Logistic Regression

Kenny Marino, Colin White and Nupur Chatterji

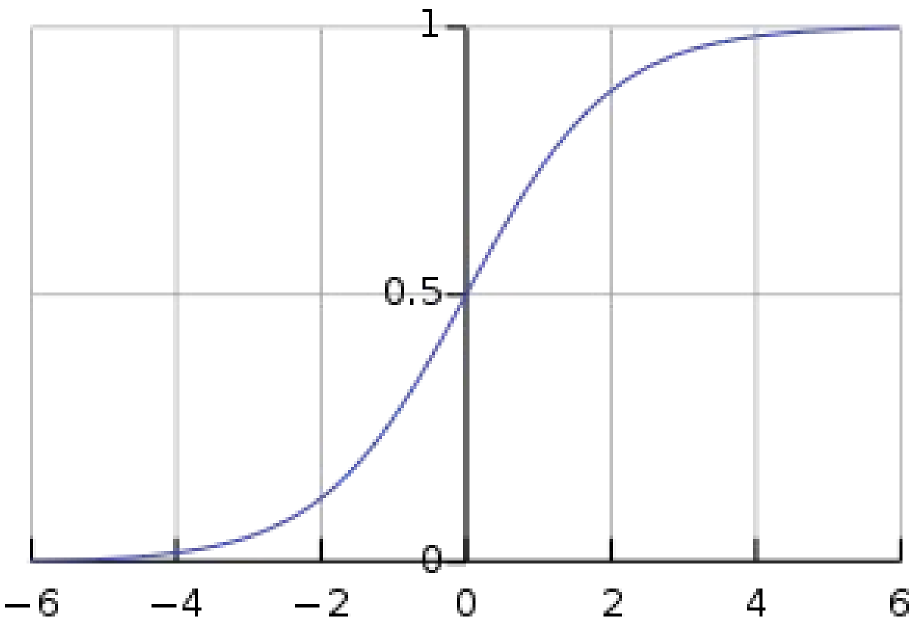
Motivation

- Generative Classifiers (like Naive Bayes)
  - Assume some functional form for P(X, Y) or for P(X|Y) and P(Y)
  - Estimate P(X|Y) and P(Y) from the training data
  - Calculate P(Y|X) using Bayes’ Rule
- **WHY NOT LEARN P(Y|X) DIRECTLY?**
- Discriminative Classifiers (like Logistic Regression)
  - Assume some functional form for P(Y|X) or for decision boundary
  - Estimate parameters of P(Y|X) directly from training data

Functional Form

$$P(Y = 1|X) = \frac{1}{1 + \exp(-(w_0 + \sum_i w_i X_i))} = \frac{\exp(w_0 + \sum_i w_i X_i)}{\exp(w_0 + \sum_i w_i X_i) + 1}$$

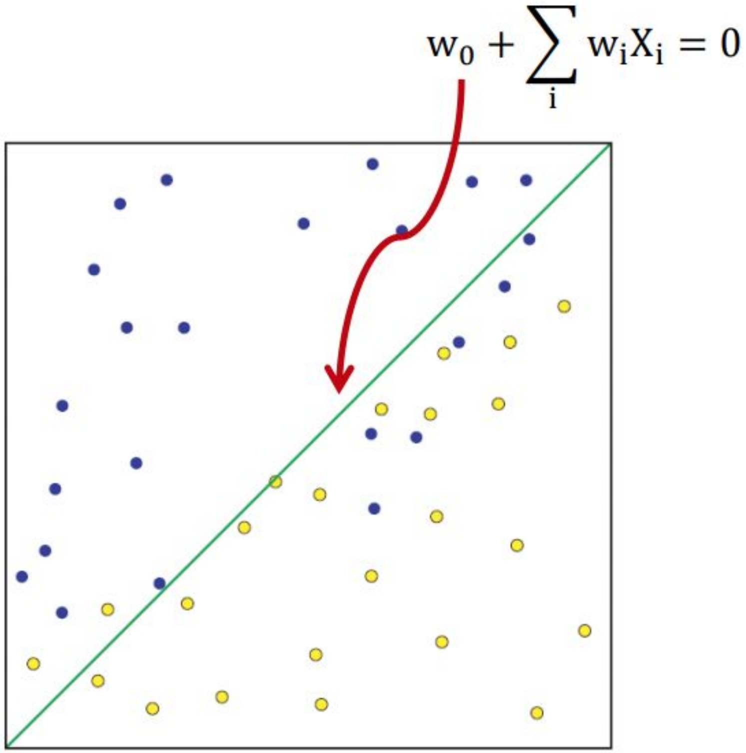
Logit/Sigmoid Function



- Large weights lead to overfitting in the model
- How can we prevent overfitting?
  - Penalize high weights

Linear Decision Boundary

- $P(Y = 1|X) > P(Y = 0|X)$
- The boundary has the equation
  - $\mathbf{w_0 + \Sigma w_i X_i = 0}$
- Then we classify points based on the question
  - $\mathbf{w_0 + \Sigma w_i X_i > 0}$



Conditional Log Likelihood

- Goal is to choose parameters **w** to maximize conditional likelihood of training data
- Do so by maximizing the conditional log likelihood function
    - No closed form, which is problematic
    - But l(**w**) is concave so we can easily find a unique maximum

$$\max_{\mathbf{w}} l(\mathbf{w}) \equiv \ln \prod_j P(y^j | \mathbf{x}^j, \mathbf{w})$$

$$= \sum_j \left[y^j \left(w_0 + \sum_{i=1}^{d} w_i x_i^j \right) - \ln \left(1 + \exp \left(w_0 + \sum_{i=1}^{d} w_i x_i^j \right) \right) \right]$$

Gradient Ascent (slide from Tom Mitchell)

$$
\begin{aligned}
l(W) &\equiv \ln \prod_l P(Y^l|X^l, W) \\
&= \sum_l Y^l (w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l))
\end{aligned}
$$

$$
\frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))
$$

Gradient ascent algorithm: iterate until change < ε

For all i, repeat

$$
w_i \leftarrow w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))
$$

M(C)LE and M(C)AP

- Know how to handle M(C)LE
- Defining priors on **w** helps to avoid overfitting (due to large weights)
- Make sure to refer back to lecture slides for exact derivations
- Next slides from Tom Mitchell

- Maximum conditional likelihood estimate

$$W \leftarrow \arg\max_W \ \ln \prod_l P(Y^l | X^l, W)$$

$$\boxed{w_i \leftarrow w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1 | X^l, W))}$$

- Maximum a posteriori estimate with prior W~N(0,σI)

$$W \leftarrow \arg\max_W \ \ln [P(W) \ \prod_l P(Y^l | X^l, W)]$$

$$\boxed{w_i \leftarrow w_i - \eta \lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1 | X^l, W))}$$

- Maximum a posteriori estimate with prior W~N(0,σI)

$$W \leftarrow \arg\max_W \ \ln[P(W) \ \prod_l P(Y^l|X^l, W)]$$

$$\boxed{w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))}$$

called a "<u>regularization</u>" term

- helps reduce overfitting, especially when training data is sparse
- keep weights nearer to zero (if P(W) is zero mean Gaussian prior), or whatever the prior suggests
- used very frequently in Logistic Regression

Main Takeaways

1. Logistic Regression is a linear classifier
2. The decision rule that is generated is a hyperplane
3. Optimize Logistic Regression by conditional likelihood
    a. No closed form solution
    b. But since it is a concave function, we can use Gradient Ascent/Descent
    c. M(C)AP corresponds to regularization

Any Questions?