Solutions provided by David Johnson. See Code for the solutions to 1-3.

4. *SV* lists the coordinates of the identified support vectors (for those who've forgotten, the support vectors are the points nearest the decision boundary—and by extension are the points that *define* the decision boundary). For the linear problem using the first test data, there should be 3 identified support vectors.

   By a strict technical definition, the support vectors should lie precisely on the margin, meaning they should all be precisely equidistant from the decision boundary. Why do we have 3, then, for our linear classifier? Shouldn't a line or plane be defined as the set of points that are all equidistant from only *two* points? Did it actually just so happen that two points on one side of the boundary were *exactly* equidistant from it in this case, and if that's the case then why do they have different alpha values (see below)? The answer is basically that this is an artifact of the quadratic programming setup and optimization process that identifies the support vectors. For reasons of computability (and likely even lower-level concerns, like floating point rounding errors) the SVM solver actually finds points that are *approximately*, rather than precisely, on the margin.

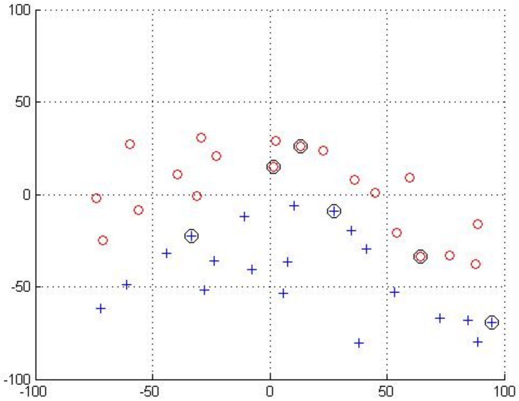
   alpha contains the weight vector—which only contains 3 values because, again, the support vector machine is defined entirely by the support vectors, so we only need weights for those points.

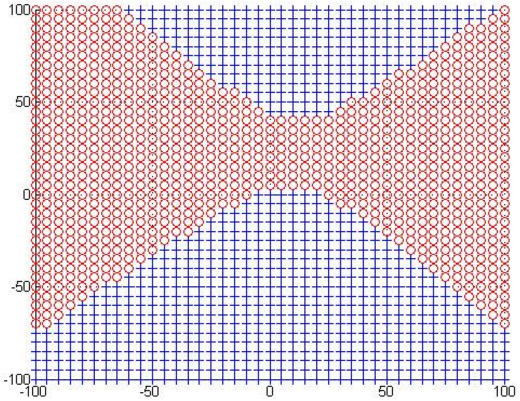
   B simply the bias term of the learned discriminant. In this case it is fairly small, since the optimal decision plane almost passes through the origin.

5. The circled points are simply the aforementioned support vectors.

6. This value defines the loss for misclassifying a point. Since this problem is linearly separable (and we thus should not have any misclassified points in an optimal solution), we do not need to worry about this, so we set the value to infinity.

7. The nonlinear plot should look something like this:

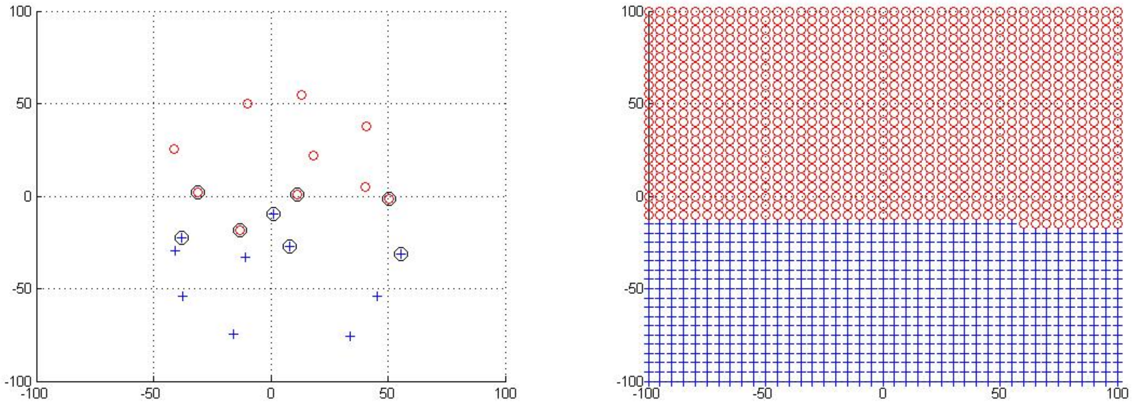
   Here the data is not linearly separable, but by applying a quadratic kernel we can effectively learn a quadratic discriminant to divide it. Aside from the new boundary shape, the particulars of these plots are the same as the linear ones—circled points represent the support vectors (those closest to the boundary), and the red and blue symbols represent the different classification regions.

   One new aspect of this method, though, is the blue region above the red region on the second graph. This region shape is a side-effect of using a quadratic discriminant, and could cause problems if we tried to apply this learned boundary to test data later. Based on this data, points located above the current set of red points should probably also be classified as red, but this classification result will not necessarily accomplish this.

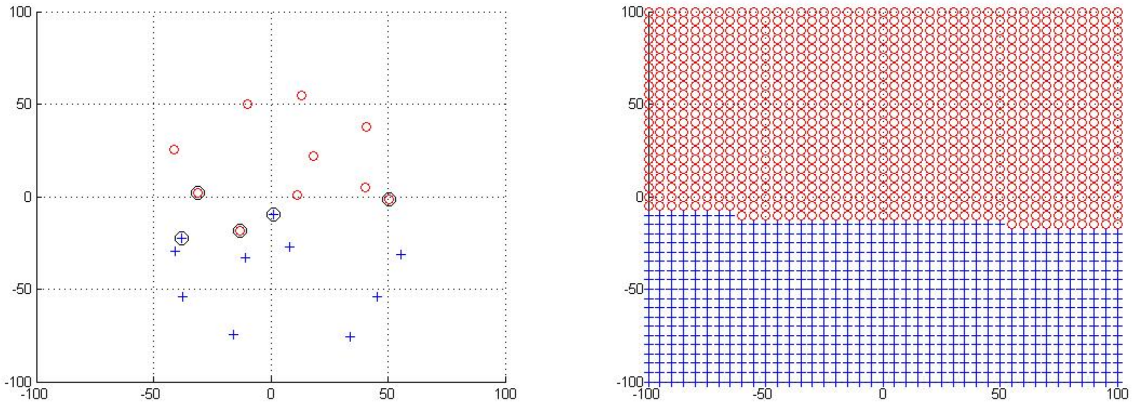
8. Here we are attempting to use a linear discriminant to solve a problem that (like most real machine learning problems) is not linearly separable. As such, we will inevitably have some misclassified points in our final solution, so we need to make use of the C parameter to determine what loss we will pay for these points. It turns out that, at least in this case, the C variable does indeed affect the results... but only to a fairly minor

extent.

C = infinity:



C = 0.1



As the value of C is decreased, the angle of the decision boundary changes to reduce the distance between the boundary and some of the more distant support vectors.

When C is infinity, the misclassified points have infinitely more weight in the optimization problem than the other points, and as such the SVM's only goal is to minimize the distance between the boundary and the misclassified points. The quadratic program effectively makes no effort to maximize the margin between the boundary and other, non-misclassified, points.

As C is decreased, the algorithm becomes more flexible with respect to the two misclassified points, and adjusts to reduce the distance to some correctly classified points, at the cost of slightly increasing the distance to one or both of the misclassified points.