

## CS 374 Assignment #7 Support Vector Machines

Due the week of October 24, 2005

This week, in which we study Support Vector Machines, will bring us to the end of our series on “the top algorithms for classification and regression”. It also brings us full circle, back to the opening topic of the semester – linear classifiers. This time, however, we consider a method that finds a linear separation in a high-dimensional feature space that is nonlinearly related to the input space.<sup>1</sup> That is, with this technique we can separate data that are, in fact, not linearly separable.

### 1 Support Vector Machines

#### 1.1 Reading

Please read the following:

- Alpaydin, Sections 10.2 and 10.9,
- Alpaydin, Section 8.2 (especially 8.2.2),
- “Support vector machines”, from IEEE Intelligent Systems, July/August 1988.

#### 1.2 Presentation of SVMs

There are many ways to gain real understanding of an algorithm – working through exercises, implementing the algorithm, etc. This week you will demonstrate your understanding by presenting the algorithm and its derivation in your tutorial session.

While you might find the readings given above to be adequate, there are many other very good sources of information on SVMs. These include:

- Notes from a tutorial on “Support Vector and Kernel Machines” presented by Nello Cristianini at ICML 2001 (which was held here at Williams).
- Lecture notes written by Andrew Ng at Stanford,
- A paper by John Platt on the Sequential Minimal Optimization (SMO) algorithm,
- A paper entitled “A Tutorial on Support Vector Machines for Pattern Recognition” by Christopher Burges, which appeared in the journal *Data Mining and Knowledge Discovery*, June 1998,
- The book *Support Vector Machines and other kernel-based learning methods* by Nello Cristianini and John Shawe-Taylor.

Pointers to most of these can be found on the “Assignments” web page for this course. The book can be found on the shelf outside my office.

Your presentation should be carefully developed and practiced. If you and your tutorial partner/group prefer to prepare a single presentation, that is perfectly fine. There is no required length of presentation, but you might want to aim for 20 minutes.

---

<sup>1</sup>Description borrowed from Schölkopf.

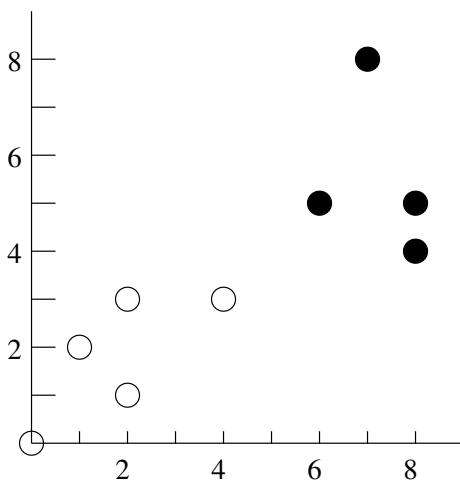
### 1.3 Exercises

- The readings for this week told you that the quadratic kernel  $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^2$  is equivalent to mapping each  $\mathbf{x}$  into a higher dimensional space where

$$\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

for the case where  $\mathbf{x} = (x_1, x_2)$ . Now consider the cubic kernel  $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^3$ .

- What is the corresponding  $\Phi$  function for the case where  $\mathbf{x} = (x_1, x_2)$ ?
  - Compare the number of arithmetic operations involved in computing the cubic kernel to the number of operations involved in computing its corresponding  $\Phi$  function. Count additions and multiplications. You may assume that the square root of a number is given to you as a constant. Don't worry about optimizing – simply count the number of arithmetic operations assuming that the functions are computed in the most straightforward manner.
- How many dot products need to be computed for M test points with N support vectors at test time?
  - Say you are given the data set shown below. This is a binary classification task in which the instances are described by two real-valued attributes.



- If the data in the graph are taken to be training data, which of the instances should be identified as support vectors?
- Using Weka, train the SMO algorithm on the data above. You can find the SMO implementation in the “functions” folder. The data can be found in

`~andrea/shared/cs374/svm-1.arff`

When training, don't normalize input values.

Based on the results obtained from running the algorithm, what are the values of  $\alpha_i$  for each of the nine instances (assuming that the algorithm identified the same support vectors that you did in part a).

4. Generate a two-dimensional example (assume two real-valued attributes and two possible classes) that cannot be separated by a linear hyperplane, but that can be separated by a polynomial kernel with degree two. You can test this with the SMO algorithm as implemented in Weka. (Just change the “exponent” setting for the algorithm to switch from linear to second degree polynomial.)

- (a) Give a graph representing your dataset.
- (b) Give the training error with the linear kernel and with the polynomial kernel of degree two.

5. Part of the power of SVMs comes from the fact that training is a convex optimization problem – there is a unique optimal value and a set of equivalent maximizing hyperplanes, each associated with a set of slack variables.

We know that maximizing the margin is equivalent to minimizing  $\|\mathbf{w}\|^2$ .  $f(\mathbf{w}) = \|\mathbf{w}\|^2$ , used in SVMs, is convex. Every convex function has a single minimum value, but, in general, many values of  $\mathbf{w}$  may achieve the same minimum value.

Following a characterization given in the assignment on decision trees, we can say that a function  $f(\mathbf{w})$  is convex, if, for all  $\lambda \in (0, 1)$ ,  $\mathbf{w}_1 \neq \mathbf{w}_2$ :

$$\lambda f(\mathbf{w}_1) + (1 - \lambda)f(\mathbf{w}_2) \geq f(\lambda\mathbf{w}_1 + (1 - \lambda)\mathbf{w}_2)$$

Prove that the optimal hyperplane parameters  $\mathbf{w}^*$  are unique for the linearly separable case. You might find it useful to proceed as follows:

- (a) Prove that the inequality above is strict for  $f(\mathbf{w}) = \|\mathbf{w}\|^2$ . That is, for  $\forall \lambda \in (0, 1)$ ,  $\mathbf{w}_1 \neq \mathbf{w}_2$ :

$$\lambda f(\mathbf{w}_1) + (1 - \lambda)f(\mathbf{w}_2) > f(\lambda\mathbf{w}_1 + (1 - \lambda)\mathbf{w}_2)$$

- (b) Using this result, prove that  $\mathbf{w}^*$  is unique.