Statistical Machine Learning from Data

Support Vector Machines

Samy Bengio

IDIAP Research Institute, Martigny, Switzerland, and
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
bengio@idiap.ch
http://www.idiap.ch/~bengio

January 18, 2006

Setup

- Training set:

$$(x_i, y_i)_{i=1\ldots n} \in \mathbb{R}^d \times \{-1, 1\}$$

- We would like to find an hyperplane

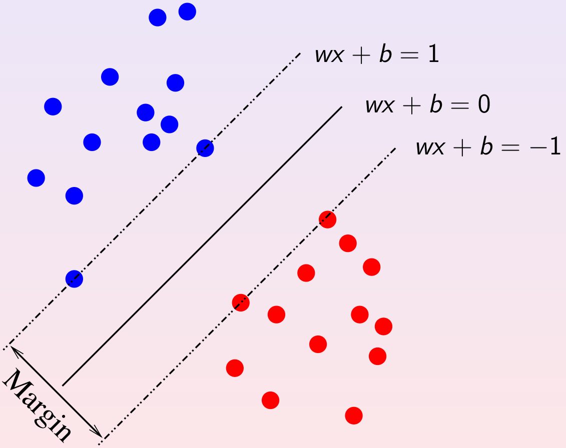
$$wx + b = 0 \quad (w \in \mathbb{R}^d,\ b \in \mathbb{R})$$

which separates the two classes.

The Margin

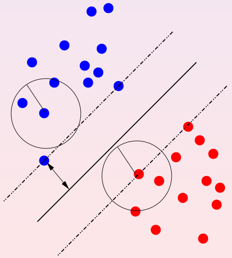
- Let d_+ be the shortest distance from the hyperplane to the closest positive example.
- Let d_- be the shortest distance from the hyperplane to the closest negative example.
- Define the margin of the hyperplane to be $d_+ + d_-$.
- The simplest SVM looks for the separating hyperplane with the largest margin.

SVMs and the Margin (Graphical View)



Why is it Good to Maximize the Margin?

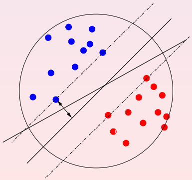
- There are several justifications to favor large margins... for instance:
- If training and test data come from the same distribution and all test data are within some Δ distance from the training points...
- Then a margin $(2 \cdot \Delta)$ is enough to classify all points:



Why is it Good to Maximize the Margin?

- If all points lie at a distance of at least Δ from the separator, and all points are in a bounded sphere, then a small perturbation of the definition of the separator will not hurt.
- Hence one can use less bits to encode the separating hyperplane.
- This is related to the Minimum Description Length principle:

> *The best description of the data, in terms of generalization error, should be the one that requires the fewest bits to store.*



Formulation of the SVM Problem

- We can define the following constraints:

$$wx_i + b \geq +1 \text{ for } y_i = +1$$

$$wx_i + b \leq -1 \text{ for } y_i = -1$$

- They can be combined as follows:

$$y_i(wx_i + b) - 1 \geq 0 \quad \forall i$$

- One can show that $d_+ = d_- = \frac{1}{\|w\|}$ with $\|w\|$ the Euclidean norm of w. Hence, the margin is simply $\frac{2}{\|w\|}$.
- So we would like to minimize:

$$\frac{\|w\|^2}{2}$$

Under the constraints:

$$y_i(wx_i + b) - 1 \geq 0 \quad \forall i$$

A Constrained Optimization Problem

- Normal way to solve an optimization problem with cost $C(w)$ and parameter w: set $\frac{\partial C}{\partial w} = 0$. Example:

$$\text{minimize } C(w) = \frac{w^2}{2} - 3w$$

hence

$$\frac{\partial C}{\partial w} = w - 3 = 0 \Longrightarrow w = 3$$

- When there are constraints $c_i \geq 0$, use Lagrange multipliers and verify the solution with the Karush-Kuhn-Tucker (KKT) conditions.

A Constrained Optimization Problem (con't)

- Form the Lagrangian by subtracting one term for each constraint $c_i \geq 0$, weighted by a positive Lagrange multiplier:

$$L(w, \alpha) = C(w) - \sum_i \alpha_i c_i$$

- We must now minimize L with respect to w subject to
  - $\frac{\partial L}{\partial \alpha_i} = 0$
  - $\alpha_i \geq 0 \;\; \forall i$
- We can equivalently solve the dual problem: maximize L with respect to α subject to
  - $\frac{\partial L}{\partial w} = 0$
  - $\alpha_i \geq 0 \;\; \forall i$
- The general problem is to find a saddle point:

$$\max_\alpha \min_w L(w, \alpha)$$

Lagrangian Formulation for SVMs

- We introduce a Lagrange multiplier $\alpha_i, i = 1, \ldots, n$, one for each inequality constraint:

$$L(w, b, \alpha) = \frac{\|w\|^2}{2} - \sum_{i=1}^{n} \alpha_i (y_i(wx_i + b) - 1)$$

