Support Vector Machines

Lecturer:
Jiří Matas

Authors:
Ondřej Drbohlav, Jiří Matas

Centre for Machine Perception
Czech Technical University, Prague
http://cmp.felk.cvut.cz

Slide credits:
Alexander Apartsin

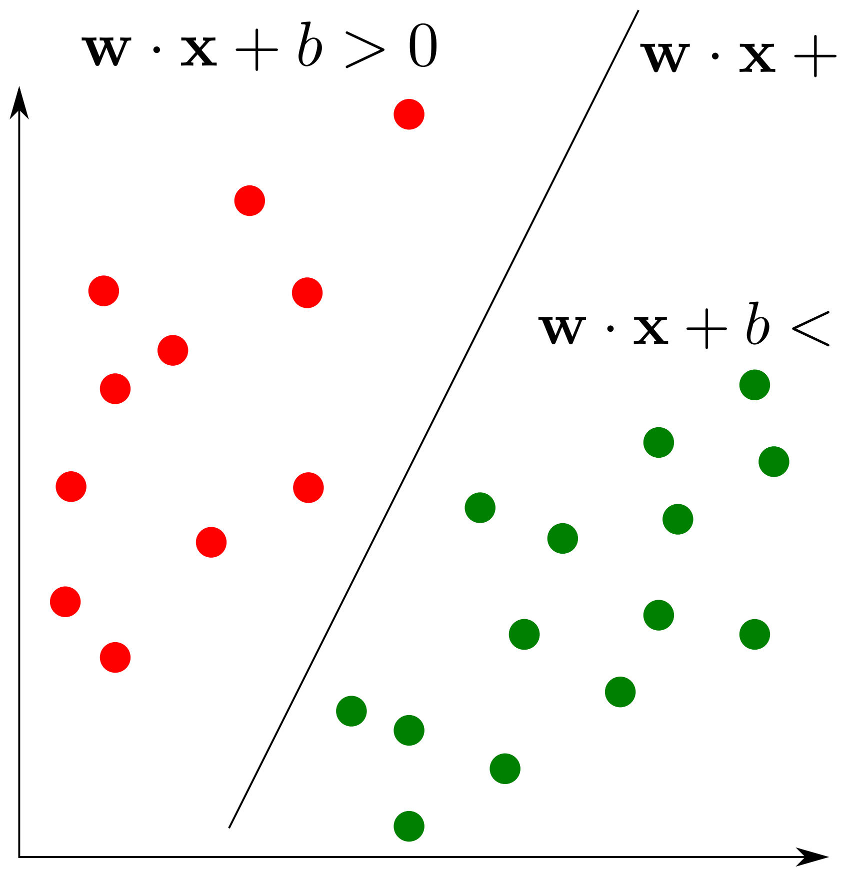
12.11.2018

A Linear Classifier

Classification according to signum of an affine function of $\mathbf{x}$:

$$q(\mathbf{x}) = \mathsf{sign}(\mathbf{w} \cdot \mathbf{x} + b) \tag{1}$$

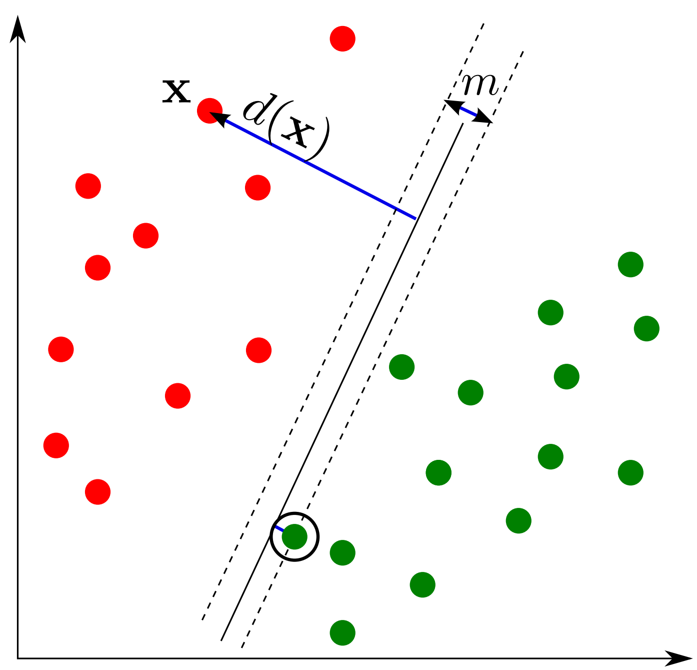
A solution for $\{\mathbf{w}, b\}$ correctly classifying the training set:



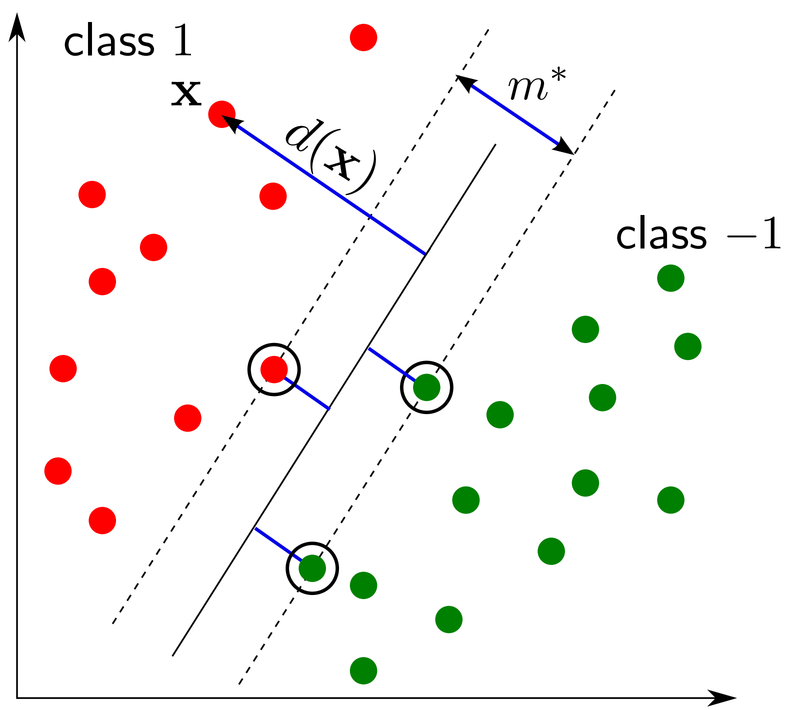
Maximum Margin Linear Classifier

- Let $d(\mathbf{x})$ denote the distance of a point $\mathbf{x} \in \mathcal{T}$ from the training set $\mathcal{T}$ to the decision boundary of a linear classifier given by parameters $(\mathbf{w}, b)$.
- The margin m of a linear classifier $(\mathbf{w}, b)$ is defined as follows:
  - (i) If the classifier classifies all data correctly then $m = 2 \min_{\mathbf{x} \in \mathcal{T}} d(\mathbf{x})$.
    Points $\mathbf{x} \in \mathcal{T}$ safisfying $m = 2d(\mathbf{x})$ are called **support vectors**.
  - (ii) If the classifier has non-zero error on $\mathcal{T}$ then $m = 0$.
- **Goal**: Find the classifier $(\mathbf{w}^*, b^*)$ maximizing the margin. Vapnik justifies the use of maximum margin from the viewpoint of Structural Risk Minimization.

Margin of a classifier $(\mathbf{w}, b)$:



Maximum margin classifier $(\mathbf{w}^*, b^*)$:

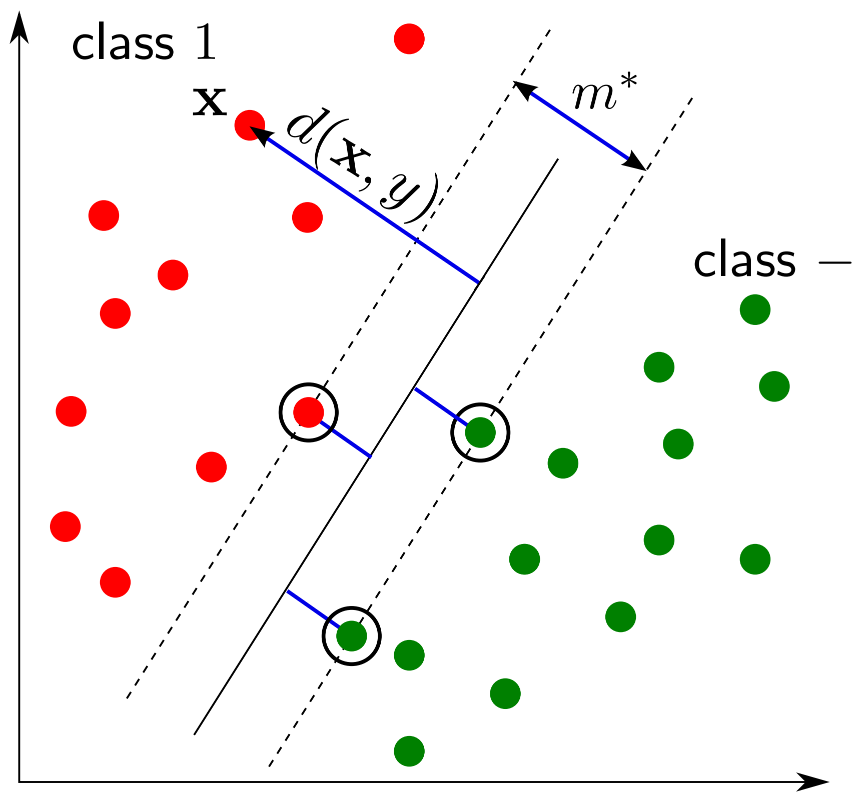


Maximizing Margin, Formulation

- Let us define signed distance $d(\mathbf{x}, y)$ of a point $\mathbf{x}$ belonging to class $y \in \{1, -1\}$ to the decision boundary of classifier $(\mathbf{w}, b)$:

$$d(\mathbf{x}, y) = \frac{y(\mathbf{w} \cdot \mathbf{x} + b)}{\|\mathbf{w}\|} \qquad (2)$$

- We search for $(\mathbf{w}, b)$ such that $d(\mathbf{x}, y) > 0$ for all training data (all training points are in their class' half-space). This is equivalent to $y(\mathbf{w} \cdot \mathbf{x} + b) > 0$.



