Support Vector Machines

Approach:

- Project instances into high dimensional space
- Learn linear separators with maximum margin
- Learning as optimizing bound on expected error

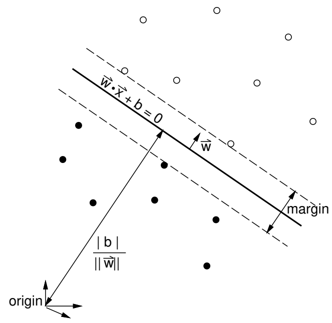
Positives:

- Good empirical results on character recognition, text classification, ...
- PAC-style theoretical grounding
- Appears to avoid overfitting in high dimensional spaces
- Global optimization method, no local optima

Negatives:

- Applying trained classifier can be expensive

Linear Separator



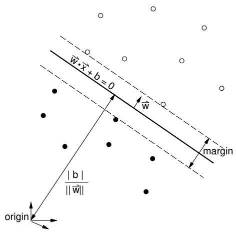
Want a linear separator. Can view this as constraint satisfaction problem:

$$\begin{array}{ll} \vec{x}_i \cdot \vec{w} + b \geq +1 & \text{if } y_i \equiv f(\vec{x}_i) = +1 \\ \vec{x}_i \cdot \vec{w} + b \leq -1 & \text{if } y_i = -1 \end{array}$$

Equivalently,

$$y_i(\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0, (\forall i)$$

Linear Separator



We'd like the hyperplane with maximum margin

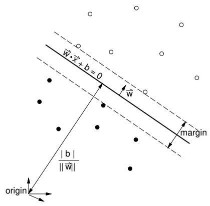
- size of margin is $\frac{2}{||\vec{w}||}$

So view our problem as a constrained optimization problem:

Minimize $||\vec{w}||^2$, subject to

$$y_i(\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0, (\forall i)$$

Linear Separator



- margin determined by just a few examples
  - call these *support vectors*
  - can define separator in terms of support vectors

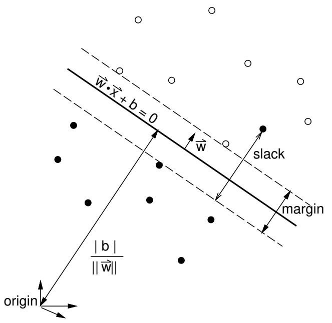
$$f(\vec{x}) \leftarrow sgn(\sum_{s_i \in \text{support vectors}} w_i \vec{s}_i \cdot \vec{x} + b)$$

- Can bound expected true error of learned hyperplane h by

$$E[error_{\mathcal{D}}(h)] \leq \frac{E[\textit{number of support vectors}]}{m}$$

Expectation on left is over training sets of size $m-1$. On right, is over all training sets of size m

Non-Separable Training Sets



Add a "slack variable" $\xi_i \geq 0$ for each $\langle x_i, y_i \rangle$. New optimization problem is

Minimize $||\vec{w}||^2 + C(\Sigma_i \xi_i)^k$, subject to

$$\begin{array}{ll} \vec{x_i} \cdot \vec{w} + b \geq +1 - \xi & \text{if } y_i = +1 \\ \vec{x_i} \cdot \vec{w} + b \leq -1 + \xi & \text{if } y_i = -1 \end{array}$$

C picked by hand

NonLinear SVMs

Suppose we have instance space $X = \Re^{n_1}$, need nonlinear separator.

$\rightarrow$ project X into some higher dimensional space $X' = \Re^{n_2}$ where data will be linearly separable

- let $\Phi : X \rightarrow X'$ be this projection.

