Learning From Data
Lecture 23
SVM's: Maximizing the Margin

M. Magdon-Ismail
CSCI 4100/6100

RECAP: Linear Models, RBFs, Neural Networks

Linear Model with Nonlinear Transform

$$h(\mathbf{x}) = \theta\left(w_0 + \sum_{j=1}^{\tilde{d}} w_j \Phi_j(\mathbf{x})\right)$$

Neural Network

$$h(\mathbf{x}) = \theta\left(w_0 + \sum_{j=1}^{m} w_j \theta\left(\mathbf{v}_j{}^{\mathrm{T}}\mathbf{x}\right)\right)$$

gradient descent

k-RBF-Network

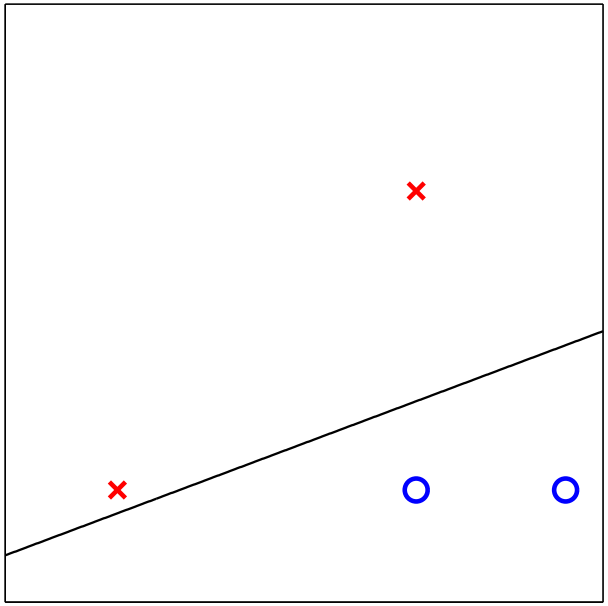
$$h(\mathbf{x}) = \theta\left(w_0 + \sum_{j=1}^{k} w_j \phi\left(\| \mathbf{x} - \boldsymbol{\mu}_j \|\right)\right)$$

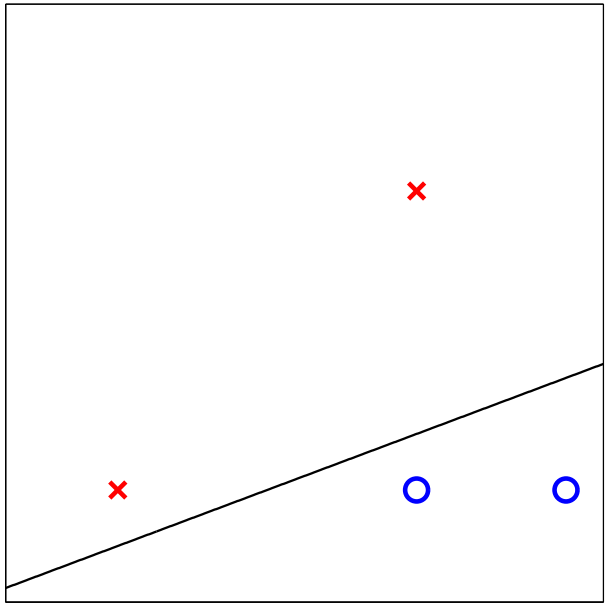
k-means

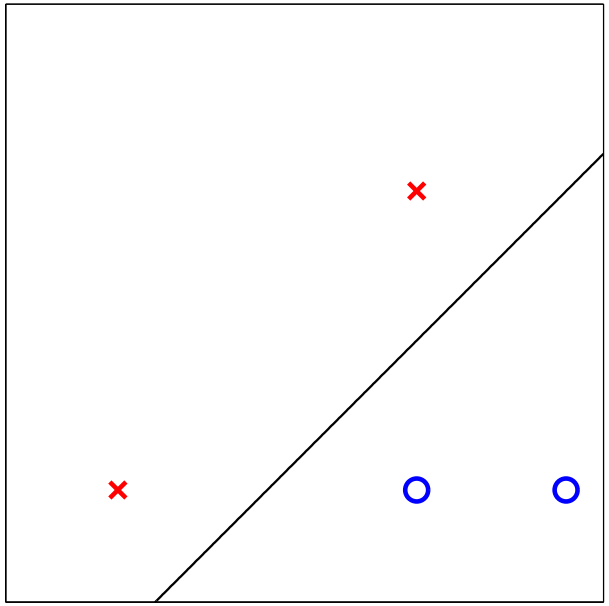
Neural Network: generalization of linear model by adding layers.

