CMSC 726 Lecture 6:Linear Classification

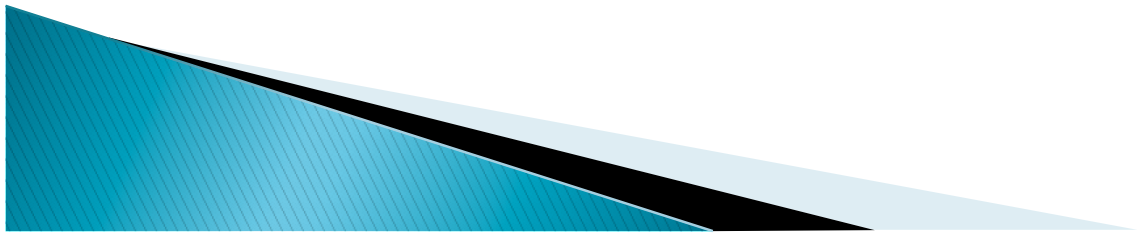
Lise Getoor

September 21, 2010

ACKNOWLEDGEMENTS: The material in this course is a synthesis of materials from many sources, including: Hal Daume III, Mark Drezde, Carlos Guestrin, Andrew Ng, Ben Taskar, Eric Xing, and others. I am very grateful for their generous sharing of insights and materials.

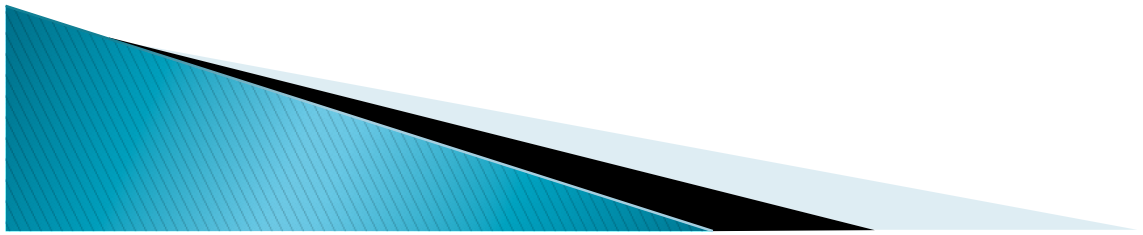
Today's Topics

- Logistic Regression
- Gradient Ascent for Logistic Regression
- Newton's Method



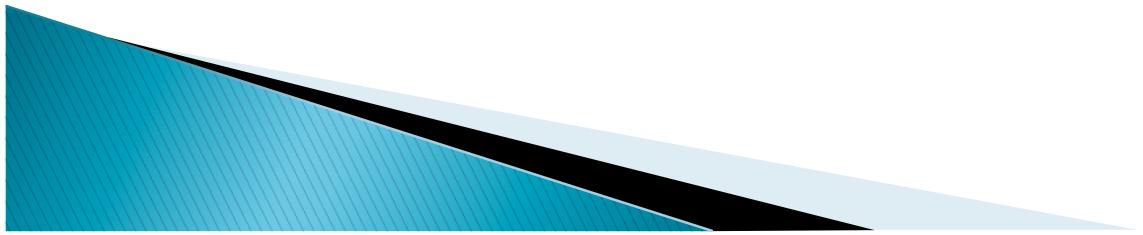
Classification

- Data $\{(x_i, y_i)\}_{i=1}^{N} \quad x_i \in \Re^M \quad y_i \in L$
- Learn: a mapping from x to discrete value y
  - h(x) = y
- Examples
  - Spam classification
  - Document topic classification
  - Identifying faces in images



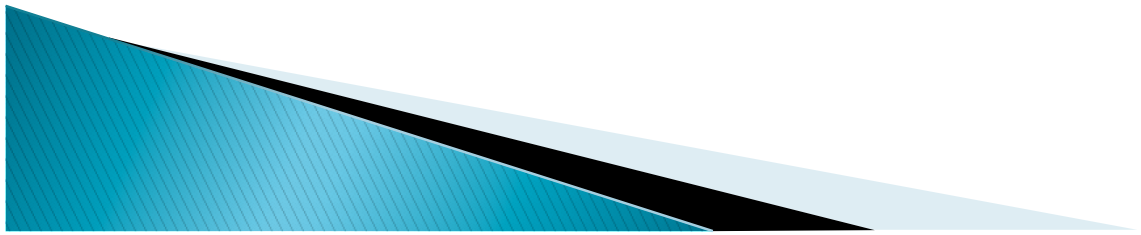
Binary Classification

- We'll focus on binary classification
  - $y_i \in \{0,1\}$
- Usually easy to generalize to multi-class classification

