# Learning Logistic Regressors by Gradient Descent

Machine Learning – CSE446
Carlos Guestrin
University of Washington

April 17, 2013




# Classification

$X \equiv (\text{GPA}, \text{ML grade})$

in reg.: $Y = \text{salary}$ (cont. value)

now: $Y$ is discrete

e.g. $Y \equiv \{\text{hired}, \text{not hired}\}$

- **Learn**: h:**X** $\mapsto$ Y
  - **X** – features
  - Y – target classes
- Conditional probability: P(Y|**X**)
  $P(Y=\text{hired} \mid \text{GPA}=3.6, \text{ML grade}=3.9)$
- Suppose you know P(Y|**X**) exactly, how should you classify?
  - Bayes optimal classifier:
    $\hat{Y} = \arg\max_y P(Y=y \mid X=x)$
- **How do we estimate P(Y|X)?**

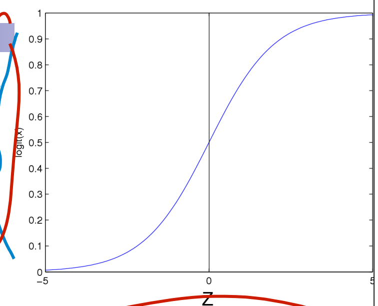
$P(\text{hired} \mid 3.6, 3.9) = 0.8$

$P(\text{not hired} \mid 3.6, 3.9) = 0.2$

$\Rightarrow$ predict hired

# Logistic Regression

**Logistic function (or Sigmoid):** $\frac{1}{1+exp(-z)}$




$\mathbb{R}$

- Learn P(Y|**X**) directly
  - Assume a particular functional form for link function
  - Sigmoid applied to a linear function of the input features:

$$\underset{\text{choice}}{P(Y=0|X,W)} = \frac{1}{1+exp(\underbrace{w_0+\sum_i w_i X_i}_{z\text{: linear just like in reg.}})}$$

$w_0 + \sum_i w_i x_i$ ← not bounded, could be neg.

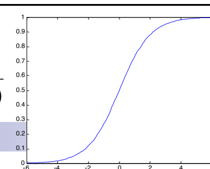
after logistic fcn, output is in [0,1]

**Features can be discrete or continuous!**

# Logistic Regression – a Linear classifier

$\frac{1}{1+exp(-z)}$



$$P(Y=0|X,W) = g(w_0+\sum_i w_i x_i) = \frac{1}{1+e^{w_0+\sum_i w_i x_i}}$$

$w_0 + \sum_i w_i x_i > 0$

$\Rightarrow g(w_0+\sum_i w_i x_i) < 0.5$

$\Rightarrow P(Y=0|X,W) < 0.5$

$\Rightarrow$ predict class = 1

$w_0 + \sum_i w_i x_i = 0$

$w_0 + \sum_i w_i x_i < 0$

$\Rightarrow g(w_0+\sum_i w_i x_i) > 0.5$

$\Rightarrow$ predict class = 0

$\ln \prod = \sum \ln$ ln is monotone

## Loss function: Conditional Likelihood

- Have a bunch of iid data of the form: $(x^j, y^j)_{1:N} = D = (D_x, D_y)$ iid

  x y

  (GPA: 3.2, hired)

  (3.4, not hired)

  (3.8, hired)

- Discriminative (logistic regression) loss function:
  **Conditional Data Likelihood**

$$\arg\max_w p(D_y \mid D_x, w) = \arg\max_x \prod_{j=1}^{N} p(y^j \mid x^j, w)$$

$$= \arg\max_w \ln \prod_{j=1}^{N} p(y^j \mid x^j, w) = \arg\max_x \sum_{j=1}^{N} \ln p(y^j \mid x^j, w)$$

$$\ln P(\mathcal{D}_Y \mid \mathcal{D}_{\mathbf{X}}, \mathbf{w}) = \sum_{j=1}^{N} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w})$$




## Maximizing Conditional Log Likelihood

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

$$P(Y=1|X,W) = \frac{exp(w_0 + \sum_i w_i X_i)}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$l(\mathbf{w}) \equiv \ln \prod_j P(y^j|\mathbf{x}^j, \mathbf{w})$$

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

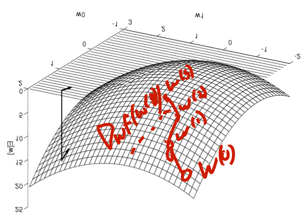
Good news: $l$(**w**) is concave function of **w**, no local optima problems

