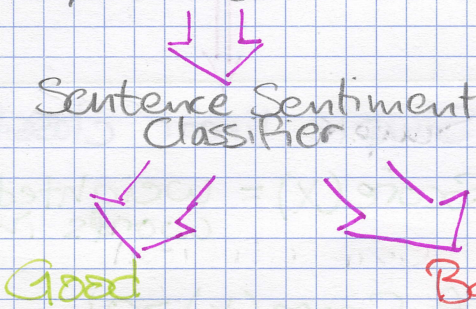
Linear Classifiers & Logistic Regression

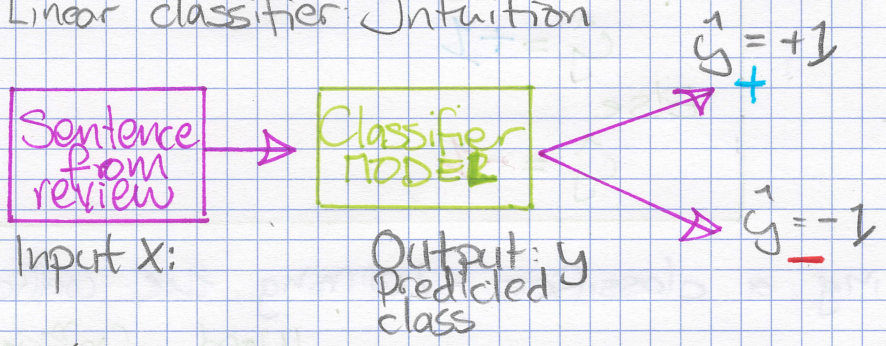
Linear classifiers: A motivating example

Classifying sentiment of review

"Easily best sushi in Seattle"



Linear classifier: Intuition



Input x_i:

Output: predicted class $\hat{y}$

A (linear) classifier

- will use training data to learn a weight or coefficient for each word

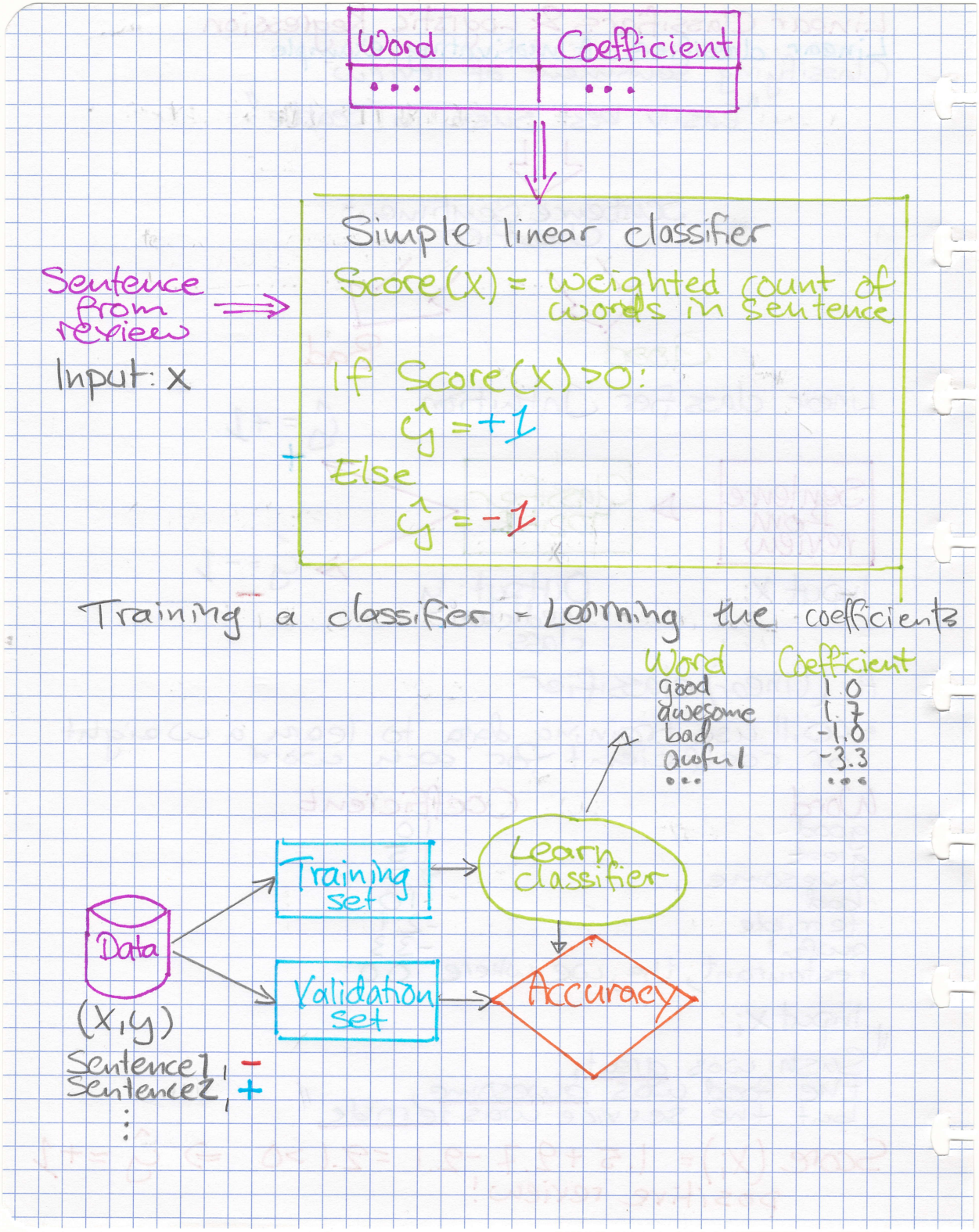
Word	Coefficient
good	1.0
great	1.5
awesome	2.7
bad	-1.0
terrible	-2.1
awful	-3.3
restaurant, the, we, where	0.0

