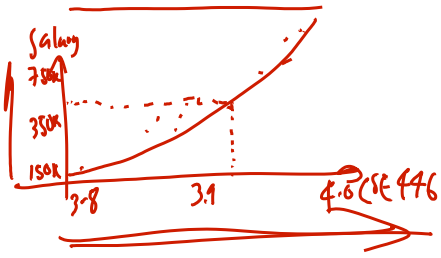
Classification
Logistic Regression

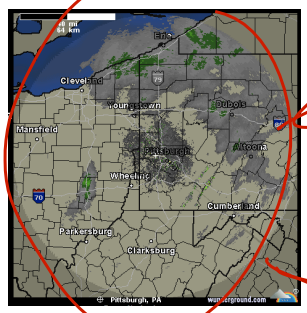
Machine Learning – CSE546
Carlos Guestrin
University of Washington

October 9, 2013



THUS FAR, REGRESSION: PREDICT A CONTINUOUS VALUE GIVEN SOME INPUTS

Weather prediction revisted



Temperature

63°F

Reading Your Brain, Simple Example

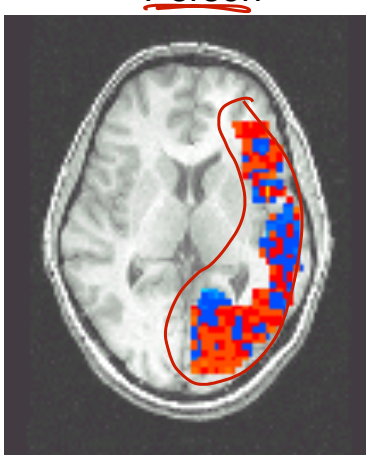
[Mitchell et al.]

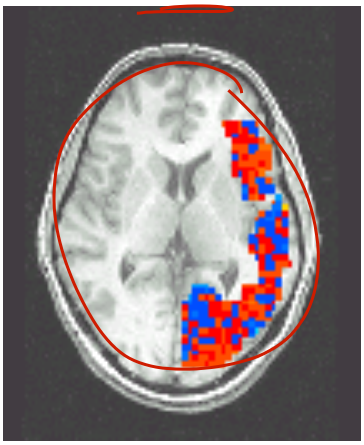
Pairwise classification accuracy: 85%

C = {Person, Animal}

Person

Animal





Classification

X = (GPA, ML grade)

in reg: Y = salary

in class: Y = (hired, not hired)

- **Learn**: h:**X** ↦ Y
  - **X** – features
  - Y – target classes
- Conditional probability: P(Y|**X**)

  P(Y = hired | X = (GPA = 3.6, ML grade = 3.9))
- Suppose you know P(Y|**X**) exactly, how should you classify?
  - Bayes optimal classifier:

    $\hat{y} = \arg\max_y P(Y = y \mid X = x)$

    P(hired | 3.6, 3.9) = 0.8

    P(not hired | 3.6, 3.9) = 0.2

    ⇒ $\hat{y}$ = hired !!
- **How do we estimate P(Y|X)?**

Link Functions

- Estimating P(Y|**X**): Why not use standard linear regression?

  $P(Y|X) = w_0 + \sum_i w_i X_i$

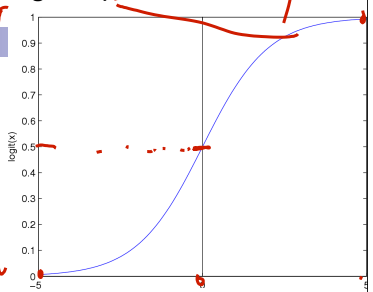
  P(Y|X) ∈ [0,1]

  $w_0 + \sum_i w_i X_i \in (-\infty, \infty)$
- Combing regression and probability?
  - Need a mapping from real values to [0,1]
  - A link function! $g: \mathbb{R} \to [0,1]$

  Many options, simple choice today

Logistic Regression

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



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

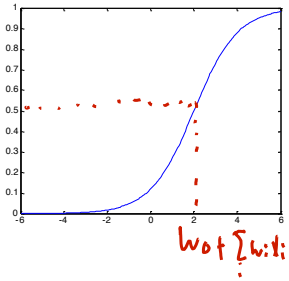
$$P(Y=0|X,W) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

Features can be discrete or continuous!

