Logistic Regression

CSE 546
Recitation 3
Oct. 15, 2013

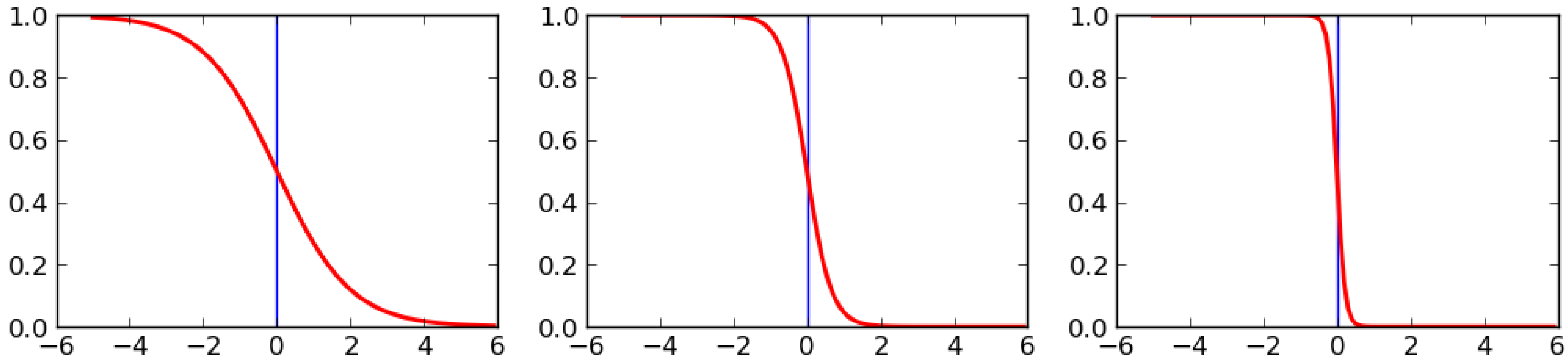
Outline

- Sigmoid
- Overfitting
- Gradient Descent

The Sigmoid

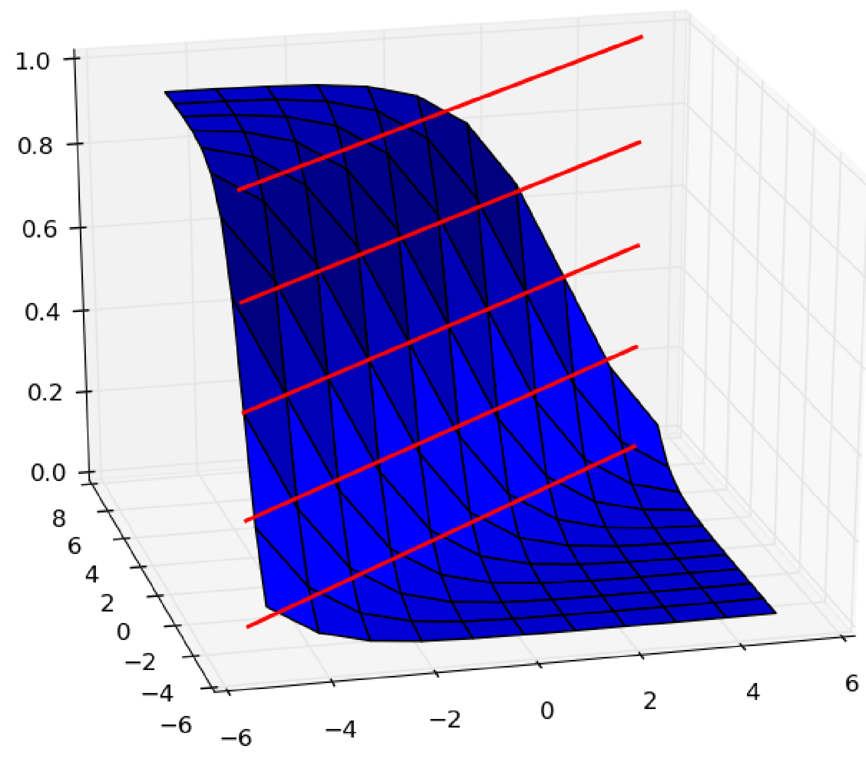
$$\Pr(y = 0) = \frac{1}{1 + \exp(wx)}$$

- Suppose there is no intercept, and w = 1,3,9
- Which graphs correspond to which w?



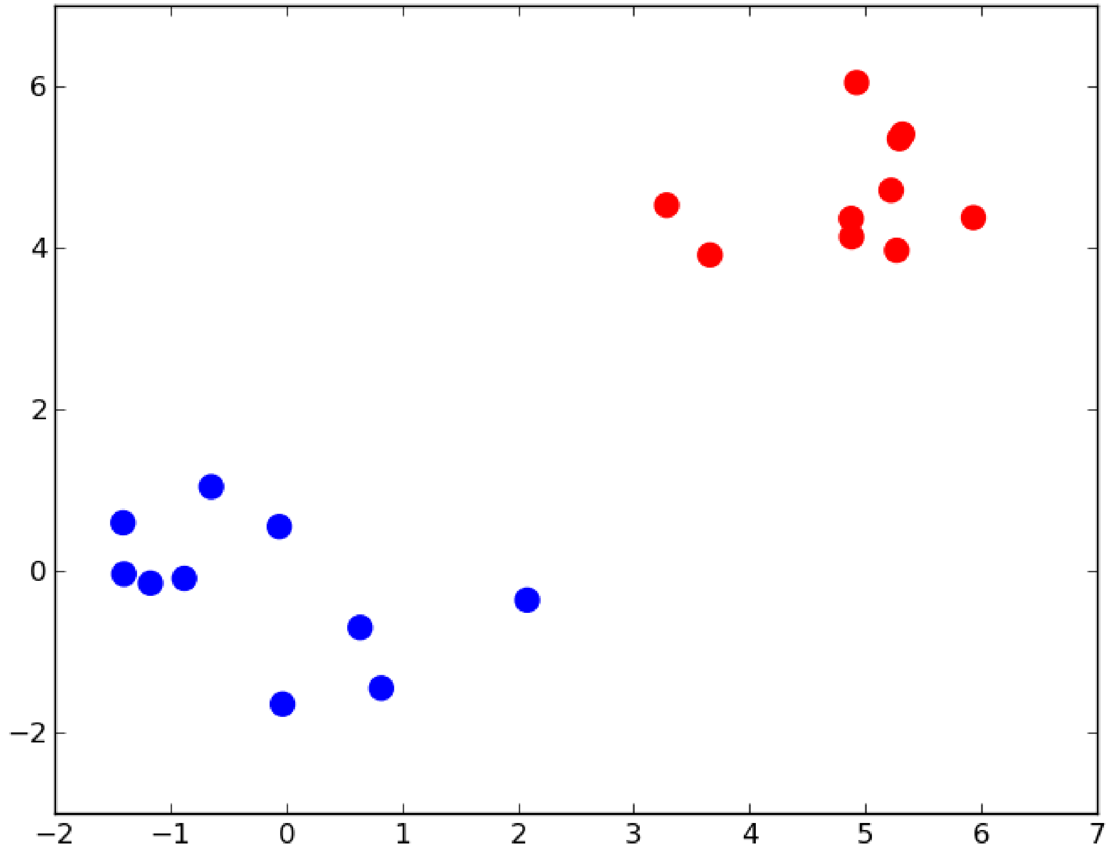
Linear Decision Boundary

- (x,y) points are classified by which side of the decision boundary they are on
- Decision boundary (red lines): $w_0 + w \cdot x = 0$

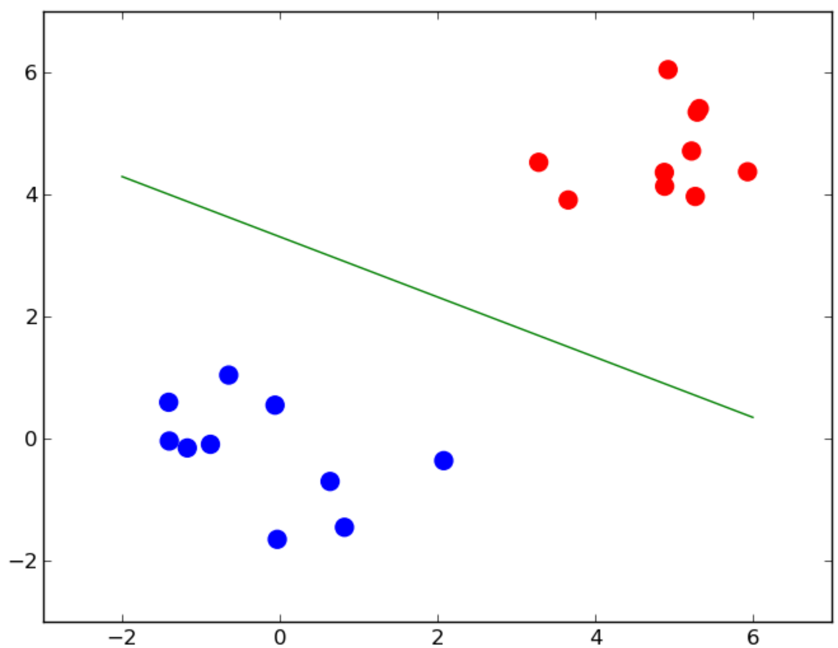


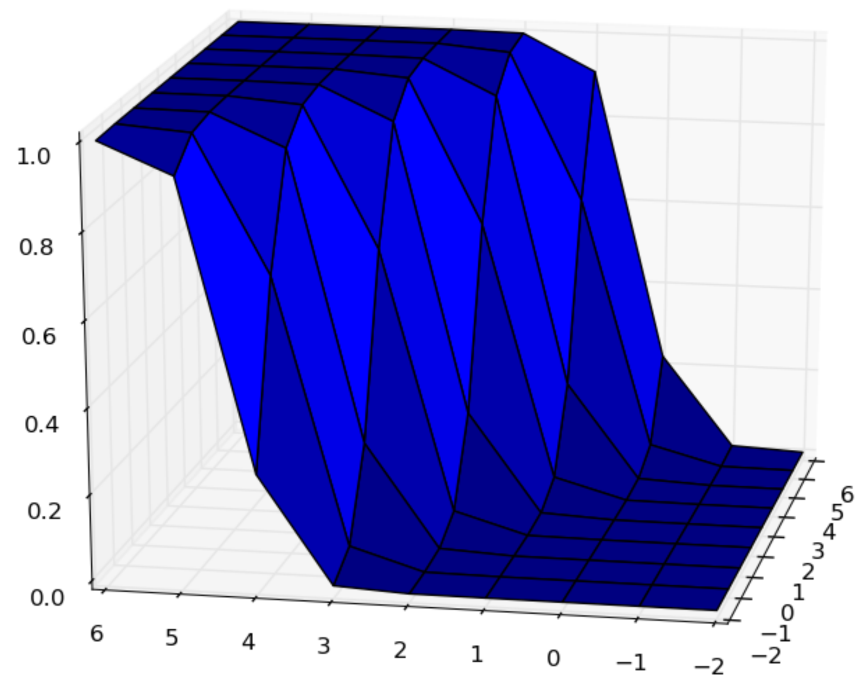
Overfitting in Logistic Regression

- Why will we overfit these data?



Overfitted Model with Linearly Separable Data





- Model should theoretically be a step function, but the package I am using prevents this

Why does linear separability cause overfitting?

