x) = (x, x^2)$$

$$\Phi_c = \begin{array}{c} \begin{array}{cc} x-\text{mean} & x^2-\text{mean} \end{array} \\ \begin{bmatrix} -1.87 & -9.98 \\ -1.74 & -9.80 \\ -1.72 & -9.77 \\ -1.35 & -9.02 \\ -0.28 & -5.33 \\ 0.8 & -0.13 \\ 2.62 & 16.15 \\ 3.66 & 27.88 \end{bmatrix} \end{array} \qquad \mathbf{Y}_c = \begin{bmatrix} 0.50 \\ 0.29 \\ 0.42 \\ 0.09 \\ -0.87 \\ -1.21 \\ 0.18 \\ 0.60 \end{bmatrix}$$



$$X = \begin{bmatrix} 0.8 \\ 0.9 \\ 1.1 \\ \vdots \\ 6.2 \end{bmatrix} \xrightarrow{\phi} \begin{array}{c} \begin{array}{cc} x & x^2 \end{array} \\ \begin{bmatrix} 0.8 & 0.64 \\ 0.9 & 0.81 \\ 1.1 & 1.21 \\ \vdots & \vdots \\ 6.2 & \end{bmatrix} \end{array}$$

$$\left(\bar{x} \quad \bar{\bar{x}} \right) \quad \hat{\mu}_x$$

Standard solution

$$\hat{\theta}_0 = 1$$

$$\hat{\theta}_1 = \begin{array}{cl} \begin{bmatrix} -0.81 \\ 0.13 \end{bmatrix} & \begin{array}{l} x \\ x^2 \end{array} \end{array}$$

$$\hat{y}(x) = 1 - 0.81x + 0.13x^2$$

Kernel-based solution

$$\alpha = \begin{bmatrix} 0.25 \\ 0.12 \\ 0.26 \\ 0.13 \\ -0.42 \\ -0.65 \\ 0.26 \\ 0.04 \end{bmatrix} \in \mathbb{R}^8$$

4

$$\phi(x) = (x, x^2)$$

$$\phi_c(x) = \phi(x) - \hat{\mu}_X = (x - \bar{x}, x^2 - \bar{x^2})$$

$$k(x, z) = \phi_c(x) \cdot \phi_c(z)$$

4

$$\hat{y}(x) = \hat{\mu}_Y + \sum_{i=1}^{N} \alpha_i k(x_i, x)$$

Question

Given an *arbitrary* function $k(x, z)$, are there conditions that guarantee the existence of a feature function $\phi(x)$ such that $\underbrace{k(x, z)}$ $= \underbrace{\phi(x) \cdot \phi(z)}$?

Answer:

Yes!

→ 1. $K(x,z)$ must be symmetric : $K(x,z) = K(z,x) \quad \forall x,z \in \mathbb{R}^D$.

→ 2. $K(x,z)$ must be "positive semi-definite".

any training data. → $\mathbb{K} \Rightarrow$ must be positive semi-definite. $x^T \mathbb{K} x \geq 0$. $\forall x \in \mathbb{R}^D$.

Example: Polynomial kernel: $$k(x, z) = \left(x^T z + 1 \right)^{d}$$

The corresponding feature vector $\phi(x)$ contains monomials of x up to order d

$$x = \begin{bmatrix} x^1 \\ \vdots \\ x^D \end{bmatrix}$$

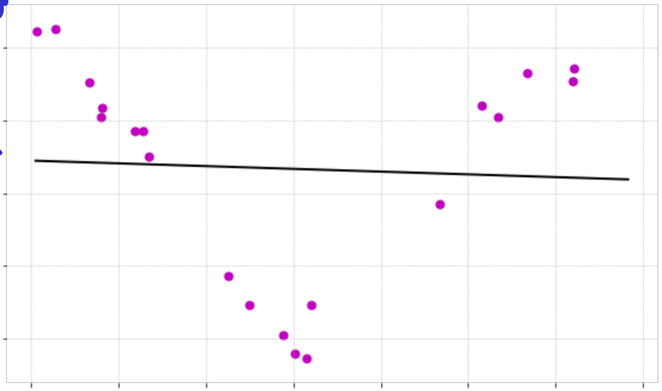
$$d=1: \quad \phi = \left(x^1, x^2, \dots, x^D \right).$$

$$d=2: \quad \phi = \left(\underbrace{x^1, x^2, \dots, x^D, (x^1)^2, \dots, (x^D)^2, x^1 \cdot x^2, x^1 \cdot x^3, \dots, x^{D-1} x^D} \right).$$