Input x_i:

"Sushi was great, the food was awesome, but the service was terrible."

Score $(x_i) = 1.5 + 2.7 - 2.1 = 2.1 > 0 \Rightarrow \hat{y} = +1$

positive review!

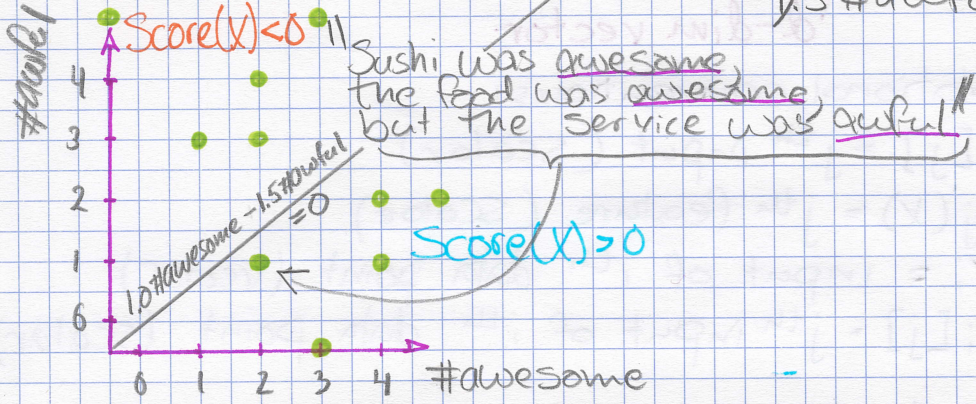

Word
Coefficient
...
...
Simple linear classifier
Score(x) = weighted count of words in sentence
If Score(x) > 0:
ŷ = +1
Else
ŷ = -1
Sentence from review
Input: x
Training a classifier = Learning the coefficients
Word
Coefficient
good
1.0
awesome
1.7
bad
-1.0
awful
-3.3
Data
(x, y)
Sentence1, -
Sentence2, +
Training set
Validation set
Learn classifier
Accuracy

Decision boundaries

Suppose only two words had non-zero coefficient

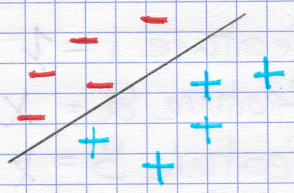
Word	Coefficient
#awesome	1.0
#awful	-1.5

$\Rightarrow$ Score(x) = 1.0 #awesome − 1.5 #awful



Decision boundary separates positive and negative predictions

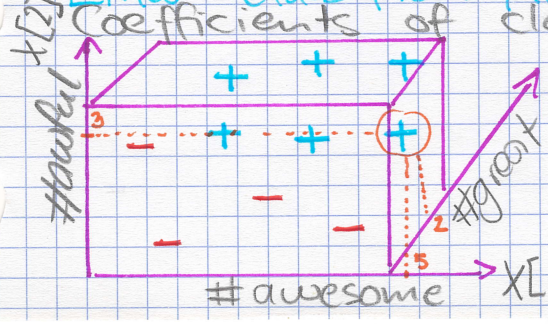
- For linear classifiers
  - When 2 coefficients are non-zero
    → line
  - When 3 coefficients are non-zero
    → plane
  - When many coefficients are non-zero
    → hyperplane



- For more general classifiers
  → more complicated shapes

Linear classifier: Model

Coefficients of classifier



$score(x) = w_0 + w_1 \#awesome^{5} + w_2 \#awful^{3} + w_3 \#great^{2}$

General notation

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

Inputs: $x = (x[1], x[2], \ldots, x[d])$

d-dim vector

Notational conventions:

$x[j]$ = j^{th} input (scalar)

$h_j(x)$ = j^{th} feature (scalar)

x_i = input of i^{th} data point (vector)

$x_i[j]$ = j^{th} input of i^{th} data point (scalar)

Simple hyperplane

Model $\hat{y}_i = \text{sign}(\text{Score}(x_i))$

$\text{Score}(x_i) = w_0 + w_1 x_i[1] + \ldots + w_d x_i[d] = w^T x_i$

feature 1 = 1

feature 2 = $x[1]$ ... eg. #awesome

feature 3 = $x[2]$ ... eg. #awful

⋮

feature $d+1$ = $x[d]$

More generic features... D-dimensional hyperplane

Model: $\hat{y}_i = \text{sign}(\text{Score}(x_i))$

$\text{Score}(x_i) = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i)$

$= \sum_{j=0}^{D} w_j h_j(x_i) = w^T h(x_i)$

feature 1 = $h_0(x)$ ... e.g., 1

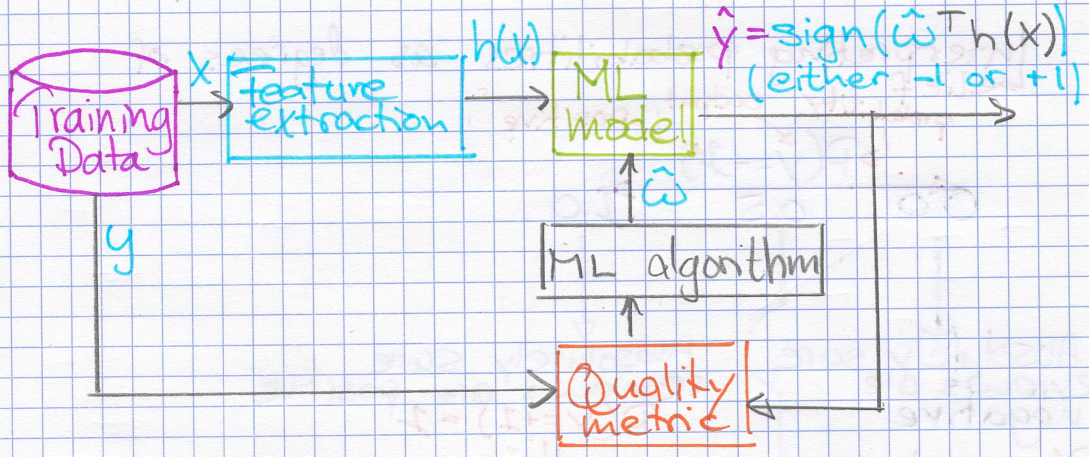
feature 2 = $h_1(x)$ ... eg $x[1]$ - # awesome

feature 3 = $h_2(x)$ ... eg $x[2]$ = # awful

or, $\log(x[7]) \times [2] = \log(\#bad) \times \#awful$

or, tf-idf("awful")

feature D+1 = $h_D(x)$ .. some other function of $x[1], \ldots, x[d]$



Predicting class probabilities

How confident is your prediction?

- Thus far, we've outputted a prediction +1 or -1
- But, how sure are you about the prediction?

"The sushi & everything else were awesome"

Definite +1

"The sushi was good, the service was OK"

not sure

Basics of probabilities

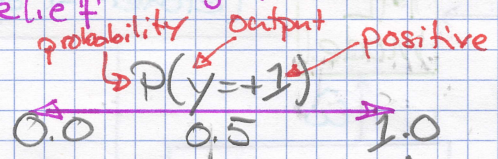
Probability a review is positive is 0.7

⇩

x = review text	y = sentiment
all the sushi was delicious! Easily best sushi in Seattle.	+1
The sushi & everything else were awesome	+1
My wife tried their ramen, it was pretty forgettable	-1
The sushi was good, the service was OK	+1

I expect 70% of rows to have $y = +1$ (Exact number will vary for each specific dataset)

Interpreting probabilities as degrees of belief



Absolutely sure reviews are negative

$P(y=+1) = 0$

⇓

$P(y=-1) = 1$

Absolutely sure reviews are positive

$P(y=+1) = 1$

⇓

$P(y=-1) = 1 - P(y=+1) = 0$

Not sure if reviews are positive or negative

$P(y=+1) = P(y=-1) = 0.5$

Property	Two class (e.g. y is +1 or -1)	Multiple classes (e.g. y is dog, cat or bird)
Probabilities always between 0 & 1	$0 \le P(y=+1) \le 1$ $0 \le P(y=-1) \le 1$	$0 \le P(y=dog) \le 1$ $0 \le P(y=cat) \le 1$ $0 \le P(y=bird) \le 1$
Probabilities sum up to 1	$P(y=+1)+P(y=-1)=1$	$P(y=dog)+P(y=cat)+P(y=bird)=1$

Conditional probabilities

Probability of review with 3 "awesome" and 1 "awful" is positive is 0.9

x = review text — y = sentiment

All the sushi was delicious! Easily best sushi in Seattle — +1

{Sushi was awesome & everything else was awesome! The service was awful, but overall awesome place} — +1

My wife tried their ramen, it was pretty forgettable — -1

The sushi was good, the service was OK — +1

...

{awesome ... awesome ... awful .. awesome} +1

...

{awesome ... awesome ... awful awesome} +1

I expect 90% of rows with reviews containing 3 "awesome" & 1 "awful" to have y = +1 (Exact number will vary for each specific dataset)

Interpreting conditional probabilities

probability → output → positive → given → input → Input sentence

$P(y=+1 \mid x_i = \text{"all the sushi was delicious"})$

Output label

0,0 — 0,5 — 1,0

Absolutely sure review "All the sushi was delicious" is negative

$P(y=+1|x_i) = 0$

$\Downarrow$

$P(y=-1|x_i) = 1$

"Absolutely sure review "All the sushi was delicious" is positive

$P(y=+1|x_i) = 1$

$\Rightarrow P(y=-1|x_i) = 1 - P(y=+1|x_i)$

$= 0$

Not sure if review "All the sushi was delicious" positive or negative

$P(y=+1|x_i) = P(y=-1|x_i) = 0.5$

Property	Two class (e.g. y is +1 or -1, x_i is review text)	Multiple classes (e.g. y is dog, cat or bird, x_i is image)
Conditional probability always between 0 & 1	$0 \le P(y=+1\|x_i) \le 1$ $0 \le P(y=-1\|x_i) \le 1$	$0 \le P(y=\text{dog}\|x_i) \le 1$ $0 \le P(y=\text{cat}\|x_i) \le 1$ $0 \le P(y=\text{bird}\|x_i) \le 1$
Conditional probabilities sum up to 1 over y, but not over x	$P(y=+1\|x_i) + P(y=-1\|x_i) = 1$ But $\sum_x P(y=+1\|x) \neq 1$	$P(y=\text{dog}\|x_i) +$ $P(y=\text{cat}\|x_i) +$ $P(y=\text{bird}\|x_i) = 1$

Using probabilities in classification

How confident is your prediction?

"The sushi & everything else were awesome."

↑

Definite +1

$P(y=+1 \mid X=$ "The sushi & everything else were awesome"$) = 0,99$

"The sushi was good, the service was OK"

↑

Not sure

$P(y=+1 \mid X=$ "The sushi was good, the service was OK"$) = 0,55$

Many classifiers provide a degree of certainty:

Output label → $P(y \mid x)$ ← Input sentence

Extremely useful in practice

Goal: Learn conditional probabilities from data

Training data: N observations (x_i, y_i)

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
0	2	-1
3	3	-1
4	1	+1

Optimize quality metric on training data

Find best model $\hat{P}$ by finding best $\hat{w}$

useful for predicting $\hat{y}$

Sentence from review ⇒

Input: x

Predict most likely class

$\hat{P}(y|x)$ = estimate of class probabilities

If $\hat{P}(y=+1|x) > 0.5$:

$\hat{y} = +1$

Else:

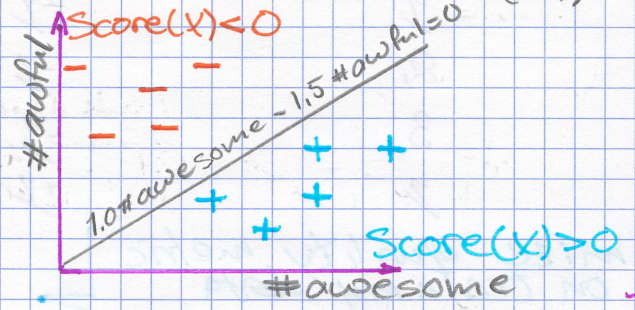
$\hat{y} = -1$

- Estimating $\hat{P}(y|x)$ improves interpretability
  - Predict $\hat{y} = +1$ and tell me how sure you are

