

CSC380: Principles of Data Science

Basic machine learning 2

Naren Doraiswamy

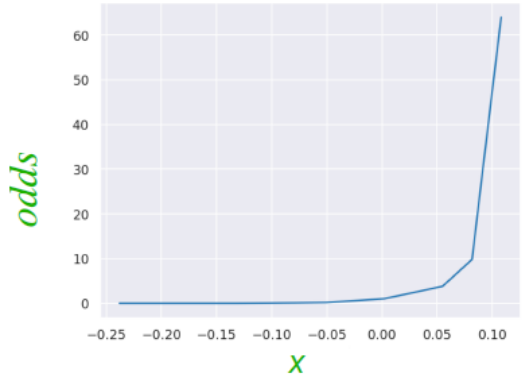
Outline

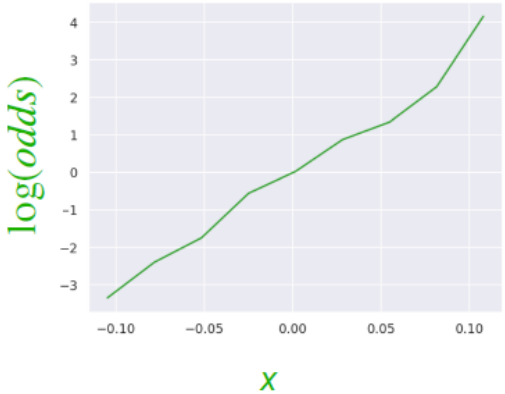
- Classification basics
- Nearest neighbor Classification
- Logistic regression

Where logistic function come from?

- $w \cdot x + b$ is unbounded
- We want to transform p into a form that produce unbounded outputs
  - $p \rightarrow \frac{p}{1-p} \rightarrow \ln \frac{p}{1-p}$

[0, 1] [0, +∞] [-∞, +∞]





Where logistic function come from?

Where does the logistic function come from?

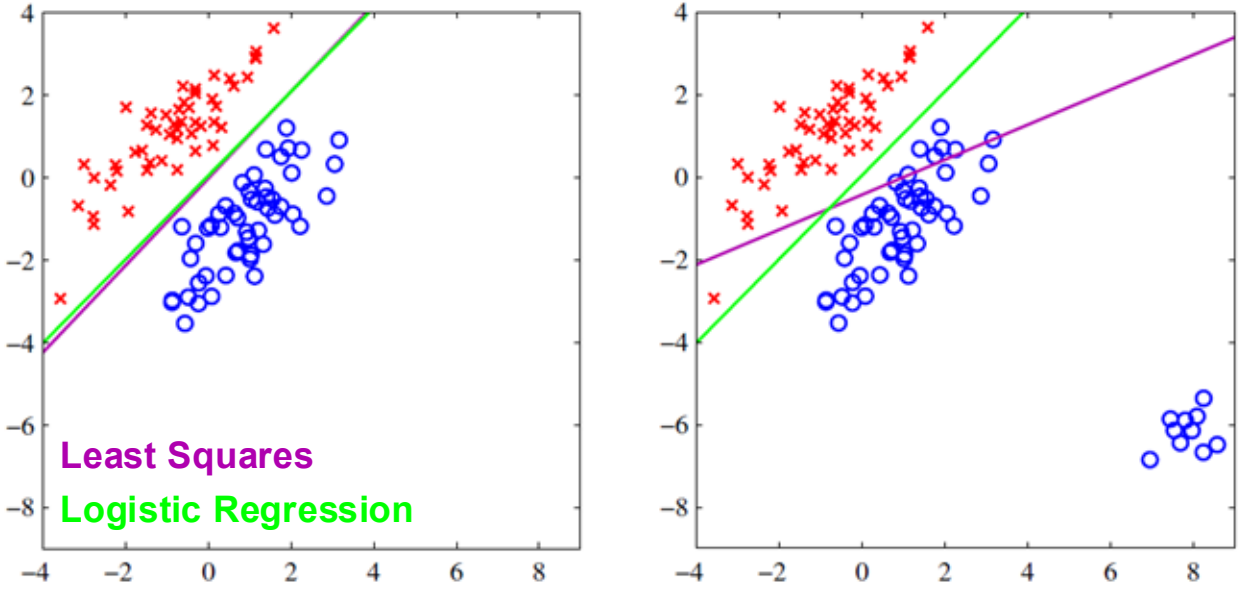
- $w \cdot x + b$ is good at predicting unbounded outputs
- $\log \text{odd} = \ln \frac{p}{1-p}$
- This now can take +/- values