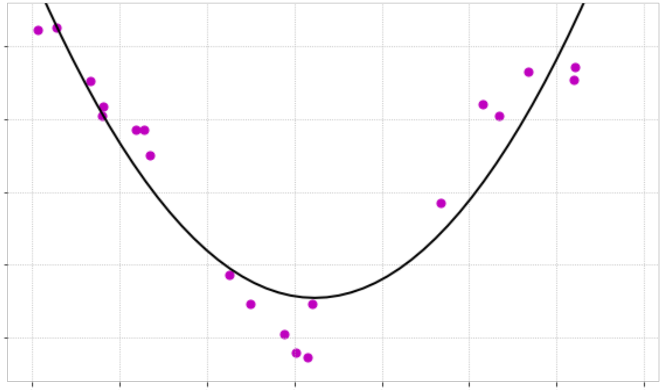
$$d=3: \quad \left(\quad , (x^1)^3, \dots (x^D)^3, x^1 (x^2)^2, \dots \right).$$

simple linear (ridge) regression.

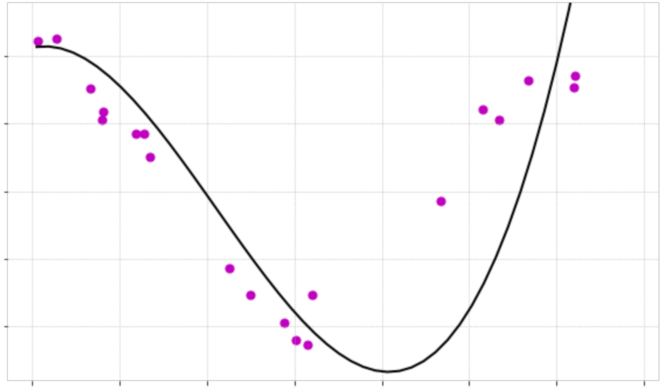
Linear $k(x, z) = x^T z$



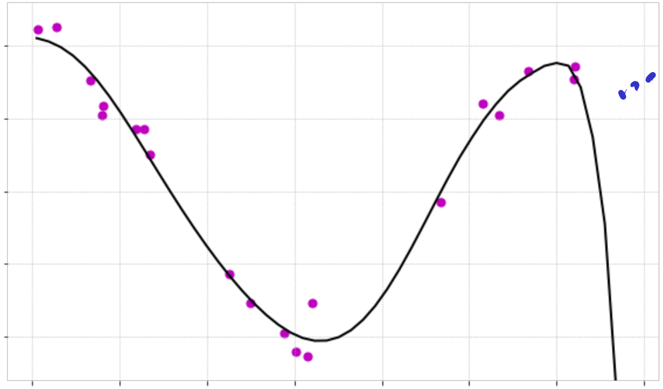
Quadratic $k(x, z) = (x^T z + 1)^2$



Poly 4 $k(x, z) = (x^T z + 1)^4$



Poly 10 $k(x, z) = (x^T z + 1)^{10}$

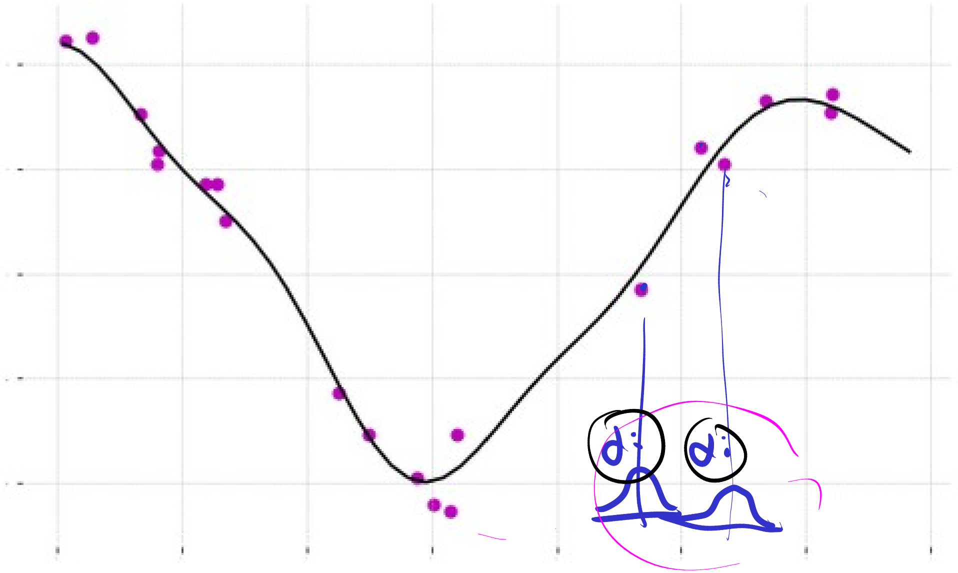


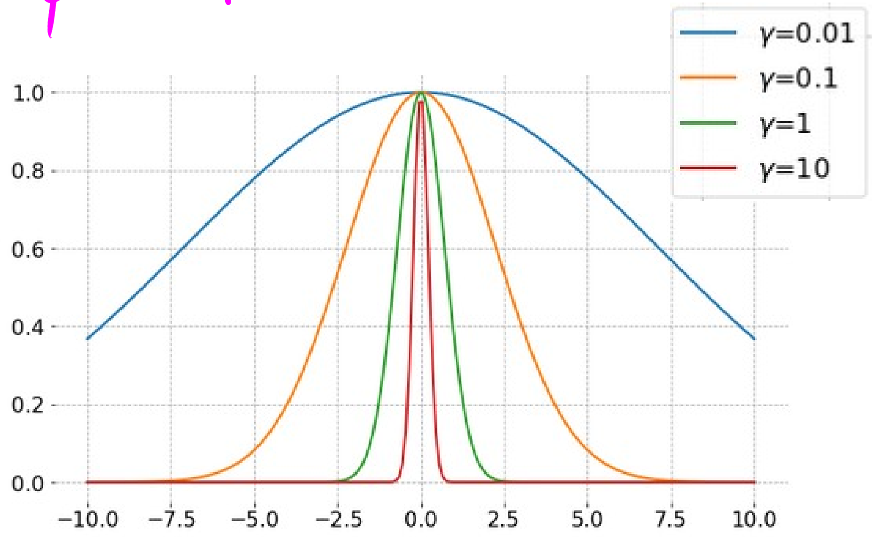
.... overfitting.

Gaussian kernel – a.k.a. Radial basis function (RBF)

$$k(x,z) = \underbrace{\exp\left(-\gamma \|x-z\|^2\right)}_{\text{"}\phi\text{" has } P=\infty}$$

$\gamma \approx 1/\sigma^2$



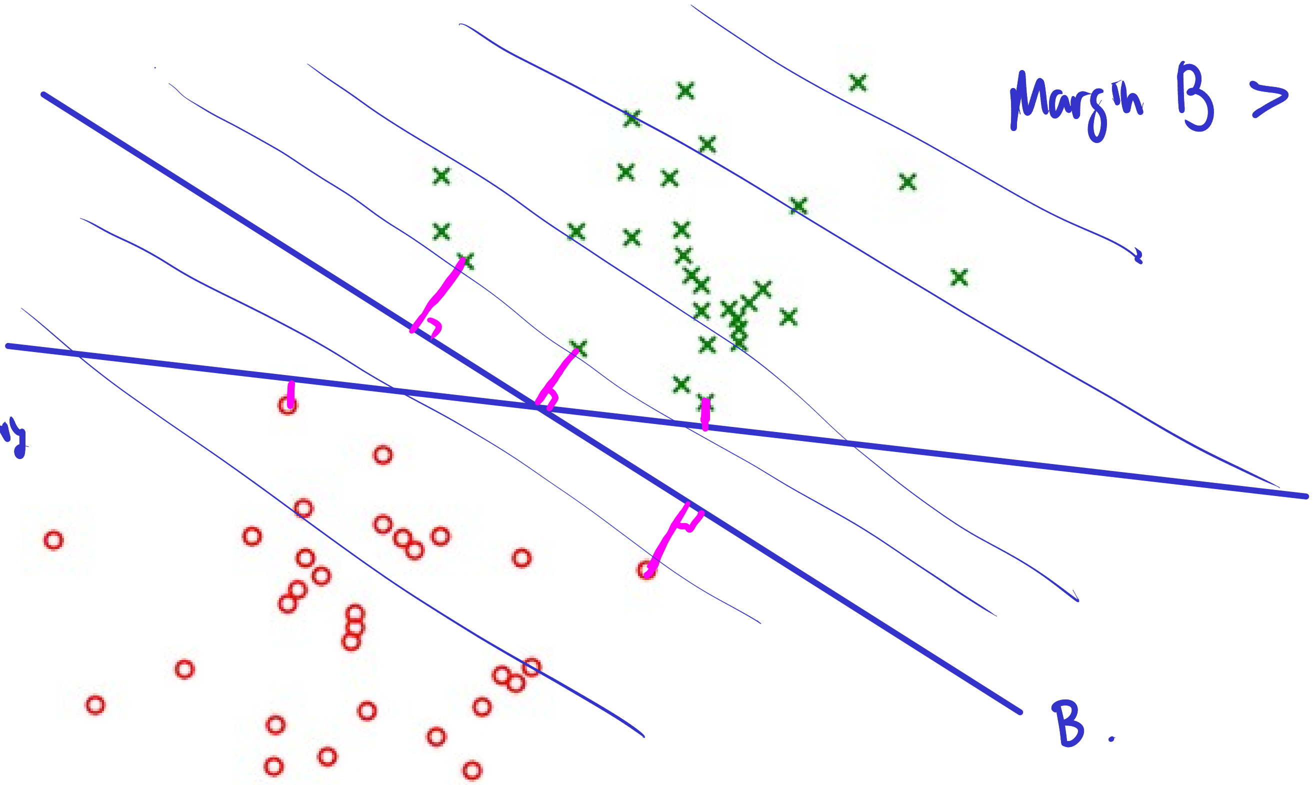


Maximum margin classifier

