Y-DATA
SCHOOL OF DATA SCIENCE

Linear & Logistic Regression

Noa Lubin & Lior Sidi

Agenda

- Recap
- Regularization
- Linear Regression
- Logistic Regression
- Code
- Summary

Y-DATA
SCHOOL OF DATA SCIENCE

Recap

Gradient Descent

The idea of gradient descent is:

Take iterative steps to update parameters in the direction of the gradient

Previous weights

gradient

$$\underset{\text{New weights}}{w^t} \leftarrow \overset{\text{Previous weights}}{w^{t-1}} - \underset{\text{Step size / learning rate}}{\eta} \cdot \overset{\text{gradient}}{\frac{\partial \ell}{\partial w^{t-1}}}$$

New weights

Step size / learning rate

Gradient Descent Training 📝

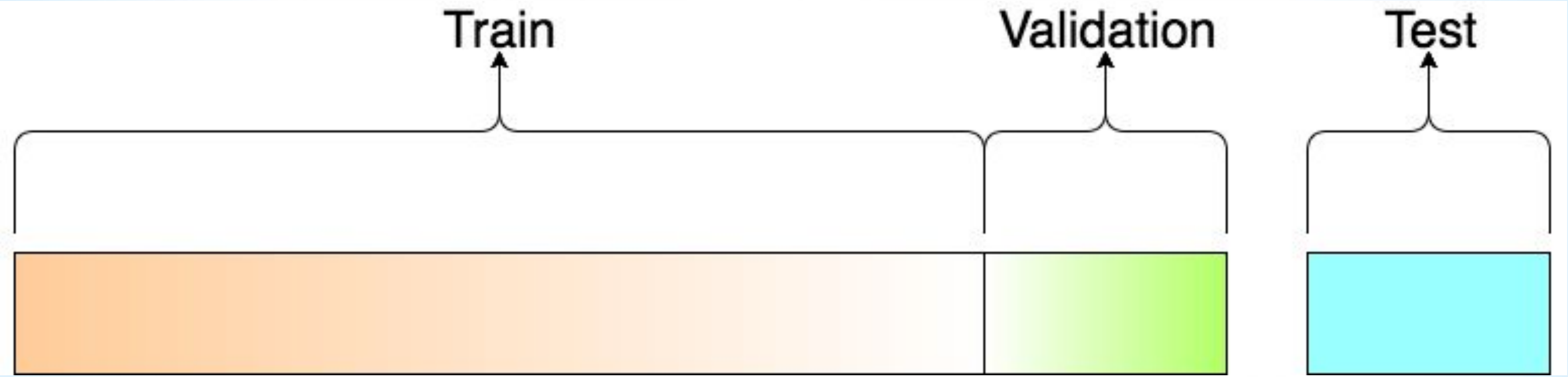
Algorithm 1 Gradient Descent Training

Input:

- Function $f(\mathbf{x}; \Theta)$ parameterized with parameters Θ.
- Training set of inputs $\mathbf{x_1}, \ldots, \mathbf{x_n}$ and desired outputs $\mathbf{y_1}, \ldots, \mathbf{y_n}$.
- Loss function L.

1: **while** stopping criteria not met **do**
2: Compute the loss $\mathcal{L}(\Theta) = \sum_i L(f(\mathbf{x_i}; \Theta), \mathbf{y_i})$ <-- slow! goes over all data.
3: $\hat{\mathbf{g}} \leftarrow$ gradients of $\mathcal{L}(\Theta))$ w.r.t Θ
4: $\Theta \leftarrow \Theta - \eta_t \hat{\mathbf{g}}$
5: **return** Θ

Reminder:



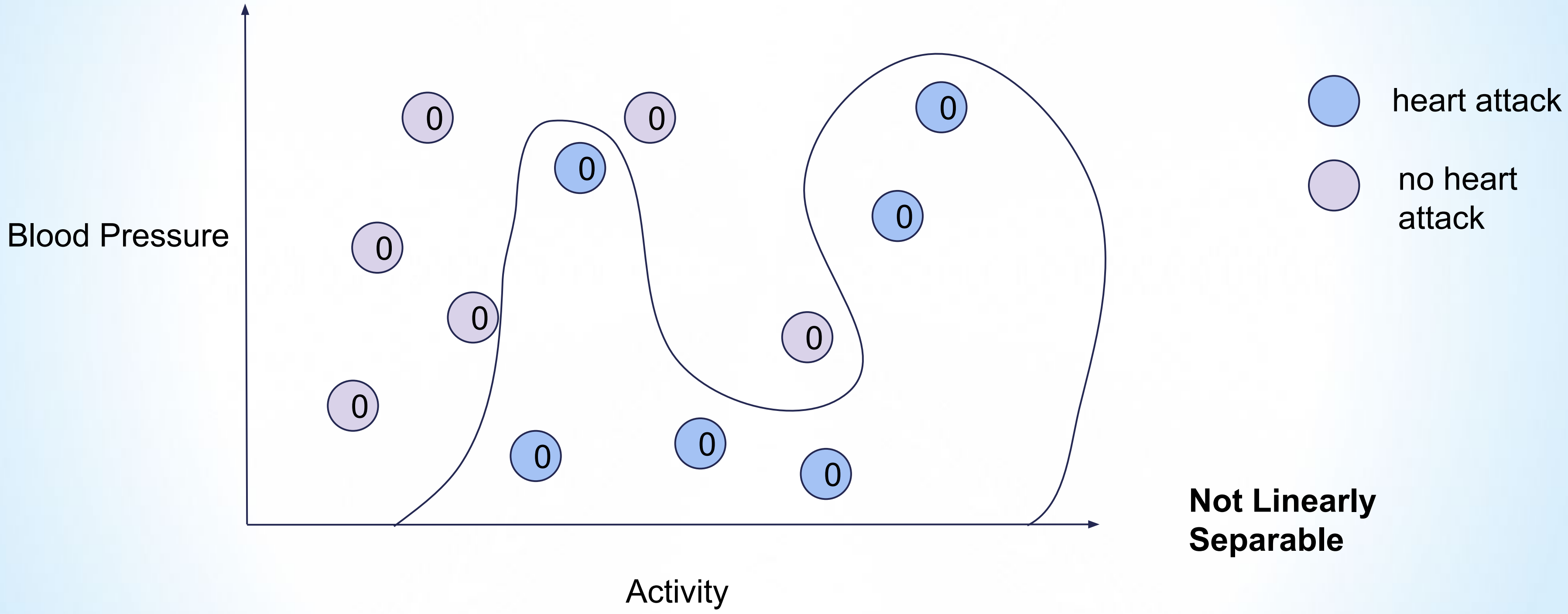
Used to learn model params
Epochs - the number of times (iterations) our model sees the train set