$$\ln \frac{P(y=1|x)}{P(y=0|x)} = \ln \frac{p}{1-p} = w \cdot x + b$$

$$\Rightarrow \frac{p}{1-p} = e^{w \cdot x + b}$$

$$\Rightarrow p = \frac{1}{1 + e^{-(w \cdot x + b)}}$$

Linear Regression vs. Logistic Regression



Linear regression may also be used for classification

- To predict a class, threshold $w \cdot x + b$ against 0.5 (labels are 0 and 1)
- Not designed for classification though -- sensitive to outliers

[Source: Bishop “PRML”]

Using Logistic Regression

Logistic Regression have two main usages

- building **predictive** classification models
- **understanding** how features relate to data classes / categories

Example South African Heart Disease (Hastie et al. 2001)
Data result from Coronary Risk-Factor Study in 3 rural areas of South Africa. Data are from white men 15-64yrs. Label is presence/absence of *myocardial infarction (MI).*

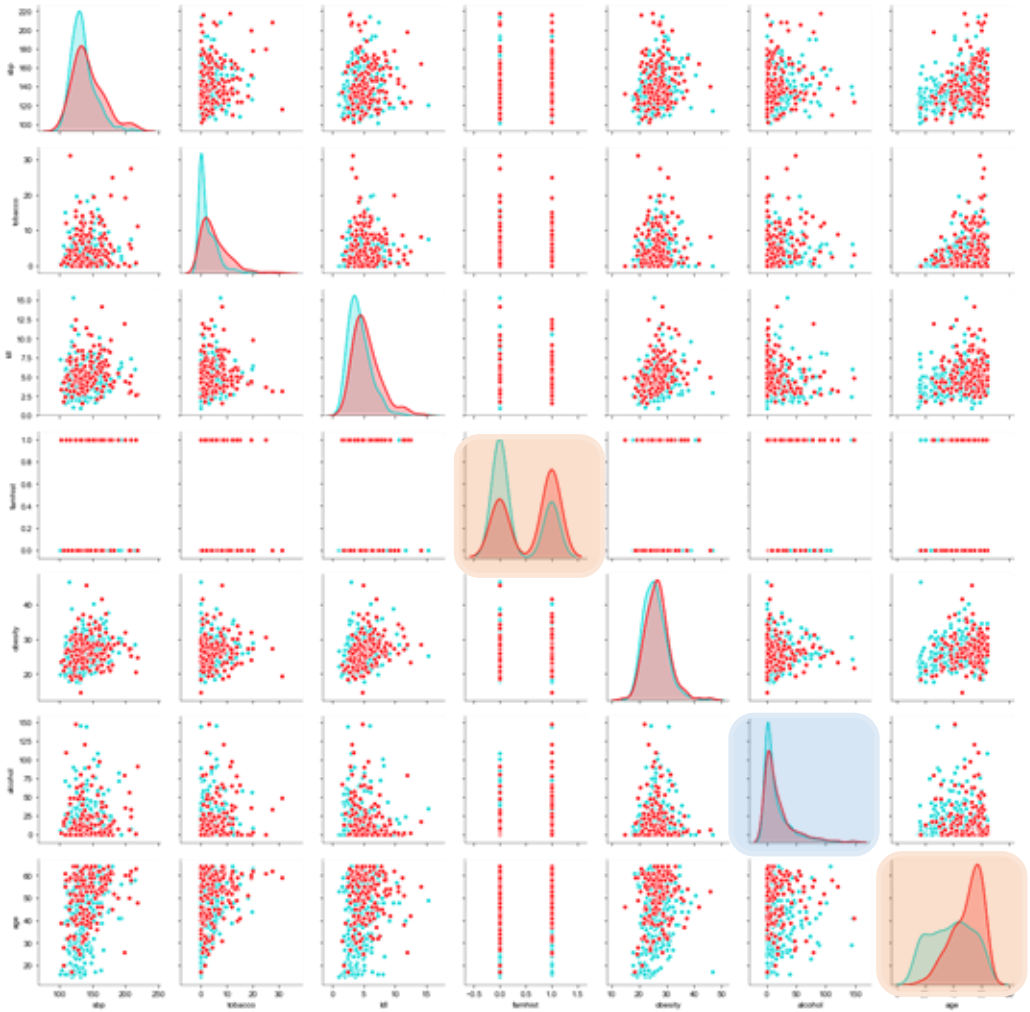
Example: African Heart Disease

	sbp	tobacco	ldl	famhist	obesity	alcohol	age	chd
0	160	12.00	5.73	1	25.30	97.20	52	1
1	144	0.01	4.41	0	28.87	2.06	63	1
2	118	0.08	3.48	1	29.14	3.81	46	0
3	170	7.50	6.41	1	31.99	24.26	58	1
4	134	13.60	3.50	1	25.99	57.34	49	1

Features

- Systolic blood pressure
- Tobacco use
- Low density lipoprotein (ldl)
- Family history (discrete)
- Obesity
- Alcohol use
- Age

Q: How predictive is each of the features to myocardial infarction?



Looking at Data

Each scatterplot shows pair of risk factors.

Cases **with** MI (**red**) and **without** (cyan)

Features

- Systolic blood pressure
- Tobacco use
- Low density lipoprotein (ldl)
- Family history (discrete)
- Obesity
- Alcohol use
- Age

[Source: Hastie et al. (2001)]

Example: African Heart Disease

	Coefficient	Std. Error	Z Score
(Intercept)	−4.130	0.964	−4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	−1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

