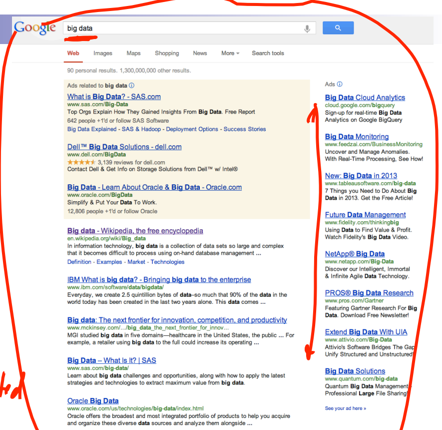
Case Study 1: Estimating Click Probabilities

Intro
Logistic Regression
Gradient Descent

Machine Learning/Statistics for Big Data
CSE599C1/STAT592, University of Washington
Carlos Guestrin
January 8th, 2013

Ad Placement Strategies

- Companies bid on ad prices
  - $C_1 \to \$10$
  - $C_2 \to \$20$
  - $C_3 \to \$100$
- Which ad wins? (many simplifications here)
  - Naively: $\$100 \to C_3$
  - But: paid when click
  - Instead: Care about expected value:



fundamental problem

$P(\text{click} \mid C_2) = 0.1 \to E[C_2] = 0.1 \times \$20 = \$2$

$P(\text{click} \mid C_3) = 0.001 \to E[C_3] = 0.001 \times \$100 = \$0.1$

Key Task: Estimating Click Probabilities

- What is the probability that user *i* will click on ad *j*
- Not important just for ads:
  - Optimize search results
  - Suggest news articles
  - Recommend products
- Methods much more general, useful for:
  - Classification
  - Regression
  - Density estimation

Learning Problem for Click Prediction

- Prediction task: $X \to [0,1]$ $P(\text{Click}=1 \mid X)$
- Features:

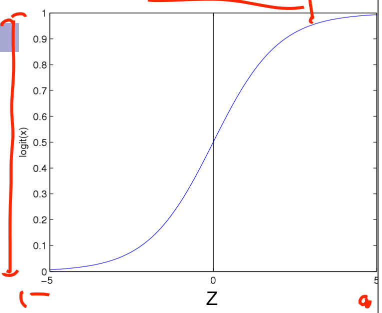
$X = (\text{feats of page, feats ad, feats user})'$

$\{0,1\}$

- Data: (x^i, y^i)
  - Batch: a fixed data set $(x^1,y^1), (x^2,y^2), \ldots (x^N,y^N)$
  - Online: $(\text{user, page, ad})_i \to$ are shown $\to$ get feedback y_i; update model on the fly..
- Many approaches (e.g., logistic regression, SVMs, naïve Bayes, decision trees, boosting,…)
  - Focus on logistic regression; captures main concepts, ideas generalize to other approaches

Logistic Regression

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



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

interval [0,1]

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

linear function of the features

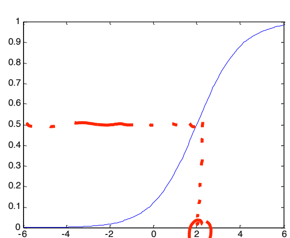
$w_0 + \sum_i w_i X_i$ not bounded could be negative

Features can be discrete or continuous!

Understanding the sigmoid

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

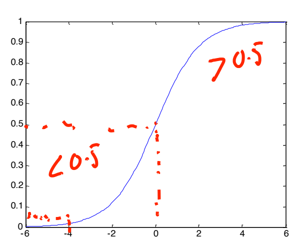
w_0=-2, w_1=-1



break even point $-w_0$

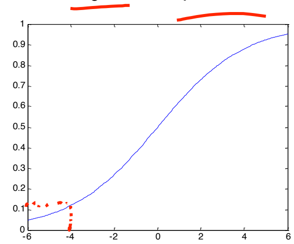
only one feature

w_0=0, w_1=-1



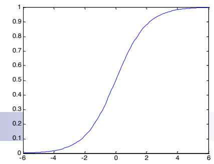
$x_1 = 0$

w_0=0, w_1=-0.5



Logistic Regression – a Linear classifier

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



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

in d dims

$w_0 + \sum_i w_i x_i > 0$

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

class: 1

$w_0 + \sum_i w_i x_i$

$w_0 + \sum_i w_i x_i < 0$

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

class = 0

Very convenient!

