Lecture 6

Classification Analysis

Ashish Dandekar

Lecture Overview

Classification and its evaluation

Confusion Matrix

ROC Curve

Linear Models

Linear Separator

Logistic Regression

Classification Analysis

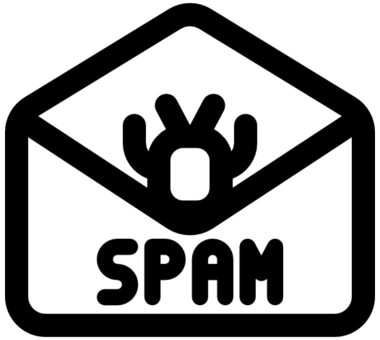
Classification

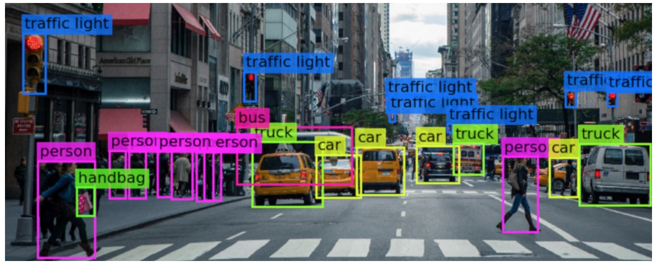
- ❯ Introduction
- Confusion matrix
- Popular metrics
- ROC Curve
- Multiclass evaluation

Classification is the task of learning a target function f that maps each datapoint x to a class label y.

- Binary classification.
- Multi-class classification
- Multi-label classification
- Outlier detection

PPT回放，adverse？





Confusion Matrix

ground truth label

predicted label	1	0
1	True Positives (TP)	False Positives (FP)
0	False Negatives (FN)	True Negatives (TN)

- True Positives (TP). The positive elements that are correctly classified.
- False Positives (FP). The negative elements that are incorrectly classified.
- True Negatives (TN). The negative elements that are correctly classified.
- False Negatives (FN). The positive elements that are incorrectly classified.

Classification Metrics

Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN}$$

$\hat{y}$ \ y	1	0
1	TP	FP
0	FN	TN

Sensitivity

$$\frac{TP}{TP + FN}$$

$\hat{y}$ \ y	1	0
1	TP	FP
0	FN	TN

Specificity

$$\frac{TN}{TN + FP}$$

$\hat{y}$ \ y	1	0
1	TP	FP
0	FN	TN

Classification Metrics

Precision

$$\frac{TP}{TP + FP}$$

y

$\hat{y}$	1	0
1	TP	FP
0	FN	TN

Recall

$$\frac{TP}{TP + FN}$$

y

$\hat{y}$	1	0
1	TP	FP
0	FN	TN

F1 Score

$$\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

y

$\hat{y}$	1	0
1	TP	FP
0	FN	TN

Why do we have so many metrics?

Class Imbalance

Many real-world datasets suffer from the class imbalance issue. For instance:

- In a dataset of infectious deceases, $1 - 5\%$ of the data comprises of infected persons.
- In a dataset for predictive maintenance, less than 2% of the data consists of anomalies.

Consider an example of the covid test that **always** returns negative. There are 10% infected people in the population.

$$Accuracy = \frac{90}{100} = 90\%$$

$$Specificity = \frac{90}{90} = 100\%$$

Why do we have so many metrics?

Qualitative analysis of FP and FN

Example 1. Consider a test to determine whether a person suffers from COVID. In this case, misclassifying a COVID positive person maybe considered worse than misclassifying a healthy person.

In this case, we need a classifier with $Recall > Precision$.

Example 2. Consider an image search engine where I search for the images with certain keywords. Let's say I am searching for images of cats. In this case, showing an image of a dog is worse that not showing all possible images of cats.

In this case, we need a classifier with $Precision > Recall$.

Thresholding

y	Scores	$\hat{y}_{0.5}$	$\hat{y}_{0.43}$
1	0.45	0	1
0	0.30	0	0
0	0.55	1	1
0	0.25	0	0
1	0.35	0	0
1	0.55	1	1

Different thresholds yield different answers!

Which threshold should we use?

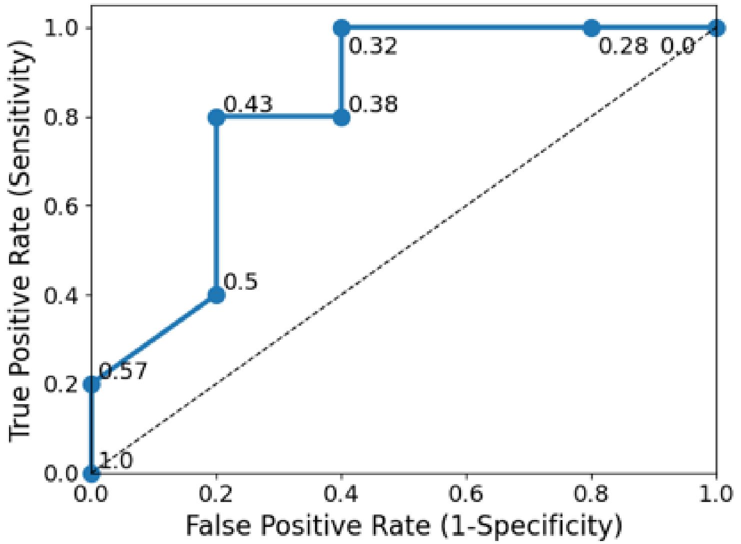
Threshold 0.5

	$y=1$	$y=0$
$\hat{y}=1$	1	2
$\hat{y}=0$	2	1

Threshold 0.43

	$y=1$	$y=0$
$\hat{y}=1$	2	2
$\hat{y}=0$	1	1

Receiver Operating Characteristics (ROC)



The curve is obtained by computing the metrics for various values of the thresholds for a classifier.

ROC Curve is a plot of True Positive Rate ($Sensitivity$) versus False Positive Rate ($1 - specificity$).

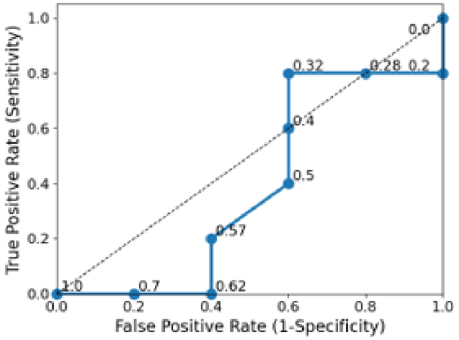
Area under the Curve (AUC/AUROC)

Area under the ROC curve is used as the quantifier of the performance of the classifier.

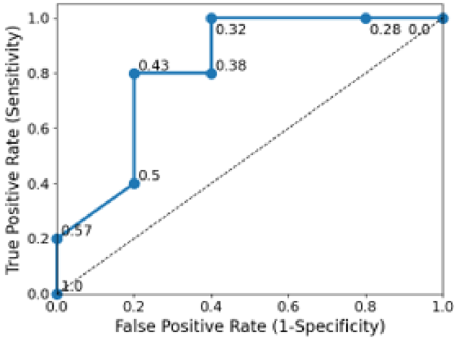
- AUROC of a random classifier is 0.5.

Receiver Operating Characteristics (ROC)

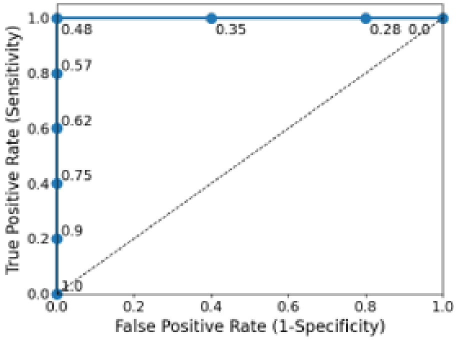
Poor Classifier



Good Classifier



Perfect Classifier



How to do it in Python?

```
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_true, y_score)
```

One-vs-Rest Confusion Matrices

ground truth label y / predicted label $\hat{y}$

$\hat{y}$ \ y	2	1	0
2	8	6	0
1	3	12	1
0	4	2	14

$$Acc = \frac{8 + 12 + 14}{8 + 12 + 14 + 6 + 3 + 4 + 2 + 1}$$

One-vs-Rest Confusion Matrices

$$\Downarrow$$

	2	$\bar{2}$
2	8	6
$\bar{2}$	7	29

	1	$\bar{1}$
1	12	4
$\bar{1}$	8	26

	0	$\bar{0}$
0	14	6
$\bar{0}$	1	29

Multiclass Evaluation

	2	$\bar{2}$
2	8	6
$\bar{2}$	7	29

	1	$\bar{1}$
1	12	4
$\bar{1}$	8	26

	0	$\bar{0}$
0	14	6
$\bar{0}$	1	29

Micro-Averaging

- Average individual TP, FP, TN, FN!
- Compute the metric on the aggregated confusion matrix.

	C	$\bar{C}$
C	11.33	5.33
$\bar{C}$	7	28

Multiclass Evaluation

	2	$\bar{2}$
2	8	6
$\bar{2}$	7	29

	1	$\bar{1}$
1	12	4
$\bar{1}$	8	26

	0	$\bar{0}$
0	14	6
$\bar{0}$	1	29

Macro-Averaging

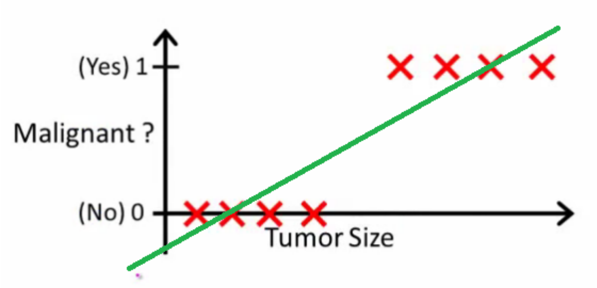
- Compute the metric on the individual matrices.
- Average the metric to compute the global metric!

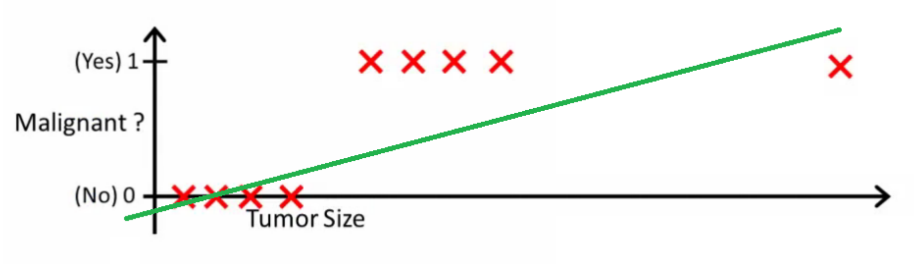
Question

What is better: macro-averaging or micro-averaging?

Linear Models

Can we use linear regression?





- Linear regression requires numerical output.
- It is equally sensitive to the unbalanced data.

Linear Separator

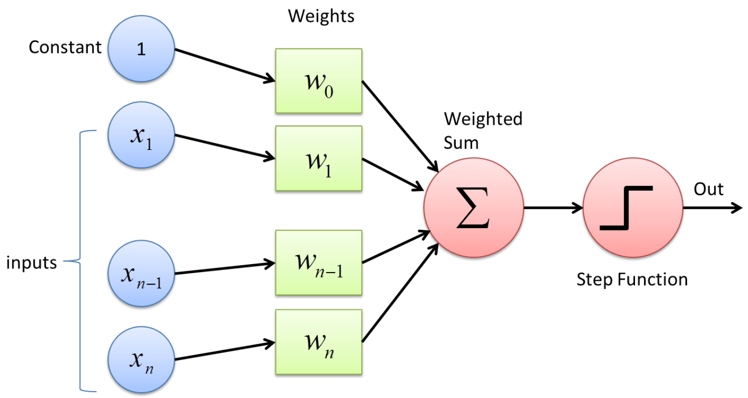
Given a labeled dataset $D = \{(x_i, y_i)\}$ of n points where $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$.