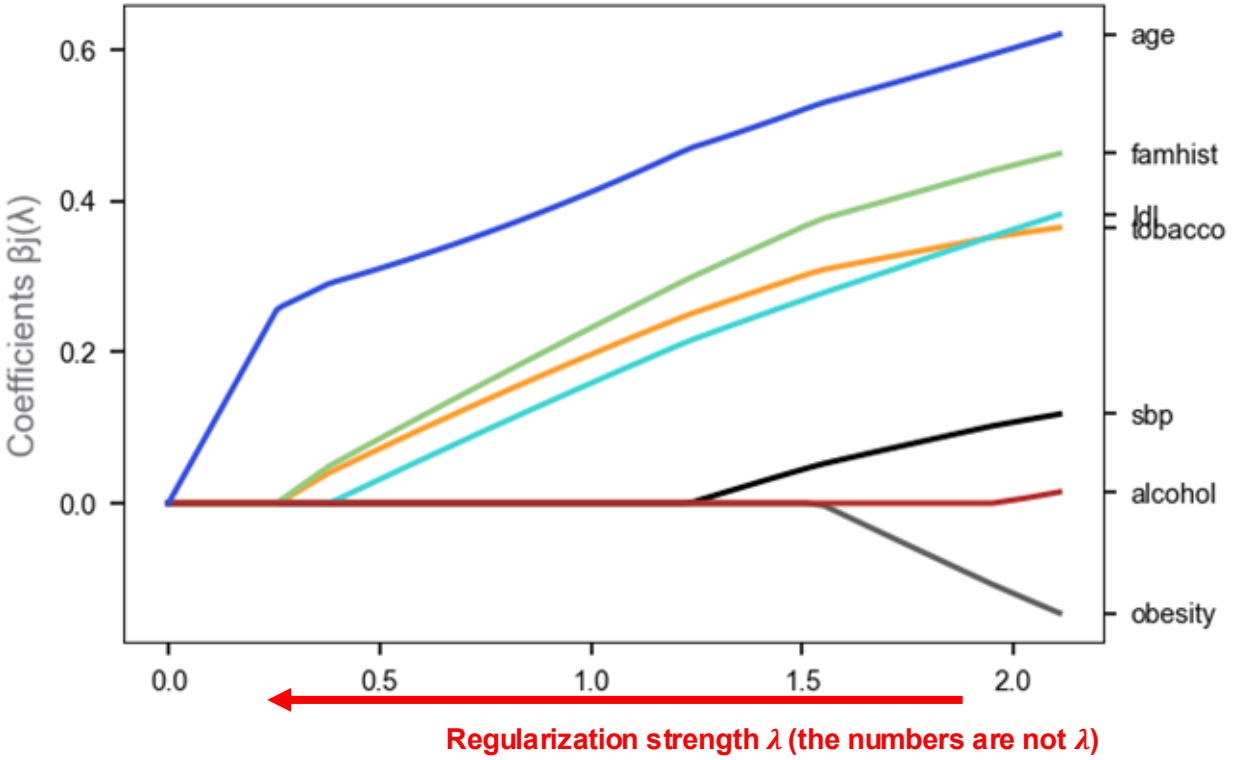
Z-score: score indicating the significance of each feature

|Z-score| > 2: feature is significant (we will learn more in *statistics*)

Finding Systolic blood pressure (sbp), obesity, and alcohol are **not significant predictors**

L1 regularized logistic regression coefficients

$$\text{argmin}_w \text{ Logistic_Loss}(w) + \lambda|w|_1$$



https://github.com/empathy87/The-Elements-of-Statistical-Learning-Python-Notebooks/blob/master/examples/South%20African%20Heart%20Disease.ipynb

Evaluation measures for classification

Pipeline for machine learning & data analysis

1. Choose a model — How should we represent the world?

2. Choose a loss function — How do we quantify prediction error?

3. Fit the model — How do we choose the best parameters of our model given our data?

4. Evaluate model performance — How do we evaluate whether this process gave rise to a good model?

Model evaluation: baseline models

Baseline models: simplest reasonable models to compare against

- Uniform classifier: predict with one label consistently (i.e., Most common label classifier, Least common label classifier)
- Blind classifier: predict random labels

	Features				Label	Most common Prediction	Least common Prediction	Random Prediction
Training data	x_{11}	x_{12}	$\cdots$	x_{1p}	Yes	Yes	No	Yes
	x_{21}	x_{22}	$\cdots$	x_{2p}	No	Yes	No	No
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$			
Test data	x_{n1}	x_{n2}	$\cdots$	x_{np}	Yes	Yes	No	Yes