Predicting class probabilities with generalized linear models

Thus far, we focused on decision boundaries

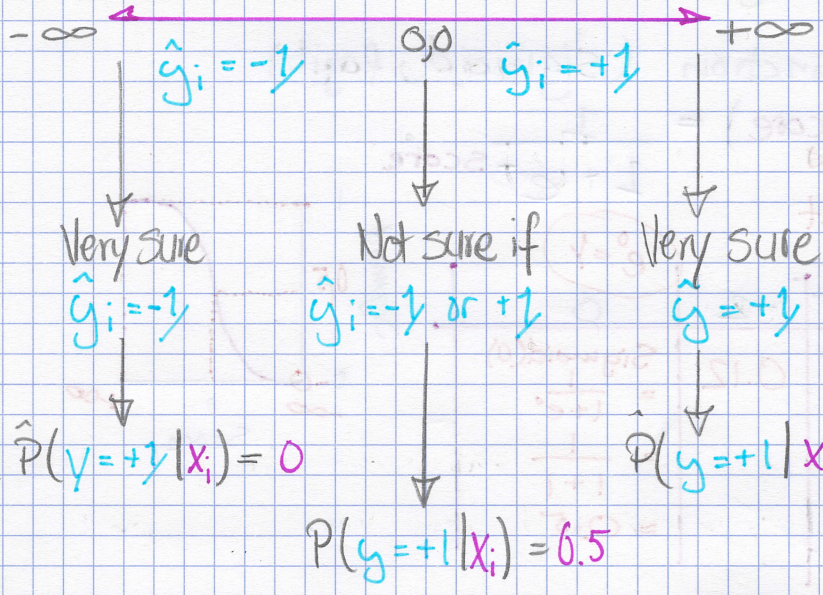
$$\text{Score}(x_i) = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i)$$
$$= w^T h(x_i)$$



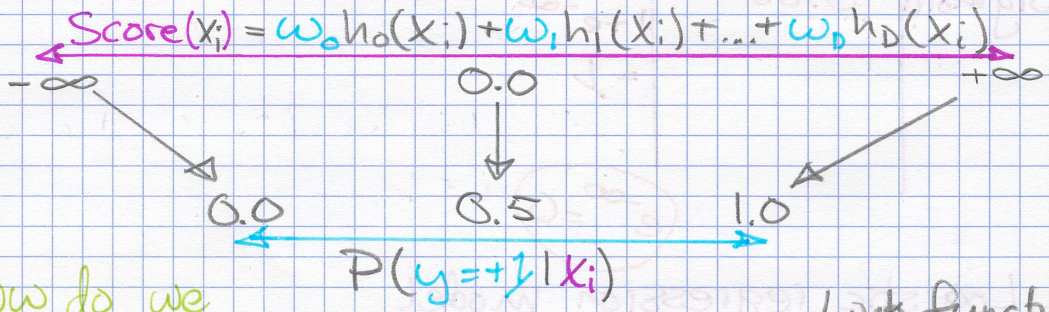
Relate $\text{Score}(x_i)$ to $\hat{P}(y=+1|x, \hat{w})$?

Interpreting Score(x_i)

$$\text{Score}(x_i) = w^T h(x_i)$$



Why not just use regression to build classifier?



How do we link $-\infty$ $+\infty$ to 0,1 ?

Link function: squeeze real line into [0,1]

$g(w^T h(x_i))$

Link function

$$\hat{P}(y=+1|x_i) = g(w^T h(x_i))$$

Generalized linear model

Logistic regression classifier:
linear score with logistic link function

Logistic function (sigmoid, logit)

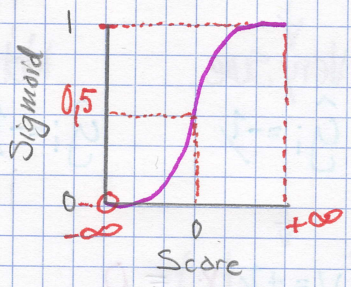
$$\text{Sigmoid}(\text{Score}) = \frac{1}{1+e^{-\text{score}}}$$

Score: input

$e^0 = 1$

Score	$-\infty$	-2	0
Sigmoid	$\frac{1}{1+e^{\infty}} = 0$	0.12	Sigmoid(0) $= \frac{1}{1+e^0} = \frac{1}{1+1} = 0,5$

