Chapter 1

Linear Regression

1.1 Fitting a robust regression model using RANSAC

Algorithm 1: RANdom SAmple Consensus (RANSAC) algorithm

begin
  repeat
    Select a random number of examples to be inliers and fit the model;
    Test all other data points against the fitted model and add those points that fall within a user-given tolerance to the inliers;
    Refit the model using all inliers;
    Estimate the error of the fitted model versus the inliers;
  until *the performance meets a certain user-defined threshold or if a fixed number of iterations was reached*;

Chapter 2

Clustering

The goal of clustering is to find a natural grouping in data so that items in the same cluster are more similar to each other than to those from different clusters.

2.1 Prototype-based clustering

Prototype-based clustering means that each cluster is represented by a prototype, which is usually either the **centroid** (average) of similar points with continuous features, or the **medoid** (the most representative or the point that minimizes the distance to all other points that belong to a particular cluster) in the case of categorical features.

2.1.1 k-means clustering

Algorithm 2: The k-means algorithm

begin
    Randomly pick k centroids from the examples as initial cluster centers;
    repeat
        Assign each example to the nearest centroid, $\mu^{(i)}, j \in \{1, \dots, k\}$;
        Move the centroids to the center of the examples that were assigned to it;
    until *the cluster assignments do not change or a user-defined tolerance or maximum number of iterations is reached*;

Algorithm 3: The k-means++ algorithm

---

begin
- Initialize an empty set, M, to store the k centroids being selected;
- Randomly choose the first centroid from the input examples and $M \leftarrow \mu^{(j)}$;
- **repeat**
  - For each example, $\boldsymbol{x}^{(i)}$, that is not in M, find the minimum squared distance, $d(x^{(i)}, M)^2$, to any of the centroids in M;
  - To randomly select the next centroid, $\mu^{(p)}$, from a weighted probability distribution equal to $\frac{d(\mu^{(p)},\mathbf{M})^2}{\sum_i d(x^{(i)},\mathbf{M})^2}$;
- **until** *k centroids are chosen*;
- Proceed with the classic k-means algorithm;

2.1.2 k-means++

2.1.3 Hard versus soft clustering

Hard clustering describes a family of algorithms where each example in a dataset is assigned to exactly one cluster, as in the algorithm 2 and algorithm 3. In contrast, algorithms for **soft clustering** (sometimes also called **fuzzy clustering**) assign an example to one or more clusters. A popular example of soft clustering is the **fuzzy C-means (FCM)** algorithm (also called **soft k-means** or **fuzzy k-means**).

2.1.4 Fuzzy C-means

Algorithm 4: The FCM algorithm

---

begin
- Specify the number of k centroids and randomly assign the cluster memberships for each point;
- **repeat**
  - Compute the cluster centroids, $\boldsymbol{\mu}^{(i)}, j \in \{1, \ldots, k\}$;
  - Update the cluster memberships for each point;
- **until** *the membership coefficients do not change or a user-defined tolerance or maximum number of iterations is reached*;

The objective function of FCM—we abbreviate it as J_m:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i,j)^m} ||\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(j)}||_2^2 \tag{2.1}$$

We added an additional exponent to $w^{(i,j)}$; the exponent m, any number greater than or equal to one (typically $m = 2$), is the so-called **fuzziness coefficient** (or simply

fuzzifier), which controls the degree of fuzziness.

The larger the value of m, the smaller the cluster membership, $w^{(i,j)}$, becomes, which leads to fuzzier clusters. The cluster membership probability itself is calculated as follows:

$$w^{(i,j)} = \left[\sum_{c=1}^{k} \left(\frac{||\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(j)}||_2}{||\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(c)}||_2} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{2.2}$$

The center, $\boldsymbol{\mu}^{(j)}$, of a cluster itself is calculated as the mean of all examples weighted by the degree to which each example belongs to that cluster ($w^{(i,j)^m}$):

$$\boldsymbol{\mu}^{(j)} = \frac{\sum_{i=1}^{n} w^{(i,j)^m} \boldsymbol{x}^{(i)}}{\sum_{i=1}^{n} w^{(i,j)^m}} \tag{2.3}$$

Chapter 3

Others

3.1 Distance

3.1.1 Euclidean distance