Midterm Friday

- **Time:** 5/2, 9:30am sharp to 10:20am (50 minutes)
  - Please be seated 5 minutes beforehand. Bring your Husky ID.
- **Locations:**
  - **CSE2 G20** (usual classroom): Sections AA, AB, AC
  - **ECE 105:** Sections AD, AE, AF
- **Topics:** Everything up to and including Lecture 8 (this lecture Classification; Logistic regression)
- **Format:** Sample exams posted
- **Cheat sheet:**
  - You may bring one 8.5x11 inch sheet of paper (can use front & back).
  - It may be handwritten ~~or typeset~~.

Classification and Logistic Regression

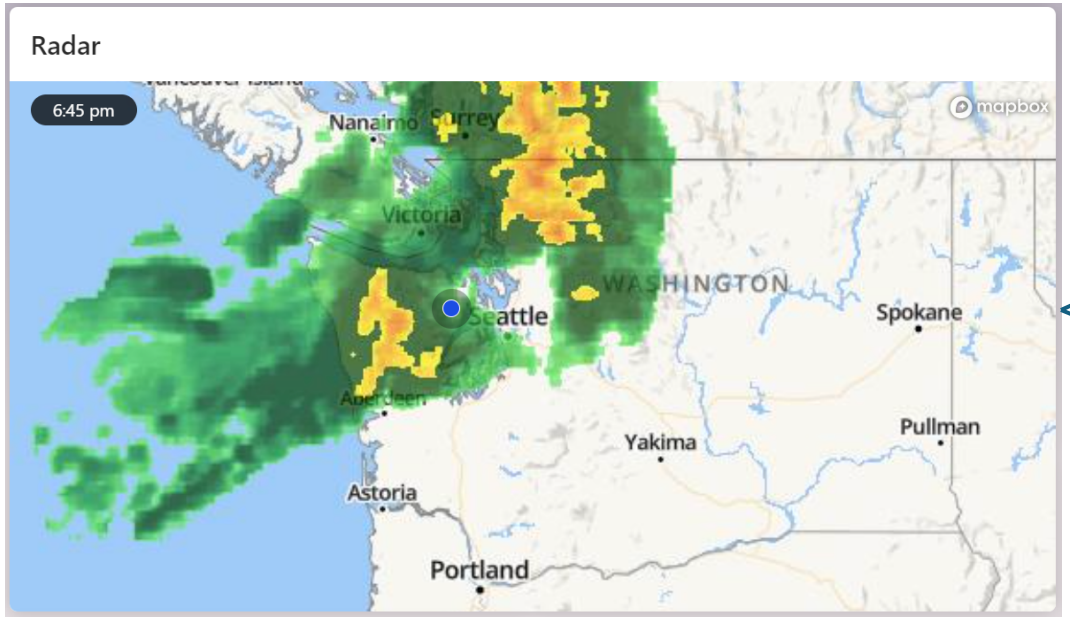
Natasha Jaques

UNIVERSITY of WASHINGTON

Regression vs classification

- Regression: Given input x, predict continuous value $y \in \mathbb{R}$
- Classification: Given input x, predict discrete value $y \in \mathcal{Y}$

x



y

Temperature: 62F

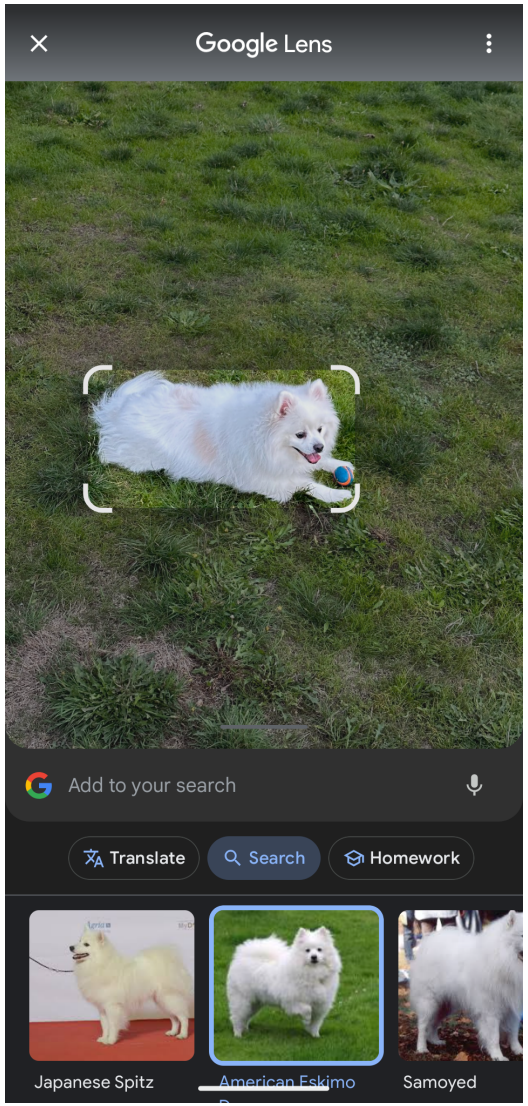
regression

classification

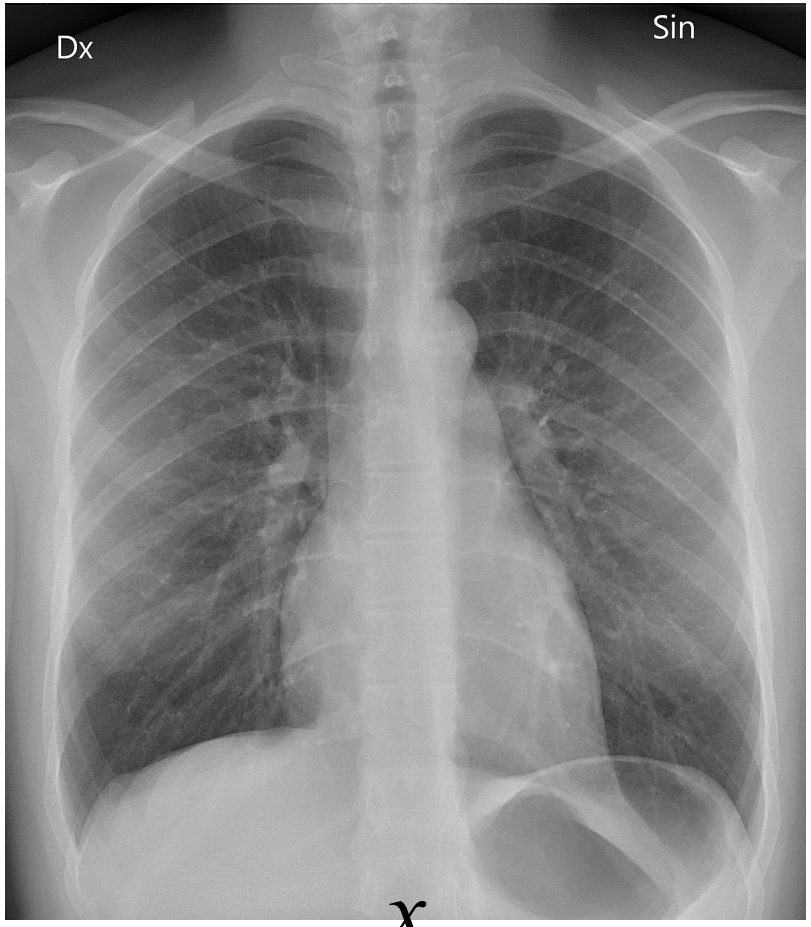
Classification problems are ubiquitous

x

y



Classification problems are ubiquitous



x

Healthy
Pneumonia
Pneumothorax
Pleural Effusion
…

y

Text generation as classification

Enter text:
One, two,

One 3198 , 11 two 734 , 11

3198 11 734 11

Prediction

#	probs	next token ID	predicted next token
0	39.71%	1115	three
1	16.97%	290	and
2	7.55%	734	two
3	3.76%	1440	four
4	2.76%	393	or
5	2.18%	1936	five
6	1.57%	530	one
7	1.43%	345	you
8	1.15%	257	a
9	0.84%	3598	seven

https://huggingface.co/blog/alonsosilva/nexttokenprediction

Text generation as classification

Enter text:

One, two, three

3198 11 734 11 1115

Prediction

#	probs	next token ID	predicted next token
0	54.42%	11	,
1	5.45%	1399	...
2	4.82%	13	.
3	4.51%	290	and
4	2.72%	986	...
5	2.51%	25	:
6	1.50%	393	or
7	1.23%	3926	...
8	0.85%	553	,"
9	0.84%	960	—

https://huggingface.co/blog/alonsosilva/nexttokenprediction

Text generation as classification

Enter text:

One, two, three,

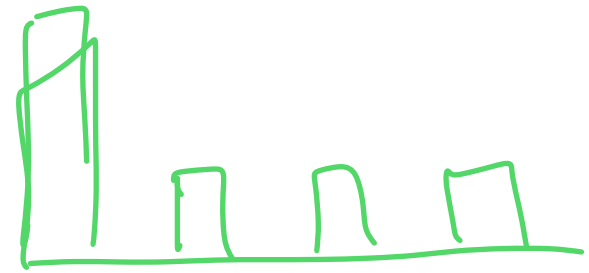
One 3198 , 11 two 734 , 11 three 1115 , 11

3198 11 734 11 1115 11

Prediction

#	probs	next token ID	predicted next token
0	46.44%	1440	four
1	7.48%	290	and
2	7.31%	1936	five
3	2.66%	393	or
4	2.54%	2237	six
5	2.09%	1115	three
6	1.86%	3863	maybe
7	1.62%	345	you
8	1.23%	257	a
9	0.92%	530	one

$P(y|\varphi)$



φ

'one, two, three'

https://huggingface.co/blog/alonsosilva/nexttokenprediction

Text generation as classification

Enter text:

One, two, three, four

3198 11 734 11 1115 11 1440

Prediction

#	probs	next token ID	predicted next token
0	50.14%	11	,
1	6.66%	13	.
2	5.91%	1399	...
3	3.15%	25	:
4	2.63%	290	and
5	2.58%	986	...
6	1.42%	3926	...
7	1.17%	553	,"
8	1.09%	960	—
9	1.08%	526	."

https://huggingface.co/blog/alonsosilva/nexttokenprediction

Classification

- Learn $f: \mathcal{X} \to \mathcal{Y}$
  - $\mathcal{X} \subset \mathbb{R}^d$: features
  - $\mathcal{Y} = \{1, \ldots, k\}$: target classes

- 0-1 loss function: $\ell(f(x), y) = \mathbf{1}\{f(x) \neq y\}$ → minimize errors

  prediction, label

- Expected loss:

$$E_{XY}[\mathbf{1}\{f(x) \neq y\}] = E_X E_{Y|X}[\mathbf{1}\{f(x) \neq y \mid X = x\}]$$

$$P(f(x) \neq y \mid X = x)$$

$$1 - P(f(x) = y \mid X = x)$$

// Expectation of indicator is just the probability of that event

$$f^*(x) = \arg\max_{y \in Y} P(Y = y \mid X = x)$$

→ ∀ x pick y that maximizes true underlying prob of y

→ BAYES OPTIMAL CLASSIFIER

Bayes-optimal classifier

$$f^*(x) = \text{argmax}_y \, P(Y = y \mid X = x)$$

unknown prob property of the true distribution

In practice, we don't know $P(Y = y \mid X = x)$, but have n i.i.d. examples:

$$x_i, y_i \Big|_{i=1}^{n}$$

training data

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

$$\hat{f}(x) = \text{argmax}_y \sum_{i=1}^{n}$$

Bayes-optimal classifier

$$f^*(x) = \text{argmax}_y \, P(Y = y \mid X = x)$$

In practice, we don't know $P(Y = y \mid X = x)$, but have n i.i.d. examples:

$$x_i, y_i{}_{i=1}^{n}$$

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

$$\hat{f}(x) = \text{argmax}_y \frac{\sum_{i=1}^{n} 1\{x_i = x, y_i = y\}}{\sum_{i=1}^{n} 1\{x_i = x\}}$$

$$P(y \mid x) = \frac{P(x, y)}{P(x)}$$

If $\mathscr{X}$ is continuous, we need a <u>model</u> to explain observations!

Maximum likelihood estimation for classification

$$f^*(x) = \text{argmax}_y \, P(Y = y \mid X = x)$$

$$\hat{f}_w(x) = \text{argmax}_y \, P_w(Y = y \mid X = x)$$

General MLE procedure:

1. Parameterize $P_w(Y = y \mid X = x)$ as a function of w
2. Learn w on i.i.d. training data $x_i, y_i{}_{i=1}^n$

max likelihood of training data

$$\hat{w}_{\text{MLE}} = \text{argmax}_w \prod_{i=1}^n P_w(y_i \mid x_i)$$

$$= \text{argmax}_w \sum_{i=1}^n \log P_w(y_i \mid x_i)$$

Modeling conditional probabilities

$$\hat{w}_{\text{MLE}} = \text{argmax}_w \sum_{i=1}^{n} \log P_w(y_i \mid x_i)$$

assuming $y \sim$ Gaussian

- Recall linear regression: $P_w(Y = y \mid X = x) = \frac{1}{2\pi} e^{y - w^\top x^2/2}$
  - Prediction: $E[Y \mid X = x] = w^\top x$

why not do this!

we want all prob mass on 0 or 1

if $y = 0$ cat or $y = 1$ dog

Modeling conditional probabilities

$$\hat{w}_{\text{MLE}} = \text{argmax}_w \sum_{i=1}^{n} \log P_w(y_i \mid x_i)$$

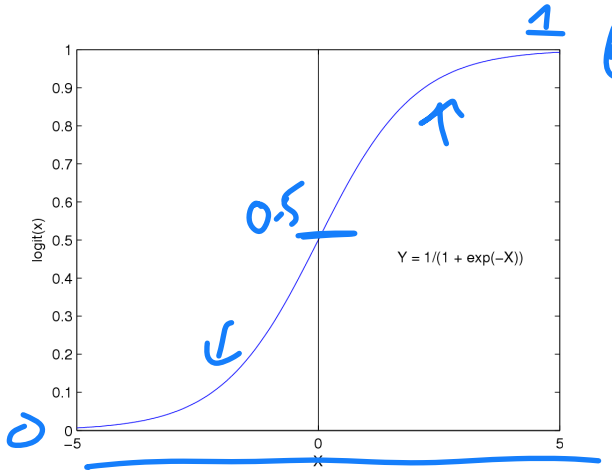
- Recall linear regression: $P_w(Y = y \mid X = x) = \frac{1}{2\pi} e^{y - w^\top x^2/2}$
  - Prediction: $E[Y \mid X = x] = w^\top x$