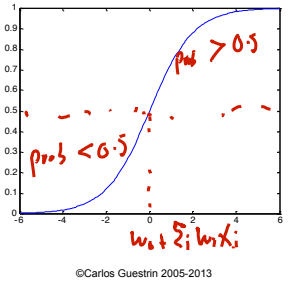
Understanding the sigmoid

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

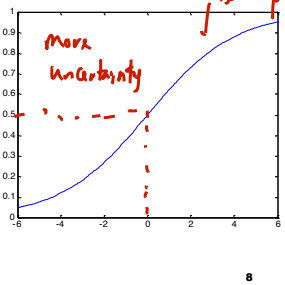
w_0=-2, w_1=-1



w_0=0, w_1=-1

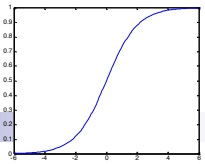


w_0=0, w_1=-0.5



Logistic Regression – a Linear classifier

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



linear fn separates positives from negatives

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

linear function

$w_0+\sum_i w_i x_i > 0$

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

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

$\Rightarrow$ predict 1

$w_0+\sum_i w_i x_i = 0$

$w_0+\sum_i w_i x_i < 0$

$\Rightarrow P(Y=0|X,w) > 0.5$

$\Rightarrow$ predict 0

Very convenient!

$f > 1 \Rightarrow \ln f > 0$

called a log linear model

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

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

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

implies

choose $Y=1$

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

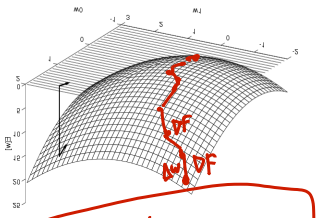
linear classification rule!

implies

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

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 can be much better

Loss function: Conditional Likelihood

- Have a bunch of iid data of the form:

$$\{(\mathbf{x}^j, y^j)\}_{j=1}^N = \mathcal{D} = (\mathcal{D}_X, \mathcal{D}_Y)$$

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

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

Expressing Conditional Log Likelihood

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

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

$$l(\mathbf{w}) \equiv \sum_{j=1}^{N} \ln P(y^j|\mathbf{x}^j,\mathbf{w})$$

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

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

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

want max this wrt w

Maximizing Conditional Log Likelihood

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

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

want max w

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

$$= \sum_j y^j(w_0+\sum_{i=1}^{n} w_i x_i^j) - \ln(1+exp(w_0+\sum_{i=1}^{n} w_i x_i^j))$$

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

Maximize Conditional Log Likelihood: Gradient ascent

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

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

weigh by how much feature i participates in data point x^j

does my prediction match

Gradient Ascent for LR

Gradient ascent algorithm: iterate until change < ε

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

For i=1,…,k,

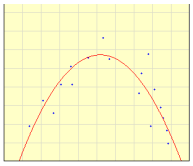
$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_{j=1}^{N} x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

repeat

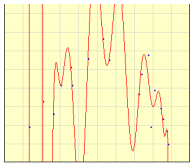
Regularization in linear regression

- Overfitting usually leads to very large parameter choices, e.g.:

