Implementation of Latent Semantic Analysis

May 4, 2017

CmpE561-Research Project
Mine Melodi Çalışkan - 2015705009
Serkan Duman - 2016700039

In this simulation Latent Semantic Analysis is applied to three different texts for summarization, Pablo Neruda-Sonnet Lxxxi, Anne Sexton-The Truth the Dead Know, Annie Flagg-The Story about the Lake in Fried Green Tomatoes.

In the application to increase the efficiency of SVD first we removed provided stop words and then for input matrix we used number of occurence approach and in the sentence selection step V^T and Σ with various number of sentences.

Source code is adapted from https://github.com/pegasos1/pyLSA.

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import re, random, pylab
        from math import *
        from operator import itemgetter
        from pattern.web import URL, Document, plaintext


        stopwords=[]
        with open('stopwords.txt') as f:
            for line in f:
                word=line.strip().split('\t')
                stopwords.append(word)

        stopwords=np.array(stopwords).flatten()


        ignore_characters = ''',:?'!''''


        # core summarization function.
        def summarize(query=None, k=4,text=None):
            j = []
            if text:
                with open(text) as f:
                    for line in f:
```

```
                    j.append(line)
            j = [word for sentence in j for word in sentence.split()
                 if re.match("^[a-zA-Z_-]*$", word) or '.' in word
                 or "'" in word or '"' in word]
            j = ' '.join(j)
            lsa1 = LSA(stopwords, ignore_characters)
            sentences = j.split('.')
            sentences = [sentence for sentence in sentences if len(sentence)>1
            for sentence in sentences:
                lsa1.parse(sentence)
        else:
            lsa1 = LSA(stopwords, ignore_characters)
            sentences = query.split('.')
            for sentence in sentences:
                lsa1.parse(sentence)

        lsa1.countMatrix()
        lsa1.getSVD()
        lsa1.printSVD()
        #Sentence selection step using Vt and Sigma matrices.
        summary =[(sentences[i],
                   np.linalg.norm(np.dot(np.diag(lsa1.S),lsa1.Vt[:,b]),2))
                  for i in range(len(sentences)) for b in range(len(lsa1.Vt))]
        sorted(summary, key=itemgetter(1))

        summary = dict((v[0],v)
                       for v in sorted(summary,
                                       key=lambda summary: summary[1]))

        return '.'.join([a for a, b in summary][len(summary)-(k):])



class LSA(object):

    def __init__(self, stopwords, ignore_characters):
        self.stopwords = stopwords
        self.ignore_characters = ignore_characters
        self.wdict = {}
        self.dcount = 0

    def parse(self, doc):
        words = doc.split();
        for w in words:
            w = w.lower().translate(None, self.ignore_characters)
            if w in self.stopwords:
                continue
            elif w in self.wdict:
```

```
                        self.wdict[w].append(self.dcount)
                    else:
                        self.wdict[w] = [self.dcount]
                self.dcount += 1

            # Create count matrix
            def countMatrix(self):
                self.keys = [k for k in self.wdict.keys()
                             if len(self.wdict[k]) > 1]
                self.keys.sort()
                self.A = np.zeros([len(self.keys), self.dcount])
                for i, k in enumerate(self.keys):
                    for d in self.wdict[k]:
                        self.A[i,d] += 1
                print "Count matrix"
                print self.A

            #Execute SVD
            def getSVD(self):
                self.U, self.S, self.Vt = np.linalg.svd(self.A,
                                                        full_matrices =False)

            def S(self):
                return self.S
            def U(self):
                return -1 * self.U
            def Vt(self):
                return -1 * self.Vt

            def printSVD(self):
                print 'Singular values: '
                print self.S
                print 'U matrix: '
                print -1*self.U[:, 0:3]
                print 'Vt matrix: '
                print -1*self.Vt[0:3, :]

        if __name__ == "__main__":
            text="lake.txt"
            summary1=summarize(text=text, k=1)
            print "\nText:\n"
            print "The Lake Story from the movie Fried Green Tomatoes\n"
            print "SENTENCE SUMMARY:\n"
            print summary1 + "\n"

Count matrix
[[ 0.  0.  0.  1.  0.  1.  0.  0.]
 [ 1.  0.  0.  1.  1.  1.  1.  0.]]
```

```
Singular values:
[ 2.44948974  1.        ]
U matrix:
[[ 0.4472136   0.89442719]
 [ 0.89442719 -0.4472136 ]]
Vt matrix:
[[ 0.36514837 -0.         -0.          0.54772256  0.36514837  0.54772256
   0.36514837 -0.        ]
 [-0.4472136  -0.         -0.          0.4472136  -0.4472136   0.4472136
  -0.4472136  -0.        ]]

Text:

The Lake Story from the movie Fried Green Tomatoes

SENTENCE SUMMARY:

 one november this big flock of ducks came in and landed on that lake



In [2]: if __name__ == "__main__":
            text="lake.txt"
            summary1=summarize(text=text, k=2)
            print "\nText:\n"
            print "The Lake Story from the movie Fried Green Tomatoes\n"
            print "2 SENTENCES SUMMARY:\n"
            print summary1 + "\n"

Count matrix
[[ 0.  0.  0.  1.  0.  1.  0.  0.]
 [ 1.  0.  0.  1.  1.  1.  1.  0.]]
Singular values:
[ 2.44948974  1.        ]
U matrix:
[[ 0.4472136   0.89442719]
 [ 0.89442719 -0.4472136 ]]
Vt matrix:
[[ 0.36514837 -0.         -0.          0.54772256  0.36514837  0.54772256
   0.36514837 -0.        ]
 [-0.4472136  -0.         -0.          0.4472136  -0.4472136   0.4472136
  -0.4472136  -0.        ]]

