



Slides adapted from Emily Fox

# Introduction to Machine Learning

Machine Learning: Jordan Boyd-Graber

University of Maryland

LOGISTIC REGRESSION FROM TEXT

## Logistic Regression: Regularized Objective

$$\mathcal{L}' \equiv \ln p(Y|X, \beta) = \sum_j \ln p(y^{(j)} | x^{(j)}, \beta) \quad (1)$$

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

## Logistic Regression: Regularized Objective

$$\mathcal{L}' \equiv \ln p(Y|X, \beta) = \sum_j \ln p(y^{(j)} | x^{(j)}, \beta) \quad (1)$$

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

$$\mathcal{L} = \mathcal{L}' - \mu \sum_i \beta_i^2 \quad (3)$$

## New Stochastic Gradient

For document  $i$ :

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

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

## New Stochastic Gradient

For document  $i$ :

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

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

## New Stochastic Gradient

For document  $i$ :

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

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

## Factorization

- Update becomes

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

## 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

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

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



## 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:

- $\beta'_j = \beta''_j \cdot (1 - 2\lambda \mu)$
- $\beta_j = \beta'_j + \lambda (y - \pi_i) x_j$

## Revised Algorithm

1. Initialize a vector  $\beta$  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  $\pi_i \equiv \Pr(y_i = 1 \mid \vec{x}_i)$
    - Set  $\beta[j] = (\beta[j] + \lambda(y_i - \pi_i)x_i)(1 - 2\lambda\mu)$
    - Keep track of last update for feature  $A[j] = T$
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  $\beta_1, \dots, \beta_d$ .