Support Vector Machine (SVM)

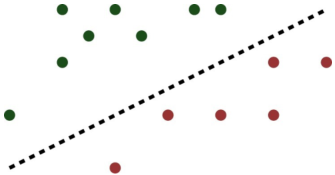
Prof. Gustavo Nonato

NYU / CUSP - GX 5006

February 28, 2017

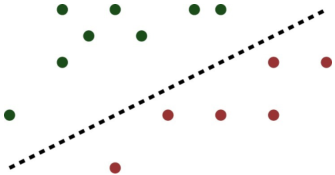
Linearly Separable Data

Suppose a given data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$ is linearly separable.



Linearly Separable Data

Suppose a given data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$ is linearly separable.

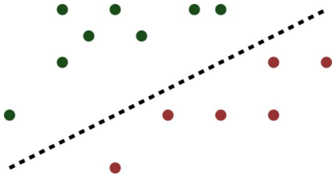


A hyperplane separating the two classes has the following equation:

$$\mathbf{w}^\top \mathbf{x} + b = 0$$

Linearly Separable Data

Suppose a given data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$ is linearly separable.



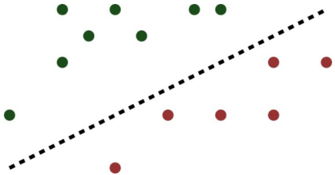
A hyperplane separating the two classes has the following equation:

$$\mathbf{w}^\top \mathbf{x} + b = 0$$

Points where $y(x) = \mathbf{w}^\top \mathbf{x} + b > 0$ are in one side of the plane while points satisfying $y(x) = \mathbf{w}^\top \mathbf{x} + b < 0$ are on the other side.

Linearly Separable Data

Suppose a given data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$ is linearly separable.



A hyperplane separating the two classes has the following equation:

$$\mathbf{w}^\top \mathbf{x} + b = 0$$

Points where $y(x) = \mathbf{w}^\top \mathbf{x} + b > 0$ are in one side of the plane while points satisfying $y(x) = \mathbf{w}^\top \mathbf{x} + b < 0$ are on the other side.

The hyperplane is not unique, there a multitude of parameters $\mathbf{w}$ and b from which $\mathbf{w}^\top \mathbf{x} + b = 0$ is a separating hyperplane.

Linearly Separable Data

Notice that any scale of $\mathbf{w}$ and b still satisfies the equation, that is

$$\lambda \mathbf{w}^\top \mathbf{x} + \lambda b = 0$$

Linearly Separable Data

Notice that any scale of $\mathbf{w}$ and b still satisfies the equation, that is

$$\lambda \mathbf{w}^\top \mathbf{x} + \lambda b = 0$$

For any $\mathbf{x}_i$,

$$\mathbf{w}^\top \mathbf{x}_i + b = c_i$$

Linearly Separable Data

Notice that any scale of $\mathbf{w}$ and b still satisfies the equation, that is

$$\lambda \mathbf{w}^\top \mathbf{x} + \lambda b = 0$$

For any $\mathbf{x}_i$,

$$\mathbf{w}^\top \mathbf{x}_i + b = c_i$$

Let $\mathbf{x}_j^+$ and $\mathbf{x}_j^-$ be the data points from classes $+1$ and -1 that are closer to a given separating hyperplane.
We can tune $\mathbf{w}$ and b such that

$$\mathbf{w}^\top \mathbf{x}_j^+ + b = 1 \quad \mathbf{w}^\top \mathbf{x}_j^- + b = -1$$

Linearly Separable Data

Notice that any scale of $\mathbf{w}$ and b still satisfies the equation, that is

$$\lambda \mathbf{w}^\top \mathbf{x} + \lambda b = 0$$

For any $\mathbf{x}_i$,

$$\mathbf{w}^\top \mathbf{x}_i + b = c_i$$

Let $\mathbf{x}_j^+$ and $\mathbf{x}_j^-$ be the data points from classes $+1$ and -1 that are closer to a given separating hyperplane.
We can tune $\mathbf{w}$ and b such that

$$\mathbf{w}^\top \mathbf{x}_j^+ + b = 1 \quad \mathbf{w}^\top \mathbf{x}_j^- + b = -1$$

Equations above can be written as:

$$y_j(\mathbf{w}^\top \mathbf{x}_j + b) = 1$$

Linearly Separable Data

Intuitively we are looking for the separating hyperplane that is as far as possible from the data points.

Linearly Separable Data

Intuitively we are looking for the separating hyperplane that is as far as possible from the data points.

The distance of $\mathbf{x}_i$ to the hyperplane is given by

$$\frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|}$$

Linearly Separable Data

Intuitively we are looking for the separating hyperplane that is as far as possible from the data points.

The distance of $\mathbf{x}_i$ to the hyperplane is given by

$$\frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|}$$

For any $\mathbf{x}_i$,

$$\frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{|\mathbf{w}\|} \geq \frac{1}{\|\mathbf{w}\|}$$

Linearly Separable Data

Intuitively we are looking for the separating hyperplane that is as far as possible from the data points.

The distance of $\mathbf{x}_i$ to the hyperplane is given by

$$\frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|}$$

For any $\mathbf{x}_i$,

$$\frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{|\mathbf{w}\|} \geq \frac{1}{\|\mathbf{w}\|}$$

Therefore, the sought hyperplane should maximize those distances, which is equivalent to minimizing $\|\mathbf{w}\|^2$.

Linearly Separable Data

Mathematically, the problem can be stated as:

$$\min \mathbf{w}^\top \mathbf{w} \qquad \textit{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

Linearly Separable Data

Mathematically, the problem can be stated as:

$$\min \mathbf{w}^\top \mathbf{w} \qquad \textit{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

This optimization can be handled by Lagrange multipliers

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{w}^\top \mathbf{w} - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \right)$$

Linearly Separable Data

The dual form of the optimization problem is given by

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x_j}$$

$$\alpha_i \geq 0 \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Linearly Separable Data

The dual form of the optimization problem is given by

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x_j}$$

$$\alpha_i \geq 0 \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

The quadratic programming problem above satisfies the KKT conditions:

$$\alpha_i \geq 0$$

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \geq 0$$

Linearly Separable Data

The dual form of the optimization problem is given by

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x_j}$$

$$\alpha_i \geq 0 \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

The quadratic programming problem above satisfies the KKT conditions:

$$\alpha_i \geq 0$$
$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \geq 0$$

$$\alpha_i \left(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \right) = 0$$

Linearly Separable Data

The dual form of the optimization problem is given by

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x_j}$$

$$\alpha_i \geq 0 \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

The quadratic programming problem above satisfies the KKT conditions:

$$\alpha_i \geq 0$$

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \geq 0$$

$$\underbrace{\alpha_i \left(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \right) = 0}_{\alpha_i = 0 \text{ or } y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1}$$

only the α_i corresponding to the *support vectors* are not zero.

Linearly Separable Data

The solution of the quadratic programming problem give us the α_i, from which we compute:

Linearly Separable Data

The solution of the quadratic programming problem give us the α_i, from which we compute:

$$b = \frac{1}{n_s} \sum_{i \in S} \left(y_i - \sum_{j \in S} \alpha_j y_j \mathbf{x}_i^\top \mathbf{x}_j \right)$$

where S is the set of indices for which $\alpha_i \neq 0$.

Linearly Separable Data

The solution of the quadratic programming problem give us the α_i, from which we compute:

$$b = \frac{1}{n_s} \sum_{i \in S} \left(y_i - \sum_{j \in S} \alpha_j y_j \mathbf{x}_i^\top \mathbf{x}_j \right)$$

where S is the set of indices for which $\alpha_i \neq 0$.

New data points can be classified by analyzing the sign of

$$y(\mathbf{x}) = \sum_{j \in S} \alpha_j y_j \mathbf{x}^\top \mathbf{x}_j + b$$

General Training Data

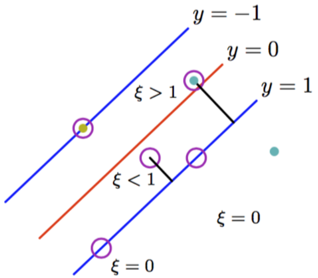
What if data points are not linearly separable?

General Training Data

What if data points are not linearly separable?

We can use slack variables.

$$y_i(\mathbf{w}^\top \mathbf{x} + b) \geq 1 - \boxed{\epsilon_i}, \quad \epsilon_i \geq 0$$

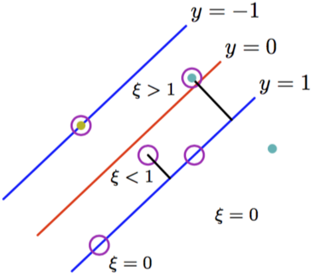


General Training Data

What if data points are not linearly separable?

We can use slack variables.

$$y_i(\mathbf{w}^\top \mathbf{x} + b) \geq 1 - \boxed{\epsilon_i}, \quad \epsilon_i \geq 0$$



Using Lagrange multipliers, the optimization becomes:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbf{w}^\top \mathbf{w} - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \epsilon_i \right) + C \sum_{i=1}^{n} \epsilon_i - \sum_{i=1}^{n} \epsilon_i \beta_i$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are Lagrange multipliers and C controls the trade-off between slack variables and the margin.

General Training Data

Interestingly, the dual formulation is identical to the separable case, with a change only on the constraints

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x_j}$$

$$\boxed{0 \leq \alpha_i \leq C} \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

From KKT conditions we have that many $\alpha_i = 0$, corresponding to points that do not contribute to the margin.