Used to choose model hyper-params

Used to evaluate model

Regularization

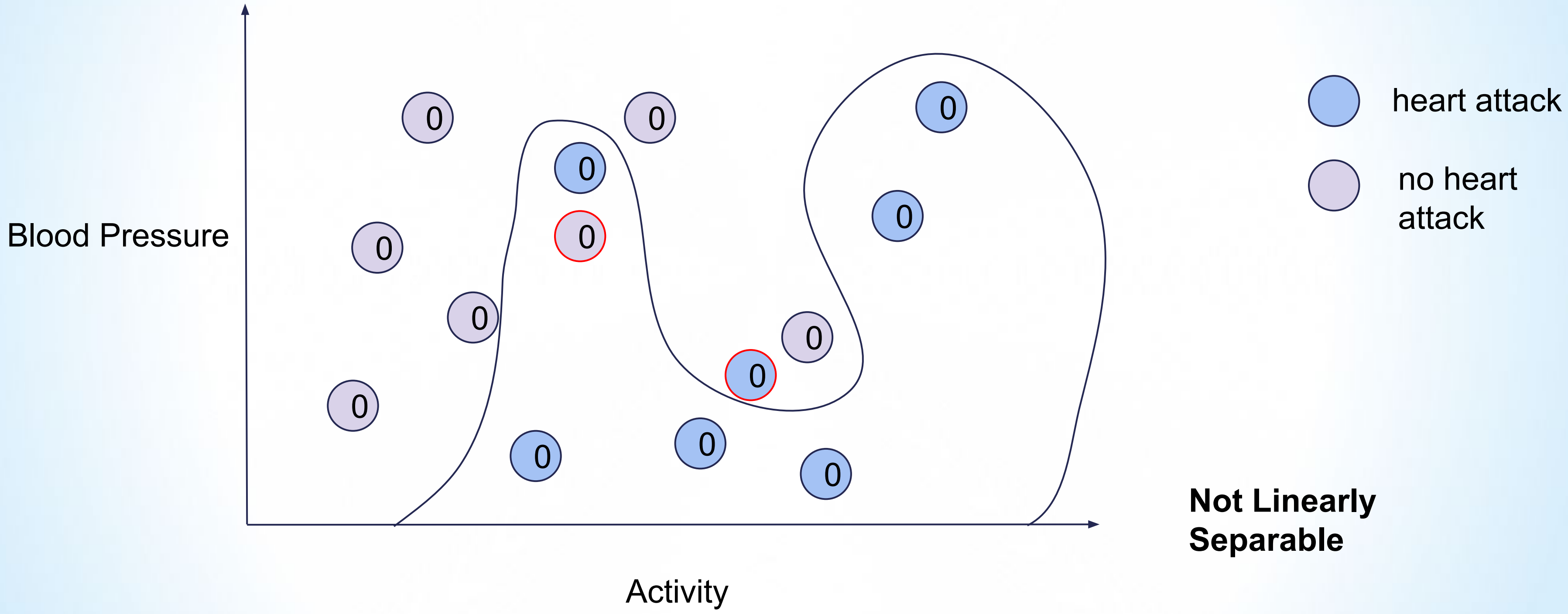
Overfitting



Not Linearly Separable

Loss = 0

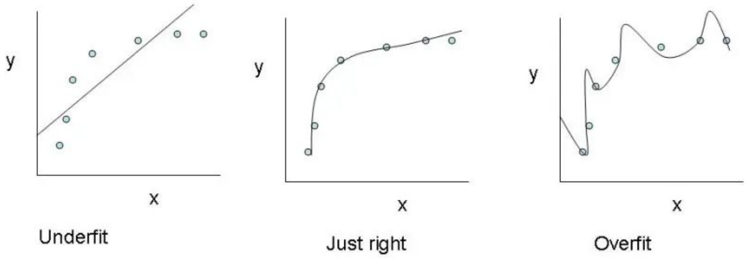
Overfitting - How Does it look on test set?



Not Linearly Separable

Loss = 0

Reminder: Underfitting & Overfitting



How to Handle Overfitting?

1. Reduce the number of features
   a. Manually select which features to keep
   b. Automatic feature selection
2. Regularization
   a. Keep all the features, but reduce the total weight of parameters θ_j
   b. Regularization works well when we have a lot of slightly useful features
3. Normalization

Minimum Description Length Principle

This principle says we'll prefer a concise hypothesis over getting the entire training set right

$$w^* = \arg\min_{w} \frac{1}{m} \sum L(x_i, y_i; w) + \lambda \Omega(w)$$

High lambda - underfitting

Low lambda - overfitting

l_p norms can be used as regularizers

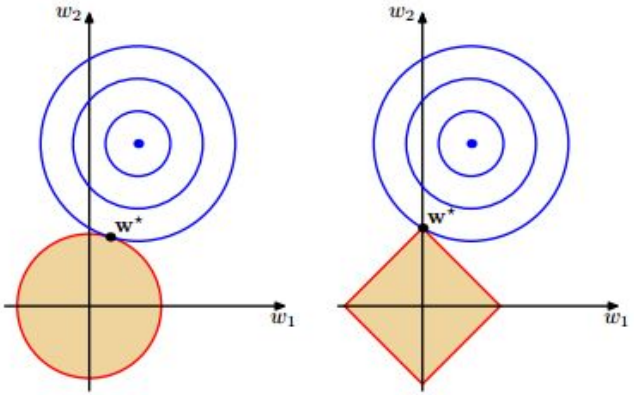
$$||\mathbf{w}||_2^2 = \sum_{d=1}^{D} w_d^2$$

$$||\mathbf{w}||_1 = \sum_{d=1}^{D} |w_d|$$

$$||\mathbf{w}||_p = (\sum_{d=1}^{D} w_d^p)^{1/p}$$

lp norms

- Can be used as regularizers
- $l2$ norm: convex, smooth, easy to optimize
- $l1$ norm: encourages sparse w, convex, but not smooth at axis points
- p < 1 : norm becomes non convex and hard to optimize

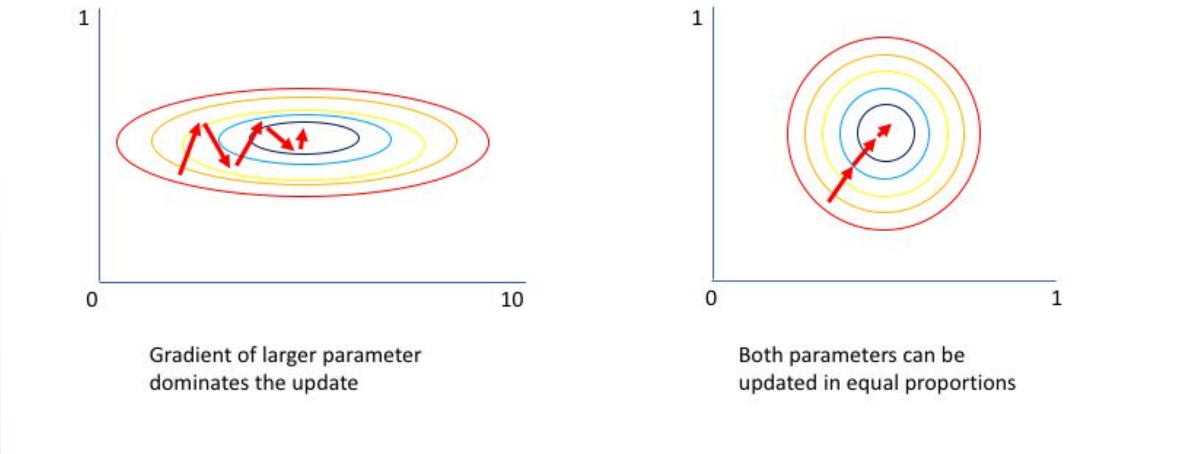


Y-DATA
SCHOOL OF DATA SCIENCE

Normalization

Why is it important?

