Logistic Regression

Learn $P(Y|X)$ directly!
- Assume a particular functional form
- Sigmoid applied to a linear function of the data:

$$P(Y = 1|X) = \frac{1}{1 + \exp(-z)}$$

Logistic function (or Sigmoid):

Features can be discrete or continuous!
Logistic Regression – a Linear classifier

\[ g(w_0 + \sum_i w_ix_i) = \frac{1}{1 + e^{-w_0 + \sum_i w_ix_i}} \]

Logistic regression for more than 2 classes

- Logistic regression in more general case, where

\[ Y \in \{y_1, \ldots, y_R\} \]

\[ P(Y = y_1 | x) \propto e^{w_0 + \sum_i w_i x_i} \]

\[ P(Y = y_2 | x) \propto e^{w_0 + \sum_i w_i x_i} \]

\[ : \]

\[ P(Y = y_R | x) = 1 - P(Y = y_1 | x) - P(Y = y_2 | x) \]
Logistic regression more generally

- Logistic regression in more general case, where
  \( Y \in \{y_1, \ldots, y_R\} \)

  for \( k < R \)
  \[
  P(Y = y_k | X) = \frac{\exp(w_{k0} + \sum_{i=1}^{n} w_{ki} x_i)}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji} x_i)}
  \]

  for \( k = R \) (normalization, so no weights for this class)
  \[
  P(Y = y_R | X) = \frac{1}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji} x_i)}
  \]

  Features can be discrete or continuous!

Loss functions: Likelihood v. Conditional Likelihood

- Generative (Naïve Bayes) Loss function:
  Data likelihood
  \[
  \ln P(D | w) = \sum_{j=1}^{N} \ln P(x^j, y^j | w) = \sum_{(i,j)} \ln P(y^j | x^i, w) + \sum_{j=1}^{N} \ln P(x^j | w)
  \]

- Discriminative models cannot compute \( P(x | w) \)!
- But, discriminative (logistic regression) loss function:
  Conditional Data Likelihood
  \[
  \ln P(D_Y | D_X, w) = \sum_{j=1}^{N} \ln P(y^j | x^j, w)
  \]

  Doesn’t waste effort learning \( P(x) \) – focuses on \( P(Y | X) \) all that matters for classification
Expressing Conditional Log Likelihood

\[ l(w) = \sum_j y_j \ln P(y_j = 1|x_j, w) + (1 - y_j) \ln P(y_j = 0|x_j, w) \]

\[ \sum_j y_j \ln \frac{e^{w_0 + \sum x_i w_i}}{1 + e^{w_0 + \sum x_i w_i}} + (1 - y_j) \ln \frac{1}{1 + e^{w_0 + \sum x_i w_i}} \]

Maximizing Conditional Log Likelihood

\[ l(w) = \ln \prod_j P(y_j|x_j, w) \]

\[ \sum_{j=1}^N y_j (w_0 + \sum_i w_i x_i^j) - \ln (1 + \exp(w_0 + \sum_i w_i x_i^j)) \]

Good news: \( l(w) \) is concave function of \( w \) and no locally optimal solutions

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

Good news: concave functions easy to optimize
Optimizing concave function –
Gradient ascent

- Conditional likelihood for Logistic Regression is concave. Find optimum with gradient ascent

Gradient:
$$\nabla_w l(w) = \left[ \frac{\partial l(w)}{\partial w_0}, \ldots, \frac{\partial l(w)}{\partial w_n} \right]^T$$

Update rule:
$$\Delta w = \eta \nabla_w l(w)$$

$$w^{(t+1)}_i = w^{(t)}_i + \eta \frac{\partial l(w)}{\partial w_i}$$

- Gradient ascent is simplest of optimization approaches
  - e.g., Conjugate gradient ascent much better (see reading)

Maximize Conditional Log Likelihood:
Gradient ascent

$$l(w) = \sum_j y_j (w_0 + \sum_i w_i x_{ij}) - \ln (1 + \exp (w_0 + \sum_i w_i x_{ij}))$$

$$\frac{\partial l(w)}{\partial w_i} = \sum_j \frac{y_j}{1 + \exp (w_0 + \sum_i w_i x_{ij})} x_{ij} - \frac{\exp (w_0 + \sum_i w_i x_{ij})}{1 + \exp (w_0 + \sum_i w_i x_{ij})} x_{ij}$$

$$\frac{\partial l(w)}{\partial w_i} = \sum_j x_{ij} \left( y_j - \frac{\exp (w_0 + \sum_i w_i x_{ij})}{1 + \exp (w_0 + \sum_i w_i x_{ij})} \right)$$
Gradient Descent for LR

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

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

For $i=1,...,n$,

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

repeat

That’s all M(C)LE. How about MAP?

