Thus far, regression: predict a continuous value given some inputs.
Weather prediction revisited

Reading Your Brain, Simple Example

Pairwise classification accuracy: 85%

[Mitchell et al.]
Classification

- Learn: \( h : X \mapsto Y \)
  - \( X \) – features
  - \( Y \) – target classes

- Conditional probability: \( P(Y|X) \)
  
- Suppose you know \( P(Y|X) \) exactly, how should you classify?
  - Bayes optimal classifier:
    \[
    g(x) = \arg\max_y P(Y=y|X=x)
    \]

- How do we estimate \( P(Y|X) \)?

Link Functions

- Estimating \( P(Y|X) \): Why not use standard linear regression?
  \[
  P(Y|X) = \omega_0 + \sum_{i} \omega_i x_i
  \]
  \( R \in (-\infty, \infty) \)

- Combing regression and probability?
  - Need a mapping from real values to \([0,1]\)
  - A link function!
    \( g : R \rightarrow [0,1] \)
    
    Many \( g \), but today a simple one
Logistic Regression

- Learn $P(Y|X)$ directly
  - Assume a particular functional form for link function
  - Sigmoid applied to a linear function of the input features:
    $$ P(Y = 0|X, W) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)} $$

$$ P(Y = 1|X, w) = 1 - P(Y = 0|X, w) = \frac{e^{w_0 + \sum_i w_i X_i}}{1 + e^{w_0 + \sum_i w_i X_i}} $$

- Features can be discrete or continuous!

Understanding the sigmoid

$$ g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}} $$

- $w_0 = -2, w_1 = -1$
- $w_0 = 0, w_1 = -1$
- $w_0 = 0, w_1 = -0.5$
Logistic Regression – a Linear classifier

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

Very convenient!

\[ P(Y = 0 \mid X = \langle X_1, \ldots, X_n \rangle) = \frac{1}{1 + \exp(w_0 + \sum_i w_iX_i)} \]

implies

\[ P(Y = 1 \mid X = \langle X_1, \ldots, X_n \rangle) = \frac{\exp(w_0 + \sum_i w_iX_i)}{1 + \exp(w_0 + \sum_i w_iX_i)} \]

implies

\[ \frac{1}{P(Y = 0 \mid X)} < \frac{P(Y = 1 \mid X)}{P(Y = 0 \mid X)} = \exp(w_0 + \sum_i w_iX_i) \]

implies

\[ \log \text{odds} \approx \sum_i w_iX_i \]

\[ \hat{y} = 1 \iff \log \text{odds} > 0 \]

linear classification rule!
Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave. Find optimum with gradient ascent
  - Gradient: $\nabla_w l(w) = \left[ \frac{\partial l(w)}{\partial w_0}, \ldots, \frac{\partial l(w)}{\partial w_n} \right]'$

  - Gradient ascent is simplest of optimization approaches
    - e.g., Conjugate gradient ascent can be much better
      - Step size, $\eta > 0$
      - Update rule: $\Delta w = \eta \nabla_w l(w)$
      - $w_i(t+1) \leftarrow w_i(t) + \eta \frac{\partial l(w)}{\partial w_i}$

- Discriminative (logistic regression) loss function:
  - Conditional Data Likelihood
  - $\ln P(D_Y | D_X, w) = \sum_{j=1}^{N} \ln \left( \prod_{i=1}^{N} P(y^j_i | x^j, w) \right) = \arg\max_w \ln \left( \prod_{i=1}^{N} P(y^j_i | x^j, w) \right)$
  - $\ln P(D_Y | D_X, w) = \sum_{j=1}^{N} \ln P(y^j_i | x^j, w)$
Expressing Conditional Log Likelihood

\[ l(w) = \sum_{j=1}^{N} y_j \ln P(Y = 1|x_j, w) + (1 - y_j) \ln \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}} \]

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

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

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

Maximizing Conditional Log Likelihood

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

\[ = \sum_j y_j (w_0 + \sum_i w_i x_i) - \ln (1 + \exp(w_0 + \sum_i w_i x_i)) \]

