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

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

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:

$$P(Y = 1|X) > P(Y = 0|X) \iff w_0 + \sum_i w_i X_i > 0$$

(Linear Decision Boundary)
Maximizing Conditional log Likelihood

$$\max_w l(w) \equiv \ln \prod_j P(y^j|x^j, w)$$

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

**Good news:** $l(w)$ is concave in $w$. Local optimum = global optimum

**Bad news:** no closed-form solution to maximize $l(w)$

**Good news:** concave functions easy to optimize (unique maximum)
Gradient Ascent for Logistic Regression

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

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

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

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

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?

\[ p(w \mid Y, X) \propto P(Y \mid X, w)p(w) \]

• One common approach is to define priors on \( w \)
  – Normal distribution, zero mean, identity covariance
  – “Pushes” parameters towards zero

• Corresponds to \textit{Regularization}
  – Helps avoid very large weights and overfitting
  – More on this later in the semester

• M(C)AP estimate

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

- Large weights lead to overfitting:
  
  \[
  \frac{1}{1 + e^{-x}} \quad \frac{1}{1 + e^{-2x}} \quad \frac{1}{1 + e^{-100x}}
  \]

- Penalizing high weights can prevent overfitting...
  
  - again, more on this later in the semester
M(C)AP – Regularization

- **Regularization**

\[
\arg \max_w \ln \left[ p(w) \prod_{j=1}^n P(y^j | x^j, w) \right]
\]

\[
w^* = \arg \max_w \sum_{j=1}^n \ln P(y^j | x^j, w) - \sum_{i=1}^d \frac{w_i^2}{2\kappa^2}
\]

- **Zero-mean Gaussian prior**

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

Penalizes large weights
M(C)AP – Gradient

- Gradient

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

Same as before

Extra term Penalizes large weights

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

Zero-mean Gaussian prior
M(C)LE vs. M(C)AP

• Maximum conditional likelihood estimate

\[ w^* = \arg \max_w \ln \left[ \prod_{j=1}^n P(y^j | x^j, w) \right] \]

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

• Maximum conditional a posteriori estimate

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

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

\[ Y \sim \text{Bernoulli}(\pi) \]

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

**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

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

Using conditionally independent assumption,

\[
\log \frac{P(X | Y = 0)}{P(X | Y = 1)} = \log \prod_{i=1}^{d} \frac{P(X_i | Y = 0)}{P(X_i | Y = 1)}
\]

Decision boundary:

\[
\log \frac{P(Y = 0 | X)}{P(Y = 1 | X)} = \log \frac{P(Y = 0)P(X | Y = 0)}{P(Y = 1)P(X | Y = 1)} = \log \frac{1 - \pi}{\pi} + \log \frac{P(X | Y = 0)}{P(X | Y = 1)}
\]

\[
= \log \frac{1 - \pi}{\pi} + \sum_i \frac{\mu_{i,1}^2 - \mu_{i,0}^2}{2\sigma_i^2} + \sum_i \frac{\mu_{i,0} - \mu_{i,1}}{\sigma_i^2} X_i = w_0 + \sum_i w_i X_i
\]

- Constant term
- First-order term
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