- Logistic regression uses a model specialized for (binary) classification:

$$\boxed{P(Y = 1 \mid X = x) = \frac{1}{1 + e^{-w^\top x}}}$$

$$P(Y = 0 \mid X = x) = \frac{e^{-w^\top x}}{1 + e^{-w^\top x}}$$

pass $w^T x$ through sigmoid func



← sigmoid

useful, widely used easy to opt..

to predict: round(sigmoid($w^T x$))

Understanding the sigmoid

learn by fitting data to max likelihood

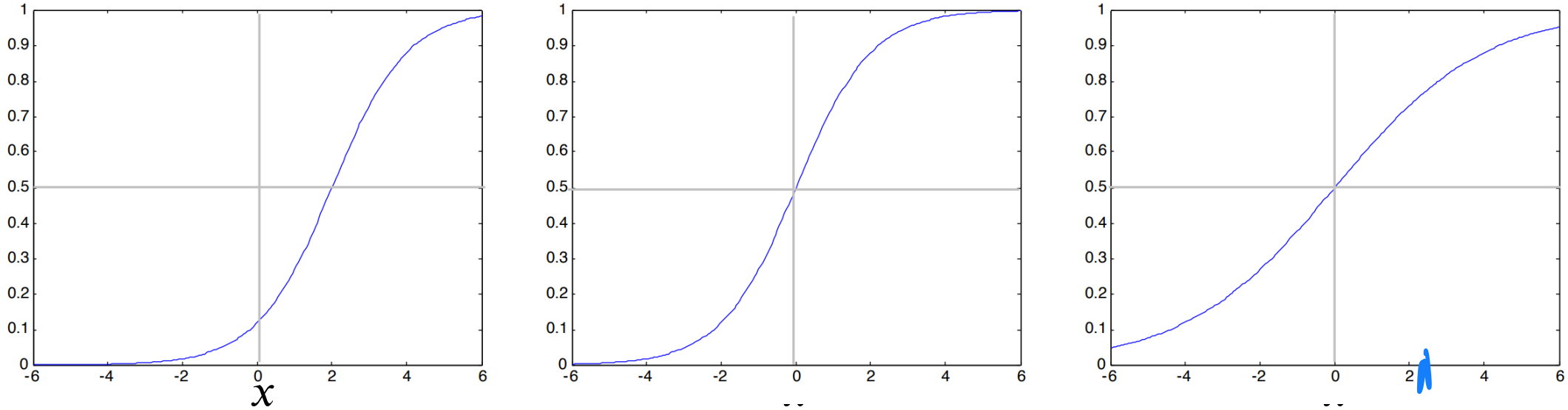
$$P(Y=1 \mid X=x) = \frac{1}{1+e^{-w^\top x}} = \sigma(w^\top x)$$

$$\sigma w_0 + \sum_k w_k x_k = \frac{1}{1+e^{-w_0+\sum_k w_k x_k}}$$

$\sigma(x \cdot 2)$ $\sigma(x)$ $\sigma(0.5\ x)$

$w_0 = -2,\ w_1 = 1$ $w_0 = 0,\ w_1 = 1$ $w_0 = 0,\ w_1 = 0.5$

$\sigma(w^\top x)$



Sigmoid for binary classification

where is the decision boundary?, where I switch from predicting 0 → 1

- What's the shape of the decision rule $P(Y = 1 \mid X) \geq P(Y = 0 \mid X)$?

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

$$P(y=0 \mid x) = \frac{e^{-w^T x}}{1+e^{-w^T x}}$$

$$\frac{P(y=1 \mid x)}{P(y=0 \mid x)} = \frac{1}{e^{-w^T x}}$$

$$\log(\ \text{"}\ ) = \log \frac{1}{e^{-w^T x}} = \boxed{w^T x}$$

→ scalar

$w^T \in 1 \times d$

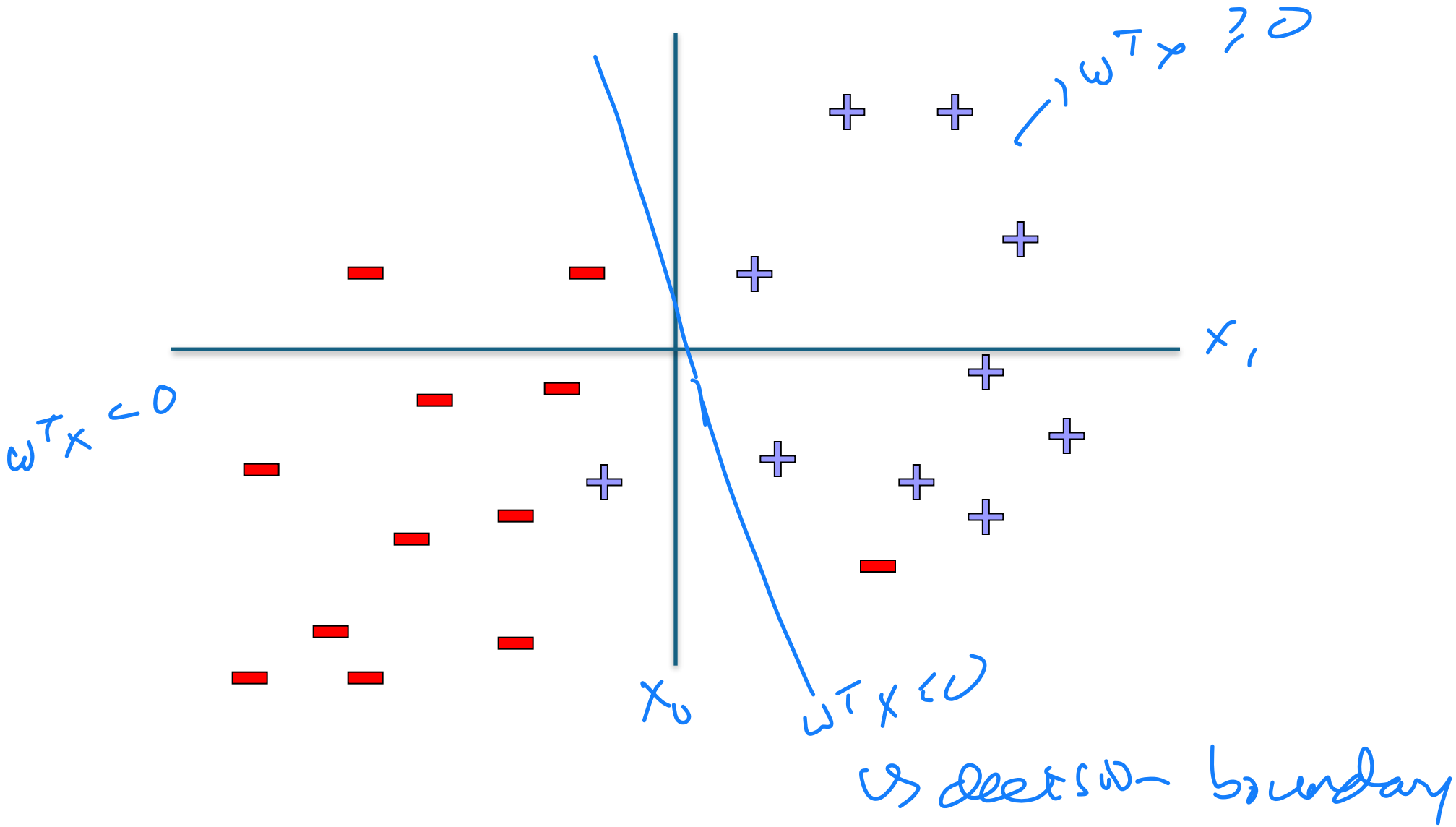
$x \in d \times 1$

Decision rule: is $w^T x \geq 0$?