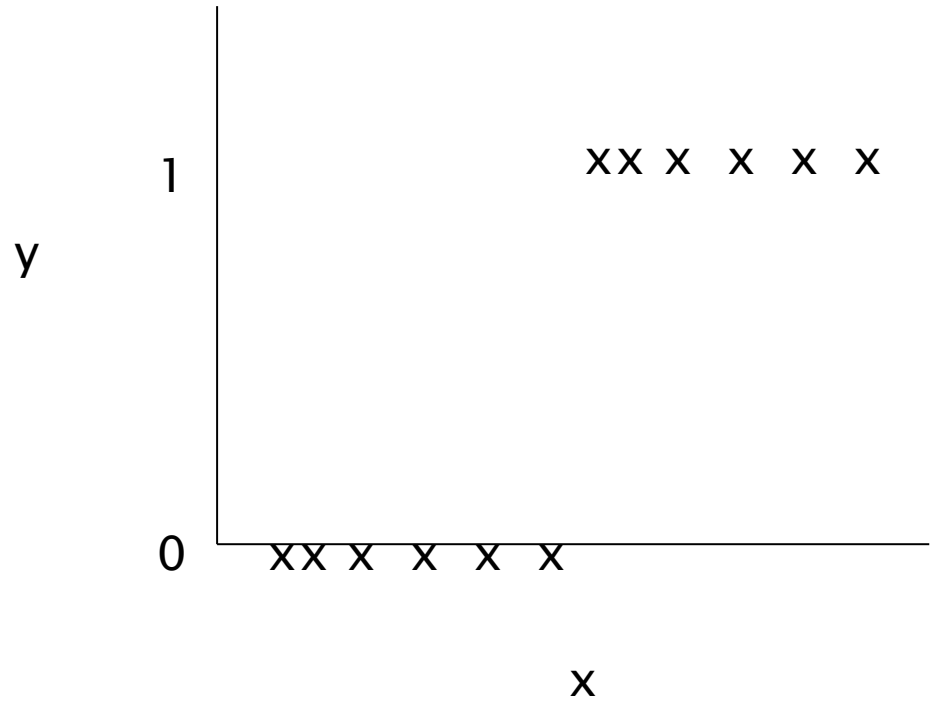


Regression

- Least squares regression
  - Outputs real number for each example
- It seems that classification should be easier!
- Let's use regression for classification
  - Learn least squares regression model $h_w(x)=y$
    - $h_w(x)=w^Tx$
  - If $h_w(x) > c$, predict "True (1)"
  - If $h_w(x) \leq c$ predict "False (0)"



Example

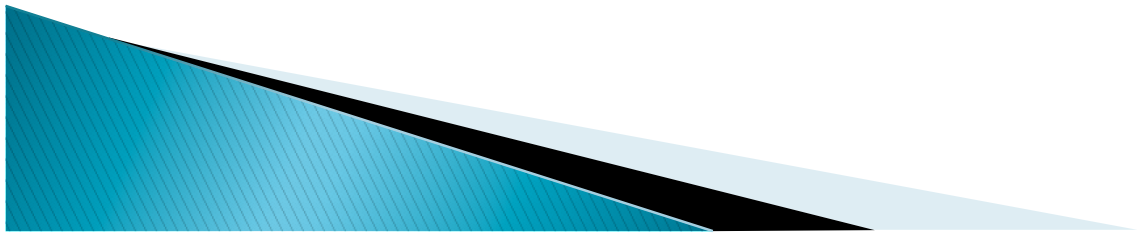


Evaluation

- Accuracy

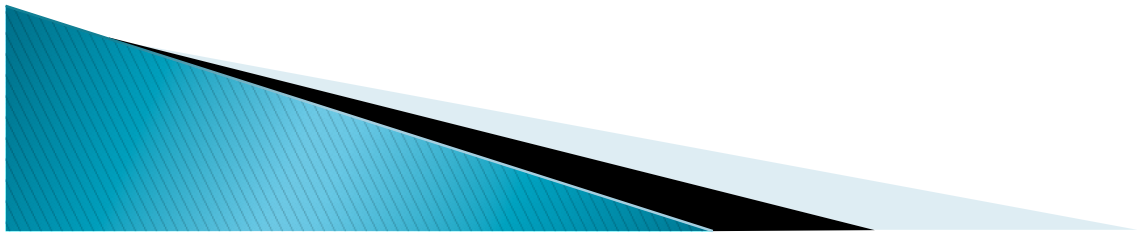
$$\frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

