LaTeX command declarations here.

EECS 445: Machine Learning

Hands On 09: Support Vector Machines

- Instructor: **Ben Bray, Chansoo Lee, Jia Deng, Jake Abernethy**
- Date: October 10, 2016

NEW: Finished Course website: [http://eecs445-f16.github.io (http://eecs445-f16.github.io)](http://eecs445-f16.github.io)

Brute Force Search for Max-Margin SVM Solution

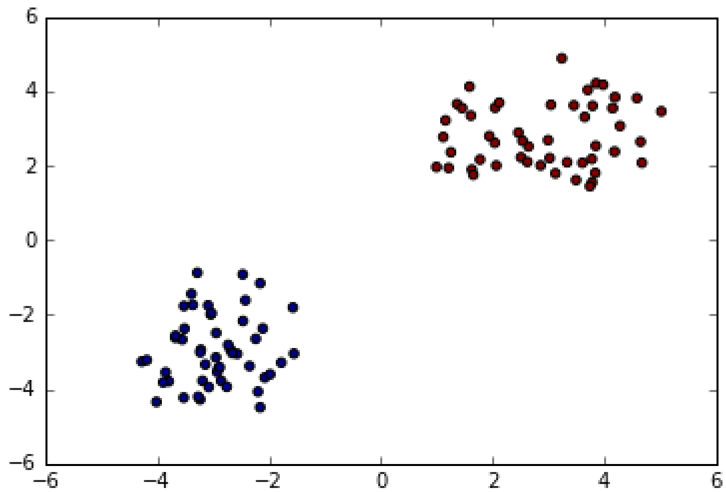
In the hard-margin support vector machine formulation, we want to try to find the hyperplane the maximizes the margin and correctly classifies the data.

Let's generate some data!

In [2]:
```
center1 = np.array([3.0,3.0])
center2 = np.array([-3.0,-3.0])
X = np.zeros((100,2)); Y = np.zeros((100,))
X[:50,:] = np.random.multivariate_normal(center1, np.eye(2),(50,))
Y[:50] = +1
X[50:,:] = np.random.multivariate_normal(center2, np.eye(2),(50,))
Y[50:] = -1

plt.scatter(X[:,0], X[:,1], c = Y)
```

Out[2]: <matplotlib.collections.PathCollection at 0x111fd4cd0>



Problem: Hard-margin SVM

1. First pick one vector and offset term $(\mathbf{w}, b)$ that correctly classifies the data
2. Determine the size of the margin for this $\mathbf{w}$
3. **Challenging**: Do a brute force search (over a grid) to find the max-margin $\mathbf{w}$!

> Note, this is not a good idea in general, since this algorithm has time complexity exponential in the dimension, but it's not so bad in 2d!

4. Find the support vectors and plot them
5. Modify the dataset above such that there is *no feasible solution* $\mathbf{w}$ (but just barely)
6. How do you know when there is no feasible $\mathbf{w}$?

In [ ]:
```
wvec = np.array([-4.0,7.0])
bval = -2.4
# Does this wvec and b correctly classify data within margin?
```

Problem: Soft-margin SVM objective

- Recall original OSMH problem is

$$\begin{aligned} \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^n \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \quad \forall i \end{aligned}$$

- Another way to write this is as follows:

$$\underset{\mathbf{w},b,\xi}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^n \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))$$

1. You modified the dataset above to ensure there is no feasible $\mathbf{w}$. Now find the $\mathbf{w}$ that minimizes the OMSH objective using a brute force search
2. Find two values of C where the support vectors of the solution are different. Plot these in both cases.

In [ ]:
```
# put some code in here!
```

OSMH *Dual* Formulation

- The previous objective function is referred to as the *Primal*
  - With N datapoints in d dimensions, the Primal optimizes over $d+1$ variables $(\mathbf{w}, b)$.
- But the *Dual* of this optimization problem has N variables, one α_i for each example i!

$$\begin{aligned} \underset{\alpha,\beta}{\text{maximize}} \quad & -\frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j t_i t_j \mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^n \alpha_i \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C/n \quad \forall i \\ & \sum_{i=1}^n \alpha_i t_i = 0 \end{aligned}$$

- Often the Dual problem is easier to solve.
- Once you solve the dual problem for $\alpha_1^*, \ldots, \alpha_N^*$, you get a primal solution as well!

1. Can you figure out a way (without using an optimization solver!) to determine the optimal dual parameters?
   - open ended, you can try different ideas
   - feel free to use the fact that you already know the support vectors
2. How do you know that you did indeed find the optimal α's?
3. How can you compute the primal variables $\mathbf{w}, b$ from these α's?

In [ ]:
```
# let's find some alphas!
```