Introduction to Machine Learning

Machine Learning: Jordan Boyd-Graber
University of Maryland
LOGISTIC REGRESSION FROM TEXT

Slides adapted from Emily Fox

Logistic Regression: Regularized Objective

$$\mathscr{L}' \equiv \ln p(Y \mid X, \beta) = \sum_j \ln p(y^{(j)} \mid x^{(j)}, \beta) \tag{1}$$

$$= \sum_j y^{(j)} \left(\beta_0 + \sum_i \beta_i x_i^{(j)} \right) - \ln \left[1 + \exp \left(\beta_0 + \sum_i \beta_i x_i^{(j)} \right) \right] \tag{2}$$

Logistic Regression: Regularized Objective

$$\mathscr{L}' \equiv \ln p(Y \mid X, \beta) = \sum_j \ln p(y^{(j)} \mid x^{(j)}, \beta) \tag{1}$$

$$= \sum_j y^{(j)} \left(\beta_0 + \sum_i \beta_i x_i^{(j)} \right) - \ln \left[1 + \exp \left(\beta_0 + \sum_i \beta_i x_i^{(j)} \right) \right] \tag{2}$$

$$\mathscr{L} = \mathscr{L}' - \mu \sum_i \beta_i^2 \tag{3}$$

New Stochastic Gradient

For document i:

$$\frac{\partial \mathcal{L}_i}{\partial \beta_j} = (y - \pi_i) - 2\mu\beta_j \tag{4}$$

Our gradient from before minus a term that brings feature weights to zero (opposite sign of β_j)

New Stochastic Gradient

For document i:

$$\frac{\partial \mathscr{L}_i}{\partial \beta_j} = (y - \pi_i) - 2\mu\beta_j \tag{4}$$

Our gradient from before minus a term that brings feature weights to zero (opposite sign of β_j)

New Stochastic Gradient

For document i:

$$\frac{\partial \mathscr{L}_i}{\partial \beta_j} = (y - \pi_i) - 2\mu\beta_j \tag{4}$$

Our gradient from before minus a term that brings feature weights to zero (opposite sign of β_j)

Factorization

- Update becomes

$$\beta_j = \beta_j' + \lambda\Big((y - \pi_i)x_j - 2\mu\beta_j'\Big) \tag{5}$$

Factorization

- Update becomes

$$\beta_j = \beta_j' + \lambda\Big((y - \pi_i)x_j - 2\mu\beta_j'\Big) \tag{5}$$

- Can be factorized as

$$\beta_j = \beta_j' + \lambda(y - \pi_i)x_j - 2\lambda\mu\beta_j' \tag{6}$$
$$\beta_j = \beta_j'(1 - 2\lambda\mu) + \lambda(y - \pi_i)x_j \tag{7}$$

Factorization

- Update becomes

$$\beta_j = \beta_j' + \lambda\left((y - \pi_i)x_j - 2\mu\beta_j'\right) \quad (5)$$

- Can be factorized as

$$\beta_j = \beta_j' + \lambda(y - \pi_i)x_j - 2\lambda\mu\beta_j' \quad (6)$$

$$\beta_j = \beta_j'(1 - 2\lambda\mu) + \lambda(y - \pi_i)x_j \quad (7)$$

- Thus, break the update into two steps:
  1. $\beta_j' = \beta_j'' \cdot (1 - 2\lambda\mu)$
  2. $\beta_j = \beta_j' + \lambda(y - \pi_i)x_j$

Revised Algorithm

1. Initialize a vector β to be all zeros
2. Initialize a vector A to be all zeros
3. For $t = 1, \dots, T$
   - For each example $\vec{x}_i, y_i$ and feature j:
     - Simulate regularization updates: $\beta[j] = \beta[j] \cdot (1 - 2\lambda\mu)^{k - A[j] - 1}$
     - Compute $\pi_i \equiv \Pr(y_i = 1 \,|\, \vec{x}_i)$
     - Set $\beta[j] = (\beta[j] + \lambda(y_i - \pi_i)x_i)(1 - 2\lambda\mu)$
     - Keep track of last update for feature $A[j] = k$
4. For each paramter, catch up on missing updates $\beta[j] = \beta[j] \cdot (1 - 2\lambda\mu)^{T - A[j]}$
5. Output the parameters $\beta_1, \dots, \beta_d$.