CS 374 Assignment #6
Support Vector Machines

Due the week of March 29, 2021

—

This week, in which we study Support Vector Machines, continues our examination of linear separators – sort of. We consider a method that finds a linear separator in a high-dimensional feature space that is nonlinearly related to the input space.[1] That is, with this technique we can separate data that are, in fact, not linearly separable. Multilayer perceptrons were able to do this, but there are several interesting new ideas here, including the fact that we don't explicitly have to consider the high-dimensional feature space into which we project the input data. Sound intriguing? Read on ....

1 Support Vector Machines

1.1 Reading

Please read the following:

- Alpaydin, Section 10.2,
- Alpaydin, Chapter 14 (through 14.6 only).

1.2 Presentation of Support Vector Machines

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 Support Vector Machines (SVMs) in your tutorial session. On one hand, this will feel familiar; at every tutorial meeting you have presented the "algorithm of the week". This, time, however, the presentation itself is the primary focus of the assignment.

You and your tutorial partner may divide the presentation in any way you'd like. You might choose to present everything jointly, take turns, etc. You may choose from a variety of presentation formats – e.g., work on the virtual whiteboard, prepare a presentation that you share from your laptop, etc.

You many organize the content in any way you'd like, but be sure to include at least the following:

- the rationale for maximizing the margin;
- the notion of a support vector;
- how the optimal separating hyperplane is found (yes, you will have to get into some math here);
- what the algorithm learns;
- how SVMs are used to classify new examples;
- kernel functions.

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

- 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,
- Lecture notes written by Andrew Ng at Stanford,
- Chapter 11, Section 5 in Daumé.

[1] Description borrowed from Schölkopf.

Pointers to these can be found on the "Assignments" web page for this course. The first two are also available on Glow.

During your tutorial session I will also ask that you present solutions to the exercises below (which you need to write up and turn in). You might find it useful to incorporate the exercises and their solutions into the presentation as examples of various ideas. (This might make your job of preparing the presentation a little easier.) Basically, "teaching" on the subject of SVMs and going through the exercises is going to be your tutorial meeting task. So plan for a combination of presentation and exercises that will be about 60 minutes in length. That will also allow time for me to ask you questions.

1.3 Exercises

1. 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

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

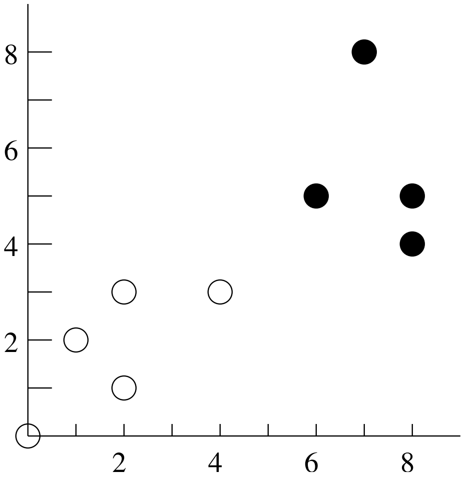
for the case where $\mathbf{x} = (x_1, x_2)$. Now consider the cubic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^3$.

   (a) What is the corresponding Φ function for the case where $\mathbf{x} = (x_1, x_2)$?

   (b) Compare the number of arithmetic operations involved in computing the cubic kernel to the number of operations involved in computing the corresponding Φ function followed by the dot product. 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.

2. How many dot products need to be computed for M test points with N support vectors at test time?

3. 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.



   (a) If the data in the graph are taken to be training data, which of the instances should be identified as support vectors?

   (b) Using Weka, train the SMO algorithm on the data above. The data can be found in

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

   as well as on Glow.

   Recall from previous weeks that typing

```
java -Xmx1g -jar /usr/share/java/weka.jar
```

will start up a GUI. Click on the “Explorer” button. Then click on “Open file...” to select the data file. Now you’re almost ready to run SMO on the data set.

Click on “Classify” and then “Choose”. Now find SMO in the “functions” folder. Before you run SMO on the data, you’ll need to modify a couple of parameters. First, you want to be sure that SMO doesn’t normalize the input features. Next to the Classifier “Choose” button, you’ll see “SMO” along with a list of parameters. Click on the text. A window should pop up that gives you options for setting parameters. For “filterType”, choose “No normalization/standardization.” Click “OK”.

Next, select the “Use training set” as the “Test option”. This way the entire data set will be used for training and testing.

Based on the results obtained from running the algorithm, what are the values of α_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 linearly separated, 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” parameter for the algorithm to 2.0.)

   (a) Give a graph representing your dataset.

   (b) Give the training error with the linear kernel and with the polynomial kernel of degree two. You should, of course, be able to achieve an error of 0 with the kernel of degree two.

2 A bit more on this week’s assignment and what you will be turning in

You might be thinking, “Hmmm, this seems a bit lighter than usual.” Not really. This week your level of responsibility for the material is much greater (but you’re all well-prepared to do this). Rather than speaking about SVMs informally and at a high level, you will effectively be giving a one-hour lecture on them. Imagine that you are teaching a machine learning course, and this is the day you’re teaching on this topic.

This week, please turn in

- A pdf with your problem set.
- (Optional) Your presentation materials. I will be giving you a grade on the presentation itself, but feel free to submit these materials, if you’d like.

To turn these in, follow the link on the course website or in the Glow module for Week 6.