Support Vector Machines

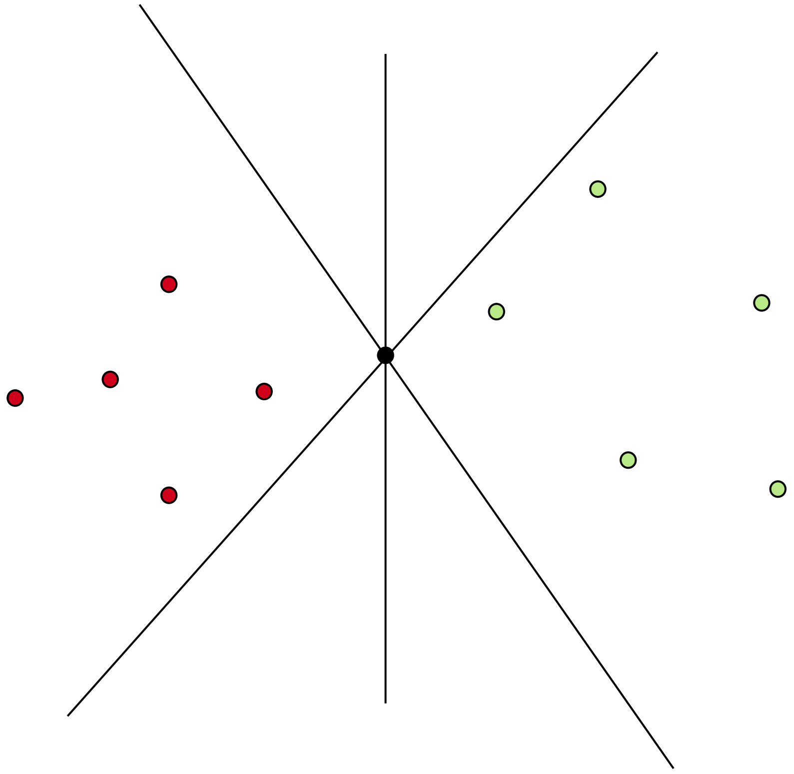
Agenda

- Hard-margin SVM
  - theory (brief)
  - implementation
- Soft-margin SVM
  - theory (brief)
  - implementation
- Closing remarks

https://tiny.cc/mlt_workshop

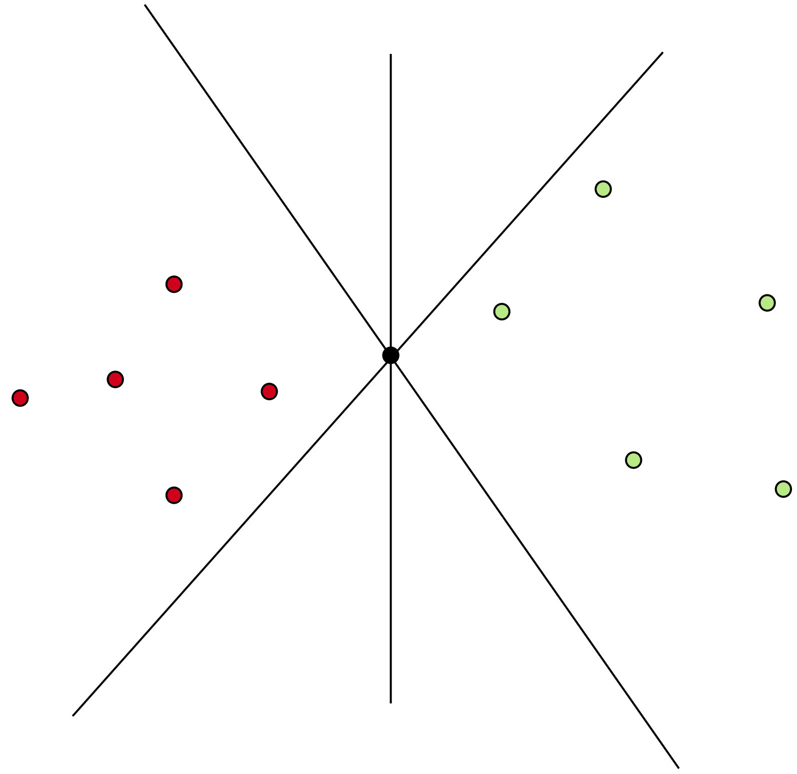
- Colabs_Slides_Day_3
  - SVM.ipynb
  - Make a copy
- Code along

From Perceptrons to SVM



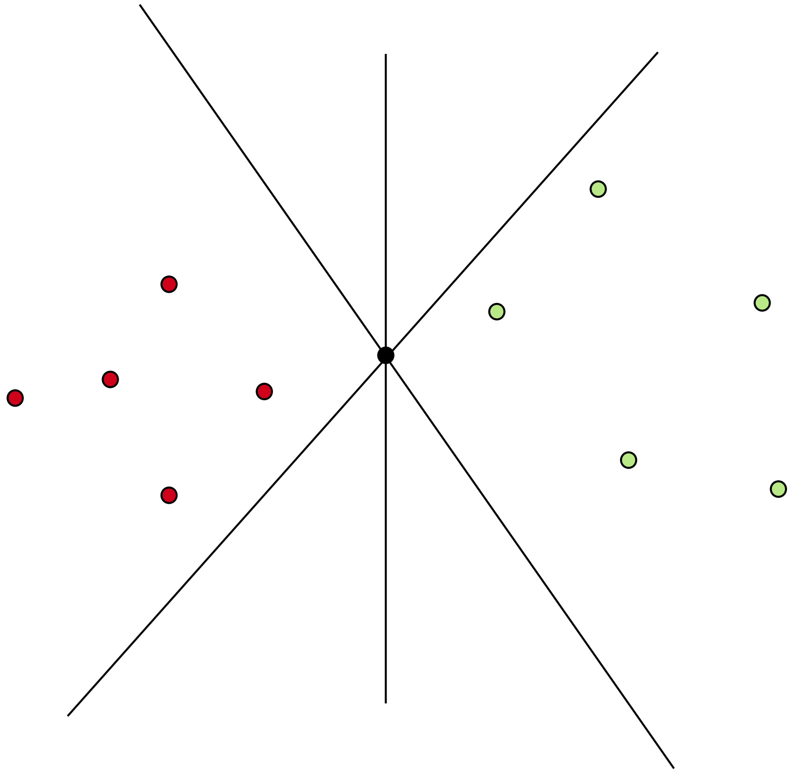
For linearly separable data with γ margin:

From Perceptrons to SVM



For linearly separable data with γ margin:
(1) Infinite number of valid linear classifiers.

From Perceptrons to SVM

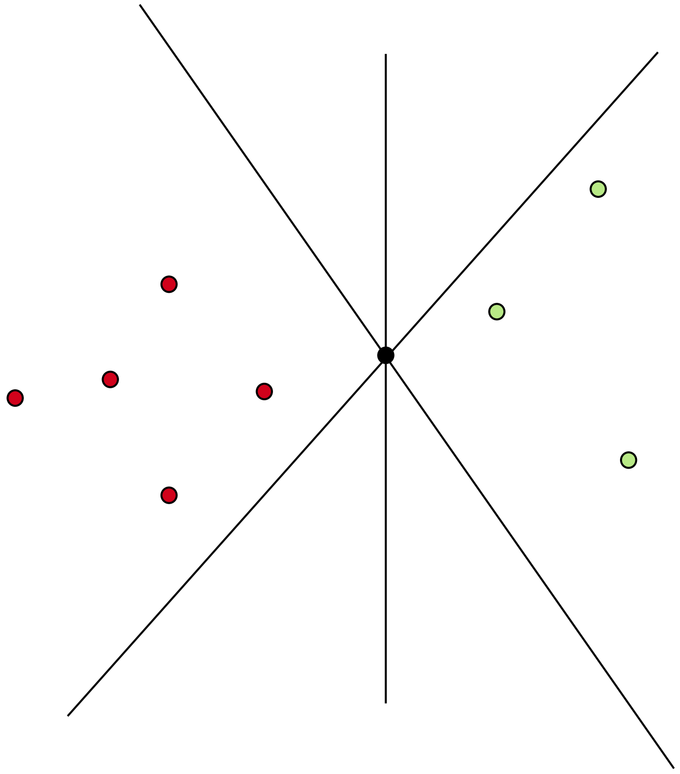


For linearly separable data with γ margin:

(1) Infinite number of valid linear classifiers.