Support Vector Machine: more 'robust' linear model

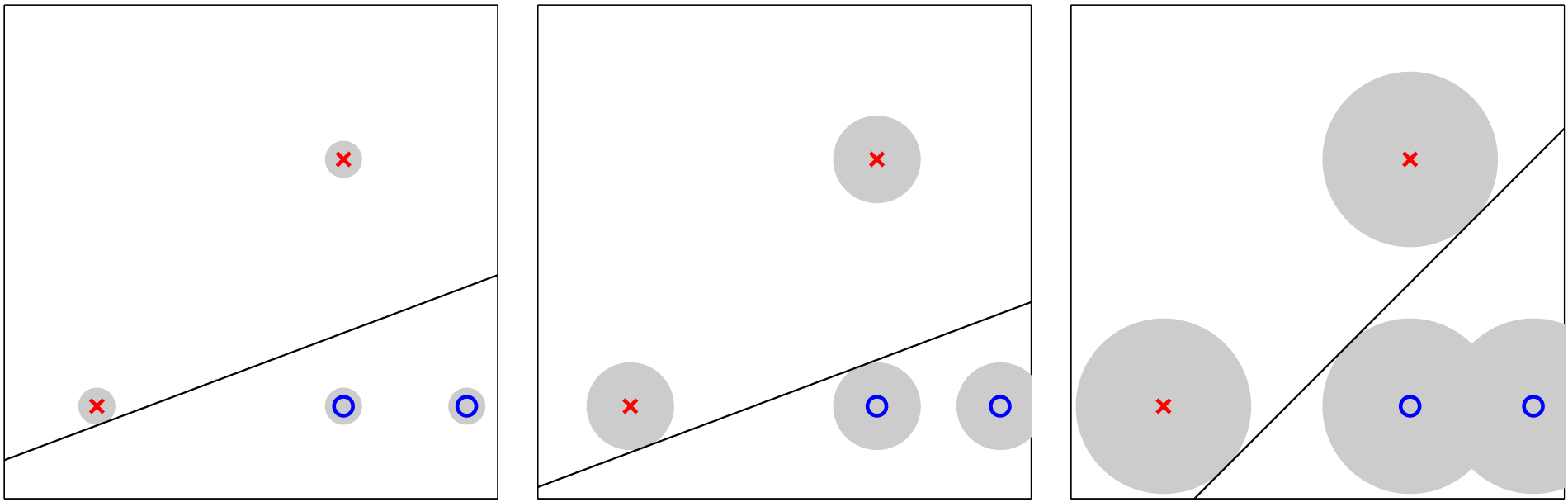
Which Separator Do You Pick?