- Other measurements appropriate for some tasks
  - Ex. we care more about certain types of mistakes



Regression for Classification

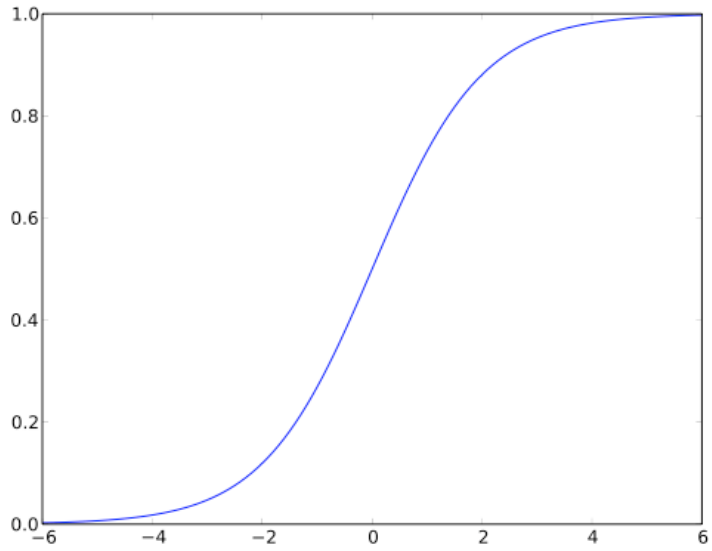
- Mismatch between regression loss and classification
  - Classification: accuracy
  - We don't care about large vs. small values of output
  - Outliers problematic
- We need output to be either 1 or 0
  - Alternative: output *between* 1 and 0



Logistic Function

- We have a function that returns unbounded values
- We want a function with output between 0 and 1
- Logistic function
  - Scaling parameter α
  - Most outputs are close to 1 or 0

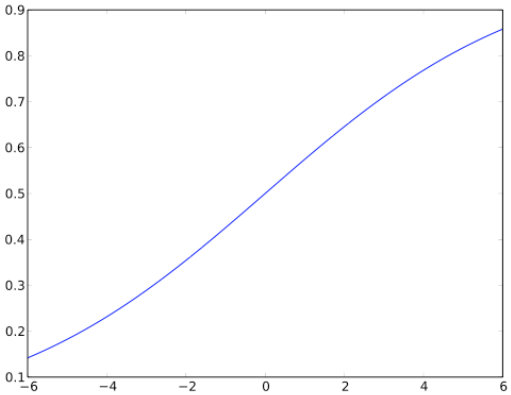
$$g_\alpha(x) = \frac{1}{1 + e^{-\alpha x}}$$



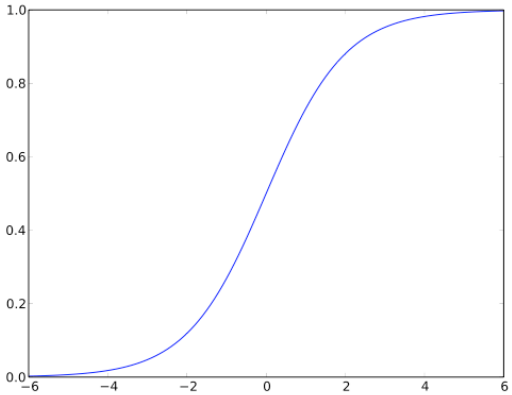
$\alpha = 1$

Logistic Function

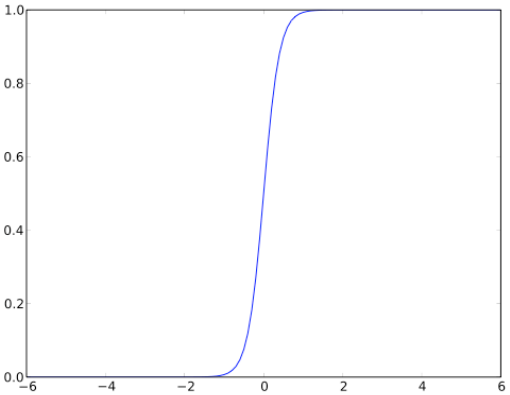
$$g_\alpha(x) = \frac{1}{1 + e^{-\alpha x}}$$