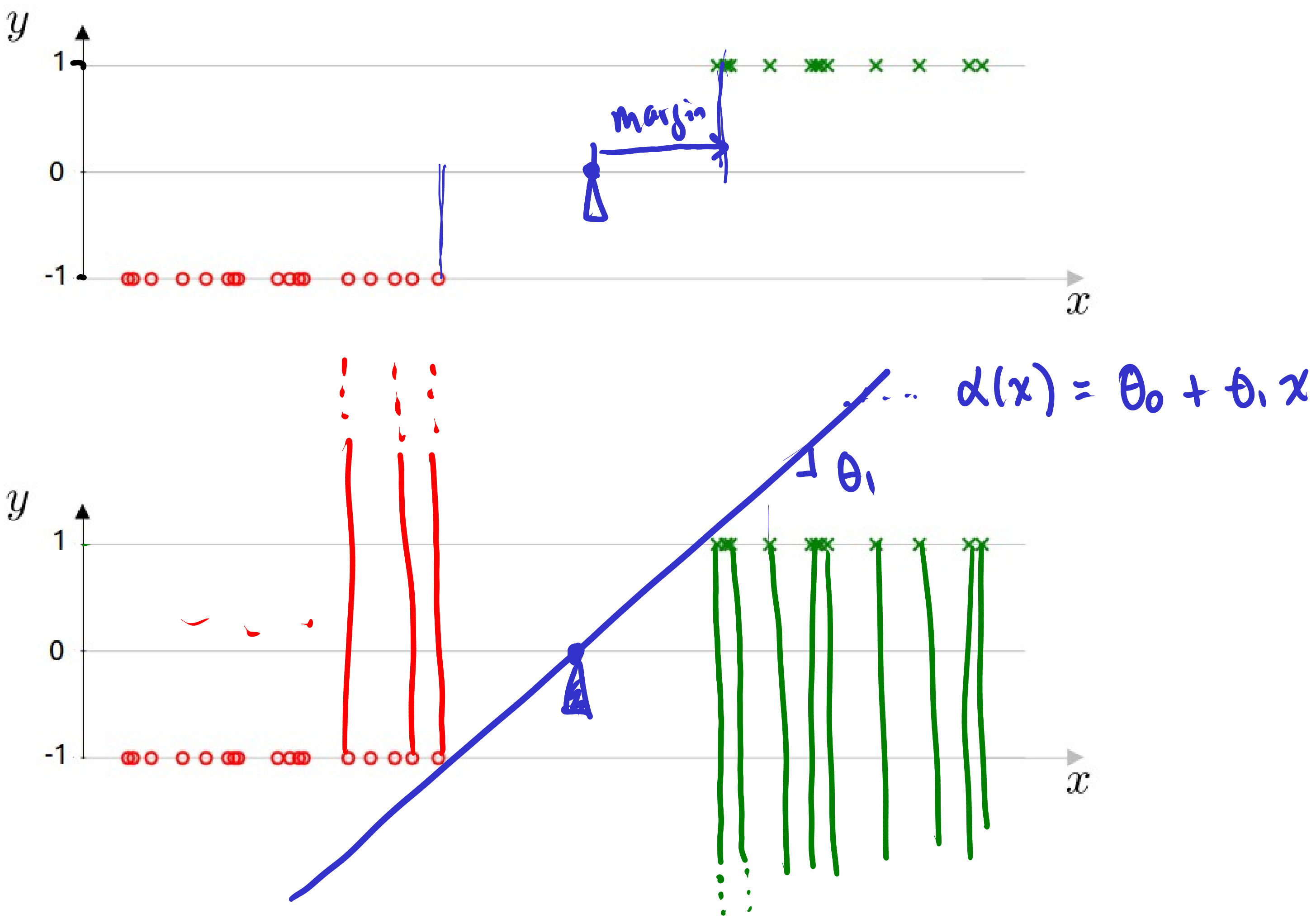
Assumption: The training data is linearly separable