Optimization task:

$$(\mathbf{w}^*, b^*) = \operatorname*{argmax}_{\mathbf{w},b} \min_{(\mathbf{x},y)\in\mathcal{T}} 2d(\mathbf{x}, y)$$

subject to:

$$y(\mathbf{w} \cdot \mathbf{x} + b) > 0, \forall(\mathbf{x}, y) \in \mathcal{T} \qquad \text{(C)}$$

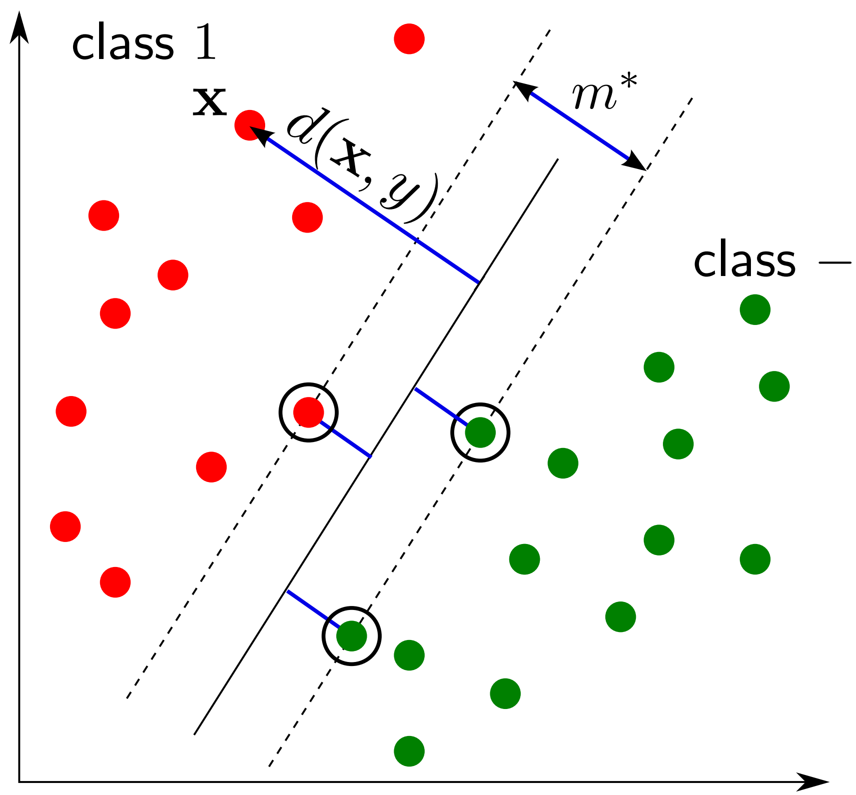
Maximizing Margin, Scale Ambiguity

- There is a scale ambiguity in the parameters $(\mathbf{w}, b)$. Any feasible $(\mathbf{w}, b)$ (that is, satisfying Eq. (C)) can be multiplied by a positive constant $(\mathbf{w}, b) \to (\sigma\mathbf{w}, \sigma b)$, and:

  (i) feasibility does not change, as

$$y(\sigma\mathbf{w} \cdot \mathbf{x} + \sigma b) = \sigma y(\mathbf{w} \cdot \mathbf{x} + b) > 0 \Leftrightarrow y(\mathbf{w} \cdot \mathbf{x} + b) > 0, \text{ and} \tag{3}$$

  (ii) signed distances do not change, as



$$d(\mathbf{x}, y) = \frac{y(\sigma\mathbf{w} \cdot \mathbf{x} + \sigma b)}{\|\sigma\mathbf{w}\|} = \frac{y(\mathbf{w} \cdot \mathbf{x} + b)}{\|\mathbf{w}\|}. \tag{4}$$

Optimization task:

$$(\mathbf{w}^*, b^*) = \operatorname*{argmax}_{\mathbf{w}, b} \min_{(\mathbf{x}, y) \in \mathcal{T}} 2d(\mathbf{x}, y)$$

subject to:

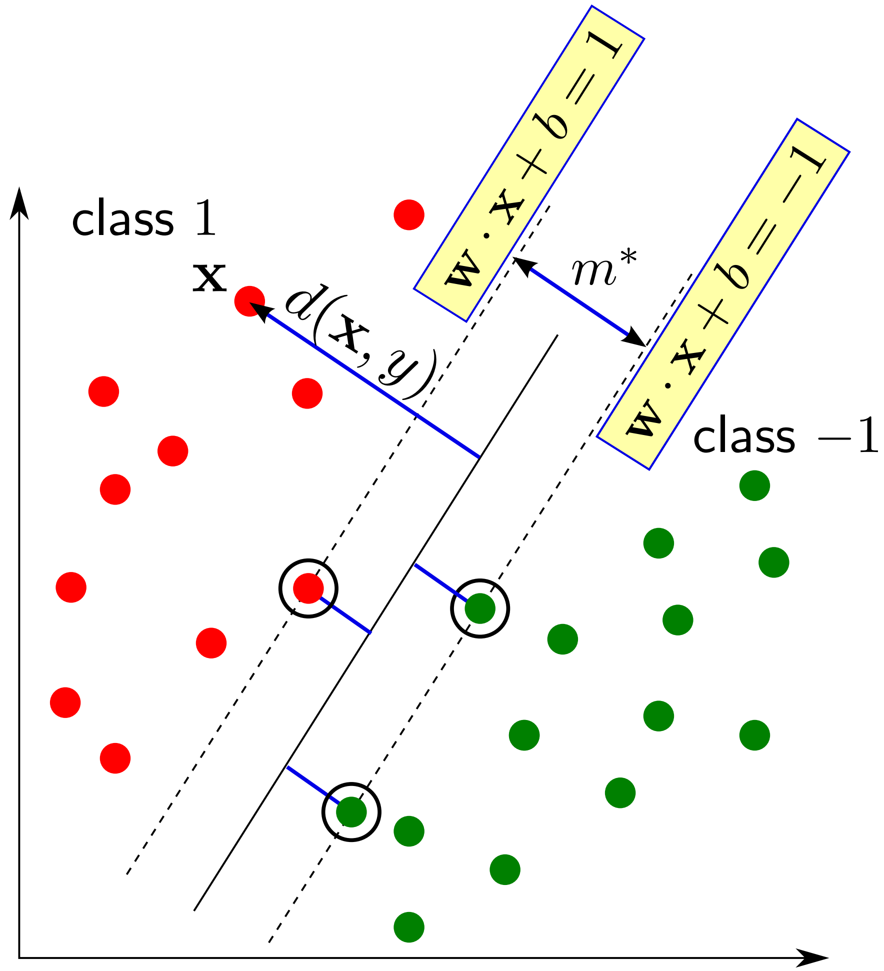
$$y(\mathbf{w} \cdot \mathbf{x} + b) > 0, \forall(\mathbf{x}, y) \in \mathcal{T} \tag{C}$$

Maximizing Margin, Fixing Scale

- Constraints $y(\mathbf{w} \cdot \mathbf{x} + b) > 0$ are equivalent to $y(\mathbf{w} \cdot \mathbf{x} + b) \geq \epsilon$ (with $\epsilon > 0$)
- Break the scale ambiguity by setting $\epsilon = 1$:

$$(\mathbf{w}^*, b^*) = \operatorname*{argmax}_{\mathbf{w},b} \min_{(\mathbf{x},y)\in\mathcal{T}} 2d(\mathbf{x}, y)$$

$$\text{subject to: } y(\mathbf{w} \cdot \mathbf{x} + b) \geq 1, \forall(\mathbf{x}, y) \in \mathcal{T} \qquad (5)$$



Optimization task (original):

$$(\mathbf{w}^*, b^*) = \operatorname*{argmax}_{\mathbf{w},b} \min_{(\mathbf{x},y)\in\mathcal{T}} 2d(\mathbf{x}, y)$$

subject to:

$$y(\mathbf{w} \cdot \mathbf{x} + b) > 0, \forall(\mathbf{x}, y) \in \mathcal{T} \qquad \text{(C)}$$

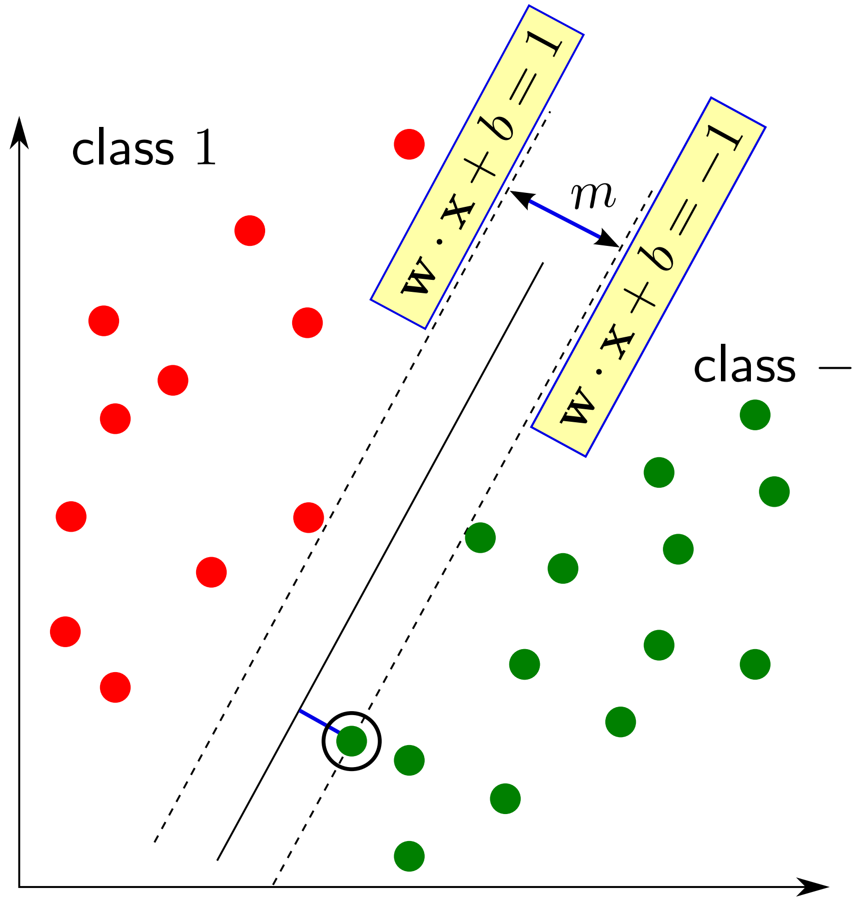
$$d(\mathbf{x}, y) = \frac{y(\mathbf{w} \cdot \mathbf{x} + b)}{\|\mathbf{w}\|}$$

Maximizing Margin, Final Optimization Formulation (1)