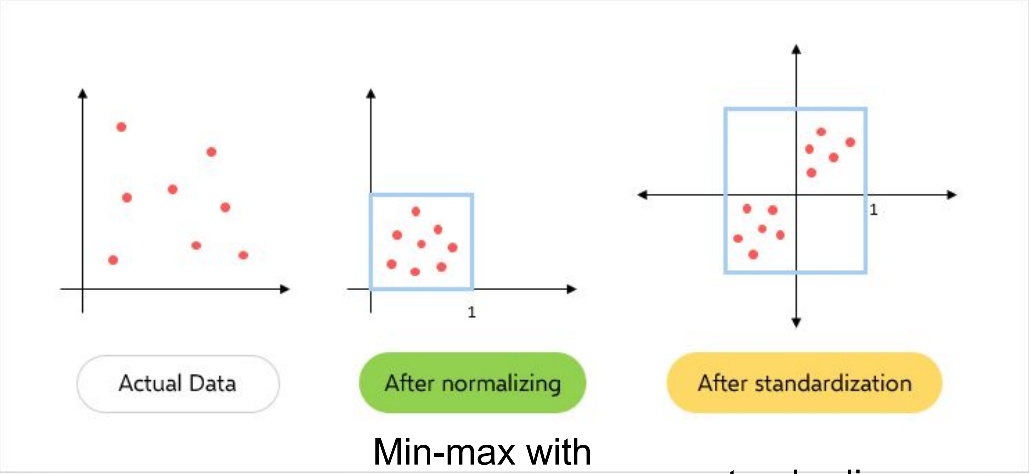
1. Helps gradient descent to converge
2. We don’t necessarily want large features to have larger impact
3. Some sort of regularization - lower hypothesis search space



Normalization methods

Min-Max: $v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$

Z-Score: $v' = \frac{v - mean_A}{stand\_dev_A}$



Min-max with new (0,1)

standardize

Linear Regression

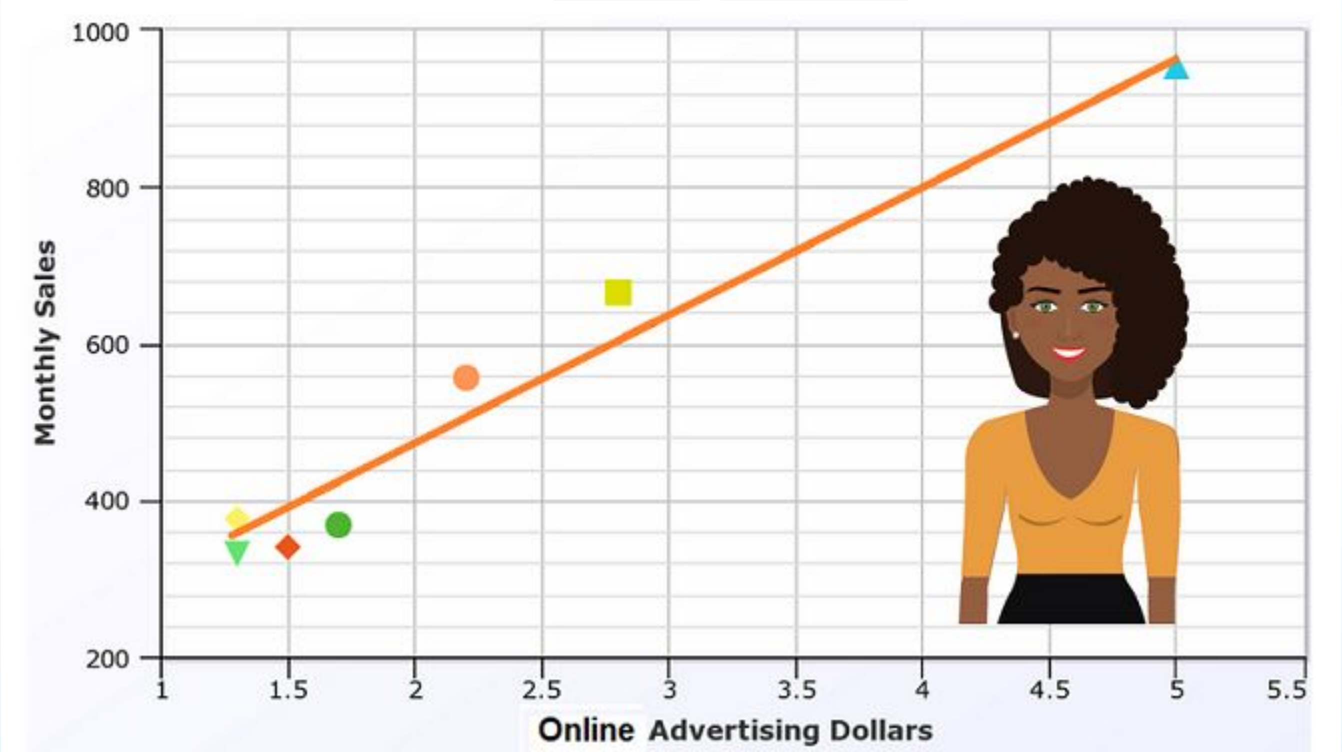
$$L = \frac{1}{2}||(Xw - y)||^2$$

Y-DATA
SCHOOL OF DATA SCIENCE

Monthly sales = w*x + b

slope

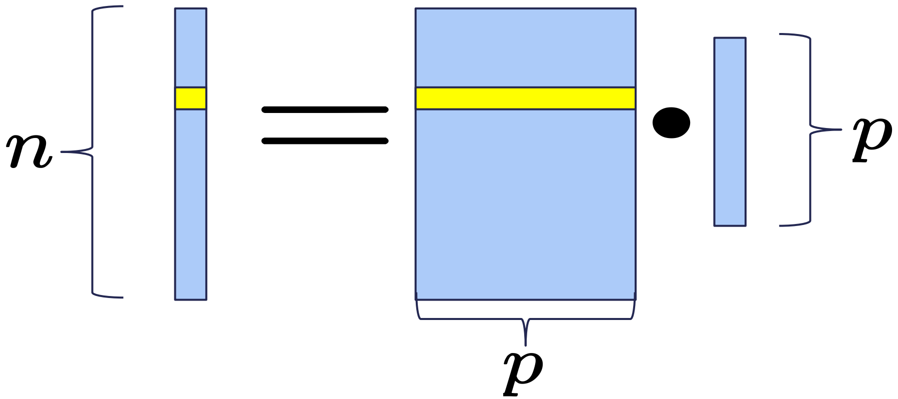
intersect



Matrix Form

1-padding for x

$$\hat{y}_{n\times 1} = X_{n\times p} w_{p\times 1}$$



Y-DATA SCHOOL OF DATA SCIENCE

Residual Sum of Squares Loss

$$L = \frac{1}{2}\|(Xw - y)\|^2 = \frac{1}{2}\underbrace{(Xw - y)^T(Xw - y)}_{\epsilon^T\epsilon}$$

Objective is to min Loss Function

is a scalar

Residual Sum of Squares Loss

$$L = \frac{1}{2}\|(Xw - y)\|^2 = \frac{1}{2}\underbrace{(Xw - y)^T(Xw - y)}_{\epsilon^T\epsilon}$$

