HW1 due thursday

Classification
Logistic Regression

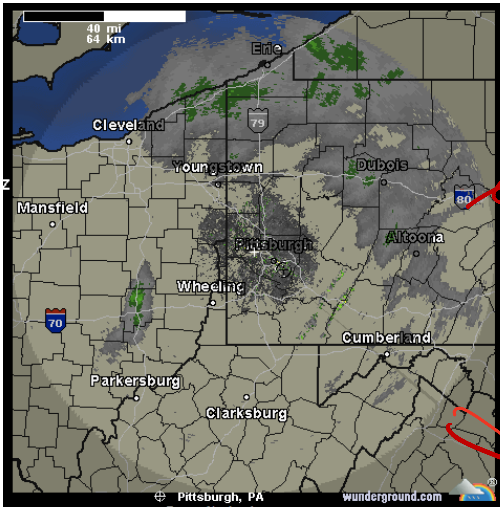
Machine Learning – CSE546

Kevin Jamieson
University of Washington

October 16, 2016

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

Weather prediction revisted



Temperature

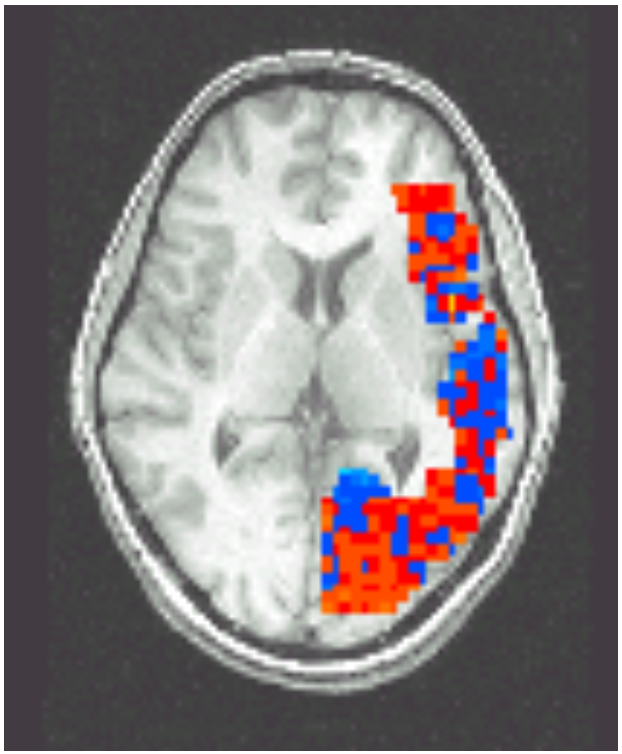
63°F

Reading Your Brain, Simple Example

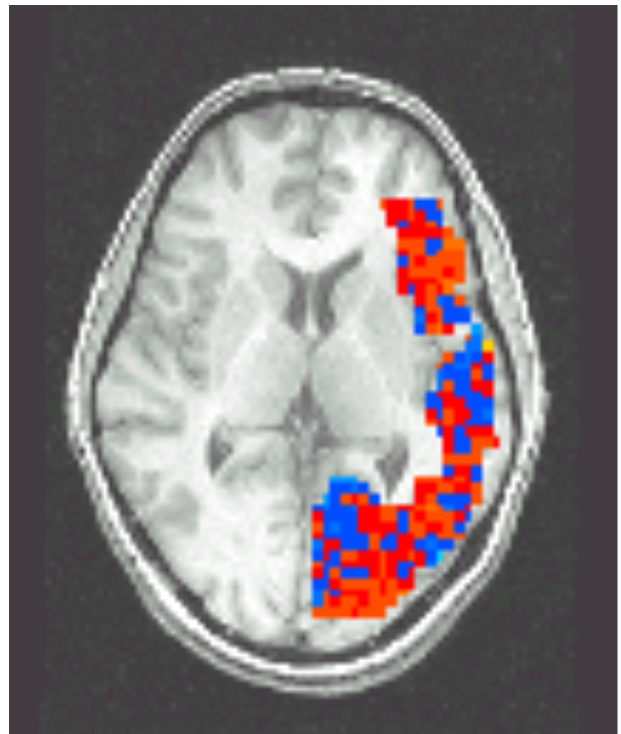
[Mitchell et al.]

Pairwise classification accuracy: 85%

Person



Animal



Binary Classification

- **Learn: f:X —>Y**
  - **X** – features
  - Y – target classes

$$Y \in \{0, 1\}$$

- **Loss function:** $\mathbb{1}\{f(x) \neq y\}$

- **Expected loss of f:**

$$\mathbb{E}_{XY}[\mathbb{1}\{f(X) \neq Y\}] = \mathbb{E}_X[\underbrace{\mathbb{E}_{Y|X}[\mathbb{1}\{f(x) \neq Y\} | X = x]}]$$