Margin : Minimum perpendicular distance from training data to decision boundary

Margin B > Margin A.



Encoding : $\{1, -1\}$.

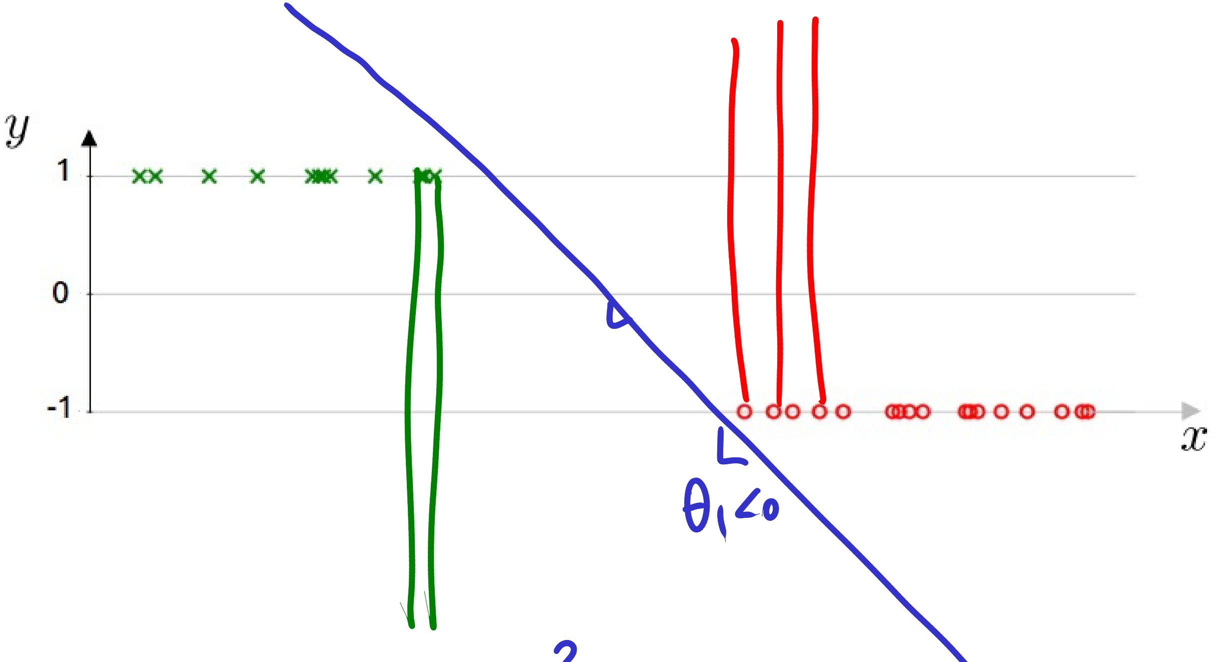


$$(\hat{\theta}_0, \hat{\theta}_1) = \underset{\theta_0, \theta_1}{\operatorname{argmin}} \quad \theta_1 \qquad \ldots \text{ convex (linear).}$$

$$\text{s.t.} \quad y_i = 1 : \quad \alpha(x_i) > 1 \qquad \text{(green)}$$

$$y_i = -1 : \quad \alpha(x_i) < -1 \qquad \text{(red)}$$

$$\alpha(x_i) \doteq \theta_0 + \theta_1 x_i$$



$$(\hat{\theta}_0, \hat{\underline{\theta}}_1) = \arg\min_{\theta_0, \theta_1} \|\underline{\theta}_1\|^2 \quad \ldots \text{ convex (quadratic).}$$

s.t. $y_i = 1: \quad \alpha(x_i) \geq 1 \quad$ (green)

