Lecture on Support Vector Machines (SVM)

Stéphane Pelletier

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

March 29, 2007

Overview of SVMs

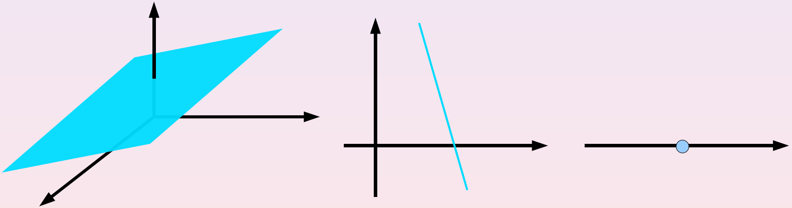
- Set of related *supervised learning* methods used for *classification* and *regression*.
- Based on simple ideas and thus provide a clear intuition of what learning from examples is about.
- Are not affected by local minima.
- Do not suffer from the *curse of dimensionality*.
- Can lead to high performances in practical applications.

Overview of SVMs (cont.)

- Simple enough to be analyzed mathematically (unlike neural net).
- Correspond to a *linear* method in a high-dimensional *feature space* nonlinearly related to input space.
- But! Does not involve computation in this high-dimensional feature space.
- By using *kernels*, all computations can be performed in input space.

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

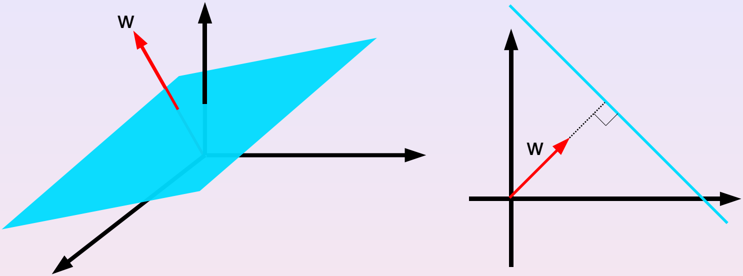
$$x_0, \ldots, x_{n-1} \mid w_0 x_0 + \ldots + w_{n-1} x_{n-1} + b = 0, \tag{1}$$

  where $w_0 \ldots w_{n-1}$ and b are scalar coefficients.
- Using vector notation, one can rewrite Equation 1 as

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

  where $\mathbf{w} = [w_0 \; w_1 \; \ldots \; w_{n-1}]^T$ and $\mathbf{x} = [x_0 \; x_1 \; \ldots \; x_{n-1}]^T$.

Hyperplane examples



- $\mathbf{w}$ is a vector perpendicular to the hyperplane.
- when $||\mathbf{w}|| = 1$, b is the distance between the hyperplane and the origin.
- Is the representation of a hyperplane unique?

Hyperplane classifiers

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

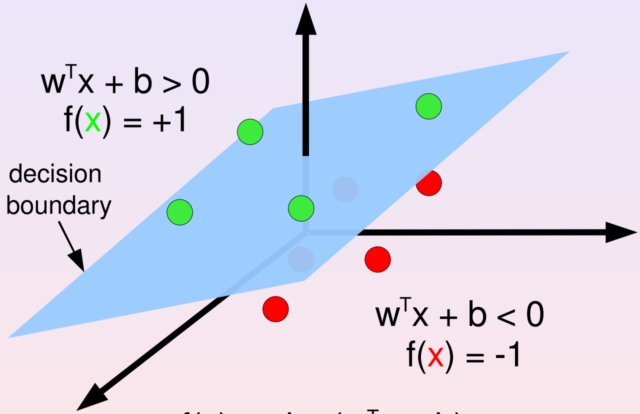
$$\mathbf{w}^T\mathbf{x} + b \leq 0 \text{ and } \mathbf{w}^T\mathbf{x} + b \geq 0.$$

- These can be used to define a function $f : \Re^n \rightarrow \{\pm 1\}$ for classifying a vector $\mathbf{x} \in \Re^n$ 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*.

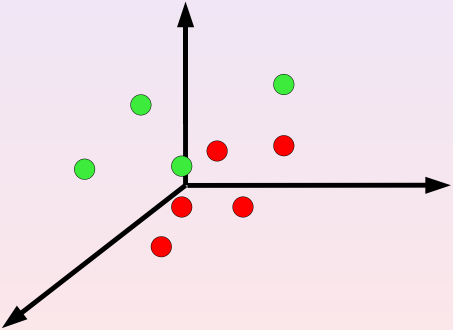
Hyperplane classifiers (cont.)



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

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 (i.e. $\mathbf{w}$ and b) that correctly classifies them?
- Motivation: learning from examples.