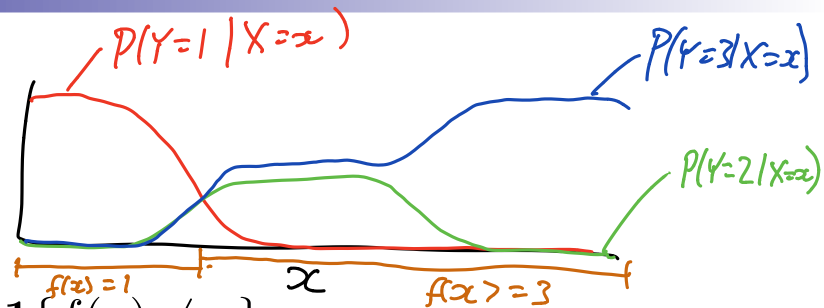
$$\sum_i \mathbb{1}\{f(x) \neq i\} \mathbb{P}(Y = i | X = x) = \sum_{i \neq f(x)} \mathbb{P}(Y = i | X = x) = 1 - \mathbb{P}(Y = f(x) | X = x)$$

- Suppose you know P(Y|**X**) exactly, how should you classify?
  - Bayes optimal classifier:

Binary Classification

- **Learn**: f:**X** —>Y
  - **X** – features
  - Y – target classes

$$Y \in \{0, 1\}$$



- **Loss function:** $\ell(f(x), y) = \mathbf{1}\{f(x) \neq y\}$

- **Expected loss of f:**

$$\mathbb{E}_{XY}[\mathbf{1}\{f(X) \neq Y\}] = \mathbb{E}_X[\mathbb{E}_{Y|X}[\mathbf{1}\{f(x) \neq Y\}|X = x]]$$

$$\mathbb{E}_{Y|X}[\mathbf{1}\{f(x) \neq Y\}|X = x] = \sum_i P(Y = i|X = x)\mathbf{1}\{f(x) \neq i\} = \sum_{i \neq f(x)} P(Y = i|X = x)$$

$$= 1 - P(Y = f(x)|X = x)$$

- Suppose you know P(Y|**X**) exactly, how should you classify?
  - Bayes optimal classifier:

$$f(x) = \arg\max_y \mathbb{P}(Y = y|X = x)$$

Link Functions

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

$$\cancel{P(Y=1 \mid X=x) = x^T w}$$

We need a function that maps $x \in \mathbb{R}^d \to [0,1]$

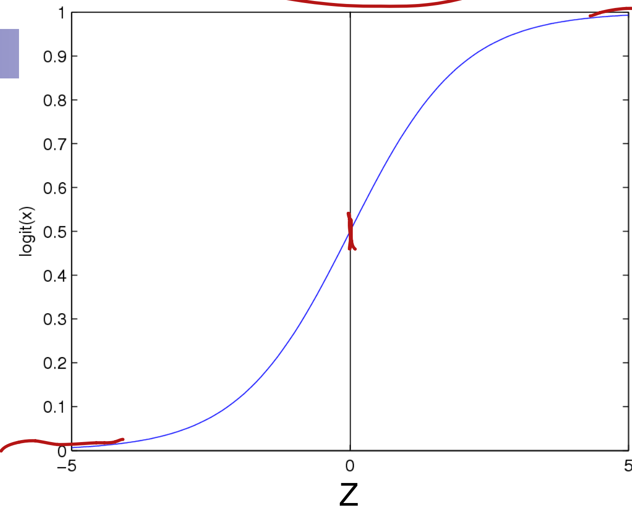
- Combining regression and probability?
  - Need a mapping from real values to [0,1]
  - A link function!

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:

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

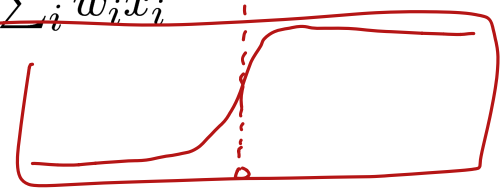


Features can be discrete or continuous!

