Lecture 14a: Latent Semantic Analysis (Part 2)

Datasets in the form of matrices

We are given \( n \) objects (documents or sentences) and \( d \) features (words, possibly stemmed) describing the objects.

**Dataset**

An \( n \)-by-\( d \) matrix \( A \), \( A_{ij} \)

Every row of \( A \) represents an object (e.g. sentence/document)

Every column represents a feature (e.g. word)

**Goal**

1. Understand the structure of the data, e.g., the underlying process generating the data.
2. Reduce the number of features representing the data

The Singular Value Decomposition (SVD)

Data matrices have \( n \) rows (one for each object) and \( d \) columns (one for each feature).

**Rows:** vectors in a Euclidean space,

Two objects are "close" if the angle between their corresponding vectors is small.

Announcements

NLP Project

- Team in-class presentation: Tuesday May 7th
- One week from Today!
- Show me something cool!

- Paper: Friday, May 10th
  - One week from tomorrow!
- Any questions?

Document matrices

Find a subset of the terms that accurately clusters the documents or terms

SVD decomposition

\[
\begin{pmatrix}
A
\end{pmatrix}
= \begin{pmatrix}
U
\end{pmatrix}
\begin{pmatrix}
S
\end{pmatrix}
\begin{pmatrix}
V^T
\end{pmatrix}
\]

\( U (V) \): orthogonal matrix containing the left (right) singular vectors of \( A \).

\( \Sigma \): diagonal matrix containing the singular values of \( A \): \( \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k \)

Exact computation of the SVD takes \( O(\min(mn^2, m^2n)) \) time.

The top \( k \) left/right singular vectors/values can be computed faster using Lanczos/Arnoldi methods.
Latent Semantic Analysis

How do we use SVD (or CX, or ICA, or ...) in NLP?

Let’s look at the text retrieval problem
- Given a corpus of documents $X$
- And a new document $A$
- Order the documents in $X$ by similarity to $A$

Not really. Usually IR is defined as returning the $N$ most similar documents. But we will look at retrieval issues later. For the moment, defining an ordering will do.

**LSA Step 1:** Analyze corpus by SVD

The left singular vectors $U$ map documents to concepts
- Discard all but the first $K$ columns of $U$
- Assuming ordered by the magnitude of the singular vectors
The 1st $K$ columns will map documents onto the $K$ major concepts in your corpus

$XU_k$ is a compacted version of the corpus
- $X$ is a documents x terms matrix
- $U_k$ is documents x $K$
- Much smaller
- Arguably more meaningful
Each row is a vector
Each term $XU_k[i,j]$ measures how much concept $j$ occurs in document $i$
**LST Step 2 (cont)**

Faster than you might think
- Typically one corpus $X$
- Many query images $A$, given one at a time

Compute $XU_k$ once
- Normalize the rows to have magnitude 1.

For every query document
- Compute $AU_k$
- Normalize it to have magnitude 1

Now compute $(XU_k)a$ using normalized versions. The result is a vector of sines of angles.

**Disambiguating Homonyms**

We started with the homonym problem
- How to select the correct word sense?

A solution: use LSA
- Collect every sentence the word occurs in from a corpus
- Perform LSA, keeping $K$ singular vectors
- $K$ should account for 85% of energy
- For every new use, find closest word sense

Advantage: we work for new terms, if you add them to corpus
Disadvantage: word sense is a column number. Linguists might still want to know what it means...
- If $CX$ decomposition is used, you can say "as used in document $y$..."

**Relating SVD to PCA**

SVD is a matrix decomposition algorithm. Applied to the raw matrix $X$, we get:

$$X = U \Sigma V^T$$

When we apply SVD to the covariance matrix, we call it PCA

$$XX^T = U \Sigma^2 U^T$$

There is therefore a second form of PCA:

$$X^T X = V \Sigma^2 V^T$$

Note: Eigenvalues are singular values squared

**Background Concepts: Variance**

Variance - the central tendency,
- variance is defined as:

$$\frac{1}{N} \sum (x_i - \bar{x})^2$$

Square root of variance is the standard deviation

**Background Concepts: Covariance**

Covariance measures if two signals vary together:

$$\Omega = \frac{1}{N} \sum (x_i - \bar{x})(y_j - \bar{y})$$

How does this differ from correlation?
The range of the covariance of two signals?
Note the covariance of two signals is a scalar

**Covariance Matrices (I)**

Let $x$ and $y$ be sets of vectors...
What if I want to know the relation between the $i$th element of $x$ and the $j$th element of $y$?

$$\frac{1}{N} \sum_{i,j} \sigma_{x_i/y_j} = \begin{bmatrix} \sigma_{x_1/y_1} & \sigma_{x_1/y_2} & \cdots \\ \sigma_{x_2/y_1} & \sigma_{x_2/y_2} & \cdots \\ \vdots & \vdots & \ddots \\ \sigma_{x_N/y_j} & \sigma_{x_{N-1}/y_j} & \cdots \\ \end{bmatrix}$$

$$\sigma_{x_i/y_j} = \sum (x_i - \bar{x})(y_j - \bar{y})$$
Background Concepts: Outer Products

Remember outer products:

\[
\begin{bmatrix}
ad & ae & af \\
bd & be & bf \\
cd & ce & cf
\end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \begin{bmatrix} d & e & f \end{bmatrix}
\]

Why?
Because if I have two vectors, their covariance term is their outer product.

