SUPPORT VECTOR MACHINES

INTRODUCTION TO SVMS

— Russell and Norvig say that the SVM framework is currently the most popular approach for off the shelf supervised learning.

— It's what you start with for an analytics problem.

— It uses optimally placed linear separators (based on the support vectors).

— When linear separators don't work, it uses the kernel trick to project to higher dimensions where they do work.

— It's a framework because you need to select the appropriate kernels.

— So, an analyst uses this framework to get a cheap lunch.

OUTLINE OF THE SVM LECTURE

- Introductory Slide
- Parametric vs. Non-Parametric Learning Algorithms
- Linear Separators
- Support Vectors
- Projecting from one set of dimensions to another
- Exercise
- Kernel Trick
- Support Vector Machines
- Making Separators
- Conclusion

PARAMETRIC VS. NON-PARAMETRIC LEARNING ALGORITHMS

— There are lots of ways to subdivide learning algorithms, but one is between parametric and non-parametric algorithms.

— Parametric algorithms use the data to learn the parameters, and then they throw the data away.

— An MLP is an example of this. You use the data to learn the weights (the parameters).

— Once you've learned the weights, it's very efficient.

— A non-parametric algorithm keeps the data around.

— So, the Euclidean distance measurement I suggested in lab 16, or that is often used in case-based reasoning systems is non-parametric.

— One problem with a non-parametric algorithm is that it can really slow down when there is a lot of "training" data.

— Is a GA parametric or non-parametric?

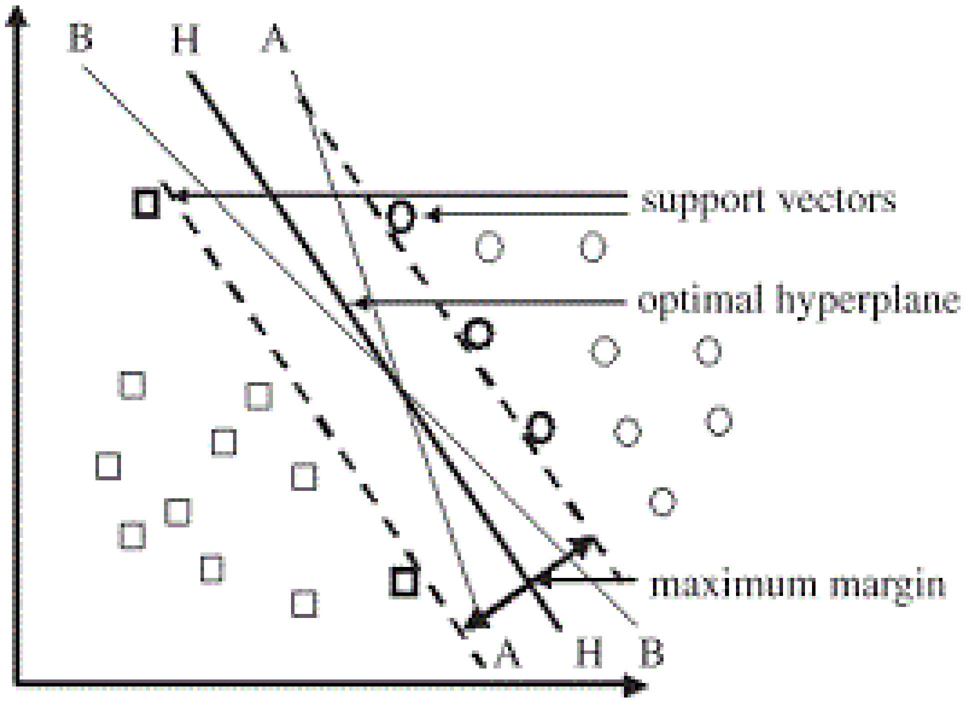
— How about conditional probability derived from input?

LINEAR SEPARATORS

- Remember the lecture on classification with lines!
- The idea is that a lot of categories can be categorised by lines.
- That's lines in two dimensions, but planes and hyper-planes in higher dimensions
- Also remember that the classification with lines lecture considered outliers,
  - multiple class tasks,
  - and classes that weren't linearly separable.
- It also referred to this lecture.

SUPPORT VECTORS

- Support vectors are lines that are used to separate categories.
- Usually, if there is one there are many lines that can separate two categories.
- What is the best one?
- You want to use the line that is furthest from both categories. Why?
- The support vectors are the lines that are on the edge of one category.
- You use the support vectors to calculate the best line, the maximum margin separator.



PROJECTING FROM ONE SET OF DIMENSIONS TO ANOTHER

- Systems can project from one dimensionality (say 3D) to another (say 2D).
- How would you project from 3D to 2D?
- If there are points in 3D space, and you shine a light on them, the shadow is projected onto a 2D shape (say the floor or a wall).
- Does it matter where the light is? The surface that it is being projected onto? Of course it does.
- Can you project from 2D to 3D?
- People do this all the time when they take a 2D image (close one eye and that's what you get) and infer depth.
- Programmatically, you can just write a function that takes the two inputs (x and y), and makes a third (z).
- You can of course do this from any N-D to any X-D.
- In high dimensions, you often use principle component analysis or independent component analysis to project data in high dimensions to 1 or a small number of dimensions.