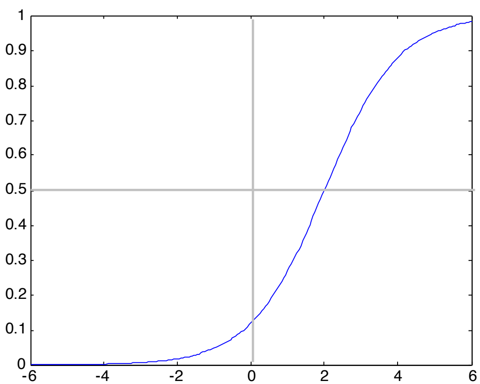
Understanding the sigmoid

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

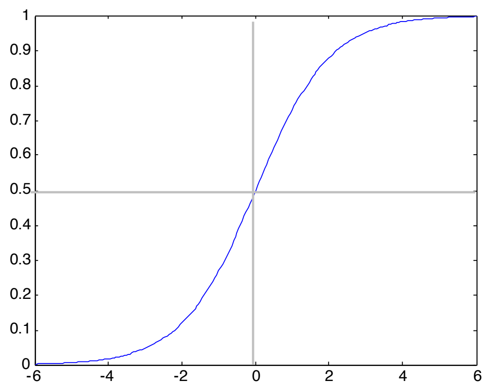
$w_0 = 0,\ w_1 = -2$



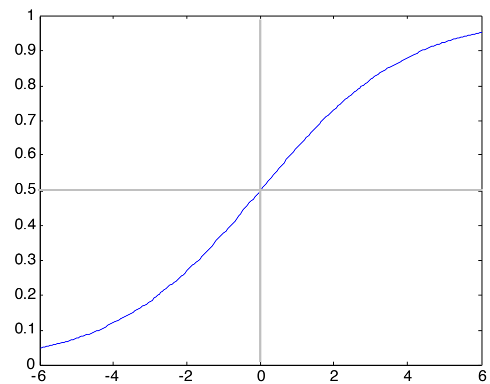
w_0=-2, w_1=-1



w_0=0, w_1=-1



w_0=0, w_1=-0.5



Sigmoid for binary classes

$$\mathbb{P}(Y=0|w,X) = \frac{1}{1+\exp(w_0+\sum_k w_k X_k)}$$

$$\mathbb{P}(Y=1|w,X) = 1-\mathbb{P}(Y=0|w,X) = \frac{\exp(w_0+\overbrace{\sum_k w_k X_k}^{w^T X})}{1+\exp(w_0+\sum_k w_k X_k)}$$

$$\frac{\mathbb{P}(Y=1|w,X)}{\mathbb{P}(Y=0|w,X)} = \exp(w_0 + w^T X) \underset{0}{\overset{1}{\gtrless}} 1$$

$$\log\left(\frac{\mathbb{P}(Y=1|w,X)}{\mathbb{P}(Y=0|w,X)} \right) = w_0 + w^T X \underset{0}{\overset{1}{\gtrless}} 0$$

Sigmoid for binary classes

$$\mathbb{P}(Y=0|w,X) = \frac{1}{1+\exp(w_0+\sum_k w_k X_k)}$$

$$\mathbb{P}(Y=1|w,X) = 1-\mathbb{P}(Y=0|w,X) = \frac{\exp(w_0+\sum_k w_k X_k)}{1+\exp(w_0+\sum_k w_k X_k)}$$

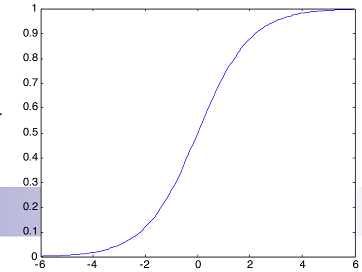
$$\frac{\mathbb{P}(Y=1|w,X)}{\mathbb{P}(Y=0|w,X)} = \exp(w_0+\sum_k w_k X_k)$$

$$\boxed{\log\frac{\mathbb{P}(Y=1|w,X)}{\mathbb{P}(Y=0|w,X)} = w_0+\sum_k w_k X_k}$$

Linear Decision Rule!

Logistic Regression – a Linear classifier

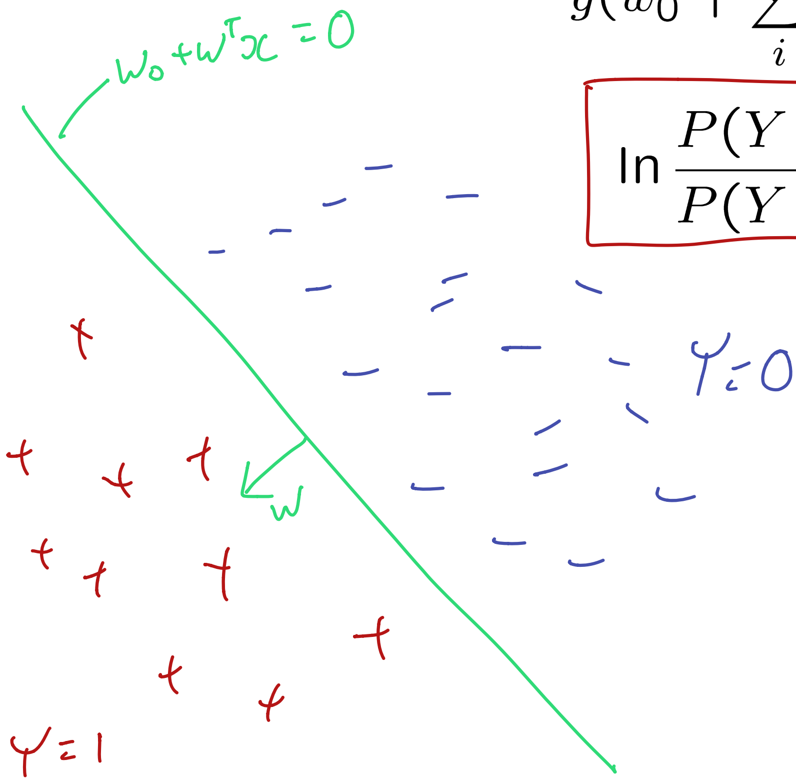
$$\frac{1}{1+exp(-z)}$$



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

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

$= w_0 + w^T X$



Loss function: Conditional Likelihood

- Have a bunch of iid data of the form: $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$

$$P(Y = -1|x, w) = \frac{1}{1 + \exp(w^T x)}$$

$$P(Y = 1|x, w) = \frac{\exp(w^T x)}{1 + \exp(w^T x)} = \frac{1}{1 + \exp(-w^T x)}$$