$\alpha = 0.3$



$\alpha = 1$



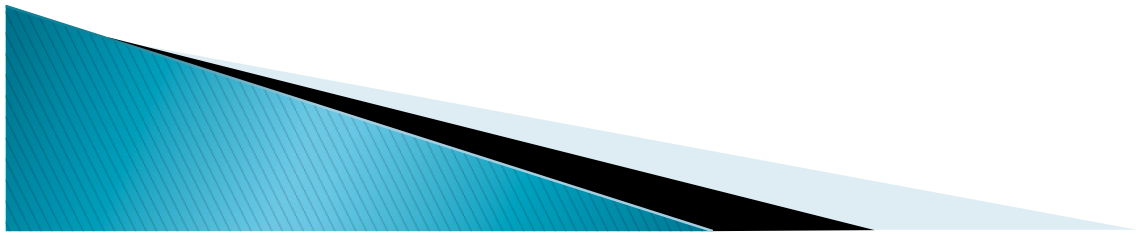
$\alpha = 5$

Logistic Regression

- We can combine the logistic function and our regression model

$$h_w(x_i) = g_w(x_i) = \frac{1}{1 + e^{-w^T x_i}}$$

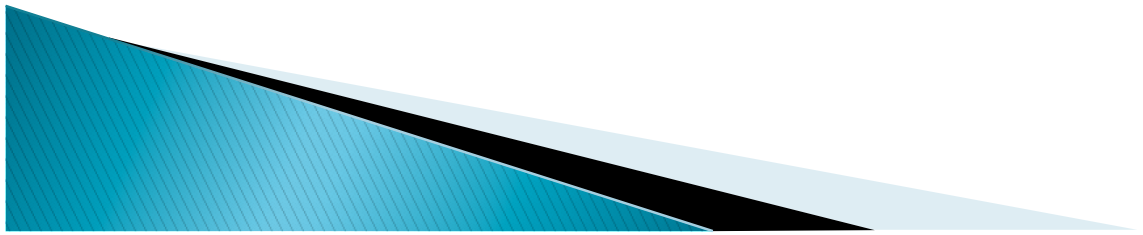
- Notice: as $w^T x_i$ becomes:
  - Large- output closer to 1
  - Small- output closer to 0



Probabilistic Interpretation

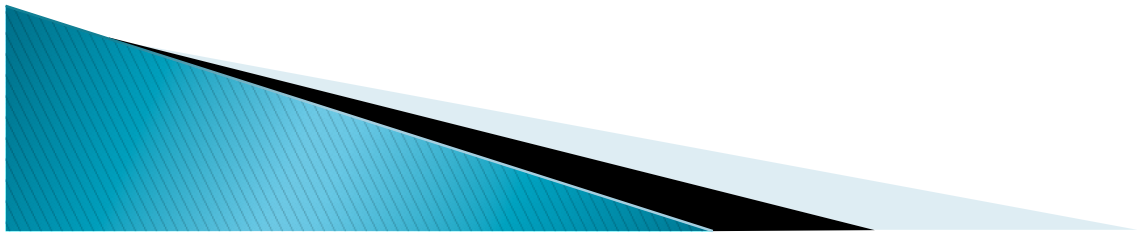
$$h_w(x) = P(y = 1 \mid x; w) = \frac{1}{1 + e^{-w^T x}}$$

$$classif\ icat\!ion = \arg\max_{y=0,1} p(y \mid x; w)$$



Discriminative Model

- We just care about p(y|x) so we can *discriminate* between classes
  - More on this next time