- That is, all points must be outside the strip delineated by the two lines $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$. The width of this strip is $\frac{2}{\|\mathbf{w}\|}$. It follows that the maximum margin m^* is

$$m^* = \max_{\mathbf{w},b} \min_{(\mathbf{x},y)\in\mathcal{T}} 2d(\mathbf{x},y) = \max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|}$$

$$\text{subject to: } y(\mathbf{w} \cdot \mathbf{x} + b) \geq 1, \forall (\mathbf{x},y) \in \mathcal{T} \qquad (6)$$



Optimization task (original):

$$(\mathbf{w}^*, b^*) = \operatorname*{argmax}_{\mathbf{w},b} \min_{(\mathbf{x},y)\in\mathcal{T}} 2d(\mathbf{x},y)$$

subject to:

$$y(\mathbf{w} \cdot \mathbf{x} + b) > 0, \forall (\mathbf{x},y) \in \mathcal{T} \qquad \text{(C)}$$

$$d(\mathbf{x},y) = \frac{y(\mathbf{w} \cdot \mathbf{x} + b)}{\|\mathbf{w}\|}$$

Maximizing Margin, Final Optimization Formulation (2)

- That is, all points must be outside the strip delineated by the two lines $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$. The width of this strip is $\frac{2}{\|\mathbf{w}\|}$. It follows that the maximum margin m^* is

$$m^* = \max_{\mathbf{w},b} \min_{(\mathbf{x},y)\in\mathcal{T}} 2d(\mathbf{x}, y) = \max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|}$$

$$\text{subject to: } y(\mathbf{w} \cdot \mathbf{x} + b) \geq 1, \forall(\mathbf{x}, y) \in \mathcal{T} \tag{7}$$

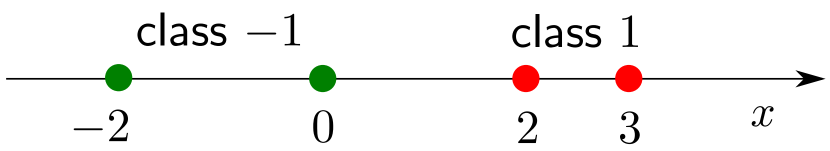
- There holds: $\operatorname*{argmax}_{\mathbf{w}} \dfrac{2}{\|\mathbf{w}\|} = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{w}\| = \operatorname*{argmin}_{\mathbf{w}} \dfrac{1}{2}\|\mathbf{w}\|^2$. Therefore, the $(\mathbf{w}^*, b^*)$ maximizing the margin are:

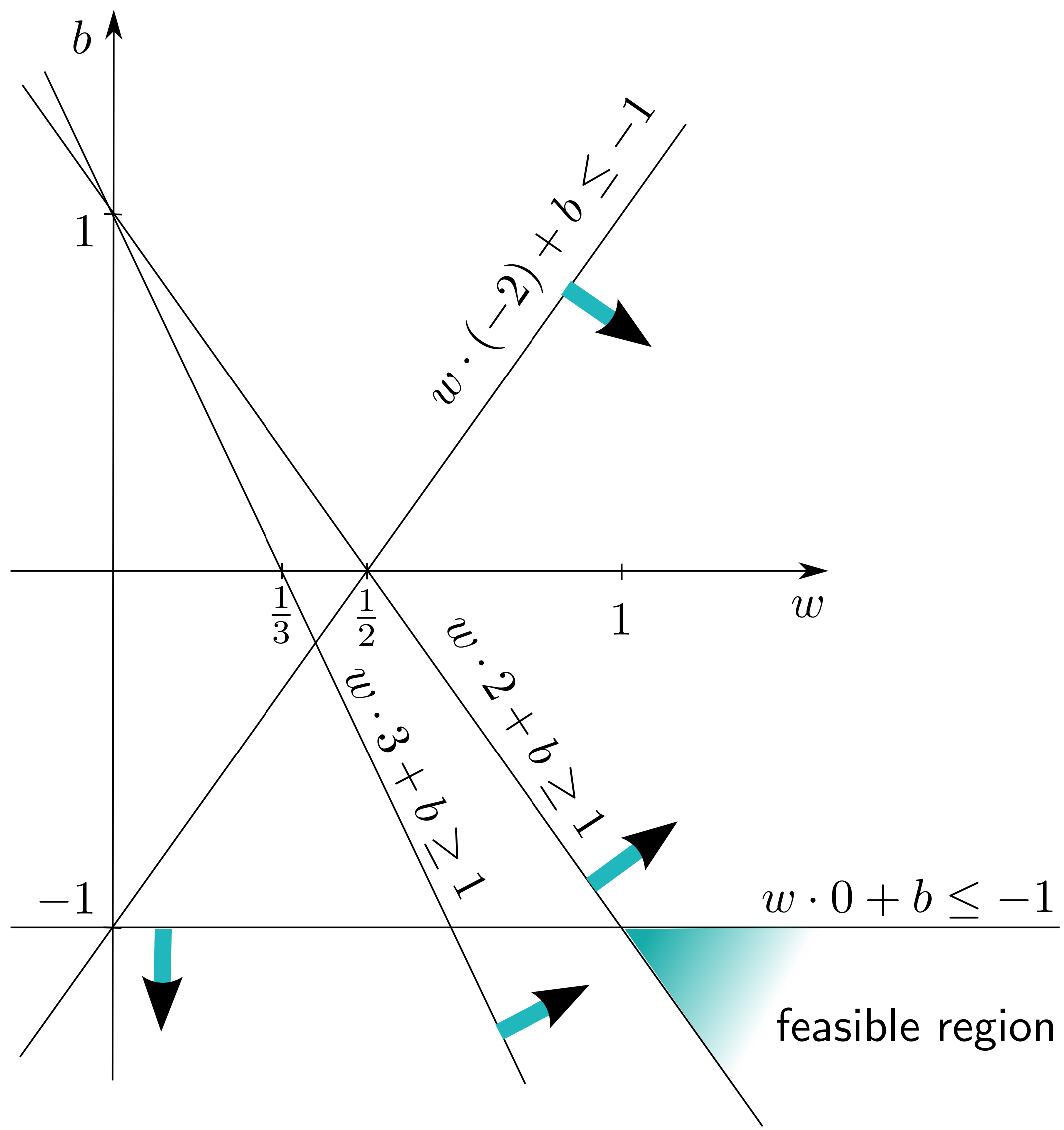
$$(\mathbf{w}^*, b^*) = \operatorname*{argmin}_{(\mathbf{w},b)} \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to: } y(\mathbf{w} \cdot \mathbf{x} + b) \geq 1, \forall(\mathbf{x}, y) \in \mathcal{T} \tag{8}$$

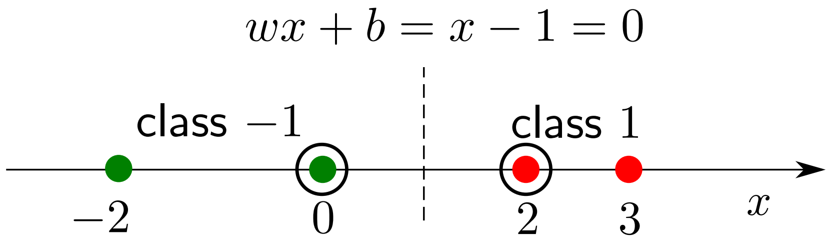
- This is a Quadratic Programming (QP) problem (more generally, it is minimization of a convex function on a convex domain.)

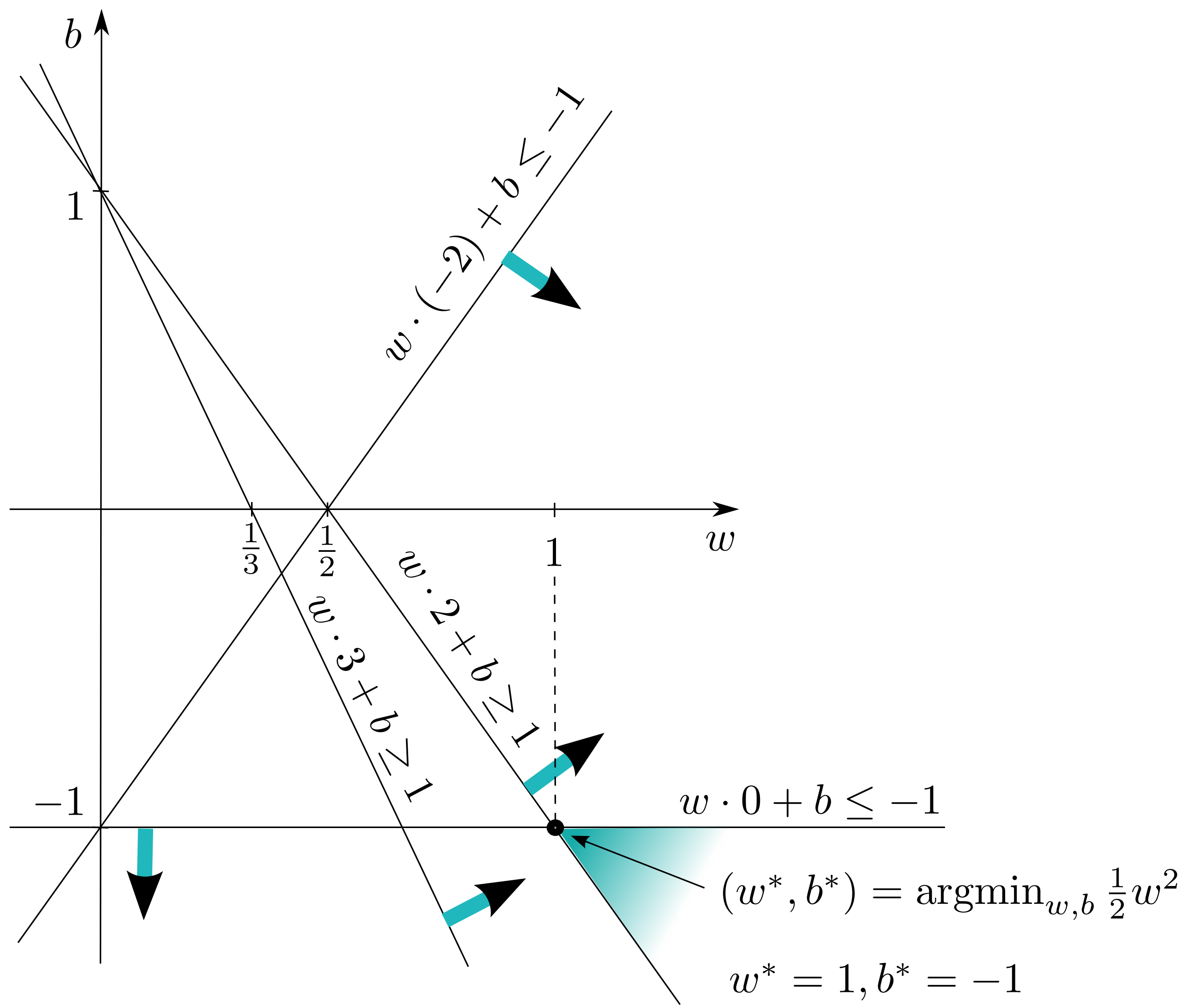
SVM, Example (1D)