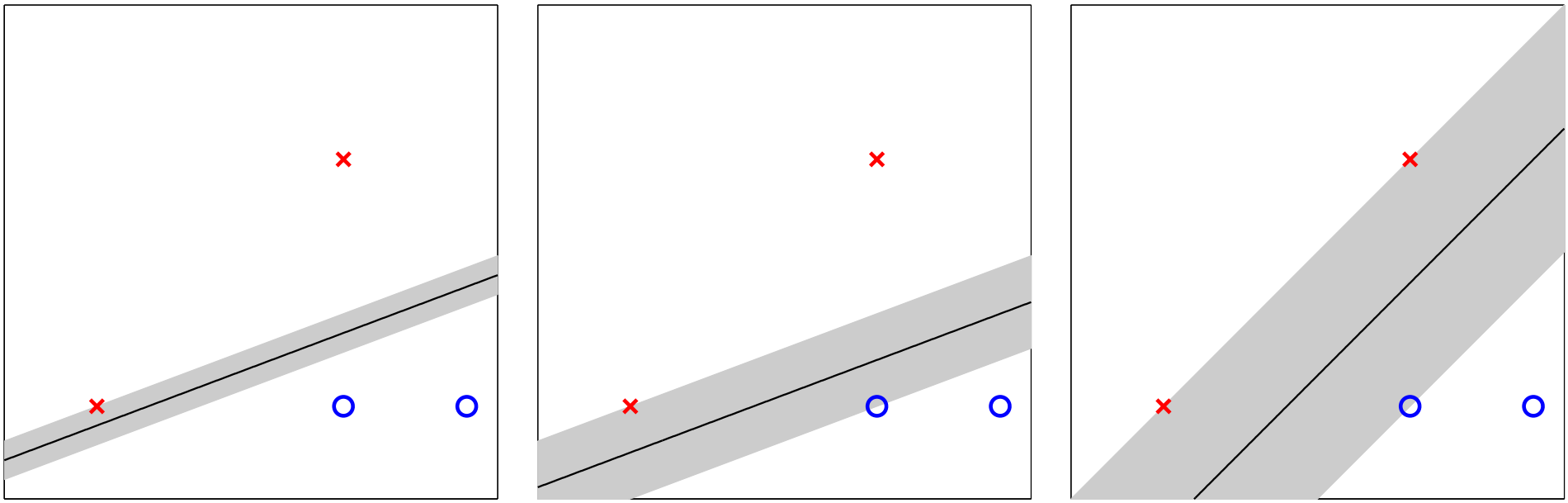


Robustness to Noisy Data



Being robust to noise (measurement error) is good (remember regularization).

Thicker Cushion Means More Robustness



We call such hyperplanes **fat**

Two Crucial Questions

1. Can we efficiently find the fattest separating hyperplane?

2. Is a fatter hyperplane better than a thin one?

Pulling Out the Bias

Before

$\mathbf{x} \in \{1\} \times \mathbb{R}^d; \ \mathbf{w} \in \mathbb{R}^{d+1}$

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}.$$

$\text{signal} = \mathbf{w}^{\mathrm{T}}\mathbf{x}$

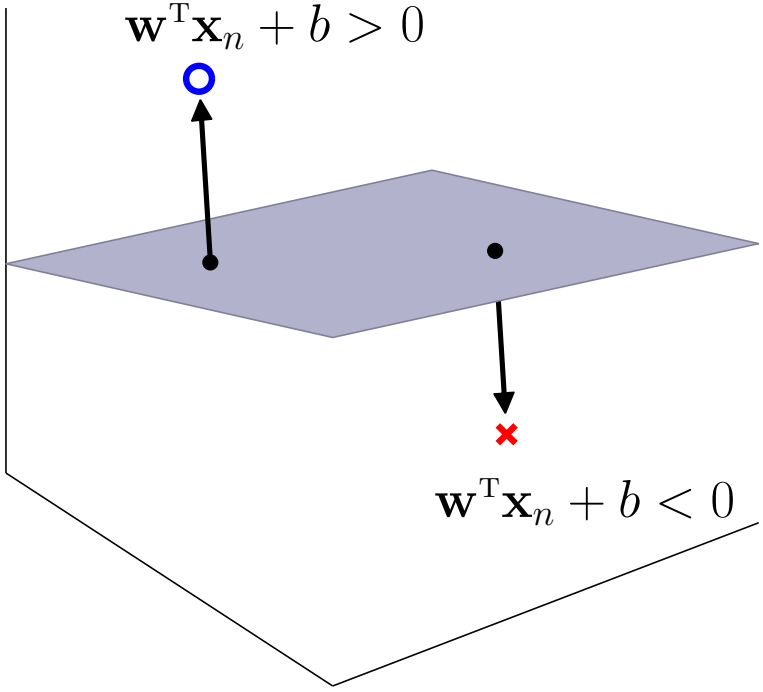
Now

$\mathbf{x} \in \mathbb{R}^d; \ b \in \mathbb{R}, \ \mathbf{w} \in \mathbb{R}^d$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} \text{bias } b \\ w_1 \\ \vdots \\ w_d \end{bmatrix}.$$

