CSE546: Logistic Regression
Winter 2012

Luke Zettlemoyer

Slides adapted from Carlos Guestrin
Let's take another probabilistic approach!!!

• Previously: directly estimate the data distribution $P(X,Y)$!
  – challenging due to size of distribution!
  – make Naïve Bayes assumption: only need $P(X_i|Y)$!

• But wait, we classify according to:
  – $\max_Y P(Y|X)$

• Why not learn $P(Y|X)$ directly?
Logistic Regression

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

\[
P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}
\]
\[
P(Y = 0|X) = \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}
\]

Logistic function (Sigmoid):

\[
\frac{1}{1 + \exp(-z)}
\]

Features can be discrete or continuous!
Logistic Regression: decision boundary

\[
P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)} \quad P(Y = 0|X) = \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}
\]

- **Prediction**: Output the \( Y \) with highest \( P(Y|X) \)
  - For binary \( Y \), output \( Y=0 \) if
    \[
    1 < \frac{P(Y = 0|X)}{P(Y = 1|X)}
    \]
    \[
    1 < \exp(w_0 + \sum_{i=1}^{n} w_i X_i)
    \]
    \[
    0 < w_0 + \sum_{i=1}^{n} w_i X_i
    \]

A Linear Classifier!
Logistic regression for discrete classification

Logistic regression in more general case, where set of possible $Y$ is $\{y_1, \ldots, y_R\}$

- Define a weight vector $w_i$ for each $y_i$, $i=1, \ldots, R-1$

\[
P(Y = 1|X) \propto \exp(w_{10} + \sum_i w_{1i}X_i)
\]
\[
P(Y = 2|X) \propto \exp(w_{20} + \sum_i w_{2i}X_i)
\]
\[
\ldots
\]
\[
P(Y = r|X) = 1 - \sum_{j=1}^{r-1} P(Y = j|X)
\]
Logistic regression: discrete Y

- Logistic regression in more general case, where \( Y \) is in the set \( \{y_1, ..., y_R\} \)

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 / Learning Objectives: Likelihood v. Conditional Likelihood

• Generative (Naïve Bayes) Loss function:
  **Data likelihood**

\[
\ln P(D \mid w) = \sum_{j=1}^{N} \ln P(x^j, y^j \mid w) \\
= \sum_{j=1}^{N} \ln P(y^j \mid x^j, w) + \sum_{j=1}^{N} \ln P(x^j \mid w)
\]

• But, discriminative (logistic regression) loss function:
  **Conditional Data Likelihood**

\[
\ln P(D_Y \mid D_X, w) = \sum_{j=1}^{N} \ln P(y^j \mid x^j, w)
\]

  – Doesn’t waste effort learning \(P(X)\) – focuses on \(P(Y \mid X)\) all that matters for classification
  – Discriminative models cannot compute \(P(x^j \mid w)\)!
Conditional Log Likelihood  
(the binary case only)

\[ l(w) \equiv \sum_j \ln P(y^j|x^j, w) \]

\[ l(w) = \sum_j y^j \ln P(y^j = 1|x^j, w) + (1 - y^j) \ln P(y^j = 0|x^j, w) \]

\[ P(Y = 0|X, w) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)} \]

\[ P(Y = 1|X, w) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)} \]

equal because \( y^j \) is in \{0, 1\}

remaining steps: substitute definitions, expand logs, and simplify

\[ \ldots \]

\[ j \]

\[ i \]

\[ i \]
Logistic Regression Parameter Estimation: Maximize Conditional Log Likelihood

\[ l(w) \equiv \ln \prod_j P(y^j | x^j, w) \]

\[ = \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + \exp(w_0 + \sum_i^n w_i x_i^j)) \]

**Good news**: \( l(w) \) is concave function of \( w \)

\[ \rightarrow \text{no locally optimal solutions!} \]

**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!

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

Update rule:
\[ \Delta w = \eta \nabla_w l(w) \]
\[ w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(w)}{\partial w_i} \]

