Logistic Regression

Classification: $\{(x^{(1)}, y^{(1)}), ..., (x^{(N)}, y^{(N)})\}, x \in R^D, y \in \{0, 1\}$

Linear classifier in higher dimension:

Hyperplace: $H = \{x : w^T x = 0\}, H^+ = \{x : w^T x > 0\}, H^- = \{x : w^T x < 0\}$

Prediction using a decision boundary:

$$h(x) = \begin{cases} 1 & w^T x \geq 0 \\ 0 & w^T x < 0 \end{cases}$$

Estimating probabilities — logistic (sigmoid) function

$z(x) = w^T x \to (-\infty, +\infty)$

$\sigma(z(x)) = \frac{1}{1+e^{-z(x)}} \to (0, 1)$

How can we find best hyperplane w?

Data → Estimation

T, H, T, H, T → to predict θ (probability of head), find θ that maximizes $(1-\theta)\theta(1-\theta)\theta(1-\theta)$

Likelihood function $L(\theta) = p(D|\theta) = \theta^{N_H}(1-\theta)^{N_T}, l(\theta) = ln(L(\theta))$

Maximum Likelihood Estimation (MLE): maximize $l(\theta)$

Extend this to conditional likelihood:

$p(y=1|x; w) = \sigma(w^T x) = \frac{1}{1+e^{-w^T x}}$

$p(y=0|x; w) = 1 - \sigma(w^T x) = 1 - \frac{1}{1+e^{-w^T x}}$

$L(w) = \Pi_{i=1}^N \sigma(w^T x^{(i)})^{y^{(i)}}(1-\sigma(w^T x^{(i)}))^{1-y^{(i)}}$

$l(w) = \Sigma_{i=1}^N [y^{(i)} \ln \sigma(w^T x^{(i)}) + (1-y(i)) \ln(1-\sigma(w^T x^{(i)}))]$

Example: $y^{(i)} = 0, \sigma(w^T x^{(i)}) = 0$ → $ln(1) = 0$

$y^{(i)} = 0, \sigma(w^T x^{(i)}) = 0.99$ → $ln(0.01) = -4.61$

Gradient Ascent

We want to maximize $\frac{1}{N}l(w)$

$w^* = \arg\max_w(\frac{1}{N}\Sigma_{i=1}^{N}[y^{(i)}\ln\sigma(w^Tx^{(i)}) + (1-y(i))\ln(1-\sigma(w^Tx^{(i)}))])$

$\sigma(z) = \frac{1}{1+e^{-z}}$

$\frac{d\sigma(z)}{dz} = \sigma(z)(1-\sigma(z))$

$\frac{\partial\sigma(w^Tx)}{\partial w_j} = \frac{\partial\sigma(w_0x_0+...+w_dx_d)}{\partial w_j} = x_j$

$\frac{\partial\sigma(z)}{\partial w_j} = \frac{\partial\sigma(w^Tx)}{\partial w^Tx}\cdot\frac{\partial w^Tx}{\partial w_j} = \sigma(w^Tx)(1-\sigma(w^Tx))x_j$

$\frac{\partial}{\partial w_j}l(w) = \Sigma_{i=1}^{N}(y_i - \sigma(w^Tx^{(i)}))x_j^{(i)}$

If $y_i \approx \sigma(w^Tx^{(i)})$, almost no change!

If $y_i - \sigma(w^Tx^{(i)} \approx \pm 1$, approx $\frac{\alpha}{N}$ times the j^{th} feature!

for i = 1 to num_iter:

$temp0 = w_0 + \frac{\alpha}{N}\frac{\partial l(w)}{\partial w_0} = w_0 + \frac{\alpha}{N}\Sigma_{i=1}^{N}(y_i - \sigma(w^Tx^{(i)}))x_0^{(i)}$ // + since ascent

$temp1 = w_1 + \frac{\alpha}{N}\frac{\partial l(w)}{\partial w_1} == w_0 + \frac{\alpha}{N}\Sigma_{i=1}^{N}(y_i - \sigma(w^Tx^{(i)}))x_1^{(i)}$

...

$tempd = w_d + \frac{\alpha}{N}\frac{\partial l(w)}{\partial w_1} = w_d + \frac{\alpha}{N}\Sigma_{i=1}^{N}(y_i - \sigma(w^Tx^{(i)}))x_d^{(i)}$

$w_0 = temp0$

$w_1 = temp1$

...

$w_d = tempd$

Vector Implementation

for i = 1 to num_iter:

$w = w + \frac{\alpha}{N}X^T(y - \sigma(Xw))$

Evaluating errors: Precision and Recall

Two types of error:

1. False positive — predict positive (has cancer), but false
2. False negative — predict negative, but false

Precision = $\frac{\text{True Positive}}{\text{True Positive + False Positive}}$

Recall = $\frac{\text{True Positive}}{\text{True Positive + False Negative}}$

F1 Score = $2\frac{\text{precision}\cdot\text{recall}}{\text{precision+recall}} \quad \in [0,1]$

Regularization of Logistic Regression

$l_{lasso}(w) = \frac{1}{N}l(w) - \lambda(||w_{1:d}||_1)$ // since ascent

$l_{ridge}(w) = \frac{1}{N}l(w) - \lambda(||w_{1:d}||_2^2)$

Multiple Classes (C_1, ..., C_k)

One-versus-One approach: $\frac{K(K-1)}{2}$ binary classification problems

classify into C_i, C_j; predict the class that wins "majority of votes", confidence scores to resolve ties

One-versus-All approach: K binary classification problems

classify into C_i and $all - C_i$; predict the class that has largest confidence score

Could our algorithm directly estimate the probability of label belonging to each of the classes?

(i.e. don't resort to a binary classification problem)

We will predict K different probabilities: $y^{(i)} = [y_1^{(i)}, ..., y_k^{(i)}]^T$

Logistic regression: $p(y=1|x;w) = \sigma(w^Tx) = \frac{1}{1+e^{-w^Tx}} = \frac{e^{w^Tx}}{e^{w^Tx}+1}$

Soft-max: $p(y=j|x;w) = \frac{e^{w_j^T x}}{\Sigma_{j=1}^{K} e^{w_j^T x}}$