Untitled8

February 26, 2016

1 Data Mining and Decision Support

1.1 by Dinara Assan and Nurdaulet Kenges

1.1.1 Singular Value Decomposition

In [11]:
```
#!/usr/bin/env python

        import numpy as np
        import matplotlib.pyplot as plt

        from sklearn.datasets import fetch_20newsgroups

        from sklearn.feature_extraction.text import TfidfVectorizer

        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.feature_extraction.text import TfidfTransformer
        from sklearn import decomposition


        corpus = ['To be, or not to be, that is the question',
                  'Whether tis nobler in the mind to suffer',
                  'The slings and arrows of outrageous fortune',
                  'Or to take arms against a sea of troubles',
                  'And by doing something',
                  'the the the the the the the'
        ]

        vectorizer = CountVectorizer(min_df=1)

        dt = vectorizer.fit_transform(corpus)

        x = vectorizer.get_feature_names()

        dt2 = vectorizer.fit_transform(corpus)
        a=dt2.toarray()
        print(a)
```

```
[[0 0 0 0 2 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 1 0 2 0 0]
 [0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1]
 [0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0]
 [1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 1 0]
 [0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0]]
```

In [19]:
```
u ,s , v = np.linalg.svd( dt.toarray () , full_matrices = False )
b = np.dot (u,np.dot (np.diag (s) , v ))
#print(b)

similarity = np.dot(a,b.T)/(np.linalg.norm(a)*np.linalg.norm(b))
print(similarity)

# Cosine similarity is simply the cosine of an angle between two given vectors,
# so it is a number between -1 and 1. If you, however, use it on matrices
# (as above) and a and b have more than 1 rows,
# then you will get a matrix of all possible cosines
# (between each pair of rows between these matrices).
# So example the result below gives us array of all possible cosines
```

```
[[  1.55555556e-01   3.33333333e-02   1.11111111e-02   3.33333333e-02
   -1.00228468e-18   7.77777778e-02]
 [  3.33333333e-02   8.88888889e-02   1.11111111e-02   1.11111111e-02
   -4.89577514e-18   7.77777778e-02]
 [  1.11111111e-02   1.11111111e-02   7.77777778e-02   1.11111111e-02
    1.11111111e-02   7.77777778e-02]
 [  3.33333333e-02   1.11111111e-02   1.11111111e-02   8.88888889e-02
    5.74386218e-18  -8.80468538e-17]
 [ -1.33380961e-17   2.55389845e-17   1.11111111e-02   2.02769900e-17
    4.44444444e-02  -1.06396373e-17]
 [  7.77777778e-02   7.77777778e-02   7.77777778e-02  -4.39174160e-17
   -4.31753398e-18   5.44444444e-01]]
```

In [ ]: