Data Science

Molin Liu

December 18, 2019

Introduction

This is a review work for the Data Science course in *Univeristy of Glasgow*.

1 Linear Algebra

1.1 Vector

Vectors can be composed (via addition), compared (via norms/inner products) and weighted (by scaling).

1.1.1 Basic Vector Operations

(Ommited)

1.1.2 Inner Product

$$\cos\theta = \frac{\mathbf{x} \bullet \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$

- The inner product of two orthogonal vectors is 0;

1.1.3 Norm

- Norm 0: Count of non-zero values;
- Norm 1: Sum of absolute values;
- Norm 2: Euclidean distance;

1.1.4 High Dimensional Vector Space

1.2 Matrix

1.2.1 Operations

(Ommited)

1.2.2 Anatomy of Matrix

1.3 Polynomial Regression

The **Polynomial Regression** can be written as:

$$t = w_0 + w_1 x + w_2 x^2 + w_3 x^2 + \ldots + w_K x^K = \sum_{k=0}^{\kappa} w_k x^k$$

Define the loss funcion:

$$\mathcal{L} = \frac{1}{N}(\mathbf{t} - \mathbf{X}\mathbf{w})^\top (\mathbf{t} - \mathbf{X}\mathbf{w})$$

1.3.1 Generalization & Overfitting

We can find out that the **loss** will always decrease as the model is made more complex.
How to choose the right model complexity? **Cross-validation**

1.3.2 Cross-Validation

2 Classification

The **Classification** task is to classify a set of *N* objects x_i with attributes. Each object has an associated label t_i

Probabilistic classifier produce a probability of class membership:
$P(t_{\text{new}} = k | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$
non-Probabilistic classifier produce a hard assignment:
$t_{new} = 1$ or $t_{new} = 0$

2.1 KNN

K-Nearest Neighbours(KNN)
- Non-probabilistic classifier;
- Supervised trainning;
- Fast;
- We can use CV to find the right K;

2.1.1 Problem

- As K increases, the small classes will disppear.

2.2 Logistic Regression

2.3 SVM

2.3.1 Hard Margin

If the training data is linearly seperable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible.

These hyperplanes can be described by the equations:

$$\vec{w} \cdot \vec{x}_i - b \geq 1, \text{ if } y_i = 1$$

or

$$\vec{w} \cdot \vec{x}_i - b \leq -1, \text{ if } y_i = -1$$

We can easily infer that

$$y_i \left(\vec{w} \cdot \vec{x}_i - b\right) \geq 1, \quad \text{for all } 1 \leq i \leq n$$

We want to maximise $\gamma = \frac{1}{\|\mathbf{w}\|}$, equivalent to minimising $\|\mathbf{w}\|$

Note: y_i is the label in the data, which is in $\{-1, 1\}$, rather than vertical axis value of the data.

2.3.2 Soft Margin

Soft-margin function for the data are not linearly seperable.

2.3.3 Inner Product

3 ROC

Sensitivity/Recall

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

Specificity

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

4 Unsupervised Learning

4.1 K-Means

4.2 Gaussion Mixture Model