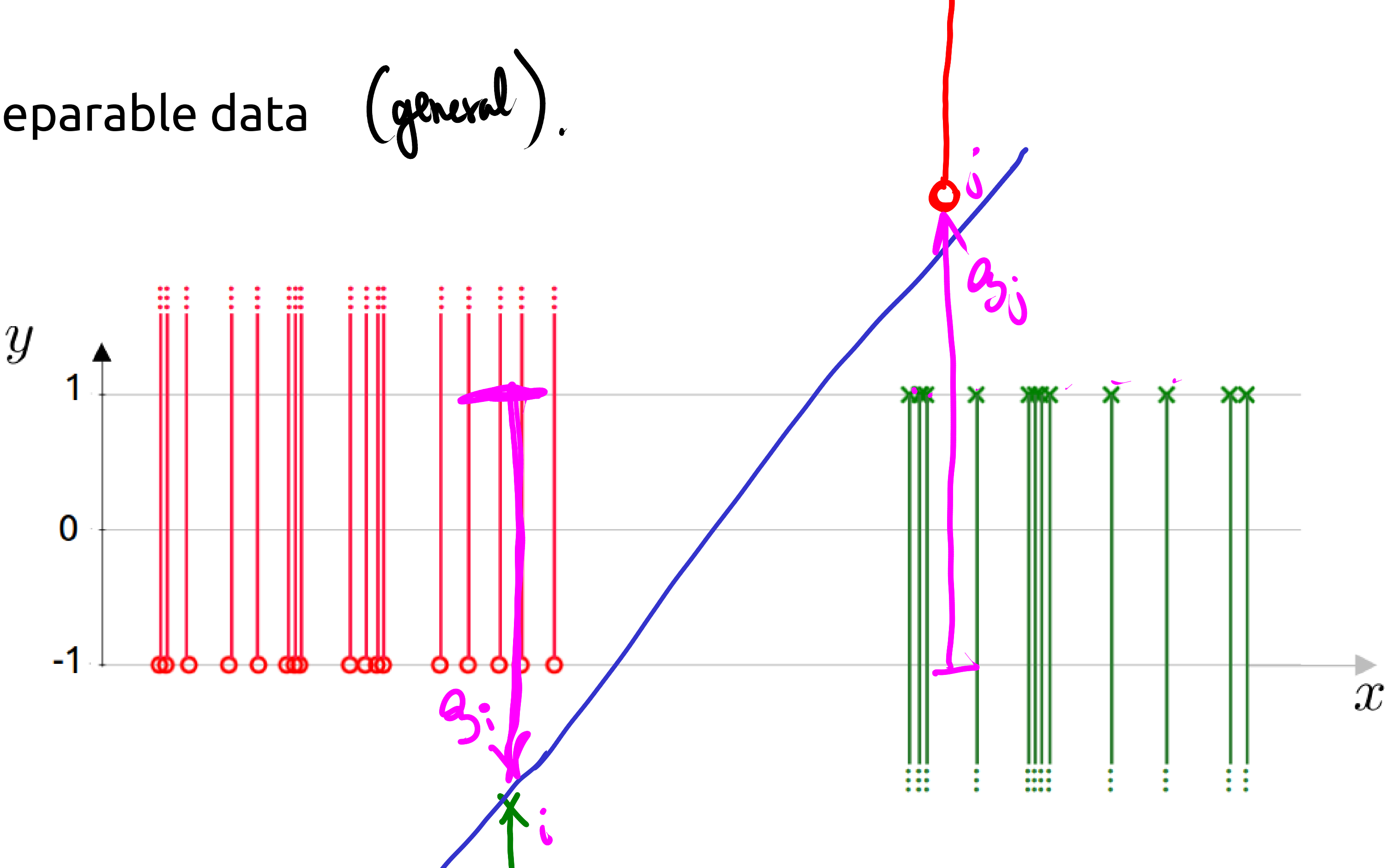
$y_i = -1: \quad \alpha(x_i) \leq -1 \quad$ (red)

$\alpha(x_i) \doteq \theta_0 + \underline{\theta}_1 x_i$

Simplification: Multiply both sides of the inequality constraints by y_i

$$\boxed{\begin{aligned} &\text{minimize} \quad \|\underline{\theta}_1\|^2 \\ &\quad \text{s.t.} \quad y_i\, \alpha(x_i) \geq 1 \end{aligned}}$$

Non-separable data (general).