Yes: predict $y=1$

No: predict $y=0$

Logistic regression – a linear classifier



Logistic regression for binary classification

switched this for notation convenience

$$\hat{w}_{\text{MLE}} = \text{argmax}_w \sum_{i=1}^{n} \log P_w(y_i \mid x_i)$$

- We have i.i.d. training data $x_i, y_i{}_{i=1}^{n}$, $x_i \in \mathbb{R}^d$, $y_i \in \{-1,1\}$ $\{0,1\}$
- $P_w y_i \mid x_i = \sigma y_i w^\top x_i = 1/(1 + \exp - y_i w^\top x_i)$
- $\ell_i(w) = -\log P_w y_i \mid x_i = \log(1 + \exp - y_i w^\top x_i)$

$y_i = -1$
$\hat{y}_i = -1$ $\to 1$

$$\boxed{\hat{w}_{\text{MLE}} = \text{argmin}_w \sum_{i=1}^{n} \log(1 + \exp - y_i w^\top x_i)}$$

$y_i = 1$
$\hat{y}_i = 1$ $\to 1$

logistic loss

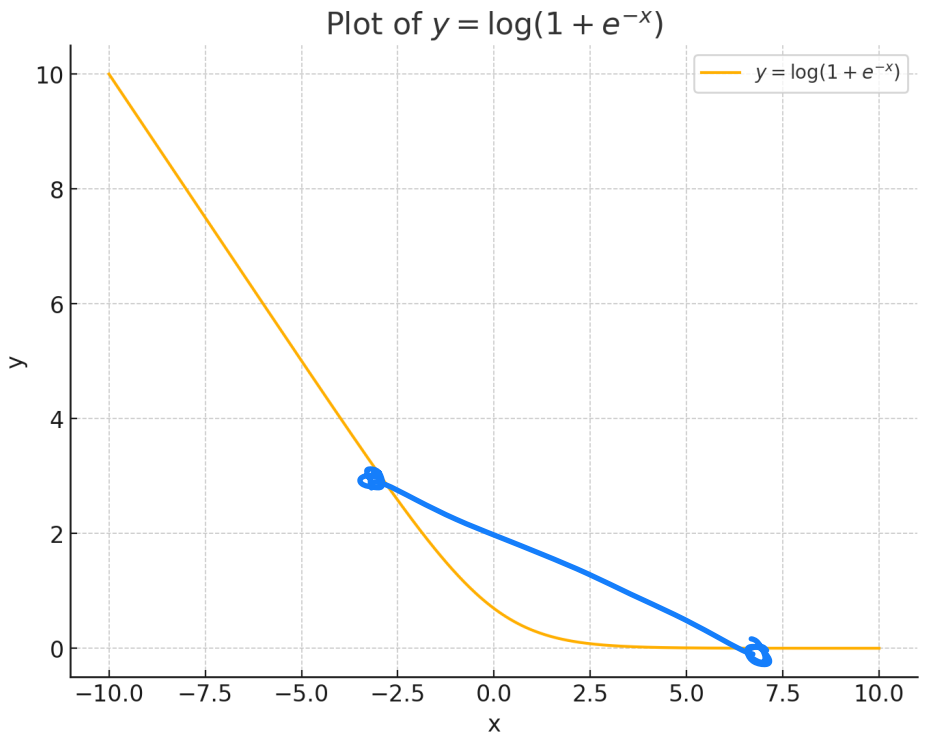
Computing the logistic regression solution

$$\hat{w}_{\text{MLE}} = \text{argmin}_{w} \sum_{i=1}^{n} \log(1 + \exp - y_i w^\top x_i)$$

Computing the logistic regression solution

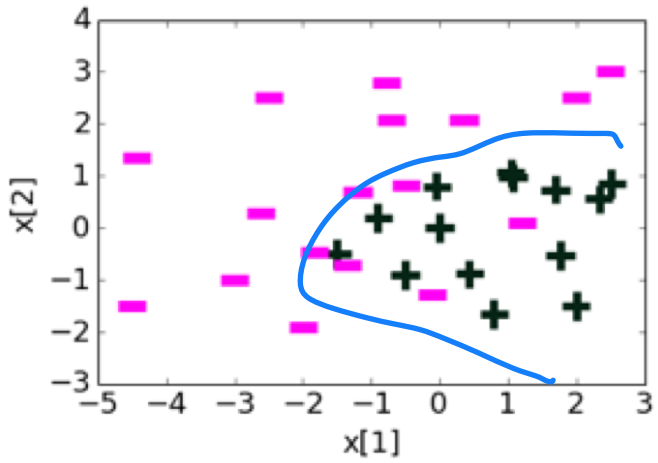
$$\hat{w}_{\text{MLE}} = \text{argmin}_{w} \sum_{i=1}^{n} \log(1 + \exp - y_i w^\top x_i)$$

- No closed form…
- But, this is a smooth convex problem!
- Optimize via gradient descent



Example: Adding more polynomial features

- $x \in \mathbb{R}^2,\ y \in \{-1, +1\}$
- Features: Polynomials
- Model: Linear on polynomial features