Objective is to min Loss Function

is a scalar

$$L = \frac{1}{2}(w^T X^T X w - w^T X^T y + y^T y - y^T X w)$$

Common Matrix Derivatives

- *Notations*
  - *x is a scalar*
  - **x** is a vector

Rule	Comments
$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$	order is reversed, everything is transposed
$(\mathbf{a}^T\mathbf{B}\mathbf{c})^T = \mathbf{c}^T\mathbf{B}^T\mathbf{a}$	as above
$\mathbf{a}^T\mathbf{b} = \mathbf{b}^T\mathbf{a}$	(the result is a scalar, and the transpose of a scalar is itself)
$(\mathbf{A}+\mathbf{B})\mathbf{C} = \mathbf{AC}+\mathbf{BC}$	multiplication is distributive
$(\mathbf{a}+\mathbf{b})^T\mathbf{C} = \mathbf{a}^T\mathbf{C}+\mathbf{b}^T\mathbf{C}$	as above, with vectors
$\mathbf{AB} \neq \mathbf{BA}$	multiplication is **not** commutative

Scalar derivative			Vector derivative		
$f(x)$	$\rightarrow$	$\frac{\mathrm{d}f}{\mathrm{d}x}$	$f(\mathbf{x})$	$\rightarrow$	$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}}$
bx	$\rightarrow$	b	$\mathbf{x}^T\mathbf{B}$	$\rightarrow$	$\mathbf{B}$
bx	$\rightarrow$	b	$\mathbf{x}^T\mathbf{b}$	$\rightarrow$	$\mathbf{b}$
x^2	$\rightarrow$	$2x$	$\mathbf{x}^T\mathbf{x}$	$\rightarrow$	$2\mathbf{x}$
bx^2	$\rightarrow$	$2bx$	$\mathbf{x}^T\mathbf{B}\mathbf{x}$	$\rightarrow$	$2\mathbf{B}\mathbf{x}$

http://www.gatsby.ucl.ac.uk/teaching/courses/sntn/sntn-2017/resources/Matrix_derivatives_cribsheet.pdf

Residual Sum of Squares Loss

$$L = \frac{1}{2}(w^T X^T X w - w^T X^T y + y^T y - y^T X w)$$

Scalar derivative $f(x)$	$\rightarrow$	$\frac{df}{dx}$	Vector derivative $f(\mathbf{x})$	$\rightarrow$	$\frac{df}{d\mathbf{x}}$
bx	$\rightarrow$	b	$\mathbf{x}^T\mathbf{B}$	$\rightarrow$	$\mathbf{B}$
bx	$\rightarrow$	b	$\mathbf{x}^T\mathbf{b}$	$\rightarrow$	$\mathbf{b}$
x^2	$\rightarrow$	$2x$	$\mathbf{x}^T\mathbf{x}$	$\rightarrow$	$2\mathbf{x}$
bx^2	$\rightarrow$	$2bx$	$\mathbf{x}^T\mathbf{B}\mathbf{x}$	$\rightarrow$	$2\mathbf{B}\mathbf{x}$

Closed Solution

$$\frac{\partial L}{\partial w} = \frac{1}{2}\frac{\partial}{\partial w}(w^T X^T X w - w^T X^T y + y^T y - y^T X w)$$
$$= X^T(Xw - y)$$

$$\boxed{\begin{aligned} \frac{\partial L}{\partial w} &= X^T(Xw - y) = 0 \\ w_{OLS} &= \underbrace{(X^T X)^{-1} X^T}_{\text{pseudo-inverse of } X} y \end{aligned}}$$

Covariance matrix

Correlation between data and labels

Why can’t we always use a closed form solution?

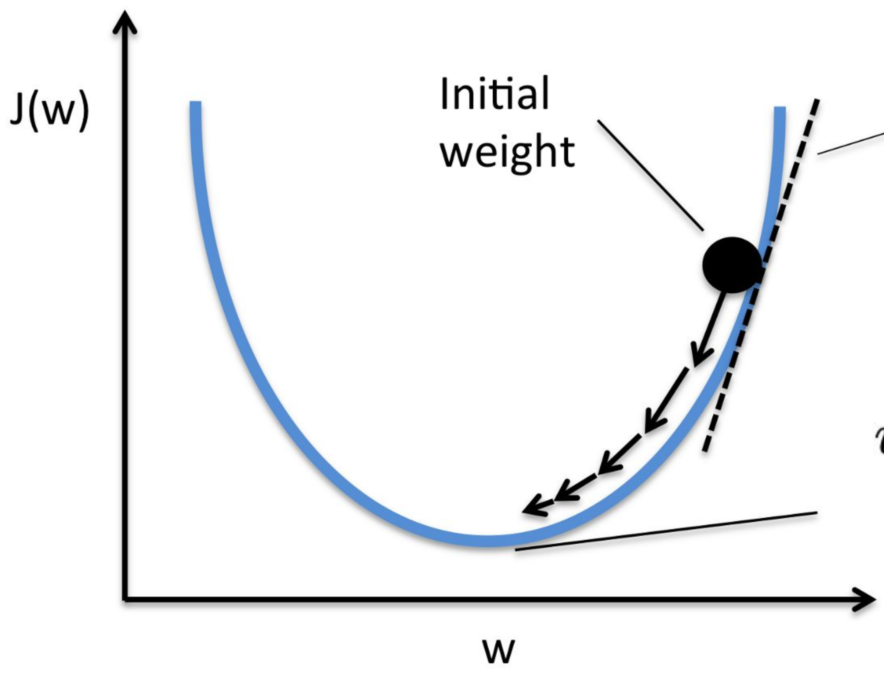
Assumes linear independence

Doesn’t necessarily exist - might be impossible to invert $(X^TX)^{-1}$

Solving inverse $(X^TX)^{-1}$ is computational expensive $O(n^3)$

Scale is an issue

Reminder: Gradient Descent for the Rescue



$$w^t \leftarrow w^{t-1} - \eta \cdot \frac{\partial \ell}{\partial w^{t-1}}$$

Gradient Descent for OLS (Ordinary Least Squares)

Let’s plug the gradient into gradient descent formula

$$\frac{\partial L}{\partial w} = X^T(Xw - y)$$

$$\underbrace{w'}_{\text{new}} = \underbrace{w}_{\text{old}} - \underbrace{\eta}_{\text{learning rate}} \underbrace{\frac{1}{n}}_{\text{normalization}} X^T(Xw - y)$$

Weighted Least Squares (WLS)

- This can be used when some samples are more valuable than others (imbalance, noise)
- W is a diagonal matrix

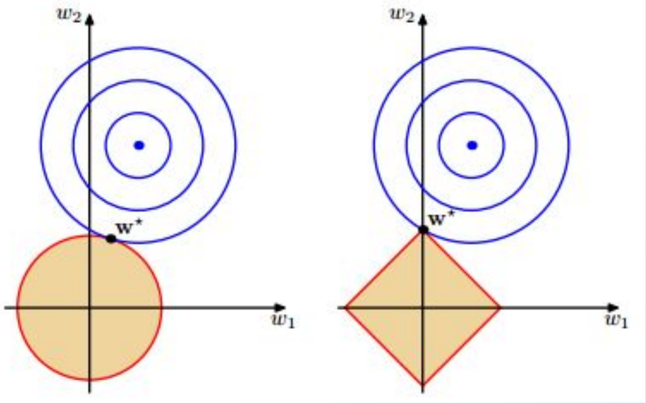
$$L = \frac{1}{2}\left\|W^{\frac{1}{2}}(Xw - y)\right\|^2$$

$$\hat{w}_{WLS} = (X^T W X)^{-1} X^T W y$$

Reminder: Regularization

l_p norms can be used as regularizers

$$||\mathbf{w}||_2^2 = \sum_{d=1}^{D} w_d^2$$
$$||\mathbf{w}||_1 = \sum_{d=1}^{D} |w_d|$$
$$||\mathbf{w}||_p = (\sum_{d=1}^{D} w_d^p)^{1/p}$$



Y-DATA
SCHOOL OF DATA SCIENCE

Ridge: L_2 Loss Regularization for OLS

- Minimize Objective function :

$$L = \underbrace{\frac{1}{2}||(Xw - y)||^2}_{\text{OLS Model fitting term}} + \underbrace{\lambda||w||^2}_{\text{Regularization term}}$$

- Where λ is the regularization parameter
- We can easily modify both our gradient descent function and our closed-form solution to fit the new loss function.