- This is equivalent to:

$$P(Y = y|x, w) = \frac{1}{1 + \exp(-y\, w^T x)}$$

- So we can compute the maximum likelihood estimator:

$$\widehat{w}_{MLE} = \arg\max_w \prod_{i=1}^n P(y_i|x_i, w)$$

Loss function: Conditional Likelihood

- Have a bunch of iid data of the form: $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$

$$\boxed{P(Y = y|x, w) = \frac{1}{1 + \exp(-y\, w^T x)}}$$

$$\widehat{w}_{MLE} = \underbrace{\arg\max_w \prod_{i=1}^n P(y_i|x_i, w)}$$

$$= \arg\min_w \sum_{i=1}^n \log(1 + \exp(-y_i\, x_i^T w))$$

Loss function: Conditional Likelihood

- Have a bunch of iid data of the form: $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$

$$\widehat{w}_{MLE} = \arg\max_w \prod_{i=1}^n P(y_i | x_i, w)$$

$$\boxed{P(Y = y | x, w) = \frac{1}{1 + \exp(-y\, w^T x)}}$$

$$= \arg\min_w \sum_{i=1}^n \log(1 + \exp(-y_i\, x_i^T w))$$

Logistic Loss: $\boxed{\ell_i(w) = \log(1 + \exp(-y_i\, x_i^T w))}$

Squared error Loss: $\ell_i(w) = (y_i - x_i^T w)^2$ (MLE for Gaussian noise)

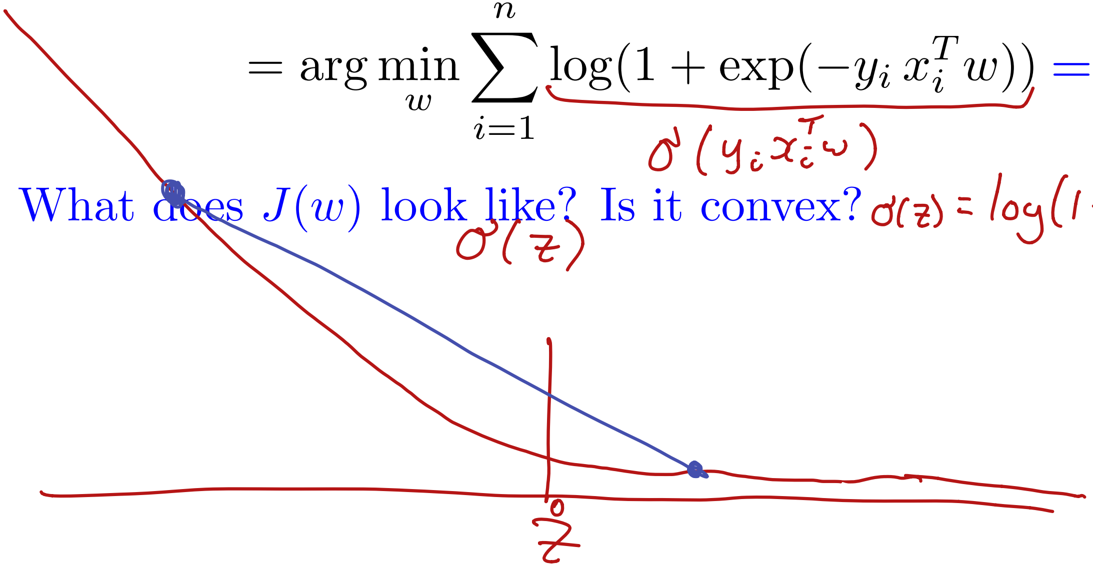
Loss function: Conditional Likelihood

- Have a bunch of iid data of the form: $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$

$$\widehat{w}_{MLE} = \arg\max_w \prod_{i=1}^n P(y_i|x_i, w) \qquad P(Y = y|x, w) = \frac{1}{1 + \exp(-y\, w^T x)}$$

$$= \arg\min_w \sum_{i=1}^n \underbrace{\log(1 + \exp(-y_i\, x_i^T w))}_{\sigma(y_i x_i^T w)} = J(w)$$

What does $J(w)$ look like? Is it convex? $\sigma(z) = \log(1 + \exp(-z))$



for $z << 0$, $\sigma(z) \approx |z|$

for $z >> 0$, $\sigma(z) = 0$

f is convex if $f(\lambda x + (1-\lambda) y) \leq \lambda f(x) + (1-\lambda) f(y)$

Loss function: Conditional Likelihood

- Have a bunch of iid data of the form: $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$

$$\widehat{w}_{MLE} = \arg\max_w \prod_{i=1}^n P(y_i | x_i, w) \qquad P(Y = y | x, w) = \frac{1}{1 + \exp(-y\, w^T x)}$$

