Solutions

Exercise 1: Data Mining: Tasks and Methods, Sets and Relations

Exercise 1-1: Data mining questions

Which data mining tasks (association rule mining, clustering, outlier detection, classification, etc.) are hiding in the following use cases? Are the tasks supervised or unsupervised?

(a) **Optical character recognition/OCR**:
When crossing the alps using the Brenner Autobahn, there is the option to pay electronically in advance. When approaching the toll station, the barrier automatically opens when the number plate was recognized. The recognition happens fully automatically by a digital camera system.

Suggested solution:
Classification

(b) **Computer Aided Diagnosis:**
Patients that suffer from blood cancer can be characterized in two categories (ALL and AML). The therapies for these two types partially differ, and the therapy for AML can sometimes be detrimental to patients suffering from ALL and the other way around. To avoid these complications, special gene expression data are used to differentiate between these two types by comparing them to the data from patients where the cancer type is already known.

Suggested solution:
Classification

(c) **Cheat Detection**
The operator of a multi player online game wants to protect his system against various violations of the terms of service. Particular problems are the use of game bot programs, the manipulation of timestamps in the communication protocol, and attempts to predict random numbers used. To prevent this misuse, data mining is used on the available user data.

Suggested solution:
Outlier detection, sometimes classification (for known bots), clustering (to recognize strategies)

(d) **Recommendation Systems**
An online shopping portal wants to determine products that are automatically offered to registered customers upon login. The available data in particular include products previously bought by the customers to predict their interests. For example a user that bought the book “Lord of the rings” might be offered the DVDs of the movie trilogy. A related task might be suggesting additional products for already chosen products as a bundled offer.

Suggested solution:
Market basket analysis, association rules

(e) **News Aggregation**
A news summary web site automatically collects current news from various sites to keep the visitor informed. However, news reports about the same subject are common and should be grouped by subject. This happens at multiple levels: there are obviously broad categories like politics and sports, and subcategories such as soccer. But even on a single soccer game, there will likely be different news sites reporting. Some articles will be identical to the report of a major agency, some will only be slightly modified, others will be original works.

Suggested solution:
Clustering and Classification (for categories)

(f) **Extraction of Data / Web Scraping**
From some movie database a list of movies and a list of actors is to be extracted (ignoring licensing problems).

Suggested solution:
Data selection – according to our definitions not "Data Mining", but the first step of the KDD process.

(g) **Identification of the most important suppliers**
A big online seller would like to know which suppliers are most important to his business, i.e., which suppliers contribute most to his revenue. The plan is to tighten the relationship to those, to take over the company, or to place a new logistic center close to the locations of such suppliers.

Suggested solution:
Data selection and simple aggregation.
We can answer this with a simple database query, e.g.:

```
SELECT SUM(Revenue) FROM data GROUP BY supplier
```

Exercise 1-2: Set operations

An algebra is defined over a base set Ω, all sets involved in the algebra are subsets of Ω.

basic operations for $S, T \subseteq \Omega$:

union $S \cup T \equiv \{x | x \in S \vee x \in T\}$

intersection $S \cap T \equiv \{x | x \in S \wedge x \in T\}$

complement $\bar{S} \equiv S^C \equiv \{x | x \notin S\}$

difference $S \setminus T \equiv \{x | x \in S \wedge x \notin T\}$

product $S \times T \equiv \{(x, y) | x \in S \wedge y \in T\}$

Powerset $\wp(S) \equiv \wp(S) \equiv 2^S \equiv \{T | T \subseteq S\}$

Now let Ω be the English alphabet lowercase letters, i.e., $\{a, b, c, d, \dots, x, y, z\}$, and let $S = \{a, b, c\}$ and $T = \{c, d\}$.

What are the values of the following expressions:

(a) $S \cup T$

Suggested solution:
$\{a, b, c, d\}$

(b) $S \cap T$

Suggested solution:
$\{c\}$

(c) $\bar{S}$

Suggested solution:
$\{d, e, f, g, h, \dots, x, y, z\}$

(d) $S \setminus T$

Suggested solution:
$\{a, b\}$

(e) $S \times T$

Suggested solution:
$\{(a, c), (a, d), (b, c), (b, d), (c, c), (c, d)\}$

(f) $\wp(S)$

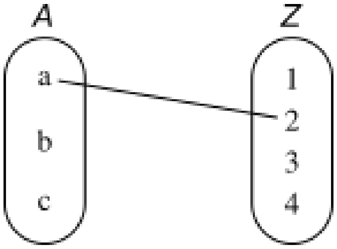
Suggested solution:
$\{\{a,b,c\},\{a,b\},\{a,c\},\{b,c\},\{a\},\{b\},\{c\},\{\}\}$