We want to find a linear separator $w_i \in \mathbb{R}^{d+1}$ such that:

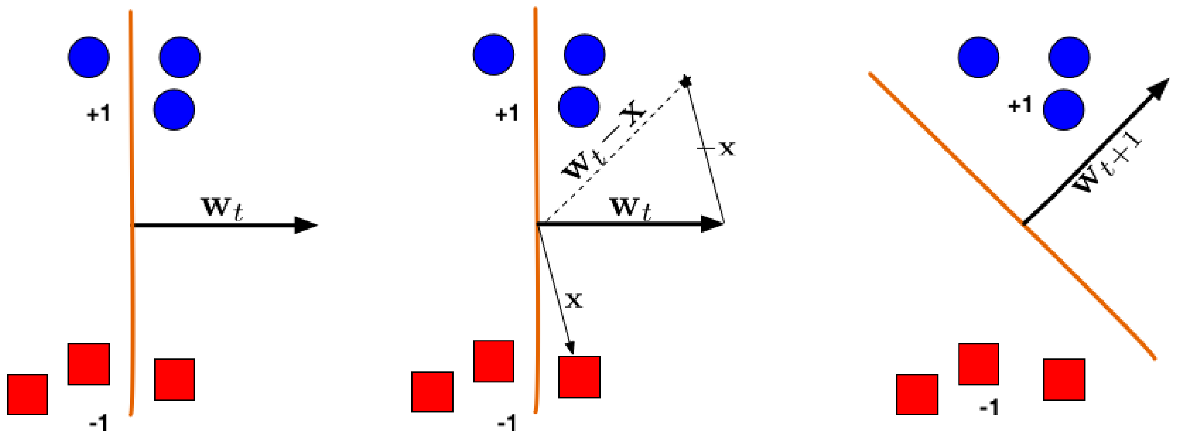
$$\hat{y} = sign(Xw)$$

Loss function (0-1 loss)

$$\ell_D(w) = \frac{1}{n}\sum_i \mathbb{I}(y_i \neq \hat{y}_i)$$



Perceptron Learning



1. Randomly initialise w_0.

2. For $t \in [1 \ldots T]$ and for each datapoint $i \in [1 \ldots n]$

$$w_{t+1} \leftarrow w_t + (y_i - sign(w^t x_i))x_i$$

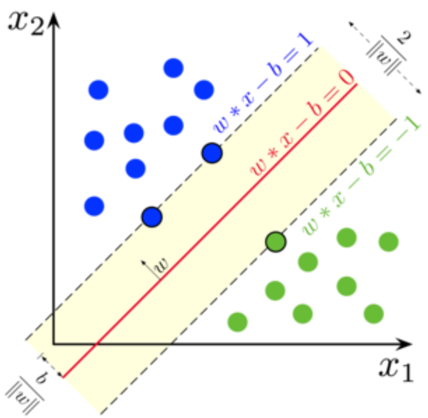
Support Vector Machine (Just an idea!)

Which linear separator to choose? - One that maximises the margin!

Hard-Margin

$$\min_{w} \| w \| , \quad \text{s.t.} \quad y_i(w^t x_i) \geq 1$$

The separator must satisfy every constraints.



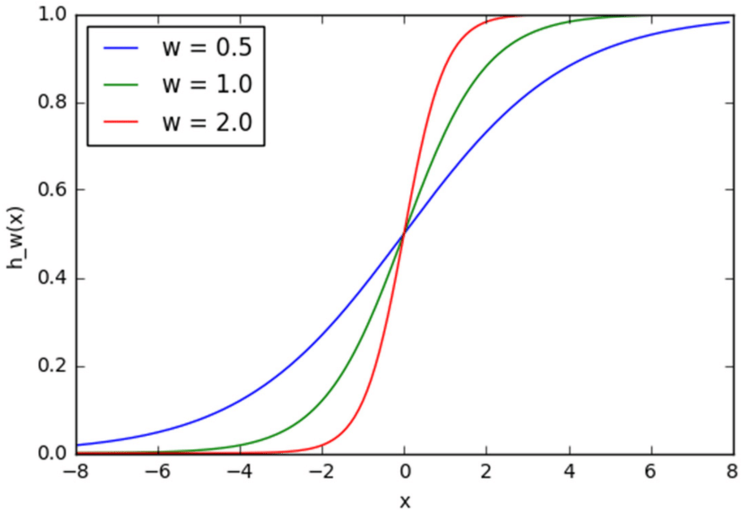
Soft-margin SVM

$$\min_{w} \frac{1}{n} \sum_{i} max(0, 1 - y_i(w^t x_i)) + \lambda \| w \|^2$$

Sigmoid Function

Sigmoid is a *smooth* function defined as follows:

$$h_w(x) = \frac{1}{1 + e^{-w^t x}} = \frac{exp(w^t x)}{1 + exp(w^t x)}$$



Why to use Sigmoid?