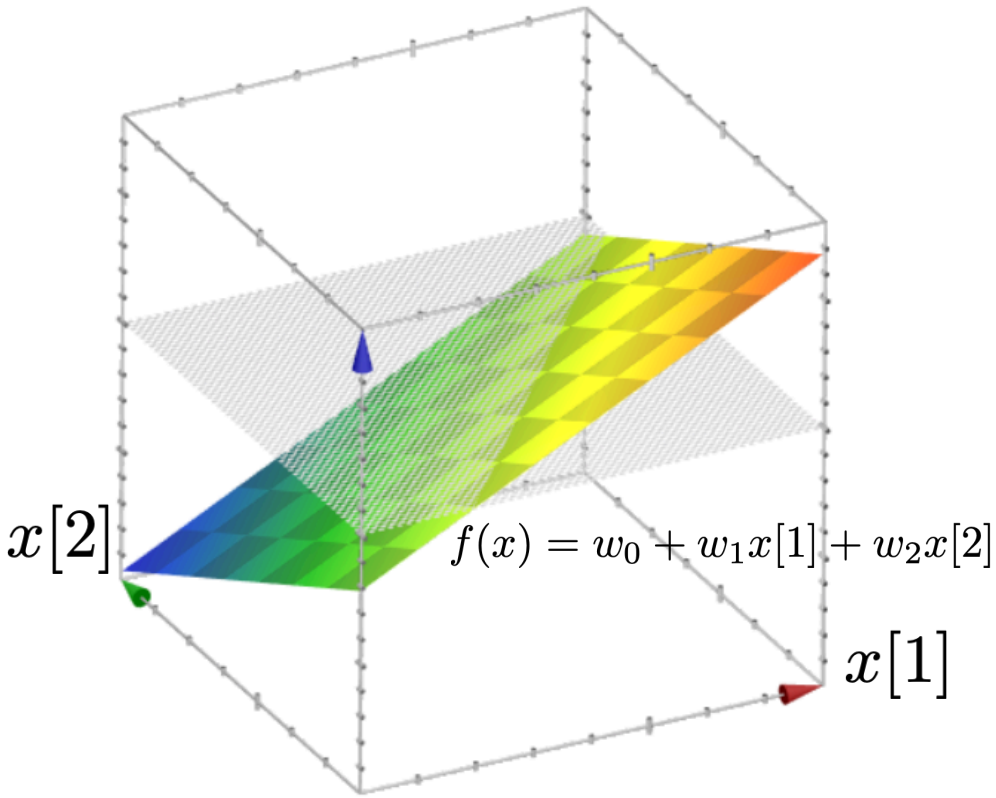


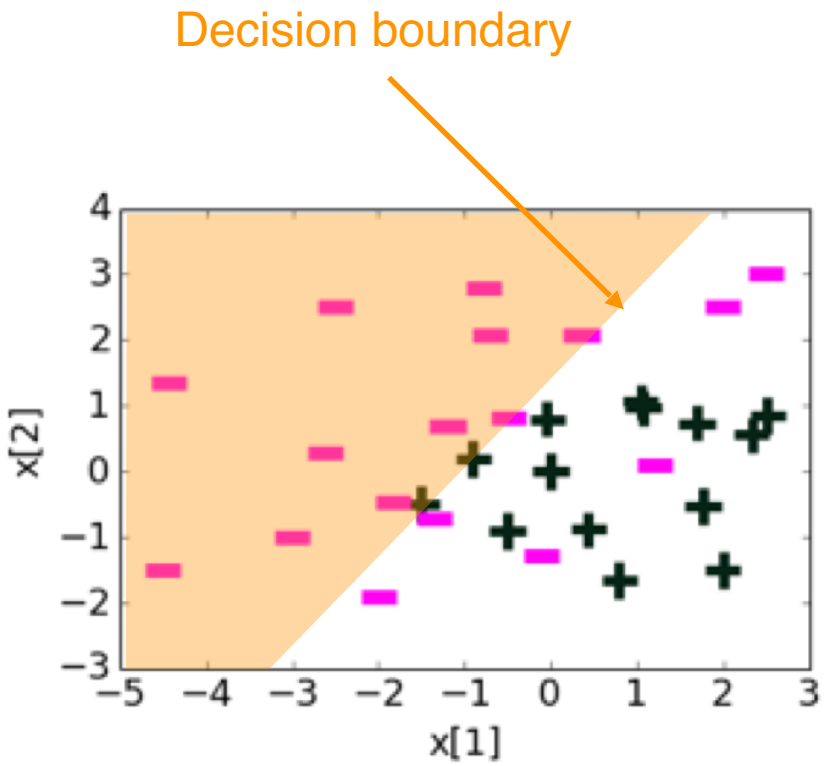
Polynomial features

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

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

Linear features

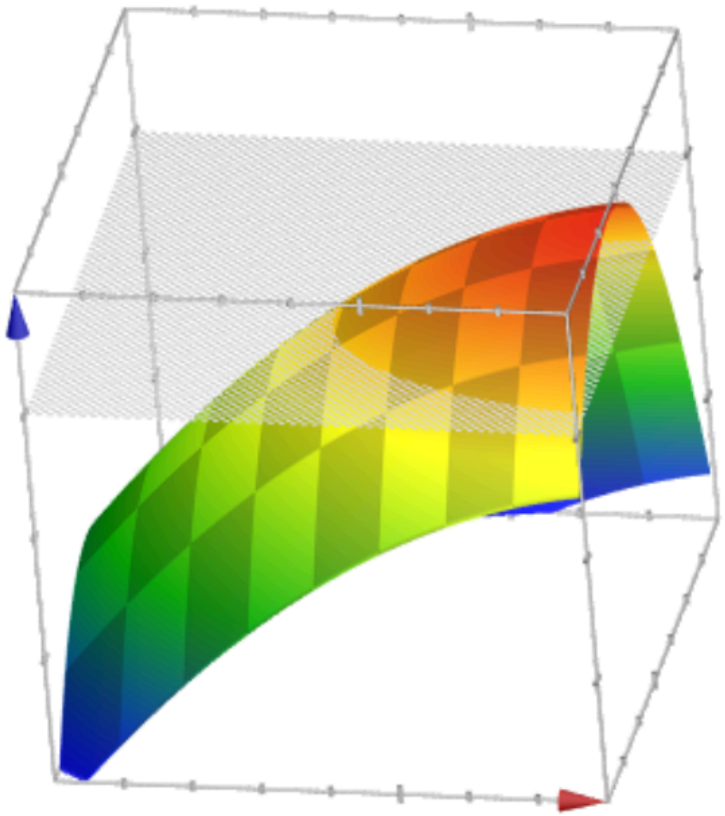


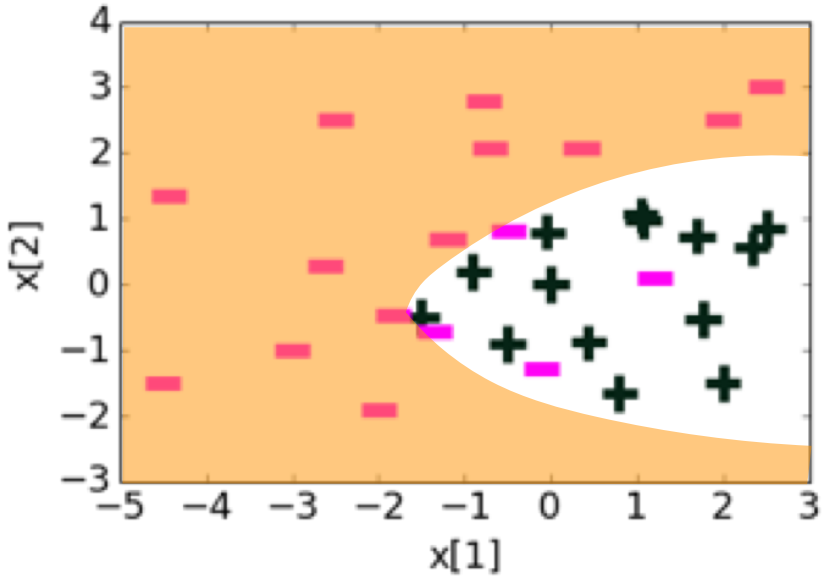


Feature	Value	Coefficient
$h_0(x)$	1	0,23
$h_1(x)$	x[1]	1,12
$h_2(x)$	x[2]	-1,07

- Simple regression models → smooth predictors
- Simple classifier models → smooth decision boundaries

