Logistic Regression

Maria-Florina Balcan

02/12/2018

Generative vs. Discriminative Classifiers

Generative classifiers (e.g. Naïve Bayes)

- Assume some functional form for P(X,Y) (or P(X|Y) and P(Y))
- Estimate parameters of P(X|Y), P(Y) directly from training data
- Use Bayes rule to calculate P(Y|X)

Why not learn P(Y|X) directly? Or better yet, why not learn the decision boundary directly?

Discriminative classifiers (e.g. Logistic Regression)

- Assume some functional form for P(Y|X) or for the decision boundary
- Estimate parameters of P(Y|X) directly from training data

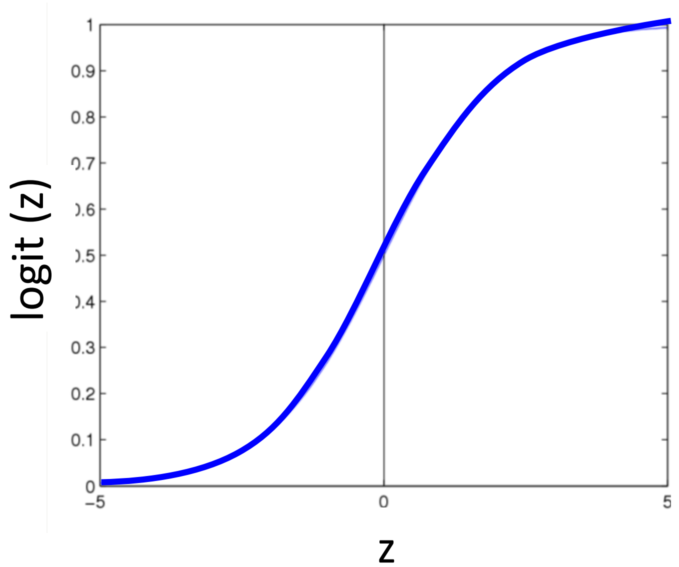
Logistic Regression

Assumes the following functional form for P(Y|X):

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

Logistic function applied to a linear function of the data

Logistic function (or Sigmoid): $\frac{1}{1+\exp(-z)}$



Features can be discrete or continuous!

Logistic Regression is a Linear Classifier!

Assumes the following functional form for P(Y|X):

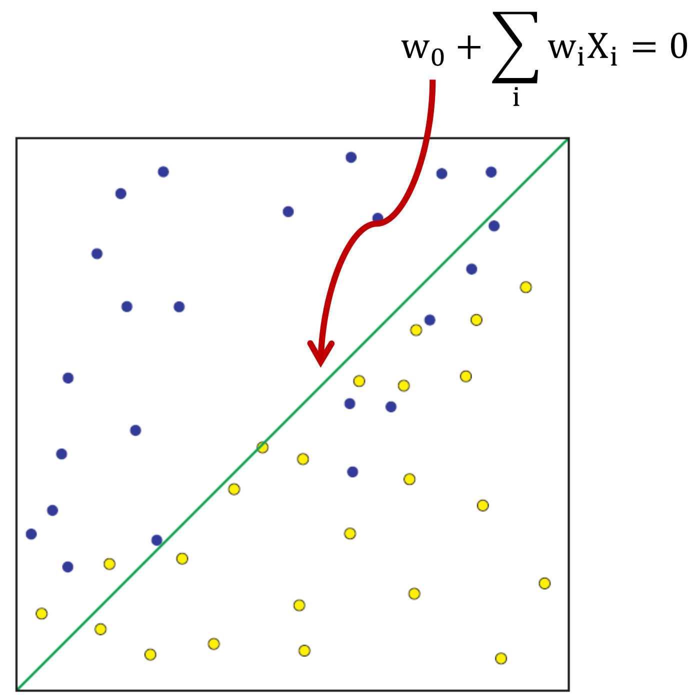
$$P(Y = 1|X) = \frac{1}{1 + \exp(-(w_0 + \sum_i w_i X_i))} = \frac{\exp(w_0 + \sum_i w_i X_i)}{\exp(w_0 + \sum_i w_i X_i) + 1}$$