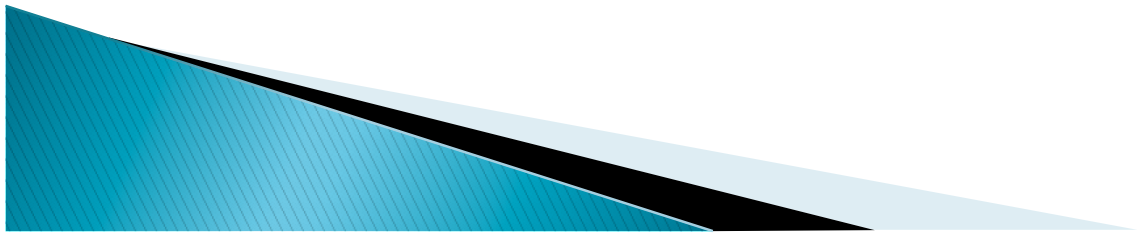


Logistic Regression Decisions

- Given parameters w, how do we make predictions?

$$p_w(y = 1 \mid x) = \frac{1}{1 + e^{-w^T \cdot x}}$$

- If output > 0.5, predict 1, else predict 0
- In addition to prediction, we have confidence in prediction
  - Confidence is the probability of the prediction



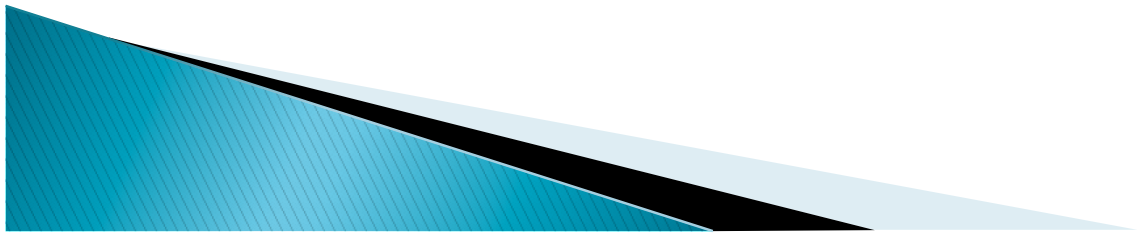
Compact Form

$$P(y = 1 \mid x; w) = h_w(x)$$

$$P(y = 0 \mid x; w) = 1 - h_w(x)$$

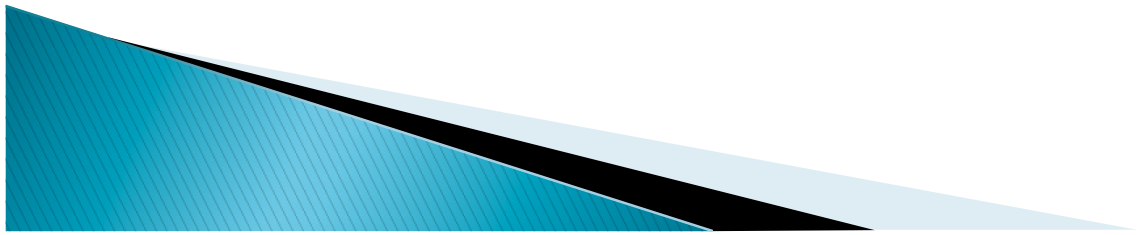
Then we can write

$$P(y \mid x; w) = h_w(x)^y (1 - h_w(x))^{1-y}$$



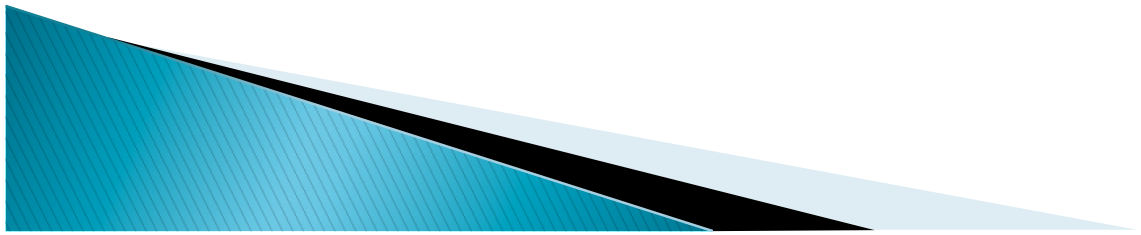
Learning the Function

- Where does w come from?
- What do we have?
  - Model
    - Probability distribution for p(y|x) parameterized by w
  - Data
    - Many pairs of example x and label y
  - Tools
    - Maximum likelihood estimation