Quadratic features

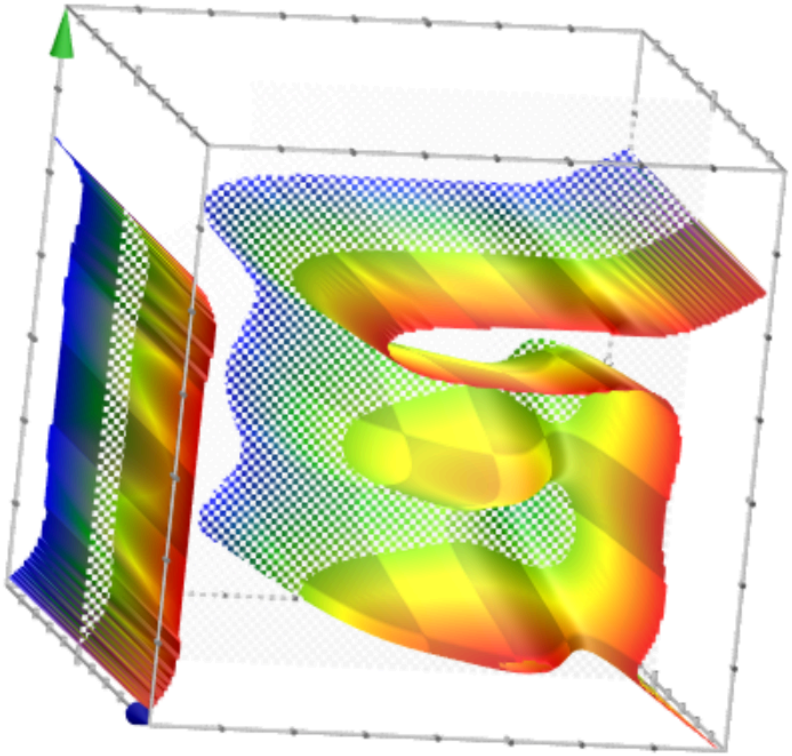


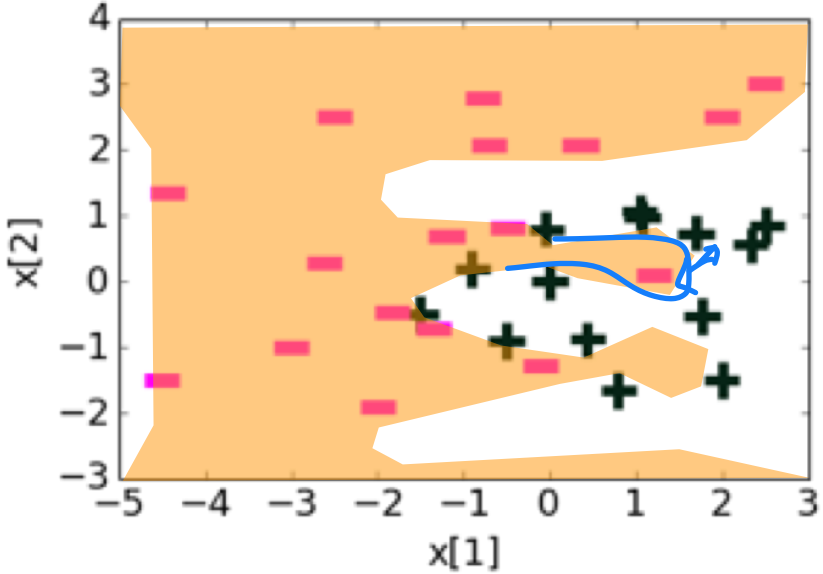


Feature	Value	Coefficient
$h_0(x)$	1	1,68
$h_1(x)$	x[1]	1,39
$h_2(x)$	x[2]	-0,59
$h_3(x)$	$(x[1])^2$	-0,17
$h_4(x)$	$(x[2])^2$	-0,96
$h_5(x)$	x[1]x[2]	Omitted

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

Higher-degree polynomial features





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

- Overfitting leads to poor generalization

Overfitting

$$\hat{w}_{\text{MLE}} = \text{argmin}_w \sum_{i=1}^{n} \log(1 + \exp(-y_i w^\top x_i))$$

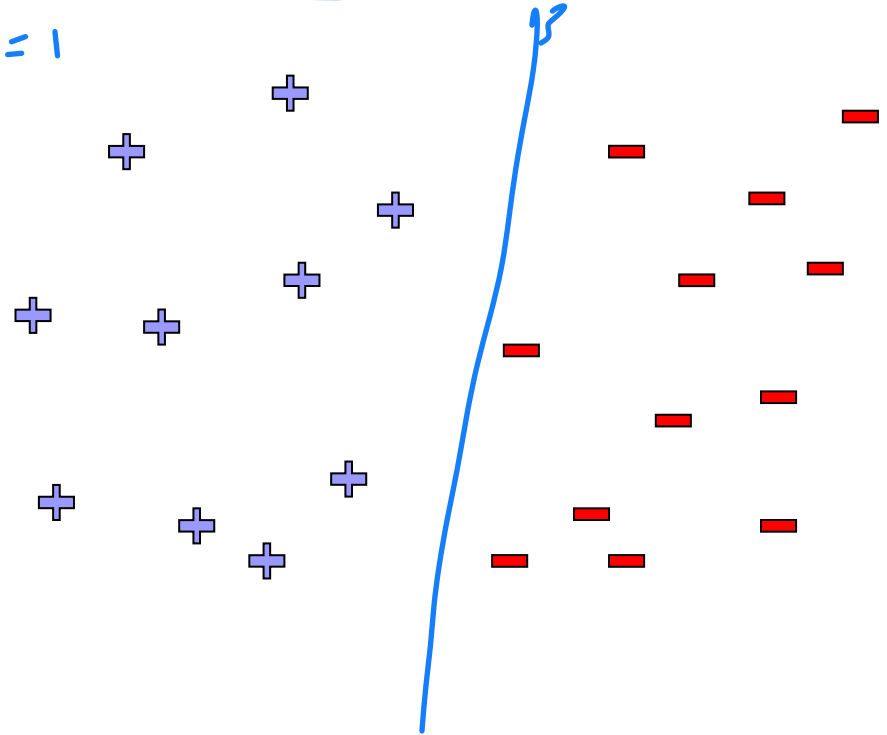
$y \in \{-1, 1\}$

if we've classified x_i correctly, then can further minimize loss by increasing w.

- When would this badly overfit?

$w^T x \geq 0 \rightarrow$ predict $y = 1$

Try to get $w^T x$ to match sign(y)

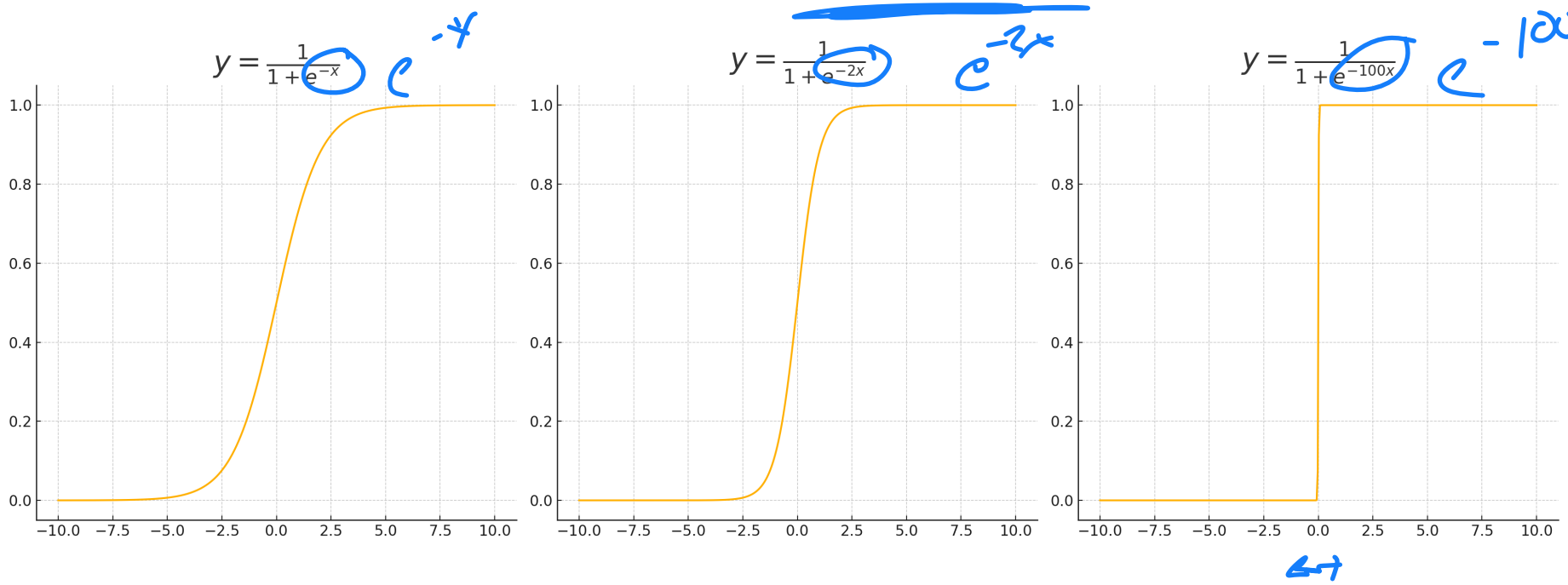


Overfitting and linear separability

$$\hat{w}_{\text{MLE}} = \text{argmin}_w \sum_{i=1}^{n} \log(1 + \exp - y_i w^\top x_i)$$

large w → high variance (small change in the input leads to a drastic change in prediction