Y-DATA
SCHOOL OF DATA SCIENCE

Ridge: L_2 Loss Regularization

$$L = \frac{1}{2}\|(Xw - y)\|^2 + \lambda\|w\|^2$$

$$\frac{\partial L}{\partial w} = X^T(Xw - y) + \lambda w = 0$$

$$w_{ridge} = \underbrace{(X^T X + \lambda I)^{-1} X^T}_{\text{pseudo-inverse of } X \text{ with diagonal loading}} \text{y}$$

$$I = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

Lasso: L_1 Loss Regularization

- OLS with L_1 penalty

$$w_{lasso} = \underset{w}{argmin} L = \underbrace{\frac{1}{2}||(Xw - y)||^2}_{\text{OLS Loss}} + \lambda \underbrace{||w||}_{\text{L1 Regularization}}$$

$$||w|| = \sum_k^p ||w_k||$$

- Causes sparse weights
- Can be treated as “automatic feature selection”
- Harder to solve (solved using **Coordinate Descent**)

Elastic Net: Ridge and Lasso Combined

$$w_{elastic} = \underset{w}{argmin} L$$

$$= \frac{1}{2} \left\| (Xw - y) \right\|^2 + \lambda_1 \left\| w \right\| + \lambda_2 \left\| w \right\|^2$$

- Was found to be equivalent to SVM (will be discussed in SVM lecture)

Zhou, Quan; Chen, Wenlin; Song, Shiji; Gardner, Jacob; Weinberger, Kilian; Chen, Yixin. *A Reduction of the Elastic Net to Support Vector Machines with an Application to GPU Computing*. Association for the Advancement of Artificial Intelligence.

Summary of Linear Regression

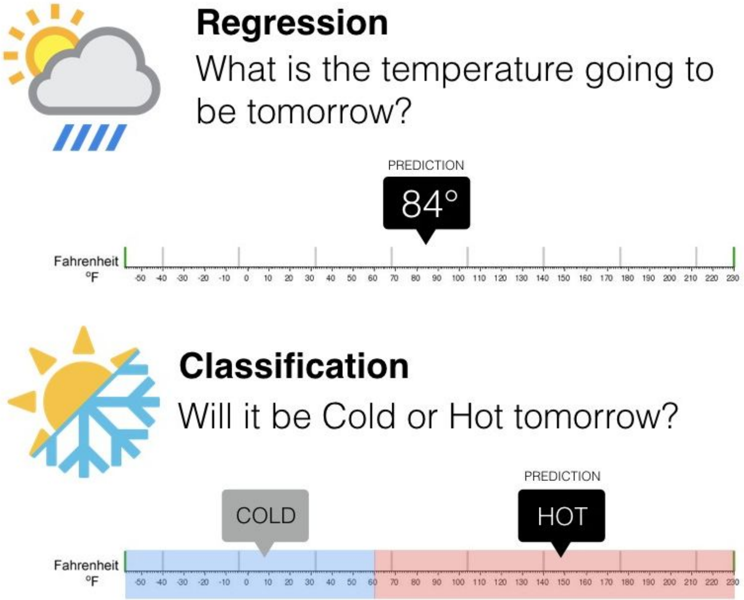
OLS	OLS + GD	Lasso	Ridge
Closed Solution	$\frac{1}{2}\|\|(Xw-y)\|\|^2$	$\frac{1}{2}\|\|(Xw-y)\|\|^2+\lambda\|\|w\|\|$	$\frac{1}{2}\|\|(Xw-y)\|\|^2+\lambda\|\|w\|\|^2$

Logistic Regression

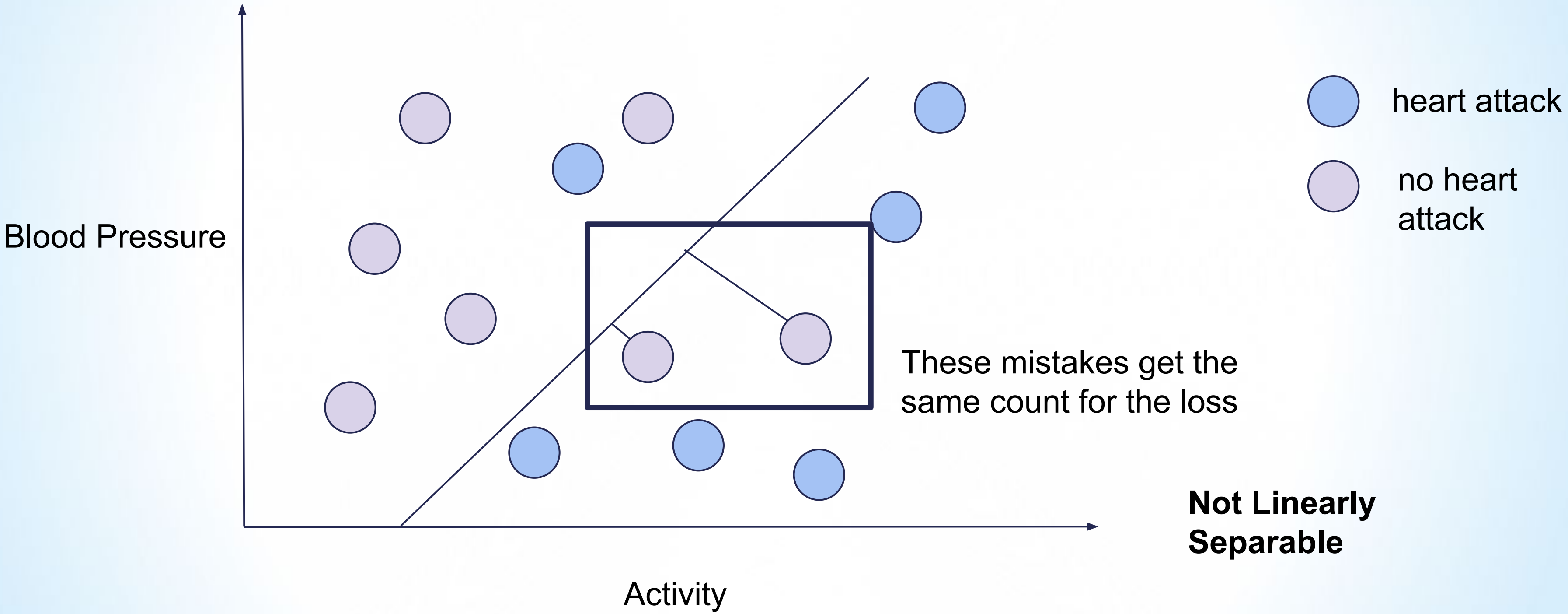
$$\ell_{\text{cross-ent}} = -\sum_k \mathbf{y}_{[k]} \log \hat{\mathbf{y}}_{[k]}$$