Learning the Function

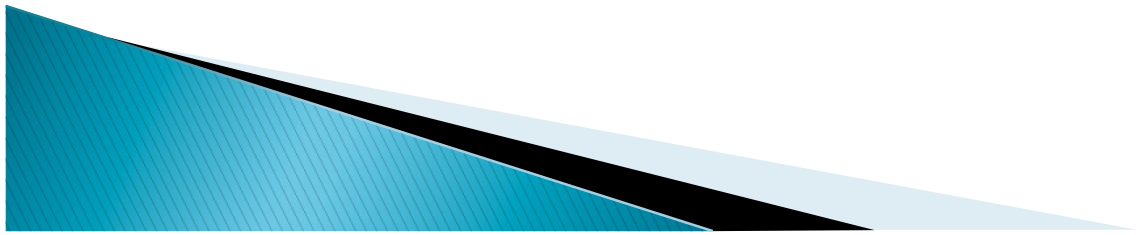
- Strategy
  - Write the likelihood of the data
  - Take the log to obtain the log likelihood
  - Use maximum likelihood to find optimal setting for w



Likelihood

- Data likelihood
  - L(w) = p(Y|X;w)

$$p(Y \mid X, w) = \prod_{i=1}^{N} p(y_i \mid x_i, w) = \prod_{i=1}^{N} h_w(x_i)^{y_i} (1 - h_w(x_i))^{1-y_i}$$



Log-Likelihood

- Data likelihood
  - L(w) = p(Y|X;w)

$$l(w) = \log L(w)$$

$$= \sum_{i=1}^{N} y_i \log h_w(x_i) + (1 - y_i) \log(1 - h_w(x_i))$$

Good news: *l(ω)* is concave function of ω

Bad news: no closed-form solution to maximize *l(ω)*

Log-Likelihood cont.

$$l(w) = \sum_{i=1}^{N} y_i \log h_w(x_i) + (1 - y_i) \log(1 - h_w(x_i))$$

$$h_w(x) = \frac{1}{1 + e^{-w^T \cdot x}}$$

$$l(w) = \sum_{i=1}^{N} (y_i - 1) w^T x_i + \log(1 + e^{-w^T x_i})$$

Good news: $l(\omega)$ is concave function of ω

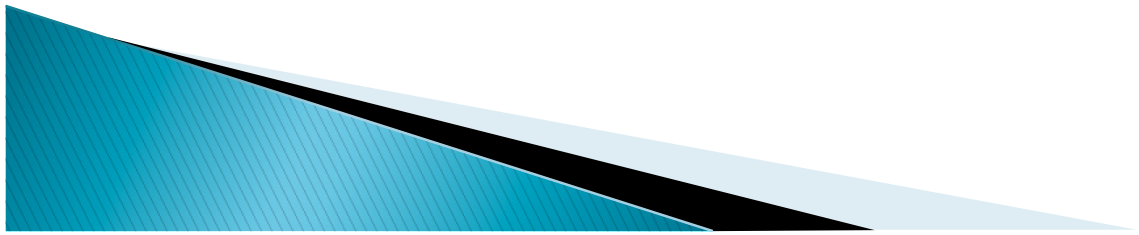
Bad news: no closed-form solution to maximize $l(\omega)$

Gradient Ascent

$$w = w + \alpha \nabla_w l(w)$$

$$\frac{\partial}{\partial w_j} l(w) = \sum_{i=1}^{N} (y_i - h_w(x_i)) x_i^j$$

$$w_j = w_j + \alpha \sum_{i=1}^{N} (y_i - h_w(x_i)) x_i^j$$

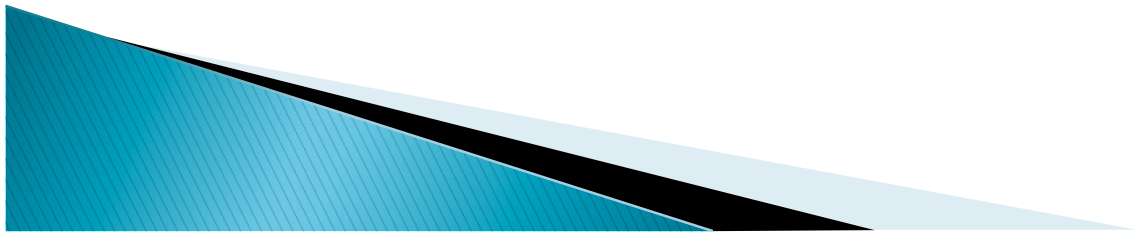


Algorithm: Logistic Regression

- Train: given data X and Y
  - Initialize w to starting value
  - Repeat until convergence
    - Compute the value of the derivative for X,Y and w
    - Update w by taking a gradient step
- Predict: given an example x
  - Using the learned w, compute p(y|x,w)