$\text{signal} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$

Separating The Data



Hyperplane $h = (b, \mathbf{w})$

h separates the data means:

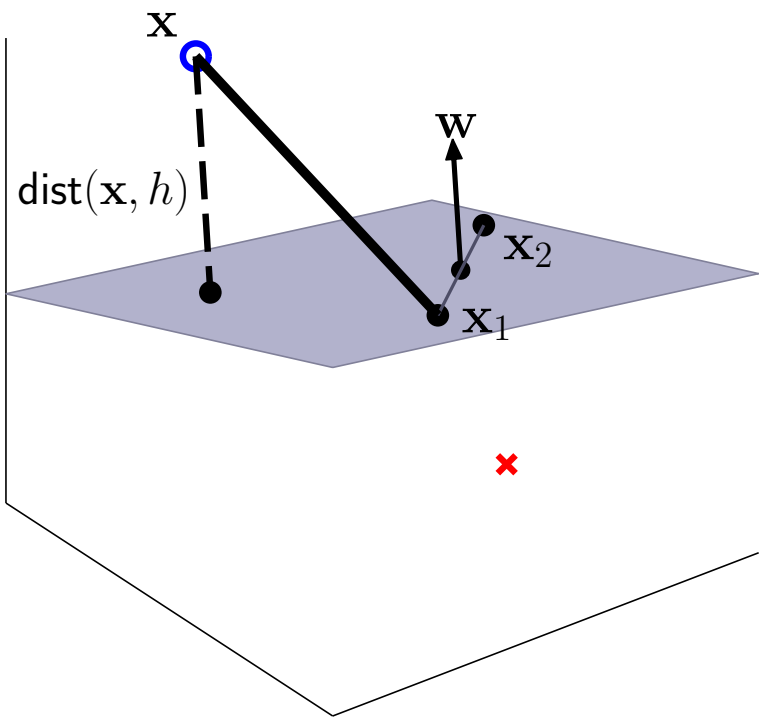
$$y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) > 0$$

By rescaling the weights and bias,

$$\min_{n=1,\ldots,N} \; y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = 1$$

(renormalize the weights so that the signal $\mathbf{w}^{\mathrm{T}}\mathbf{x}+b$ is meaningful)

Distance to the Hyperplane



$\mathbf{w}$ is normal to the hyperplane:

$$\mathbf{w}^{\mathrm{T}}(\mathbf{x}_2 - \mathbf{x}_1) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_2 - \mathbf{w}^{\mathrm{T}}\mathbf{x}_1 = -b + b = 0.$$

(because $\mathbf{w}^{\mathrm{T}}\mathbf{x} = -b$ on the hyperplane)

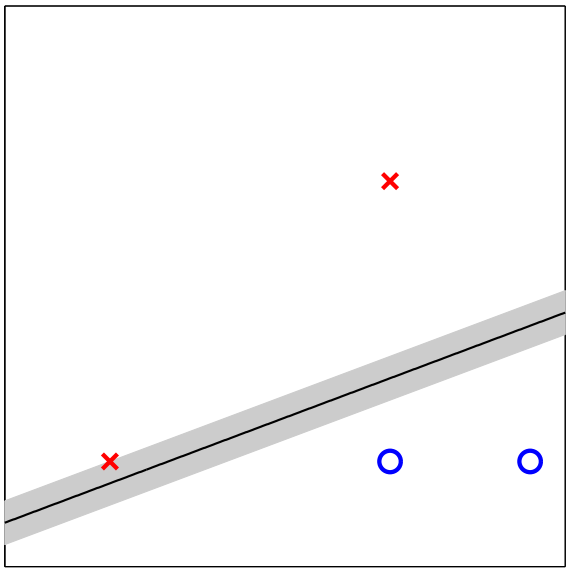
Unit normal $\mathbf{u} = \mathbf{w}/\| \mathbf{w} \|$.