-2.2 + 3.1 X – 0.30 X^2



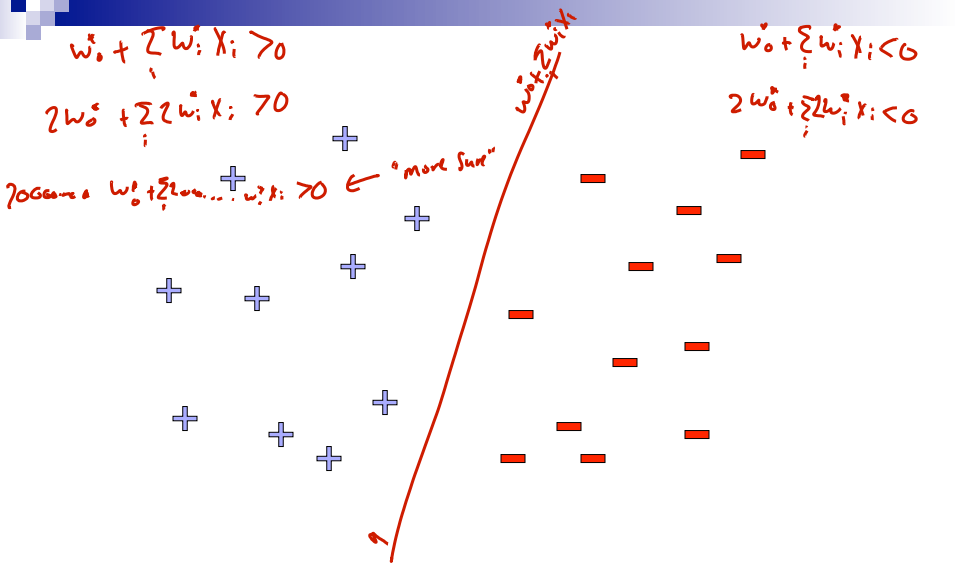
-1.1 + 4,700,910.7 X – 8,585,638.4 X^2 + …



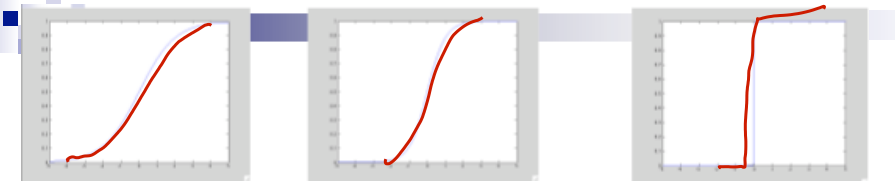
- Regularized least-squares (a.k.a. ridge regression), for λ>0:

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

Linear Separability



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

$P(y=0|w,x) = \frac{1}{1+\exp(w_0+\sum_i w_i x_i)} \to 1$ for negative examples (and vice versa) as $\|w\| \to \infty$

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

Regularized Conditional Log Likelihood

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

$$\ell(\mathbf{w}) = \ln \prod_{j=1}^{N} P(y^j|\mathbf{x}^j,\mathbf{w}) - \frac{\lambda}{2}||\mathbf{w}||_2^2$$

$\frac{\lambda}{2}||\mathbf{w}||_2^2 = \frac{\lambda}{2}\sum_i w_i^2$

- Practical note about w_0:

don't regularize w_0

- Gradient of regularized likelihood:

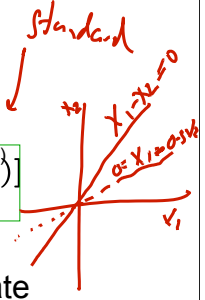
$\frac{\partial \ell}{\partial w_i} = \frac{\partial}{\partial w_i}\left[\sum_j \ln P(y^j|x^j,w)\right] - \frac{\lambda}{2}\frac{\partial}{\partial w_i}\|w\|_2^2$ $\quad 2w_i$

Standard v. Regularized Updates

- Maximum conditional likelihood estimate

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

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



- Regularized maximum conditional likelihood estimate

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

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

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

$$\ell(\mathbf{w}^*) - \ell(\mathbf{w}) \leq \frac{1}{2\lambda} ||\nabla \ell(\mathbf{w})||_2^2$$

- Thus, stop when:

Digression: Logistic regression for more than 2 classes

- Logistic regression in more general case (C classes), where *Y in* {0,…,C-1}

For c class, need $(C-1)(k+1)$ params need

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

$$P(Y=c|X,w) \propto e^{w_{c0} + \sum_i w_{ci} X_i}$$

For $C=0$

$$P(c=0|X,w) = 1 - \sum_{c=1}^{C-1} P(Y=c|X,w)$$

$C=2$

# of params learned $= k+1$ for k features

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

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

Digression: Logistic regression more generally

- Logistic regression in more general case, where *Y in* {0,…,C-1}

for *c>0*

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

normalized

for *c=0* (normalization, so no weights for this class)

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

just like in 2 class case

Learning procedure is basically the same as what we derived!

derivative of ln(h) function