Classification: model evaluation

Confusion matrix

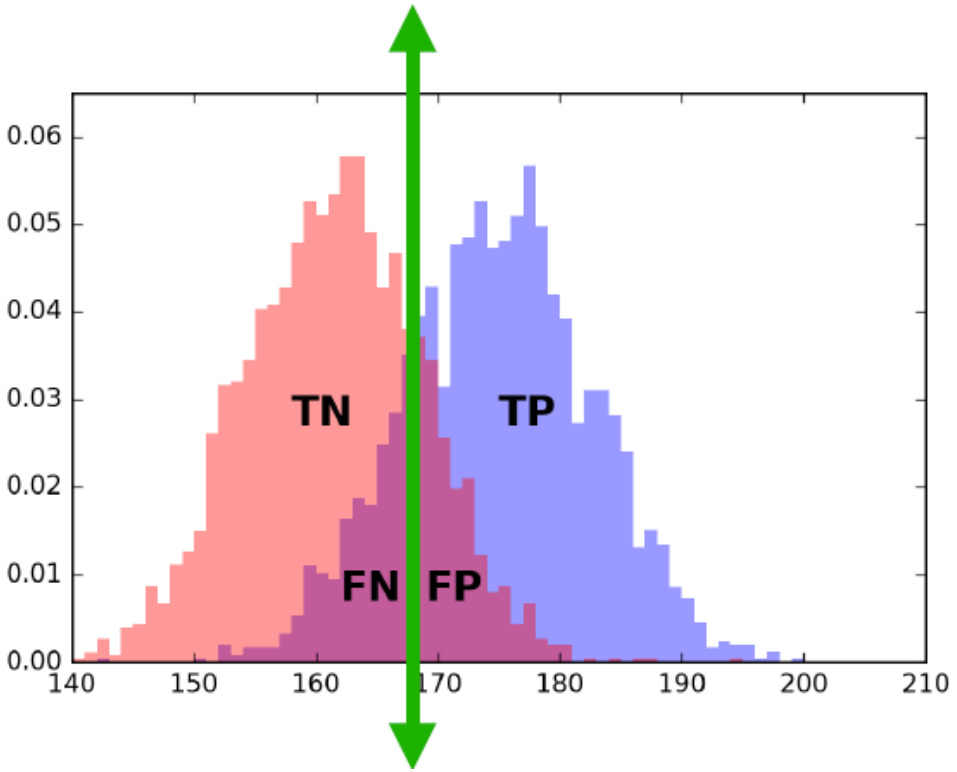
Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

- True positives (TP): + labeled as +
- True negatives (TN): - labeled as –
- False positives (FP): - labeled as +
- False negatives (FN): + labeled as -

Classification: model evaluation

- Example: classify everyone of height ≥168cm as male (1)



Fill out the tables: TP, TN, FP, FN

Cancer diagnosis. Positive class 5%, consider 100 data points.

- Left: Blind classifier (random label)
- Middle: Most common label classifier (always negative)
- Right: Least common label classifier (always positive)

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	2.5	47.5
Predicted Negative (0)	2.5	47.5

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	0	0
Predicted Negative (0)	5	95

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	5	95
Predicted Negative (0)	0	0

Accuracy

Accuracy ratio of correct predictions over all predictions

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} = P(f(x) = y)$$

Case 1: classes balanced, Positive class 50%

- Blind classifier accuracy = 50%
  - Predict random label
- Most common label classifier accuracy = 50%
  - Predict majority class (doesn't matter 1 or 0)

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Fill out the tables: TP, TN, FP, FN

Cancer diagnosis. Positive class 50%, consider 100 data points.

- Left: Blind classifier (random label)
- Middle: Most common label classifier (doesn't matter positive or negative , let's say always negative)

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	25	25
Predicted Negative (0)	25	25

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	0	0
Predicted Negative (0)	50	50

Fill out the tables: TP, TN, FP, FN

Cancer diagnosis. Positive class 5%, consider 100 data points.

- Left: Blind classifier (random label)
- Middle: Most common label classifier (always negative)

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	2.5	47.5
Predicted Negative (0)	2.5	47.5

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	0	0
Predicted Negative (0)	5	95

Accuracy

Accuracy ratio of correct predictions over all predictions

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} = P(f(x) = y)$$

Case 2: classes are imbalanced, Positive class 5%

- Blind classifier accuracy = 50%
- Most common label classifier accuracy = 95%
  - Always predict 0

Accuracy can be misleading in the presence of class imbalance!

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Precision

Precision ratio of correct positive predictions over positive predictions

$$\text{Precision} = \frac{TP}{TP + FP} = P(y = 1 \mid f(x) = 1)$$

Example cancer diagnosis. Positive class 5%

Blind classifier precision = $P(y = 1) = 5\%$

Most common label classifier precision

= $P(y = 1 \mid f(x) = 1) =$ undefined

Least common label classifier precision

= $P(y = 1) = 5\%$

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Precision

Cancer diagnosis. Positive class 5%, 100 data points

- Left: Blind classifier
- Middle: Most common label classifier
- Right: Least common label classifier

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	2.5	47.5
Predicted Negative (0)	2.5	47.5

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	0	0
Predicted Negative (0)	5	95

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	5	95
Predicted Negative (0)	0	0

Recall

Recall ratio of correct positive predictions over all positive instances

$$\text{Recall} = \frac{TP}{P} = \frac{TP}{TP + FN} = P(f(x) = 1 \mid y = 1)$$

Example cancer diagnosis. Positive class 5%

Blind classifier recall $= P(f(x) = 1) = 50\%$

f(x) and y are independent

Most common label classifier recall

$= P(f(x) = 1 \mid y = 1) = 0$

Least common label classifier recall

$= P(f(x) = 1 \mid y = 1) = 100\%$

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Precision vs Recall

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Recall} = \frac{TP}{TP+FN}$$

Are false positives more important than false negatives?

Diagnostic screening for cancer: Recall is more important

FP: OK FN: Not OK (cancer not caught)

Spam filtering: Precision is more important

FP: Not OK (important email marked spam) FN: OK

F1-score

F1-score Harmonic mean of precision and recall

$$\mathrm{F1} = \frac{2\ \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

F1 score high => need both precision and recall to be high

	0.0	0.2	0.4	0.6	0.8	1.0
0.0	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.00	0.20	0.26	0.30	0.32	0.33
0.4	0.00	0.26	0.40	0.48	0.53	0.57
0.6	0.00	0.30	0.48	0.60	0.68	0.74
0.8	0.00	0.32	0.53	0.68	0.80	0.88
1.0	0.00	0.33	0.57	0.74	0.88	1.00

Table 5.2: Table of f-measures when varying precision and recall values.

Summary of recall & precision

- Class imbalanced ⇒ accuracy is misleading.
- High recall is easy to achieve
  - just predict least common label always
- High precision is hard to achieve in imbalanced classification
- F-score does the best job of any single statistic, but all four work together to describe the performance of a classifier.

Multiclass classification

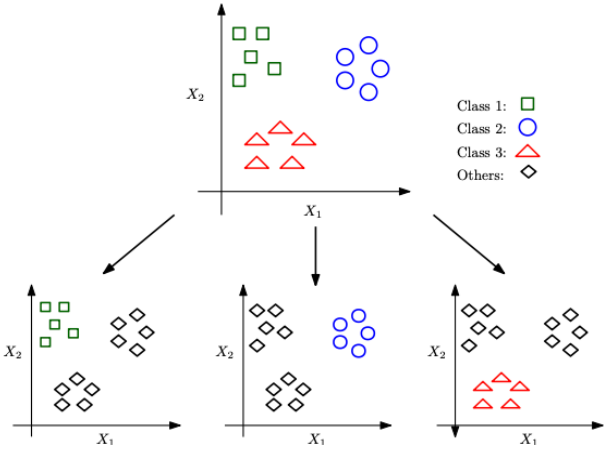
Multiclass classification

- Multiclass classification naturally happens in many real-world applications
  - E.g. movies: “drama”, “action”, “documentary”
  - E.g. objects: “automobile”, “airplanes”, “trees”

- Some of our methods are designed for binary classification, e.g. logistic regression
  - How to extend them to multiclass prediction?

Approach 1: reducing multiclass to binary

The “One-vs-rest” approach:



Training

For each class i:

Train f_i that classifies class i against the rest

Test

For test example x:

Use each f_i to predict its probability of belonging class i

Pick class i that has the highest probability

Alternative: “all pairs” approach

Approach 2: modifying the learning algorithm

Case study How to extend logistic regression to multiclass?