- L has to be minimized w.r.t. the primal variables w and b and maximized w.r.t. the dual variables α_i.
- At the extremum, we have

$$\frac{\partial L}{\partial w} = 0 \text{ and } \frac{\partial L}{\partial b} = 0$$

Solve the Lagrangian

- We have:

$$L = \frac{\|w\|^2}{2} - \sum_{i=1}^{n} \alpha_i (y_i (w x_i + b) - 1)$$

- We want $\frac{\partial L}{\partial w} = 0$:

$$\begin{aligned} \frac{\partial L}{w} &= w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \\ w &= \sum_{i=1}^{n} \alpha_i y_i x_i \end{aligned}$$

- We want $\frac{\partial L}{\partial b} = 0$:

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

Substituting to get the Dual

$$
\begin{aligned}
L &= \frac{\|w\|^2}{2} - \sum_{i=1}^{n} \alpha_i (y_i (w x_i + b) - 1) \\
&= \frac{\sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j}{2} - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\sum_{j=1}^{n} \alpha_j y_j x_j x_i + b \right) - 1 \right) \\
&= \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j x_i - b \sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i \\
&= -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j + \sum_{i=1}^{n} \alpha_i \\
L &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j
\end{aligned}
$$

The Dual Formulation

- We need to maximize the following:

$$L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j$$

- subject to

$$\alpha_i \geq 0, \quad \forall i$$

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

- This can be solved using classical quadratic programming optimization packages, based for instance on constrained gradient descent.

The KKT Conditions

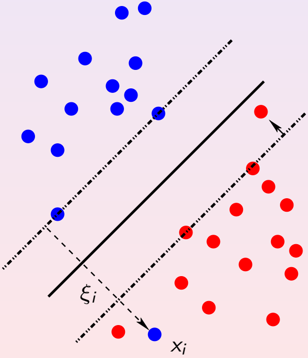
The following KKT conditions are satisfied at the solution.

- $\dfrac{\partial L}{\partial w} = w - \displaystyle\sum_{i=1}^{n} \alpha_i y_i x_i = 0$
- $\dfrac{\partial L}{\partial b} = \displaystyle\sum_{i=1}^{n} \alpha_i y_i = 0$
- $y_i(wx_i + b) - 1 \geq 0, \quad \forall i$
- $\alpha_i \geq 0, \quad \forall i$
- $\alpha_i(y_i(wx_i + b) - 1) = 0, \quad \forall i$

This can be used to estimate b after w has been found during training.

A Bug

This minimization problem does not have any solution if the two classes are not separable.



Fixing The Bug: "Soft" Margin

- Relax the constraints: use a soft margin instead of a hard margin.
- We would like to minimize:

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

Under the constraints:

$$y_i(wx_i + b) \geq 1 - \xi_i \quad \forall i$$

$$\xi_i \geq 0 \quad \forall i$$

The Non-Separable Dual Formulation

- We need to maximize the following:

$$L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j$$

- subject to

$$0 \leq \alpha_i \leq C, \quad \forall i, \ldots, n$$

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

- We then obtain w and b as follows:

$$w = \sum_i \alpha_i y_i x_i$$

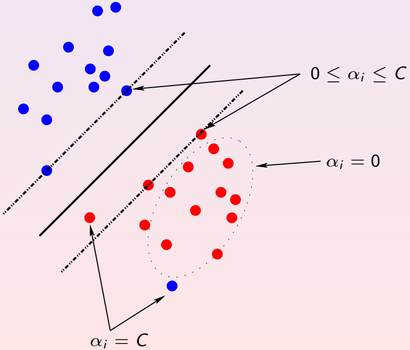
$$\alpha_i [1 - \xi_i - y_i (w x_i + b)] \; = \; 0$$

Support Vector Terminology

- Note that the decision function can be rewritten as:

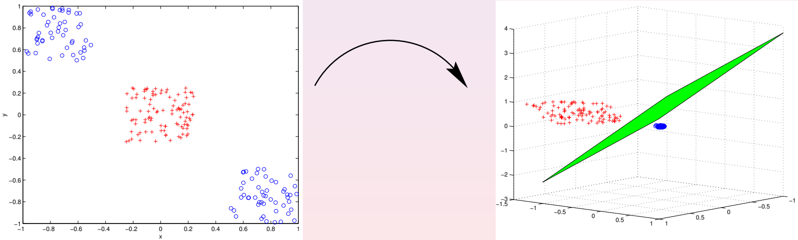
$$\hat{y} = \text{sign}\left(\sum_i \alpha_i y_i \, x_i x + b\right)$$

- Training examples x_i with $\alpha_i \neq 0$ are support vectors.



Non-Linear SVMs

- Project the data into a higher dimensional space: it should be easier to separate the two classes.
- Given a function $\phi : \mathbb{R}^d \to F$, work with $\phi(x_i)$ instead of working with x_i.