EXERCISE

— Write a function that projects X,Y,Z coordinates onto the XY plane.

— Write a function that takes any 2D point, and makes a 3D one.

— Write a function that takes a 2D point, and makes a 1D point.

— Write a good function that takes a 2D point, and makes a 1D point (PCA).

— How would you use an SVM for the MNIST coursework?

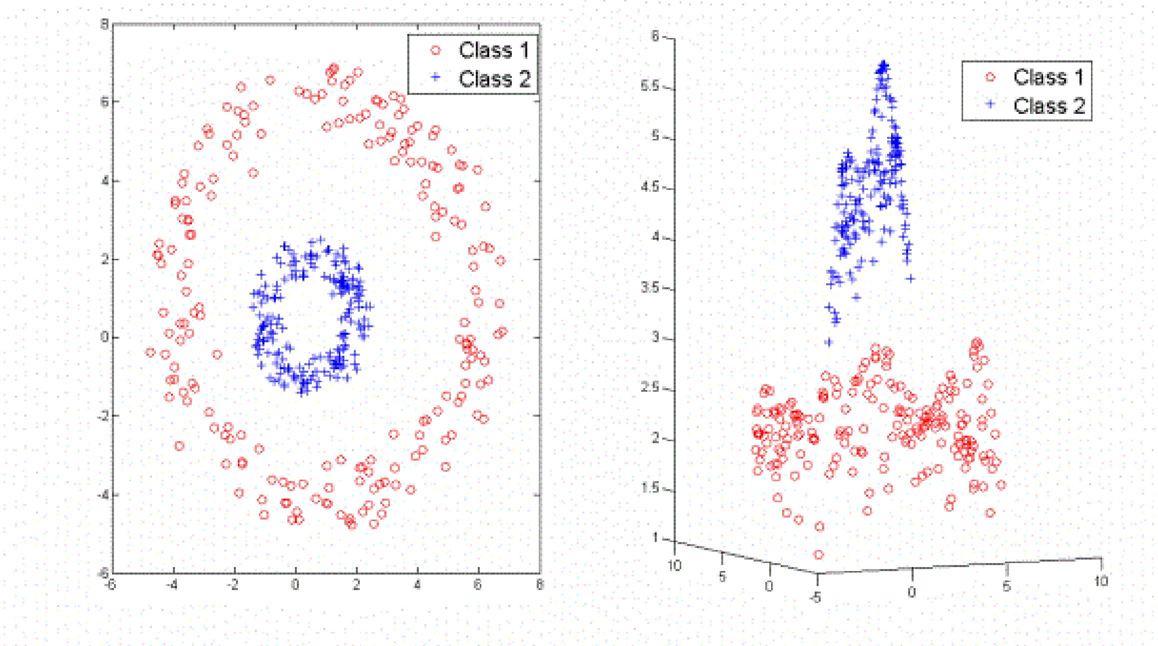
— How would you learn the hyperplanes?

— How many dimensions would you use?

— Could you project into more dimensions?

KERNEL TRICK

- The Kernel Trick is to project data to a higher dimension.
- Typically, you use the trick when there is not a good linear separator in you current dimension scheme.
- In this case, the kernel function translates the x and y coordinates to z coordinates.
- It's going to be something like 7-(distance from 0,0).
- The general problem is that you don't know which kernel to use.
- I just looked for a picture that involved a circle, because it's pretty easy to explain.
- The problem is made even more difficult by outliers.
- You could use the training data to make a kernel.



Middlesex University London

SUPPORT VECTOR MACHINES

— SVMs combine support vectors and the kernel trick.

— They find the support vectors to calculate the maximum margin separators.

— If they t can't find a line to separate the classes, they might use the kernel trick to project to higher dimensions where it can find a linear (hyper-planer) separator.

— This is a really popular technique.

— You might need lots of lines to break up the classes.

— In fact, there is no limit.

— So, is this parametric or non-parametric?

— Also note that SVMs aren't just a single algorithm; it's more a set of tools.

— That's good for analytics.

MAKING LINES

— The canonical way to make lines (or hyperplaner separators) is to use quadratic programming.

— I still haven't been able to find a good description of that, but a student managed it for last year’s categorisation coursework.

— Support Vector Machines Succinctly has a good an explanation of it all and pointers to some code.

— A simpler way is to use the old perceptron learning rule.

— This just starts out with a separator, and then moves it so one new point is correctly categorised, then repeats until it classifies correctly.

— These are hard margin separators, and don't work well with categories that aren't linearly separable.

— For these, you want to use a soft margin separator, with error tolerance specified.

CONCLUSION OF THE SVM LECTURE

Take Home Points:

Learning algorithms can be parametric or non-parametric.

Support Vector Machines make use of support vectors to generate maximum margin separators.

If you can't find good linear separators, you can use the kernel trick to project to higher dimensions where

you can find a linear separator.

SVMs are a framework using linear separators and kernels.

— This lecture has used mostly two dimensions for support vectors for illustration. This all works in higher dimensions.

Reading for this week: Russell and Norvig's Learning from Examples Chapter section 9 (pp. 755-758).

Reading for next week: Russell and Norvig's Making Simple Decisions Chapter (pp. 621-647) (particularly section 3).