Classification
Logistic Regression

Matt Golub
Hunter Schafer

UNIVERSITY of WASHINGTON

Thus far, regression:

predict a continuous value given some inputs

Weather prediction revisited

Classification

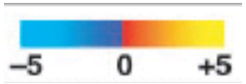
Temperature

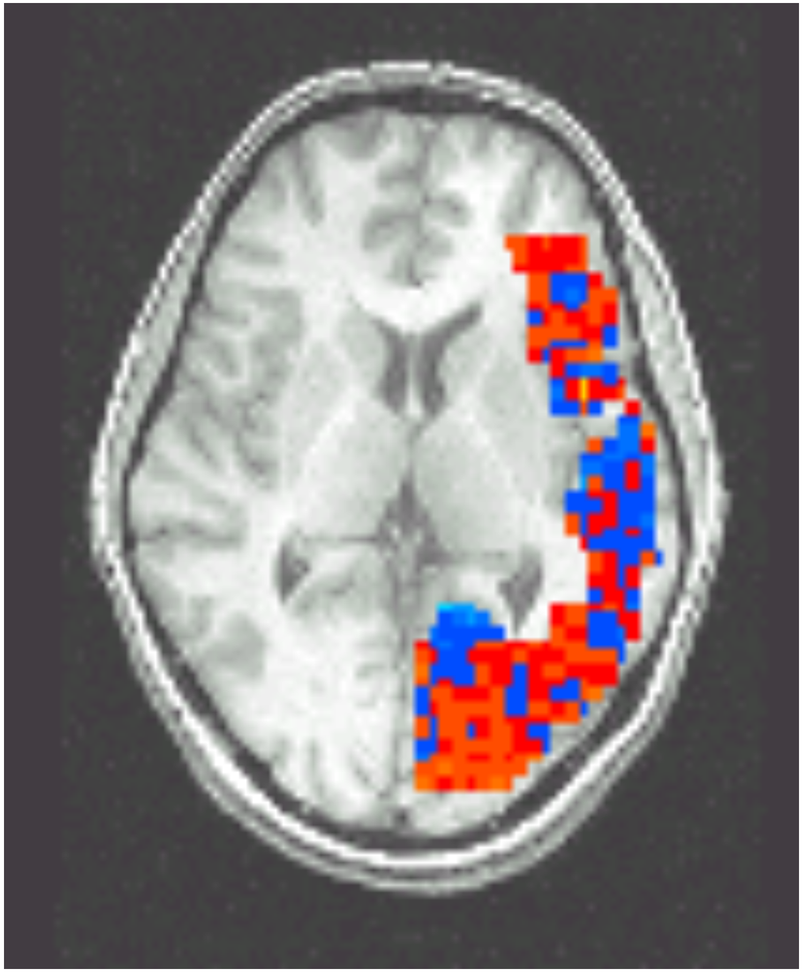
Regression

Reading Your Brain

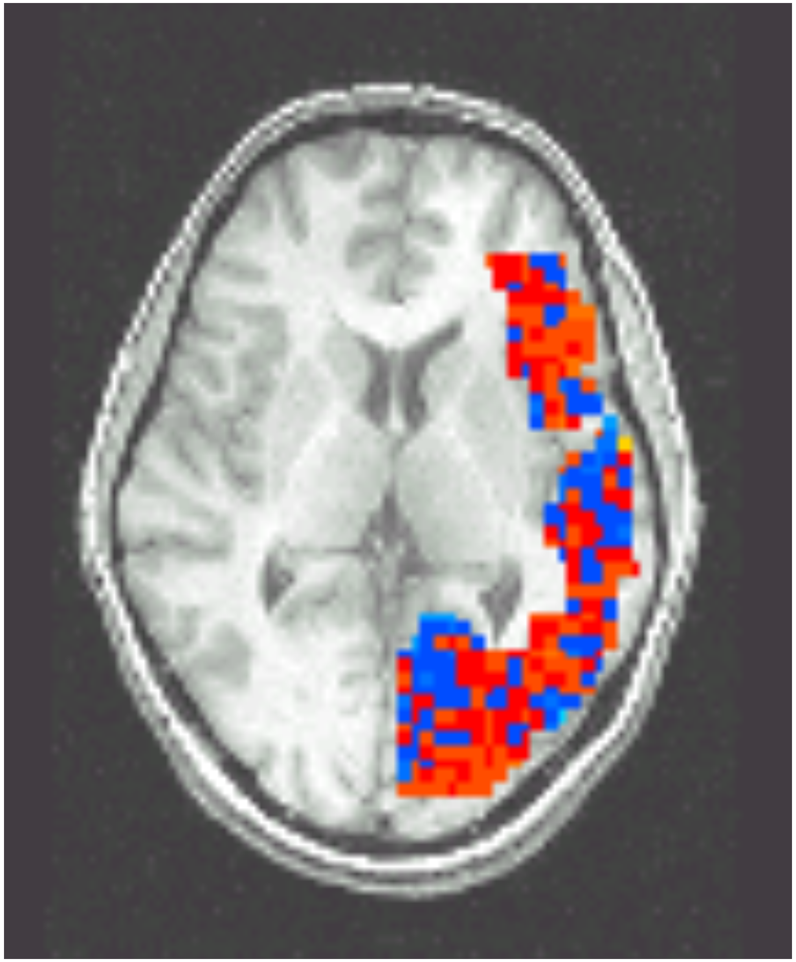
Pairwise classification accuracy: 85%

Person





Animal



[Mitchell et al.]

Classification

- **Learn** $f: \mathcal{X} \rightarrow \mathcal{Y}$
  - **Features:** $\mathcal{X} \subset \mathbb{R}^d$
  - **Target classes:** $\mathcal{Y} = \{1, ..., k\}$

- **Loss Function:** $\mathscr{L}(f(\mathbf{x}), y) = \mathbf{1}\{f(\mathbf{x}) \neq y\}$
- **Expected loss of** f**:**

$$(X_i, Y_i) \overset{iid}{\sim} D \qquad \mathbb{E}_{XY}\left[\mathbb{1}\{f(X) \neq Y\}\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|X}\left[\mathbb{1}\{f(x) \neq y\} \mid X = x\right]\right]$$

Classification

- **Learn** $f: \mathcal{X} \rightarrow \mathcal{Y}$
  - **Features:** $\mathcal{X} \subset \mathbb{R}^d$
  - **Target classes:** $\mathcal{Y} = \{1, ..., k\}$

- **Loss Function:** $\mathscr{L}(f(\mathbf{x}), y) = \mathbf{1}\{f(\mathbf{x}) \neq y\}$
- **Expected loss of** f**:** $\mathbb{E}_{XY}[\mathbf{1}\{f(X) \neq Y\}] = \mathbb{E}_X\left[\mathbb{E}_{Y|X}[\mathbf{1}\{f(X) \neq Y\}|X = \mathbf{x}]\right]$

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

$$= 1 - P(Y = f(X)|X = \mathbf{x})$$

Use definition of expectation

Can remove term corresponding to $f(X) = i$ b/c $\mathbb{1}(\cdots) = 0$ for that i. Then, drop the $\mathbb{1}$ altogether, since only keeping i s.t. $\mathbb{1}(\cdots) = 1$

since $\sum_i P(Y=i|X=x) = 1$

- **Suppose you knew P(Y | X) exactly, how should you classify?**

Classification

- **Learn** $f: \mathcal{X} \to \mathcal{Y}$
  - **Features:** $\mathcal{X} \subset \mathbb{R}^d$
  - **Target classes:** $\mathcal{Y} = \{1, ..., k\}$