Pattern recognition from examples

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

$$(\mathbf{x}_i, y_i),\ i \in \{0, 1, \ldots, p-1\},$$

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 $f : \Re^n \rightarrow \{\pm 1\}$ that correctly classifies all patterns.
- This implies finding a hyperplane (i.e. $\mathbf{w}$ and b) such that

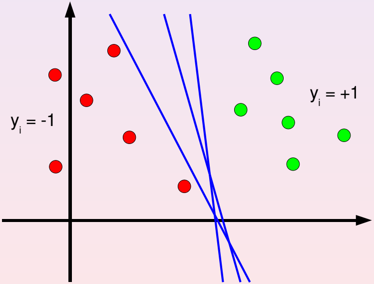
$$\begin{aligned} \mathbf{w}^T\mathbf{x}_i + b \geq 0 \quad & \text{if } y_i = +1, \\ \mathbf{w}^T\mathbf{x}_i + b \leq 0 \quad & \text{if } y_i = -1, \end{aligned}$$

which is equivalent to

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

Classification example

- 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?

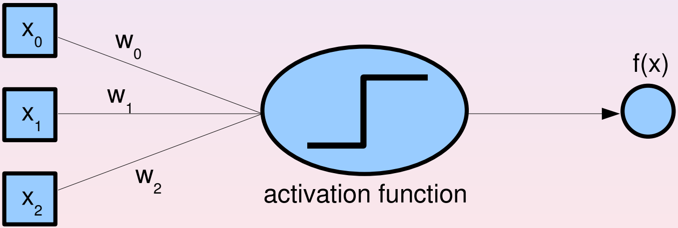


Perceptron

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

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

- Seems related to a hyperplane, no?



Perceptron learning

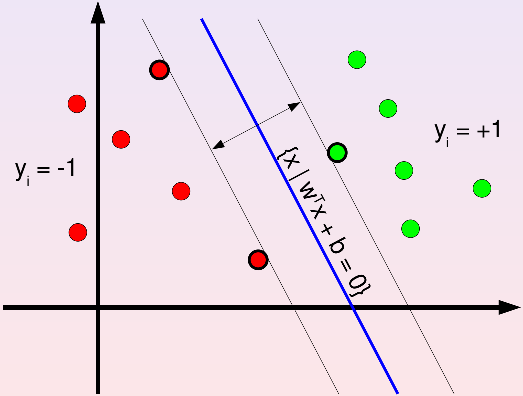
- The update rule for a perceptron is

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

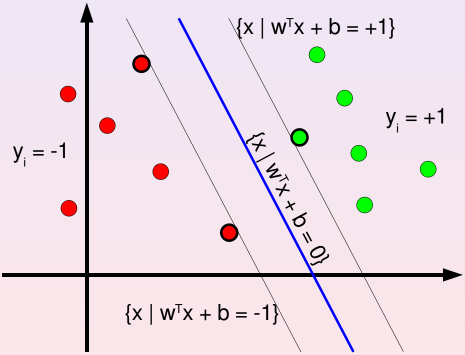
  where $\mathbf{x}_i(i)$ is element i of $\mathbf{x}_i$.
- When the current decision boundary correctly classifies all examples, the learning algorithm stops.
- Previous update rule converges to *any* hyperplane satisfying $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 0$ for all i.
- Decision boundary depends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

SVM goal

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



- As seen before, the representation of a hyperplane is not unique.
- We rescale $\mathbf{w}$ and b such 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

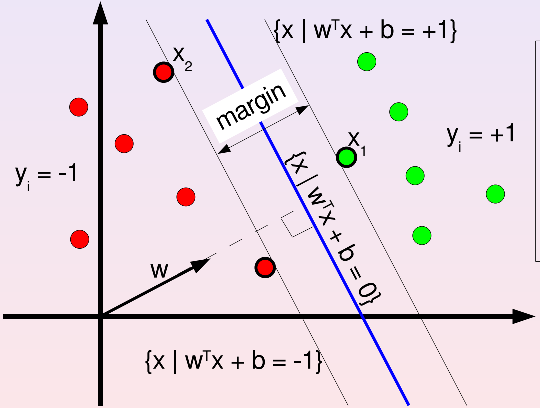
$$\mathbf{w}^T(\mathbf{x}_1 - \mathbf{x}_2) = 2$$

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

- 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.



$$w^T x_1 + b = +1$$
$$w^T x_2 + b = -1$$
$$w^T(x_1 - x_2) = 2$$
$$\frac{w^T}{\|w\|}(x_1 - x_2) = \frac{2}{\|w\|}$$

Problem to solve

- Minimize

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

subject to the constraints

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

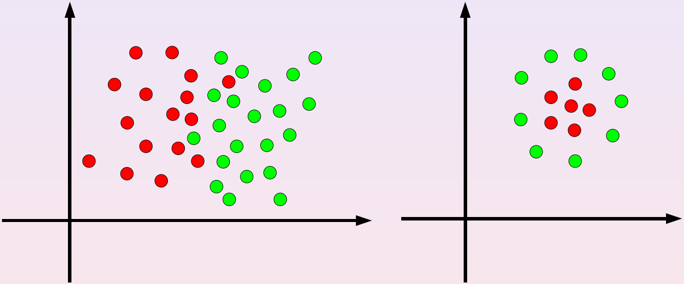
- Solution to this constrained optimization problem can be found using method of Lagrange multipliers.
- It has the form

$$\mathbf{w} = \sum_j v_j \mathbf{x}_j,\ j < p$$

where $\mathbf{x}_j$ are the support vectors.

