DSCI 320: Optimization Methods in Data Science

Homework assignment 5 – due Friday 11/15/2019

Note: This homework is all about Support Vector Machines (SVMs). There are many implementations of SVMs out there on the internet. But it's worthwhile writing a simple implementation once, so you understand the general principle. As a consequence, I'd like you to implement the necessary algorithms for this homework yourself, rather than relying on an implementation you find somewhere.

Problem 1 (Support vector machines with hard counting). Let's play with hard counting first. To this end, create two data clouds by randomly drawing 100 points each from 2-dimensional normal distributions with standardard deviations 1 and mean values equal to $(0,0)^T$ and $(5,0)^T$ respectively. Plot these data points (colored by type) and make sure that the two point clouds are sufficiently separated so that you can put a straight line that separates the two – if that is not the case, feel free to move the centers of the two point clouds, play with the standard deviations, or with the random number generators until the two clouds are indeed separate.

Then formulate the "hard-count" SVM classification problem as a constrained optimization problem. It contains inequality constraints for which you have seen how to use the logarithmic barrier method to convert the problem into an unconstrained one in which the constraints are incorporated into the objective function. State the form of this problem.

Solve this reformulation with a minimization algorithm of your choice to obtain the classifier. Generate a plot that contains the data points, the classifier line, and the exclusion zone. **(50 points)**

Problem 2 (Support vector machines with soft counting). Repeat the previous problem with two point clouds that do overlap, i.e., for which there is no hard-count classifier. To this end, generate again two point clouds drawn from a normal distribution with standard deviation 1 and mean values at $(0,0)^T$ and $(2,0)^T$.

Then formulate the soft-counting problem and solve it using a method of your choice. There are two general approaches you can follow:

- The original formulation of the soft-counting formulation had a non-smooth objective function but no constraints; derivative-based methods such as Newton's method or Steepest Descent might work if you employ careful line search methods, but you can also just use a stochastic algorithm if you want.
- Using slack variables s_i, the non-smooth problem can be converted into a smooth one with inequality constraints, and these constraints can then again be incorporated into the objective function using the logarithmic barrier method. You can then apply a smooth, unconstrained optimization method to this problem again.

Discuss your choice of formulation and what method you use to solve it.

Again generate a plot that contains the data points (colored by type), the classifier line, and the exclusion zone (which now no longer actually exclused everything if the point clouds overlap). Also discuss your choice of the balance factor λ in the objective function that decides whether you care more about the size of the gap or the correct classification of data points. **(50 points)**

Bonus problem. The way to use advanced algorithms is to implement a simple method yourself so you understand the general principle, and then use one of the existing high-quality implementations that other people have already written, optimized, and debugged. If you want, use an existing implementation of SVMs on the two problems above. Discuss what software you used and show relevant results.

(5 bonus points for each of the two problems)

If you have comments on the way I teach – in particular suggestions how I can do things better, if I should do more or less examples, powerpoint slides vs whiteboard, etc – or on other things you would like to critique, feel free to hand those in with your homework as well. I want to make this as good a class as possible, and all comments are certainly much appreciated!