$e^{\infty} = \infty$



Score	$+2$	$+\infty$
Sigmoid	0.88	$\frac{1}{1+e^{-\infty}} = 1$

$e^{-\infty} = 0$

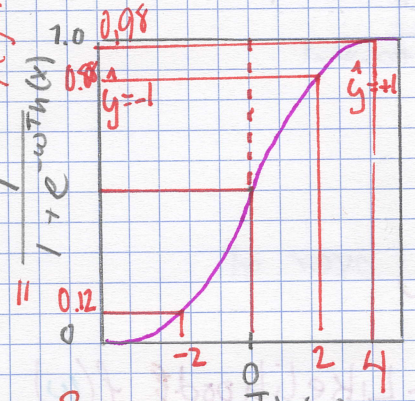
Logistic regression model

$$P(y=+1 \mid x_i, w) = \text{sigmoid}(\text{Score}(x_i)) = \frac{1}{1+e^{-\text{Score}(x_i)}} = \frac{1}{1+e^{-w^T h(x_i)}}$$

Understanding the logistic regression model

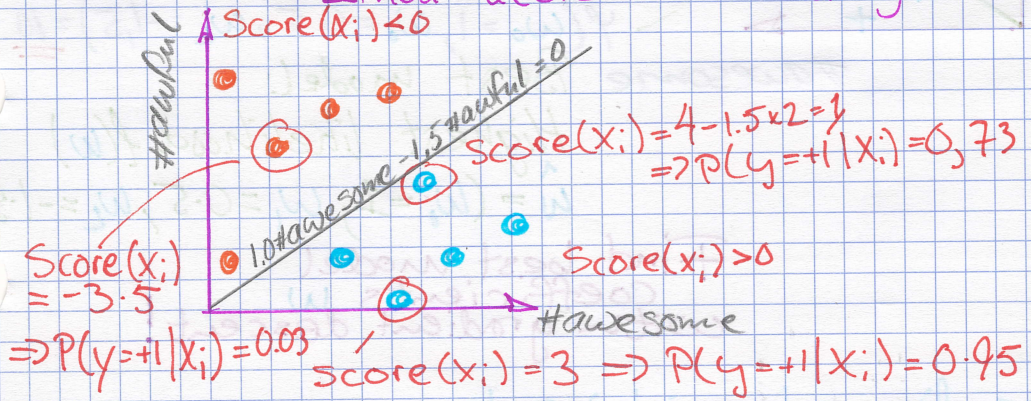
$$P(y=+1 \mid x, w) = \frac{1}{1 + e^{-w^T h(x)}}$$

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

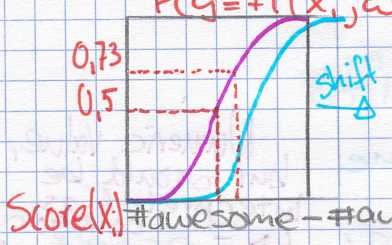


Score(x_i)	$P(y=+1 \mid x_i, w)$
0	0.5
-2	0.12 < 0.5 ⇒ ŷ=-1
2	0.88
4	0.98 ⇒ ŷ=+1

Logistic regression → Linear decision boundary

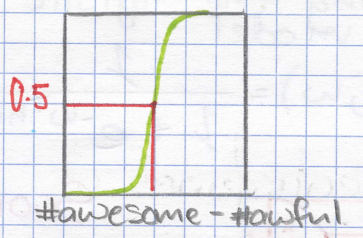


Effect of coefficients on logistic regression model



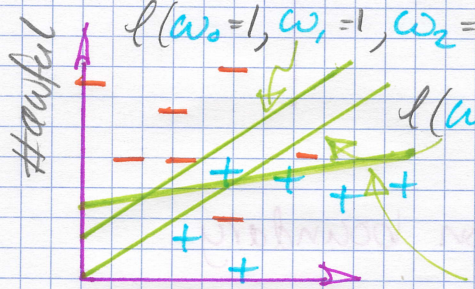
w_0	-2	w_0	0
$w_{\#awesome}$	+1	$w_{\#awesome}$	+1
$w_{\#awful}$	-1	$w_{\#awful}$	-1

w_0 0
$w_{\#awesome}$ +3
$w_{\#awful}$ -3



Overview of learning logistic regression model

Find "best" classifier = Maximize quality metric over all possible w_0, w_1, w_2



$\ell(w_0=1, w_1=1, w_2=-1.5) = 10^{-5}$ Likelihood $\ell(w)$

$\ell(w_0=0, w_1=1, w_2=-1.5) = 10^{-6}$

$\ell(w_0=1, w_0=0.5, w_2=-1.5) = 10^{-4}$

Best model: Highest likelihood $\ell(w)$

$\hat{w} = (w_0=1, w_1=0.5, w_2=-1.5)$

Find best model coefficients w with gradient descent!