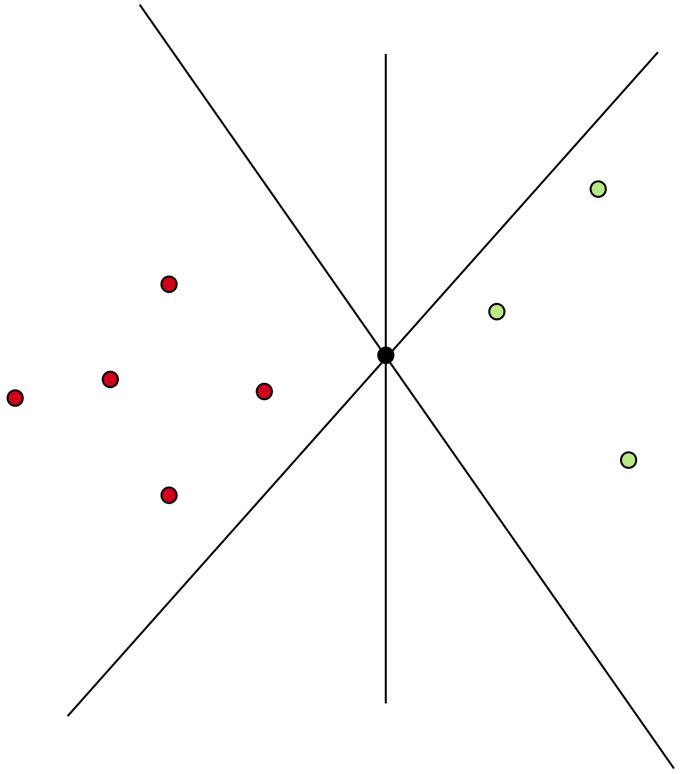
(2) Perceptron returns a valid linear classifier.

From Perceptrons to SVM



For linearly separable data with γ margin:
(1) Infinite number of valid linear classifiers.
(2) Perceptron returns a valid linear classifier.
(3) Is it the "best"?

From Perceptrons to SVM



For linearly separable data with γ margin:

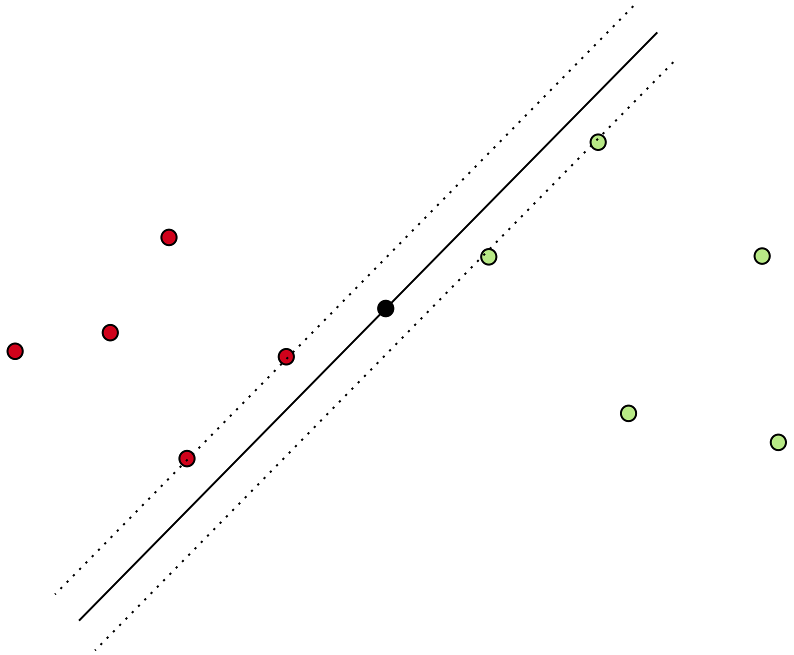
(1) Infinite number of valid linear classifiers.

(2) Perceptron returns a valid linear classifier.

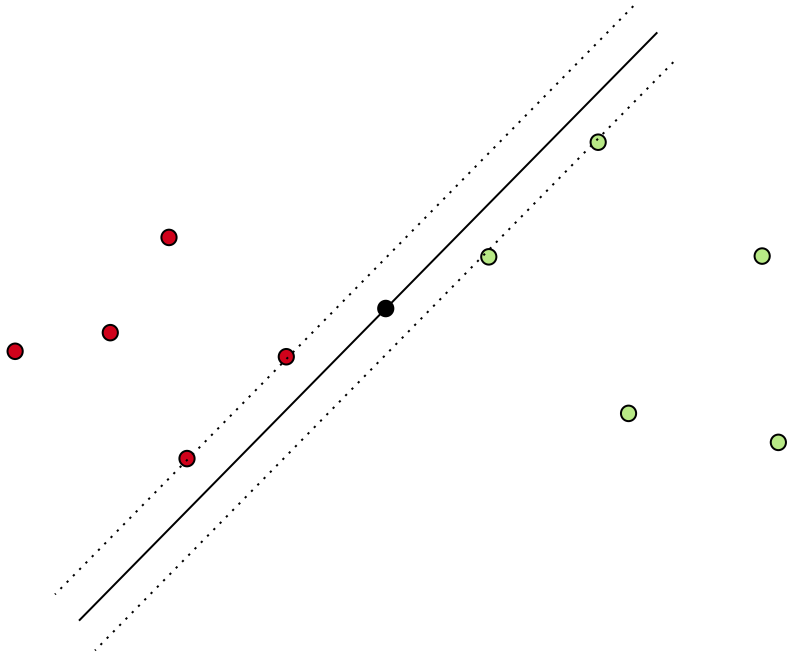
(3) Is it the "best"?

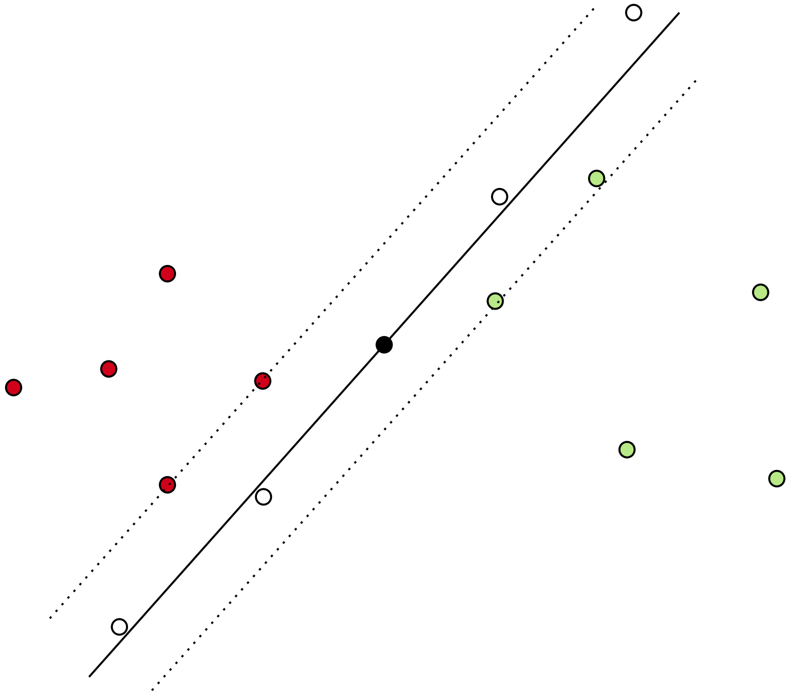
(4) What is a good notion of "best"?