Covariance Matrices (II)

The covariance between two vectors isn’t too interesting, just a set of scalars, but...

What if we have two sets of vectors:
- Let \( X = \{(a_1, b_1, c_1), (a_2, b_2, c_2)\} \)
- Let \( Y = \{(d_1, e_1, f_1), (d_2, e_2, f_2)\} \)

And assume the vectors are centered.
- Meaning that the average X vector is subtracted from the X set, and the average Y is subtracted from the Y set.

What is the covariance between the sets of vectors?

Covariance Matrices (III)

The covariance matrix is the outer product:

\[
\begin{bmatrix}
a_1 & a_2 \\
b_1 & b_2 \\
c_1 & c_2
\end{bmatrix}
\begin{bmatrix}
e_1 & f_1 \\
e_2 & f_2 \\
e_3 & f_3
\end{bmatrix}
= \begin{bmatrix}
a_1e_1 + a_2e_2 & a_1f_1 + a_2f_2 & a_1e_3 + a_2e_3 \\
b_1e_1 + b_2e_2 & b_1f_1 + b_2f_2 & b_1e_3 + b_2e_3 \\
c_1e_1 + c_2e_2 & c_1f_1 + c_2f_2 & c_1e_3 + c_2e_3
\end{bmatrix}
\]

\( \Omega_{ij} \) is the covariance of position \( i \) in set \( X \) with position \( j \) in set \( Y \),
assumes pair wise matches

Why would you do this?

Imagine the first set of vectors was demographics
- Age, weight, income...

The second set of vectors was medical info:
- Blood pressure, cholesterol, sugar levels, etc.

Then the covariance matrix tells you about relations between demographics and medical conditions
- For example, does weight correlate to cholesterol?

This is similar to the NLP (documents vs terms) case
- And can be approached with LSA

Covariance Matrices (IV)

It is interesting & meaningful to look at the covariance of a set with itself:

\[
\begin{bmatrix}
a_1 & a_2 \\
b_1 & b_2 \\
c_1 & c_2
\end{bmatrix}
\begin{bmatrix}
a_1 & a_2 \\
b_1 & b_2 \\
c_1 & c_2
\end{bmatrix} = \begin{bmatrix}
a_1a_1 + a_2a_2 & a_1a_2 + a_2a_2 & a_1a_3 + a_2a_3 \\
a_1b_1 + a_2b_2 & a_1b_2 + a_2b_2 & a_1b_3 + a_2b_3 \\
a_1c_1 + a_2c_2 & a_1c_2 + a_2c_2 & a_1c_3 + a_2c_3
\end{bmatrix}
\]

Now how do you interpret \( \Omega_{ij} \)?
Does \( \Sigma \) have any special properties?

Covariance Matrices (V)

Covariance matrices of 2D data sets (easy to draw)

What can you tell me about \( \Omega_{xy} \)?
What can you tell me about \( \Omega_{yx} \)?
Principal Component Analysis

PCA = SVD(Cov(X)) = SVD(XX'/(n-1))

SVD: XX' = RΛR^{-1}

- R is a rotation matrix (the Eigenvector matrix)
- Λ is a diagonal matrix (diagonal values are the Eigenvalues)

The Eigenvalues capture how much the dimensions in X co-vary
The Eigenvectors show which combinations of dimensions tend to vary together

PCA (II)

The Eigenvector with the largest Eigenvalue is the direction of maximum variance
The Eigenvector with the 2nd largest Eigenvalue is orthogonal to the 1st vector and has the next greatest variance.
And so on...
The Eigenvalues describe the amount of variance along the Eigenvectors

PCA vs LSA

Assume A is N x D (terms by documents)
AA' is the term covariance matrix
- SVD(AA') gives the linear relations among terms
- This is PCA
A'A is the document covariance matrix
- SVD(A'A) gives the linear relations among documents
- This is also PCA
SVD(A) gives Uk
- X = AUk gives linear relations between terms and documents
- This is LSA

The CX-decomposition

Find C that contains subset of the columns of A to minimize:

\[ \min_C \left\| A - CX \right\|_F^2 \]

\[ \left\| A \right\|_F^2 = \sum_{i,j} A_{ij}^2 \]

Given C it is easy to find X from standard least squares. However, finding C is now hard!!!

Why CX-decomposition

If A is an object-feature matrix, then selecting "representative" columns is equivalent to selecting "representative" features

This leads to easier interpretability; compare to eigenfeatures, which are linear combinations of all features.