General Training Data

Interestingly, the dual formulation is identical to the separable case, with a change only on the constraints

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x_j}$$

$$\boxed{0 \leq \alpha_i \leq C} \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

From KKT conditions we have that many $\alpha_i = 0$, corresponding to points that do not contribute to the margin.

$\alpha_i < C \longrightarrow$ points lying on the margin.

General Training Data

Interestingly, the dual formulation is identical to the separable case, with a change only on the constraints

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x_j}$$

$$\boxed{0 \le \alpha_i \le C} \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

From KKT conditions we have that many $\alpha_i = 0$, corresponding to points that do not contribute to the margin.

$\alpha_i < C \longrightarrow$ points lying on the margin.

$\alpha_i = C \longrightarrow$ points inside the margin (correctly or misclassified).

General Training Data

$$b = \frac{1}{n_s} \sum_{i \in S} \left(y_i - \sum_{j \in S} \alpha_j y_j \mathbf{x}_i^\top \mathbf{x}_j \right)$$

where S is the set of indices for which $\alpha_i \neq 0$.

General Training Data

$$b = \frac{1}{n_s} \sum_{i \in S} \left(y_i - \sum_{j \in S} \alpha_j y_j \mathbf{x}_i^\top \mathbf{x}_j \right)$$

where S is the set of indices for which $\alpha_i \neq 0$.

New data points can be classified by analyzing the sign of

$$y(\mathbf{x}) = \sum_{j \in S} \alpha_j y_j \mathbf{x}^\top \mathbf{x}_j + b$$

Kernels

There are two questions that naturally appears from the previous development:

Kernels

There are two questions that naturally appears from the previous development:

- Why to use the dual formulation when we could directly optimize the primal one?

Kernels

There are two questions that naturally appears from the previous development:

- Why to use the dual formulation when we could directly optimize the primal one?
- Why so much sophistication to derived a classifier with linear decision boundary?

Kernels

There are two questions that naturally appears from the previous development:

- Why to use the dual formulation when we could directly optimize the primal one?
- Why so much sophistication to derived a classifier with linear decision boundary?

The answer for both questions comes from the concept of kernels.

Kernels

Let $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a function that assigns to any pair of points $\mathbf{x}_i, \mathbf{x}_j$ a scalar value $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ (a measure of similarity between those points).

Kernels

Let $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a function that assigns to any pair of points $\mathbf{x}_i, \mathbf{x}_j$ a scalar value $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ (a measure of similarity between those points).

The function κ is called a kernel if there is a mapping $\phi : \mathbb{R}^d \to \mathcal{H}$ such that:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \text{ (dot product)}$$

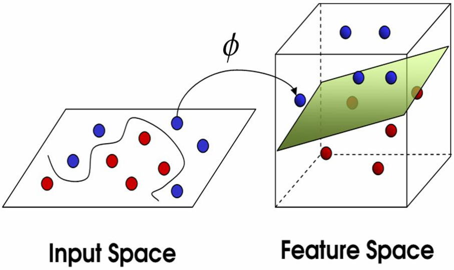
Kernels

Let $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a function that assigns to any pair of points $\mathbf{x}_i, \mathbf{x}_j$ a scalar value $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ (a measure of similarity between those points).

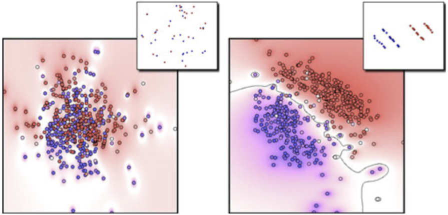
The function κ is called a kernel if there is a mapping $\phi : \mathbb{R}^d \to \mathcal{H}$ such that:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \text{ (dot product)}$$

Typically, ϕ and $\mathcal{H}$ are implicitly defined from the kernel.

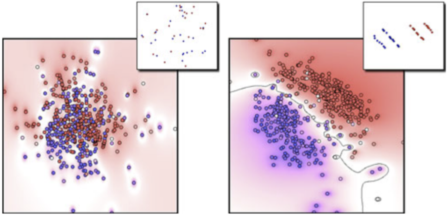


Kernels



(a) (b)

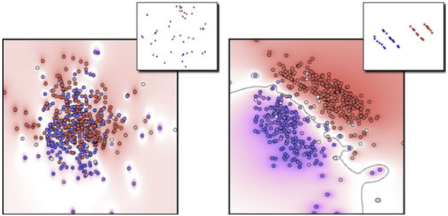
Kernels


(a) (b)

Examples:

$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$ (Identity)

Kernels



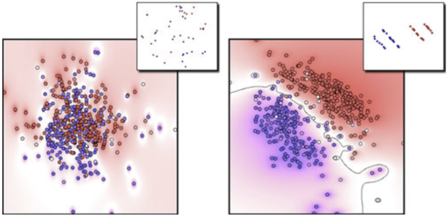
(a) (b)