Margin



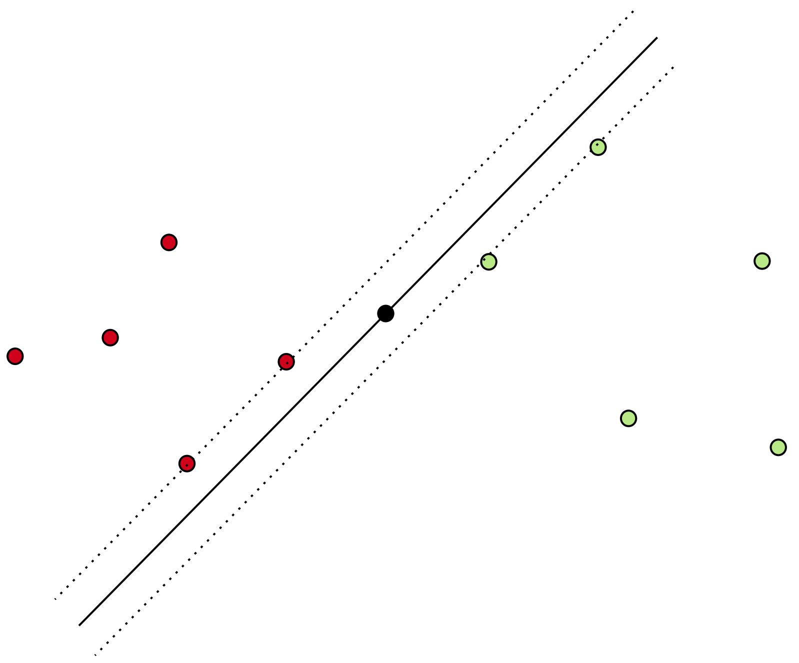
Margin

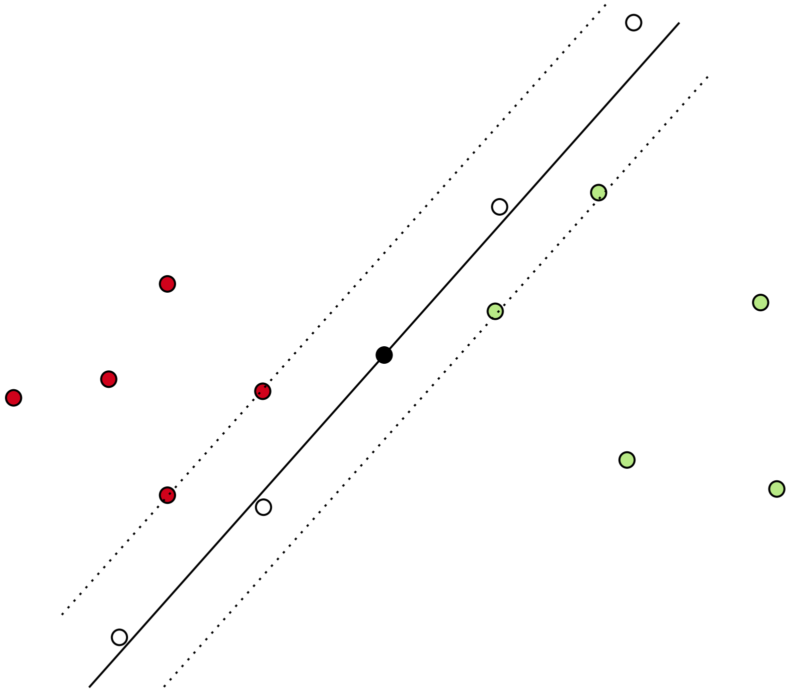




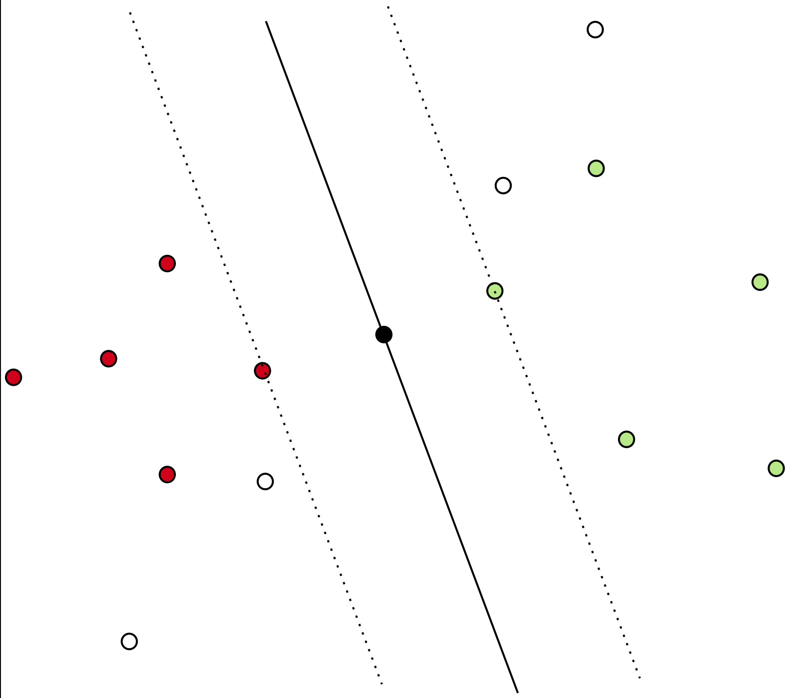
Small margin
Doesn't generalize well

Margin



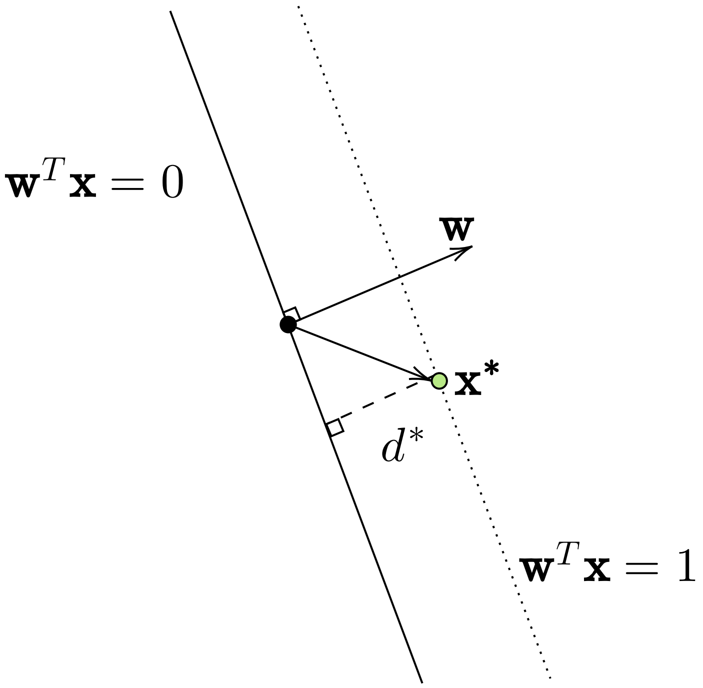


Small margin
Doesn't generalize well



Large margin
Better generalization

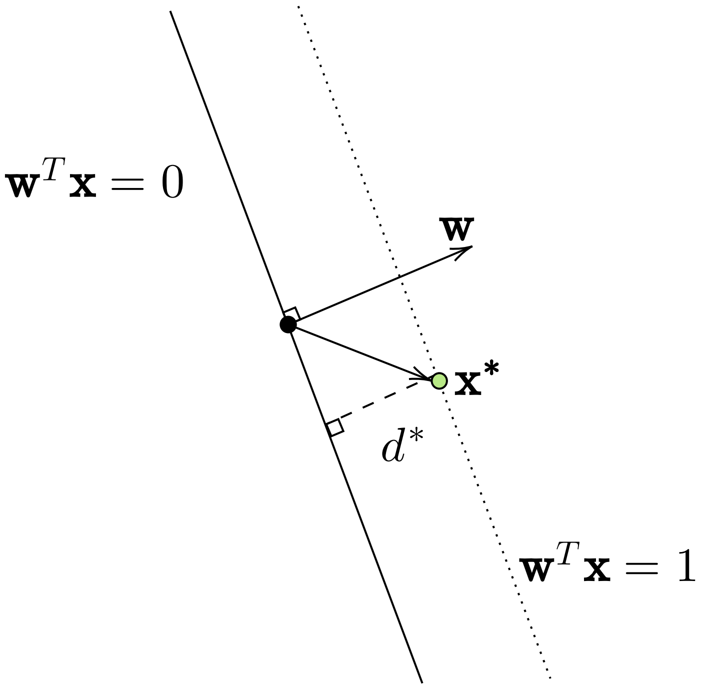
Computing the Margin



$$d^* = \frac{1}{||\mathbf{w}||}$$

For any linear classifier represented by **w**:

Computing the Margin

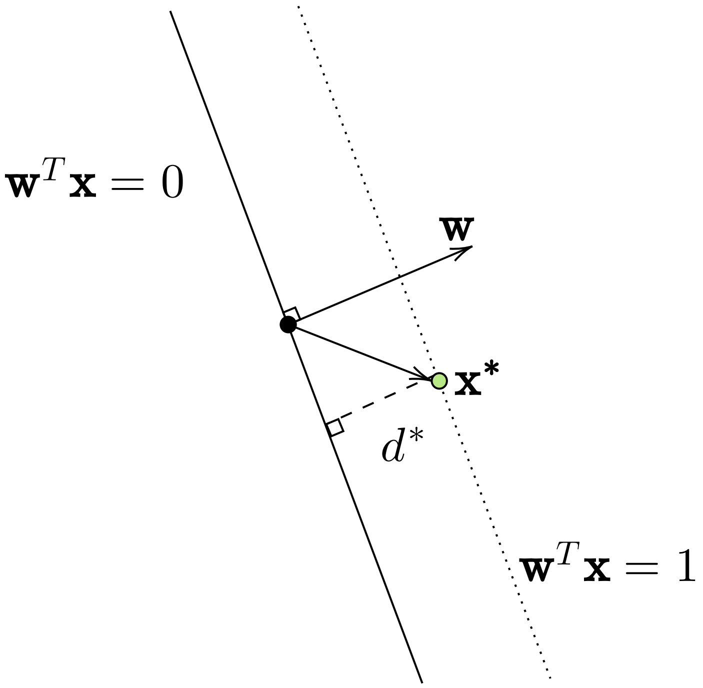


$$d^* = \frac{1}{||\mathbf{w}||}$$

For any linear classifier represented by **w**:

(1) Find the point closest to it → **x***

Computing the Margin



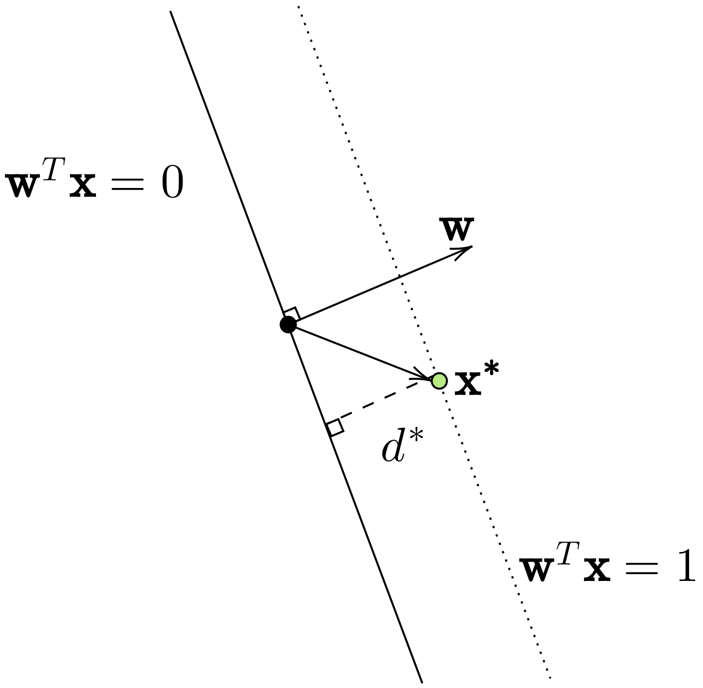
$$d^* = \frac{1}{||\mathbf{w}||}$$

For any linear classifier represented by $\mathbf{w}$:

(1) Find the point closest to it → $\mathbf{x^*}$

(2) Scale $\mathbf{w}$ such that $\mathbf{x^*}$ lies on $\mathbf{w}^T\mathbf{x} = 1$

Computing the Margin



$$d^* = \frac{1}{||\mathbf{w}||}$$

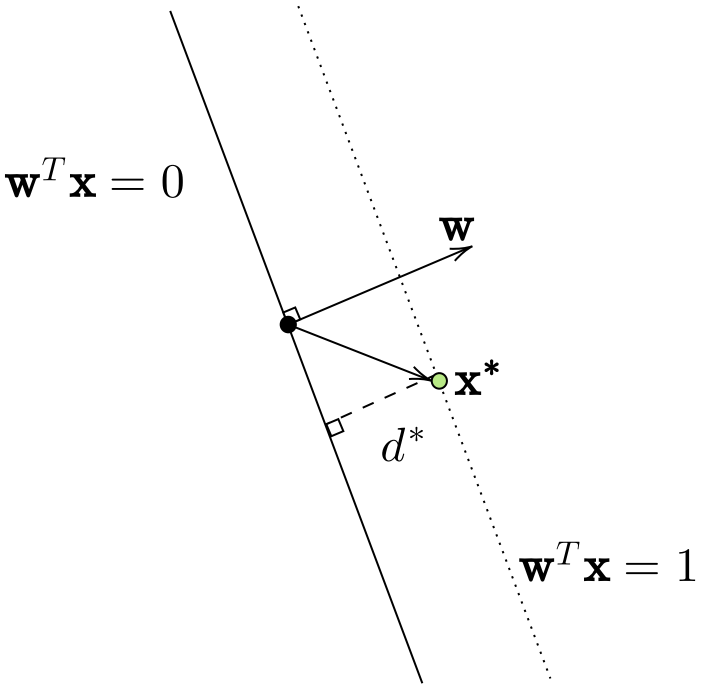
For any linear classifier represented by $\mathbf{w}$:

(1) Find the point closest to it → $\mathbf{x^*}$

(2) Scale $\mathbf{w}$ such that $\mathbf{x^*}$ lies on $\mathbf{w}^T\mathbf{x} = 1$

(3) Distance of $\mathbf{x^*}$ from the line is $\frac{1}{||\mathbf{w}||}$

Computing the Margin



$$d^* = \frac{1}{||\mathbf{w}||}$$

For any linear classifier represented by $\mathbf{w}$:

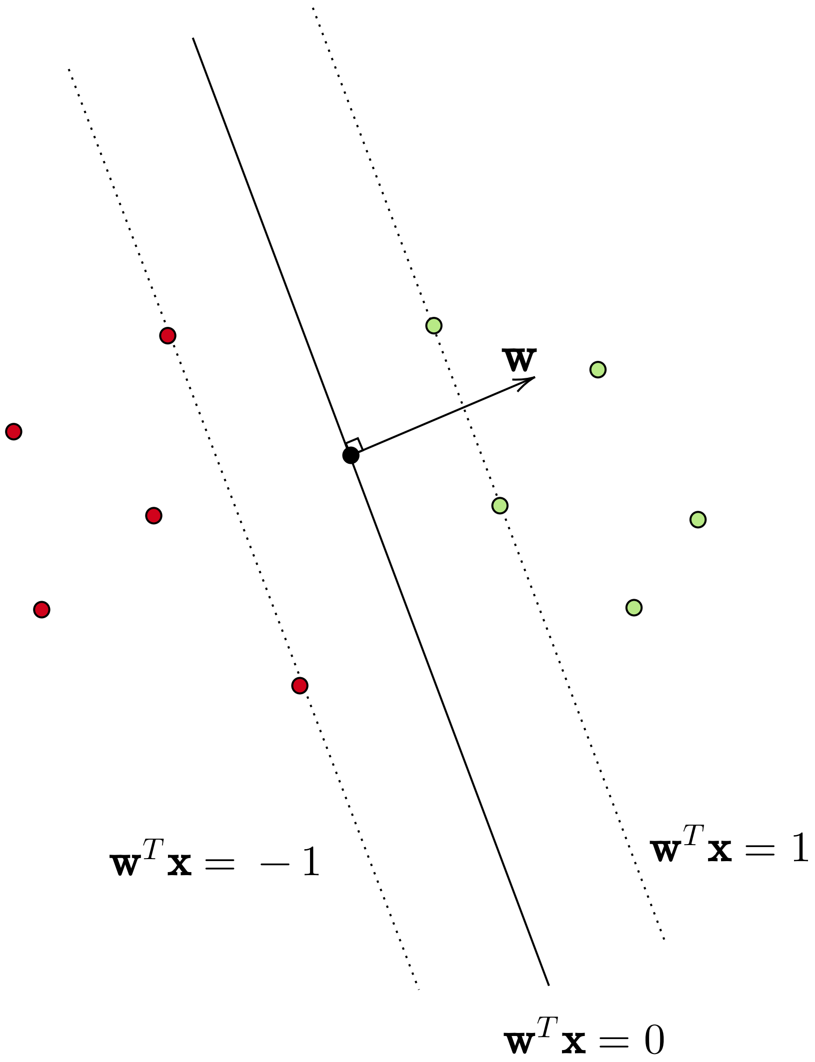
(1) Find the point closest to it → $\mathbf{x}^*$

(2) Scale $\mathbf{w}$ such that $\mathbf{x}^*$ lies on $\mathbf{w}^T\mathbf{x} = 1$