Decision boundary:

$$P(Y = 1|X) > P(Y = 0|X) \ ?$$

$$w_0 + \sum_i w_i X_i > 0 \ ?$$

(Linear Decision Boundary)



Maximizing Conditional log Likelihood

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

$$= \sum_j \left[\mathrm{y}^j \left(\mathrm{w}_0 + \sum_{i=1}^{d} \mathrm{w}_i \mathrm{x}_i^j \right) - \ln \left(1 + \exp \left(\mathrm{w}_0 + \sum_{i=1}^{d} \mathrm{w}_i \mathrm{x}_i^j \right) \right) \right]$$

Good news: $\mathrm{l}(\mathbf{w})$ is concave in w. Local optimum = global optimum

Bad news: no closed-form solution to maximize $l(\mathbf{w})$

Good news: concave functions easy to optimize (unique maximum)

Gradient Ascent for Logistic Regression

Gradient ascent algorithm: iterate until change $< \epsilon$

$$w_0^{(t+1)} = w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})]$$

For $i = 1, \ldots, d$:

$$w_i^{(t+1)} = w_i^{(t)} + \eta \sum_j x_i^j [y^j - \underbrace{\hat{P}(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})}]$$

repeat

Predict what current weight thinks label Y should be

look at actual labels of the examples, compare them to our current predictions, and then for each example j we multiply that difference by the feature value x_i^j and then add them up.

That's all M(C)LE. How about MAP?

$$p(\mathbf{w} \mid Y, \mathbf{X}) \;\propto\; P(Y \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})$$

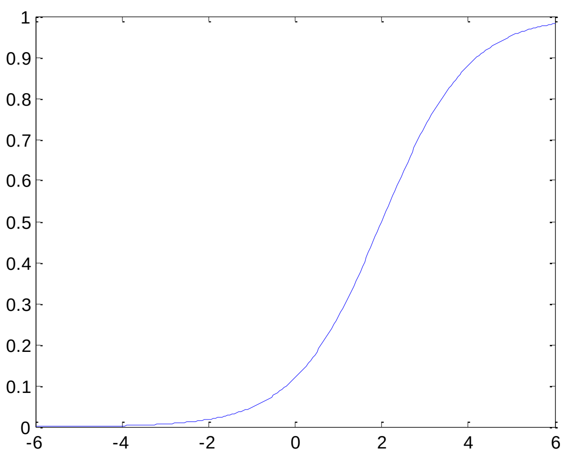
- One common approach is to define priors on **w**
  - Normal distribution, zero mean, identity covariance
  - "Pushes" parameters towards zero
- Corresponds to ***Regularization***
  - Helps avoid very large weights and overfitting
  - More on this later in the semester
- M(C)AP estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{n} P\left(y^j \mid \mathbf{x}^j, \mathbf{w}\right) \right]$$

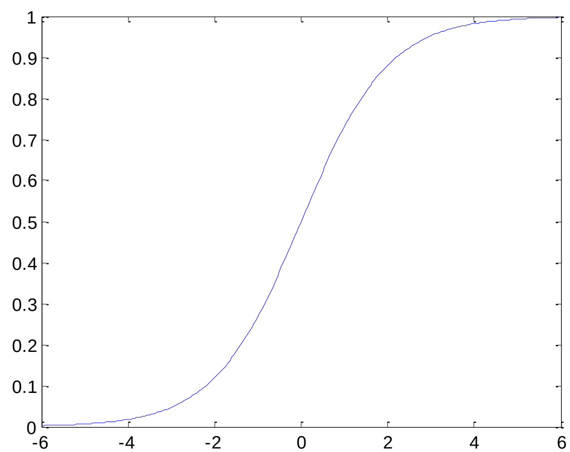
Understanding the sigmoid

$$g\left(w_0 + \sum_i w_i x_i\right) = \frac{1}{1 + \exp(w_0 + \sum_i w_i x_i)}$$

