

Support Vector Machines (SVMs)

Stéphane Pelletier

Department of Electrical and Computer Engineering
McGill University, Montréal, Canada
`stephane.pelletier@mail.mcgill.ca`

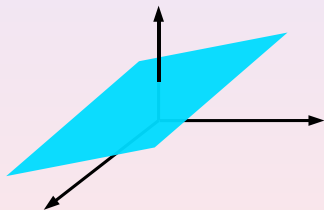
March 30, 2010

Overview of SVMs

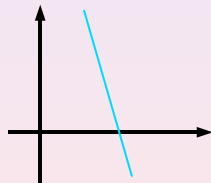
- Set of related **supervised learning** methods used for **classification** and **regression** .
- Based on simple ideas and thus provide a clear intuition of what learning from examples is about.
- Simple enough to be analyzed mathematically.
- Are not affected by local minima.
- Correspond to a **linear** method in a high-dimensional **feature space** nonlinearly related to the **input space** .
- All computations performed in input space using **kernels**
- Do not suffer from the **curse of dimensionality** .

Hyperplane concept

- A **hyperplane** is a higher-dimensional generalization of a plane in 3D.
- It has **codimension** 1, i.e., it has dimension $n - 1$ in an n -dimensional space,
- A hyperplane divides...



a space in two half-spaces



a plane in two half-planes



a line in two rays

Mathematical representation of a hyperplane

- A hyperplane in an n -dimensional space is defined by

$$x_1, \dots, x_n \mid w_1 x_1 + \dots + w_n x_n + b = 0, \quad (1)$$

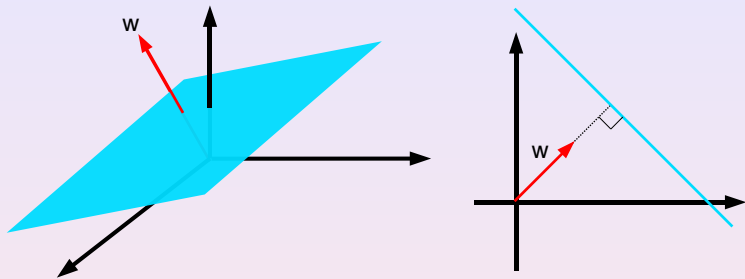
where $w_1 \dots w_n$ and b are scalar coefficients.

- Using vector notation, one can rewrite Equation 1 as

$$\mathbf{x} \mid \mathbf{w}^T \mathbf{x} + b = 0, \quad (2)$$

where $\mathbf{w} = [w_1 \ w_2 \ \dots \ w_n]^T$ and $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$.

Hyperplane examples



- w is a vector perpendicular to the hyperplane.
- The representation of a hyperplane is not unique.
- When $\|w\| = 1$, b is the distance between the hyperplane and the origin.

Hyperplane classifiers

- The **two** half-spaces defined by a hyperplane are

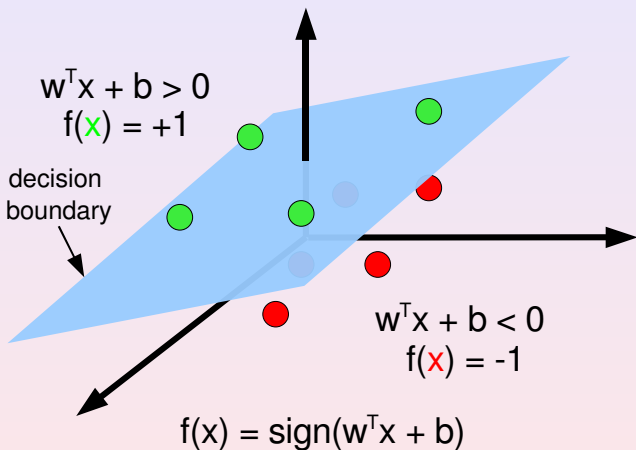
$$\mathbf{w}^T \mathbf{x} + b \leq 0 \text{ and } \mathbf{w}^T \mathbf{x} + b \geq 0.$$

- One can define a decision function $f(\mathbf{x})$ for classifying a vector $\mathbf{x} \in \mathbb{R}^n$ into one of two classes, namely -1 or 1 :

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b).$$

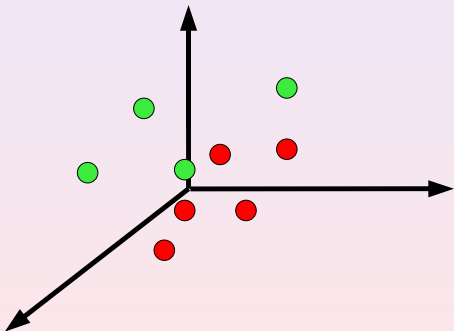
- The hyperplane $\mathbf{w}^T \mathbf{x} + b = 0$ is called the **decision boundary**.