$$\begin{aligned}
\mathsf{dist}(\mathbf{x}, h) &= |\mathbf{u}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}_1)| \\
&= \frac{1}{\| \mathbf{w} \|} \cdot |\mathbf{w}^{\mathrm{T}}\mathbf{x} - \mathbf{w}^{\mathrm{T}}\mathbf{x}_1| \\
&= \frac{1}{\| \mathbf{w} \|} \cdot |\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|
\end{aligned}$$

Fatness of a Separating Hyperplane

$$\mathsf{dist}(\mathbf{x}, h) = \frac{1}{\| \mathbf{w} \|} \cdot |\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|$$

Fatness = Distance to the closest point



Since $|\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b| = |y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)| = y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)$,

$$\mathsf{dist}(\mathbf{x}_n, h) = \frac{1}{\| \mathbf{w} \|} \cdot y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b).$$

$$\begin{aligned}
\text{Fatness} &= \min_n \mathsf{dist}(\mathbf{x}_n, h) \\
&= \frac{1}{\| \mathbf{w} \|} \cdot \min_n y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) && \longleftarrow \text{separation condition} \\
&= \frac{1}{\| \mathbf{w} \|} && \longleftarrow \text{the } \textit{margin } \gamma(h)
\end{aligned}$$

Maximizing the Margin

$$\text{margin} \quad \gamma(h) = \frac{1}{\|\mathbf{w}\|}$$

$\longleftarrow$ bias b does not appear here

$$\begin{aligned} \underset{b,\mathbf{w}}{\text{minimize}} \quad & \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ \text{subject to:} \quad & \min_{n=1,\ldots,N} \; y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = 1. \end{aligned}$$

Maximizing the Margin

$$\text{margin} \quad \gamma(h) = \frac{1}{\|\,\mathbf{w}\,\|}$$

$\longleftarrow$ bias b does not appear here

$$\begin{aligned} &\underset{b,\mathbf{w}}{\text{minimize}} \quad \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &\text{subject to:} \quad \min_{n=1,\ldots,N} \; y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = 1. \end{aligned}$$

$$\begin{aligned} &\underset{b,\mathbf{w}}{\text{minimize}} \quad \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &\text{subject to:} \quad y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \;\text{ for } n = 1, \ldots, N. \end{aligned}$$

Example – Our Toy Data Set

$$\mathrm{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1$$

$$\begin{aligned} -b &\geq 1 \quad (i) \\ -(2w_1 + 2w_2 + b) &\geq 1 \quad (ii) \\ 2w_1 + b &\geq 1 \quad (iii) \\ 3w_1 + b &\geq 1 \quad (iv) \end{aligned}$$

(i) and (iii) gives $w_1 \geq 1$

(ii) and (iii) gives $w_2 \leq -1$

So, $\frac{1}{2}(w_1^2 + w_2^2) \geq 1$ $(b = -1, w_1 = 1, w_2 = -1)$

Optimal Hyperplane

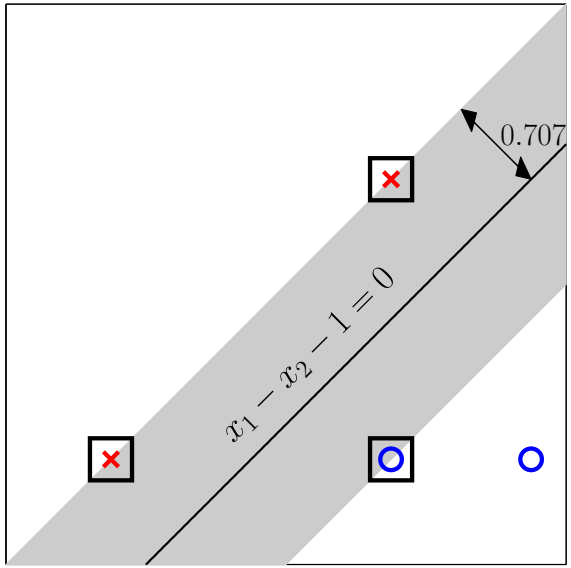
$$g(\mathbf{x}) = \mathrm{sign}(x_1 - x_2 - 1)$$

$$\text{margin:} \quad \frac{1}{\| \mathbf{w}^* \|} = \frac{1}{\sqrt{2}} \approx 0.707.$$

