Practice Exercises for Support Vector Machines

LOrtiz (Orig. by KKoile)

Part 1: Using the SVM demonstration program

Note: You can access the SVM demonstration program from the 6.034 web page.

1. How does changing the kernel function affect each of the test cases? (e.g. see last page)

2. Does changing the kernel change the location of a decision boundary?
   Most of the time, yes. (See boundaries shown on last page.) Sometimes, changing the sigma value on a radial basis kernel will not change the decision boundary location, e.g. if the points are symmetric. (See 2001 final exam question.)

3. How does the distance between positive and negative data points affect the width of the street and the support vector weights?
   The farther apart the closest pairs of opposite points, the wider the street, and the lower the weights on the support vectors, which form the gutters of the street. Intuitively, the farther apart the support vectors, the less "influence" they have to exert on points in the street to "pull" those points into the support vector's class, so the lower the weights (which represent the influence).

4. What would a diagram look like for a support vector machine that has overfit the data?
   Most, or all, of the data points would be support vectors. (e.g. see last page)

5. How can you tell by comparing diagrams which kernel does the best job at building a classifier for a particular set of data?
   The best classifier for a particular set of training data will have all data points classified correctly. (In our SVM demonstration program, that means that none of the points are shown as red circles.) Also, not all (or not all but a few) of the data points are support vectors.

Part 2: Thinking about the math

1. What equations are used for classification in a support vector machine?
   $w \cdot u + b > 0$ for positive class (i.e. on one side of the boundary line)
   $w \cdot u + b \leq 0$ for negative class (i.e on the other side)

   Also recall that the support vectors are specified as having values of 1 and –1 for this equation:
   $w \cdot x_+ + b = 1$ for positive support vectors
   $w \cdot x_- + b = -1$ for negative support vectors
   The above two equations are often combined into:
   $y_i(w \cdot x_i + b) - 1 = 0$, with $y_i = 1$ for positive support vectors, –1 for negative support vectors

2. Use the fact that a line can be represented by a normal vector and a distance from the origin to explain how the above equation classifies points on either side of a line.

   The equation of a line is $n \cdot [x\ y] + b = 0$, where n is the unit normal vector, b is the distance from the origin, and [x y] is the vector from the origin to the point. All points further from the origin than the line, satisfy the equation $n \cdot [x\ y] + b > 0$; all those closer to the origin, satisfy the equation $n \cdot [x\ y] + b < 0$. If we consider w a normal vector, and b the negative of the distance (scaled by the magnitude of w) from the origin, we can see that the classification equation looks like a test for determining on which side a line the point u lies. (Note that how a support vector machine classifies points that fall on a boundary line is implementation dependent. In our discussions, we have said that points falling on the line will be considered negative examples, so the classification equation is $w \cdot u + b \leq 0$.)

3. When describing the placement of decision boundaries using a support vector machine, what function are we maximizing in our LaGrangian formulation of the problem? What do our constraints represent?

We are maximizing the width of the street, and the constraints say that our gutter points (i.e. support vectors) will have classification values of 1 and –1.

4. Dot products are used inside the classifier of a support vector machine. Are they used in the training step as well? If so, where?

Yes, we train the classifier by maximizing the dual of a LaGrangian formulation. The dual contains a term that depends on the dot product of sample points: $L_D = \sum\alpha_i - 1/2\sum\sum\alpha_i\alpha_j y_i y_j x_i . x_j$

5. The best decision boundary yields the widest street. To maximize the width of the street, we end up maximizing an equation written in terms of what variables?

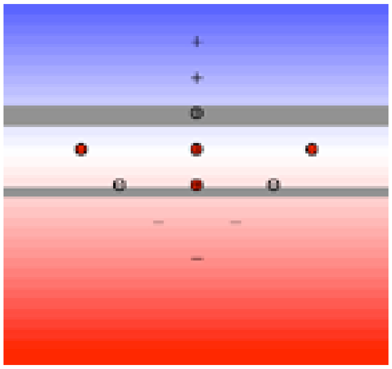
The first term is a sum of support vector weights; the second term is a weighted sum of dot products of sample points (see equation shown in answer for 4.)

6. Once we have found support vectors and their weights, how do we find a classification vector (which we have called *w*)?

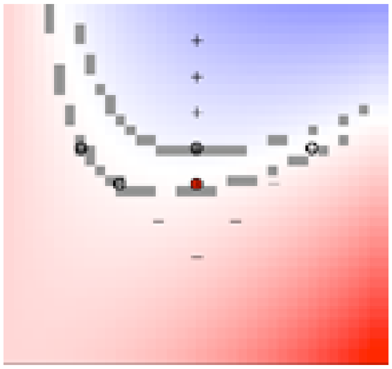
$w = \sum\alpha_i y_i x_i$

Since α_i values are 0 for non-support vectors, only support vectors contribute to this computation. If we do not know the α_i values, for simple cases they may be computed using known sample points and the classification equation, constraint equations, and the fact that $\sum\alpha_i y_i = 0$. For more complex cases (which is most of the time), the α_i values are found by maximizing L_D (see equation shown in answer for 4).

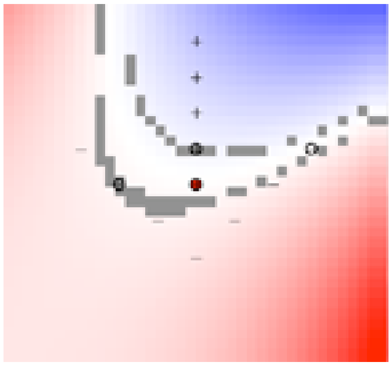
Admiral’s delight data set with different kernels



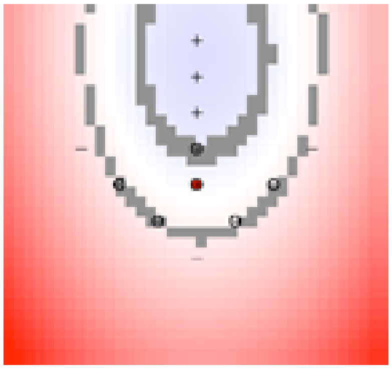
linear



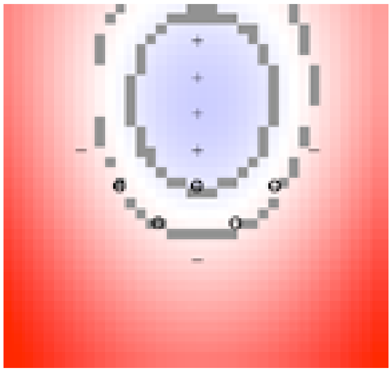
second order polynomial



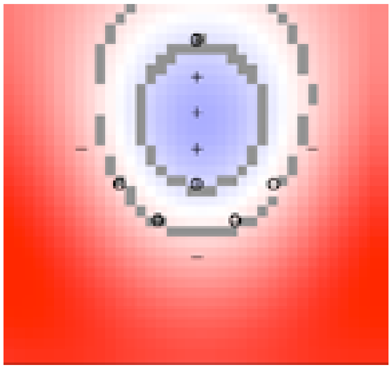
third order polynomial



radial basis, 2.0



radial basis, 0.5



radial basis, 0.08

SVM that has overfit training data