New variable... $\xi_i > 0$
and as small as possible.

$$(\hat{\theta}_0, \hat{\underline{\theta}}_1) = \underset{\theta_0, \theta_1, \xi_i}{\arg\min} \|\vec{\theta}_1\|^2 + C \sum_{i=1}^{N} \xi_i$$

s.t. $\xi_i \geq 0$

$y_i = 1$: $\alpha(x_i) \geq 1 - \xi_i$ (green)

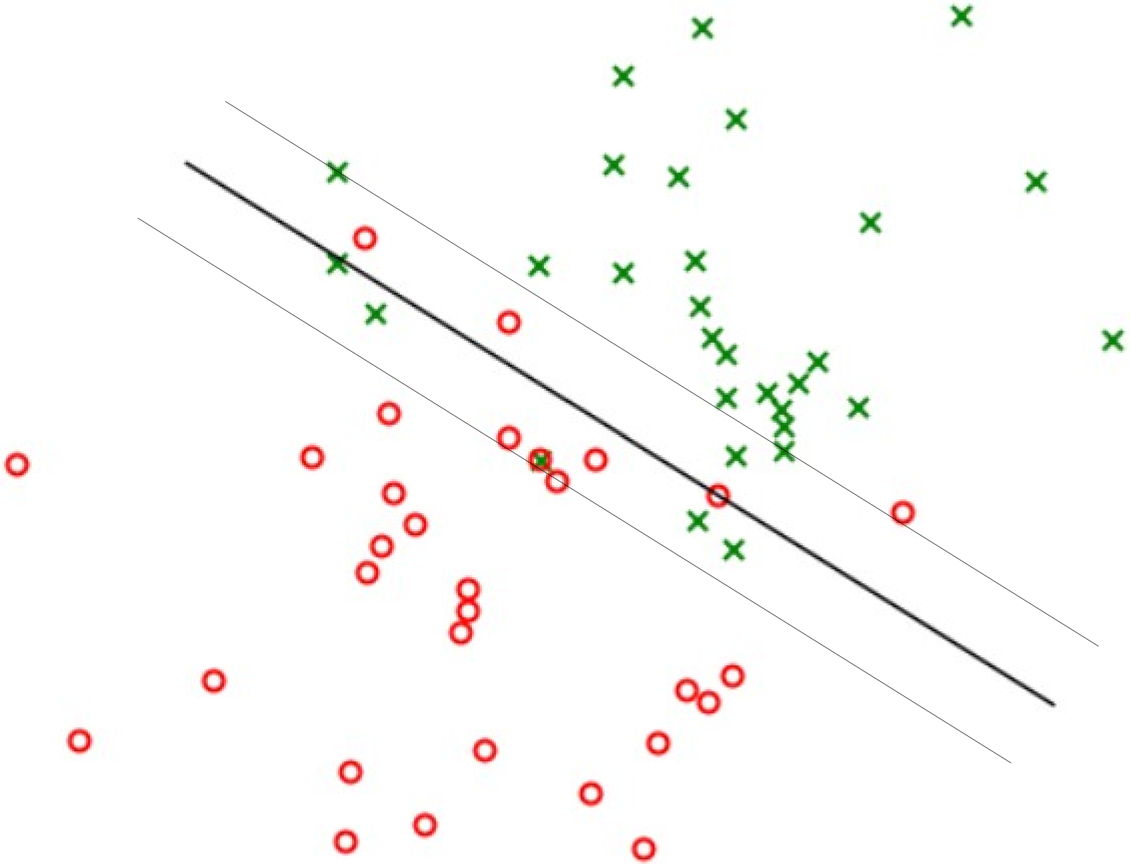
$y_i = -1$: $\alpha(x_i) \leq +1 + \xi_i$ (red)

$\alpha(x_i) = \theta_0 + \underline{\theta}_1 x_i$

$$(\hat{\theta}_0, \hat{\underline{\theta}}_1, \hat{\xi}) = \boxed{\underset{\theta_0, \underline{\theta}_1, \xi}{\arg\min} \|\underline{\theta}_1\|^2 + C \sum_{i=1}^{N} \xi_i \quad \text{s.t.} \quad \xi_i \geq 0, \quad y_i \alpha_i(x_i) \geq 1 - \xi_i}$$

MMC.

CONVEX.