For data points (i), (ii) and (iii) $y_n(\mathbf{w}^{*\mathrm{T}}\mathbf{x}_n + b^*) = 1$

↑

Support Vectors



Quadratic Programming

$$\begin{aligned} \underset{\mathbf{u}\in\mathbb{R}^q}{\text{minimize}} \quad & \tfrac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathrm{T}}\mathbf{u} \\ \text{subject to:} \quad & \mathrm{A}\mathbf{u} \geq \mathbf{c} \end{aligned}$$

$$\mathbf{u}^* \leftarrow \mathsf{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$$

($\mathrm{Q} = 0$ is linear programming)

Maximum Margin Hyperplane is QP

$$\underset{b,\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{subject to: } y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \text{ for } n = 1, \ldots, N.$$

$$\underset{\mathbf{u}\in\mathbb{R}^q}{\text{minimize}} \quad \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{c}^{\mathrm{T}}\mathbf{u}$$

$$\text{subject to: } \mathrm{A}\mathbf{u} \geq \mathbf{a}$$

$$\mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{d+1}$$

$$\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} = \begin{bmatrix} b & \mathbf{w}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix} \begin{bmatrix} b \\ \mathbf{w}^{\mathrm{T}} \end{bmatrix} = \mathbf{u}^{\mathrm{T}} \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix} \mathbf{u} \implies \mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}, \ \mathbf{p} = \mathbf{0}_{d+1}$$

$$y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \equiv \begin{bmatrix} y_n & y_n\mathbf{x}_n^{\mathrm{T}} \end{bmatrix} \mathbf{u} \geq 1 \implies \begin{bmatrix} y_1 & y_1\mathbf{x}_1^{\mathrm{T}} \\ \vdots & \vdots \\ y_N & y_N\mathbf{x}_N^{\mathrm{T}} \end{bmatrix} \mathbf{u} \geq \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \implies \mathrm{A} = \begin{bmatrix} y_1 & y_1\mathbf{x}_1^{\mathrm{T}} \\ \vdots & \vdots \\ y_N & y_N\mathbf{x}_N^{\mathrm{T}} \end{bmatrix}, \ \mathbf{c} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Back To Our Example

Exercise:

$$\mathrm{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$\begin{aligned} & & y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 & \\ -b & \geq 1 & & (i) \\ -(2w_1 + 2w_2 + b) & \geq 1 & & (ii) \\ 2w_1 + b & \geq 1 & & (iii) \\ 3w_1 + b & \geq 1 & & (iv) \end{aligned}$$

Show that

$$\mathrm{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathrm{A} = \begin{bmatrix} -1 & 0 & 0 \\ -1 & -2 & -2 \\ 1 & 2 & 0 \\ 1 & 3 & 0 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Use your QP-solver to give