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

- One common approach is to define priors on $\mathbf{w}$
  - 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

- MAP estimate

$$\mathbf{w}^* = \arg \max_w \ln \left[ p(\mathbf{w}) \prod_{j=1}^{N} P(y^j | \mathbf{x}^j, \mathbf{w}) \right]$$
M(C)AP as Regularization

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

= \ln p(w) + \sum_{j} \ln P(y^j | x^j, w)

= \ln \prod_{i} \frac{1}{\sqrt{2\pi}} e^{-\frac{w^2}{2\sigma^2}}

= \ln \prod_{i} \frac{1}{\sqrt{2\pi}} e^{-\frac{w^2}{2\sigma^2}}

= \text{Zero mean Gaussian prior}

Penalizes high weights, also applicable in linear regression

Large parameters → Overfitting

- If data is linearly separable, weights go to infinity
- Leads to overfitting:
  - Penalizing high weights can prevent overfitting…
    - again, more on this later in the semester

If \( \sigma = 0 \), then \( p(y^j | x^j, w) \) is a step function.

If \( \sigma = 1 \), then \( p(y^j | x^j, w) \) is a sigmoid function.

If \( \sigma = 100 \), then \( p(y^j | x^j, w) \) is almost a step function.
Gradient of M(C)AP

\[ \frac{\partial}{\partial w_i} \ln \left[ p(w) \prod_{j=1}^{N} P(y_j \mid x_j, w) \right] \]

\[ p(w) = \prod_{i} \frac{1}{\sqrt{2\pi} \sigma_i} e^{-\frac{(w_i - \mu_i)^2}{2\sigma_i^2}} \]

\[ \frac{\partial}{\partial w_i} \ln p(w) + \frac{\partial}{\partial w_i} \ln \left( \frac{1}{\sqrt{2\pi} \sigma_i} \right) = -w_i \]

As before:

\[ \frac{\partial}{\partial w_i} \left( \text{const.} + \sum_{i} \frac{w_i^2}{2\sigma_i^2} \right) = \frac{-w_i}{\sigma_i^2} \]

Extra turn:

- Push: \( w_i \) towards \( \phi \)

If \( w_i > 0 \): \( \frac{-w_i}{\sigma_i^2} \) is very negative \( \Rightarrow \) push \( w_i \) to \( \phi \)

If \( w_i < 0 \): \( \frac{-w_i}{\sigma_i^2} \) is very positive \( \Rightarrow \) push \( w_i \) to \( \phi \)

MLE vs MAP

in practice, ALWAYS REGULARIZE YOUR LR

Typically don't regularize \( w_0 \)

Maximum conditional likelihood estimate

\[ w^* = \arg \max_w \ln \left[ \prod_{j=1}^{N} P(y_j \mid x_j, w) \right] \]

\[ w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_{ij} [y_j - \hat{P}(Y^j = 1 \mid x_i, w)] \]

Maximum conditional a posteriori estimate

\[ w^* = \arg \max_w \ln \left[ p(w) \prod_{j=1}^{N} P(y_j \mid x_j, w) \right] \]

\[ \lambda = \frac{1}{\sigma_i^2} \]

\[ w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\lambda w_i^{(t)} + \sum_j x_{ij} [y_j - \hat{P}(Y^j = 1 \mid x_i, w)] \right\} \]
Logistic regression v. Naïve Bayes

- Consider learning \( f: X \rightarrow Y \), where
  - \( X \) is a vector of real-valued features, \( < X_1 \ldots X_n > \)
  - \( Y \) is boolean
- Could use a Gaussian Naïve Bayes classifier
  - assume all \( X_i \) are conditionally independent given \( Y \)
  - model \( P(X_i | Y = y_k) \) as Gaussian \( N(\mu_{ik}, \sigma_i) \)
  - model \( P(Y) \) as Bernoulli(\( \theta, 1-\theta \))

- What does that imply about the form of \( P(Y|X) \)?

\[
P(Y = 1|X = < X_1, \ldots X_n >) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}
\]

Cool!!!!

