About this class

Separating hyperplanes (again)

The perceptron algorithm

The perceptron convergence theorem

Maximum margin classifiers

The Perceptron

Weight vector $W =< w_0, w_1, \ldots, w_n >$ such that for an input $x_1, \ldots x_n$, predict $Y = 1$ if $w_0 + \sum_{i=1}^{n} w_i x_i > 0$ and $Y = -1$ (equivalently 0, but it's easier to think about it this way) otherwise

For notational convenience add an additional imaginary $x_0 = 1$ for every input so now the classifier is sgn$(W.X)$

Perceptron update rule:

$$w_i \leftarrow w_i + \Delta w_i$$

where

$$\Delta w_i = \eta(Y - W.X)x_i$$

For the perceptron criterion, η does not matter

Decision surface is a hyperplane. w_0 affects the distance from origin, but not the angle. The rest of w is perpendicular to the separating hyperplane

Similar to logistic regression – historically very important!

The Perceptron Convergence Theorem

Using capital letters for the entire vector, let R be $\max_t ||X_t||$

Suppose we repeatedly iterate through the data and perform the perceptron criterion update

Suppose the data are linearly separable. That is

$$\exists W^* Y_t * (W^*.X_t) = 1$$

The perceptron algorithm will converge to a linear separator

We have $W_{t+1} = W_t + (X_t.Y_t)$

After running the algorithm for some time, let's say that input (X_n, Y_n) has been misclassified τ_n times

$$W = \sum_n \tau_n X_n Y_n$$

$$\Rightarrow W^{*T}.W = \sum_n \tau_n W^{*T} X_n Y_n$$

$$\geq \tau \min_n W^{*T} X_n Y_n$$

where τ is now the total number of mistakes made. So we have established the boundedness below of the projection of W on W^*

Now the next step is to show boundedness above of W after τ errors

$$||W_{\tau+1}||^2 = ||W^\tau + X_n Y_n||^2$$

$$= ||W_n||^2 + 2Y_n(W_n.X) + ||X_n||^2$$

Since we made a mistake, $Y_n(W_n.X) < 0$, and we have $||X_n||^2 < R^2$, so

$$||W_{\tau+1}||^2 < ||W_n||^2 + R^2$$

Therefore, after τ updates

$$||W||^2 \leq \tau R^2$$

So the length of W increases no faster than $\sqrt{\tau}$, but it increases at least as fast as τ. Therefore, it must stop changing for sufficiently large τ

Things to think about:

1. Decrease in error is not monotonic!

2. Basic generalization result – this algorithm will converge after a finite number of errors. Therefore it *must* be able to generalize (assuming concept is in the hypothesis space)

Maximizing the Margin

Picture of large and small margin hyperplanes

Intuition: large margin condition acts as a regularizer and should generalize better

The Support Vector Machine (SVM) makes this formal. Not only that, it is amenable to the kernel trick which will allow us to get much greater representational power!

Deriving the SVM

(Derivation based on Ryan Rifkin's slides in MIT 9.520 from Spring 2003)

Assume we classify a point x as sgn$(w.x)$

Let x be a datapoint on the margin, and z the point on the separating hyperplane closest to x

We want to maximize $||x - z||$

For some k (assumed positive)

$$w.x = k$$

$$w.z = 0$$

$$\Rightarrow w.(x - z) = k$$

Since $x - z$ is parallel to w (both perpendicular to the separating hyperplane)

$$k = w.(x - z)$$

$$\Rightarrow k = ||w||||x - z||$$

$$\Rightarrow ||x - z|| = \frac{k}{||w||}$$

So now, maximizing $||x - z||$ is equivalent to minimizing $||w||$

We can fix $k = 1$ (this is just a rescaling)

Now we have an optimization problem:

$$\min_{w \in \mathbb{R}^n} ||w||^2$$

subject to:

$$y_i(w.x_i) \geq 1, i = 1, \ldots, l$$

Can be solved using quadratic programming

Think about this expression in terms of training set error and inductive bias!

Typically we also use a bias term to shift the hyperplane around (so it doesn't have to pass through the origin) Now $f(x) = \text{sgn}(w.x + b)$

When a Separating Hyperplane Does Not Exist

We introduce *slack variables*. The new optimization problem becomes

$$\min_{w \in \mathbb{R}^n, \xi \in \mathbb{R}^l} C \sum_{i=1}^{l} \xi_i + \frac{1}{2}||w||^2$$

subject to:

$$y_i(w.x_i + b) \geq 1 - \xi_i, i = 1, \ldots, l$$

$$\xi_i \geq 0, i = 1, \ldots, l$$

Now we are trading the error off against the margin

The Dual Formulation

$$\max_{\alpha \in \mathbb{R}^l} \sum_{i=1}^{l} \alpha_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i . x_j)$$

subject to:

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

$$0 \leq \alpha_i \leq C, i = 1, \ldots, l$$

The hypothesis is then:

$$f(x) = \text{sgn}(\sum_{i=1}^{l} \alpha_i y_i (x . x_i))$$

Sparsity: it turns out that:

$$y_i f(x_i) > 1 \Rightarrow \alpha_i = 0$$

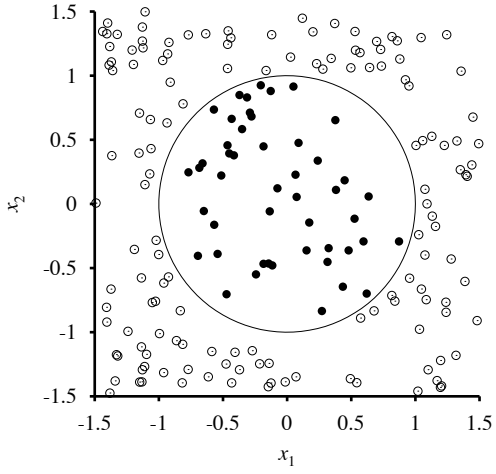
$$y_i f(x_i) < 1 \Rightarrow \alpha_i = C$$

This allows for more efficient solution of the QP than we could get otherwise

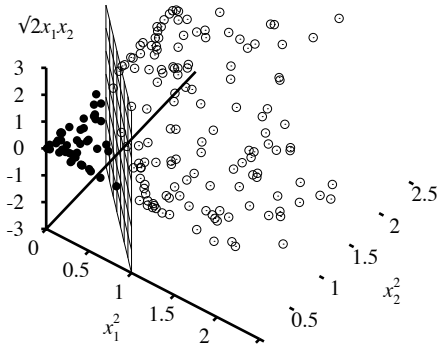
The Kernel Trick

The really nice thing: optimization depends only on the dot product between examples.

An example from Russell & Norvig



Now suppose we go from representation $x =< x_1, x_2 >$ to representation $F(x) =< x_1^2, x_2^2, \sqrt{2}x_1x_2 >$



Now $F(x_i).F(x_j) = (x_i.x_j)^2$

We don't need to compute the actual feature representation in the higher dimensional space, because of Mercer's theorem.

For a Mercer Kernel K, the dot product of $F(x_i)$ and $F(x_j)$ is given by $K(x_i, x_j)$.

What is a Mercer kernel? Continuous, symmetric, and positive definite

Positive definiteness: for any m-size subset of the input space, the matrix K where $K_{ij} = K(X_i, X_j)$ is positive definite

Remember positive definiteness: for all non-zero vectors $z, z^T K z > 0$

Allows us to work with very high-dimensional spaces!

Examples:

1. Polynomial: $K(X_i, X_j) = (1 + x_i.x_j)^d$ (feature space is exponential in d!)

2. Gaussian: $e^{-\frac{||x_i - x_J||^2}{2\sigma^2}}$ (infinite dimensional feature space!)

3. String kernels, protein kernels!

How do we choose which kernel and which λ to use? (The first could be harder!)

Selecting the Best Hypothesis

Based on notes from Poggio, Mukherjee and Rifkin

Define the performance of a hypothesis by a loss function V

Commonly used for regression: $V(f(x), y) = (f(x) - y)^2$

Could use absolute value: $V(f(x), y) = |f(x) - y|$

What about classification? 0-1 loss: $V(f(x), y) = I[y = f(x)]$

Hinge loss: $V(f(x), y) = (1 - y.f(x))_+$

Hypothesis space: space of functions that we search

Expected error of a hypothesis: Expected error on a sample drawn from the underlying (unknown) distribution

$$I[f] = \int V(f(x), y) d\mu(x, y)$$

In discrete terms we would replace with a sum and μ with P

Empirical error, or empirical risk, is the average loss over the training set

$$I_S[f] = \frac{1}{l} \sum V(f(x_i), y_i)$$

Empirical risk minimization: find the hypothesis in the hypothesis space that minimizes the empirical risk

$$\min_{f \in H} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$

For most hypothesis spaces, ERM is an *ill-posed* problem. A problem is ill-posed if it is not *well-posed*. A problem is *well-posed* if its solution exists, is unique, and depends continuously on the data

Regularization restores well-posedness. Ivanov regularization directly constrains the hypothesis space, and Tikhonov regularization imposes a penalty on hypothesis complexity

Ivanov regularization:

$$\min_{f \in H} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i) \text{ subject to } \omega(f) \leq \tau$$

Tikhonov regularization:

$$\min_{f \in H} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i) + \lambda \omega(f)$$

ω is the regularization or smoothness functional. The mathematical machinery for defining this is complex, and we won't get into it much more, but the interesting thing is that if we use the hinge loss and the linear kernel, the SVM comes out of solving the Tikhonov regularization problem!

Meaning of using an unregularized bias term?

Punish function complexity but not an arbitrary translation of the origin

However, in the case of SVMs, the answer will end up being different if we add a fictional "1" to each example, because now we punish the weight we put on it!

Generalization Bounds

Important concepts of error:

1. Sample (estimation) error: difference between hypothesis we find in H and the best hypothesis in H

2. Approximation error: difference between best hypothesis in H and the true function in some other space T

3. Generalization error: difference between hypothesis we find in H and the true function in T, which is the sum of the two above

Tradeoff: making H bigger makes the approximation error smaller, but the estimation error larger