Examples:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j \text{ (Identity)}$$

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^\top \mathbf{x}_j)^p, \quad p > 0 \text{ (Polynomial)}$$

Kernels



(a) (b)

Examples:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j \text{ (Identity)}$$

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^\top \mathbf{x}_j)^p, \quad p > 0 \text{ (Polynomial)}$$

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \quad \gamma > 0 \text{ (RBF)}$$

Kernel SVM

Kernel SVM

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \ \mathbf{x}_i^\top \mathbf{x_j}$$

Kernel SVM

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i^\top \mathbf{x_j}}$$

Kernel SVM

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i^\top \mathbf{x_j}}$$

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \underbrace{\kappa(\mathbf{x}_i, \mathbf{x_j})}_{\textit{kernel trick}}$$

Kernel SVM

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i^\top \mathbf{x}_\mathbf{j}}$$

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \underbrace{\kappa(\mathbf{x}_i, \mathbf{x}_\mathbf{j})}_{\textit{kernel trick}}$$

$$b = \frac{1}{n_s} \sum_{i \in S} \left(y_i - \sum_{j \in S} \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \right)$$

Kernel SVM

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i^\top \mathbf{x}_\mathbf{j}}$$

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \underbrace{\kappa(\mathbf{x}_i, \mathbf{x}_\mathbf{j})}_{\textit{kernel trick}}$$

$$b = \frac{1}{n_s} \sum_{i \in S} \left(y_i - \sum_{j \in S} \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \right)$$

New data points can be classified by analyzing the sign of

$$y(\mathbf{x}) = \sum_{j \in S} \alpha_j y_j \kappa(\mathbf{x}, \mathbf{x}_j) + b$$

Multiple Classes

Multiple Classes

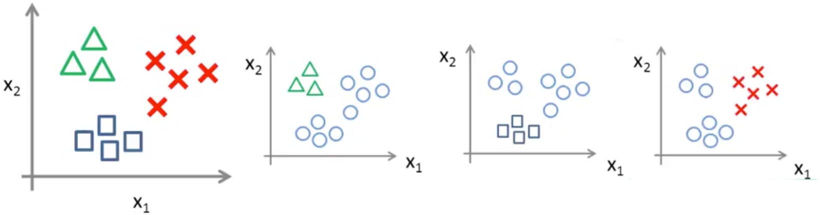
Most schemes build upon binary classification:

Multiple Classes

Most schemes build upon binary classification:

- One-vs-All: $k-1$ classifiers

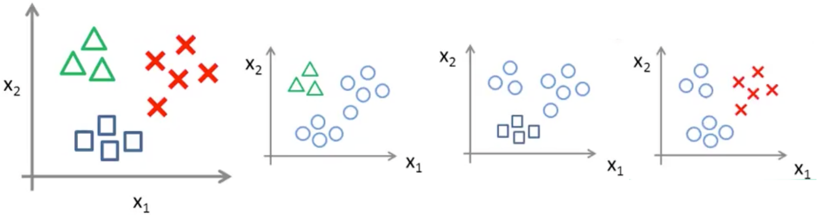
$$\boxed{y = \arg\max_i f_i(x)}$$



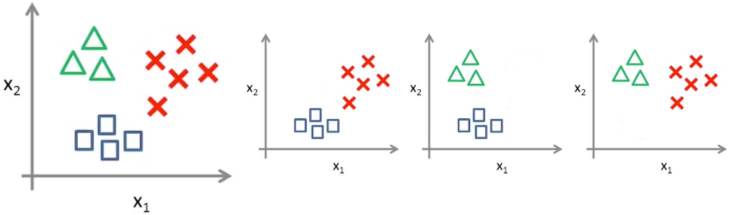
Multiple Classes

Most schemes build upon binary classification:

- One-vs-All: $k-1$ classifiers $\boxed{y = \arg\max_i f_i(x)}$



- One-vs-One: $k(k-1)/2$ classifiers $\boxed{y = \arg\max_i (\sum_j f_{ij}(x))}$



Recap

Summary of the Lecture:

Recap

Summary of the Lecture:

- SVM looks for a linear decision boundary with optimal margin

Recap

Summary of the Lecture:

- SVM looks for a linear decision boundary with optimal margin
- Support vectors are found based on the coefficients α_i

Recap

Summary of the Lecture:

- SVM looks for a linear decision boundary with optimal margin
- Support vectors are found based on the coefficients α_i
- The dual formulation allows for using the kernel trick

Recap

Summary of the Lecture:

- SVM looks for a linear decision boundary with optimal margin
- Support vectors are found based on the coefficients α_i
- The dual formulation allows for using the kernel trick
- With kernels SVM can perform non-linear classification