(g) $S \cap \bar{T}$

Suggested solution:
$\{a,b\}$

Exercise 1-3: Sets, Relations, Functions – Visualization

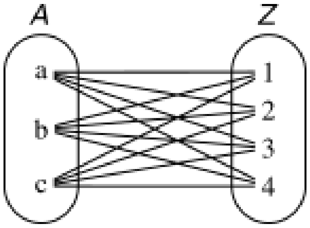
Consider the sets $A = \{a, b, c\}$ and $Z = \{1, 2, 3, 4\}$ and some binary relation over them.
If for example the elements $a \in A$ and $2 \in Z$ are in relation R could we write: $aR2$ or $(a, 2) \in R$.
As a graphical visualization we can draw the two sets and a line connecting a and 2:



Given such a visualization, the mathematical definitions basically tell us which lines to draw.
Give such a visualization for:

(a) The Cartesian product $A \times Z$

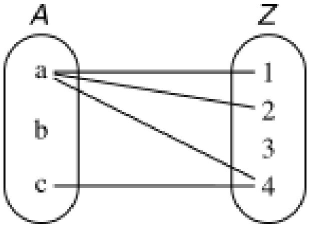
Suggested solution:



Each element in A is partner to each element in Z.

(b) A binary relation over A and Z, that is not a function.

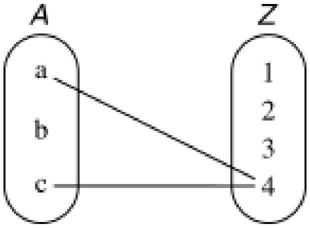
Suggested solution:



Elements in A may have none, one, or several partners in Z.
Elements in Z may have none, one, or several partners in A.

(c) A non-total function $A \to Z$

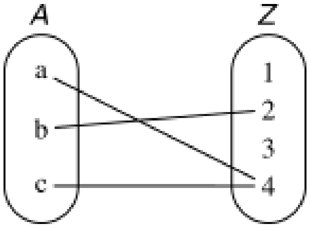
Suggested solution:



Each element in A has none or one partner in Z, but not several. At least one element in A does not have a partner (non-total).
Elements in Z may have none, one, or several partners in A.

(d) A total function $A \to Z$

Suggested solution:



Each element in A has one partner in Z, but not several.

Elements in Z may have none, one, or several partners in A.

(e) We called the case c) *non-total*. Could we have called it *partial*, or would that make a difference?

Suggested solution:

Intuitively we would expect that a function is either partial or total, but not both.

The condition for a partial function is however, that each element in A has none *or* one partner in Z.

This condition is given for non-total functions as well as for total functions.

Given the definition, *partial* is therefore a generalization of *total*, not the opposite of *total*.

Exercise 1-4: Tools and Data

(a) Install python 3.6 or higher

I. Install Anaconda with installation guide

II. Create a virtual environment named `dm566` with python 3.9 using environments guide and activate it.

Suggested solution:

```
» conda create -n dm566 python=3.9
» conda activate dm566
```

(b) Install python packages: `scikit-learn, numpy pandas, seaborn, matplotlib` with conda package installation guide.

Suggested solution:

```
» pip install scikit-learn numpy pandas seaborn matplotlib
```

(c) Load Iris dataset from `sklearn`. Assign data as X, and target as y then investigate the shapes of the data.

Suggested solution:

```
from sklearn import datasets     # import sklearn datasets

iris = datasets.load_iris()      # load iris dataset
X = iris.data
y = iris.target

print(X.shape, y.shape)
```

(d) Using `numpy, pandas, seaborn` and `matplotlib` libraries pairplot the data.

Suggested solution:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

data = np.hstack([X, y.reshape(-1, 1)])       # merge data & target
df = pd.DataFrame(data,                       # generate dataframe
    columns=iris.feature_names + ['species'])
sns.pairplot(df, hue='species')               # plot pairplot
plt.show()                                    # show plot
```

(e) Using `KMeans` from `sklearn.cluster`, try different clusters by choosing different attributes and changing the number of cluster centroids.

Suggested solution:

```
from sklearn.cluster import KMeans

Xp = X[:, :2]                                # select 2 features

kmeans = KMeans(n_clusters=3)                # create kmeans object
kmeans.fit(Xp)                               # fit kmeans object
y_pred = kmeans.predict(Xp)                  # predict cluster

plt.scatter(Xp[:, 0], Xp[:, 1], c=y_pred)    # plot scatter
plt.scatter(kmeans.cluster_centers_[:, 0],
            kmeans.cluster_centers_[:, 1],
            marker=’x’, s=200)               # plot centroid
plt.show()
```

(f) Using `StandardScaler` from `sklearn.preprocessing`, repeat the clustering. Observe the difference.

Suggested solution:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()                    # create scaler object
X_normalized = scaler.fit_transform(X)       # normalize data

Xp = X_normalized[:, :2]                     # select 2 features

kmeans = KMeans(n_clusters=3)                # create kmeans object
kmeans.fit(Xp)                               # fit kmeans object
y_pred = kmeans.predict(Xp)                  # predict cluster

plt.scatter(Xp[:, 0], Xp[:, 1], c=y_pred)    # plot scatter
plt.scatter(kmeans.cluster_centers_[:, 0],
            kmeans.cluster_centers_[:, 1],
            marker=’x’, s=200)               # plot centroid
plt.show()
```