• Gradient ascent is simplest of optimization approaches
  – e.g., Conjugate gradient ascent much better (see reading)
Maximize Conditional Log Likelihood: Gradient ascent

\[
l(w) = \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + \exp(w_0 + \sum_i^n w_i x_i^j))
\]

\[
\frac{\partial l(w)}{\partial w_i} = \sum_j \left[ \frac{\partial}{\partial w} y^j (w_0 + \sum_i^n w_i x_i^j) - \frac{\partial}{\partial w} \ln \left(1 + \exp(w_0 + \sum_i^n w_i x_i^j)\right) \right]
\]

\[
= \sum_j \left[ y^j x_i^j - \frac{x_i^j \exp(w_0 + \sum_i^n w_i x_i^j)}{1 + \exp(w_0 + \sum_i^n w_i x_i^j)} \right]
\]

\[
= \sum_j x_i^j \left[ y^j - \frac{\exp(w_0 + \sum_i^n w_i x_i^j)}{1 + \exp(w_0 + \sum_i^n w_i x_i^j)} \right]
\]

\[
\frac{\partial l(w)}{\partial w_i} = \sum_j x_i^j (y^j - P(Y^j = 1|x^j, w))
\]

\[
P(Y = 1|X, W) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)}
\]
Gradient Descent for LR

Gradient ascent algorithm: (learning rate $\eta > 0$)

do:

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

For $i=1\ldots n$: (iterate over weights)

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

until "change" < $\varepsilon$

Loop over training examples!
Large parameters...

\[
\frac{1}{1 + e^{-ax}}
\]

- Maximum likelihood solution: prefers higher weights
  - higher likelihood of (properly classified) examples close to decision boundary
  - larger influence of corresponding features on decision
  - *can cause overfitting***

- Regularization: penalize high weights
  - again, more on this later in the quarter
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

- Often called **Regularization**
  - Helps avoid very large weights and overfitting

- MAP estimate:
  \[
  w^* = \arg \max_w \ln \left[ p(w) \prod_{j=1}^N P(y^j \mid x^j, w) \right]
  \]
M(C)AP as Regularization

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

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

- Add \( \log p(w) \) to objective:

\[ \ln p(w) \propto -\frac{\lambda}{2} \sum_i w_i^2 \]

\[ \frac{\partial \ln p(w)}{\partial w_i} = -\lambda w_i \]

- Quadratic penalty: drives weights towards zero
- Adds a negative linear term to the gradients

Penalizes high weights, also applicable in linear regression
MLE vs. MAP

• Maximum conditional likelihood estimate

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

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

• Maximum conditional a posteriori estimate

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

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

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

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

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

$$= \frac{1}{1 + \exp\left(\ln \frac{1-\theta}{\theta} + \sum_i \ln \frac{P(X_i|Y = 0)}{P(X_i|Y = 1)}\right)}$$

Looks like a setting for $w_0$? Can we solve for $w_i$?

- Yes, but only in Gaussian case
Ratio of class-conditional probabilities

\[
\ln \frac{P(X_i | Y = 0)}{P(X_i | Y = 1)} = \ln \left[ \frac{\frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x_i - \mu_{i0})^2}{2\sigma_i^2}}}{\frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x_i - \mu_{i1})^2}{2\sigma_i^2}}} \right]
\]

\[
= -\frac{(x_i - \mu_{i0})^2}{2\sigma_i^2} + \frac{(x_i - \mu_{i1})^2}{2\sigma_i^2}
\]

\[
= \mu_{i0} + \mu_{i1} \frac{\mu_{i0}^2 + \mu_{i1}^2}{2\sigma_i^2} x_i
\]

Linear function! Coefficients expressed with original Gaussian parameters!
Derive form for $P(Y|X)$ for continuous $X_i$
Gaussian Naïve Bayes vs. 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, $X = <X_1, ..., X_n>$

**Number of parameters:**
- Naïve Bayes: $4n + 1$
- Logistic Regression: $n + 1$

**Estimation method:**
- Naïve Bayes parameter estimates are uncoupled
- Logistic Regression parameter estimates are coupled
Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

• Generative vs. 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
Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

• Generative vs. Discriminative classifiers
• Non-asymptotic analysis
  – convergence rate of parameter estimates, 
    \( (n = \# \text{ of attributes in } X) \)
  • Size of training data to get close to infinite data solution
  • Naïve Bayes needs \( O(\log n) \) samples
  • Logistic Regression needs \( O(n) \) samples

• GNB converges more quickly to its (perhaps less helpful) asymptotic estimates
Some experiments from UCI data sets

Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. $m$ (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.
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