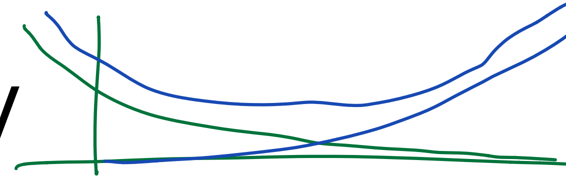
$$= \arg\min_w \sum_{i=1}^n \log(1 + \exp(-y_i\, x_i^T w)) = J(w)$$

Good news: $J(\mathbf{w})$ is convex function of $\mathbf{w}$, no local optima problems

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

Good news: convex functions easy to optimize

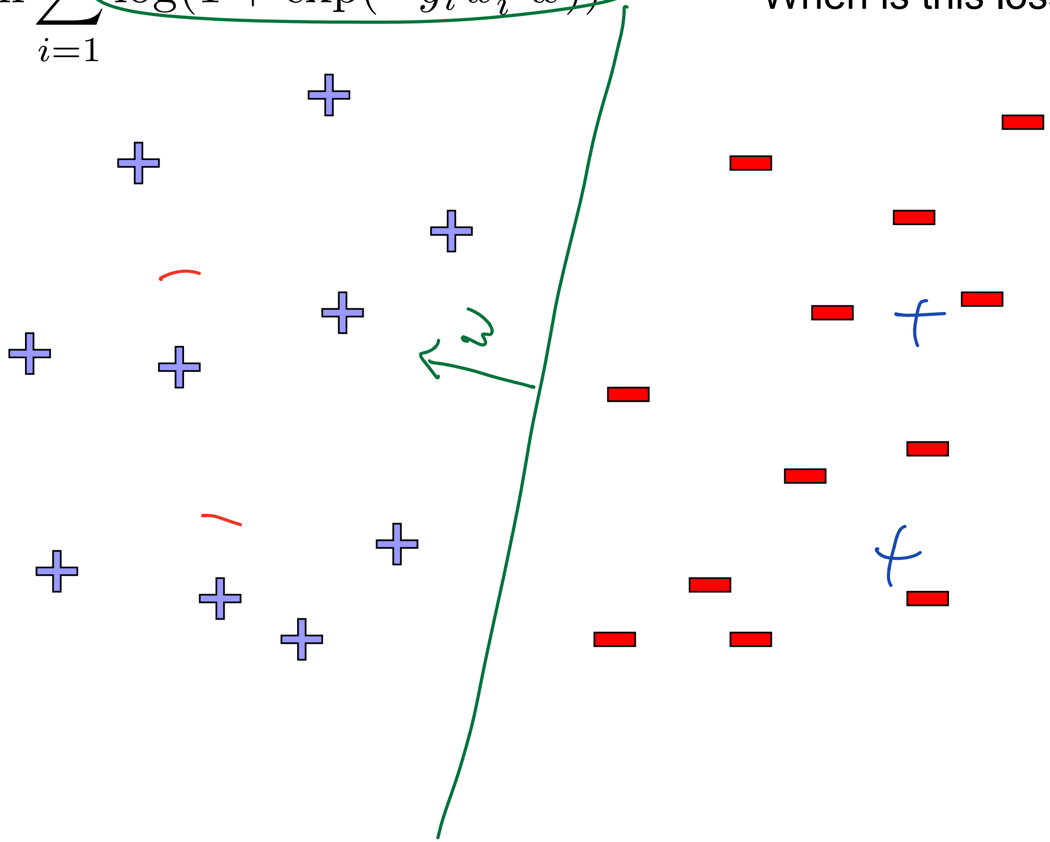
Linear Separability



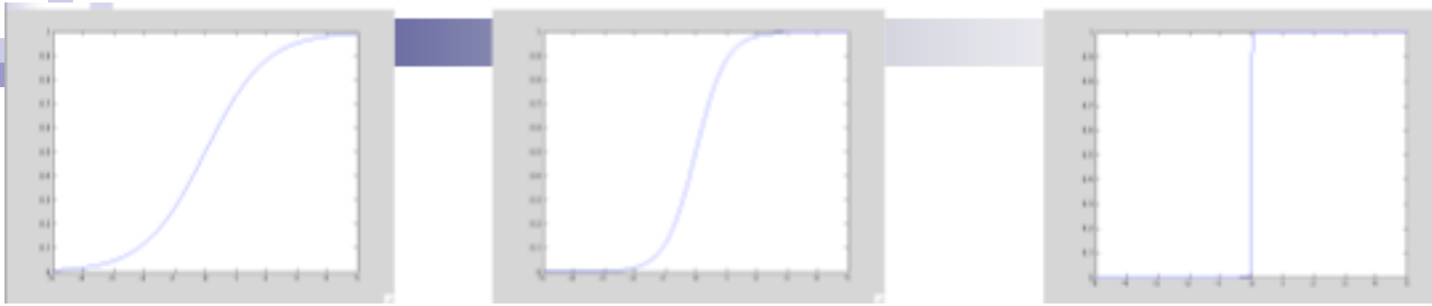
$\|w\|$ increases

$$\arg\min_{w} \sum_{i=1}^{n} \log(1 + \exp(-y_i \, x_i^T w))$$

When is this loss small?