- **Loss Function:** $\mathscr{L}(f(\mathbf{x}), y) = \mathbf{1}\{f(\mathbf{x}) \neq y\}$
- **Expected loss of** f**:** $\mathbb{E}_{XY}[\mathbf{1}\{f(X) \neq Y\}] = \mathbb{E}_X\left[\mathbb{E}_{Y|X}[\mathbf{1}\{f(X) \neq Y\}|X = \mathbf{x}]\right]$

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

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

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

Bayes Optimal Binary Classifier

- **Bayes-Optimal classifier:** $f(\mathbf{x}) = \arg\max_y \mathbb{P}(Y = y | X = \mathbf{x})$
- **Suppose we don't know** $P(Y = y \mid X = \mathbf{x})$**, but have n iid examples**

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^n \qquad Y \in \{0,1\}$$

- **Suppose** $\mathcal{X}$ **is discrete so that** $X \in \{1,2,\ldots,m\}$**. What is a natural estimator for** $P(Y = y \mid X = x)$**?**

Bayes Optimal Binary Classifier

- **Bayes-Optimal classifier:** $f(\mathbf{x}) = \arg\max_y \mathbb{P}(Y = y | X = \mathbf{x})$
- **Suppose we don't know $P(Y = y | X = \mathbf{x})$, but have n iid examples**

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^n \qquad Y \in \{0,1\}$$

- **Suppose $\mathcal{X}$ is discrete so that $X \in \{1,2,\ldots,m\}$. What is a natural estimator for $P(Y = y | X = x)$?**

"Counting" approximation

Lots of issues with this!

$$\hat{f}(x) = \arg\max_{y \in \{0,1\}} \frac{\sum_{i=1}^n \mathbf{1}[x_i = x, y_i = y]}{\sum_{i=1}^n \mathbf{1}[x_i = x]} \approx \frac{P(X=x, Y=y)}{P(X=x)}$$

$$= P(Y=y | X=x)$$

What if $\mathcal{X}$ is continuous? That is, what if $X \in \mathbb{R}^d$?

Bayes Optimal Binary Classifier

- **Bayes-Optimal classifier:** $f(\mathbf{x}) = \arg\max_y \mathbb{P}(Y = y | X = \mathbf{x})$
- **Suppose we don't know $P(Y = y | X = \mathbf{x})$, but have n iid examples**

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^n \qquad Y \in \{0,1\}$$

- **Suppose $\mathcal{X}$ is discrete so that $X \in \{1,2,\ldots,m\}$. What is a natural estimator for $P(Y = y | X = x)$?**

$$\hat{f}(x) = \arg\max_{y \in \{0,1\}} \frac{\sum_{i=1}^n \mathbf{1}[x_i = x, y_i = y]}{\sum_{i=1}^n \mathbf{1}[x_i = x]}$$

What if $\mathcal{X}$ is continuous? That is, what if $X \in \mathbb{R}^d$?

We need a <u>model</u> to explain observations

Logistic Regression

Recall linear regression:

We assumed that for any $\mathbf{x}$, we have: $p(Y = y \,|\, \mathbf{X} = \mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y - \mathbf{w}^T\mathbf{x})^2}$

Given data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ we then computed the MLE for $\mathbf{w}$.

Logistic Regression

Recall linear regression:

We assumed that for any $\mathbf{x}$, we have: $p(Y = y \,|\, \mathbf{X} = \mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y - \mathbf{w}^T\mathbf{x})^2}$

Given data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ we then computed the MLE for $\mathbf{w}$.

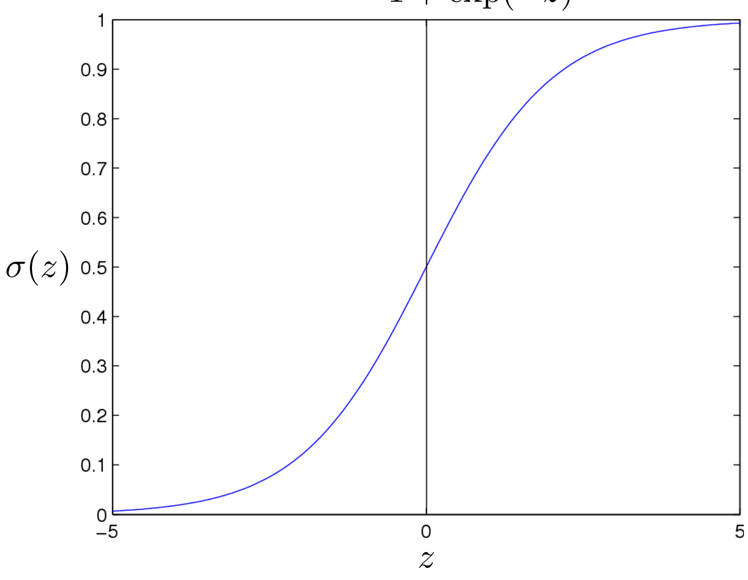
Logistic regression uses a model specialized for classification:

$$\mathbb{P}\left[Y = 1 | \mathbf{X} = \mathbf{x}, \mathbf{w}\right] = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

$$\mathbb{P}\left[Y = 0 | \mathbf{X} = \mathbf{x}, \mathbf{w}\right] = 1 - \sigma(\mathbf{w}^T\mathbf{x}) = \frac{\exp(-\mathbf{w}^T\mathbf{x})}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

$$= \frac{1}{1 + \exp(\mathbf{w}^T\mathbf{x})}$$

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



Logistic Regression

Recall linear regression:

We assumed that for any $\mathbf{x}$, we have: $p(Y = y \,|\, \mathbf{X} = \mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y - \mathbf{w}^T\mathbf{x})^2}$

Given data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ we then computed the MLE for $\mathbf{w}$.

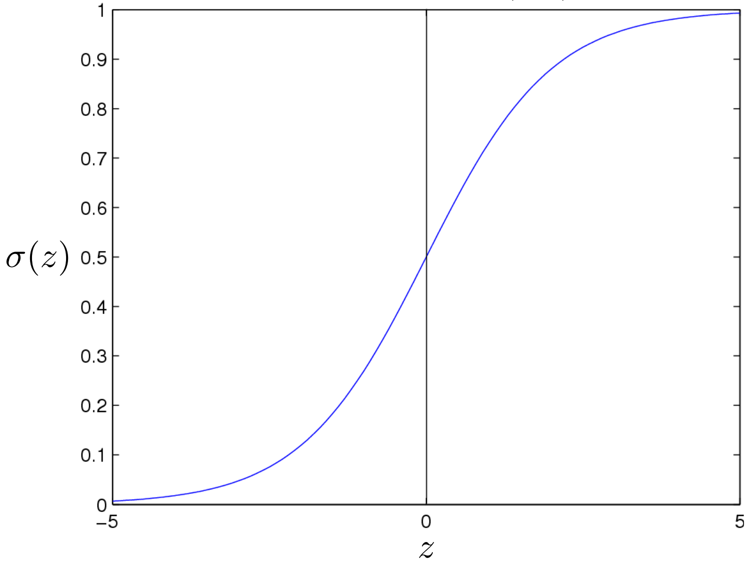
Logistic regression uses a model specialized for classification:

$$\mathbb{P}[Y = 1 | \mathbf{X} = \mathbf{x}, \mathbf{w}] = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

$$\mathbb{P}[Y = 0 | \mathbf{X} = \mathbf{x}, \mathbf{w}] = 1 - \sigma(\mathbf{w}^T\mathbf{x}) = \frac{\exp(-\mathbf{w}^T\mathbf{x})}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