Interestingly,

- Training depends only on dot products of form $\Phi(\vec{x_i}) \cdot \Phi(\vec{x_j})$
- So we can train in $\Re^{n_2}$ with same computational complexity as in $\Re^{n_1}$, provided we can find a function K such that $K(\vec{x_i}, \vec{x_j}) = \Phi(\vec{x_i}) \cdot \Phi(\vec{x_j})$
- Classifying new $\vec{x}$ requires calculating sign of

$$f(\vec{x}) \leftarrow \sum_{s_i \in \text{support vectors}} w_i y_i K(\vec{s_i}, \vec{x}) + b$$

NonLinear Support Vector Machines

Example1: $X = \Re^2$, $X' = \Re^3$

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

Then we can solve optimization problem using

$$K(\vec{x_i}, \vec{x_j}) = \Phi(\vec{x_i}) \cdot \Phi(\vec{x_j}) = (\vec{x_i} \cdot \vec{x_j})^2$$

NonLinear Support Vector Machines

Example2: $X = \Re^n, X' = \Re^\infty$

We can use

$$K(\vec{x_i}, \vec{x_j}) = e^{-||\vec{x_i} \cdot \vec{x_j}||^2 / 2\sigma^2}$$

... which corresponds to radial basis function network with Gaussian kernel function!

$$f(\vec{x}) \leftarrow \sum_{s_i \in \text{support vectors}} w_i y_i K(\vec{s_i}, \vec{x}) + b$$

Note SVM automatically chooses RBF weights w_i, Gaussian centers $\vec{s_i}$, number of centers, and threshold b. Just not σ ...

NonLinear Support Vector Machines

Other kernel functions that have been used:

- Polynomial classifier of degree p

$$K(\vec{x_i}, \vec{x_j}) = (\vec{x_i} \cdot \vec{x_j} + 1)^p$$

- Gaussian radial basis function classifier

$$K(\vec{x_i}, \vec{x_j}) = e^{-||\vec{x_i} \cdot \vec{x_j}||^2 / 2\sigma^2}$$

- This one doesn't quite have a Φ

$$K(\vec{x_i}, \vec{x_j}) = tanh(\kappa \vec{x_i} \cdot \vec{x_j} - \delta)$$

Which Φ to Choose?

From PAC theory (Vapnik, 1995) we know that with probability $(1 - \delta)$

$$err_{\mathcal{D}} \leq err_D + \sqrt{\frac{VC(H)(log(2m/VC(H)) + 1) - log(\delta/4)}{m}}$$

- $err_{\mathcal{D}}$ is true error of h
- err_D is error of h on training set D
- m is number of training examples in D
- $VC(H)$ is VC dimension of hypothesis space H

So let us select Φ that minimizes this expression (called Structural Risk Minimization principle)

- trades off $VC(H)$ and $err_D(h)$
- similar to Min Description Length methods

What is VC dim of Separators with Margins?

Consider Gap Tolerant classifiers that

- Require a minimum margin M
- Classify only examples outside margin and inside sphere of diameter D

VC dim of Gap Tolerant classifiers is at most $1 + min(d, D^2/M^2)$

- d is dimension of $X' = \Re^d$
- M is min margin allowed by classifier
- D is diameter in instance space of classifier

suggests choosing the Φ_i that minimizes structural risk, where substitute $1 + min(d, D^2/M^2)$ for VC dim

Summary: Support Vector Machines

Learn linear separators (e.g., perceptrons)

- Pick separator that maximizes margin
- Use slack parameters ξ_i to accomodate unseparable data
- Can write separating plane in terms of support vectors

Learning non-linear functions

- Project instance space X into higher dimension X'
- Use kernel functions for efficiency (to train directly in X)
- Choose hypothesis (including choice of Φ) by minimizing total "risk"

Further Reading

The idea of maximizing the margin goes back to work by Vapnik published in 1982. Support Vector Machines were first introduced in [Cortes and Vapnik, 1995]. An excellent SVM tutorial is available online [Burgess 1998]. A new edited collection of articles is available in [Scholkopf et al., 1998]. SVM code may be downloaded from the site http://www-ai.informatik.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html

- Burgess, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Boston, MA: Kluwer Academic Publishers, to appear. http://svm.research.bell-labs.com/SVMdoc.html
- Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273 - 297.
- Joachims, T., (1997). Text categorization with support vector machines, *Proceedings of the 1997 European Conference on Machine Learning (ECML)*, http://www-ai.informatik.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html.
- Scholkopf, B., Burges, C., & Smola, A. (eds.) (1998). *Advances in Kernel Methods - Support Vector Learning*, MIT-Press.
- Vapnik, V. (1995) *The nature of statistical learning theory*, New York: Springer.