(3) Distance of $\mathbf{x}^*$ from the line is $\frac{1}{||\mathbf{w}||}$

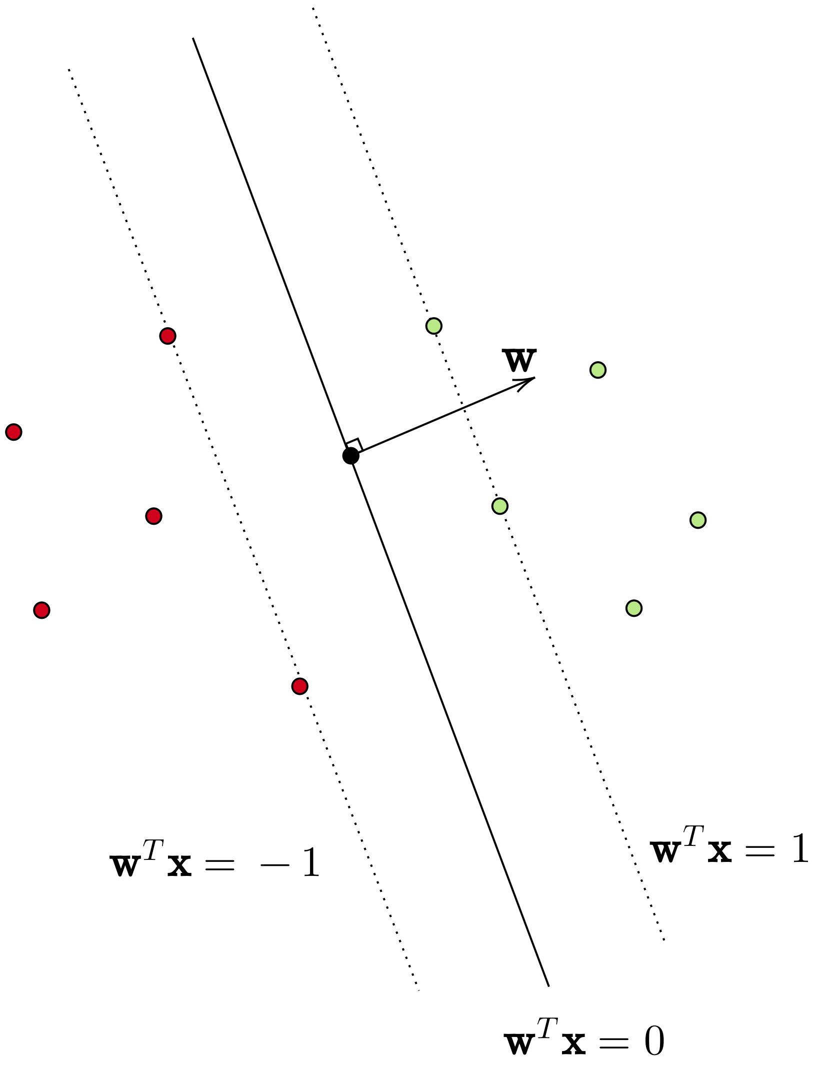
(4) This is the (geometric) margin for this linear classifier.

Beyond the "margin"



$$\left(\mathbf{w}^T\mathbf{x}_i\right)y_i \geq 1, \quad 1 \leq i \leq n$$

Max-Margin Classifier

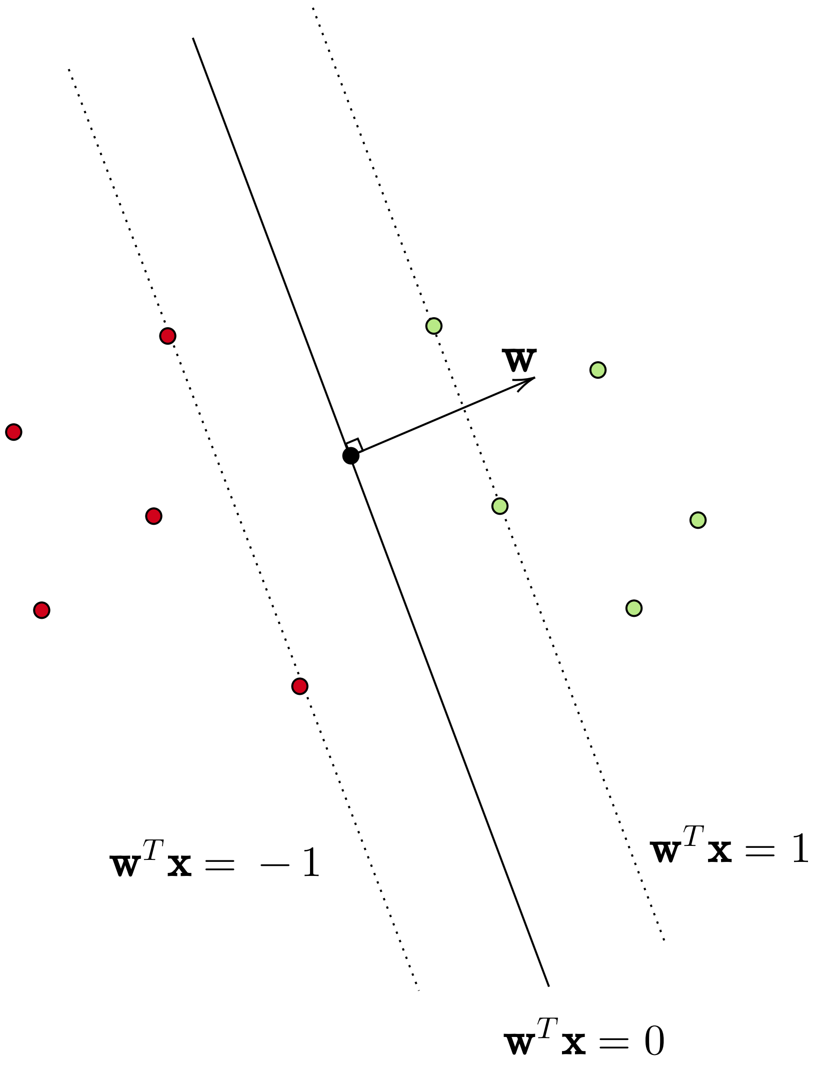


$$\max_{\mathbf{w}} \quad \frac{1}{||\mathbf{w}||}$$

sub. to

$$\left(\mathbf{w}^T\mathbf{x}_i\right)y_i \geq 1, \quad 1 \leq i \leq n$$

Max-Margin Classifier



$$\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2}$$

sub. to

$$\left(\mathbf{w}^T\mathbf{x}_i\right)y_i \geq 1, \quad 1 \leq i \leq n$$

Primal and Dual

$$\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2}$$

sub. to

$$\left(\mathbf{w}^T \mathbf{x}_i\right) y_i \geq 1, \quad 1 \leq i \leq n$$

$$\equiv \quad \min_{\mathbf{w}} \max_{\boldsymbol{\alpha} \geq 0} \quad \frac{||\mathbf{w}||^2}{2} + \sum_{i=1}^{n} \alpha_i \Big[1 - \left(\mathbf{w}^T \mathbf{x}_i\right) y_i\Big] \quad \equiv \quad \max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + \sum_{i=1}^{n} \alpha_i \Big[1 - \left(\mathbf{w}^T \mathbf{x}_i\right) y_i\Big]$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}$$

Formulating the Dual

$$\max_{\boldsymbol{\alpha}\geq 0} \boxed{\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + \sum_{i=1}^{n} \alpha_i \left[1 - \left(\mathbf{w}^T \mathbf{x}_i\right) y_i\right]}$$

$$\boxed{\mathbf{w} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i y_i}$$

Formulating the Dual

$$\max_{\boldsymbol{\alpha}\geq 0} \boxed{\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + \sum_{i=1}^{n} \alpha_i \Big[1 - \big(\mathbf{w}^T \mathbf{x}_i\big) y_i\Big]}$$

$$\boxed{\mathbf{w} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i y_i}$$

$$d \times n$$

$$\mathbf{X} = \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_n \\ | & & | \end{bmatrix}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}$$

$$n \times n$$

$$\mathbf{Y} = \begin{bmatrix} y_1 & & 0 \\ & \ddots & \\ 0 & & y_n \end{bmatrix}$$

$$\mathbf{XY}\boldsymbol{\alpha} = \begin{bmatrix} | & & | \\ y_1\mathbf{x}_1 & \cdots & y_n\mathbf{x}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \sum_{i=1}^{n} \alpha_i (y_i \mathbf{x}_i) = \mathbf{w}$$

