Classification using (linear) Support Vector Machines

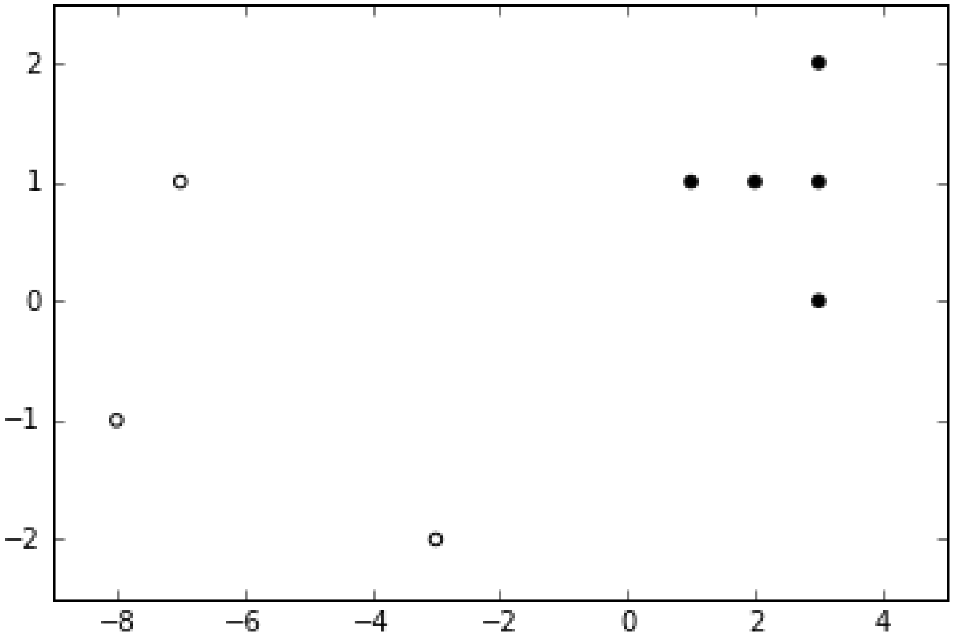
"Support vector machines" are one way of classifying data observations.

- Support: the support of a function is the set where the function isn?t zero
- Vector: an arrow that points to a point; a direction and a magnitude
- Machine: it sounds cool!

We?ll start with *linear* support vector machines, and in fact the simplest version: the "maximal margin classifier". The idea is that you pick a linear function like $f(x) = 3x + 2y - 1$, and then you split your data into two classes using the line where that linear function equals zero. One of the classes should be on the side where $f(x) > 0$ and the other class should be on the side where $f(x) < 0$. (Yes, just two classes – if you want to deal with more classes, you iterate this again and again.)

Big idea: Pick linear functions to separate your groups

Fake data first:
Here is some fake data, designed to be nice. Can you draw a line to separate the two groups? (Can you draw more than one line?) Based on yesterday?s activity, you know you can! But let?s get more specific:



Compare the line you drew with the lines your neighbor drew. Some yes-no questions:

□ If I removed the white point at (-8,-1), would you change your separating line?

□ If I removed the black point at (3,2), would you change your separating line?

□ If I removed the white point at (-7,1), would you change your separating line?

□ If I removed the black point at (3,0), would you change your separating line?

Talk with your neighbor – do you agree?

Given your discussion, which points do you think matter the most?

The points that matter the most are the ones that give us the *support vectors*.

The Support Vector Machine Algorithm

How do we mathematically decide where the line between two groups should go? This is an *optimization* problem. "Optimal" means "best," measured in a specific mathematical way.
The people who invented support vector machines (SVM) decided that they wanted the widest possible ?street? between the two groups of data. This is called the margin. I want to use a ?street? analogy because you want a lane on each side of your separating line!

- This is a supervised learning problem, so you need data separated into two classes, labeled by 1 and -1.
- You want to find the margin M (the ?width of the lanes in the street?) that is maximal, as big as possible. Remember the ?street? can contain no data.
- Here are the constraints for your data: