Latent Semantic Indexing

Chapter 18

LSI

- Applications in Search Engine Optimization (SEO) video
  http://www.youtube.com/watch?v=LOPY1hPcZEM
Latent Semantic Indexing (LSI) (AKA “Latent Semantic Analysis” (LSA))

- Problem: How to capture semantic similarity between documents in a natural corpus (e.g., problems of homonymy, polysemy, synonymy, etc.)

- “LSA assumes that there exists a LATENT structure in word usage – obscured by variability in word choice”

(http://ir.dcs.gla.ac.uk/oldseminars/Girolami.ppt)

The Problem

- Example: Vector Space Model
  - (from Lillian Lee)

  From www.cs.nmsu.edu/~mmartin/LSA_Intro_AI_Seminar.ppt
Principal Components Analysis

- PCA used to reduce dimensions of data without much loss of information.

- Used in machine learning and in signal processing and image compression (among other things).

PCA is “an orthogonal linear transformation that transfers the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (first principal component), the second greatest variance lies on the second coordinate (second principal component), and so on.”
Background for PCA

• Suppose attributes are $A_1$ and $A_2$, and we have $n$ training examples. $x$’s denote values of $A_1$ and $y$’s denote values of $A_2$ over the training examples.

• Variance of an attribute:

$$\text{var}(A_i) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{(n-1)}$$

• Covariance of two attributes:

$$\text{cov}(A_1, A_2) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{(n-1)}$$

• If covariance is positive, both dimensions increase together. If negative, as one increases, the other decreases. Zero: independent of each other.
• Covariance matrix
  – Suppose we have $n$ attributes, $A_1$, ..., $A_n$.

  – Covariance matrix:

$$C_{n \times n} = (c_{i,j})$$, where $c_{i,j} = \text{cov}(A_i, A_j)$
Eigenvectors:
- Let $M$ be an $n \times n$ matrix.
  - $v$ is an eigenvector of $M$ if $M \times v = \lambda v$
  - $\lambda$ is called the eigenvalue associated with $v$
- For any eigenvector $v$ of $M$ and scalar $a$,
  \[ M \times av = \lambda av \]
- Thus you can always choose eigenvectors of length 1:
  \[ \sqrt{v_1^2 + ... + v_n^2} = 1 \]
- If $M$ has any eigenvectors, it has $n$ of them, and they are
  orthogonal to one another.
- Thus eigenvectors can be used as a new basis for a $n$-dimensional vector space.

PCA

1. Given original data set $S = \{x^1, ..., x^k\}$, produce new set by subtracting the mean of attribute $A_i$ from each $x_i$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>2.4</td>
<td>.69</td>
<td>.49</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7</td>
<td>-1.31</td>
<td>-1.21</td>
</tr>
<tr>
<td>2.2</td>
<td>2.9</td>
<td>.39</td>
<td>.99</td>
</tr>
<tr>
<td>1.9</td>
<td>2.2</td>
<td>.09</td>
<td>.29</td>
</tr>
<tr>
<td>Data = 3.1</td>
<td>3.0</td>
<td>DataAdjust = 1.29</td>
<td>1.09</td>
</tr>
<tr>
<td>2.3</td>
<td>2.7</td>
<td>.49</td>
<td>.79</td>
</tr>
<tr>
<td>2</td>
<td>1.6</td>
<td>.19</td>
<td>-.31</td>
</tr>
<tr>
<td>1</td>
<td>1.1</td>
<td>-.81</td>
<td>-.81</td>
</tr>
<tr>
<td>1.5</td>
<td>1.6</td>
<td>-.31</td>
<td>-.31</td>
</tr>
<tr>
<td>1.1</td>
<td>0.9</td>
<td>-.71</td>
<td>-1.01</td>
</tr>
</tbody>
</table>

Mean: 1.81   1.91   Mean: 0     0
2. Calculate the covariance matrix:

\[
\begin{pmatrix}
  x & y \\
 0.61655556 & 0.61544444 \\
0.61544444 & 0.71655556
\end{pmatrix}
\]

3. Calculate the (unit) eigenvectors and eigenvalues of the covariance matrix:

Eigenvectors:

\[
\begin{pmatrix}
-0.735178656 & -0.677873399 \\
0.677873399 & -0.735178656
\end{pmatrix}
\]

Eigenvalues:

\[
\begin{pmatrix}
0.0490833989 \\
1.28402771
\end{pmatrix}
\]
Order eigenvectors by eigenvalue, highest to lowest.

\[ \mathbf{v}_1 = \begin{pmatrix} -0.677873399 \\ -0.735178956 \end{pmatrix} \quad \lambda = 1.28402771 \]

\[ \mathbf{v}_2 = \begin{pmatrix} -0.735178956 \\ 0.677873399 \end{pmatrix} \quad \lambda = 0.0490833989 \]

In general, you get \( n \) components. To reduce dimensionality to \( p \), ignore \( n-p \) components at the bottom of the list.
Construct new feature vector. 
Feature vector = \((v_1, v_2, \ldots, v_p)\)

\[\text{FeatureVector}_1 = \begin{pmatrix} -0.677873399 & 0.735178956 \\ -0.735178956 & 0.677873399 \end{pmatrix}\]

or reduced dimension feature vector:

\[\text{FeatureVector}_2 = \begin{pmatrix} -0.677873399 \\ -0.735178956 \end{pmatrix}\]

5. Derive the new data set.

\[\text{TransformedData} = \text{RowFeatureVector} \times \text{RowDataAdjust}\]

\[\text{RowFeatureVector}_1 = \begin{pmatrix} -0.677873399 & -0.735178956 \\ -0.735178956 & 0.677873399 \end{pmatrix}\]

\[\text{RowFeatureVector}_2 = \begin{pmatrix} -0.677873399 & -0.735178956 \end{pmatrix}\]

\[\text{RowDataAdjust} = \begin{pmatrix} 0.69 & -1.31 & 0.39 & 1.29 & 0.49 & 0.19 & -0.81 & -0.31 & -0.71 \\ 0.49 & -1.21 & 0.99 & 0.29 & 1.09 & 0.79 & -0.31 & -0.81 & -0.31 & -1.01 \end{pmatrix}\]

This gives original data in terms of chosen components (eigenvectors)—that is, along these axes.
Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.
Reconstructing the original data

We did:

$$\text{TransformedData} = \text{RowFeatureVector} \times \text{RowDataAdjust}$$

so we can do

$$\text{RowDataAdjust} = \text{RowFeatureVector}^{-1} \times \text{TransformedData}$$

$$= \text{RowFeatureVector}^T \times \text{TransformedData}$$

and

$$\text{RowDataOriginal} = \text{RowDataAdjust} + \text{OriginalMean}$$
Example: Linear discrimination using PCA for face recognition

1. Preprocessing: “Normalize” faces

   • Make images the same size
   • Line up with respect to eyes
   • Normalize intensities
2. Raw features are pixel intensity values (2061 features)

3. Each image is encoded as a vector $\Gamma_i$ of these features

4. Compute “mean” face in training set:

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$$

- Subtract the mean face from each face vector
  $$\Phi_i = \Gamma_i - \Psi$$
- Compute the covariance matrix $C$
- Compute the (unit) eigenvectors $v_i$ of $C$
- Keep only the first $K$ principal components (eigenvectors)
The eigenfaces encode the principal sources of variation in the dataset (e.g., absence/presence of facial hair, skin tone, glasses, etc.).

We can represent any face as a linear combination of these "basis" faces.

Use this representation for:

• Face recognition  
  (e.g., Euclidean distance from known faces)

• Linear discrimination  
  (e.g., "glasses" versus "no glasses",  
  or "male" versus "female")
Latent Semantic Analysis
(Landauer et al.)

• From training data (large sample of documents), create word-by-document matrix.

Technical Memo Example

Titles:
c1: Human machine interface for Lab ABC computer applications
c2: A survey of user opinion of computer system response time
c3: The EPS user interface management system
c4: System and human system engineering testing of EPS
c5: Relation of user-perceived response time to error measurement

m1: The generation of random, binary, unordered trees
m2: The intersection graph of paths in trees
m3: Graph minors IV: Widths of trees and well-quasi-ordering
m4: Graph minors: A survey

A sample dataset consisting of the titles of 9 technical memoranda. Terms occurring in more than one title are italicized. There are two classes of documents - five about human-computer interaction (c1-c5) and four about graphs (m1-m4). This dataset can be described by means of a term by document matrix where each cell entry indicates the frequency with which a term occurs in a document.

From Deerwester et al., Indexing by latent semantic analysis
• Now apply “singular value decomposition” to this matrix

• SVD is similar to principal components analysis

• Basically, reduce dimensionality of the matrix by re-representing matrix in terms of “features” (derived from eigenvalues and eigenvectors), and using only the ones with highest value.

• Result: Each document is represented by a vector of features obtained by SVD.

• Given a new document (or query), compute its representation vector in this feature space, compute its similarity with other documents using cosine between vector angles. Retrieve documents with highest similarities.
Example 18.3  We now illustrate the singular-value decomposition of a $4 \times 2$ matrix of rank 2; the singular values are $\Sigma_{11} = 2.236$ and $\Sigma_{22} = 1$.

$$
C = \begin{pmatrix}
1 & -1 \\
0 & 1 \\
1 & 0 \\
-1 & 1 \\
\end{pmatrix} = \begin{pmatrix}
-0.632 & 0.000 \\
0.316 & -0.707 \\
-0.316 & -0.707 \\
0.632 & 0.000 \\
\end{pmatrix} \begin{pmatrix}
2.236 & 0.000 \\
0.000 & 1.000 \\
-0.707 & -0.707 \\
\end{pmatrix} \begin{pmatrix}
-0.707 & 0.707 \\
\end{pmatrix}.
$$

- SVD
  - can be viewed as a method for rotating the axes in n-dimensional space, so that the first axis runs along the direction of the largest variation among the documents
    - the second dimension runs along the direction with the second largest variation
    - and so on
LSI

- Four basic steps
  - Rank-reduced Singular Value Decomposition (SVD) performed on matrix
    - all but the k highest singular values are set to 0
    - produces k-dimensional approximation of the original matrix
    - this is the “semantic space”
  - Compute similarities between entities in semantic space (usually with cosine)

\[
\begin{array}{cccccccc}
\text{c1} & \text{c2} & \text{c3} & \text{c4} & \text{c5} & \text{m1} & \text{m2} & \text{m3} & \text{m4} \\
\text{human} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{interface} & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\text{computer} & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{user} & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
\text{system} & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 \\
\text{response} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\text{time} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\text{EPS} & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
\text{survey} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\text{trees} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
\text{graph} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\text{minors} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{array}
\]

\[
\begin{align*}
\text{r} (\text{human.user}) &= -.38 \\
\text{r} (\text{human.minors}) &= -.29
\end{align*}
\]
• Singular Value Decomposition
\[ \mathbf{C} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \]

• Dimension Reduction
\[ \sim \mathbf{C} = \sim \mathbf{U} \sim \mathbf{\Sigma} \sim \mathbf{V} \]

\[ \begin{bmatrix}
0.22 & -0.11 \\
0.20 & -0.07 \\
0.24 & 0.04 \\
0.40 & 0.06 \\
0.64 & -0.17 \\
0.27 & 0.11 \\
0.27 & 0.11 \\
0.30 & -0.14 \\
0.21 & 0.27 \\
0.01 & 0.49 \\
0.04 & 0.62 \\
0.03 & 0.45
\end{bmatrix} \]

\[ \begin{bmatrix}
0.29 & -0.41 & -0.11 & -0.34 & 0.52 & -0.06 & -0.41 \\
0.14 & -0.55 & 0.28 & 0.50 & -0.07 & -0.01 & -0.11 \\
-0.16 & -0.59 & -0.11 & -0.25 & -0.30 & 0.06 & 0.49 \\
0.10 & 0.33 & 0.38 & 0.00 & 0.00 & 0.01 \\
0.33 & -0.16 & -0.21 & -0.17 & 0.03 & 0.27 \\
-0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\
-0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\
0.33 & 0.19 & 0.11 & 0.27 & 0.03 & -0.02 & -0.17 \\
-0.18 & -0.03 & -0.54 & 0.08 & -0.47 & -0.04 & -0.58 \\
0.23 & 0.03 & 0.59 & -0.39 & -0.29 & 0.25 & -0.23 \\
0.22 & 0.00 & -0.07 & 0.11 & 0.16 & -0.68 & 0.23 \\
0.14 & -0.01 & -0.30 & 0.28 & 0.34 & 0.68 & 0.18
\end{bmatrix} \]
\[ \Sigma = \begin{bmatrix} 3.34 & 2.54 & 2.35 & 1.64 & 1.50 & 1.31 & 0.85 & 0.56 & 0.36 \\ 2.54 & 2.35 & 1.64 & 1.50 & 1.31 & 0.85 & 0.56 & 0.36 & 0.17 \\ 2.35 & 1.64 & 1.50 & 1.31 & 0.85 & 0.56 & 0.36 & 0.17 & 0.06 \\ 1.64 & 1.50 & 1.31 & 0.85 & 0.56 & 0.36 & 0.17 & 0.06 & 0.01 \\ 1.50 & 1.31 & 0.85 & 0.56 & 0.36 & 0.17 & 0.06 & 0.01 & 0.00 \\ 1.31 & 0.85 & 0.56 & 0.36 & 0.17 & 0.06 & 0.01 & 0.00 & 0.00 \\ 0.85 & 0.56 & 0.36 & 0.17 & 0.06 & 0.01 & 0.00 & 0.00 & 0.00 \\ 0.56 & 0.36 & 0.17 & 0.06 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.36 & 0.17 & 0.06 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix} \]

\[ \begin{array}{ccccccccccc} \text{V} = & 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.01 & 0.02 & 0.08 \\ & -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \\ & 0.11 & -0.50 & 0.21 & 0.57 & -0.51 & 0.10 & 0.19 & 0.25 & 0.08 \\ & -0.05 & -0.03 & 0.04 & 0.27 & 0.15 & 0.02 & 0.02 & 0.01 & -0.03 \\ & 0.05 & -0.21 & 0.38 & -0.21 & 0.33 & 0.39 & 0.35 & 0.15 & -0.60 \\ & -0.08 & -0.26 & 0.72 & -0.37 & 0.03 & -0.30 & -0.21 & 0.00 & 0.36 \\ & 0.18 & -0.43 & -0.24 & 0.26 & 0.67 & -0.34 & -0.15 & 0.25 & 0.04 \\ & -0.01 & 0.05 & 0.01 & -0.02 & -0.06 & 0.45 & -0.76 & 0.45 & -0.07 \\ & -0.06 & 0.24 & 0.02 & -0.08 & -0.26 & -0.62 & 0.02 & 0.52 & -0.45 \end{array} \]
Correlation

<table>
<thead>
<tr>
<th></th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>c4</th>
<th>c5</th>
<th>m1</th>
<th>m2</th>
<th>m3</th>
<th>m4</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>0.16</td>
<td>0.40</td>
<td>0.38</td>
<td>0.47</td>
<td>0.18</td>
<td>-0.05</td>
<td>-0.12</td>
<td>-0.16</td>
<td>-0.09</td>
</tr>
<tr>
<td>interface</td>
<td>0.14</td>
<td>0.37</td>
<td>0.33</td>
<td>0.40</td>
<td>0.16</td>
<td>-0.03</td>
<td>-0.07</td>
<td>-0.10</td>
<td>-0.04</td>
</tr>
<tr>
<td>computer</td>
<td>0.15</td>
<td>0.51</td>
<td>0.36</td>
<td>0.41</td>
<td>0.24</td>
<td>0.02</td>
<td>0.06</td>
<td>0.09</td>
<td>0.12</td>
</tr>
<tr>
<td>user</td>
<td>0.26</td>
<td>0.84</td>
<td>0.61</td>
<td>0.70</td>
<td>0.39</td>
<td>0.03</td>
<td>0.08</td>
<td>0.12</td>
<td>0.19</td>
</tr>
<tr>
<td>system</td>
<td>0.45</td>
<td>1.23</td>
<td>1.05</td>
<td>1.27</td>
<td>0.56</td>
<td>-0.07</td>
<td>-0.15</td>
<td>-0.21</td>
<td>-0.05</td>
</tr>
<tr>
<td>response</td>
<td>0.16</td>
<td>0.58</td>
<td>0.38</td>
<td>0.42</td>
<td>0.28</td>
<td>0.06</td>
<td>0.13</td>
<td>0.19</td>
<td>0.22</td>
</tr>
<tr>
<td>time</td>
<td>0.16</td>
<td>0.58</td>
<td>0.38</td>
<td>0.42</td>
<td>0.28</td>
<td>0.06</td>
<td>0.13</td>
<td>0.19</td>
<td>0.22</td>
</tr>
<tr>
<td>EPS</td>
<td>0.22</td>
<td>0.55</td>
<td>0.51</td>
<td>0.63</td>
<td>0.24</td>
<td>-0.07</td>
<td>-0.14</td>
<td>-0.20</td>
<td>-0.11</td>
</tr>
<tr>
<td>survey</td>
<td>0.10</td>
<td>0.53</td>
<td>0.23</td>
<td>0.21</td>
<td>0.27</td>
<td>0.14</td>
<td>0.31</td>
<td>0.44</td>
<td>0.42</td>
</tr>
<tr>
<td>trees</td>
<td>-0.06</td>
<td>0.23</td>
<td>-0.14</td>
<td>-0.27</td>
<td>0.14</td>
<td>0.24</td>
<td>0.55</td>
<td>0.77</td>
<td>0.66</td>
</tr>
<tr>
<td>graph</td>
<td>-0.06</td>
<td>0.34</td>
<td>-0.15</td>
<td>-0.30</td>
<td>0.20</td>
<td>0.31</td>
<td>0.69</td>
<td>0.98</td>
<td>0.85</td>
</tr>
<tr>
<td>minors</td>
<td>-0.04</td>
<td>0.25</td>
<td>-0.10</td>
<td>-0.21</td>
<td>0.15</td>
<td>0.22</td>
<td>0.50</td>
<td>0.71</td>
<td>0.62</td>
</tr>
</tbody>
</table>

\[ r_{(human.user)} = .94 \]
\[ r_{(human.minors)} = -.83 \]
Semantic Similarity Measure

- To find similarity between two documents, project them in LS space
- Then calculate the cosine measure between their projection

Summary

- Some Issues
  - SVD Algorithm complexity O(n^2k^3)
    - n = number of terms
    - k = number of dimensions in semantic space (typically small ~50 to 350)
    - for stable document collection, only have to run once
    - dynamic document collections: might need to rerun SVD, but can also “fold in” new documents
Summary

• Some issues
  – Finding optimal dimension for semantic space
    • precision-recall improve as dimension is increased until hits optimal, then slowly decreases until it hits standard vector model
    • run SVD once with big dimension, say $k = 1000$
      – then can test dimensions $\leq k$
    • in many tasks 150-350 works well, still room for research

Summary

• Some issues
  – SVD assumes normally distributed data
    • term occurrence is not normally distributed
    • matrix entries are weights, not counts, which may be normally distributed even when counts are not
Some General LSA Based Applications
From http://lsa.colorado.edu/~quesadaj/pdf/LSATutorial.pdf

Information Retrieval

Text Assessment
   Compare document to documents of known quality / content

Automatic summarization of text
   Determine best subset of text to portray same meaning

Categorization / Classification
   Place text into appropriate categories or taxonomies

Application: Automatic Essay Scoring
   (in collaboration with Educational Testing Service)

Create domain semantic space

Compute vectors for essays, add to vector database

To predict grade on a new essay, compare it to ones previously scored by humans

From http://lsa.colorado.edu/~quesadaj/pdf/LSATutorial.pdf
Mutual information between two sets of grades:

human – human .90
LSA – human .81

From http://lsa.colorado.edu/~quesadaj/pdf/LSATutorial.pdf

Demo
http://www.pearsonkt.com/
http://www.pearsonkt.com/prodIEA.shtml
http://www.pearson.com/investors/our-news/?i=772
http://www.youtube.com/results?search_query=Karen+Lochbaum&aq=f