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

Regularized Conditional Log Likelihood

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

$$\underbrace{\arg\min_{w,b} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i\,(x_i^T w + b))\right) + \lambda ||w||_2^2}$$

$\textcolor{blue}{\text{Be sure to not regularize the offset } b\text{!}}$

Gradient Descent

Machine Learning – CSE546
Kevin Jamieson
University of Washington

October 16, 2016

Machine Learning Problems

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters: $$\sum_{i=1}^n \ell_i(w)$$

  Each $\ell_i(w)$ is convex.

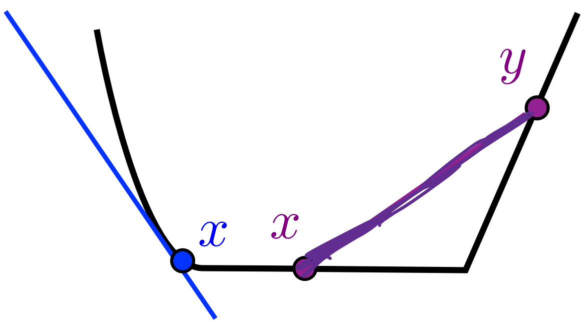
Machine Learning Problems

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters: $\sum_{i=1}^n \ell_i(w)$

Each $\ell_i(w)$ is convex.



g is a subgradient at x if

$$f(y) \geq f(x) + g^T(y - x)$$

f convex:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \qquad \forall x, y, \lambda \in [0,1]$$

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \qquad \forall x, y$$

Machine Learning Problems

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters:

Each $\ell_i(w)$ is convex.

$$\boxed{\sum_{i=1}^n \ell_i(w)}$$

Logistic Loss: $\ell_i(w) = \log(1 + \exp(-y_i\, x_i^T w))$

Squared error Loss: $\ell_i(w) = (y_i - x_i^T w)^2$

Least squares

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters:

$$\sum_{i=1}^n \ell_i(w)$$

Each $\ell_i(w)$ is convex.

Squared error Loss: $\ell_i(w) = (y_i - x_i^T w)^2$

How does software solve: $\frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

$\equiv (X^T X)w = X^T y$

Find x : $Ax = b$

Least squares

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters: $\sum_{i=1}^n \ell_i(w)$

Each $\ell_i(w)$ is convex.

Squared error Loss: $\ell_i(w) = (y_i - x_i^T w)^2$

How does software solve: $\frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

…its complicated:
(LAPACK, BLAS, MKL…)

Do you need high precision?
Is X column/row sparse?
Is $\widehat{w}_{LS}$ sparse?
Is $\mathrm{X}^T\mathrm{X}$ "well-conditioned"?
Can $\mathrm{X}^T\mathrm{X}$ fit in cache/memory?

Taylor Series Approximation

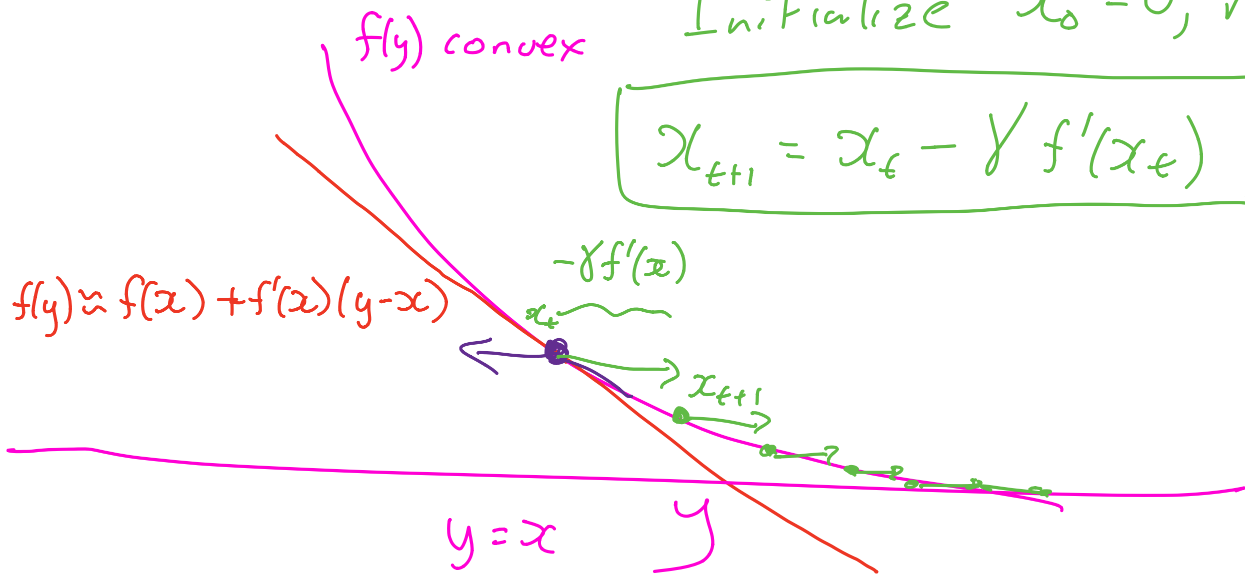
- Taylor series in one dimension:

$$f(x+\delta) = \underbrace{f(x) + f'(x)\delta}_{} + \tfrac{1}{2} f''(x)\delta^2 + \ldots$$