Linear separability

- What if the training data is not linearly separable?

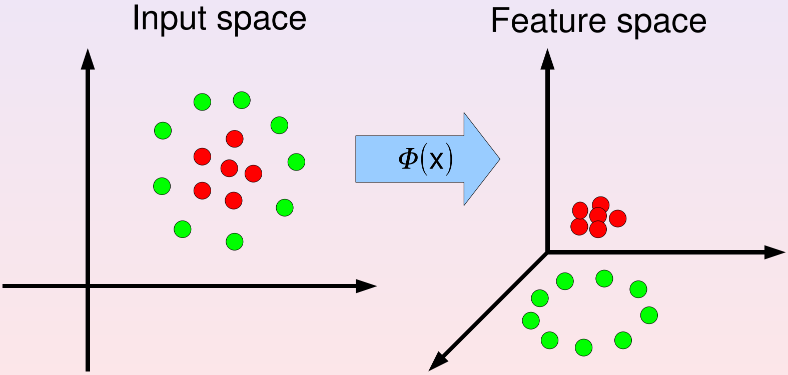


Soft Margin Hyperplane

- Minimize
$$\mathbf{w}^T\mathbf{w} + \lambda \sum_i \epsilon_i^\delta, \ \delta \geq 0$$
subject to the constraints
$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \epsilon_i, \ \epsilon_i \geq 0.$$
where ϵ_i allows for some error.
- This is not a convex optimization problem.

Input space vs Feature space

- Map the input vectors nonlinearly into a higher-dimensional *feature space*.
- Compute the hyperplane in this feature space.



Kernel functions

- Use a nonlinear map

$$\Phi : \Re^n \rightarrow \Re^m, m > n$$

- If m is huge, the dot product in the feature space can be very expensive to compute.
- Fortunately, we can use *kernel* functions.
- All computations performed in input space!

Kernel function example

- One can use the *polynomial* kernel

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

- When $d = 2$, we have

$$(\mathbf{x}^T \mathbf{y})^2 = \left(\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \cdot \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} \right)^2 = \left(\begin{bmatrix} x_0^2 \\ \sqrt{2} x_0 x_1 \\ x_1^2 \end{bmatrix} \cdot \begin{bmatrix} y_0^2 \\ \sqrt{2} y_0 y_1 \\ y_1^2 \end{bmatrix} \right).$$

  which defines

$$\Phi(\mathbf{x}) = [x_0^2,\ \sqrt{2} x_0 x_1,\ x_1^2]^T$$

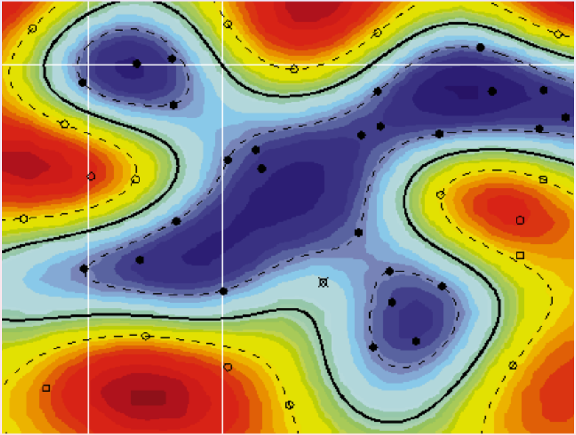
- All dot products can be done in 2D (input) space instead of 3D (feature) space.

Generalization hability

- Generalization can arise from
  - small dimensionality of feature space,
  - large separating margin,
  - small number of support vectors.
- SVM rely on the last two.

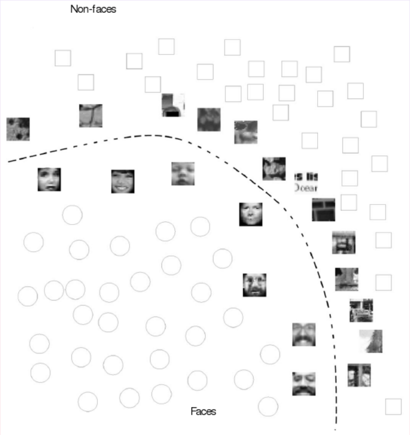
Example of a SV classifier

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



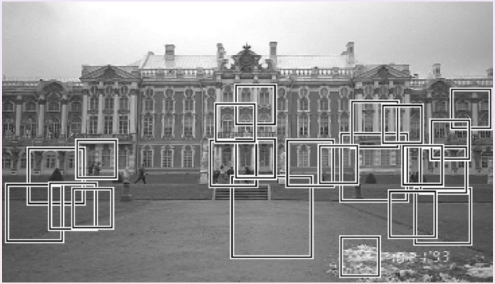
Example of SVM applied to face detection

- Geometrical interpretation of how the SVM separates the face and nonface classes.



Example of SVM applied to face detection (cont.)

- A few nonface examples used for training



Example of SVM applied to face detection (cont.)

- Face detection in a *new* image