Good news: \( l(w) \) is concave function of \( w \), no local optima problems

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

Good news: concave functions easy to optimize

\[ \text{gradient ascent} \]
Maximize Conditional Log Likelihood: Gradient ascent

\[
\frac{\partial}{\partial \omega} \ln l(\omega) = \frac{f(\omega)}{\exp(f(\omega))}
\]

\[
l(\omega) = \sum_{j=1}^{N} y_j(\omega_0 + \sum_{i=1}^{k} w_i x_{ij}) - \ln(1 + \exp(\omega_0 + \sum_{i=1}^{k} w_i x_{ij}))
\]

\[
\frac{\partial l}{\partial \omega} = \sum_{i=1}^{N} \left[ y_i x_i - \frac{e^{\omega_0 + \sum_{i=1}^{k} w_i x_{ij}}}{1 + e^{\omega_0 + \sum_{i=1}^{k} w_i x_{ij}}} \right] p(y=1|x_i, \omega)
\]

\[
\frac{\partial l}{\partial w_k} = \sum_{i=1}^{N} x_{ik} (y_i - p(y=1|x_i, \omega))
\]

Gradient Ascent for LR

Gradient ascent algorithm: iterate until change < \varepsilon

\[
w_{0}^{(t+1)} \leftarrow w_{0}^{(t)} + \eta \sum_{j=1}^{N} [y_i - \hat{P}(Y=1|x_i, w_0)]
\]

For \(i=1, \ldots, k,\)

\[
w_{i}^{(t+1)} \leftarrow w_{i}^{(t)} + \eta \sum_{j=1}^{N} x_{ij} [y_i - \hat{P}(Y=1|x_i, w_i)]
\]

repeat
Regularization in linear regression

- Overfitting usually leads to very large parameter choices, e.g.:
  \[-2.2 + 3.1 X - 0.30 X^2\]
  \[-1.1 + 4,700,910.7 X - 8,585,638.4 X^2 + \ldots\]

- Regularized least-squares (a.k.a. ridge regression), for $\lambda > 0$:
  \[w^* = \arg \min_w \sum_j \left( t(x_j) - \sum_i w_i h_i(x_j) \right)^2 + \lambda \sum_i w_i^2 \]

Linear Separability

- $w_0 + \sum_i w_i x_i > 0$
- $2 w_0 + \sum_i w_i x_i > 0$
- $2 w_0 + \sum_i w_i x_i < 0$
- $w_0 + \sum_i w_i x_i < 0$

- More confident
- Log likelihood will be high
Large parameters $\rightarrow$ Overfitting

- If data is linearly separable, weights go to infinity:
  $P(y=0 \mid w, x) \rightarrow 1$, $\|w\| \rightarrow \infty$

- In general, leads to overfitting:
  - Penalizing high weights can prevent overfitting...

Regularized Conditional Log Likelihood

- Add regularization penalty, e.g., $L_2$:
  $\ell(w) = \ln \prod_{j=1}^{N} P(y^j \mid x^j, w) - \frac{\lambda}{2} \|w\|_2^2$

- Practical note about $w_0$:
  - Don't regularize $w_0$.

- Gradient of regularized likelihood:
  $\frac{\partial \ell}{\partial w} = \text{same as before} + \frac{\partial}{\partial w_0}$
Standard v. Regularized Updates

Maximum conditional likelihood estimate

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

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

Regularized maximum conditional likelihood estimate

\[ w^* = \arg \max_w \ln \prod_{j=1}^{N} P(y^j|x^j, w) - \frac{\lambda}{2} \sum_{i=1}^{k} w_i^2 \]

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

Please Stop!! Stopping criterion

\[ \ell(w) = \ln \prod_j P(y^j|x^j, w)) - \lambda ||w||_2^2 \]

- When do we stop doing gradient descent? \( \epsilon > 0 \)
  \[ \ell(w^*) - \ell(w^{(t)}) < \epsilon \]

- Because \( \ell(w) \) is strongly concave:
  - i.e., because of some technical condition
  \[ \ell(w^*) - \ell(w) \leq \frac{1}{2\lambda} || \nabla \ell(w) ||_2^2 < \epsilon \]

- Thus, stop when:
  \[ || \nabla \ell(w) ||_2^2 < 2\lambda \epsilon \]

©Carlos Guestrin 2005-2014
Digression: Logistic regression for more than 2 classes

- Logistic regression in more general case (C classes), where $Y \in \{0, \ldots, C-1\}$

  For $c > 0$
  \[
  P(Y = c| x, w) = \frac{\exp(w_c + \sum_{i=1}^{k} w_{ci}x_i)}{1 + \sum_{c' = 1}^{C-1} \exp(w_{c'} + \sum_{i=1}^{k} w_{c'i}x_i)}
  \]

  For $c = 0$ (normalization, so no weights for this class)
  \[
  P(Y = 0| x, w) = \frac{1}{1 + \sum_{c' = 1}^{C-1} \exp(w_{c'} + \sum_{i=1}^{k} w_{c'i}x_i)}
  \]

Learning procedure is basically the same as what we derived!