Encoding categorical inputs

Categorical inputs

- Numeric inputs:
  - #awesome, age, salary, ...
  - Intuitive when multiplied by coefficient
    - e.g. 1.5 #awesome
- Categorical inputs:
  - Gender <male, female>
  - Country of birth

Numeric value, but should be interpreted as category
↳ Zipcode

How do we multiply category by coefficient?
Must convert categorical inputs into numeric features

Encoding categories as numeric features

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

1-hot encoding

196 categories

x	Argentina $h_1(x)$	Brazil $h_2(x)$	...	... $h_{195}(x)$	Zimbabwe $h_{196}(x)$
Brazil	0	1	0	0	0
Zimbabwe	0	0	0	0	1

196 features

x = Restaurant reviews (text data)

bag of word

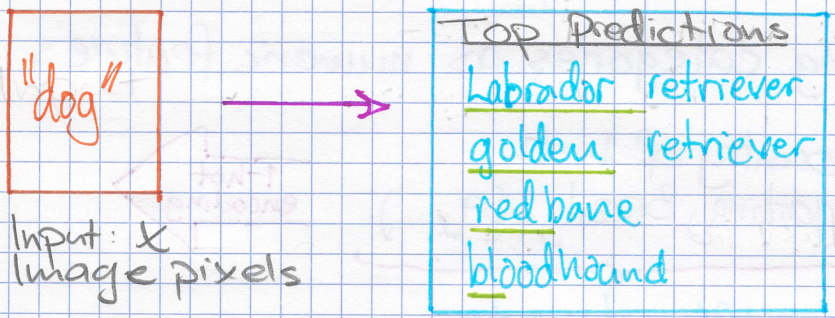
10,000 words in vocabulary

x	#awesome $h_1(x)$	#awful $h_2(x)$	...	$h_{9999}(x)$	#Sushi $h_{10000}(x)$
Text	2	0	0	0	3

10,000 features

Multiclass classification using 1 versus all

Multiclass classification

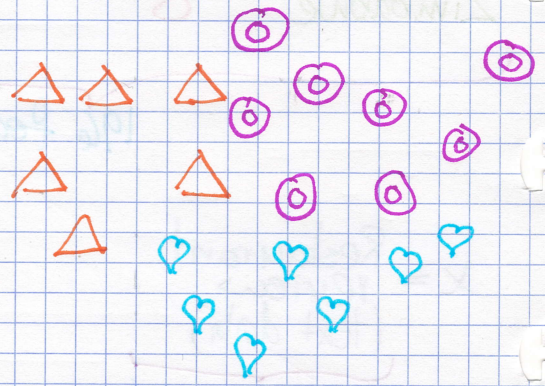


Input: x
Image pixels

Output: y
Object in image

Multiclass classification formulation

- C possible classes
  - y can be $1, 2, \ldots, C$
- N datapoints:



Data point	x[1]	x[2]	y
x_1, y_1	2	1	△
x_2, y_2	0	2	♡
x_3, y_3	3	3	◎
x_4, y_4	4	1	◎

Learn:

$\hat{P}(y = \triangle \mid x)$

$\hat{P}(y = \heartsuit \mid x)$

$\hat{P}(y = \text{◎} \mid x)$

1 versus all:

Estimate $\hat{P}(y=\triangle|x)$ using 2-class model

+1 class: points with $y_i = \triangle$

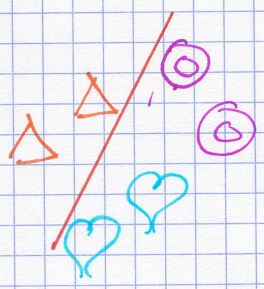
−1 class: points with $y_i = \heartsuit$ OR $\odot$

Train classifier: $\hat{P}_{\triangle}(y=+1|x)$

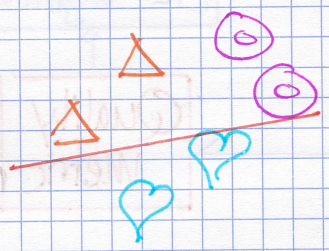
Predict: $\hat{P}_{\triangle}(y=+1|x_i) = \hat{P}(y=\triangle|x_i)$

1 versus all: Simple multiclass classification using C 2-class models

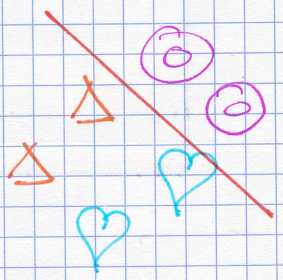
$\hat{P}(y=\triangle|x_i) = \hat{P}_{\triangle}(y=+1|x_i, w_{\triangle})$

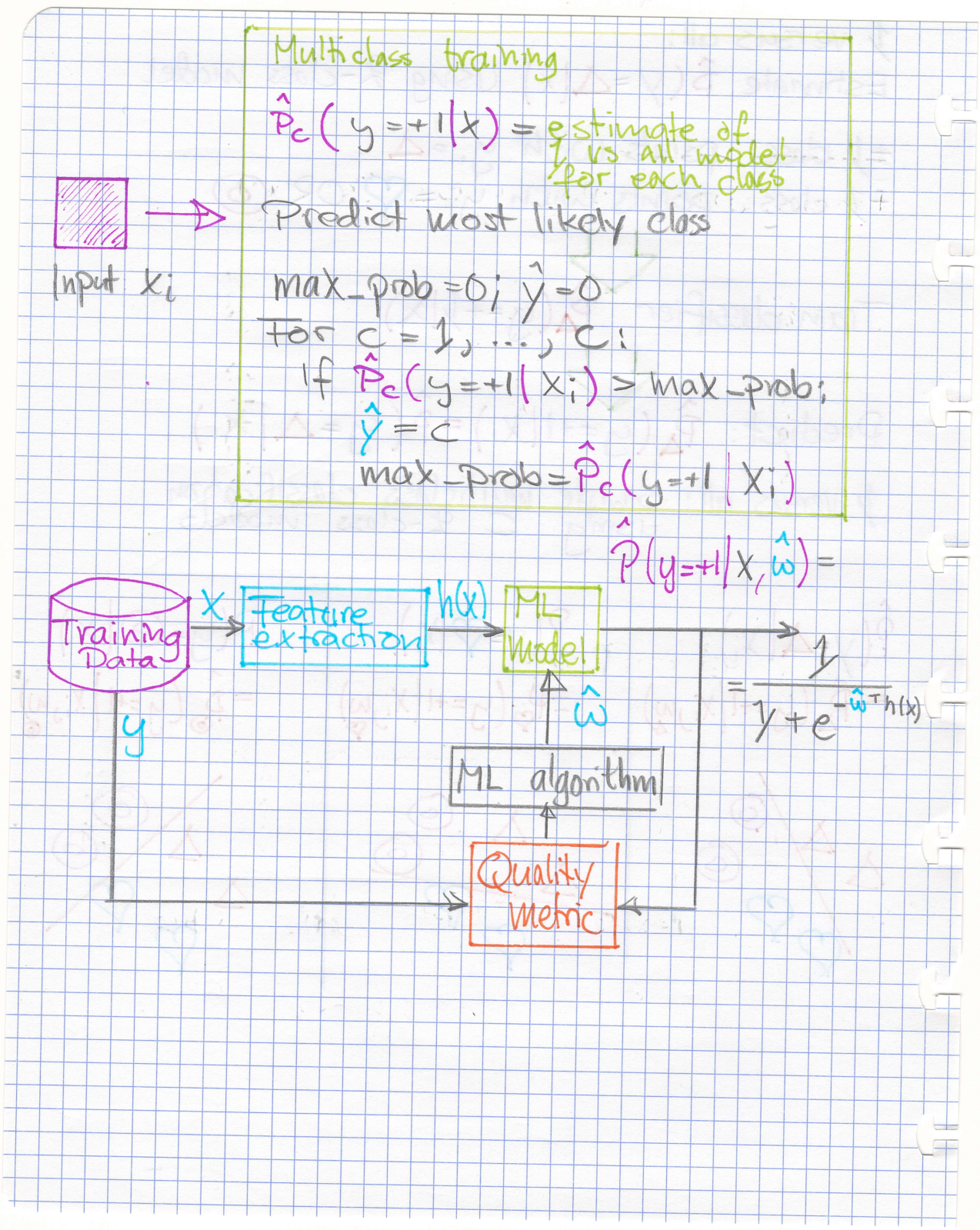


$\hat{P}(y=\heartsuit|x_i) = \hat{P}_{\heartsuit}(y=+1|x_i, w_{\heartsuit})$



$\hat{P}(y=\odot|x_i) = \hat{P}_{\odot}(y=+1|x_i, w_{\odot})$




Multiclass training
$\hat{P}_c(y=+1|x)$ = estimate of 1 vs all model for each class
Predict most likely class
max_prob = 0; $\hat{y}$ = 0
for c = 1, ..., C:
if $\hat{P}_c(y=+1|x_i)$ > max_prob:
$\hat{y}$ = c
max_prob = $\hat{P}_c(y=+1|x_i)$
input x_i
$\hat{P}(y=+1|x,\hat{w})$ =
$= \frac{1}{1+e^{-\hat{w}^T h(x)}}$
Training Data
x
Feature extraction
h(x)
ML model
$\hat{w}$
ML algorithm
Quality metric
y