- Gradient descent:

Initialize $x_0 = 0$, randomly

$$\boxed{x_{t+1} = x_t - \gamma f'(x_t)}$$



Taylor Series Approximation

- Taylor series in **d** dimensions:

$$f(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v + \dots$$

- Gradient descent:

$$x_{t+1} = x_t - \gamma \nabla f(x_t)$$

Gradient Descent $f(w) = \frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

$$\boxed{w_{t+1} = w_t - \eta \nabla f(w_t)}$$

$$\nabla f(w) = X^T(Xw - y) = X^TXw - X^Ty$$

$$w_{t+1} = w_t - \eta\underbrace{\left(\underbrace{X^T}_{d\times n}\underbrace{(Xw_t - y)}_{n\times 1}\right)}_{d\times 1}$$

$$= w_t - \eta X^TXw_t + \eta X^Ty$$

$$= (I - \eta X^TX)w_t + \eta X^Ty$$

$$w_{t+1} - w_* = (I - \eta X^TX)(w_t - w_*) \cancel{- \eta X^TXw_* + \eta X^Ty}$$

$$\Sigma X^T X w_* + \Sigma X^T y = \Sigma X^T (X w_* + y)$$
$$= \Sigma \nabla f(w_*)$$
$$= 0$$

Gradient Descent $\quad f(w) = \underbrace{\frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2}$

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

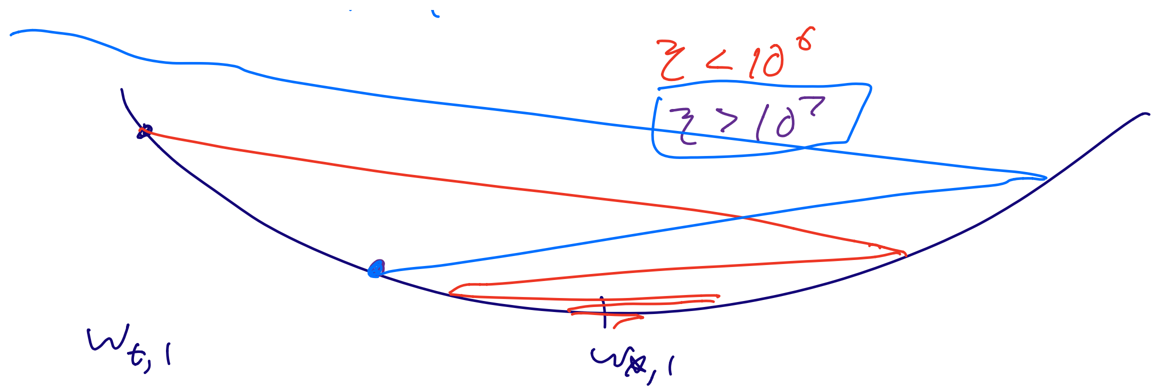
$$(w_{t+1} - w_*) = (I - \eta \mathrm{X}^T \mathrm{X})(w_t - w_*)$$
$$= \boxed{(I - \eta \mathrm{X}^T \mathrm{X})^{t+1}}(w_0 - w_*)$$

Example: $\mathrm{X} = \begin{bmatrix} 10^{-3} & 0 \\ 0 & 1 \end{bmatrix} \quad \mathrm{y} = \begin{bmatrix} 10^{-3} \\ 1 \end{bmatrix} \quad w_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad w_* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$X^TX = \begin{bmatrix} 10^{-6} & 0 \\ 0 & 1 \end{bmatrix}$ $\quad D$ diagonal $\Rightarrow D^k$ = kth power of diagonal

$$(w_{t+1,1} - w_{*,1}) = \underbrace{(1 - 2\,10^{-6})}_{\text{abs. value} < 1 \text{ --}}{}^{t+1}(w_{0,1} - w_{*,1})$$

$$(w_{t+1,2} - w_{*,1}) = (1 - 2)^{t+1}(w_{0,2} - w_{*,2})$$


$\eta < 10^{6}$
$\eta > 10^{7}$
$w_{t,1}$
$w_{*,1}$

Taylor Series Approximation

- Taylor series in one dimension:

$$f(x+\delta) = \underbrace{f(x) + f'(x)\delta + \tfrac{1}{2} f''(x)\delta^2}_{} + \dots$$