- When data is linearly separable, $\|w\| \to \infty$



Regularized logistic regression

$$\hat{w}_{\text{reg}} = \text{argmin}_w \sum_{i=1}^{n} \log(1 + \exp - y_i w^\top x_i)$$

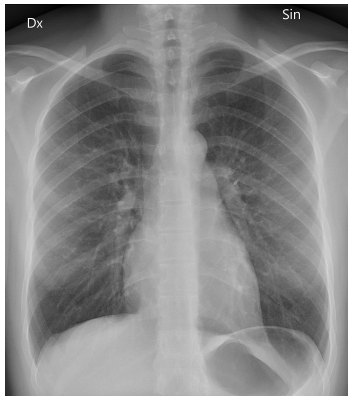
$+\lambda \|w\|_2^2$ Ridge

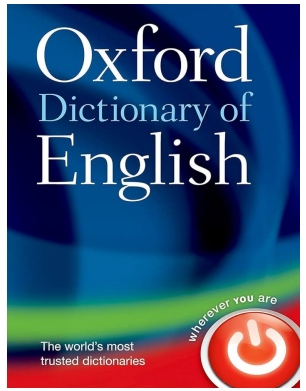
$\lambda \|w\|_1$ Lasso

prevents $\|w\| \to \infty$
(when linearly separable)
↳ for data & model class

Multi-class classification

- So far: binary $y \in -1, +1$
- In general: $y \in \{c_1, c_2, \ldots, c_k\}$
- c_j's are called <u>classes</u> or <u>labels</u>





- A <u>k-class classifier</u> predicts y given x

Encoding categorical labels c_j

5 classes $k=5$

labels $\begin{bmatrix} 3 \\ 2 \\ \vdots \\ \vdots \end{bmatrix} \rightarrow \begin{matrix} [0\,0\,0\,1\,0] \\ [0\,0\,1\,0\,0] \end{matrix}$

- For optimization, we need to embed raw c_j into real-valued vectors
- We typically use one-hot embeddings (a.k.a. one-hot encodings)
  - Each class is a standard basis vector in $\mathbb{R}^k$

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	0	1	0	0	0
Zimbabwe	0	0	0	0	1

196 features

0 0 1
0 1 0
1 0 0

Multi-class logistic regression (aka softmax classification)

features

- Data: features $x_i \in \mathbb{R}^d$, categorical $y \in \{c_1, \ldots c_k\}$
- One-hot encoding $\in \mathbb{R}^k$ s.t. $y = [1,0,0,\ldots]$ implies $y = c_1$
- Model: linear prediction $\hat{y}_i = f x_i = \text{softmax}(w^\top x) \in \mathbb{R}^k$
- Parameter matrix w $\in \mathbb{R}^{d \times k}$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

y true label

$$\begin{bmatrix} \text{---} & w_1^T & \text{---} \\ \text{---} & w_2^T & \text{---} \\ & \vdots & \\ \text{---} & w_k^T & \text{---} \end{bmatrix} \begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} 3.6 \\ -4.2 \\ \\ \\ \end{bmatrix} \xrightarrow{\text{softmax}} \begin{bmatrix} 0.4 \\ 0.3 \\ \vdots \\ \\ \end{bmatrix}$$

$W^T \in \mathbb{R}^{k \times d}$ $\quad x_i \in \mathbb{R}^{d \times 1}$ $\quad$ logits $\in \mathbb{R}^k$ $\quad \hat{y} \in \mathbb{R}^k$

normalized

sum = 1

$p(y = c_i \mid x) \geq 0$

Softmax classification

without loss of generality, set k = 2, $w_1 = 0$

loss

output y

why softmax?

$\begin{bmatrix}1\\10\\2\end{bmatrix} \to \begin{bmatrix}\text{small}\\\text{big}\\\text{small}\end{bmatrix} \to$

$\sim \begin{bmatrix}0\\1\\0\end{bmatrix}$

$\begin{bmatrix}2\\2\\2\end{bmatrix} \to \begin{bmatrix}0.33\\0.33\\0.33\end{bmatrix}$

2 classes

sigmoid

k classes

Conditional probabilities

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

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

equivalent if k = 2

$p(y=1 \mid x) = 1 - p(y=-1 \mid x)$

$$P(y = c_j \mid x) = \frac{e^{w_j^\top x}}{\sum_{j'} e^{w_{j'}^\top x}}$$

softmax

MLE

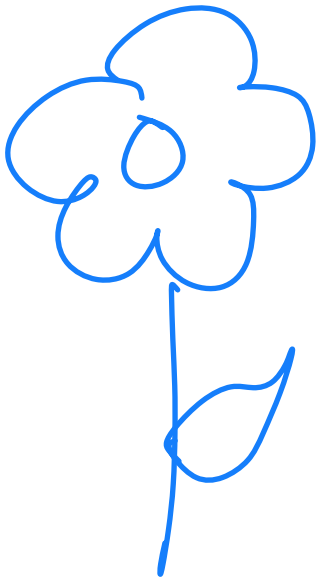
$$\text{argmax}_w \sum_{i=1}^{n} \log \frac{1}{1 + e^{-y_i w^\top x_i}}$$

This is LLMs

$$\text{argmax}_w \sum_{i=1}^{n} \sum_{j=1}^{k} 1\{y_i = c_j\} \log \left[\frac{e^{w_j^\top x_i}}{\sum_{j'=1}^{k} e^{w_{j'}^\top x_i}}\right]$$

binary cross entropy

x_1



x_2

$$\begin{bmatrix} 0 \\ 0.73 \\ 0.66 \end{bmatrix}$$

$y_1 = [0\ 0\ 1]$

x_2

$y_2 = [0\ 1\ 0]$

x_1

$\{x_i, \underline{y_i}\}$

(linear)

Regression and classification

model

$w^T x$

regression $\varepsilon \sim N(0, I)$

classification sigmoid

- ML paradigm: define prediction $f_w(x)$ and loss $\ell(f_w(x), y)$

data

- Then optimize:

$$\hat{w} = \text{argmin}_w \sum_{i=1}^{n} \ell(f_w(x_i), y_i)$$

pick model

pick loss

min loss on training data

⋮

check val err & select hparams

- Squared error loss: $\ell(f_w(x), y) = (y - f_w(x))^2$
- Logistic loss: $\ell(f_w(x), y) = \log(1 + \exp(-y f_w(x)))$

Regression and classification

- Can we treat classification as a regression problem?

Yes

bulldog → $c_1 = 1$

poodle $c_2 = 2$

→ regress to c → round to a valid label

Not a good idea

→ no smooth relationship b/w label values

- Can we treat regression as a classification problem?