Derive form for \( P(Y|X) \) for continuous \( X_i \)

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

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

\[
= \frac{1}{1 + \exp(\ln \frac{P(Y = 0)P(X|Y = 0)}{P(Y = 1)P(X|Y = 1)})}
\]

\[
= \frac{1}{1 + \exp(\ln \frac{1}{\theta} + \sum_i \ln \frac{P(X_i|Y = 0)}{P(X_i|Y = 1)})}
\]

\[
\text{only assumption}
\]

\[
\text{this is } \text{NB}
\]

\[
\text{let's check}
\]

\[
\text{only assumption}
\]

\[
\text{NB}
\]

\[
\text{reminds me of}
\]

\[
\text{w} \cdot X
\]
Ratio of class-conditional probabilities

\[
\ln \frac{P(X_i | Y = 0)}{P(X_i | Y = 1)} = \frac{-(x_i - \mu_i)^2}{2\sigma_i^2}
\]

\[
= \frac{-(x_i - \mu_i)^2}{2\sigma_i^2} + \frac{(x_i - \mu_i)^2}{2\sigma_i^2}
\]

\[
= -\frac{2\mu_i x_i - \mu_i^2}{2\sigma_i^2} + \frac{\mu_i^2}{2\sigma_i^2}
\]

\[
= \frac{(w_i X_i - \mu_i)}{\sigma_i^2} + \frac{\mu_i^2}{2\sigma_i^2}
\]

Derive form for \( P(Y|X) \) for continuous \( X_i \)

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

\[
= \frac{1}{1 + \exp(\ln \frac{1}{\theta} + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}
\]

\[
P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}
\]

\[
w_0 = \ln \left( \frac{1}{\theta} \right) + \sum_i \frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2}
\]
Gaussian Naïve Bayes v. Logistic Regression

- 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

Naïve Bayes vs Logistic Regression

Consider $Y$ boolean, $X_i$ continuous, $X = \langle X_1 \ldots X_n \rangle$

Number of parameters:
- NB: $4n + 1$
- LR: $n + 1$

Estimation method:
- NB parameter estimates are uncoupled
- LR parameter estimates are coupled
G. Naïve Bayes vs. Logistic Regression 1

Generative and Discriminative classifiers

Asymptotic comparison (# training examples → infinity)
- when model correct
  - GNB (with class independent variances) and LR produce identical classifiers
- when model incorrect
  - LR is less biased – does not assume conditional independence
    - therefore LR expected to outperform GNB

G. Naïve Bayes vs. Logistic Regression 2

Generative and Discriminative classifiers

Non-asymptotic analysis
- convergence rate of parameter estimates, \( n = \# \) of attributes in \( X \)
  - Size of training data to get close to infinite data solution
    - GNB needs \( O(\log n) \) samples
    - LR needs \( O(n) \) samples
  - GNB converges more quickly to its (perhaps less helpful) asymptotic estimates
Some experiments from UCI data sets

What you should know about Logistic Regression (LR)

- Gaussian Naïve Bayes with class-independent variances representationally equivalent to LR
  - Solution differs because of objective (loss) function
- In general, NB and LR make different assumptions
  - NB: Features independent given class → assumption on \( P(X|Y) \)
  - LR: Functional form of \( P(Y|X) \), no assumption on \( P(X|Y) \)
- LR is a linear classifier
  - Decision rule is a hyperplane
- LR optimized by conditional likelihood
  - No closed-form solution
  - Concave! Global optimum with gradient ascent
  - Maximum conditional a posteriori corresponds to regularization
- Convergence rates
  - GNB (usually) needs less data
  - LR (usually) gets to better solutions in the limit

Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning Repository. Plots are of generalization error vs. \( m \) (averaged over 1000 runs). Open circles: dashed line is logistic regression; solid line is naive Bayes.