$$= \frac{1}{1 + \exp(\mathbf{w}^T\mathbf{x})}$$

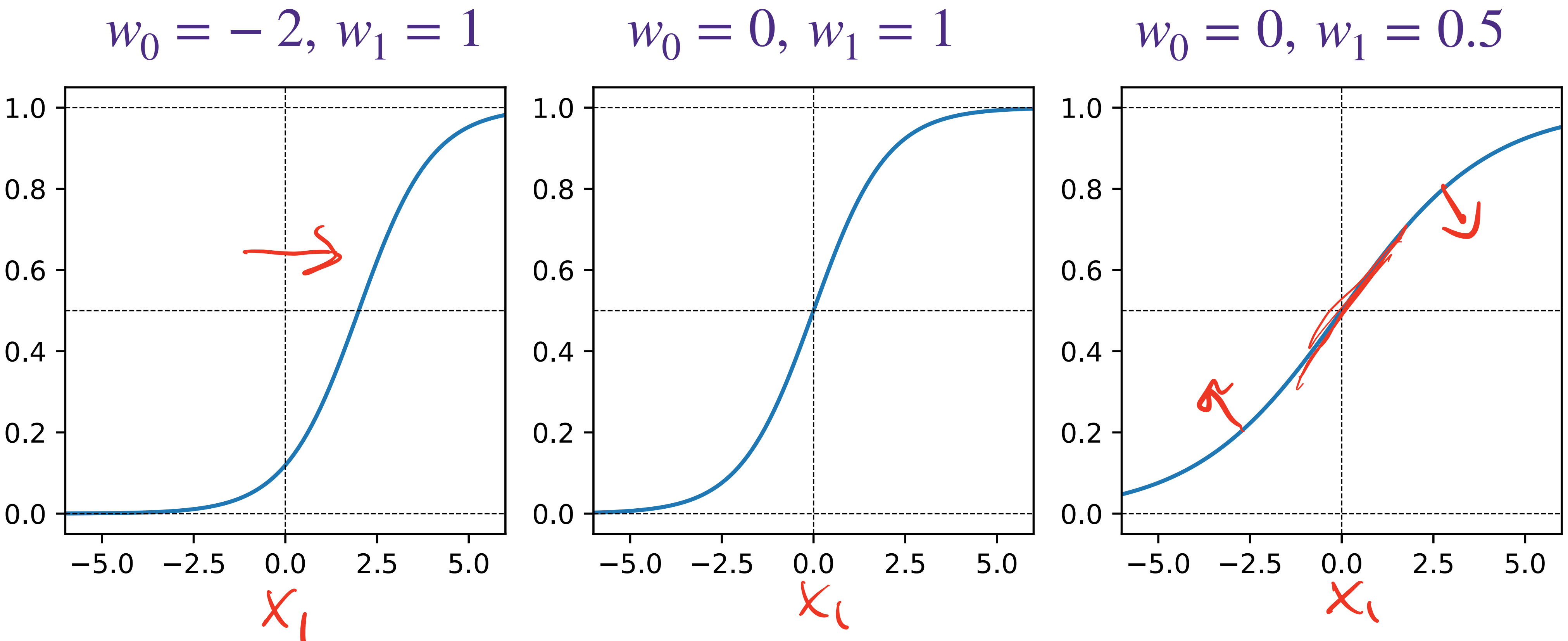
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



Features can be discrete or continuous!

Understanding the sigmoid

$$\sigma\left(w_0 + \sum_{k=1}^{d} w_k x_k\right) = \frac{1}{1 + e^{-(w_0 + \sum_k w_k x_k)}}$$



Sigmoid for binary classes

$$\mathbb{P}[Y = 1 | \mathbf{X} = \mathbf{x}] = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x} - w_0)}$$

$$\mathbb{P}[Y = 0 | \mathbf{X} = \mathbf{x}] = \frac{\exp(-\mathbf{w}^T \mathbf{x} - w_0)}{1 + \exp(-\mathbf{w}^T \mathbf{x} - w_0)}$$

"odds"

$$\frac{\mathbb{P}[Y = 1 | \mathbf{X} = \mathbf{x}]}{\mathbb{P}[Y = 0 | \mathbf{X} = \mathbf{x}]} = \frac{1}{\exp(-w^T x - w_0)} = \exp(w^T x + w_0)$$

$$\log(\text{"odds"}) = w^T x + w_0$$

Sigmoid for binary classes

$$\mathbb{P}\left[Y=1|\mathbf{X}=\mathbf{x}\right]=\frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x}-w_0)}$$

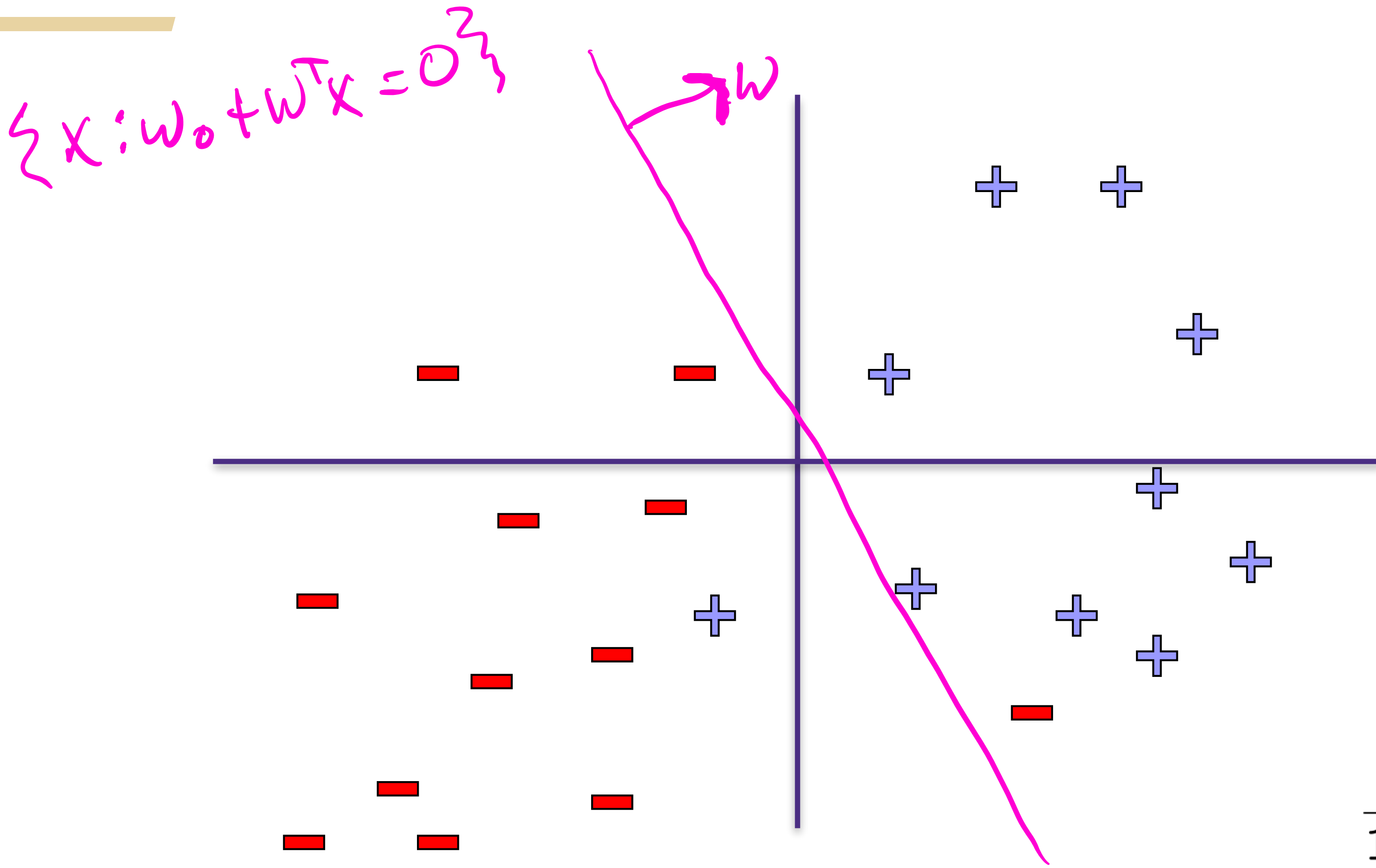
$$\mathbb{P}\left[Y=0|\mathbf{X}=\mathbf{x}\right]=\frac{\exp(-\mathbf{w}^T\mathbf{x}-w_0)}{1+\exp(-\mathbf{w}^T\mathbf{x}-w_0)}$$

$$\frac{\mathbb{P}\left[Y=1|\mathbf{X}=\mathbf{x}\right]}{\mathbb{P}\left[Y=0|\mathbf{X}=\mathbf{x}\right]}=\exp(w_0+\mathbf{w}^T\mathbf{x})=\exp(w_0+\sum_{k=1}^{d}w_kx_k)$$

Linear Decision Rule!

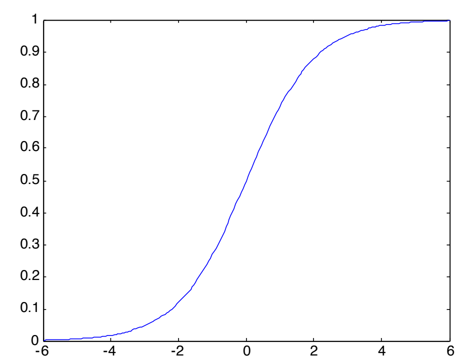
$$\log\frac{\mathbb{P}\left[Y=1|\mathbf{X}=\mathbf{x}\right]}{\mathbb{P}\left[Y=0|\mathbf{X}=\mathbf{x}\right]}=w_0+\sum_{k=1}^{d}w_kx_k$$

Logistic Regression – A Linear Classifier



$$\log \frac{\mathbb{P}\left[Y=1|\mathbf{X}=\mathbf{x}\right]}{\mathbb{P}\left[Y=0|\mathbf{X}=\mathbf{x}\right]} = w_0 + \sum_{k=1}^{d} w_k x_k$$

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



Loss function: Conditional Likelihood

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

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

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

- **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:** $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$

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

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

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

Maximize likelihood

$\log \frac{1}{z} = \log z^{-1} = -\log z$

=

Minimize negative log likelihood

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

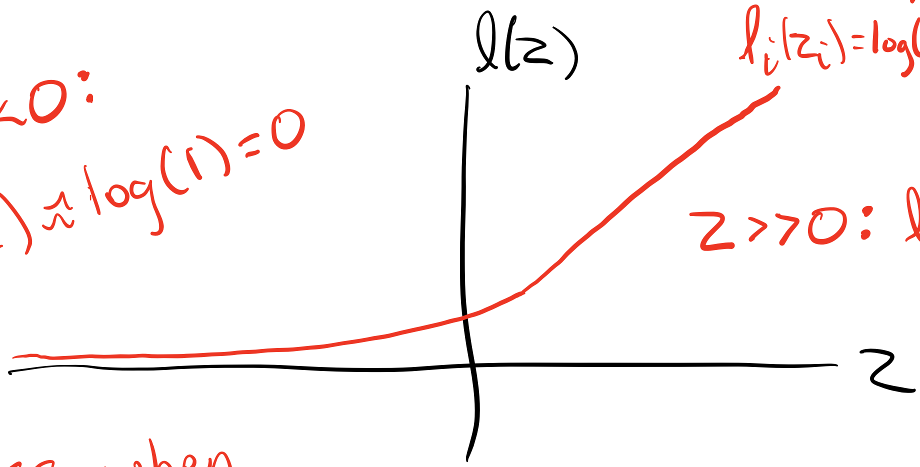
Squared error Loss: $\ell_i(w) = (y_i - x_i^T w)^2$

(MLE for Gaussian noise)

Logistic loss: $l_i(w) = \log(1 + \exp(-y_i x_i^T w))$

$z_i = -y_i x_i^T w$

$l_i(z_i) = \log(1 + \exp(z_i))$



$z << 0$: $l(z) \approx \log(1) = 0$

$z >> 0$: $l(z) \approx \log(\exp(z)) = z$ (Linear)

low loss when

$\text{sign}(y_i) = \text{sign}(x_i^T w)$

Then, if possible, increase $||w||$ without affecting $\text{sign}(x_i^T w)$.

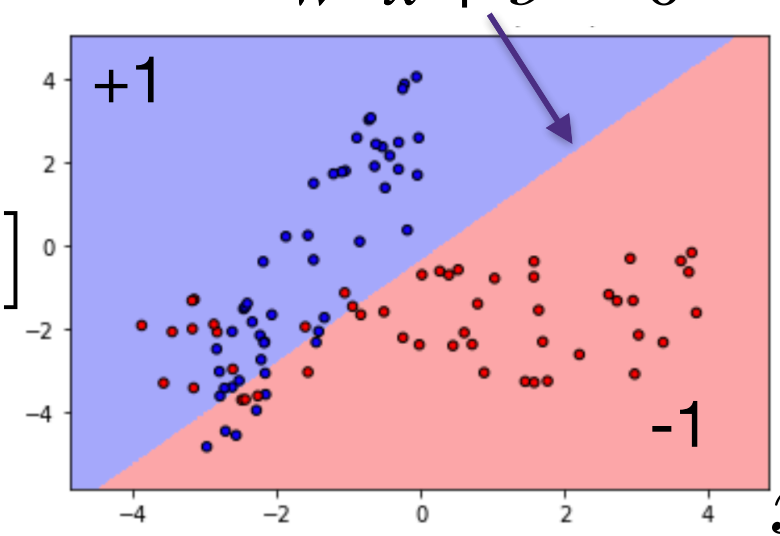
Logistic regression for binary classification

- Data $\mathcal{D} = \{(x_i \in \mathbb{R}^d, y_i \in \{-1, +1\})\}_{i=1}^n$
- Model: $P\left(Y = y \,|\, x, w\right) = \dfrac{1}{1 + \exp(-y(w^T x + b))}$
- Loss function: logistic loss $\ell(w, b) = \log(1 + e^{-y_i(w^T x + b)})$
- Optimization: solve for

$$(\widehat{b}, \widehat{w}) = \arg\min_{b,w} \sum_{i=1}^{n} \log(1 + e^{-y_i(w^T x + b)})$$

- As this is a **smooth convex** optimization, it can be solved efficiently using gradient descent
- Prediction: $\text{sign}(w^T x + b)$

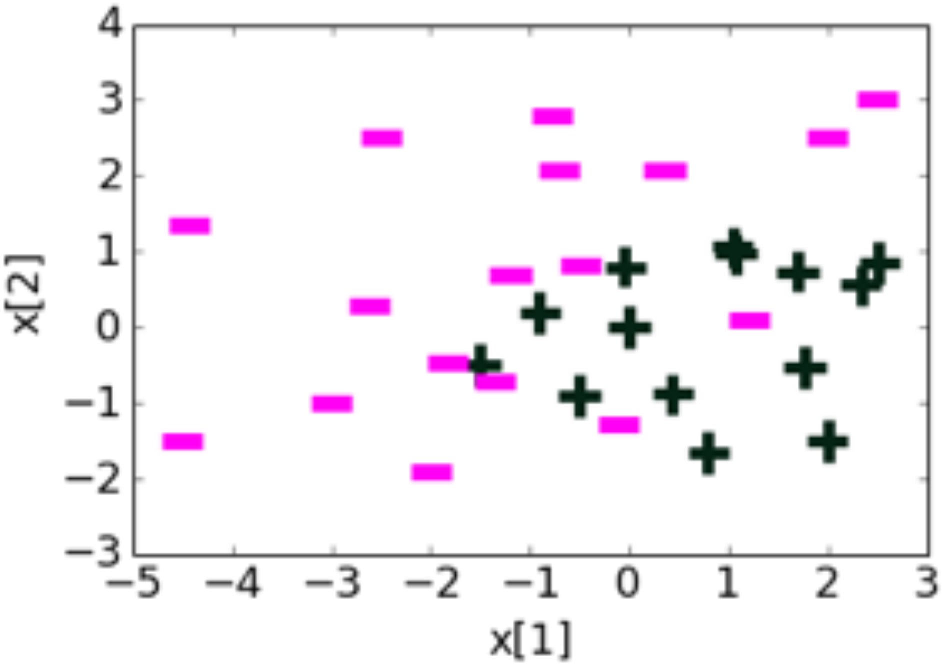
decision boundary at $w^T x + b = 0$



$x[2]$

$x[1]$

Example: adding more polynomial features



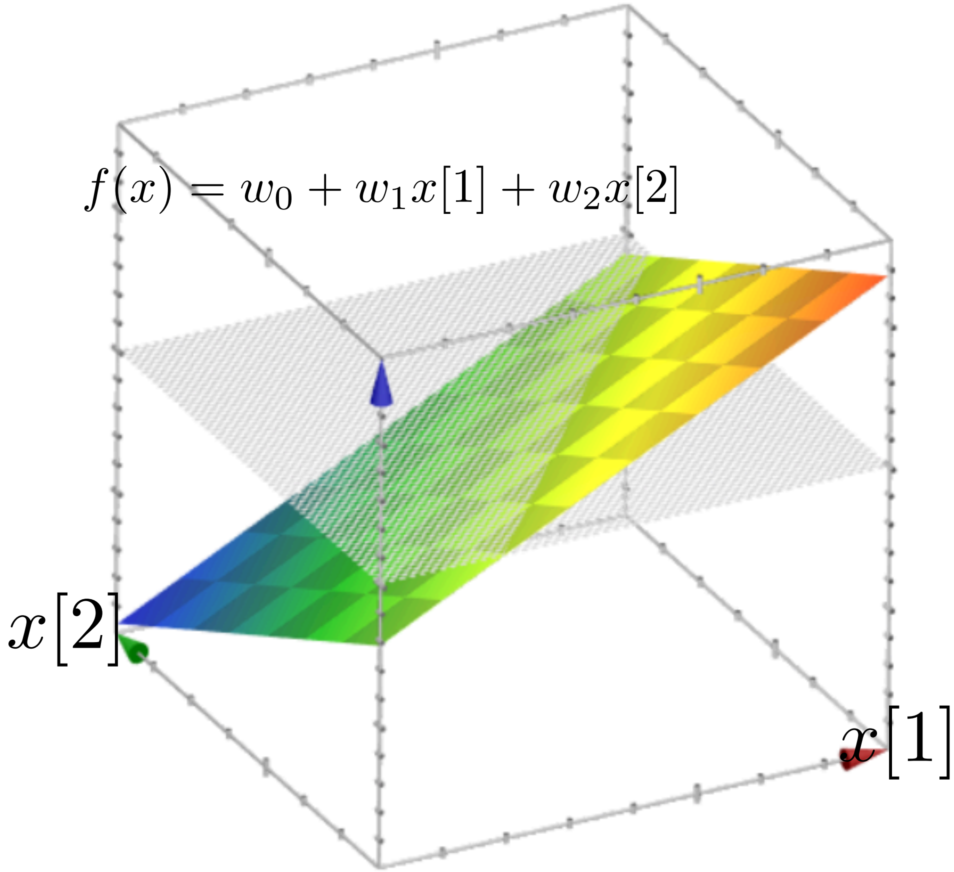
Polynomial features

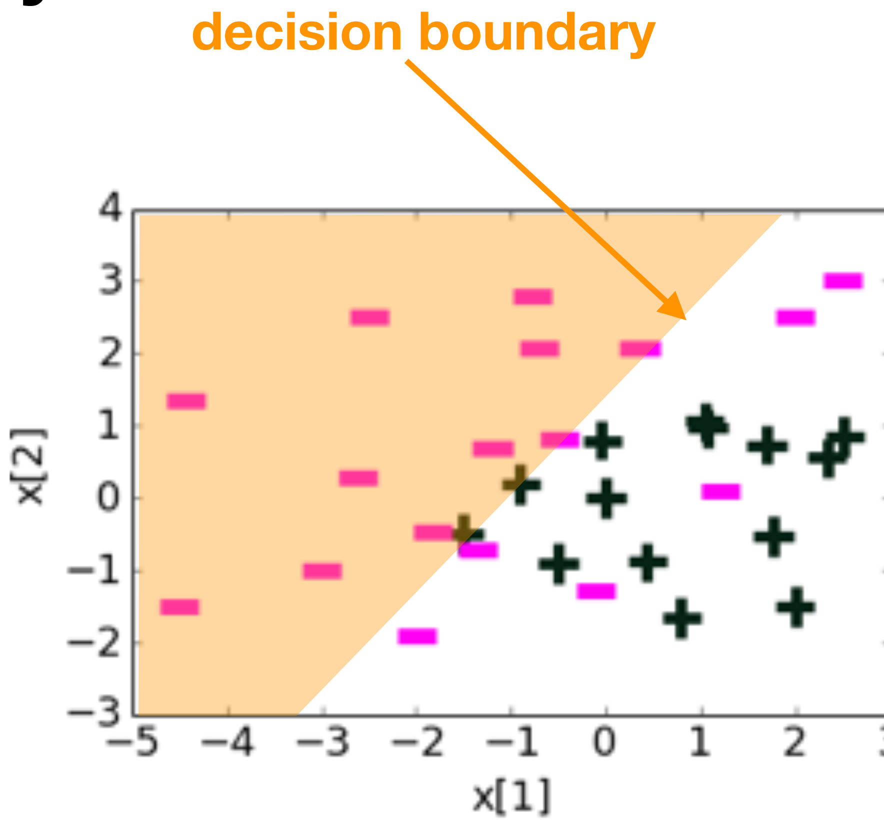
$$\begin{bmatrix} h_0(x) = 1 \\ h_1(x) = x[1] \\ h_2(x) = x[2] \\ h_3(x) = x[1]^2 \\ h_4(x) = x[2]^2 \\ \vdots \end{bmatrix}$$

- data: ***x*** in 2-dimensions, ***y*** in {+1,-1}
- features: polynomials
- model: linear on polynomial features

$$f(x) \; = \; w_0 h_0(x) + w_1 h_1(x) + w_2 h_2(x) + \cdots$$

Learned decision boundary

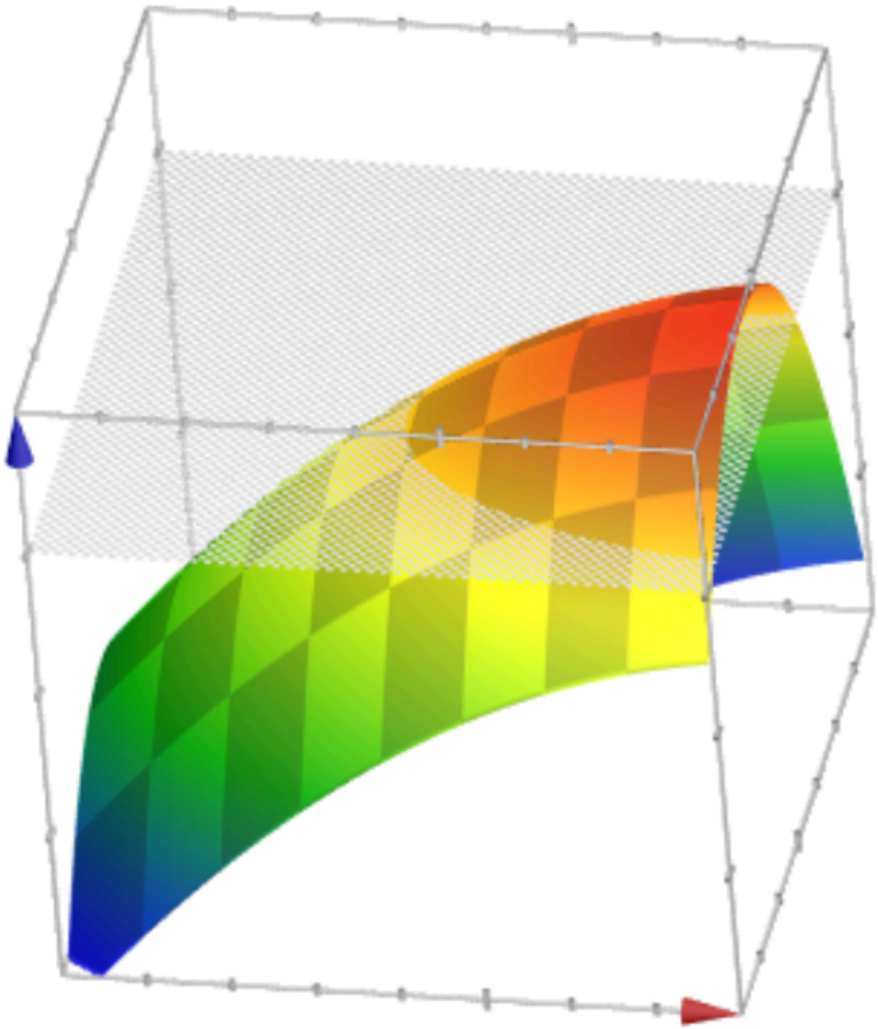


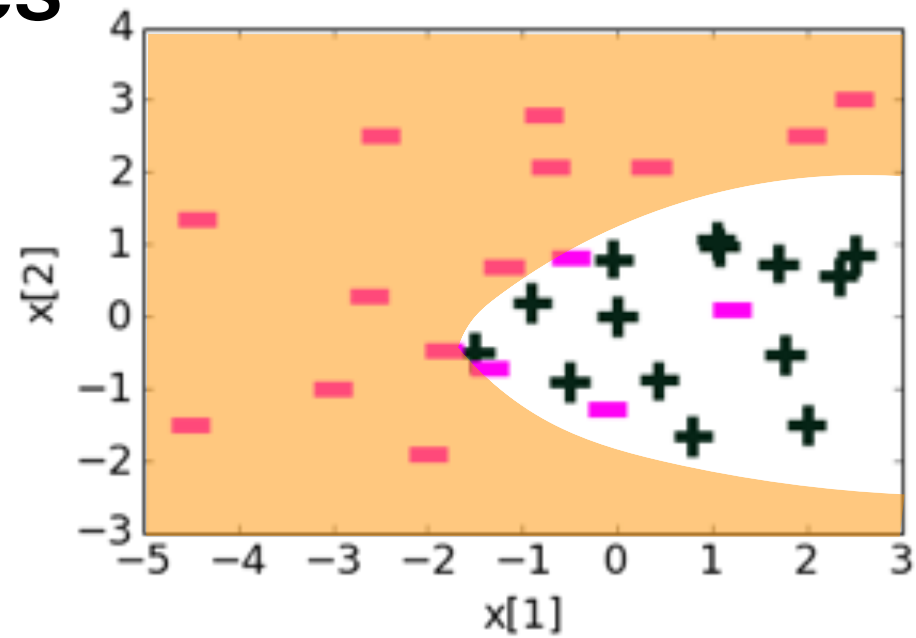


Feature	Value	Coefficient
$h_0(x)$	1	0.23
$h_1(x)$	x[1]	1.12
$h_2(x)$	x[2]	-1.07

- Simple **regression** models had **smooth** predictors
- Simple **classifier** models have **smooth** decision boundaries

Adding quadratic features



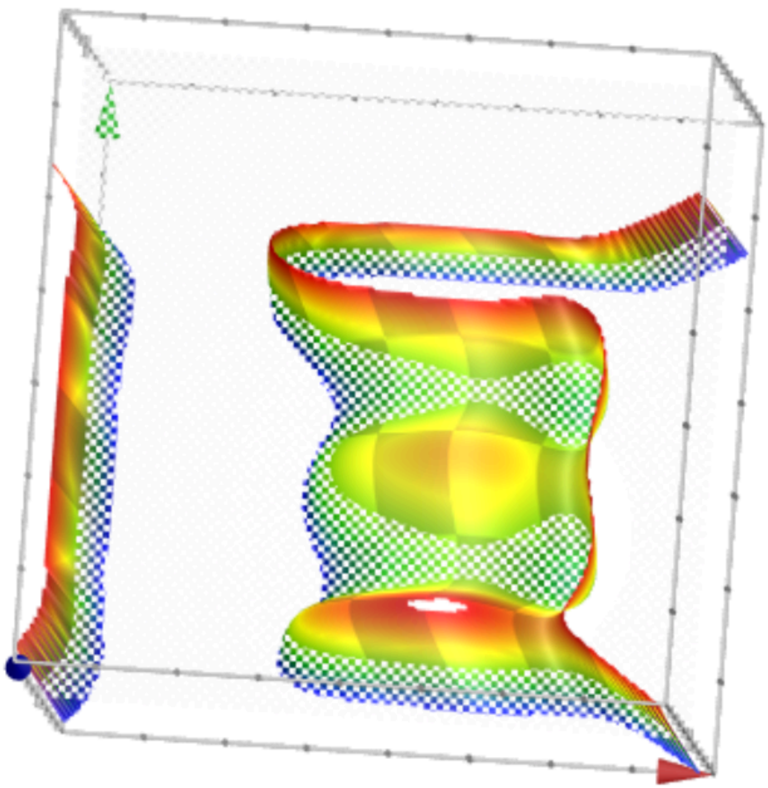


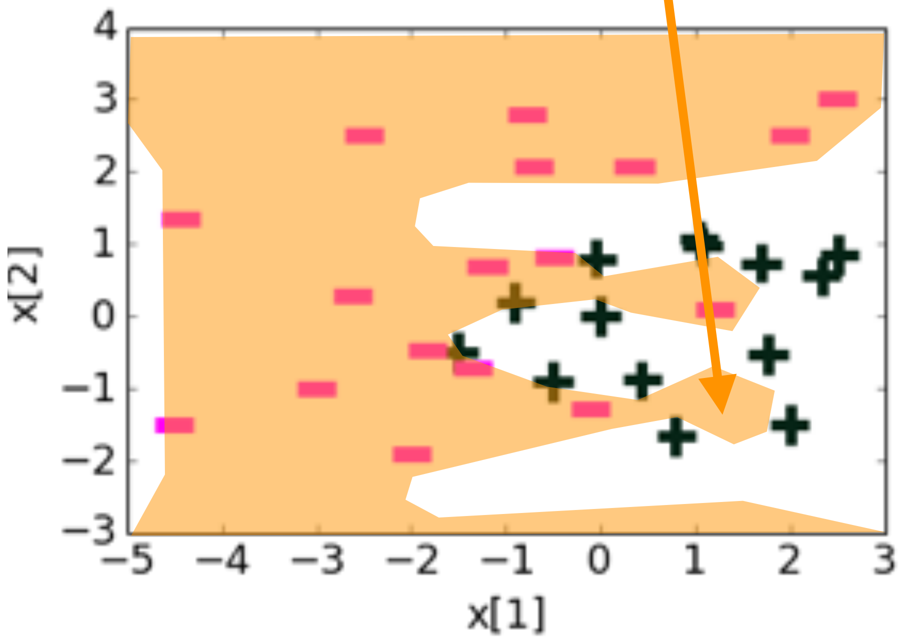
Feature	Value	Coefficient
$h_0(x)$	1	1.68
$h_1(x)$	x[1]	1.39
$h_2(x)$	x[2]	-0.59
$h_3(x)$	$(x[1])^2$	-0.17
$h_4(x)$	$(x[2])^2$	-0.96
$h_5(x)$	x[1]x[2]	Omitted

- Adding more features gives more complex models
- Decision boundary becomes more complex

Adding higher degree polynomial features

Overfitting leads to non-generalization





Feature	Value	Coefficient learned
$h_0(x)$	1	21.6
$h_1(x)$	x[1]	5.3
$h_2(x)$	x[2]	-42.7
$h_3(x)$	$(x[1])^2$	-15.9
$h_4(x)$	$(x[2])^2$	-48.6
$h_5(x)$	$(x[1])^3$	-11.0
$h_6(x)$	$(x[2])^3$	67.0
$h_7(x)$	$(x[1])^4$	1.5
$h_8(x)$	$(x[2])^4$	48.0
$h_9(x)$	$(x[1])^5$	4.4
$h_{10}(x)$	$(x[2])^5$	-14.2
$h_{11}(x)$	$(x[1])^6$	0.8
$h_{12}(x)$	$(x[2])^6$	-8.6

Coefficient values getting large

- Overfitting leads to very large values of

$$f(x) = w_0 h_0(x) + w_1 h_1(x) + w_2 h_2(x) + \cdots$$

Loss function: Conditional Likelihood

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

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

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

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

What does $J(w)$ look like? Is it convex?

Loss function: Conditional Likelihood

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

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

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

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

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

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

Good news: convex functions easy to optimize

One other concern... overfitting.

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

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

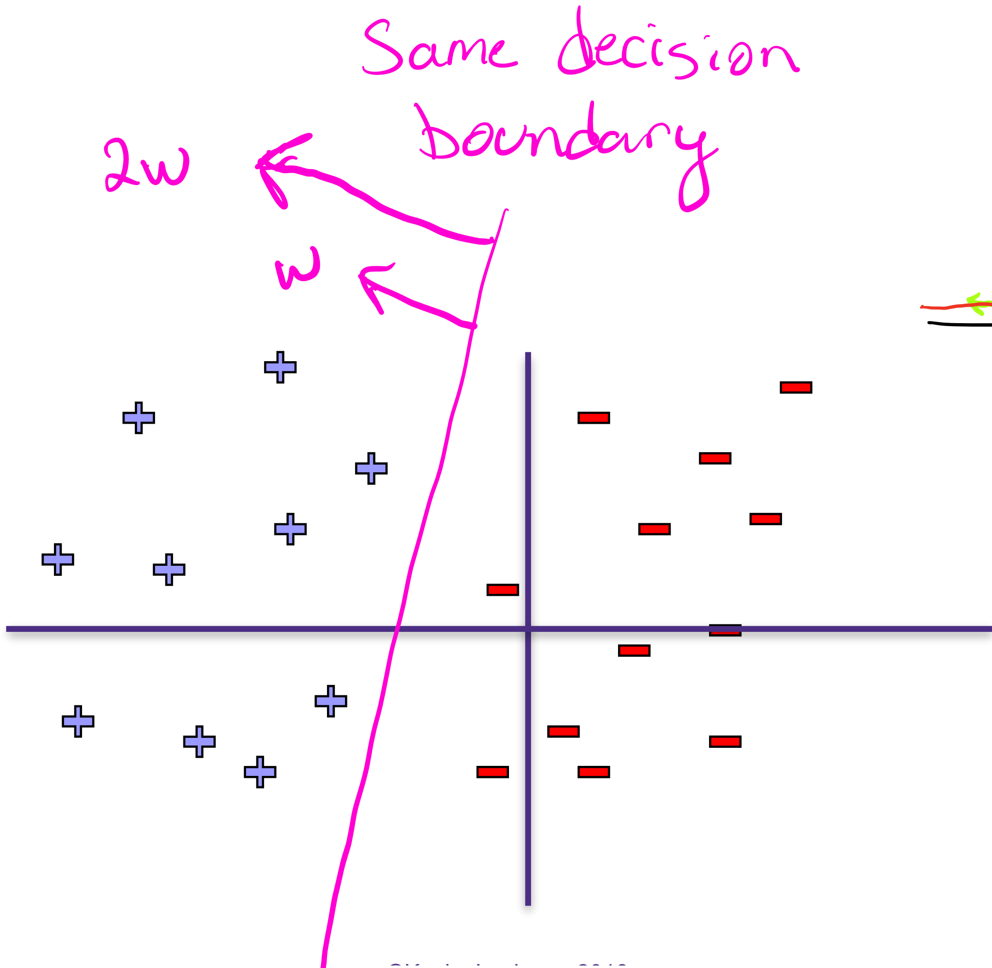
$$\begin{aligned}\widehat{w}_{MLE} &= \arg\max_w \prod_{i=1}^n P(y_i|x_i, w) \\ &= \arg\min_w \sum_{i=1}^n \log(1 + \exp(-y_i\, x_i^T w))\end{aligned}$$

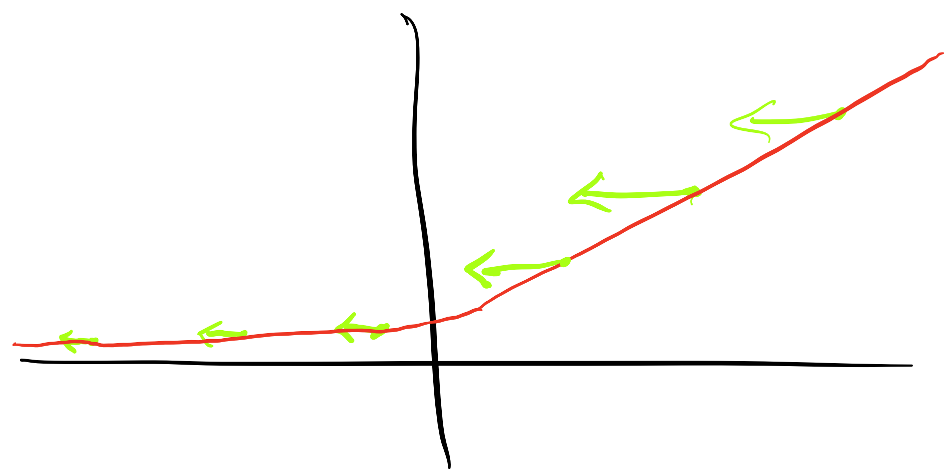
Does anyone see a situation when this minimization might overfit?

Overfitting and Linear Separability

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

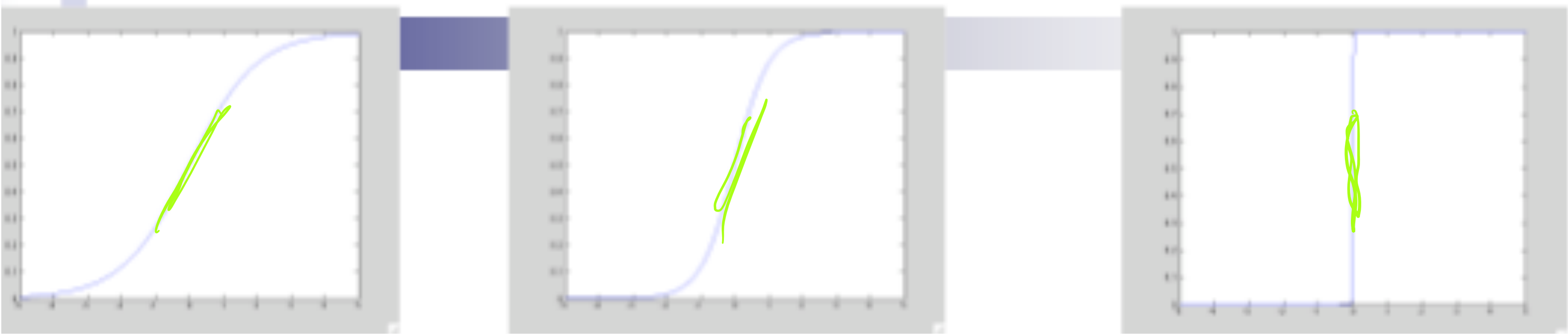
When is this loss small?





Large parameters → Overfitting

When data is linearly separable, weights $\Rightarrow \infty$



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

Overfitting

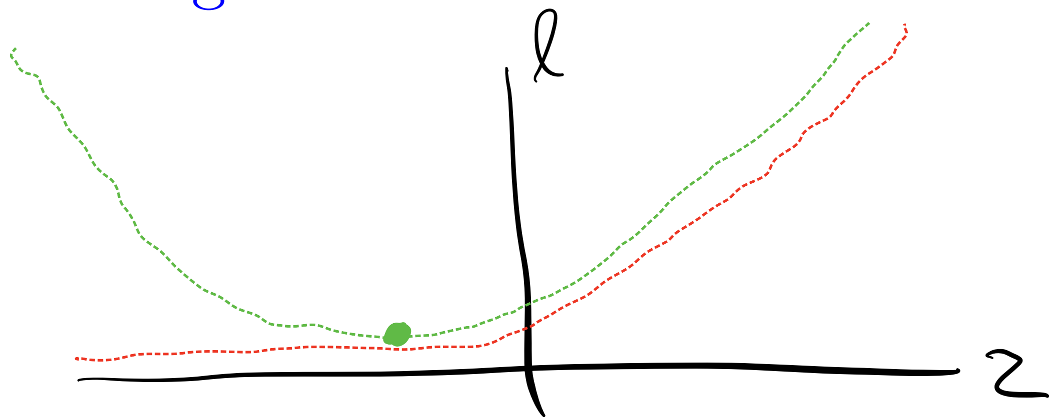
Penalize high weights to prevent overfitting?

Regularized Conditional Log Likelihood

Add a penalty to avoid high weights/overfitting?:

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

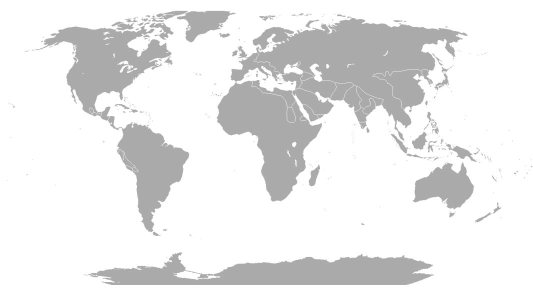
Be sure to not regularize the offset b!



How do we encode categorical data *y*?

- so far, we considered Binary case where there are two categories
- encoding y is simple: {+1,-1}

- multi-class classification predicts categorial y
- taking values in $C = \{c_1, \ldots, c_k\}$
- c_j's are called **classes** or **labels**
- examples:



Country of birth
(Argentina, Brazil, USA,...)

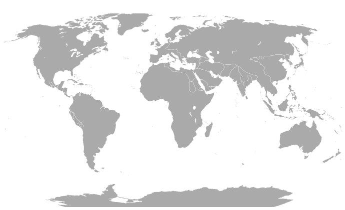
Zipcode
(10005, 98195,...)

All English words

- a **k-class classifier** predicts y given x

Embedding c_j's in real values

- for optimization we need to **embed** raw categorical c_j's into real valued vectors
- there are many ways to embed categorial data
  - True->1, False->-1
  - Yes->1, Maybe->0, No->-1
  - Yes->(1,0), Maybe->(0,0), No->(0,1)
  - Apple->(1,0,0), Orange->(0,1,0), Banana->(0,0,1)
  - Ordered sequence:
    (Horse 3, Horse 1, Horse 2) -> (3,1,2)
- we use **one-hot embedding** (a.k.a. **one-hot encoding**)
  - each class is a standard basis vector in $k-$dimension



Country of birth
(Argentina, Brazil, USA,...)

196 categories

1-hot encoding

x	$h_1(x)$	$h_2(x)$	...	$h_{195}(x)$	$h_{196}(x)$
Brazil		1			
Zimbabwe					1

196 features

Multi-class logistic regression

- data: categorical y in $\{c_1, \ldots, c_k\}$ with k categories

  we use one-hot encoding, s.t. $y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ implies that $y = c_1$

- model: linear vector-function makes a linear prediction $\hat{y} \in \mathbb{R}^k$

$$\hat{y}_i \ = \ f(x_i) \ = \ W^T x_i \quad \in \mathbb{R}^k$$

with model parameter matrix $W \in \mathbb{R}^{d \times k}$ and sample $x_i \in \mathbb{R}^d$

$$f(x_i) \ = \ \begin{bmatrix} f_1(x_i) \\ f_2(x_i) \\ \vdots \\ f_k(x_i) \end{bmatrix} \ = \ \underbrace{\begin{bmatrix} w_{1,0} & w_{1,1} & w_{1,2} & \cdots \\ w_{2,0} & w_{2,1} & w_{2,2} & \cdots \\ \vdots & & & \\ w_{k,0} & w_{k,1} & w_{k,2} & \cdots \end{bmatrix}}_{W^T} \underbrace{\begin{bmatrix} 1 \\ x_i[1] \\ \vdots \\ x_i[d] \end{bmatrix}}_{x_i} \ = \ \begin{bmatrix} w_{1,0} + w_{1,1}x_i[1] + w_{1,2}x_i[2] + \cdots \\ w_{2,0} + w_{2,1}x_i[1] + w_{2,2}x_i[2] + \cdots \\ \vdots \\ w_{k,0} + w_{k,1}x_i[1] + w_{k,2}x_i[2] + \cdots \end{bmatrix}$$

$$W = \begin{bmatrix} w[:,1] & w[:,2] & \cdots & w[:,k] \end{bmatrix}$$

- Logistic regression

2 classes

$$\mathbb{P}(y_i = -1 \,|\, x_i) = \frac{1}{1 + e^{w^T x_i}}$$

$$\mathbb{P}(y_i = +1 \,|\, x_i) = \frac{1}{1 + e^{-w^T x_i}} = \frac{e^{w^T x_i}}{1 + e^{w^T x_i}}$$

k classes

$$\mathbb{P}(y_i = c_1 \,|\, x_i) = \frac{e^{w[:,1]^T x_i}}{e^{w[:,1]^T x_i} + \cdots + e^{w[:,k]^T x_i}}$$

$$\vdots$$

$$\mathbb{P}(y_i = c_k \,|\, x_i) = \frac{e^{w[:,k]^T x_i}}{e^{w[:,1]^T x_i} + \cdots + e^{w[:,k]^T x_i}}$$

Without loss of generality setting w[:,1]=0 when $k = 2$ recovers the original binary class case

Maximum Likelihood Estimator

$$\text{maximize}_w \ \frac{1}{n} \sum_{i=1}^{n} \log(\mathbb{P}(y_i \,|\, x_i))$$

$$\text{maximize}_{w \in \mathbb{R}^d} \ \frac{1}{n} \sum_{i=1}^{n} \log\left(\frac{1}{1 + e^{-y_i w^T x_i}} \right)$$

$$\text{maximize}_{W \in \mathbb{R}^{d \times k}} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{I}\{y_i = c_j\} \log\left(\frac{e^{w[:,j]^T x_i}}{\sum_{j'=1}^{k} e^{w[:,j']^T x_i}} \right)$$

$\mathbf{I}\{y_i = j\}$ is an indicator that is one only if $y_i = j$