Tutorial on k-means, neural networks and support vector machines

Introduction

- An outline on k-means, back-propagation and support vector machines are given.
- A brief description of each approach is given, with some algorithmic layouts provided.
- For the practicals, please provide your own code and answers

Note that text in blue are web references that can be clicked.

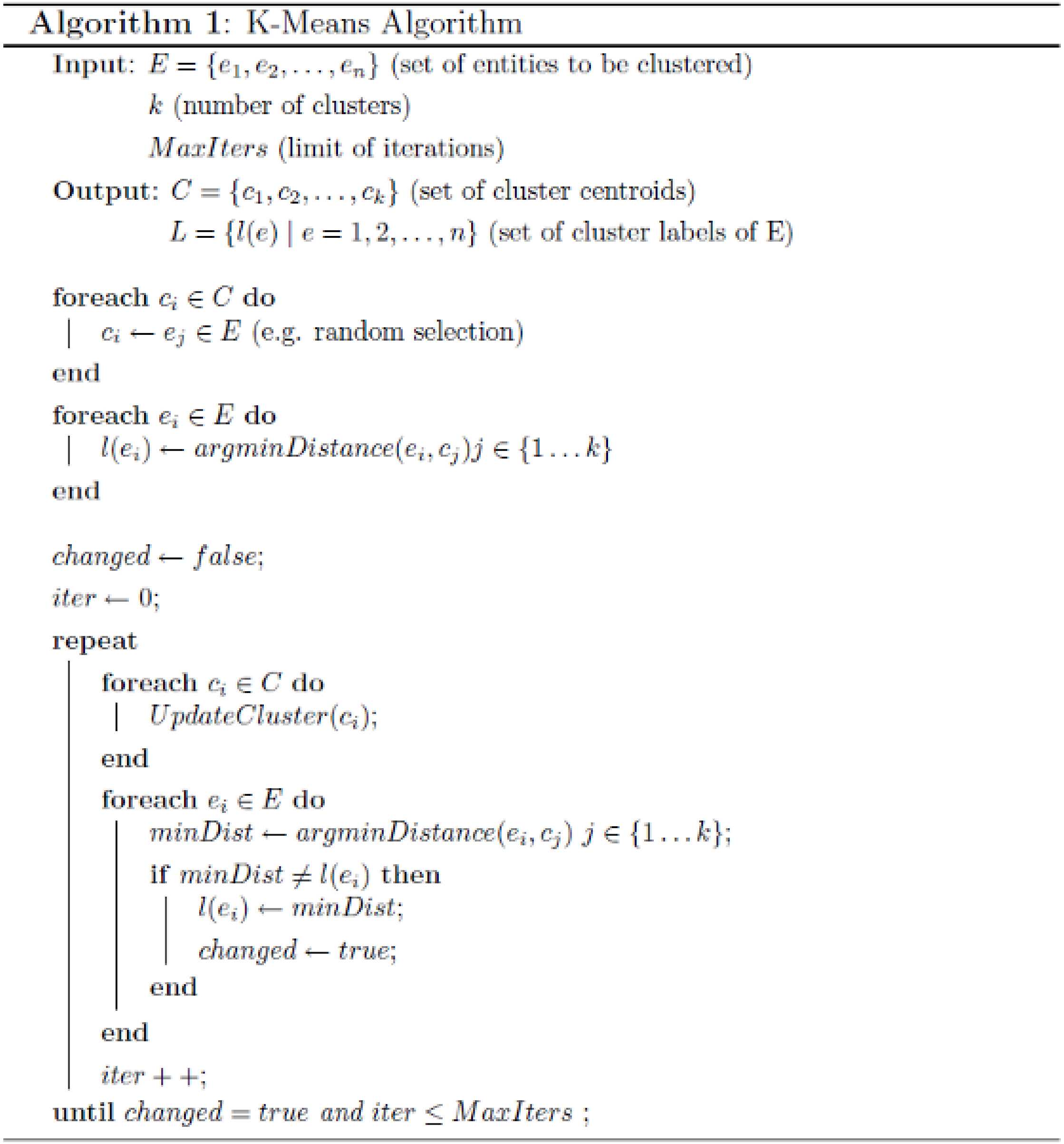
k-means: Data Mining Algorithm

- k-means is used to cluster or group data, such as drawing a circle around data that looks similar
- Formally, to partition data into said groups minimize a cluster sum of squares $\Sigma_{j=1}^{k}\Sigma_{i=1}^{n}||x_i^j - c_j||^2$ where c is the center point or centroid.
- Note that k-means is not an optimal clustering technique since clustering cannot be well defined.
- Also note that the *CalculateCentroid* and *UpdateCluster* steps are optimal.
- For the practicals, please provide your own code and answers

k-means algorithm

A visualisation of how the centroid move during the update centroid step can be seen in Centroid Update Visualisation.

Please note that the centroid of each group or cluster of data defines each group associated with that centroid. Said differently, the centroid c_1 is closest to the data associated with centroid c_1.

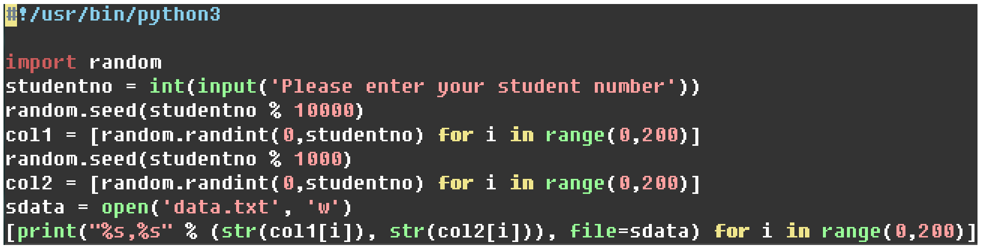


```
Algorithm 1: K-Means Algorithm
Input: $E = \{e_1, e_2, \ldots, e_n\}$ (set of entities to be clustered)
       $k$ (number of clusters)
       $MaxIters$ (limit of iterations)
Output: $C = \{c_1, c_2, \ldots, c_k\}$ (set of cluster centroids)
        $L = \{l(e) \mid e = 1, 2, \ldots, n\}$ (set of cluster labels of E)

foreach $c_i \in C$ do
    $c_i \leftarrow e_j \in E$ (e.g. random selection)
end
foreach $e_i \in E$ do
    $l(e_i) \leftarrow argminDistance(e_i, c_j) j \in \{1 \ldots k\}$
end

$changed \leftarrow false$;
$iter \leftarrow 0$;
repeat
    foreach $c_i \in C$ do
        $UpdateCluster(c_i)$;
    end
    foreach $e_i \in E$ do
        $minDist \leftarrow argminDistance(e_i, c_j)\ j \in \{1 \ldots k\}$;
        if $minDist \neq l(e_i)$ then
            $l(e_i) \leftarrow minDist$;
            $changed \leftarrow true$;
        end
    end
    $iter++$;
until $changed = true$ and $iter \leq MaxIters$ ;
```

Figure 1: k-means retrieved from `wikibooks.com` in pseudo-code[1]



```python
#!/usr/bin/python3

import random
studentno = int(input('Please enter your student number'))
random.seed(studentno % 10000)
col1 = [random.randint(0,studentno) for i in range(0,200)]
random.seed(studentno % 1000)
col2 = [random.randint(0,studentno) for i in range(0,200)]
sdata = open('data.txt', 'w')
[print("%s,%s" % (str(col1[i]), str(col2[i])), file=sdata) for i in range(0,200)]
```

Figure 2: Data generation algorithm for the practical hand-in

Practical 1: Implement k-means

Please read to the end for handing in information.

- Task 1

  Implement k-means Figure 1 from page in python. Python via Jupyter Notebook can be used for this and subsequent tasks.

  Note: installing extra python modules can be done via `pip install <module> --user` if python works on the command line. Otherwise consult the web for more information. Regardless, the following packages are of use: pandas, sklearn, numpy, matplotlib, seaborn. See eg 'pip3 install seaborn –user' on linux.

- Task 2

  Use your student number and run the code snippet in Figure 2 on your student number to generate the text file listed. Use the data in the *generated* text file to find the centroids.

Read the next page

Practical 1: Implement k-means

Please read to the end for handing in information.

- Task 3

  Explain in words your code and why and how you implemented your code the way you did.

- Task 4 Explain in words how and why the elbow method is used and where the method is useful.

- For handing in:

  Please hand in: your implementation in python of k-means, the coordinates describing your *student number generated data* centroid, your word-based explanation, and an explanation of the elbow method.

References

[1] Wikibooks, “Data mining algorithms in r/clustering/k-means.”