The multinomial logistic regression model:

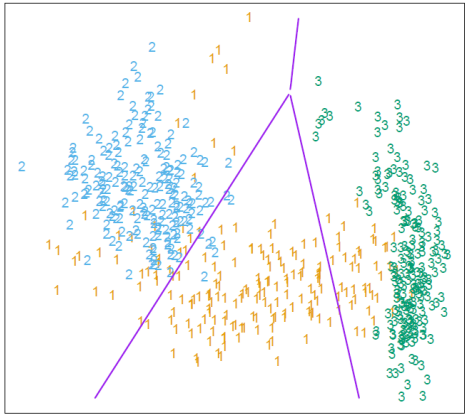
$p(C = i \mid x)$'s sum to 1

$$\log \frac{p(C = 1 \mid x)}{p(C = K \mid x)} = w_1^T x$$

$$\log \frac{p(C = 2 \mid x)}{p(C = K \mid x)} = w_2^T x$$

$$\vdots$$

$$\log \frac{p(C = K - 1 \mid x)}{p(C = K \mid x)} = w_{K-1}^T x$$



Can define multiclass cross entropy loss accordingly, and minimize it

Comparison

- Reducing multiclass to binary
  - Pros: simple & general – binary classification algorithm as ‘black box’
- Modifying the learning algorithm
  - Pros: can exploit the structure of the problem more

sklearn.linear_model.LogisticRegression

class sklearn.linear_model.LogisticRegression(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

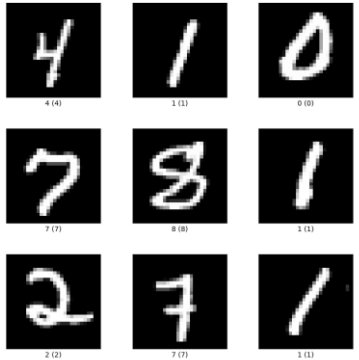
multi_class=‘ovr’ implements one-vs-rest

multi_class=‘multinomial’ implements multinomial

Multiclass classification: performance metrics

Multiclass confusion matrix

Example Optical Character Recognition



Predicted label $f(x)$

Actual label y

Digits	0	1	2	3	4	5	6	7	8	9
0	351	0	5	4	2	7	2	1	6	0
1	0	254	0	0	2	0	0	1	1	2
2	1	1	166	4	5	1	3	2	2	1
3	1	2	4	142	0	5	0	1	4	0
4	3	3	8	1	180	3	2	5	4	4
5	0	0	3	11	0	140	3	0	7	1
6	0	2	2	0	4	0	158	0	1	0
7	0	0	2	2	1	0	0	132	2	1
8	2	1	8	0	0	0	2	1	137	1
9	1	1	0	2	6	4	0	4	2	167

Multiclass classification: performance metrics

We can precision & recall separately for each class i

$$\text{Precision}_i = P(\, y = i \mid f(x) = i \,) = \frac{C[i,i]}{\sum_j C[j,i]}$$

$$\text{Recall}_i = P(\, f(x) = i \mid y = i \,) = \frac{C[i,i]}{\sum_j C[i,j]}$$

Example

$$\text{Precision}_0 = \frac{351}{359}$$

$$\text{Recall}_0 = \frac{351}{378}$$

Predicted label $f(x)$

Digits	0	1	2	3	4	5	6	7	8	9
0	351	0	5	4	2	7	2	1	6	0
1	0	254	0	0	2	0	0	1	1	2
2	1	1	166	4	5	1	3	2	2	1
3	1	2	4	142	0	5	0	1	4	0
4	3	3	8	1	180	3	2	5	4	4
5	0	0	3	11	0	140	3	0	7	1
6	0	2	2	0	4	0	158	0	1	0
7	0	0	2	2	1	0	0	132	2	1
8	2	1	8	0	0	0	2	1	137	1
9	1	1	0	2	6	4	0	4	2	167

Actual label y