$$(b^*, w_1^*, w_2^*) = (-1, 1, -1)$$

Primal QP algorithm for linear-SVM

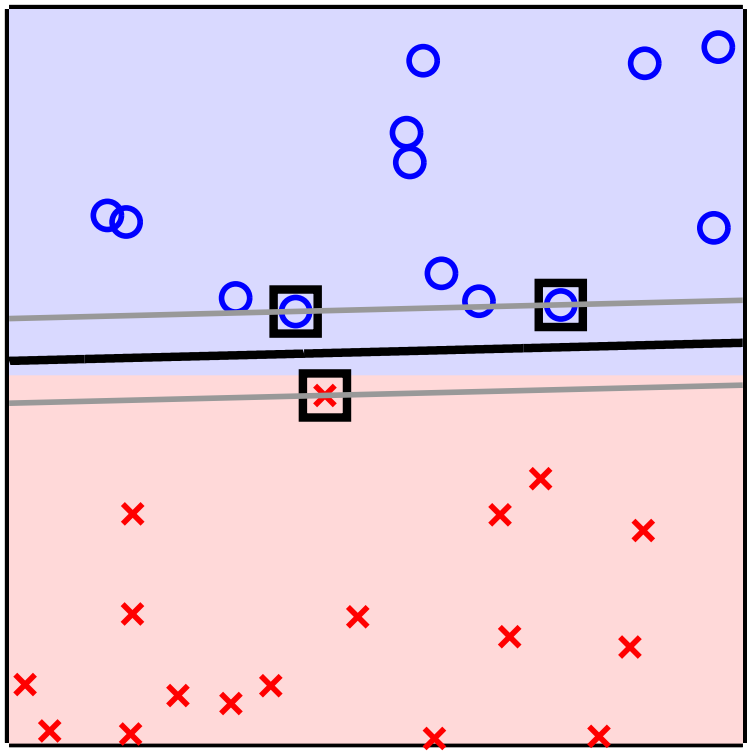
1: Let $\mathbf{p} = \mathbf{0}_{d+1}$ be the $(d+1)$-vector of zeros and $\mathbf{c} = \mathbf{1}_N$ the N-vector of ones. Construct matrices Q and A, where

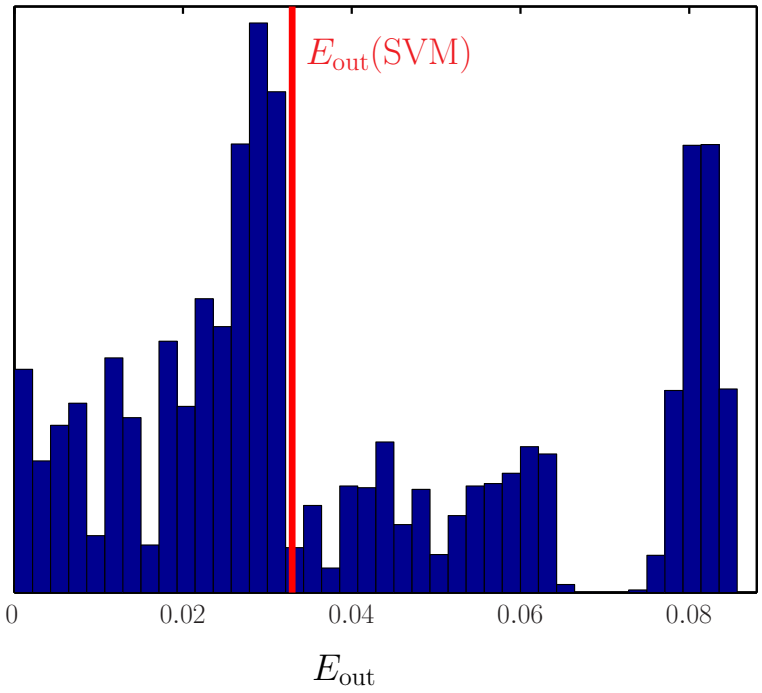
$$\mathrm{A} = \underbrace{\begin{bmatrix} y_1 & -y_1\mathbf{x}_1^{\mathrm{T}}- \\ \vdots & \vdots \\ y_N & -y_N\mathbf{x}_N^{\mathrm{T}}- \end{bmatrix}}_{\text{signed data matrix}}, \qquad \mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}$$

2: Return $\begin{bmatrix} b^* \\ \mathbf{w}^* \end{bmatrix} = \mathbf{u}^* \leftarrow \mathsf{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$.

3: The final hypothesis is $g(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^{*\mathrm{T}}\mathbf{x} + b^*)$.

Example: SVM vs PLA





PLA depends on the ordering of data (e.g. random)

Link to Regularization

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}} \quad E_{\text{in}}(\mathbf{w}) \\ &\text{subject to: } \mathbf{w}^{\text{T}}\mathbf{w} \leq C. \end{aligned}$$

	optimal hyperplane	regularization
minimize	$\mathbf{w}^{\text{T}}\mathbf{w}$	E_{in}
subject to	$E_{\text{in}} = 0$	$\mathbf{w}^{\text{T}}\mathbf{w} \leq C$