Visualizing LSA

Demo

- Create a fake TFIDF matrix with a strong concept
- Plot the matrix on a two term axis
- Perform SVD decomposition
- Plot the new axes
- Reduce the dimensionality of SVD
- Plot the new axes
Demo

- Create a fake TFIDF matrix with a strong concept
- Plot the matrix on a two term axis

```matlab
>> x = [1:100];
>> y = random('norm',0,5,1,100);
>> y = x + y;
>> y(70:85) = [70:85];
>> plot(x,y,'ro')
```

```matlab
>> size(x)
an =
    1     100
>> size(y)
an =
    1     100
```
Visualizing LSA

Demo

- Perform SVD decomposition

```matlab
>> C = [x;y];
>> size(C)
ans =
   2  100
>> [U S V] = svd(C);
>> size(U)
ans =
   2   2
>> size(S)
ans =
   2  100
>> S(1:2,1:2)
ans =
   822.6330  0
   0   30.2548
>> size(M)
ans =
   2   2
>> size(V)
ans =
   100  100
>> size(Cc)
ans =
   2  100
>> Sk=S(1:2,1:2);
>> Uk=U(:,1:2);
>> M = inv(Sk)*Uk';
>> Cc = M*C;
>> size(Uk)
ans =
   2   2
```
Visualizing LSA

Demo

- Plot the new axes

```matlab
>> plot(Cc(1,:),Cc(2,:), 'g*')
```
Using MATLAB For LSA

Demo

- Calculate the TFIDF score
- Plot the documents on a two term axis
- Perform SVD decomposition
  - Validate decomposition
- Reduce rank of system
- Show “M”
  - Demonstrate what SVD is capturing
- Execute a query
Using MATLAB For LSA

Demo

- Calculate the TFIDF score

<table>
<thead>
<tr>
<th>Term</th>
<th>wiki: baseballBat</th>
<th>wiki: bat</th>
<th>wiki: coffee</th>
<th>djp3 paper</th>
<th>starbucks</th>
<th>wiki:starbucks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ball</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>baseball</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bat</td>
<td>24</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>California</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>coffee</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>company</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>encyclopedia</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Irvine</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>run</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>species</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>starbucks</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>stores</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>university</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>users</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

| df            | 1                 | 1         | 2            | 1          | 2         | 1              |
Demo

• Calculate the TFIDF score

```matlab
>> !more computeTFIDF.m

% demo computeTFIDF.m

% This program calculates the TFIDF scores for a document.
% The function takes in two input matrices, one for the term frequencies
% and another for the document frequencies. It returns a matrix containing
% the TFIDF scores.

% Load term frequencies and document frequencies
% tf = load('tf.txt','-ASCII');
% df = load('df.txt','-ASCII');
% tfidf = zeros(size(tf));

for i = 1:size(tf,1)
    for j = 1:size(tf,2)
        if tf(i,j) == 0
            tfidf(i,j) = (0) * log2(c/df(i));
        else
            tfidf(i,j) = (1+log2(tf(i,j))) * log2(c/df(i));
        end
    end
end
```
Using MATLAB For LSA

Demo

- Calculate the TFIDF score

<table>
<thead>
<tr>
<th></th>
<th>wiki: baseball</th>
<th>bat</th>
<th>coffee</th>
<th>dip3 paper</th>
<th>starbucks</th>
<th>wiki: starbucks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ball</td>
<td>9.2670</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>baseball</td>
<td>11.1720</td>
<td>5.2651</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bat</td>
<td>8.8520</td>
<td>0</td>
<td>0</td>
<td>9.2670</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>California</td>
<td>0</td>
<td>0</td>
<td>8.5466</td>
<td>0</td>
<td>0</td>
<td>6.8501</td>
</tr>
<tr>
<td>coffee</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>company</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10.3399</td>
</tr>
<tr>
<td>encyclopedia</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Irvine</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9.2670</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>run</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>species</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.7549</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>starbucks</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7.6195</td>
<td>0</td>
<td>9.8419</td>
</tr>
<tr>
<td>stores</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>university</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>users</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Demo

- Perform SVD decomposition
- Validate decomposition

```matlab
>> [U S V] = svd(tfidf);
>> size(U)
ans =
   14   14

>> size(S)
ans =
   14    6

>> size(V)
ans =
    6    6

>> U*S*V'
ans =
   9.2670   -0.0000    0.0000    0.0000    0.0000   -0.0000
  11.1720   -0.0000    0.0000    0.0000    0.0000    0.0000
   8.8520    5.2651   -0.0000   -0.0000   -0.0000    0.0000
  -0.0000    0.0000    0.0000    9.2670   -0.0000   -0.0000
   0.0000   -0.0000    8.5466   -0.0000    0.0000    6.8501
   0.0000   -0.0000   -0.0000   -0.0000    0.0000    10.3399
   0.5850    0.5850    0.5850    0.5850    0.0000    0.5850
  -0.0000    0.0000    0.0000    9.2670   -0.0000   -0.0000
   8.5871   -0.0000    0.0000    0.0000    0.0000    0.0000
  -0.0000    4.0971    3.1699    0.0000    0.0000   -0.0000
   0.0000   -0.0000   -0.0000    0.0000    4.7549    7.6195
   0.0000   -0.0000   -0.0000   -0.0000    0.0000    9.8419
  -0.0000    0.0000    0.0000    9.2670   -0.0000   -0.0000
  -0.0000    0.0000    0.0000    8.5871   -0.0000   -0.0000
```
## Demo