SVM, Example (1D), Result





SVM, Primal Problem

The derived optimization problem for $\mathbf{w}$ and b is

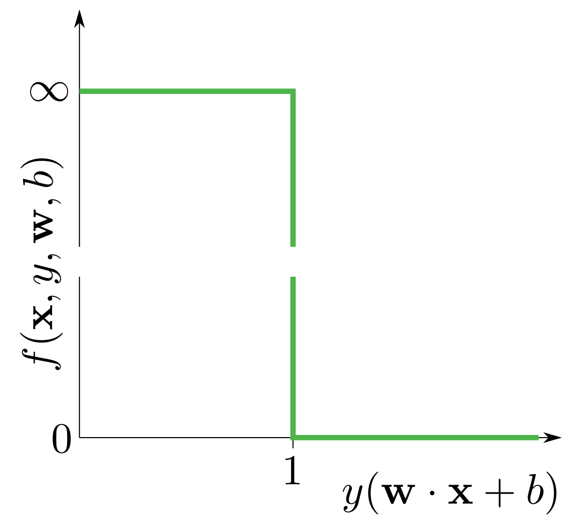
$$(\mathbf{w}^*, b^*) = \operatorname*{argmin}_{(\mathbf{w},b)} \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to: } y(\mathbf{w} \cdot \mathbf{x} + b) \geq 1, \forall (\mathbf{x}, y) \in \mathcal{T} \tag{9}$$

It is called *primal* problem. We will also soon derive the *dual* problem. For now, note that the above optimization task can be equivalently regarded as solving an unconstrained problem (this observation will become handy when deriving the dual problem):

$$(\mathbf{w}^*, b^*) = \operatorname*{argmin}_{(\mathbf{w},b)} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{(\mathbf{x},y)\in\mathcal{T}} f(\mathbf{x}, y, \mathbf{w}, b) \right\}, \text{ where} \tag{10}$$

$$f(\mathbf{x}, y, \mathbf{w}, b) = \begin{cases} 0 & \text{if } y(\mathbf{w} \cdot \mathbf{x} + b) \geq 1, \\ \infty, & \text{otherwise} \end{cases} \tag{11}$$



Note that $f(\mathbf{x}, y, \mathbf{w}, b)$ for a given $(\mathbf{x}, y)$ is a convex function of $\mathbf{w}, b$.

The Dual Formulation (1)

Start with just discussed primal formulation. Let $\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$ be the training set. We want to solve

$$(\mathbf{w}^*, b^*) = \underset{(\mathbf{w},b)}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} f(\mathbf{x}_i, y_i, \mathbf{w}, b) \right\}, \text{ where}$$

$$f(\mathbf{x}_i, y_i, \mathbf{w}, b) = \begin{cases} 0 & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \\ \infty, & \text{otherwise} \end{cases} \tag{12}$$

This is the same as (α_i's are non-negative multipliers):

$$(\mathbf{w}^*, b^*) = \underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \max_{\substack{\{\alpha_i\} \\ \alpha_i \geq 0 \\ i \in \{1,..,N\}}} \left(-\sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \right) \right\}. \tag{13}$$

because

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 1 \;\Rightarrow\; \max_{\alpha_i}(-\alpha_i[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]) = 0 \text{ for } \alpha_i = 0\,, \tag{14}$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) < 1 \;\Rightarrow\; \max_{\alpha_i}(-\alpha_i[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]) = \infty \text{ for } \alpha_i = \infty\,, \tag{15}$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 \;\Rightarrow\; \max_{\alpha_i}(-\alpha_i[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]) = 0 \text{ for any } \alpha_i \geq 0\,. \tag{16}$$

The Dual Formulation (2)

This is in turn the same as

$$(\mathbf{w}^*, b^*) = \operatorname*{argmin}_{\mathbf{w},b} \max_{\substack{\{\alpha_i\} \\ \alpha_i \geq 0 \\ i \in \{1,..,N\}}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \right\} . \tag{17}$$

There holds, in full generality, that $\max_p \min_q f(p,q) \leq \min_q \max_p f(p,q)$. For our case,

$$\min_{\mathbf{w},b} \max_{\substack{\{\alpha_i\} \\ \alpha_i \geq 0 \\ i \in \{1,..,N\}}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \right\} \geq$$

$$\geq \max_{\substack{\{\alpha_i\} \\ \alpha_i \geq 0 \\ i \in \{1,..,N\}}} \min_{\mathbf{w},b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \right\} \tag{18}$$

This is the essence of converting the primal problem to the dual one. And, our case is even better: strong duality holds, and the two terms are equal (duality gap is zero). Denote the inner term by $L(\mathbf{w}, b, \alpha)$ (corresponds to what's commonly known as the Lagrangian):

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \tag{19}$$

The Dual Formulation (3)

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \tag{20}$$

We want to find $\mathrm{argmax}_{\alpha \geq 0} \min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha)$. First, for fixed α, find $\min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha)$:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \;\Rightarrow\; \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{21}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{22}$$

Put this to Lagrangian:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = \tag{23}$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i\right) \cdot \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i b + \sum_{i=1}^{N} \alpha_i \tag{24}$$

$$= -\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \alpha_i = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{25}$$

The Dual Formulation, Result and Insights

The dual optimization problem:

$$\alpha = \underset{\alpha}{\operatorname{argmax}} \left(\min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha) \right) = \underset{\alpha}{\operatorname{argmax}} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \right\} \tag{26}$$

$$\text{subject to: } \sum_i \alpha_i y_i = 0; \;\; \alpha_i \geq 0, \;\; \forall i \in \{1, 2, ..., N\} \tag{27}$$

- Number of optimization variables α_i's is N (the number of training data). But at the solution, all α_i's but those of support vectors are zero.
- Once the solution is obtained, the primal variables can be computed as

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \qquad \text{only support vectors } (\alpha_i > 0) \text{ contribute} \tag{28}$$

$$y^S[\mathbf{w} \cdot \mathbf{x}^S + b] = 1 \text{ for any support vector } (\mathbf{x}^S, y^S) \;\Rightarrow\; b = y^S - \mathbf{w} \cdot \mathbf{x}^S \tag{29}$$

- The discriminant function $\mathbf{w} \cdot \mathbf{x} + b$ thus takes the form ($\mathcal{P}$ are indices of all support vectors):

$$\mathbf{w} \cdot \mathbf{x} + b = \sum_{i \in \mathcal{P}} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + \underbrace{y^S - \sum_{i \in \mathcal{P}} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}^S)}_{\text{constant, independent of } \mathbf{x}} \tag{30}$$

- Both the dual classification problem and the discriminant function involve data points **only** in the form of **dot products**.

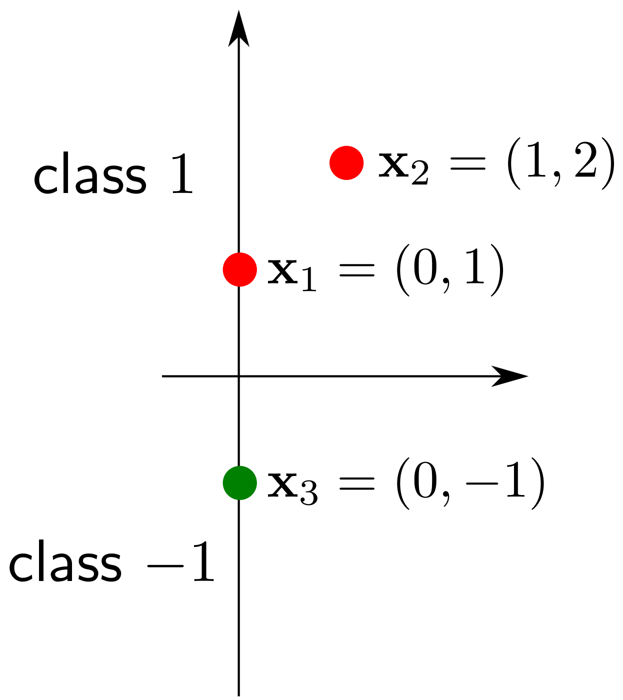
The Dual Problem, Example (1)

Consider the 3 points as below

Objective: maximize

$$\alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}^T \begin{bmatrix} y_1 y_1 \mathbf{x}_1 \cdot \mathbf{x_1} & y_1 y_2 \mathbf{x}_1 \cdot \mathbf{x_2} & y_1 y_3 \mathbf{x}_1 \cdot \mathbf{x_3} \\ y_2 y_1 \mathbf{x}_2 \cdot \mathbf{x_1} & y_2 y_2 \mathbf{x}_2 \cdot \mathbf{x_2} & y_2 y_3 \mathbf{x}_2 \cdot \mathbf{x_3} \\ y_3 y_1 \mathbf{x}_3 \cdot \mathbf{x_1} & y_3 y_2 \mathbf{x}_3 \cdot \mathbf{x_2} & y_3 y_3 \mathbf{x}_3 \cdot \mathbf{x_3} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

subject to: $\alpha_1, \alpha_2, \alpha_3 \geq 0$; $\alpha_1 + \alpha_2 - \alpha_3 = 0$



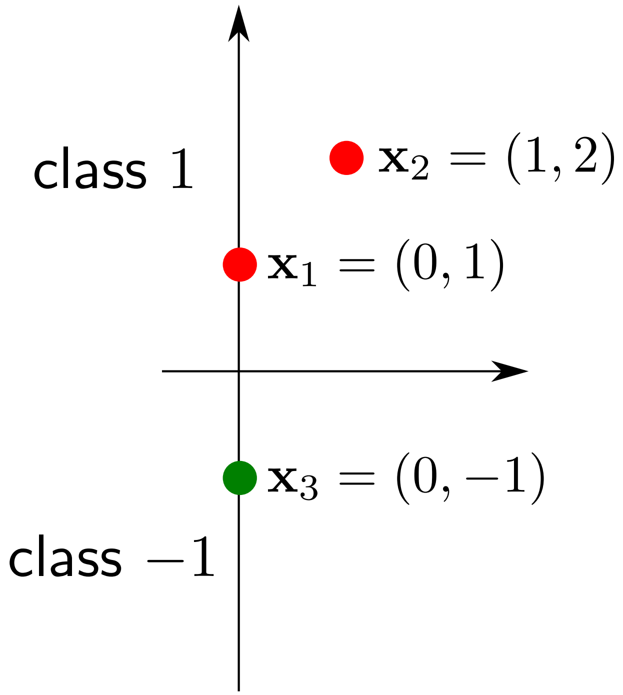
The Dual Problem, Example (2)