Yes

buckets / bins

[0-10] [11-20] ...

c=1 c=2 ...

is it a good idea?

→ loss of precision

sometimes ~~yes~~ → be easier to opt.

→ kaggle competition

→ "distribution KL"

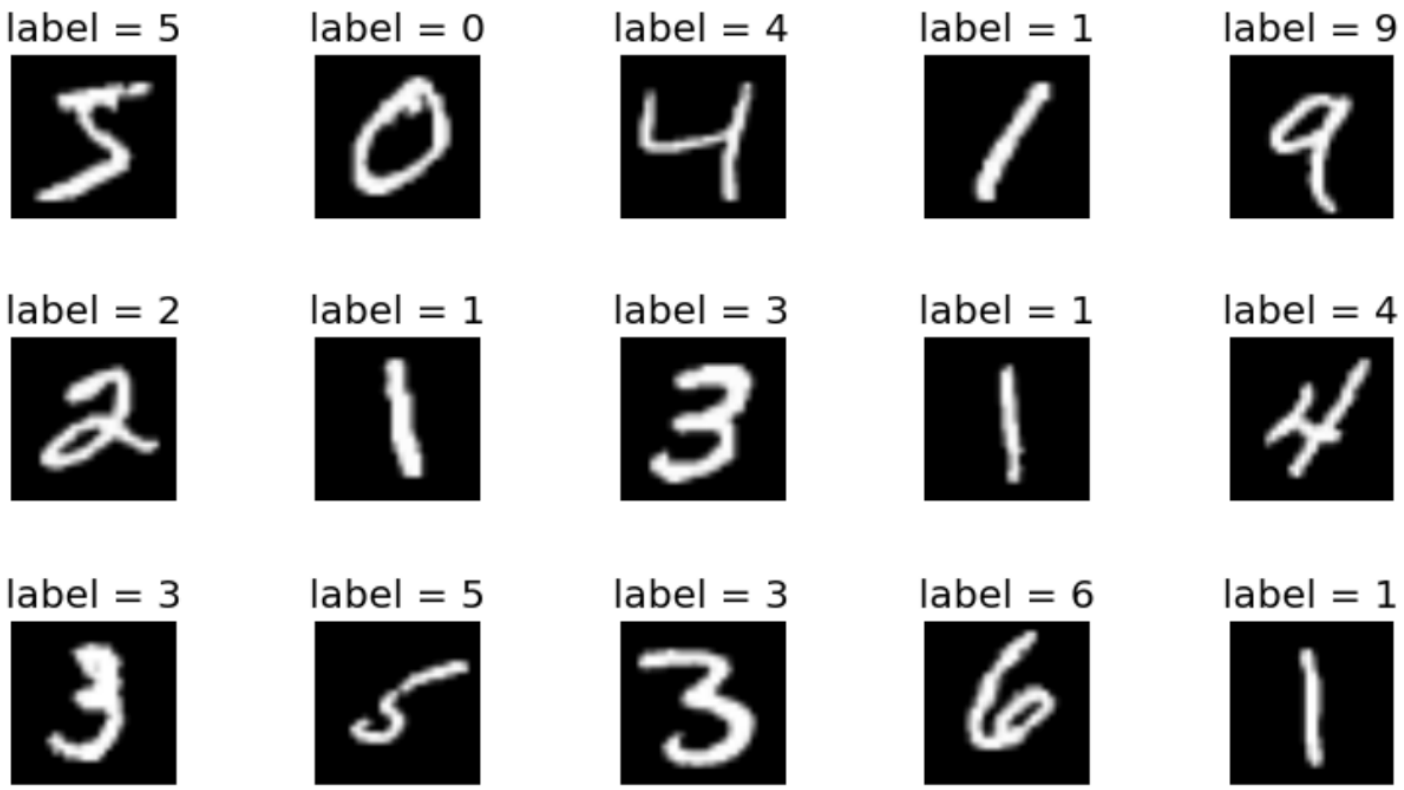
Regression and classification

Temperature: 62F

Regression and classification

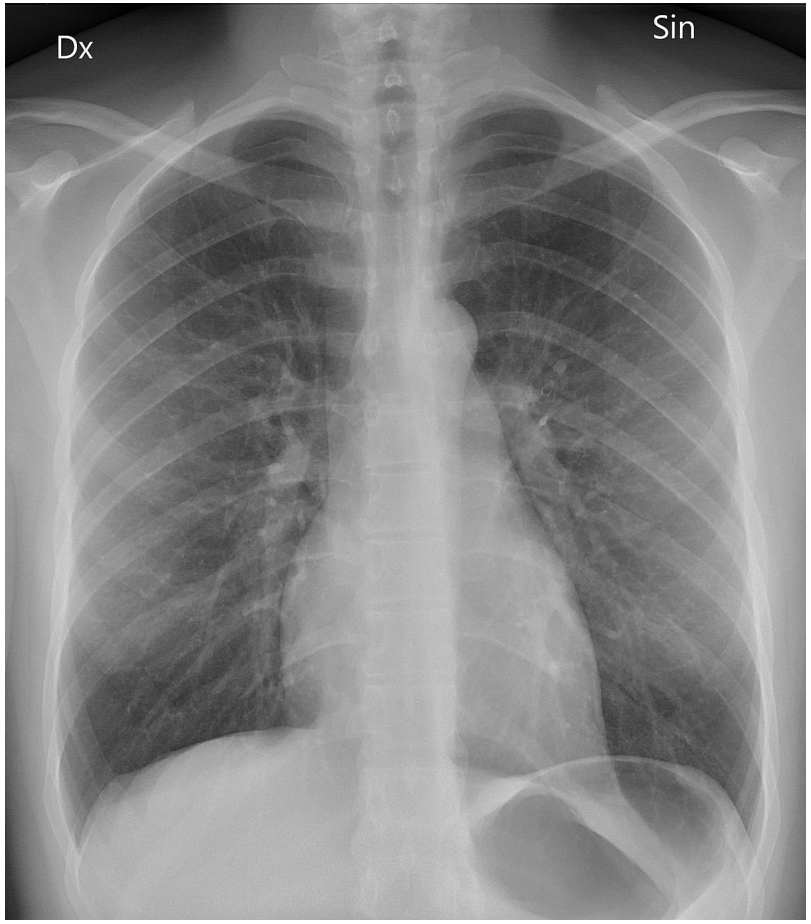
98105

Regression and classification



Multi-class vs multi-label classification

what if multiple labels can be true?



Healthy
Pneumonia
Pneumothorax
Pleural Effusion
...

↳ no softmax
↳ multiple sigmoid outputs

Summary

- Classification problems (where y is categorical) are everywhere
- Regression losses are not typically appropriate
- Logistic regression: model conditional probability P(y|x) as sigmoid (then apply usual MLE machinery)
- Softmax classification: generalized to $k > 2$ classes
- Regularization is still important

Recap: The ML pipeline

1. Define the **task** (what type of data, what type of eval metrics?)
2. Collect and preprocess **data** → feature engineering → polynomial degree
3. Choose **model** family/parameterization $w^T x$
4. Choose **training loss** → sigmoid
4. For each choice of hyperparameters:
   - **Optimize** model (minimize loss) on training data
   - **Evaluate** on validation data
5. Pick best hyperparameters according to validation performance
6. **Evaluate** final model on test data

   → held out

The ML pipeline

1. Task
2. Data
3. Model family
4. Training loss
4. Optimize
5. Pick hyperparameters
6. Evaluate