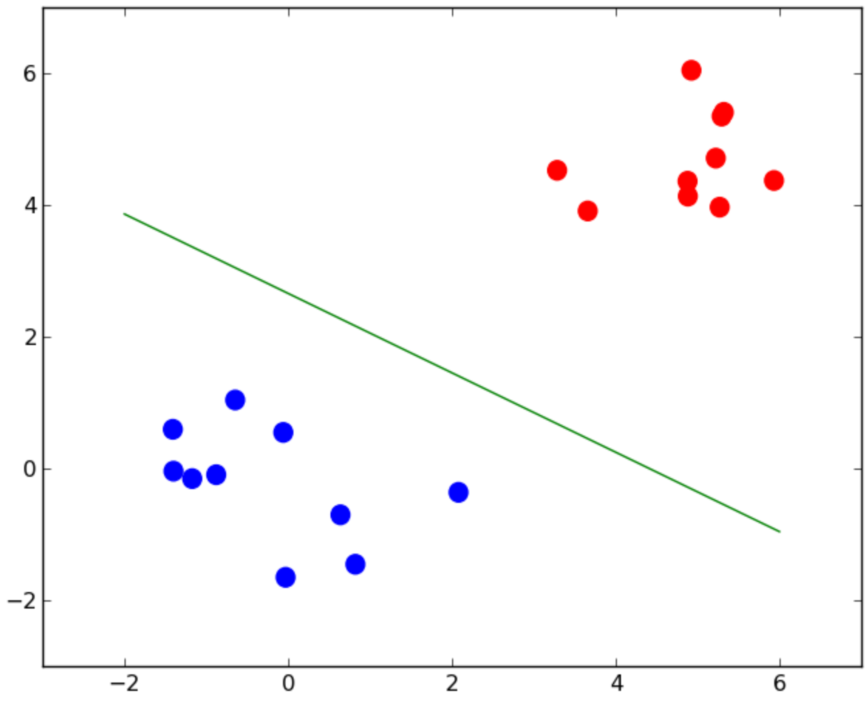
- Logistic regression's objective function, conditional likelihood, is maximized if every point is classified correctly:

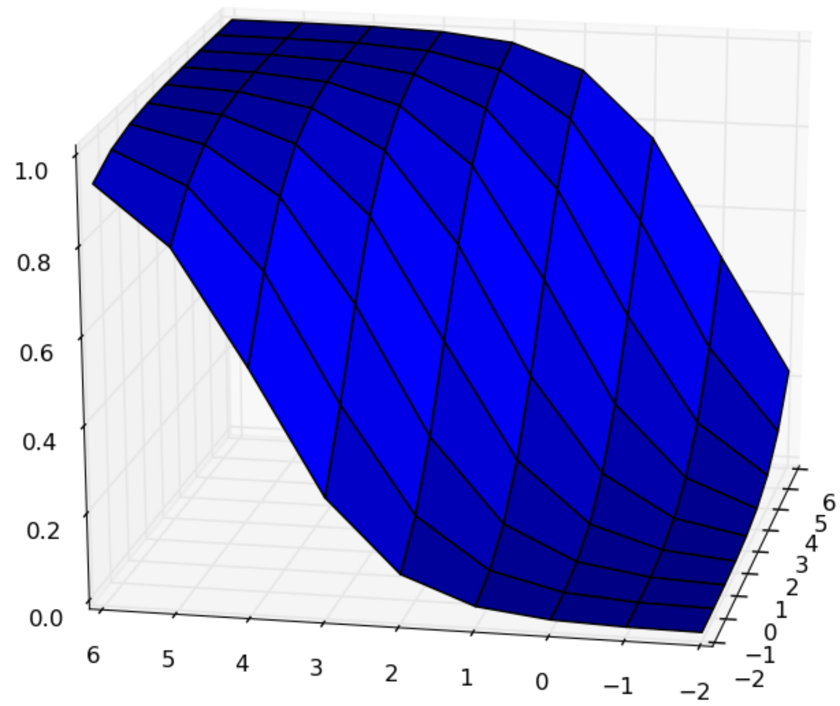
$$\Pr(y_i | x_i, w, w_0) = 1, i = 1, \ldots, n$$

$$\Pr(y_i = 0 | x_i, w, w_0) = \frac{1}{1 + \exp(w_0 + w \cdot x_i)}$$

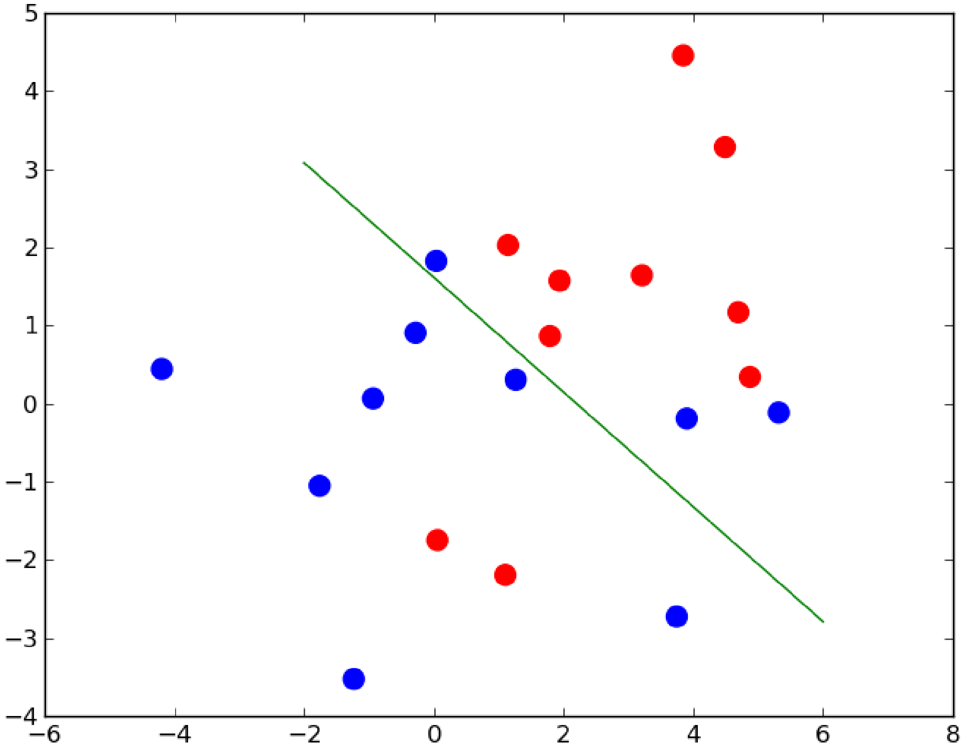
- Possible if and only if linearly separable data
- $\exp(w_0 + w \cdot x)$ must be 0 or infinity, so w0 and/or w are infinite
- Creates 0-1 step function with step at $w_0 + w \cdot x = 0$

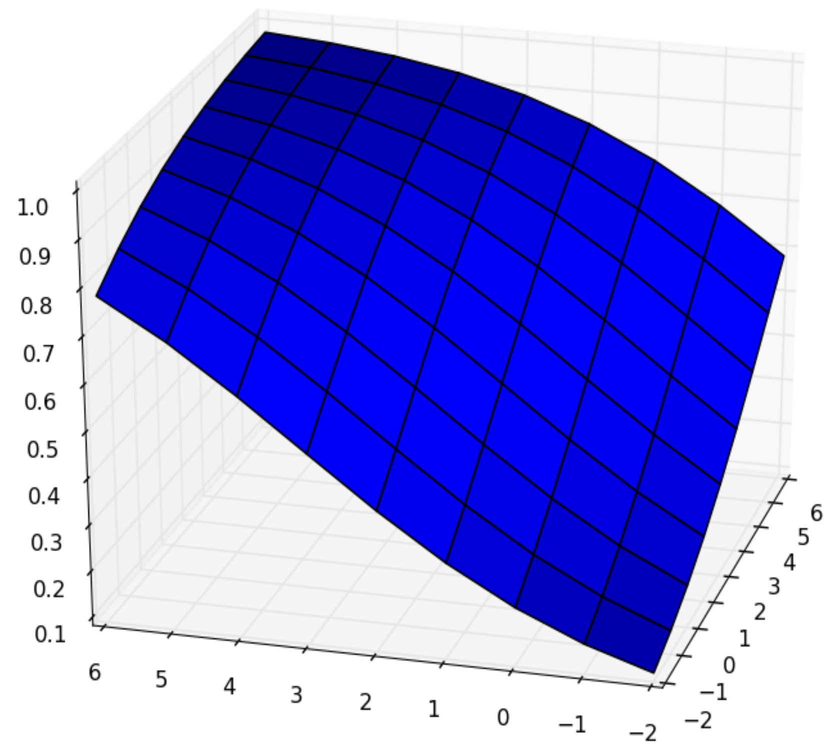
Regularized Model with Linearly Separable Data