Consider the 3 points as below

Objective: maximize

$$\alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}^T \begin{bmatrix} 1 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

subject to: $\alpha_1, \alpha_2, \alpha_3 \geq 0; \;\; \alpha_1 + \alpha_2 - \alpha_3 = 0$

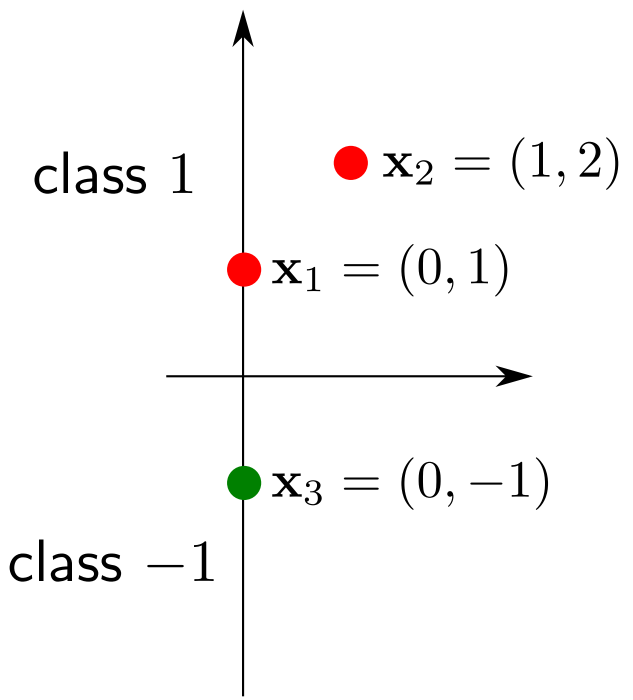


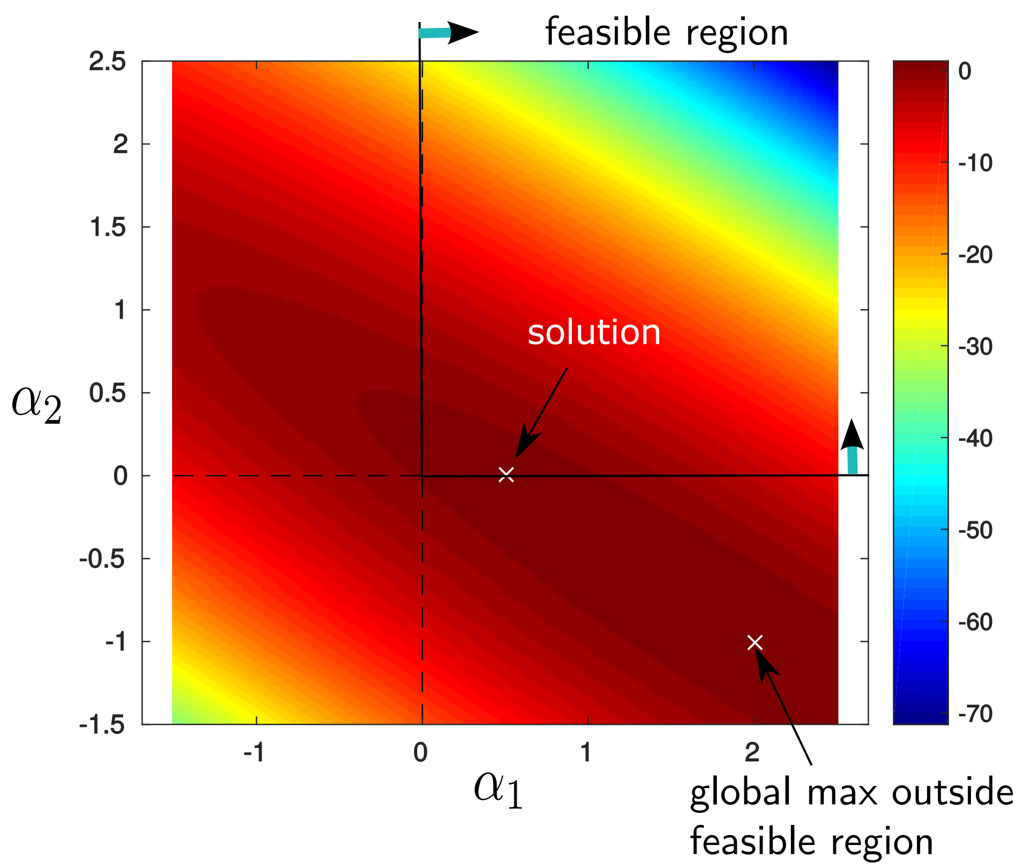
The Dual Problem, Example (3)

Substitute $\alpha_3 = \alpha_1 + \alpha_2$ and search for solution as a problem in α_1, α_2. After some straightforward computation, the original problem turns to:

$$\text{maximize } 2(\alpha_1 + \alpha_2) - \frac{1}{2} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}^T \begin{bmatrix} 4 & 6 \\ 6 & 10 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

subject to: $\alpha_1, \alpha_2 \geq 0$. **Solution**: $(\alpha_1, \alpha_2) = (\frac{1}{2}, 0)$, $\alpha_3 = \frac{1}{2} + 0 = \frac{1}{2}$.





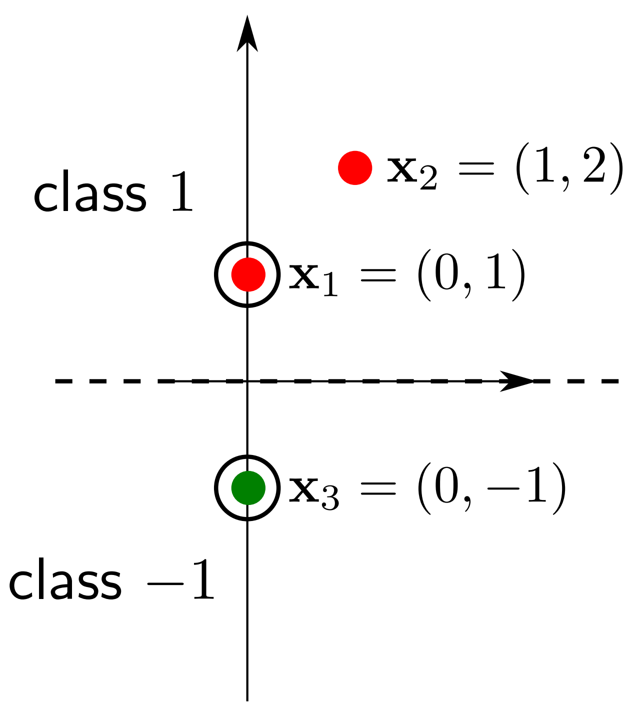
The Dual Problem, Example, Result

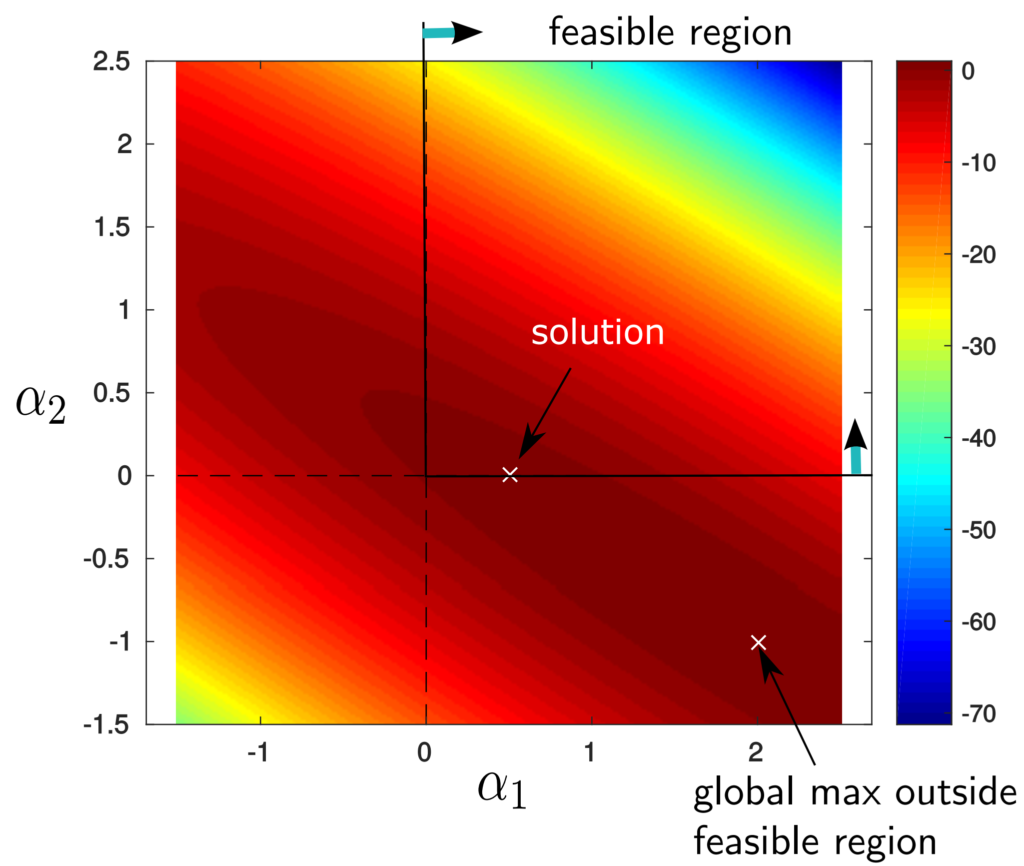
Result: $(\alpha_1, \alpha_2, \alpha_3) = (\frac{1}{2}, 0, \frac{1}{2})$. The support vectors are $\mathbf{x}_1$ and $\mathbf{x}_3$ because their $\alpha_i > 0$.