$P(Y=1|X) = 1 - P(Y=0|X) \geq 0$

estimate these probabilities from data

$$P(Y=0\,|X=<X_1,...X_n>) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$P(Y=1\,|X=<X_1,...X_n>) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

Y = 1 if

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

linear classification rule!

implies

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

Digression: Logistic regression for more than 2 classes

- Logistic regression in more general case (k+1 classes), where *Y* in { $y_1,\ldots,y_R$}

$\forall j \in 1 \ldots R\text{-}1$ $$P(Y=y_j|X) \propto e^{w_{0j} + \sum_i w_{ij} X_i}$$

$$P(Y=y_R|X) = 1 - \sum_{j=1}^{R-1} P(Y=y_j|X)$$

 9

Digression: Logistic regression more generally

- Logistic regression in more general case, where *Y* in {$y_1,\ldots,y_R$}

for *k<R*

$$P(Y=y_k|X) = \frac{\exp(w_{k0} + \sum_{i=1}^n w_{ki}X_i)}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^n w_{ji}X_i)}$$

for *k=R* (normalization, so no weights for this class)

$$P(Y=y_R|X) = \frac{1}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^n w_{ji}X_i)}$$

Features can be discrete or continuous!

 10

$\ln \to$ monotonic $\quad \arg\max_w f(w) = \arg\max_w \ln f(w)$

Loss function: Conditional Likelihood

$\ln \prod = \sum \ln$

- Have a bunch of iid data of the form: $(x^i, y^i)_{1:N} \equiv D \equiv (D_X, D_Y)$

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

$$\arg\max_w P(D_Y \mid D_X^w) = \arg\max_w \prod_{i=1}^{N} P(y^i \mid x^i_w) = \arg\max_w \ln \prod_{i=1}^{N} P(y^i \mid x^i_w)$$

$$= \arg\max_w \sum_{i=1}^{N} \ln P(y^i \mid x^i_w)$$

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

$\ln \frac{a}{b} = \ln a - \ln b, \quad \ln 1 = 0, \quad \ln e^{\text{something}} = \text{something}$

Expressing Conditional Log Likelihood

conditional log likelihood

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

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

$$l(\mathbf{w}) \equiv \sum_j \ln P(y^j|\mathbf{x}^j,\mathbf{w}) \quad y^j \in \{0,1\}$$

$$= \sum_j \begin{cases} P(Y=1 \mid x^j, w) & \text{if } y^j = 1 \\ P(Y=0 \mid x^j, w) & \text{if } y^j = 0 \end{cases}$$

$$\ell(\mathbf{w}) = \sum_j y^j \ln P(Y=1|\mathbf{x}^j,\mathbf{w}) + (1-y^j)\ln P(Y=0|\mathbf{x}^j,\mathbf{w})$$

$$= \sum_j (1-y^j) \ln \frac{1}{1+e^{w_0+\sum_i w_i x_i^j}} + y^j \ln \frac{e^{w_0+\sum_i w_i x_i^j}}{1+e^{w_0+\sum_i w_i x_i^j}}$$

$$= \sum_j (y^j - 1)\ln\left(1+e^{w_0+\sum_i w_i x_i^j}\right) + y^j\left[w_0+\sum_i w_i x_i^j - \ln\left(1+e^{w_0+\sum_i w_i x_i^j}\right)\right]$$

$$= \sum_j y^j\left(w_0+\sum_i w_i x_i^j\right) - \ln\left(1+e^{w_0+\sum_i w_i x_i^j}\right)$$

likelihood of data

Maximizing Conditional Log Likelihood

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

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

max w

$$\begin{aligned} l(\mathbf{w}) &\equiv \ln \prod_j P(y^j|\mathbf{x}^j, \mathbf{w}) \\ &= \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j)) \end{aligned}$$

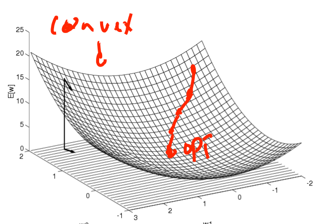
Good news: $l(\mathbf{w})$ is concave function of **w**, no local optima problems

Bad news: no closed-form solution to maximize $l(\mathbf{w})$

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

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

- Gradient ascent is simplest of optimization approaches
  - e.g., Conjugate gradient ascent much better (see reading)

Maximize Conditional Log Likelihood: Gradient ascent

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

$$l(\mathbf{w}) = \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j))$$

$$\frac{\partial l(w)}{\partial w_i} = \sum_j y^j x_i^j - \frac{x_i^j\, e^{w_0 + \sum_i w_i x_i^j}}{1 + e^{w_0 + \sum_i w_i x_i^j}}$$

$P(Y=1 \mid x^j, w)$

$$\frac{\partial l(w)}{\partial w_i} = \sum_j x_i^j \left[y^j - P(Y=1 \mid x^j, w) \right]$$

how much x_i plays a role in j

how much of a mistake I am making in point j

Gradient Ascent for LR

we will revisit this later

Gradient ascent algorithm: iterate until change < ε

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

For i=1,…,~~n~~ d,

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

repeat

Test set error as a function of model complexity