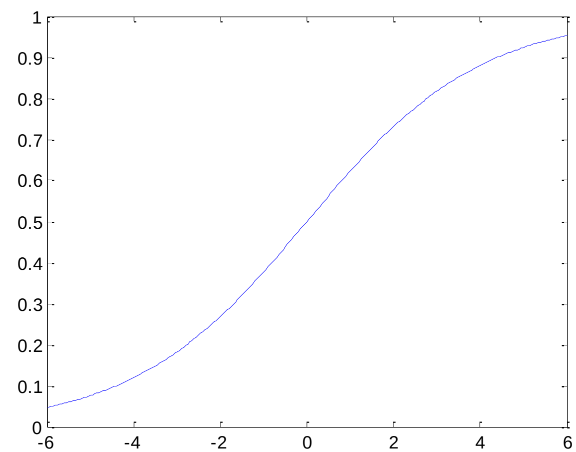
w_0=-2, w_1=-1



w_0=0, w_1=-1

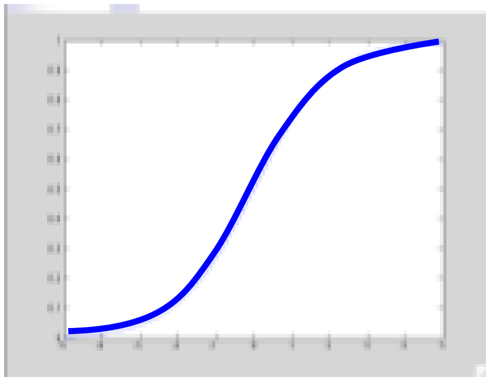


w_0=0, w_1=-0.5

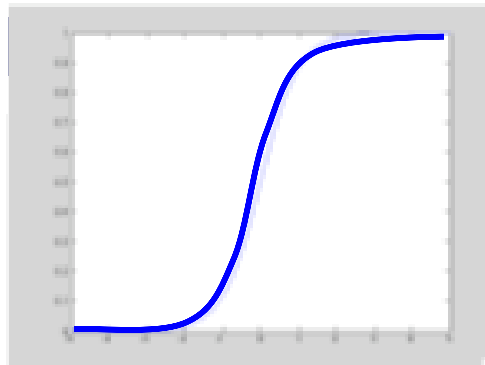


$$z = w_0 + \sum_i w_i x_i$$

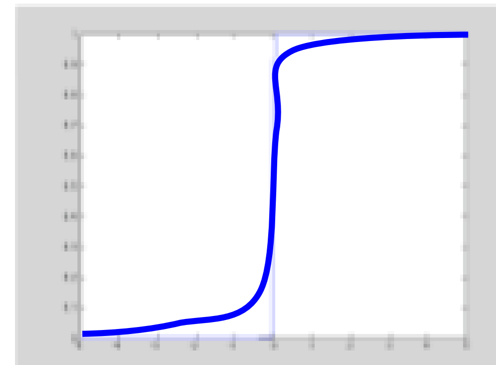
Large weights → Overfitting



$$\frac{1}{1+e^{-x}}$$

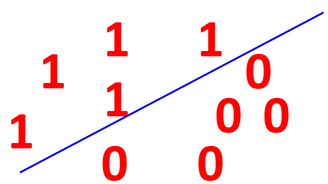


$$\frac{1}{1+e^{-2x}}$$



$$\frac{1}{1+e^{-100x}}$$

- Large weights lead to overfitting:



- Penalizing high weights can prevent overfitting…
  - again, more on this later in the semester

M(C)AP – Regularization

- Regularization

$$p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} e^{-w_i^2/2\kappa^2}$$

Zero-mean Gaussian prior

$$\arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \sum_{j=1}^{n} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w}) - \underbrace{\sum_{i=1}^{d} \frac{w_i^2}{2\kappa^2}}_{\boxed{\text{Penalizes large weights}}}$$