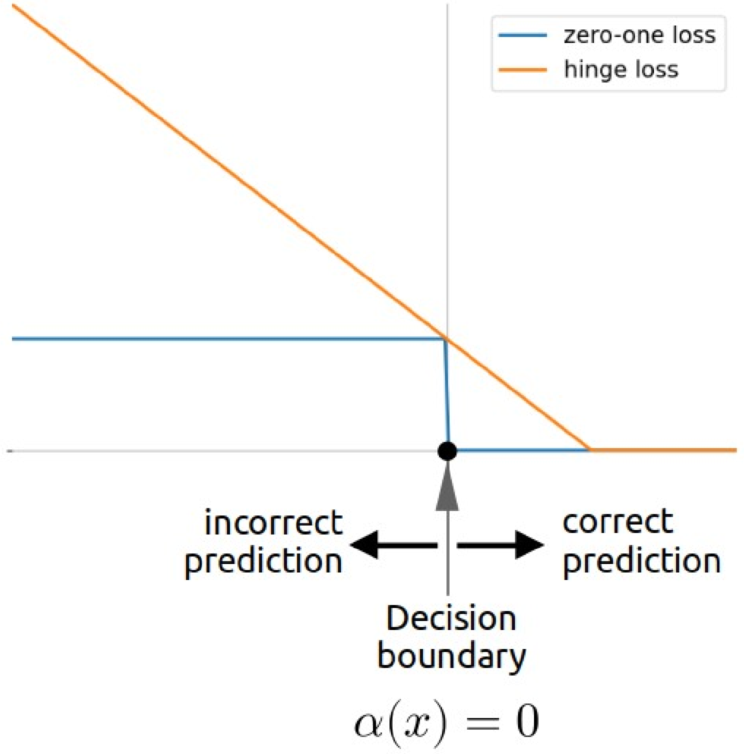
SVM as loss minimization

$$(\hat{\theta}_0, \hat{\underline{\theta}}_1) = \underset{\theta_0, \underline{\theta}_1}{\text{argmin}} \left(\sum_{i=1}^{N} L(y_i, \alpha(x_i)) \; + \; \lambda \, \|\underline{\theta}_1\|^2 \right)$$

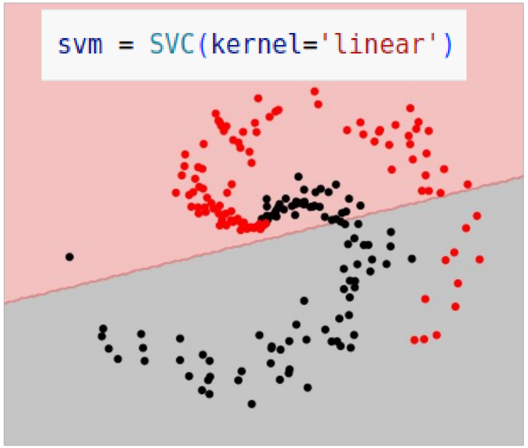
with

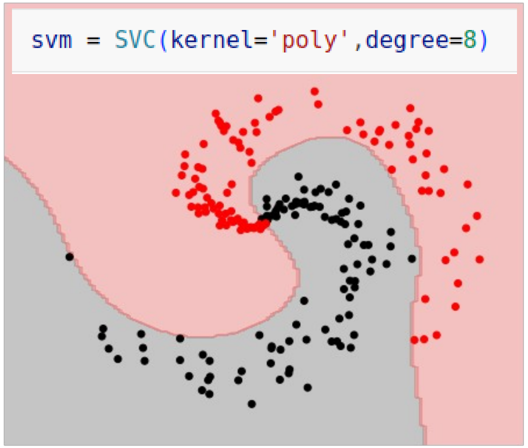
$$L(y_i, \hat{y}_i) = \text{max}(0, 1 - y_i \, \hat{y}_i)$$

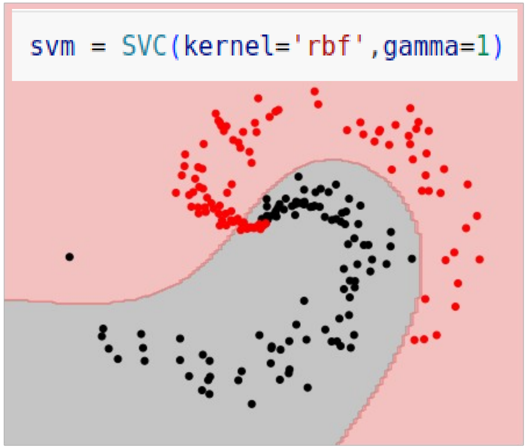
$$\alpha(x_i) = \theta_0 + x_i \underline{\theta}_1$$



Example: SVM on spiral dataset







Tuning the RBF kernel.