TDA231 - Algorithms for Machine Learning & Inference

Chalmers University of Technology

LP3– 2017-03-06

By:
Karabo Ikaneng, 891024-8239, kara.ikaneng@gmail.com
Elias Svensson, 920406-3052, elias.svensson.1992@gmail.com

Goal

K-means clustering

1 Practical problems

1.1 k-Means Implementation, 8 points

a)

See code.

b)

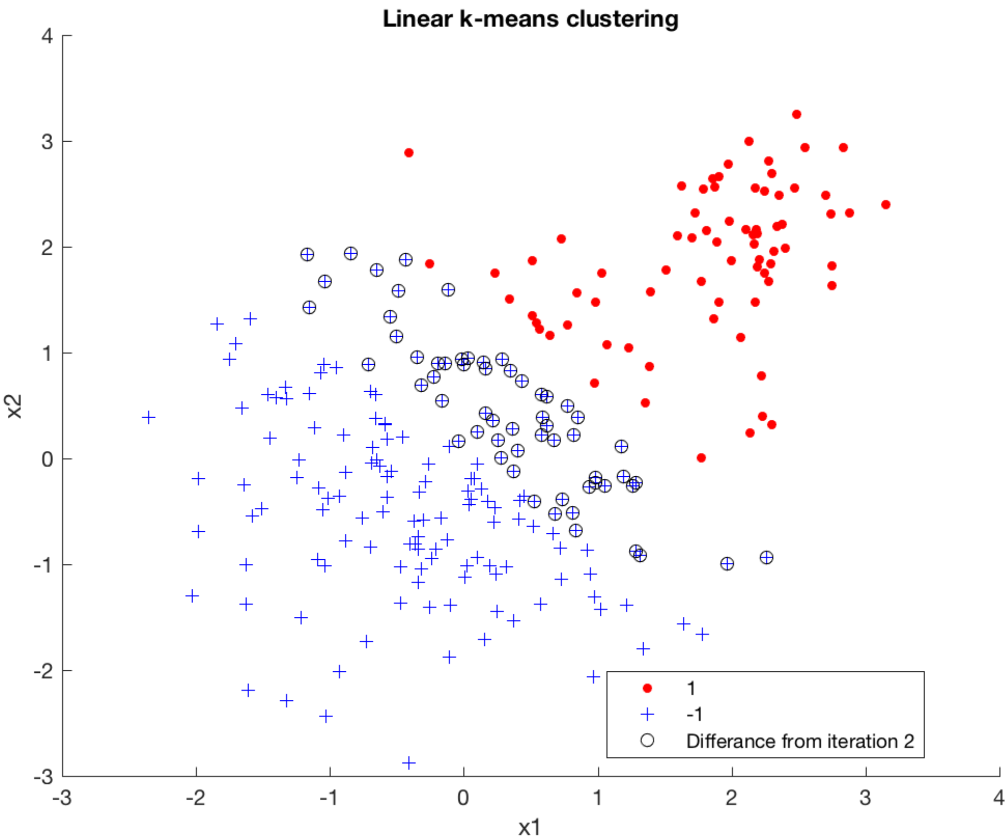

Figure 1: Linear k-means cluster assignments, stored at iteration 2 and at convergence.

c)

See code.

d)

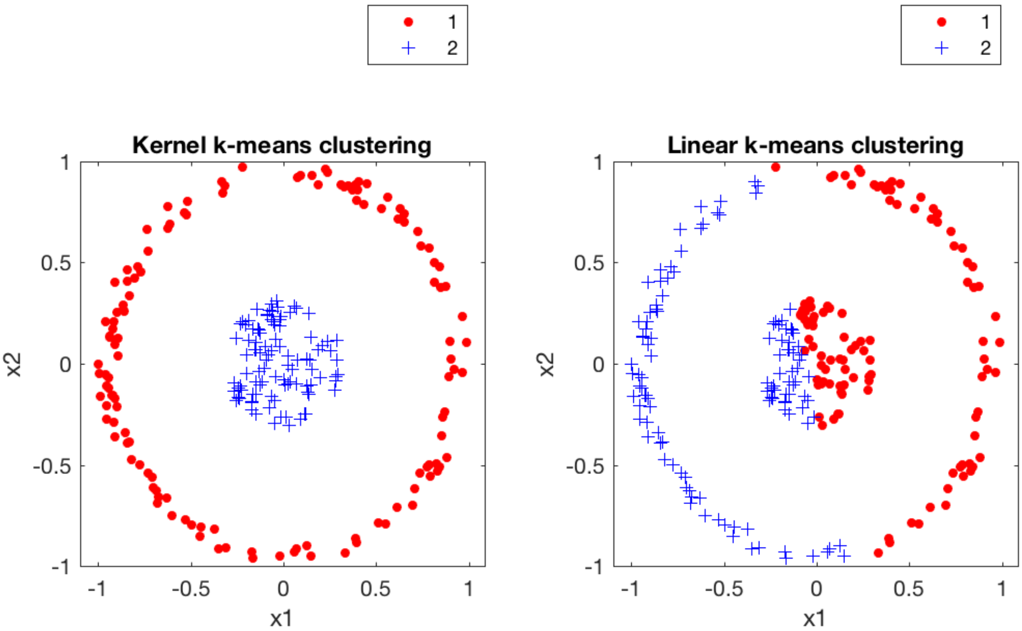

Figure 2: Linear & RBF kernel k-means cluster assignments

1.2 k-Means Analysis, 12 points

a)

The 10 words that are closest to the centroids in each cluster are shown below as printed in the command window:

```
Cluster 1
public
business
making
for
financial
new
private
to
full
well
```

Cluster 2
put
brought
to
that
after
nonetheless
would
did
finally
ultimately

Cluster 3
be
that
more
even
to
some
an
similarly
instance
well

Cluster 4
ever
next
last
start
previous
coming
making
putting
going
getting

Cluster 5
curtis
allen
miller
smith
frank
scott
warren
walker

harris
oliver

Cluster 6
another
back
up
making
with
to
be
off
turn
when

Cluster 7
now
near
area
nearby
part
today
still
from
to
it

Cluster 8
england
london
preston
bradford
kent
bedford
james
whilst
thomas
barton

Cluster 9
country
countries
foreign
abroad
international
to

```
well
elsewhere
making
with

Cluster 10
something
nothing
come
own
what
happy
gone
telling
seeing
supposed
```

b)

When the code was run, the average fraction of word pairs that remained in the same cluster was f = 0.5042. Therefore, since the words were not perfectly classified (perfect grouping would have yielded f = 1), some cluster centers were situated close to each other and tended to overlap in different kmean runs. If fewer groups were used for the the word pairs, the stability would be better, since there would be less groups to which the words could be grouped and the centers of the groups would also lay further away, which would make it more likely for the kmeans clustering algorithm to converge to the same clusters for different runs.

c)

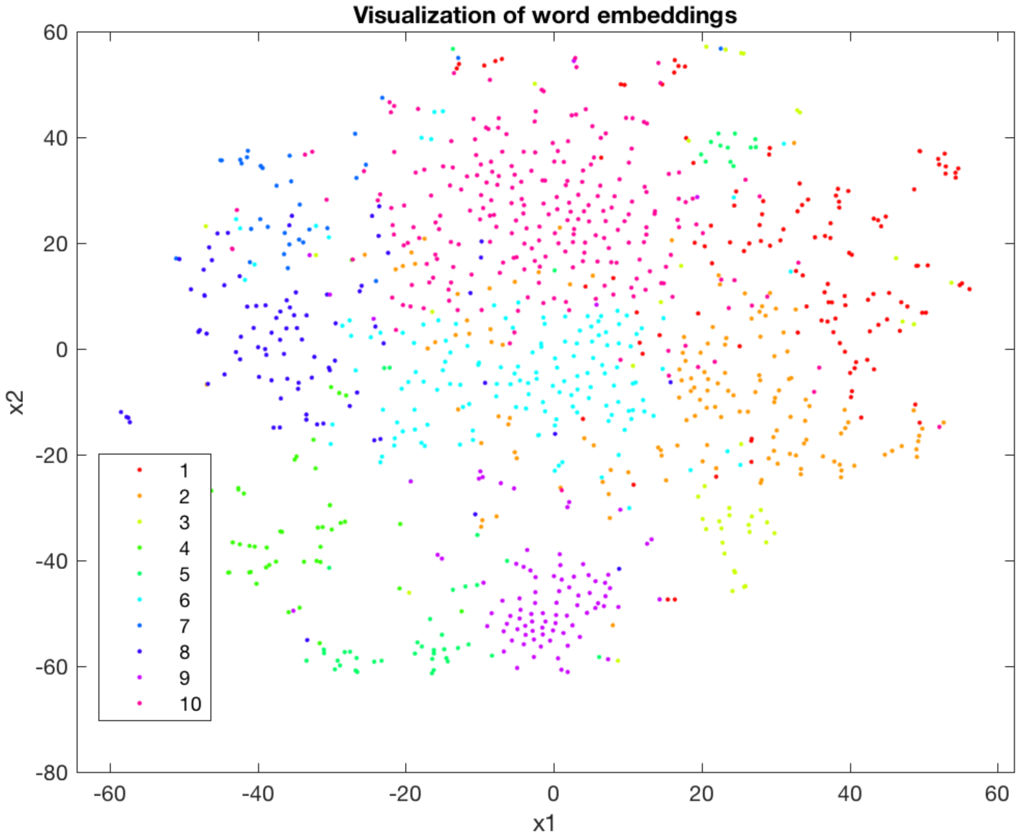

Figure 3: Visualization of the word embeddings of 1000 randomly sampled words as a point cloud.

The matlab function tsne was downloaded to project the embeddings into 2D, and then a scatter plot produced, as seen in Figure 3. Different colors were assigned to each of the computed clusters. When the data is projected and plotted in 2D, it turns out that there are no clear boundaries between the groups. This fact reflects the answer in 1.2)b) where the fraction f = 0.5042, when grouping the word embeddings.