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. Find optimum with gradient ascent




**Gradient:** $\nabla_{\mathbf{w}} l(\mathbf{w}) = [\frac{\partial l(\mathbf{w})}{\partial w_0}, \ldots, \frac{\partial l(\mathbf{w})}{\partial w_n}]'$

Step size, $\eta > 0$

**Update rule:** $\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w}^{(t)})}{\partial w_i}$$

for HW2 $\eta$ will be constant

- Gradient ascent is simplest of optimization approaches
  - e.g., Conjugate gradient ascent can be much better

Often, especially in proofs, $\eta$ gets smaller with iterations

e.g. $\eta_t = \frac{\alpha}{t}$  $\alpha \leftarrow$ constant



# Maximize Conditional Log Likelihood: Gradient ascent

$\frac{\partial}{\partial s} \ln f(s) = \frac{f'(s)}{f(s)}$  $\frac{\partial}{\partial s} e^{f(s)} = f'(s)\, e^{f(s)}$

$$l(\mathbf{w}) = \sum_{j=1}^{N} y^j (w_0 + \sum_{i=1}^{k} w_i x_i^j) - \ln(1 + exp(w_0 + \sum_{i=1}^{k} w_i x_i^j))$$

$$\nabla_w l: \quad \frac{\partial l}{\partial w_i} = \sum_{j=1}^{N} \left[ y^j x_i^j - \underbrace{\frac{x_i^j \exp(w_0 + \sum w_i x_i^j)}{1 + \exp(w_0 + \sum_i w_i x_i^j)}}_{\hat{P}(Y=1 \mid x^j, w)} \right]$$

$$\frac{\partial l}{\partial w_i} = \sum_{j=1}^{N} x_i^j \underbrace{\left(y^j - \hat{P}(Y=1 \mid x^j, w)\right)}$$

weighted by contribution of i th feature to point j

how far is my prediction from the truth

# Gradient Ascent for LR

revisit soon

Start from some $w^{(0)}$ e.g. $\emptyset$

Gradient ascent algorithm: iterate until change < ε

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

step size

For i=1,…,k,

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

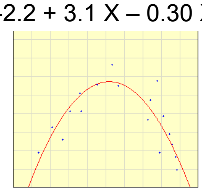
repeat



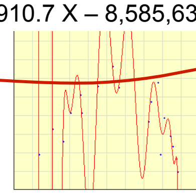

# Regularization in linear regression

- 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 $X^2$ + …



- Regularized least-squares (a.k.a. ridge regression), for λ>0:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left( t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

# Linear Separability

$\exists w$ : all positive examples $w_0 + \sum_i w_i x_i > 0$

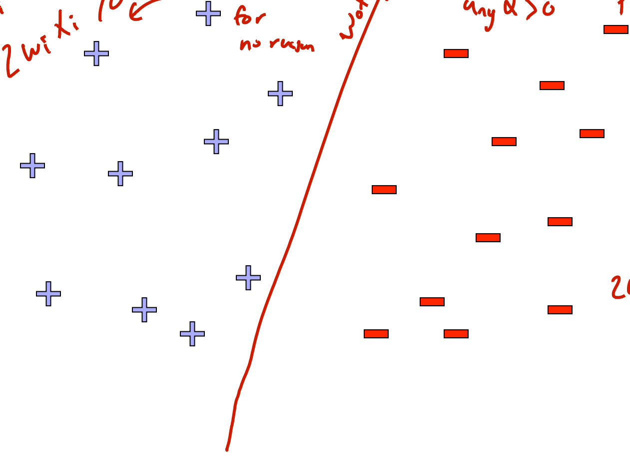
all negative : $w_0 + \sum_i w_i x_i < 0$

$\Rightarrow \alpha w$ is also a separating plane, any $\alpha > 0$

$w_0 + \sum_i w_i x_i > 0$ ← same separating plane

$2w_0 + \sum_i 2w_i x_i > 0$ ← "more sure" for no reason

$w_0 + \sum_i w_i x_i = 0$

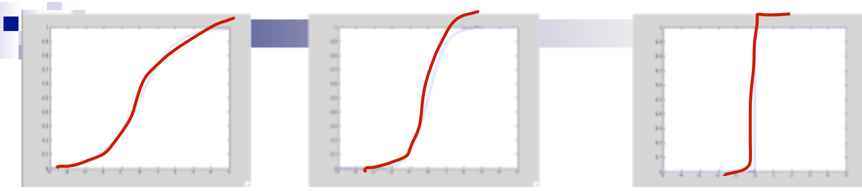


$w_0 + \sum_i w_i x_i < 0$

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




# Large parameters → Overfitting



$$\frac{1}{1+e^{-x}} \qquad \frac{1}{1+e^{-2x}} \qquad \frac{1}{1+e^{-100x}}$$

- If data is linearly separable, weights go to infinity

$P(Y=0 \mid w, x) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}}$ ← increase as $\|w\| \to \infty$

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

Regularization → $\|w\|_2^2$, $\|w\|_1$, …

# Regularized Conditional Log Likelihood

- Add regularization penalty, e.g., $L_2$:

$$\max_w \quad \ell(\mathbf{w}) = \ln \prod_{j=1}^{N} P(y^j|\mathbf{x}^j, \mathbf{w}) - \frac{\lambda}{2} \underbrace{||\mathbf{w}||_2^2}_{\sum_{i=1}^{k} w_i^2}$$

- Practical note about $w_0$:

don't regularize

- Gradient of regularized likelihood: $\overbrace{\phantom{xxxx}}^{\lambda w_i}$

$$\frac{\partial \ell}{\partial w_i} = \frac{\partial}{\partial w_i}\left[ \ln \prod P(y^j|x^j, w) \right] - \frac{\lambda}{2} \frac{\partial}{\partial w_i} ||w||_2^2$$



# Standard v. Regularized Updates

- Maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \quad \ln \prod_{j=1}^{N} P(y^j|\mathbf{x}^j, \mathbf{w})$$

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

without regularization

- Regularized maximum conditional likelihood estimate

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

push towards $\phi$

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

$w^*$ is optimal solution to learning problem

## Please Stop!! Stopping criterion

$$\ell(\mathbf{w}) = \ln \prod_j P(y^j|\mathbf{x}^j, \mathbf{w})) - \lambda||\mathbf{w}||_2^2$$

- When do we stop doing gradient ascent or descent? $\varepsilon > 0$ user-specified tolerance

$$\ell(w^*) - \ell(w^{(t)}) < \varepsilon$$

- Because $l(\mathbf{w})$ is strongly concave:
  - i.e., because of some technical condition

$$\underbrace{\ell(\mathbf{w}^*) - \ell(\mathbf{w})}_{\text{I don't know}} \leq \frac{1}{2\lambda}||\nabla \ell(\mathbf{w})||_2^2 < \varepsilon$$

- Thus, stop when: $\frac{1}{2\lambda}||\nabla \ell(w^{(t)})||_2^2 < \varepsilon$





## Digression: Logistic regression for more than 2 classes

- Logistic regression in more general case (C classes), where *Y in* {1,…,C}

for C classes $(C-1)(k+1)$ params

$\forall$ class $c \in \{1, \ldots, C-1\}$

$$P(Y=c \mid X, w) \propto e^{w_{c0} + \sum_{i=1}^{k} w_{ci} X_i}$$

$$P(Y=C \mid X, w) = 1 - \sum_{c'=1}^{C-1} P(Y=c' \mid X, w)$$

$C=2$

# params to learn: $K+1$

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

normalizer

Probs add up to 1

$$P(Y=0|X,w) = 1 - P(Y=1|X,w)$$

$$= \frac{1}{1 + e^{w_0 + \sum_i w_i X_i}}$$

# Digression: Logistic regression more generally

$$W = \begin{array}{c|c} & 0 \; \cdots \; k \\ \hline 1 & \\ \vdots & \boxed{\qquad\qquad} \; w_c \\ C-1 & \end{array}$$

- Logistic regression in more general case, where *Y in* {1,…,C}

for *c<C*

$$P(Y=c|\mathbf{x},\mathbf{w}) = \frac{\exp(w_{c0}+\sum_{i=1}^{k} w_{ci}x_i)}{\underbrace{1+\sum_{c'=1}^{C-1}\exp(w_{c'0}+\sum_{i=1}^{k} w_{c'i}x_i)}_{\text{normalize}}}$$

for *c=C* (normalization, so no weights for this class)

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

**Learning procedure is basically the same as what we derived!** Slightly longer derivative