Vector $\mathbf{w} = \sum_{i=\{1,3\}} \alpha_i y_i \mathbf{x}_i = \frac{1}{2}(0,1) - \frac{1}{2}(0,-1) = (0,1)$.

Offset $b = y^S - \mathbf{w}\mathbf{x}^S = 1 - \mathbf{w}\mathbf{x}_1 = -1 - \mathbf{w}\mathbf{x}_3 = 0$.

Decision boundary $(0,1)^T \cdot \mathbf{x} = 0\,.$

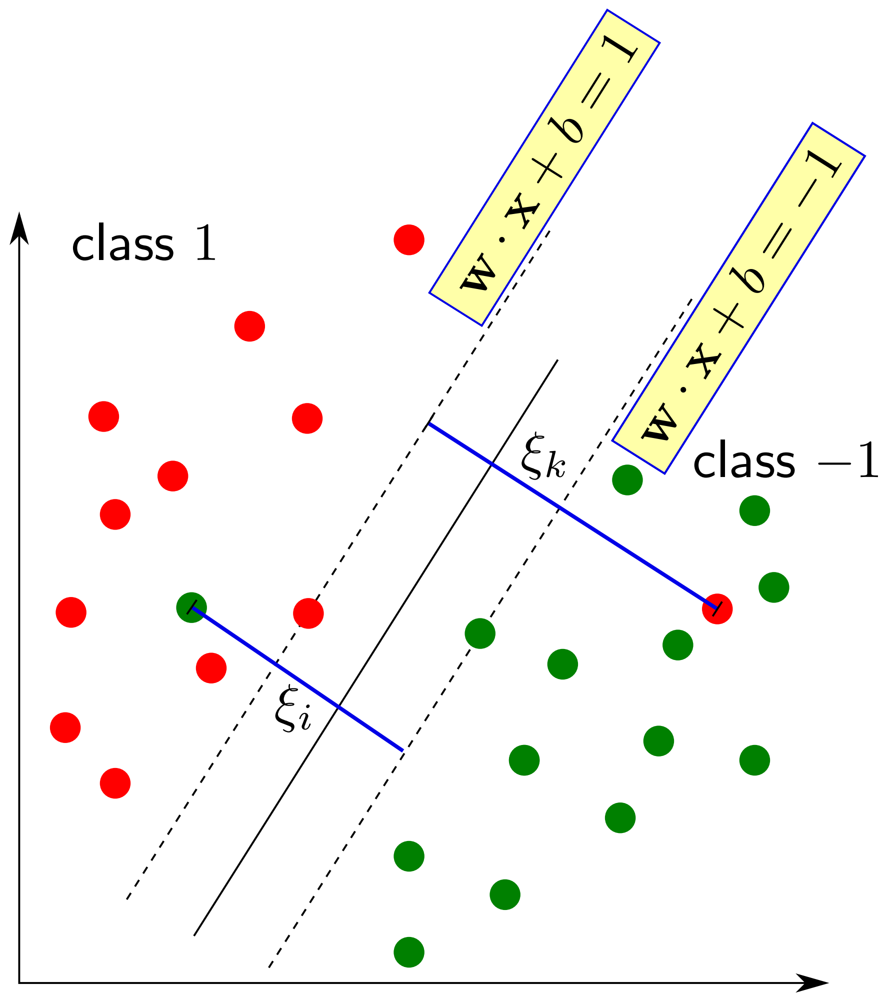




Soft Margin SVM

If the data are not linearly separable, *slack variables* ξ_i need to be introduced.

- Position and size of margin is implied by $\mathbf{w}$ and b, as before.
- If a point $(\mathbf{x}, y)$ fulfills the condition $y(\mathbf{w}\cdot\mathbf{x}+b)\geq 1$ then no penalty is paid.
- Otherwise, the condition is relaxed to $y(\mathbf{w}\cdot\mathbf{x}+b)\geq 1-\xi$ and penalty $C\cdot\xi$ is paid



$$(\mathbf{w}^*, b^*) = \underset{(\mathbf{w},b)}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \qquad (31)$$

subject to:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1-\xi_i, \qquad (32)$$

$$\xi_i \geq 0, \qquad (33)$$

$$\forall i = 1, ..., N$$

Soft Margin SVM

The primal problem

$$(\mathbf{w}^*, b^*) = \underset{(\mathbf{w},b)}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to: } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i,\ \ \forall i = 1, ..., N \tag{34}$$

$$\xi_i \geq 0,\ \ \forall i = 1, ..., N \tag{35}$$

The dual problem:

$$\alpha = \underset{\alpha}{\operatorname{argmax}} \left\{ \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j \right\} \tag{36}$$

$$\text{subject to: } \sum_i \alpha_i y_i = 0 \tag{37}$$

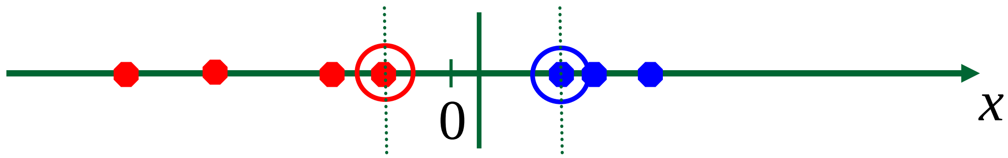
$$0 \leq \alpha_i \leq C,\ \ \forall i \in \{1, 2, ..., N\} \tag{38}$$

Linear SVMs: Overview

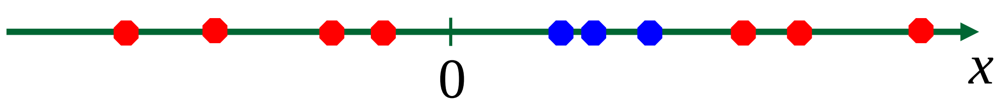
- The classifier is a *separating hyperplane.*
- Most "important" training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\mathbf{x_i}$ are support vectors with non-zero Lagrangian multipliers h_i.
- Both in the dual formulation of the problem and in the solution training points appear only inside inner-products.

Who really need linear classifiers

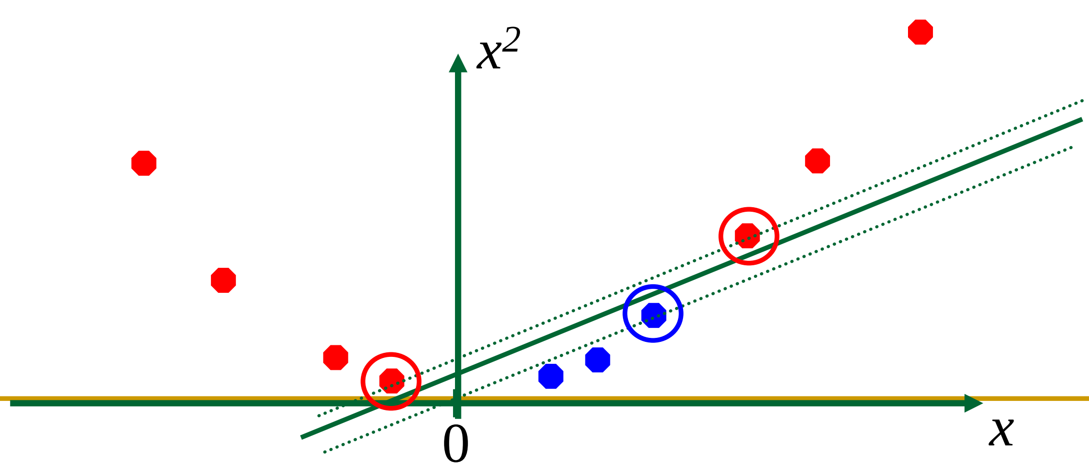
- Datasets that are linearly separable with some noise, linear SVM work well:



- But if the dataset is non-linearly separable?

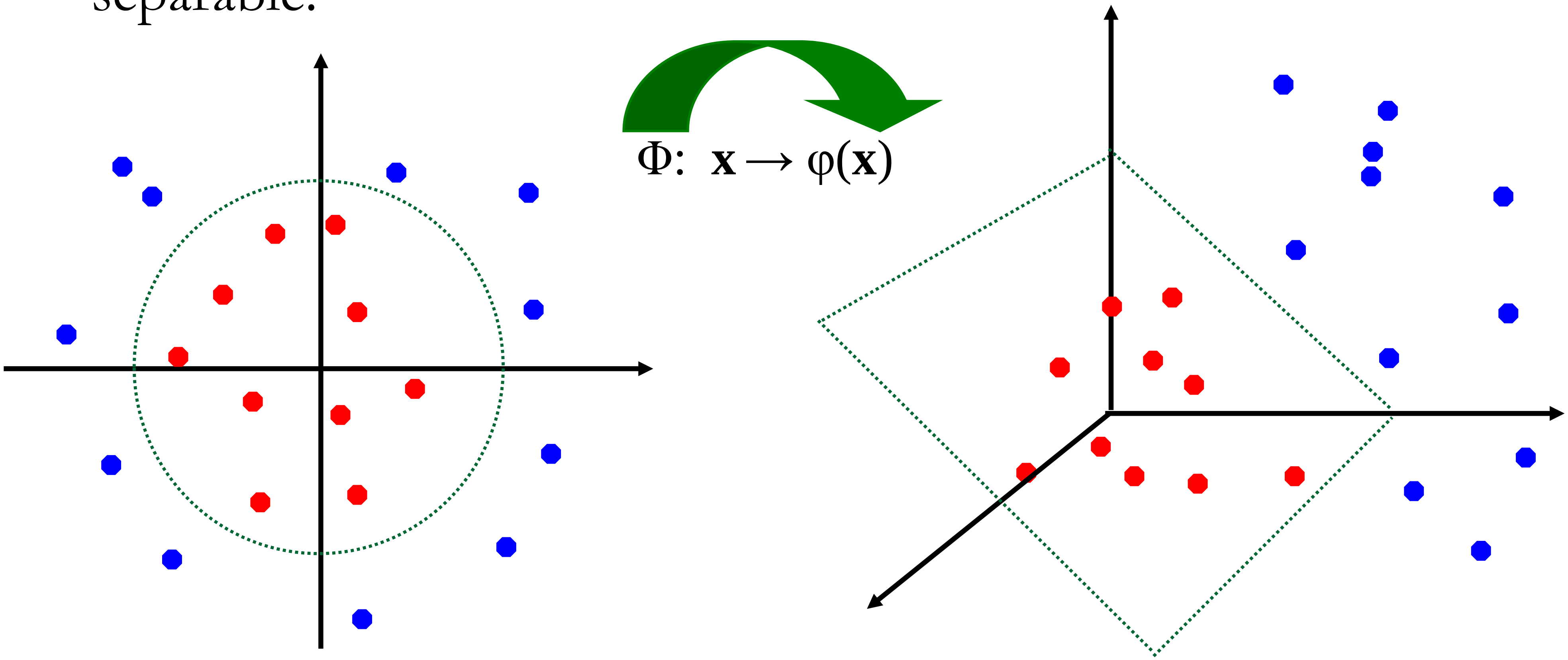


- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

- General idea: the original space can always be mapped to some higher-dimensional feature space where the training set becomes separable:



The "Kernel Trick"

- The SVM only relies on the inner-product between vectors $\mathbf{x}_\mathrm{i} \cdot \mathbf{x}_\mathrm{j}$
- If every datapoint is mapped into high-dimensional space via some transformation Φ: $\mathbf{x} \longrightarrow \varphi(\mathbf{x})$, the inner-product becomes:

$$K(\mathbf{x}_\mathrm{i},\mathbf{x}_\mathrm{j}) = \varphi(\mathbf{x}_\mathrm{i}) \cdot \varphi(\mathbf{x}_j)$$

- $K(\mathbf{x}_\mathrm{i},\mathbf{x}_j)$ is called the kernel function.
- For SVM, we only need specify the kernel $K(\mathbf{x}_\mathrm{i},\mathbf{x}_j)$, without need to know the corresponding non-linear mapping, $\varphi(\mathbf{x})$.