$$\hat{\mathbf{y}} = \operatorname{softmax}(\mathbf{x}\mathbf{W} + \mathbf{b})$$

Classification vs Regression

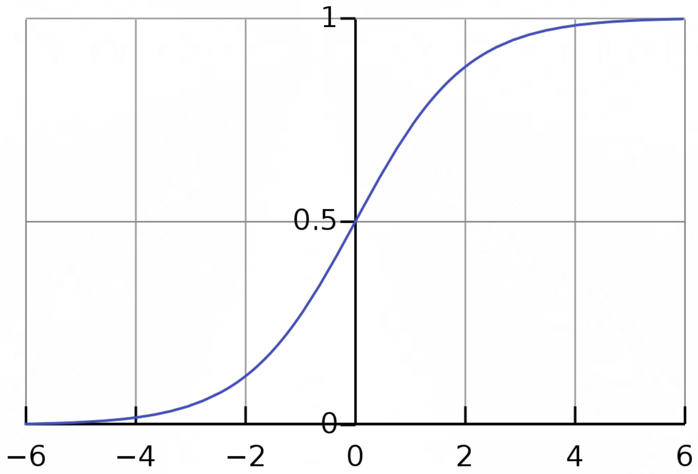


Why not use the 1-0 loss from before?



Can we do better? Sigmoid

$$\sigma(z_i) = \frac{1}{1+e^{-z_i}}$$



Y-DATA
SCHOOL OF DATA SCIENCE

Derivative of Sigmoid

$$\sigma(z_i) = \frac{1}{1+e^{-z_i}}$$

What is the derivative of a sigmoid?

- *Where z=wx*

Derivative of Sigmoid

$$\sigma(z_i) = \frac{1}{1+e^{-z_i}}$$

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

- *Where z=wx*

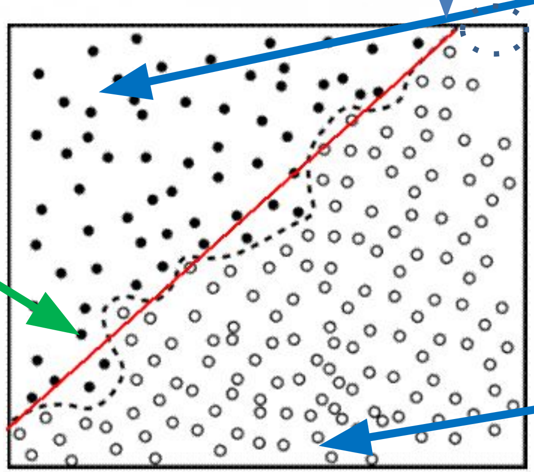
Sigmoid:

$$\left[\frac{1}{1+e^{-x}}\right]' = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1+e^{-x}-1}{(1+e^{-x})^2} = \frac{1+e^{-x}}{(1+e^{-x})^2} - \frac{1}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}} - \frac{1}{(1+e^{-x})^2}$$
$$= \frac{1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right) = sigmoid * (1 - sigmoid)$$

Logistic Regression with Sigmoid Intuition

- Different/Better Hypothesis & Objective

Linear separator



If X is near the linear separator w then
- wX is small
- $e^{-wX} \approx 1$
- $h_w(X) \approx 0.5$ (i.e. 50% probability)

If X is positive and far from the linear separator w then
- wX is big positive
- $e^{-wX} \approx 0$
- $h_w(X) \approx 1$

$h_w(x)$

If X is negative and far from the linear separator w then
- wX is big negative
- e^{-wX} >>10000
- $h_w(X) \approx 0$

x

Another Intuition for Logistic Regression

The logistic regression model tries to predict the odds of an event:

$$\frac{p(X)}{1-p(X)}$$

$$\frac{p(X)}{1-p(X)} = e^{\beta^T X} = e^{\beta_1 X_1 + \ldots + \beta_p X_p}$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta^T X$$

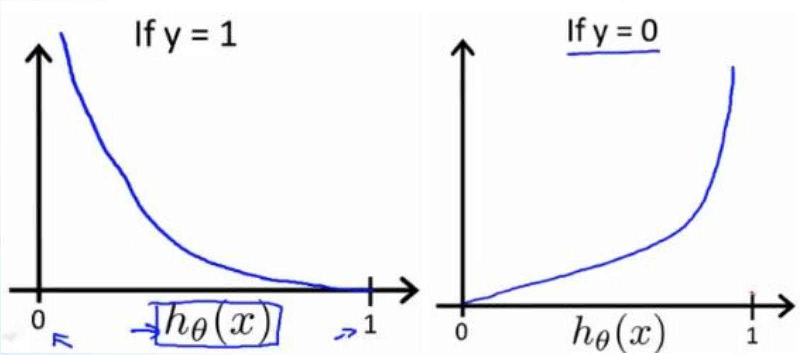
$$p(X) = \frac{\exp(\beta^T X)}{1+\exp(\beta^T X)}$$

Cross Entropy Loss/Negative Log Likelihood

- Let's define a loss for each observation based on a fix separator

  - Loss(x,y) = $\begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$

Loss(x,y)

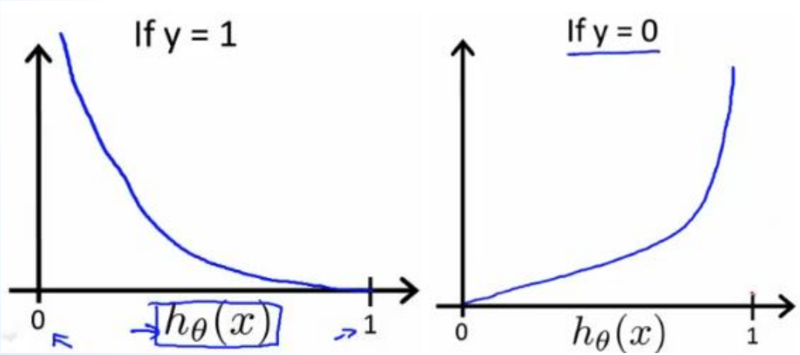


Cross Entropy Loss/Negative Log Likelihood

- Let's define a loss for each observation based on a fix separator
  - Loss(x,y) = $\begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$

$$\boxed{-y\log(H_\theta(x)) - (1-y)\log(1 - H_\theta(x))}$$

Loss(x,y)



LR Derivation Simplified

- Optimize $$G = y \cdot \log(h) + (1-y) \cdot \log(1-h)$$
- Where $$h = 1/(1+e^{-z}) \quad z(\theta) = x\theta:$$
- Derivation $$\frac{dG}{d\theta} = \frac{dG}{dh}\frac{dh}{dz}\frac{dz}{d\theta}$$

$$\frac{dG}{\partial h} = \frac{y}{h} - \frac{1-y}{1-h} = \frac{y-h}{h(1-h)} \qquad \frac{dh}{dz} = h(1-h) \qquad \frac{dz}{d\theta} = x$$