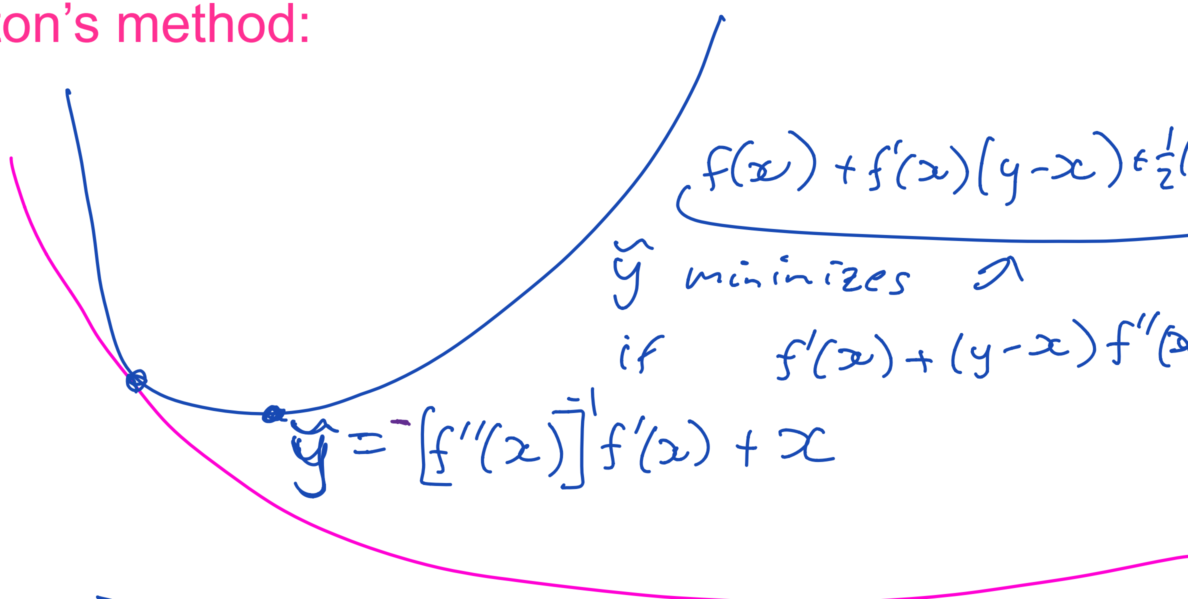
- Newton's method:

$$\underbrace{f(x) + f'(x)(y-x) + \tfrac{1}{2}(y-x)^2 f''(x)}_{}$$

$\hat{y}$ minimizes

if $f'(x) + (y-x) f''(x) = 0$

$$\hat{y} = -[f''(x)]^{-1} f'(x) + x$$



$y = x$

Taylor Series Approximation

- Taylor series in **d** dimensions:

$$f(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v + \dots$$

- Newton's method:

$$x_{t+1} = x_t + \eta v_t$$

$$v_t = -\left[\nabla^2 f(x)\right]^{-1} \nabla f(x)$$

Newton's Method $f(w) = \frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

$$\nabla f(w) =$$

$$\nabla^2 f(w) =$$

$$v_t \text{ is solution to} : \nabla^2 f(w_t) v_t = -\nabla f(w_t)$$

$$w_{t+1} = w_t + \eta v_t$$

Newton's Method $\quad f(w) = \frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

$$\nabla f(w) = \mathrm{X}^T(\mathrm{X}w - \mathrm{y})$$

$$\nabla^2 f(w) = \mathrm{X}^T\mathrm{X}$$

$$v_t \text{ is solution to} : \nabla^2 f(w_t)v_t = -\nabla f(w_t)$$

$$w_{t+1} = w_t + \eta v_t$$

For quadratics, Newton's method converges in one step! (Not a surprise, why?)

$$w_1 = w_0 - \eta(\mathrm{X}^T\mathrm{X})^{-1}\mathrm{X}^T(\mathrm{X}w_0 - y) = w_*$$

General case

In general for Newton's method to achieve $f(w_t) - f(w_*) \leq \epsilon$:

So why are ML problems overwhelmingly solved by gradient methods?

Hint: v_t is solution to : $\nabla^2 f(w_t) v_t = -\nabla f(w_t)$

General Convex case $f(w_t) - f(w_*) \leq \epsilon$

Clean converge nice proofs: Bubeck

Newton's method:

$$t \approx \log(\log(1/\epsilon))$$

Gradient descent:

- f is *smooth* and *strongly convex:* $aI \preceq \nabla^2 f(w) \preceq bI$
- f is *smooth*: $\nabla^2 f(w) \preceq bI$
- f is potentially non-differentiable: $||\nabla f(w)||_2 \leq c$

Nocedal +Wright, Bubeck

Other: BFGS, Heavy-ball, BCD, SVRG, ADAM, Adagrad,…

Revisiting… Logistic Regression

Machine Learning – CSE546
Kevin Jamieson
University of Washington

October 16, 2016

Loss function: Conditional Likelihood

- Have a bunch of iid data of the form: $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$

$$\widehat{w}_{MLE} = \arg\max_w \prod_{i=1}^n P(y_i | x_i, w) \qquad P(Y = y | x, w) = \frac{1}{1 + \exp(-y\, w^T x)}$$

$$f(w) = \arg\min_w \sum_{i=1}^n \log(1 + \exp(-y_i\, x_i^T w))$$

$$\nabla f(w) =$$