M(C)AP – Gradient

- Gradient

$$p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} e^{-w_i^2/2\kappa^2}$$

Zero-mean Gaussian prior

$$\frac{\partial}{\partial w_i} \ln \left[p(\mathbf{w}) \prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$= \underbrace{\frac{\partial}{\partial w_i} \ln p(\mathbf{w})}_{\propto -\frac{w_i}{\kappa^2}} + \underbrace{\frac{\partial}{\partial w_i} \ln \left[\prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]}_{\text{Same as before}}$$

$\boxed{\text{Extra term Penalizes large weights}}$

M(C)LE vs. M(C)AP

- Maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln\left[\prod_{j=1}^{n} P\left(y^j \mid \mathbf{x}^j, \mathbf{w}\right)\right]$$

$$w_i^{(t+1)} = w_i^{(t)} + \eta \sum_j x_i^j \left[y^j - \hat{P}\left(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)}\right)\right]$$

- Maximum conditional a posteriori estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln\left[p(\mathbf{w}) \prod_{j=1}^{n} P\left(y^j \mid \mathbf{x}^j, \mathbf{w}\right)\right]$$

$$w_i^{(t+1)} = w_i^{(t)} + \eta \left(-\frac{1}{\kappa^2} w_i^{(t)} + \sum_j x_i^j \left[y^j - \hat{P}\left(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)}\right)\right]\right)$$

Connection to Gaussian Naïve Bayes

There are several distributions that can lead to a linear decision boundary.

As another example, consider a generative model (GNB):

$$Y \sim \mathrm{Bernoulli}(\pi)$$

$$P(X_i \mid Y = y) = \frac{1}{\sqrt{2\pi\sigma_{i,y}^2}} \exp\left(\frac{-(X_i - \mu_{i,y})^2}{2\sigma_{i,y}^2}\right)$$

Gaussian class conditional densities

Assume variance is independent of class, i.e. $\sigma_{i,0}^2 = \sigma_{i,1}^2$

Connection to Gaussian Naïve Bayes

$$P(X_i \mid Y = y) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-(X_i - \mu_{i,y})^2}{2\sigma_i^2}\right)$$

Using conditionally independent assumption,

$$\log \frac{P(X \mid Y = 0)}{P(X \mid Y = 1)} = \log \prod_{i=1}^{d} \frac{P(X_i \mid Y = 0)}{P(X_i \mid Y = 1)}$$

Decision boundary:

$$\log \frac{P(Y = 0 \mid X)}{P(Y = 1 \mid X)} = \log \frac{P(Y = 0)P(X \mid Y = 0)}{P(Y = 1)P(X \mid Y = 1)} = \log \frac{1 - \pi}{\pi} + \log \frac{P(X \mid Y = 0)}{P(X \mid Y = 1)}$$

$$= \underbrace{\log \frac{1 - \pi}{\pi} + \sum_i \frac{\mu_{i,1}^2 - \mu_{i,0}^2}{2\sigma_i^2}}_{\text{Constant term}} + \underbrace{\sum_i \frac{\mu_{i,0} - \mu_{i,1}}{\sigma_i^2} X_i}_{\text{First-order term}} = w_0 + \sum_i w_i X_i$$

Gaussian Naïve Bayes vs. Logistic Regression

Set of Gaussian Naïve Bayes parameters (feature variance independent of class label)

⇔

Set of Logistic Regression parameters

- Representation equivalence
  - **But only in a special case!!!** (GNB with class-independent variances)
- But what's the difference???
- **LR makes no assumptions about** P(**X**|Y) **in learning**!!!
- **Loss function!!!**
  - Optimize different functions ! Obtain different solutions

What you should know

- LR is a linear classifier: decision rule is a hyperplane
- LR optimized by conditional likelihood
  - no closed-form solution
  - concave $\Rightarrow$ global optimum with gradient ascent
  - Maximum conditional a posteriori corresponds to regularization
- Gaussian Naïve Bayes with class-independent variances representationally equivalent to LR
  - Solution differs because of objective (loss) function