Hyperplane classification example



Learning the classification function parameters

- Given \mathbf{w} and b , we know we can classify any point \mathbf{x} .
- Given a set of points, can we find a hyperplane that correctly classifies them?
- Motivation: learning from examples.



Pattern recognition from examples

- Suppose we have a set of p **classified** patterns

$$(\mathbf{x}_i, y_i), i \in \{1, 2, \dots, p\},$$

where \mathbf{x}_i is a n -dimensional pattern (vector) and $y_i \in \{\pm 1\}$ is its class label.

- We would like to find a function $f(\mathbf{x})$ that correctly classifies all patterns.
- This implies finding a hyperplane (i.e. \mathbf{w} and b) such that

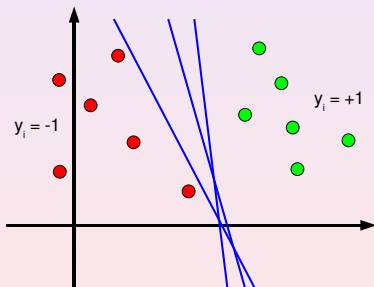
$$\begin{aligned} \mathbf{w}^T \mathbf{x}_i + b &\geq 0 && \text{if } y_i = +1, \\ \mathbf{w}^T \mathbf{x}_i + b &\leq 0 && \text{if } y_i = -1, \end{aligned}$$

which is equivalent to

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 0.$$

Choice of decision boundary

- A perfect classification is possible only if the training data is **linearly** separable, which is the case when the constraint $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 0$ is satisfied for all (\mathbf{x}_i, y_i) .
- In general, many hyperplanes satisfy this constraint.
- Which one should we choose?

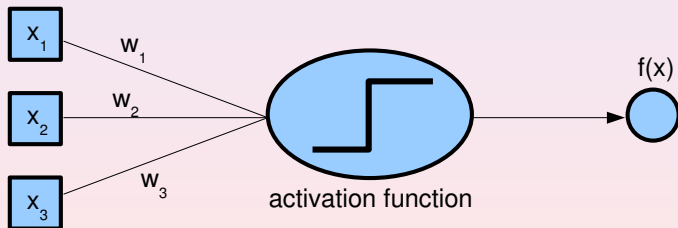


Comparison with perceptrons

- A perceptron is also a linear classifier.
- When the activation function is the sign function, we have:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b).$$

- Same function $f(\mathbf{x})$ as with hyperplanes



Perceptron learning

- The update rule for a perceptron is

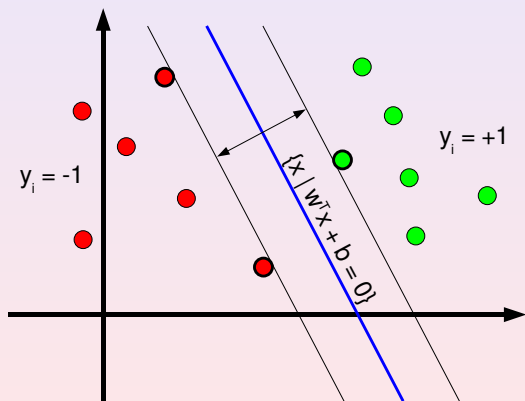
$$w_i \leftarrow w_i + \alpha \times \mathbf{x}_i(i) \times (y_i - f(\mathbf{x}_i))$$

where $\mathbf{x}_i(i)$ is element i of \mathbf{x}_i .

- When the current decision boundary correctly classifies all examples, the learning algorithm stops.
- Previous update rule converges to **any** hyperplane satisfying $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 0$ for all i .
- Decision boundary depends on **all** training patterns and initial solution.