Derivative of sigmoid

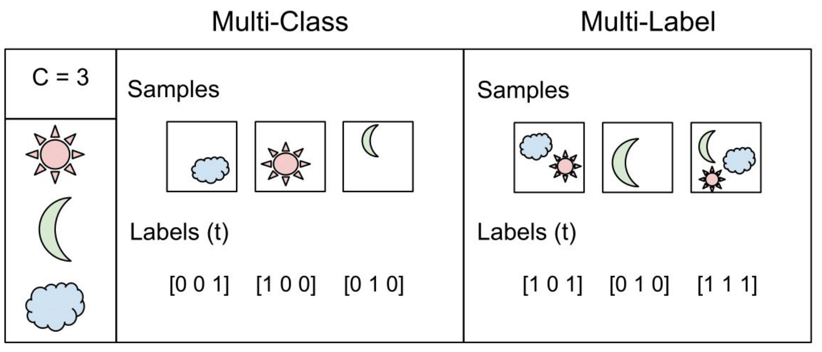
- Result $$\frac{dG}{d\theta} = (y-h)x$$

Y-DATA
SCHOOL OF DATA SCIENCE

Reminder: Multi-Class & Multi-Label

Multiclass classification

Assigns a single class out of m possible classes (y output an integer between 1 and m)



Multilabel classification

Assign a 0 or 1 labels for each of the m possible classes (y output a binary vector of size m)

Y-DATA SCHOOL OF DATA SCIENCE

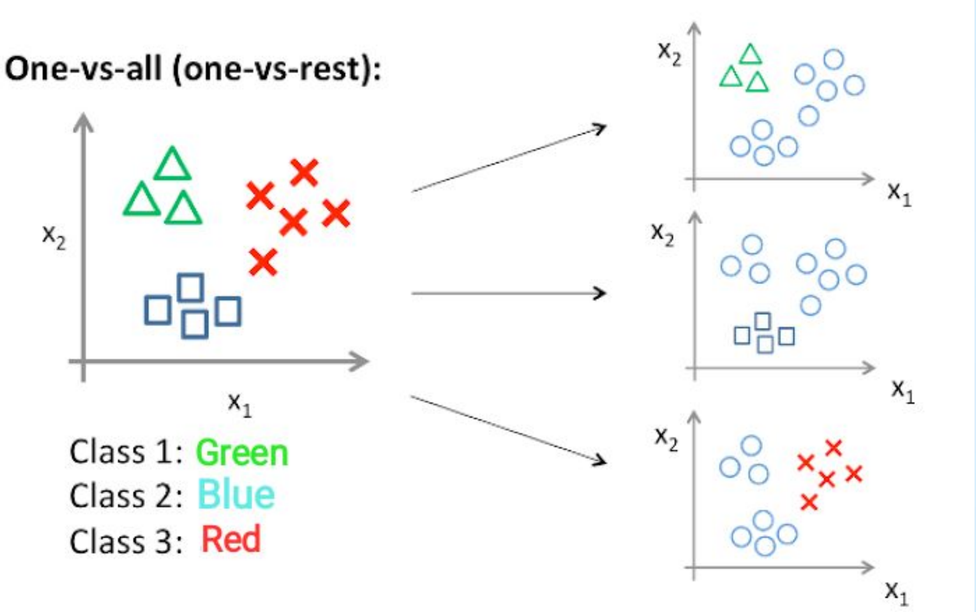
Heuristic Solutions to Multiclass Problems

One vs All (One vs Rest)

Training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives - then choose the class with maximal confidence

One vs One

Train K (K − 1) / 2 binary classifiers, each receives the samples of a pair of classes from the original training set, and learn to distinguish these two classes. During prediction time, a voting scheme is applied



Cross Entropy Multiclass

$$\ell_{\text{cross-ent}} = -\log \hat{\mathbf{y}}_{[t]}$$

$$\ell_{\text{cross-ent}} = -\sum_{k} \mathbf{y}_{[k]} \log \hat{\mathbf{y}}_{[k]}$$

Sigmoid for Multiclass (Softmax)

$$\sigma(z_i) = \frac{1}{1+e^{-z_i}}$$

$$\text{softmax}(\mathbf{v})_{[i]} = \frac{e^{\mathbf{v}[i]}}{\sum_{i'} e^{\mathbf{v}[i']}}$$

Log-linear model
(aka "logistic regression")

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{xW} + \mathbf{b})$$

$$\hat{\mathbf{y}}_{[i]} = \frac{e^{(\mathbf{xW}+\mathbf{b})_{[i]}}}{\sum_i e^{(\mathbf{xW}+\mathbf{b})_{[i]}}}$$

Y-DATA
SCHOOL OF DATA SCIENCE

Cross Entropy Multiclass Derivative - Try this @ home

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right]$$

$$\underset{\text{linearity}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{\partial}{\partial \theta_j} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \frac{\partial}{\partial \theta_j} \log(1 - h_\theta(x^{(i)})) \right]$$

$$\underset{\text{chain rule}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{\frac{\partial}{\partial \theta_j} h_\theta(x^{(i)})}{h_\theta(x^{(i)})} + (1 - y^{(i)}) \frac{\frac{\partial}{\partial \theta_j} (1 - h_\theta(x^{(i)}))}{1 - h_\theta(x^{(i)})} \right]$$

$$\underset{h_\theta(x) = \sigma(\theta^\top x)}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{\frac{\partial}{\partial \theta_j} \sigma(\theta^\top x^{(i)})}{h_\theta(x^{(i)})} + (1 - y^{(i)}) \frac{\frac{\partial}{\partial \theta_j} (1 - \sigma(\theta^\top x^{(i)}))}{1 - h_\theta(x^{(i)})} \right]$$

$$\underset{\sigma'}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{\sigma(\theta^\top x^{(i)})(1 - \sigma(\theta^\top x^{(i)})) \frac{\partial}{\partial \theta_j} (\theta^\top x^{(i)})}{h_\theta(x^{(i)})} - (1 - y^{(i)}) \frac{\sigma(\theta^\top x^{(i)})(1 - \sigma(\theta^\top x^{(i)})) \frac{\partial}{\partial \theta_j} (\theta^\top x^{(i)})}{1 - h_\theta(x^{(i)})} \right]$$

$$\underset{\sigma(\theta^\top x) = h_\theta(x)}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{h_\theta(x^{(i)})(1 - h_\theta(x^{(i)})) \frac{\partial}{\partial \theta_j} (\theta^\top x^{(i)})}{h_\theta(x^{(i)})} - (1 - y^{(i)}) \frac{h_\theta(x^{(i)})(1 - h_\theta(x^{(i)})) \frac{\partial}{\partial \theta_j} (\theta^\top x^{(i)})}{1 - h_\theta(x^{(i)})} \right]$$

$$\underset{\frac{\partial}{\partial \theta_j} (\theta^\top x^{(i)}) = x_j^{(i)}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} (1 - h_\theta(x^{(i)})) x_j^{(i)} - (1 - y^{i}) \, h_\theta(x^{(i)}) x_j^{(i)} \right]$$

$$\underset{\text{distribute}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{i} - y^{i} h_\theta(x^{(i)}) - h_\theta(x^{(i)}) + y^{(i)} h_\theta(x^{(i)}) \right] x_j^{(i)}$$

$$\underset{\text{cancel}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} - h_\theta(x^{(i)}) \right] x_j^{(i)}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left[h_\theta(x^{(i)}) - y^{(i)} \right] x_j^{(i)}$$