Non-linear SVMs

- The dual problem:

$$\text{Maximizing}: \ L(\mathbf{h}) = \sum_{i=1}^{N} h_i - \frac{1}{2}\mathbf{h}\cdot\mathbf{D}\cdot\mathbf{h}$$

$$\text{Subject to}: \mathbf{h}\cdot\mathbf{y} = 0$$

$$0 \le \mathbf{h} \le \mathrm{C}$$

$$\text{where } D_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

- Optimization techniques for finding h_i's remain the same!
- The solution is:

$$\mathbf{w}^* = \sum_{i \in SV} h_i y_i \varphi(\mathbf{x}_i)$$

$$f(\mathbf{x}) = \mathbf{w}^* \cdot \varphi(\mathbf{x}) + b^*$$

$$= \sum_{i \in SV} h_i y_i K(\mathbf{x}_i, \mathbf{x}) + b^*$$

Examples of Kernel Trick (1)

- For the example in the previous figure:
  - The non-linear mapping

$$x \rightarrow \varphi(x) = (x, x^2)$$

  - The kernel

$$\varphi(x_i) = (x_i, x_i^2), \quad \varphi(x_j) = (x_j, x_j^2)$$
$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$$
$$= x_i x_j (1 + x_i x_j)$$

- Where is the benefit?

Examples of Kernel Trick (2)

- Polynomial kernel of degree 2 in 2 variables
  - The non-linear mapping:

$$\mathbf{x} = (x_1, x_2)$$
$$\varphi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

  - The kernel

$$\varphi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$
$$\varphi(\mathbf{y}) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1y_2)$$
$$\begin{aligned} K(\mathbf{x}, \mathbf{y}) &= \varphi(\mathbf{x}) \cdot \varphi(\mathbf{y}) \\ &= (1 + \mathbf{x} \cdot \mathbf{y})^2 \end{aligned}$$

Examples of kernel trick (3)

- Gaussian kernel: $$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2}$$

  - The mapping is of infinite dimension:

$$\varphi(\mathbf{x}) = (\ldots, \varphi_\omega(\mathbf{x}), \ldots), \quad \mathit{for} \;\; \omega \in R^d$$
$$\varphi_\omega(\mathbf{x}) = Ae^{-B\omega^2} e^{-i\mathbf{w}\mathbf{x}}$$
$$K(\mathbf{x}, \mathbf{y}) = \int \varphi_\omega(\mathbf{x}) \varphi^*{}_\omega(\mathbf{y}) d\omega$$

- The moral: very high-dimensional and complicated non-linear mapping can be achieved by using a simple kernel!

What Functions are Kernels?

- For some functions $K(\mathbf{x}_i,\mathbf{x}_j)$ checking that $K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$ can be cumbersome.
- Mercer's theorem:

Every semi-positive definite symmetric function is a kernel

Examples of Kernel Functions

- Linear kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$

- Polynomial kernel of power *p*: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^p$

- Gaussian kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2}$
  - In the form, equivalent to RBFNN, but has the advantage of that the center of basis functions, i.e., support vectors, are optimized in a supervised.

- Two-layer perceptron: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\alpha \mathbf{x}_i \cdot \mathbf{x}_j + \beta)$

Lifting Dimension by Polynomial Mapping of Degree d

Let $d \in \mathbb{N}$ and $\mathbf{x} = [x_1, x_2, ..., x_D]^\top \in \mathbb{R}^D$.

Let $\phi_d(\mathbf{x})$ denote the mapping which lifts $\mathbf{x}$ to the space containing all monomials of degree d', $1 \leq d' \leq d$ in the components of $\mathbf{x}$:

For example, when $\mathbf{x} = [x_1, x_2]^\top \in \mathbb{R}^2$,

$$\phi_1(\mathbf{x}) = [x_1, x_2]^\top , \tag{39}$$

$$\phi_2(\mathbf{x}) = [x_1, x_2, x_1^2, x_1 x_2, x_2^2]^\top , \tag{40}$$

$$\phi_3(\mathbf{x}) = [x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3]^\top . \tag{41}$$

The number of monomials of degree d' of $\mathbf{x} \in \mathbb{R}^D$ is $\binom{d'+D-1}{d'}$. The dimensionality L of the output space of $\phi_d(\mathbf{x})$ is thus

$$L = \sum_{d'=1}^{d} \binom{d' + D - 1}{d'} . \tag{42}$$

Lifting Dimension by Polynomial Mapping of Degree d

Feature space dimensionality D, lifting by $\phi_d(\mathbf{x})$

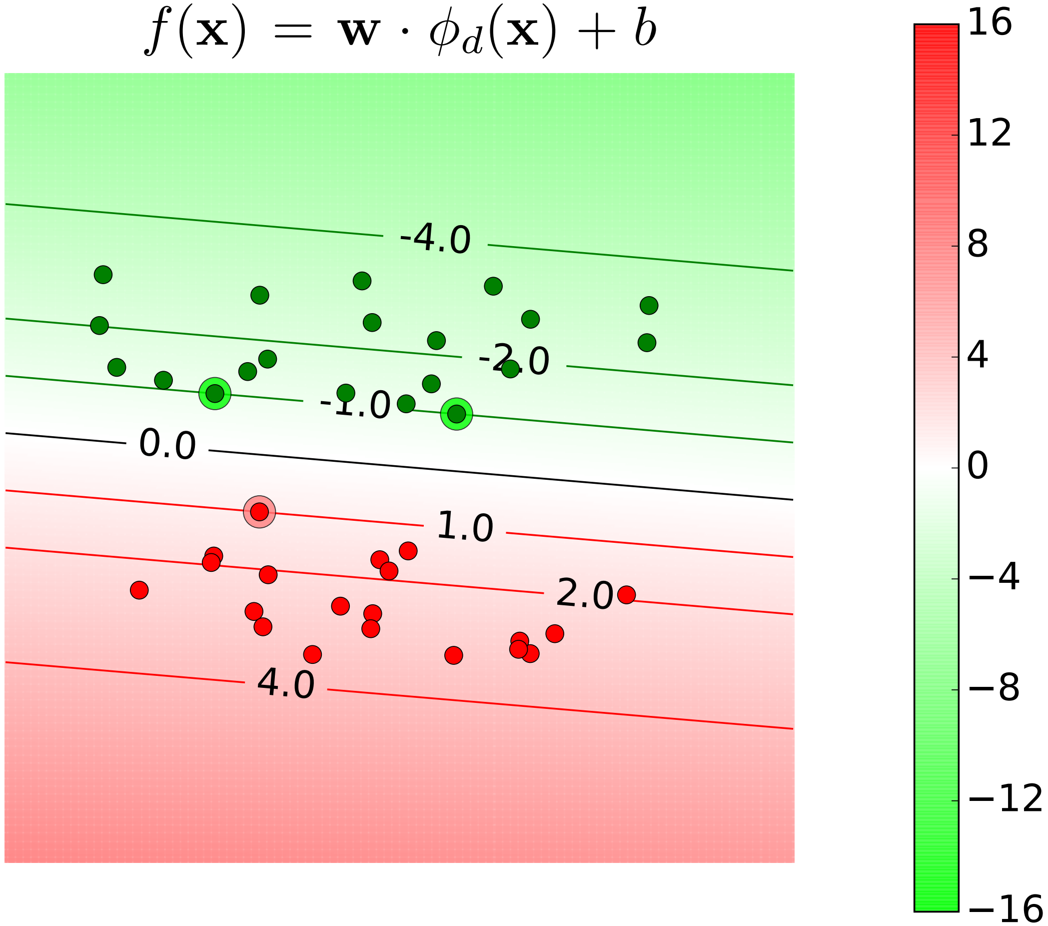
dimensionality of feature space after lifting (L)

D \ d	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	2	5	9	14	20	27	35	44
3	3	9	19	34	55	83	119	164
4	4	14	34	69	125	209	329	494
5	5	20	55	125	251	461	791	1286
6	6	27	83	209	461	923	1715	3002
7	7	35	119	329	791	1715	3431	6434
8	8	44	164	494	1286	3002	6434	12869