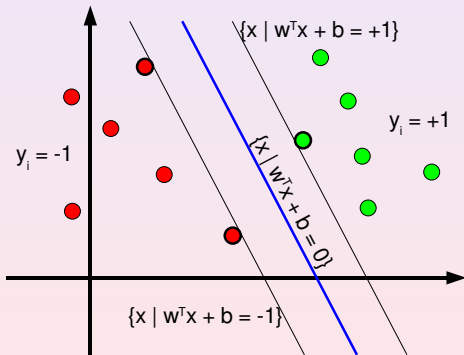
SVMs goal

- SVMs maximize distance between decision boundary and **closest** sample(s), called **support vectors** (SV).
- Only the SVs affect the location of the decision boundary.



Determining the decision boundary

- Remember: the representation of a hyperplane is not unique.
- \mathbf{w} and b are usually chosen so that SVs satisfy $|\mathbf{w}^T \mathbf{x}_i + b| = 1$.



Margin

- Let \mathbf{x}_1 and \mathbf{x}_2 be two SVs from different sets.
- From our rescaling assumption, we have

$$\mathbf{w}^T \mathbf{x}_1 + b = +1$$

$$\mathbf{w}^T \mathbf{x}_2 + b = -1,$$

- which leads to

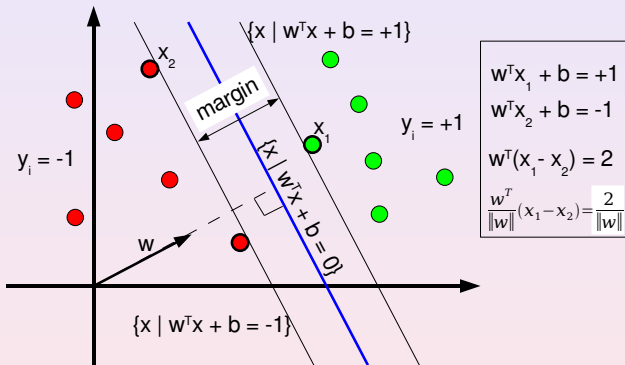
$$\mathbf{w}^T (\mathbf{x}_1 - \mathbf{x}_2) = 2$$

$$\frac{\mathbf{w}^T}{\|\mathbf{w}\|} (\mathbf{x}_1 - \mathbf{x}_2) = \frac{2}{\|\mathbf{w}\|}$$

- The quantity $\frac{2}{\|\mathbf{w}\|}$ is called the **margin**.

Geometrical interpretation of the margin

- The margin is the distance measured perpendicularly to the hyperplane between SVs from different sets.



- Generalization capabilities of SVMs rely on margin maximization.

Problem to solve

- Find \mathbf{w} and b minimizing

$$\|\mathbf{w}\|^2 = \mathbf{w}^T \mathbf{w}$$

subject to the constraints

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \text{ for all } i.$$

- Equivalent to finding coefficients α_i maximizing

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

subject to the constraints

$$\alpha_i > 0 \text{ and } \sum_i \alpha_i y_i = 0.$$

Things to know

- Single global maximum.
- \mathbf{w} depends only on SVs and has the form

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i,$$

where α_i is zero if \mathbf{x}_i is not a SV.

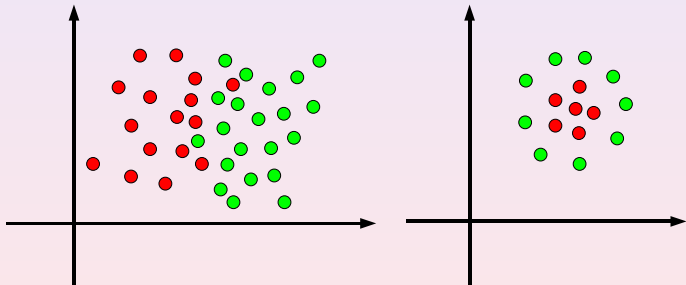
- Consequently, the decision rule $f(\mathbf{x})$ can be rewritten as:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b) = \text{sign}\left(\sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b\right).$$

- Solution process involves dot products $\mathbf{x}_i^T \mathbf{x}_j$.