Y-DATA
SCHOOL OF DATA SCIENCE

Other Loss Functions

- Hinge: $\max(0, 1 - ywx)$ $\rightarrow$ You'll discuss this at SVM lecture
- Exponential loss
- C-loss
- …

Y-DATA
SCHOOL OF DATA SCIENCE

Explainability

ELI5 - Global explanation is the model W

Bank Marketing Data Set — [LINK]()

y=1 top features

Weight?	Feature
+1.170	month__mar
+1.117	month__dec
+0.968	education__illiterate
+0.920	month__oct
+0.711	contact__cellular
+0.619	month__sep
+0.615	job__retired
+0.580	job__student
+0.564	default__no
+0.528	<BIAS>
+0.424	poutcome__success
+0.372	marital__unknown
+0.208	job__unknown
+0.201	housing__no
+0.193	day_of_week__wed
+0.188	housing__unknown
… 17 more positive …	
… 21 more negative …	
-0.682	month__jul
-0.761	month__may
-0.798	month__aug
-0.886	month__nov

Explainability

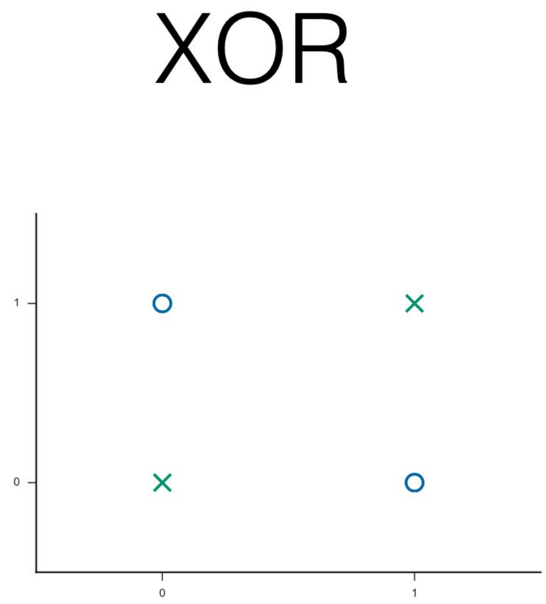
Local explanation is wx

y=1 (probability **0.961**, score **3.203**) top features

Contribution?	Feature	Value
+0.711	contact__cellular	1.000
+0.564	default__no	1.000
+0.528	<BIAS>	1.000
+0.424	poutcome__success	1.000
+0.363	previous	2.000
+0.208	job__unknown	1.000
+0.193	day_of_week__wed	1.000
+0.188	marital__single	1.000
+0.156	loan__no	1.000
+0.139	housing__yes	1.000
+0.129	age	27.000
+0.024	education__university.degree	1.000
-0.005	pdays	3.000
-0.146	month__jun	1.000
-0.271	campaign	4.000

Y-DATA
SCHOOL OF DATA SCIENCE

Can we build a linear model for XOR?



$$(0,0)\cdot\mathbf{w}+b<0$$

$$(0,1)\cdot\mathbf{w}+b\geq 0$$

$$(1,0)\cdot\mathbf{w}+b\geq 0$$

$$(1,1)\cdot\mathbf{w}+b<0$$

$$\mathbf{w} = ?$$

Can we build a linear model for XOR?

Linear Models will underfit xor, we need non-linearity which can be achieved:

* data pre-processing
* kernels
* adding non-linearity to the model

Code

Let’s Think About This Together

1. What are the main hyper-parameters?
2. Can it work for multi-class data (relevant only for logistic)?
3. How does it handle categorical data?
4. How does it handle missing data?
5. Is it sensitive to outliers?
6. What if some features are correlated?
7. Is it prone to overfitting?
8. Is it Interpretable?
9. Can it be parallelized?
10. Speed of training
11. Speed of prediction

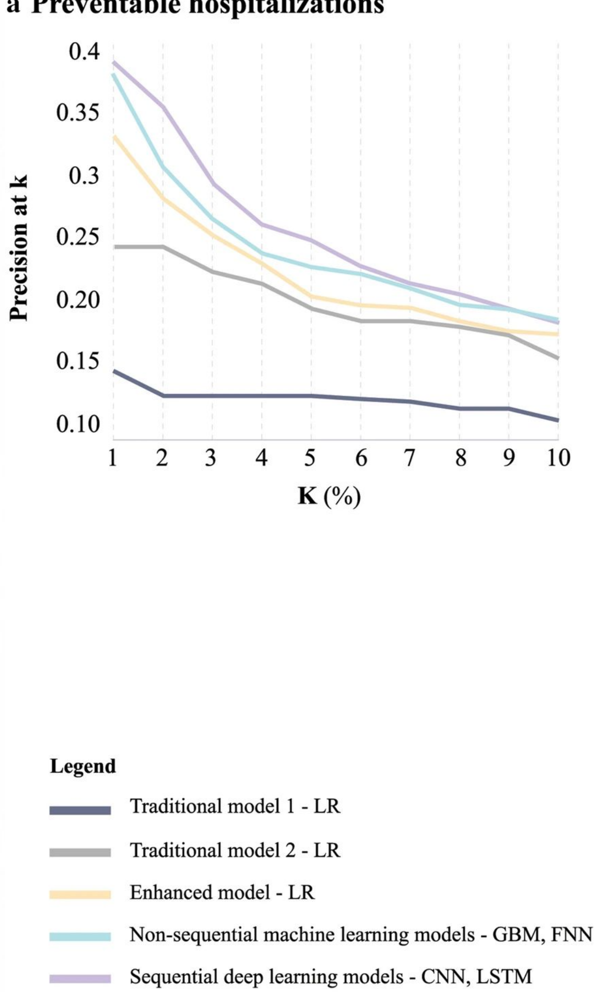
Let’s Think About This Together

1. What are the main hyper-parameters? Optimizer: LR, stopping criteria, initial weights, epochs, regularization lambda
2. Can it work for multi-class data (relevant only for logistic)? Yes
3. How does it handle categorical data? We need to make it numeric
4. How does it handle missing data? No
5. Is it sensitive to outliers? Yes
6. What if some features are correlated? Doesn’t handle well
7. Is it prone to overfitting? Regularization
8. Is it Interpretable? Yes
9. Can it be parallelized? Not off the shelf
10. Speed of training - depends on optimizer and hyper-params
11. Speed of prediction - linear

Y-DATA
SCHOOL OF DATA SCIENCE

Congestive Heart Failure

Comparison of deep learning with traditional models to predict preventable acute care use and spending among heart failure patients



Summary

Y-DATA
SCHOOL OF DATA SCIENCE

Summary

	Linear Regression	Logistic Regression
Target Type	Regression	Classification
Loss	$\frac{1}{2}\|(Xw - y)\|^2$	$\ell_{\text{cross-ent}} = -\sum_k \mathbf{y}_{[k]} \log \hat{\mathbf{y}}_{[k]}$ $\hat{\mathbf{y}} = \text{softmax}(\mathbf{x}\mathbf{W} + \mathbf{b})$

Y-DATA
SCHOOL OF DATA SCIENCE

Pros and Cons

Pros	Cons
1. Fast 2. Simple 3. Explainable	1. Can’t handle missing data - needs imputation 2. Categorical data needs translation to numeric 3. Assumption of linear relations 4. Sensitivity to correlated features