Lifting by Polynomial Mapping of Degree d, Example

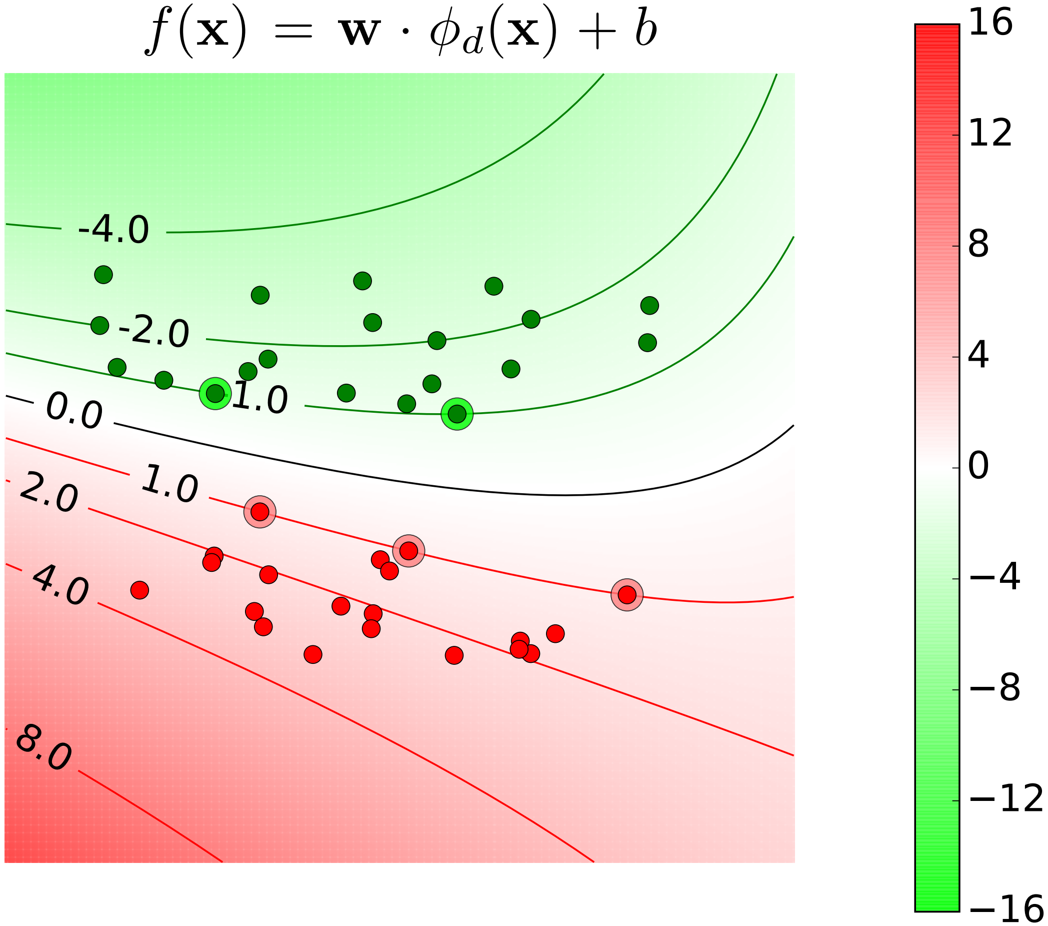
$d = 1$, dim$(\phi_d(\mathbf{x})) = 2$
support vectors : 3

$f(\mathbf{x}) = \mathbf{w} \cdot \phi_d(\mathbf{x}) + b$



$d = 2$, dim$(\phi_d(\mathbf{x})) = 5$
support vectors : 5

$f(\mathbf{x}) = \mathbf{w} \cdot \phi_d(\mathbf{x}) + b$

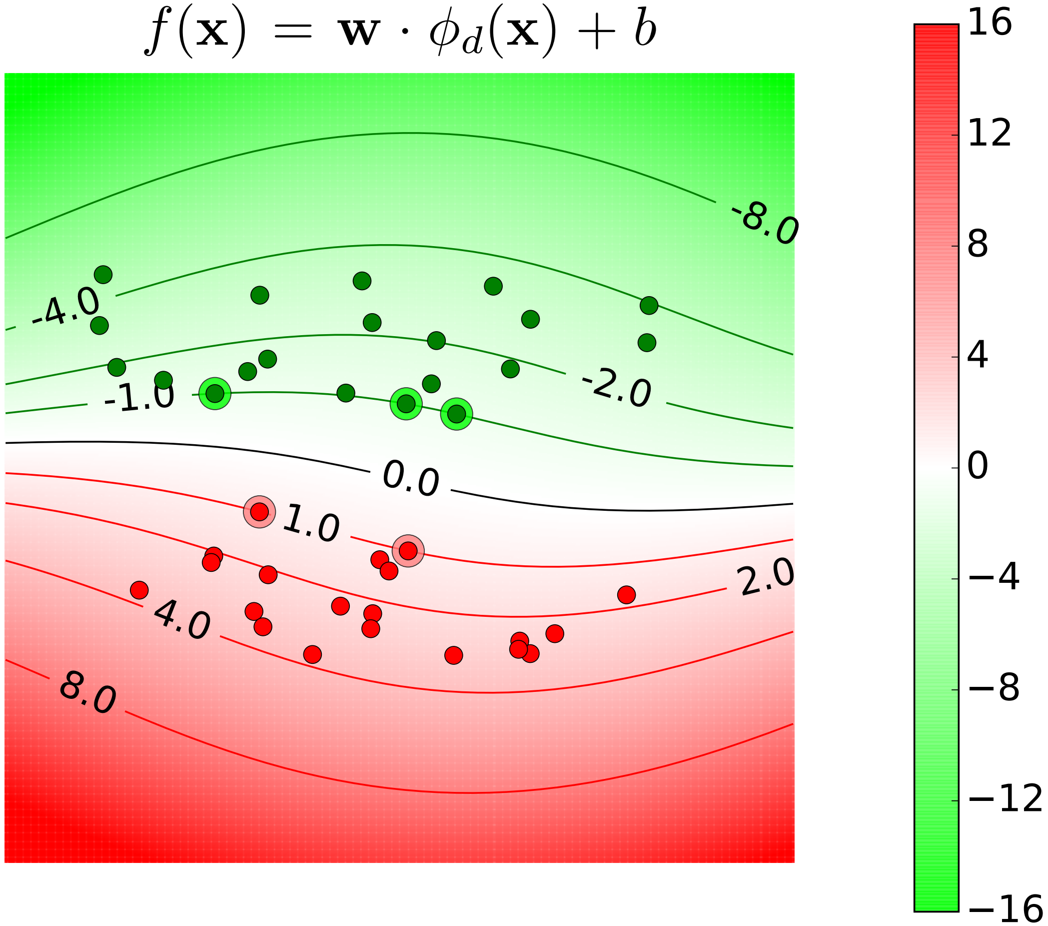


Lifting by Polynomial Mapping of Degree d, Example

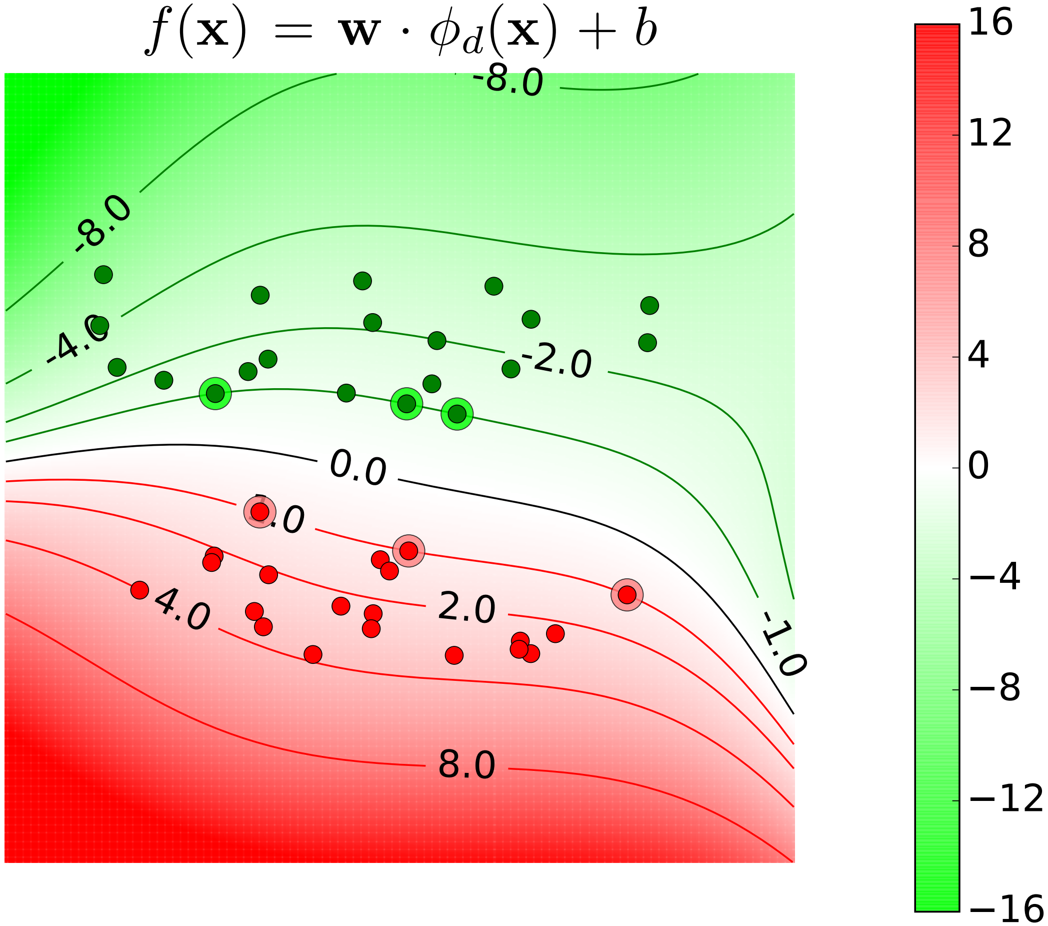
$d = 3$, $\dim(\phi_d(\mathbf{x})) = 9$
support vectors : 5

$d = 4$, $\dim(\phi_d(\mathbf{x})) = 14$
support vectors : 6

$f(\mathbf{x}) = \mathbf{w} \cdot \phi_d(\mathbf{x}) + b$



$f(\mathbf{x}) = \mathbf{w} \cdot \phi_d(\mathbf{x}) + b$



SVM Overviews

- Main features:
  - By using the kernel trick, data is mapped into a high-dimensional feature space, without introducing much computational effort;
  - Maximizing the margin achieves better generation performance;
  - Soft-margin accommodates noisy data;
  - Not too many parameters need to be tuned.
- Demos(http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml)

SVM so far

- SVMs were originally proposed by Boser, Guyon and Vapnik in 1992 and gained increasing popularity in late 1990s.
- SVMs are currently among the best performers for many benchmark datasets.
- SVM techniques have been extended to a number of tasks such as regression [Vapnik *et al.* '97].
- Most popular optimization algorithms for SVMs are SMO [Platt '99] and SVM^{light} [Joachims' 99], both use *decomposition* to handle large size datasets.
- It seems the kernel trick is the most attracting site of SVMs. This idea has now been applied to many other learning models where the inner-product is concerned, and they are called 'kernel' methods.
- Tuning SVMs remains to be the main research focus: how to an optimal kernel? Kernel should match the smooth structure of data.

Appendix

Online demo: http://cs.stanford.edu/people/karpathy/svmjs/demo/