Linear separability

- What if the training data is not linearly separable?
- Possible reasons:
 - mislabeling
 - nature of the problem



Soft Margin Hyperplanes

- Useful to deal with mislabeled examples.
- Split examples as cleanly as possible.
- Minimize

$$\mathbf{w}^T \mathbf{w} + C \sum_i \epsilon_i$$

subject to the constraints

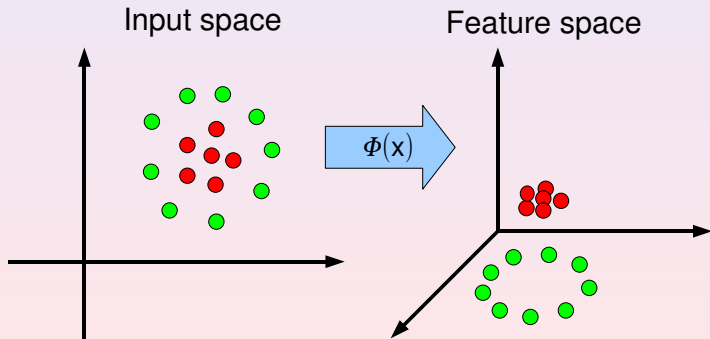
$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \epsilon_i, \epsilon_i \geq 0.$$

where ϵ_i are **slack** variables and C is a constant.

- Variables ϵ_i measure the degree of misclassification.

Input space vs Feature space

- We have an input space of size n .
- Define a **feature space** of size $m > n$.
- Map the patterns x_i into this higher-dimensional space.
- Compute the hyperplane in the feature space.



Nonlinear mapping and kernel functions

- Use a nonlinear mapping

$$\Phi : \mathcal{R}^n \rightarrow \mathcal{R}^m$$

- Dot products $\mathbf{x}_i^T \mathbf{x}_j$ are replaced with $\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$.
- If m is huge, dot products are expensive to compute.
- Fortunately, we can use **kernel** functions.
- All computations performed in input space!

Kernel function example

- Define the nonlinear mapping $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$

$$\Phi(\mathbf{x}) = [x_1^2, \sqrt{2}x_1x_2, x_2^2]^T.$$

- One can use the **polynomial** kernel

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y})^d.$$

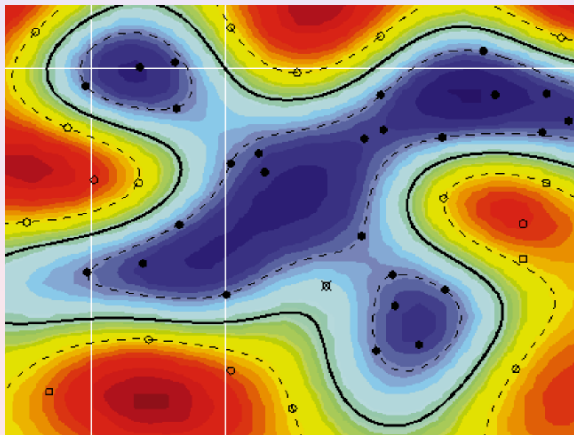
- When $d = 2$, we have

$$(\mathbf{x}^T \mathbf{y})^2 = \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right)^2 = \left(\begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix} \cdot \begin{bmatrix} y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{bmatrix} \right).$$

- All dot products $\Phi(\mathbf{x})^T \Phi(\mathbf{y})$ can be done in input space by computing $(\mathbf{x}^T \mathbf{y})^2$.

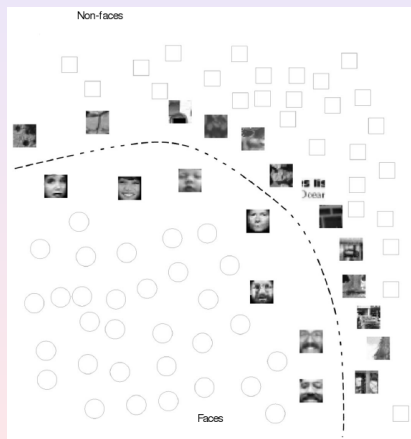
Example of a SV classifier

- Classification between circles and disks using a radial basis function kernel [Support vector machines, M.A. Hearst, IEEE Intelligent Systems, 1998].



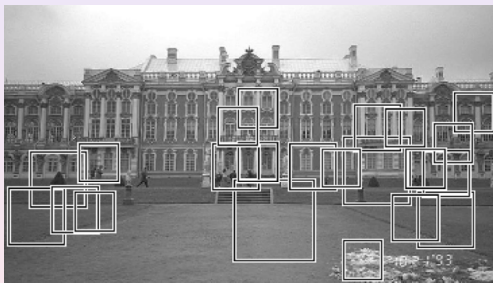
Example of SVMs applied to face detection

- Geometrical interpretation of how the SVMs separate the face and nonface classes.



Example of SVMs applied to face detection (cont.)

- A few nonface examples used for training



Example of SVMs applied to face detection (cont.)

- Face detection in a **new** image