The Kernel Trick

- Note that we have only dot products $\phi(x_i)\phi(x_j)$ to compute.
- Unfortunately, this could be very expensive in a high dimensional space.
- Use instead a kernel: a function $k(x, z)$ which represents a dot product in a "hidden" feature space.

$$k(x, z) = \phi(x)\phi(z)$$

- Example: instead of

$$\phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

use

$$k(x, z) = (xz)^2$$

Common Kernels

- Polynomial:

$$k(x, z) = (u\,xz + v)^p \quad (u \in \mathbb{R},\ v \in \mathbb{R},\ p \in \mathbb{N}^*_+)$$

- Gaussian:

$$k(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right) \quad (\sigma \in \mathbb{R}^*_+)$$

- ⚠ The function

$$k(x, z) = \tanh(u\,xz + v)$$

is not a kernel!

Mercer's Condition

- Which functions are kernels???
- There exists a mapping ϕ and an expansion

$$k(x, z) = \sum_i \phi(x)_i \phi(z)_i$$

if and only if, for any $g(x)$ such that

$$\int g(x)^2 dx \text{ is finite}$$

then

$$\int k(x, z) g(x) g(z) dx dz \geq 0$$

- In practice, a kernel gives rise to a positive semi-definite matrix (example a symmetric similarity matrix).

Final Solution

- Maximize

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

under the constraints

$$0 \ \leq \ \alpha_i \ \leq \ C \quad \text{and} \quad \sum_i \alpha_i y_i = 0$$

- For $0 \ < \ \alpha_i \ < \ C$, compute b using

$$1 - y_i \left[\sum_j \alpha_j y_j \, k(x_j, \, x_i) + b \right] \ = \ 0$$

- Decision function: $\hat{y} = \text{sign} \left(\sum_i \alpha_i y_i \, k(x_i, x) + b \right)$

The KKT Conditions

The following KKT conditions are satisfied at the solution.

- $y_i(wx_i + b) - 1 \geq 0, \;\; \forall i$
- $0 < \alpha_i < C, \;\; \forall i \;\; s.t. \;\; y_i(wx_i + b) = 1$
- $\alpha_i = C, \;\; \forall i \;\; s.t. \;\; y_i(wx_i + b) \leq 1$
- And note that $\alpha_i = 0$ for all non-support vectors.

Facts to Remember

- SVMs maximize the margin (*in the feature space*)
- Use the soft margin trick
- Project the data into a higher dimensional space for non-linear relations
- Kernels simplify the computation
- A Lagrangian method leads to a "nice" quadratic optimization problem under constraints.

SVMs in Practice

- In order to tune the capacity, the kernel is the most important parameter to choose.
  - Polynomial kernel: increasing the degree will increase the capacity.
  - Gaussian kernel: increasing σ will decrease the capacity.
- Tune C, the trade-off between the margin and the errors.
  - For non-noisy data sets, C usually has not much influence.
  - Carefully choose C for noisy data sets: small values usually give better results.

1. Linear Support Vector Machines
2. Kernels for Non-Linear Support Vector Machines
3. **Training Support Vector Machines**
4. Other Kernel Methods

Complexity of the QP Problem

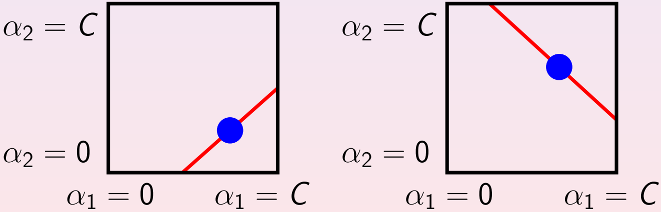
- You need n^2 in memory just to keep the kernel matrix!
- Naive optimization technique would then be at least n^3 to n^4.
- What about datasets of 100000 examples or more???
- Various approaches have been proposed:
  - Chunking: at each step, solve the QP problem with all non-zero α_i from previous step, and the M worst examples violating the KKT conditions.
  - Decomposition: Solve a series of smaller QP problems, where each one adds an example that violates the KKT conditions.
  - Sequential Minimal Optimization (SMO): solve the smallest optimization problem at each iteration.

SMO Framework

- At every step, choose two Lagrange multipliers α_i to jointly optimize, with at least one violating the KKT conditions.

  there are several tricks to select the most violating ones...

- Find the optimal value for these two α_i and update the SVM model.



$$y_1 \neq y_2 \longrightarrow \alpha_1 - \alpha_2 = k \qquad y_1 = y_2 \longrightarrow \alpha_1 + \alpha_2 = k$$

This procedure converges to the optimum.

Other Kernel Methods

A Zoo of Kernel Methods in the Literature:

- Control explicitely the number of SVs: ν-SVMs
- For regression problems: Support Vector Regression
- For density estimation or representation: Kernel PCA
- For generative models: Fisher kernel
- For discrete sequences: String kernel
- ...

How to design a kernel? Prior knowledge!!!

- choosing a similarity measure between 2 examples in the data
- choosing a linear representation of the data
- choosing a feature space for learning