Text:

The Lake Story from the movie Fried Green Tomatoes

2 SENTENCES SUMMARY:
```

now they say that lake is somewhere over in georgia. one november this big flock o

```
In [5]: if __name__ == "__main__":
            text="neruda.txt"
            summary1=summarize(text=text, k=1)
            print "\nText:\n"
            print "Sonnet Lxxxi Poem by Pablo Neruda\n"
            print "SENTENCE SUMMARY:\n"
            print summary1 + "\n"

Count matrix
[[ 2.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.]
 [ 0.  0.  0.  0.  1.  0.  1.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.]
 [ 0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
Singular values:
[  2.23606798e+00   2.17532775e+00   1.41421356e+00   1.12603250e+00
   1.12002301e-16]
U matrix:
[[ -1.00000000e+00  -0.00000000e+00  -0.00000000e+00]
 [ -0.00000000e+00   4.59700843e-01   3.72567276e-17]
 [ -0.00000000e+00   6.27963030e-01  -9.31418190e-17]
 [ -0.00000000e+00   2.96647460e-17   1.00000000e+00]
 [ -0.00000000e+00   6.27963030e-01   1.86283638e-17]]
Vt matrix:
[[ -8.94427191e-01  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
   -0.00000000e+00  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
   -0.00000000e+00  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
   -0.00000000e+00  -4.47213595e-01]
 [ -0.00000000e+00   5.77350269e-01   6.45305451e-17  -1.01529336e-32
    7.88675135e-01  -0.00000000e+00   2.11324865e-01  -0.00000000e+00
   -0.00000000e+00  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
   -1.46285326e-32  -0.00000000e+00]
 [ -0.00000000e+00   4.54418092e-17   7.07106781e-01   4.63278219e-49
    1.52888787e-17  -0.00000000e+00  -1.52888787e-17  -0.00000000e+00
   -0.00000000e+00  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
    7.07106781e-01  -0.00000000e+00]]

Text:

Sonnet Lxxxi Poem by Pablo Neruda

SENTENCE SUMMARY:
```

```
 without you i am your dream only that and that is all
```

```
In [6]: if __name__ == "__main__":
            text="neruda.txt"
            summary1=summarize(text=text, k=2)
            print "\nText:\n"
            print "Sonnet Lxxxi Poem by Pablo Neruda\n"
            print "2 SENTENCES SUMMARY:\n"
            print summary1 + "\n"

Count matrix
[[ 2.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.]
 [ 0.  0.  0.  0.  1.  0.  1.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.]
 [ 0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
Singular values:
[  2.23606798e+00   2.17532775e+00   1.41421356e+00   1.12603250e+00
   1.12002301e-16]
U matrix:
[[ -1.00000000e+00  -0.00000000e+00  -0.00000000e+00]
 [ -0.00000000e+00   4.59700843e-01   3.72567276e-17]
 [ -0.00000000e+00   6.27963030e-01  -9.31418190e-17]
 [ -0.00000000e+00   2.96647460e-17   1.00000000e+00]
 [ -0.00000000e+00   6.27963030e-01   1.86283638e-17]]
Vt matrix:
[[ -8.94427191e-01  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
   -0.00000000e+00  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
   -0.00000000e+00  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
   -0.00000000e+00  -4.47213595e-01]
 [ -0.00000000e+00   5.77350269e-01   6.45305451e-17  -1.01529336e-32
    7.88675135e-01  -0.00000000e+00   2.11324865e-01  -0.00000000e+00
   -0.00000000e+00  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
   -1.46285326e-32  -0.00000000e+00]
 [ -0.00000000e+00   4.54418092e-17   7.07106781e-01   4.63278219e-49
    1.52888787e-17  -0.00000000e+00  -1.52888787e-17  -0.00000000e+00
   -0.00000000e+00  -0.00000000e+00  -0.00000000e+00  -0.00000000e+00
    7.07106781e-01  -0.00000000e+00]]

Text:

Sonnet Lxxxi Poem by Pablo Neruda

2 SENTENCES SUMMARY:

and now you are mine rest with your dream in my dream. without you i am your dream
```

```
In [7]: if __name__ == "__main__":
            text="sexton.txt"
            summary1=summarize(text=text, k=1)
            print "\nText:\n"
            print "The Truth the Dead Know- Poem by Anne Sexton\n"
            print "SENTENCE SUMMARY:\n"
            print summary1 + "\n"

Count matrix
[[ 0.  0.  0.  1.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  1.  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.  2.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  2.  0.  0.  0.  0.  0.]]
Singular values:
[ 2.          2.          1.41421356  1.41421356]
U matrix:
[[-0. -0. -1.]
 [-0. -0. -0.]
 [-1. -0. -0.]
 [-0. -1. -0.]]
Vt matrix:
[[-0.         -0.         -0.         -0.         -0.         -0.         -0.
  -0.         -0.         -0.         -0.         -0.         -0.         -0.
  -0.         -1.        ]
 [-0.         -0.         -0.         -0.         -0.         -0.         -0.
  -0.         -0.         -0.         -1.         -0.         -0.         -0.
  -0.         -0.        ]
 [-0.         -0.         -0.         -0.70710678 -0.         -0.         -0.
  -0.70710678 -0.         -0.         -0.         -0.         -0.         -0.
  -0.         -0.        ]]

Text:

The Truth the Dead Know- Poem by Anne Sexton

SENTENCE SUMMARY:

 and I turn to you and am bright and young
```