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

- Note: many other optimization routines available



The Newton's method

- Finding a zero of a function

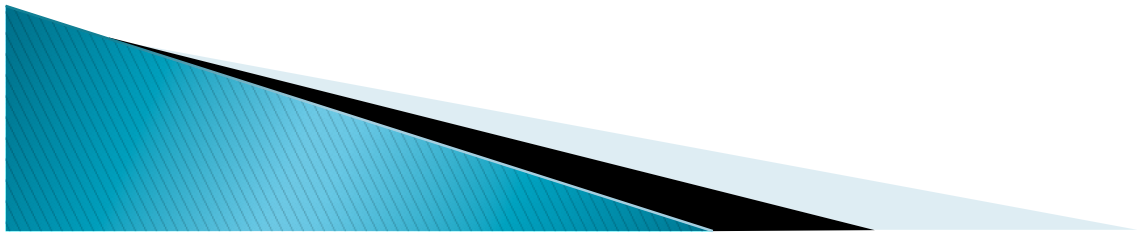
$$w^{t+1} = w^t - \frac{f(w^t)}{f'(w^t)}$$

The Newton's method (con'd)

- To maximize the conditional likelihood $l(\omega)$, since l is convex, we need to find ω where $l'(w)=0$!

- So we can perform the following iteration:

$$w^{t+1} = w^t - \frac{f'(w^t)}{f''(w^t)}$$



The Newton-Raphson method

- In LR the θ is vector-valued, thus we need the following generalization:

$$w^{t+1} = w^t + H^{-1}\nabla_{w^t} l(w^t)$$

- ∇ is the gradient operator over the function

- H is known as the Hessian of the function

$$H_{ij} = \frac{\partial^2 l(w^t)}{\partial w_i \partial w_j}$$

Iterative reweighed least squares (IRLS)

- Recall in the least square est. in linear regression, we have:

$$\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- Now for logistic regression:

$$\theta^{t+1} = (\mathbf{X}^T\mathbf{R}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}\mathbf{z}$$

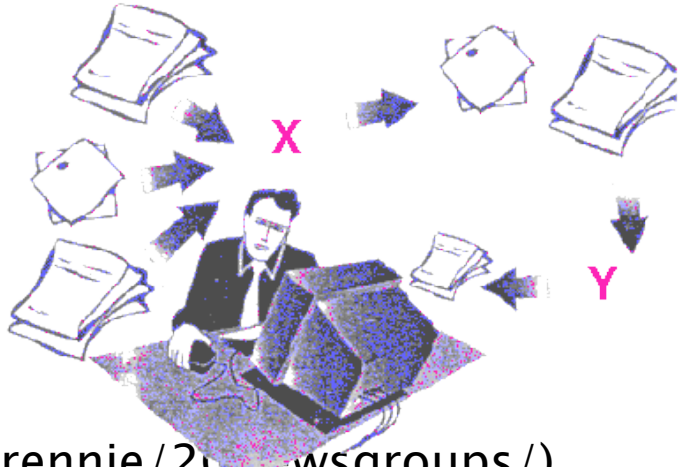
$$\text{where} \quad \mathbf{z} = \mathbf{X}\theta^t - \mathbf{R}^{-1}(\mathbf{u} - \mathbf{y})$$

$$\text{and} \quad R_{ii} = u_i(1 - u_i)$$

Logistic regression: practical issues

- NR (IRLS) is slower than gradient, but takes fewer steps
- Quasi-Newton methods, that approximate the Hessian, work faster.
- Conjugate gradient takes $O(Nm)$ per iteration, and usually works best in practice.
- Stochastic gradient descent can also be used if N is large

Case study

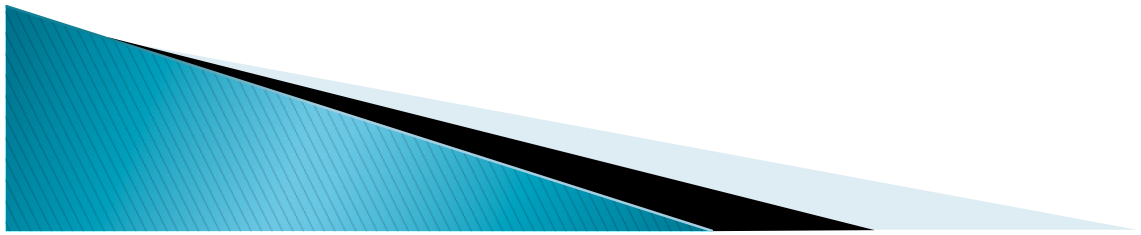


- Dataset
  - 20 News Groups (20 classes)
  - Download :(http://people.csail.mit.edu/jrennie/20Newsgroups/)
  - 61,118 words, 18,774 documents
  - Class labels descriptions

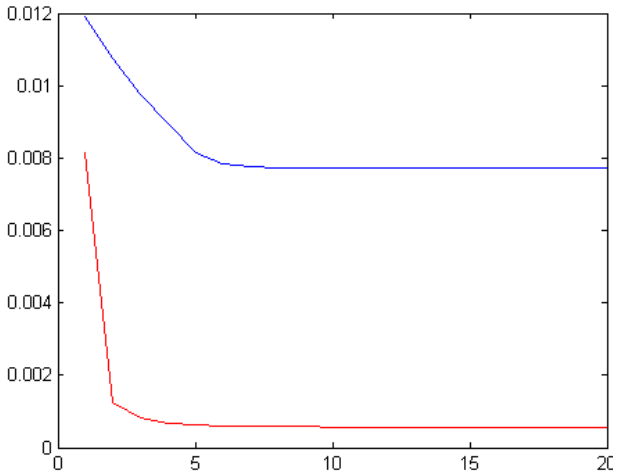
comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

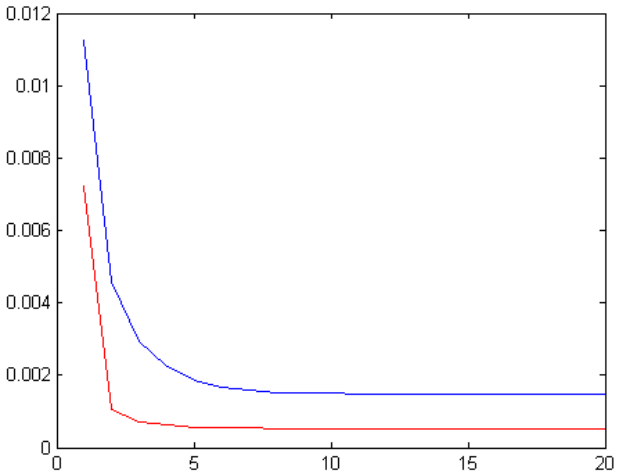
Experimental setup

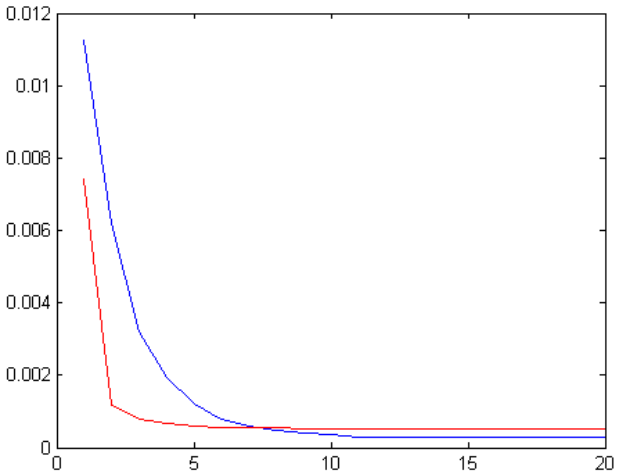
- Parameters
  - Binary features
  - Using (stochastic) Gradient descent vs. IRLS to estimate parameters
- Training/Test Sets:
  - Subset of 20ng (binary classes)
  - Use SVD to do dimension reduction (to 100)
  - Random select 50% for training
  - 10 run and report average result
- Convergence Criteria: RMSE in training set



Convergence curves







alt.atheism
vs.
comp.graphics

rec.autos
vs.
rec.sport.baseball

comp.windows.x
vs.
rec.motorcycles

Legend: - X-axis: Iteration #; Y-axis: error
- In each figure, red for IRLS and blue for gradient descent