Formulating the Dual

$$\max_{\boldsymbol{\alpha}\geq 0} \boxed{\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + \sum_{i=1}^{n} \alpha_i \left[1 - \left(\mathbf{w}^T \mathbf{x}_i\right) y_i\right]}$$

$$\boxed{\mathbf{w} = \mathbf{X}\mathbf{Y}\boldsymbol{\alpha}}$$

$d \times n$

$$\mathbf{X} = \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_n \\ | & & | \end{bmatrix}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}$$

$n \times n$

$$\mathbf{Y} = \begin{bmatrix} y_1 & & 0 \\ & \ddots & \\ 0 & & y_n \end{bmatrix}$$

$$\boxed{\max_{\boldsymbol{\alpha}\geq 0} \ \boldsymbol{\alpha}^T \mathbf{1} - \frac{\boldsymbol{\alpha}^T \left(\mathbf{Y}^T \mathbf{X}^T \mathbf{X}\mathbf{Y}\right) \boldsymbol{\alpha}}{2}}$$

Support Vectors

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \alpha_1^* \\ \vdots \\ \alpha_n^* \end{bmatrix} \qquad \mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* \mathbf{x}_i y_i$$

Support Vectors

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \alpha_1^* \\ \vdots \\ \alpha_n^* \end{bmatrix} \qquad \mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* \mathbf{x}_i y_i$$

Definition: A support vector is a point for which $\alpha_i^* > 0$

Support Vectors

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \alpha_1^* \\ \vdots \\ \alpha_n^* \end{bmatrix}$$

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* \mathbf{x}_i y_i$$

Definition: A support vector is a point for which $\alpha_i^* > 0$

$$\alpha_i^* g_i(\mathbf{w^*}) = 0 \Longrightarrow \alpha_i^* \left[1 - \left((\mathbf{w}^*)^T \mathbf{x}_i \right) y_i \right] = 0$$

Complementary Slackness

Support Vectors

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \alpha_1^* \\ \vdots \\ \alpha_n^* \end{bmatrix} \qquad \mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* \mathbf{x}_i y_i$$

Definition: A support vector is a point for which $\alpha_i^* > 0$

$$\alpha_i^* g_i(\mathbf{w}^*) = 0 \Longrightarrow \alpha_i^* \left[1 - \left((\mathbf{w}^*)^T \mathbf{x}_i\right) y_i\right] = 0$$

Complementary Slackness

Every support vector lies on one of the two supporting hyperplanes $(\mathbf{w}^*)^T \mathbf{x} = \pm 1$

Support Vectors

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \alpha_1^* \\ \vdots \\ \alpha_n^* \end{bmatrix} \qquad \mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* \mathbf{x}_i y_i$$

Definition: A support vector is a point for which $\alpha_i^* > 0$

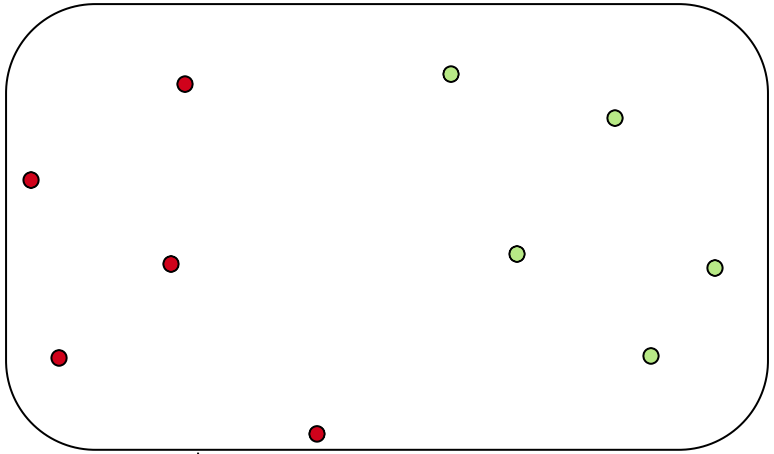
$$\alpha_i^* g_i\left(\mathbf{w}^*\right) = 0 \Longrightarrow \alpha_i^* \left[1 - \left(\left(\mathbf{w}^*\right)^T \mathbf{x}_i\right) y_i\right] = 0$$

Complementary Slackness

Every support vector lies on one of the two supporting hyperplanes $\left(\mathbf{w}^*\right)^T \mathbf{x} = \pm 1$

Every point that is **not** on one of the two supporting hyperplanes has $\alpha_i^* = 0$.

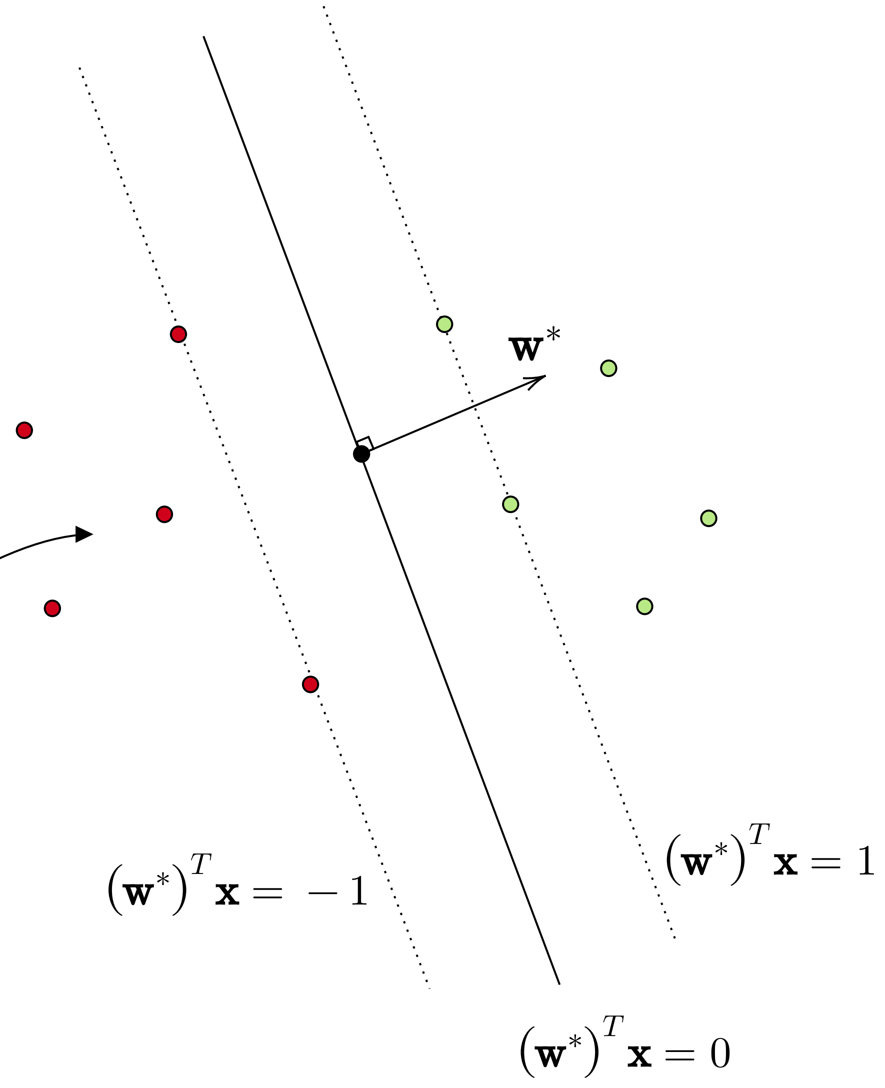
Hard-Margin, Linear-SVM



SVM Solver

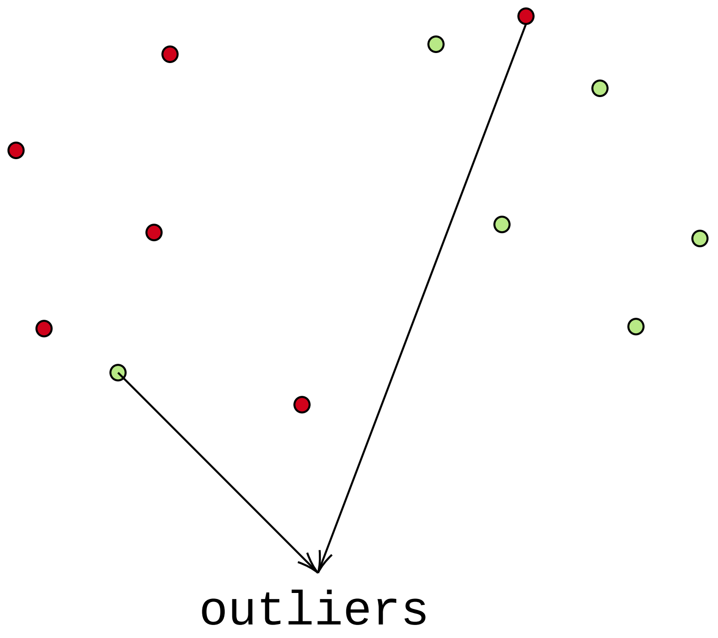
$$\boldsymbol{\alpha}^* = \begin{bmatrix} \alpha_1^* \\ \vdots \\ \alpha_n^* \end{bmatrix}$$

$$\widehat{y} = \begin{cases} 1, & (\mathbf{w}^*)^T \mathbf{x} \geqslant 0 \\ -1, & (\mathbf{w}^*)^T \mathbf{x} < 0 \end{cases}$$