- Differentiability and convexity properties
- Probabilistic interpretation of the

Binary Logistic Regression

Given a labeled dataset $D = \{(x_i, y_i)\}$ of n points where $x_i \in R^d$, $y_i \in \{0, 1\}$.

Let p_i denotes the proability of item i to be labeled as 1.

Hypothesis

Binary logistic regression works on the following hypothesis:

$$p = \sigma(X\mathbf{w}) = \frac{exp(X\mathbf{w})}{1 + exp(X\mathbf{w})}$$

where $X \in R^{n \times (d+1)}$ is the data matrix and $\mathbf{w} \in R^{d+1}$.

Thresholding

The actual labels can be assigned to the datapoints by *tuning* a threshold (α) for the probabilistic output:

$$y = \begin{cases} 0 & \sigma(w^t x) < \alpha \\ 1 & otherwise \end{cases}$$

Likelihood

Likelihood over the dataset (D) can be computed as follows:

$$L(y \mid X, \mathbf{w}) = \prod_{i=1}^{n} p^{y_i}(1-p)^{1-y_i}$$

Taking log on both sides:

$$\ell_D(\mathbf{w}) = \sum_{i=1}^{n} y_i \log p + (1-y_i)\log(1-p)$$

Thus the machine learning problems becomes to estimate **w** that maximises the likelihood. It is typically solved using optimisation technique such as *Gradient Descent*.

$$\hat{\mathbf{w}} = \arg\max_{w} \ell_D(\mathbf{w})$$

Odds-Ratio

Odds ratio for an event is defined as follows:

$$odds = \frac{\text{Probability that event happens}(p)}{\text{Probability that event does not happen}(1-p)}$$

- Odds of getting a head are 1:1
- Odds of getting a spade are 1:3
- Odds of finding an employee at work place are 5:2

Odds-Ratio

Rearranging the hypothesis for the logistic regression:

$$p = \frac{exp(Xw)}{1 + exp(Xw)}$$

$$exp(Xw) = \frac{p}{1 - p}$$

$$Xw = log\left(\frac{p}{1 - p}\right)$$

Thus log-odds for a datapoint i are given as,

$$log\left(\frac{p_i}{1 - p_i}\right) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d$$

Interpreting Logistic Regression

We train a logistic regression on the dataset of graduate program acceptance in a univeristy. We find that:

$$log\left(\frac{p}{1-p}\right) = -1.34 - 0.56(\text{Gender}) + 1.13(\text{Score})$$

where p is the probability of getting accepted in the program.

Predictor	Odds	Interpretation
Gender	$e^{-0.56} = 0.57$	Odds of men getting accepeted is 43% lower
Score	$e^{1.13} = 3.09$	Odds of getting accepted are 209% higher for every extra GPA

Interpreting Logistic Regression

(5 points) Suppose we have a dataset of 10 patients with their blood pressure and a target variable that represents whether the patient has a heart disease or not (represented as 1 for disease and 0 for no disease). The dataset is as follows:

Blood Pressure	**Heart Disease**
135	1
132	1
120	0
140	1
110	0

Blood Pressure	**Heart Disease**
115	0
128	1
135	0
125	1
120	0

Any blood pressure that is higher than 130 is labeled as high and otherwise as normal. We fit a logistic regression model to classify whether a person has heart disease based on the blood pressure. It follows the following hypothesis:

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

where p denotes the probability of having the heart disease. We encode the blood pressure variable as 1 when it is high and 0 if it is normal. Compute the values b_0 and b_1 by hand. You may keep the solutions in their transcendental forms. For instance: You may use $e^0.3$ or $\log 0.6$ instead of computing their actual values.

Thank you!

Feel free to reach out to me at

dcsashi (at) nus (dot) edu (dot) sg