- Reduce rank of system

```
>> Sk = S(1:3,1:3)
Sk =
   19.2339       0       0
      0    18.2035       0
      0       0    18.1004

>> Uk = U(:,1:3)
Uk =
   -0.4767    -0.0000    0.0116
   -0.5747    -0.0000    0.0140
   -0.4946       0.0000    0.0086
      0    -0.5091       0.0000
   -0.0119       0.0000    -0.4755
   -0.0102       0.0000    -0.5514
   -0.0354       0.0000    -0.0383
      0    -0.5091       0.0000
   -0.4417    -0.0000    0.0107
   -0.0325    -0.0000    -0.0427
   -0.0080    -0.0000    -0.4365
   -0.0097       0.0000    -0.5249
      0    -0.5091       0.0000
      0    -0.4717       0.0000
```
Using MATLAB For LSA

Demo

- Show “M”

```matlab
>> M = inv(Sk)*Uk;

M =

Columns 1 through 10
-0.0248   -0.0299   -0.0257   0.0000   -0.0006   -0.0005   -0.0018   0.0000   -0.0230   -0.0017
-0.0000   -0.0000   0.0000   -0.0280   0.0000   0.0000   0.0000   -0.0280   -0.0000   -0.0000
 0.0006   0.0008   0.0005   0.0000  -0.0263  -0.0305  -0.0021   0.0000   0.0006  -0.0024

Columns 11 through 14
-0.0004   -0.0005   0.0000   0.0000
-0.0000   0.0000  -0.0280  -0.0259
-0.0241  -0.0290   0.0000   0.0000
```
Demo

• Demonstrate what SVD is capturing

• 1st concept (1st row of M)

First concept is selecting for Wikipedia?

First concept is selecting for baseball?
Demo

- Demonstrate what SVD is capturing
- 2nd concept (2nd row of M)
Using MATLAB For LSA

Demo

- Demonstrate what SVD is capturing
- 3rd concept (3rd row of M)

First concept is selecting for wikipedia?

Third concept is selecting for coffee?
Using MATLAB For LSA

Demo

• Execute a query “coffee stores”

```matlab
>> q=[ 0 0 0 0 (1+log2(1))*log2(c/df(5)) 0 0 0 0 0 (1+log(1))*log2(c/df(12)) 0 0 ]

q =

Columns 1 through 10

0 0 0 0 0 1.5850 0 0 0 0

Columns 11 through 14

0 2.5850 0 0

>> qc=M*q'

qc =

-0.0023
-0.0000
-0.1166
```
Using MATLAB For LSA

Demo

- Execute a query

\[ \text{sim}(q, d_i) = \frac{\vec{V}(q) \cdot \vec{V}(d_i)}{|\vec{V}(q)||\vec{V}(d_i)|} \]

>> Cc = inv(Sk)*Uk'*tfidf
Cc =
    -0.9894   -0.1434   -0.0117    0.0000    -0.0020    -0.0189
    0.0000    0.0000    0.0000   -1.0000    -0.0000    -0.0000
   0.0227   -0.0084   -0.2332    0.0000   -0.1147   -0.9653

>> sim = qc'*Cc;
>> sim = sim ./ [norm(Cc(:,1)) norm(Cc(:,2)) norm(Cc(:,3)) norm(Cc(:,4)) norm(Cc(:,5)) norm(Cc(:,6))]

sim =
    -0.0004    0.0091    0.1166   -0.0000    0.1166    0.1166
Demo

• Execute a query “coffee stores”

• Answer:
  • starbucks (0.1166)
  • wiki:starbucks(0.1166)
  • wiki:coffee (0.1166)
  • wiki:bat (0.0091)
  • djp3 paper (0.0)
  • wiki:baseballBat (-0.0004)
Demo

• Execute a query “baseball bat”

```matlab
>> q=[(1+log2(1))*log2(c/df(1)) (1+log2(1))*log2(c/df(2)) 0 0 0 0 0 0 0 0 0 0 0 0 0]'

q =

Columns 1 through 10

2.5850  2.5850  0  0  0  0  0  0  0  0

Columns 11 through 14

0  0  0  0

>> qc=M*q'

qc =

-0.1413
-0.0000
-0.0000
 0.0037
Using MATLAB For LSA

Demo

• Execute a query

\[
sim(q, d_i) = \frac{\vec{V}(q) \cdot \vec{V}(d_i)}{|\vec{V}(q)||\vec{V}(d_i)|}
\]

>> Cc = inv(Sk)*Uk'*tfidf
Cc =

-0.9894  -0.1434  -0.0117  0.0000  -0.0020  -0.0189
 0.0000   0.0000   0.0000  -1.0000  -0.0000  -0.0000
 0.0227  -0.0084  -0.2332  0.0000  -0.1147  -0.9653

>> sim = qc'*Cc;
>> sim = sim ./ [norm(Cc(:,1)) norm(Cc(:,2)) norm(Cc(:,3)) norm(Cc(:,4)) norm(Cc(:,5)) norm(Cc(:,6))]

sim =

 0.1414  0.1408  0.0035  0.0000  -0.0012  -0.0009
Demo

• Execute a query “baseball bat”

• Answer:
  • wiki:baseballBat (0.1414)
  • wiki:bat (0.1408)
  • wiki:coffee (0.0035)
  • djp3 paper (0.000)
  • wiki:starbucks (-0.0009)
  • starbucks (-0.0012)