$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* \mathbf{x}_i y_i$$

Soft-Margin, SVM

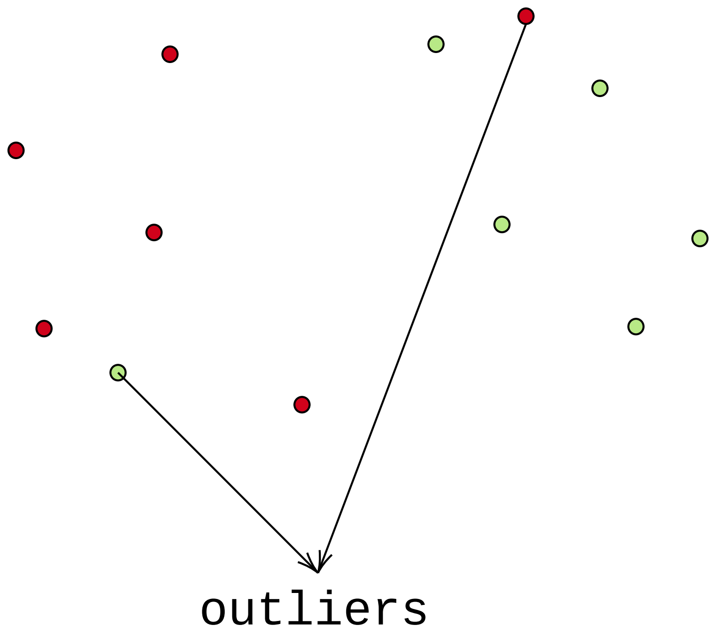


Not linearly separable

(1) Structural → Hard-margin, Kernel-SVM

(2) Statistical (outliers)

Soft-Margin, SVM



Not linearly separable

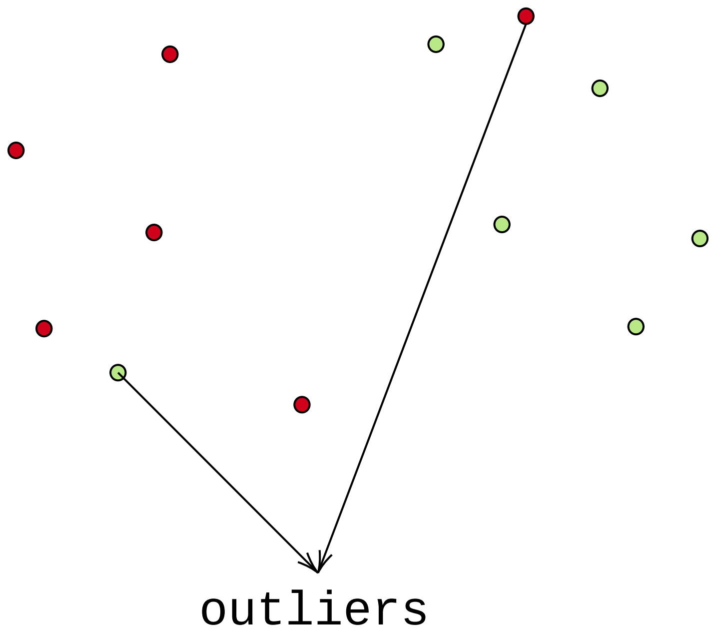
(1) Structural → Hard-margin, Kernel-SVM

(2) Statistical (outliers)

- Ideally, we want $\left(\mathbf{w}^T\mathbf{x}_i\right)y_i \geqslant 1$.
- Not true for outliers.
- Use a non-negative bribe to push them

$$\left(\mathbf{w}^T\mathbf{x}_i\right)y_i + \xi_i \geqslant 1$$

Soft-Margin, SVM



Not linearly separable

(1) Structural → Hard-margin, Kernel-SVM

(2) Statistical (outliers)

- Ideally, we want $\left(\mathbf{w}^T\mathbf{x}_i\right)y_i \geqslant 1$.
- Not true for outliers.
- Use a non-negative bribe to push them

$$\left(\mathbf{w}^T\mathbf{x}_i\right)y_i + \xi_i \geqslant 1$$

$$\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + C \cdot \sum_{i=1}^{n} \xi_i$$

sub. to

$$\left(\mathbf{w}^T\mathbf{x}_i\right)y_i + \xi_i \geq 1, \quad 1 \leq i \leq n$$

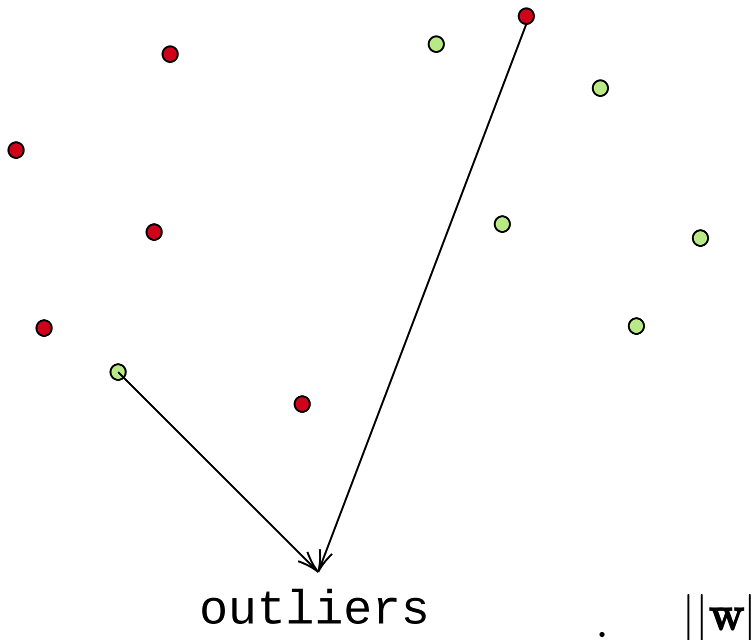
$$\xi_i \geq 0, \quad 1 \leq i \leq n$$

Soft-Margin, SVM

Not linearly separable

(1) Structural → Hard-margin, Kernel-SVM

(2) Statistical (outliers)



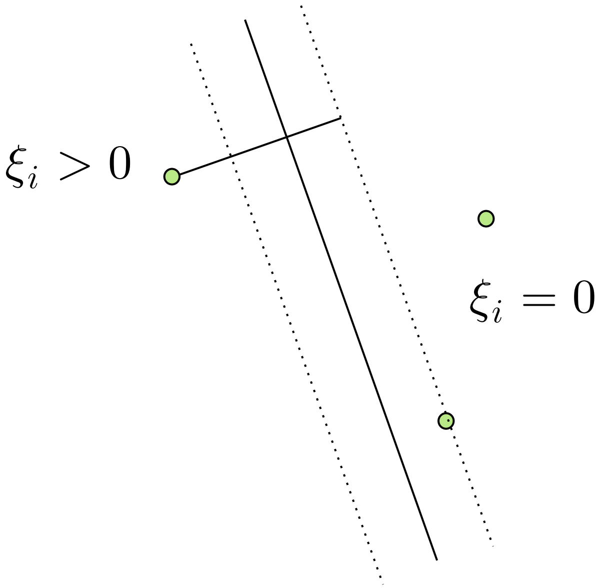
- Ideally, we want $\big(\mathbf{w}^T\mathbf{x}_i\big)y_i \geqslant 1$.
- Not true for outliers.
- Use a non-negative bribe to push them

$$\big(\mathbf{w}^T\mathbf{x}_i\big)y_i + \xi_i \geqslant 1$$

$$\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + C \cdot \sum_{i=1}^{n} \xi_i$$

sub. to

$$\begin{aligned} \big(\mathbf{w}^T\mathbf{x}_i\big)y_i + \xi_i \geq 1, &\quad 1 \leq i \leq n \\ \xi_i \geq 0, &\quad 1 \leq i \leq n \end{aligned}$$



Dual for Soft-SVM

$$\max_{0 \leqslant \boldsymbol{\alpha} \leqslant C} \boldsymbol{\alpha}^T \mathbf{1} - \frac{\boldsymbol{\alpha}^T \left(\mathbf{Y}^T \mathbf{X}^T \mathbf{X} \mathbf{Y}\right) \boldsymbol{\alpha}}{2}$$

- Only change is the constraints
- Earlier there was only a lower-bound
- Now, we have an upper-bound
- These are called box-constraints

Soft-Margin, SVM: Hinge-loss formulation

$$\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + C \cdot \sum_{i=1}^{n} \xi_i$$

sub. to

$$\left(\mathbf{w}^T \mathbf{x}_i\right) y_i + \xi_i \geq 1, \quad 1 \leq i \leq n$$

$$\xi_i \geq 0, \quad 1 \leq i \leq n$$

$$\equiv$$

$$\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + C \cdot \sum_{i=1}^{n} \max\Big(0,\ 1 - \left(\mathbf{w}^T \mathbf{x}_i\right) y_i\Big)$$

Regularization Hinge Loss

Soft-Margin, SVM: Hinge-loss formulation

$$\min_{\mathbf{w}} \quad \underbrace{\frac{||\mathbf{w}||^2}{2}}_{(1)} + C \cdot \underbrace{\sum_{i=1}^{n} \max\Big(0,\ 1 - \big(\mathbf{w}^T \mathbf{x}_i\big) y_i\Big)}_{(2)}$$

Terms in the objective:

(1) $\frac{||\mathbf{w}||^2}{2}$ controls the width of the margin

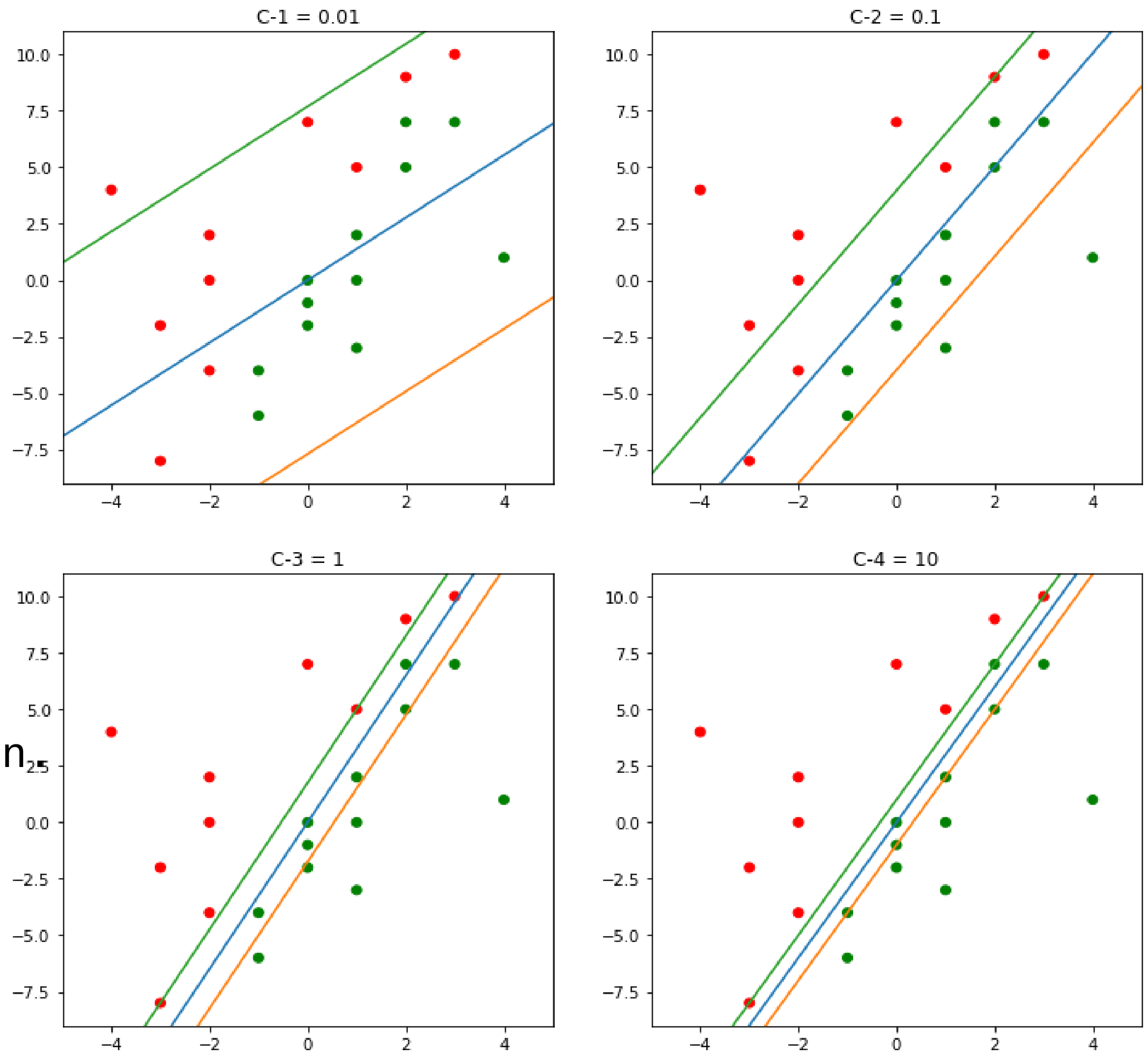
Smaller the value of $||\mathbf{w}||$, wider the margin

(2) $\sum_{i=1}^{n} \max\Big(0,\ 1 - \big(\mathbf{w}^T \mathbf{x}_i\big) y_i\Big)$ is the hinge-loss. Wider the margin, larger the loss.

Soft-Margin, SVM: Hinge-loss formulation

$$\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + C \cdot \sum_{i=1}^{n} \max\Big(0,\ 1 - \big(\mathbf{w}^T \mathbf{x}_i\big) y_i\Big)$$

(1) (2)



- (1) and (2) work in opposite directions
- If $||\mathbf{w}||$ decreases, the margin becomes wider, which increases the hinge-loss.
- C controls the tradeoff between (1) and (2):
  - If C is small, we are fine with a wide margin.
  - If C is large, we prefer a narrow margin.
  - If $C \to \infty$, we do not tolerate bribery at all.

Closing Remarks

- Everything cannot remain a black-box.

Closing Remarks

- Everything cannot remain a black-box.
- Sometimes it is necessary to peep into the box.

Closing Remarks

- Everything cannot remain a black-box.
- Sometimes it is necessary to peep into the box.
- Sometimes it is necessary to build your own box.

Closing Remarks

- Everything cannot remain a black-box.
- Sometimes it is necessary to peep into the box.
- Sometimes it is necessary to build your own box.
- But we must know when to stop:

Closing Remarks

- Everything cannot remain a black-box.
- Sometimes it is necessary to peep into the box.
- Sometimes it is necessary to build your own box.
- But we must know when to stop:
  - Model

Closing Remarks

- Everything cannot remain a black-box.
- Sometimes it is necessary to peep into the box.
- Sometimes it is necessary to build your own box.
- But we must know when to stop:
  - Model
  - NumPy over sklearn

Closing Remarks

- Everything cannot remain a black-box.
- Sometimes it is necessary to peep into the box.
- Sometimes it is necessary to build your own box.
- But we must know when to stop:
  - Model
  - NumPy over sklearn
  - Native Python over NumPy

Closing Remarks

- Everything cannot remain a black-box.
- Sometimes it is necessary to peep into the box.
- Sometimes it is necessary to build your own box.
- But we must know when to stop:
  - Model
  - NumPy over sklearn
  - Native Python over NumPy
  - C over Python

Closing Remarks

- Everything cannot remain a black-box.
- Sometimes it is necessary to peep into the box.
- Sometimes it is necessary to build your own box.
- But we must know when to stop:
  - Model
  - NumPy over sklearn
  - Native Python over NumPy
  - C over Python
  - Assembly language over C

Closing Remarks

- Everything cannot remain a black-box.
- Sometimes it is necessary to peep into the box.
- Sometimes it is necessary to build your own box.
- But we must know when to stop:
  - Model
  - NumPy over sklearn
  - Native Python over NumPy
  - C over Python
  - Assembly language over C
  - Build a computer
  - ...