Unregularized Model on Non-Linearly Separable Data





Gradient Descent

- How can we optimize a convex function f(w) if there is no closed form solution to

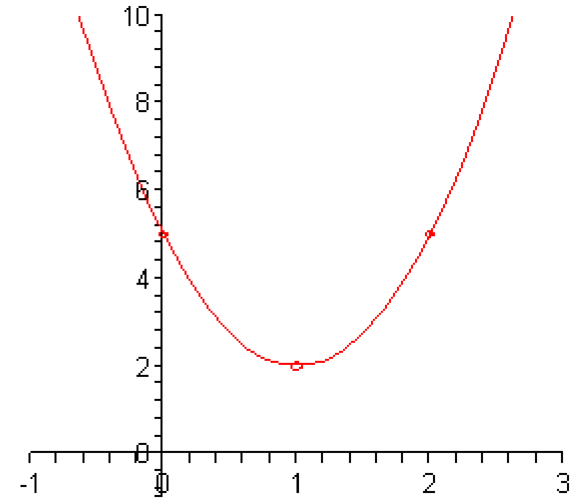
$$\nabla_w f(w) = 0$$

- Logistic regression objective has this problem (but concave)
- Must use numerical approximation algorithm such as gradient descent

Update Rule

- Update estimate by subtracting gradient evaluated at that point (with step size parameter $\eta^{(t)}$)

$$w^{(t+1)} \leftarrow w^{(t)} - \eta^{(t)} \nabla_w f(w^{(t)})$$



- Let w^* be argmin of optimum
- In dimension k, if estimate is $w_k < w_k^*$, the derivative is negative, so subtracting it increases w_k
- If $w_k > w_k^*$, subtracting derivative decreases w_k

Gradient Descent for Linear Regression

- Linear regression has convex objective, mean-squared error, so we can use gradient descent
- MSE: $f(w) = \frac{1}{n}(Y - Xw)^T(Y - Xw)$
- Update rule for GD:

$$w^{(t+1)} \leftarrow w^{(t)} - 2\eta^{(t)}\frac{1}{n}X^T(Xw^{(t)} - Y)$$

- Update rule for SGD replaces mean over all points with only one point (x_t is a 1xd vector for the t-th point):

$$w^{(t+1)} \leftarrow w^{(t)} - 2\eta^{(t)}x_t^T(x_t w^{(t)} - y_t)$$

Linear Regression Update Rule Derivation

- For regular gradient descent,

$$w^{(t+1)} \leftarrow w^{(t)} - \eta^{(t)} \nabla_w f(w^{(t)})$$
$$f(w) = \frac{1}{n}(Y - Xw)^T (Y - Xw)$$
$$\nabla_w f(w) = 2\frac{1}{n}(-X^T)(Y - Xw) = 2\frac{1}{n}X^T(Xw - Y)$$

- In dimension j, $$\frac{\partial}{\partial w_j} f(w) = 2\frac{1}{n}\sum_{i=1}^{n} x_{ij}(x_i w - y_i)$$

- For stochastic gradient descent, replace this sum/mean with a single point

$$2x_{tj}(x_t w - y_t)$$

- In matrix form with all dimensions,

$$2x_t^T(x_t w - y_t)$$