Applications of SVD: PCA & MDS

Szymon Rusinkiewicz
COS 302, Fall 2020
- Singular Value Decomposition
- Solving linear least-squares...
  - without incurring condition-squaring effect of normal equations ($A^T A x = A^T b$)
  - when $A$ is singular, “fat”, or otherwise poorly-specified?
- Total least squares
Today: More Applications of SVD

- Principal Components Analysis
- Multi-dimensional Scaling
Principal Components Analysis (PCA)

- Approximating a high-dimensional data set with a lower-dimensional linear subspace
- Also converts possibly-correlated attributes into uncorrelated attributes
SVD and PCA

- Data matrix with points/examples as rows
- Center data by subtracting mean
- Compute (reduced) SVD
- Columns of $V$ are normalized principal components
- Each $w_i$ indicates importance of corresponding component
- Rows of $U$ are data points expressed in terms of principal components
Dimensionality Reduction

• Map points in high-dimensional space to lower number of dimensions
• (Try to) preserve structure: pairwise distances, etc.
• Useful for further processing:
  – Less computation, fewer parameters
  – Easier to understand, visualize
SVD for Rank-$k$ approximation

- $A$ is $m \times n$ matrix of rank $> k$
- Suppose you want to find best rank-$k$ approximation to $A$
- Take SVD: $A = U W V^T$
- Set all but the largest $k$ singular values of $W$ to 0
- Can form compact representation by eliminating columns of $U$ and $V$ corresponding to zeroed $w_i$
PCA on Images

- **Compression**: each new image can be approximated by projection onto first few principal components
- **Recognition**: for a new image, project onto first few principal components, match feature vectors
- **Generation**: Adjust contributions of a few principal components to generate new plausible data points
PCA on Images

\[
A = U \begin{bmatrix}
w_1 & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & w_m
\end{bmatrix} V^T
\]
PCA for Relighting

- Images under different illumination

[Image: PCA for Relighting]
PCA for Relighting

- Images under different illumination
- Most variation captured by first 5 principal components – can re-illuminate by combining only a few images

[Matusik & McMillan]
Face Recognition

• Suppose you want to recognize a *particular* face

• How does *this* face differ from average face
  – Not all variations equally important
    (variation in a single pixel relatively unimportant)

• If images are high-dimensional vectors, want to find directions in this space with high variation
  – PCA!
PCA on Faces: “Eigenfaces”

For all except average,
“gray” = 0,
“white” > 0,
“black” < 0
Using PCA for Recognition

• Compute PCA basis using training set
• Store each person as coefficients of projection onto first few principal components

\[
\text{image} = \text{average} + \sum_{i=1}^{i_{\text{max}}} a_i \text{Eigenface}_i
\]
Using PCA for Recognition

- Compute PCA basis using training set
- Store each person as coefficients of projection onto first few principal components
- For a new image: calculate coefficients

\[ a_i = (\text{image} - \text{average}) \cdot \text{Eigenface}_i \]
Using PCA for Recognition

• Compute PCA basis using training set
• Store each person as coefficients of projection onto first few principal components
• For a new image: calculate coefficients
• Is it a face?

$$\left\| \text{image} - \left( \text{average} + \sum_{i=1}^{i_{\text{max}}} a_i \text{Eigenface}_i \right) \right\| < \text{threshold?}$$
Using PCA for Recognition

- Compute PCA basis using training set
- Store each person as coefficients of projection onto first few principal components
- For a new image: calculate coefficients
- Is it a face?
- If a face, find image in database with closest $a_i$
  - “Nearest-neighbor classifier”
Choosing the Dimension $k$

- How many eigenfaces to use?
- Look at the decay of the singular values
  - Singular value gives the amount of variance “in the direction” of that eigenface
PCA for DNA Microarrays

- Measure gene activation under different conditions
PCA for DNA Microarrays

- Measure gene activation under different conditions
PCA for DNA Microarrays

- PCA shows patterns of correlated activation
  - Genes with same pattern might have similar function
PCA for DNA Microarrays

- PCA shows patterns of correlated activation
  - Genes with same pattern might have similar function
PCA for Music
Practical Considerations for PCA

• Sensitive to scale of each attribute (column)
  – In practice, may “standardize” by scaling each attribute to have unit variance

• Sensitive to noisy attributes
  – Just because a dimension is highly weighted by PCA doesn’t mean it’s relevant, informative, etc.
Multidimensional Scaling
Multidimensional Scaling

• In some experiments, can only measure similarity or dissimilarity
  – e.g., are responses to stimuli similar or different? How different are they?
  – Frequent in psychophysical experiments, preference surveys, etc.

• Want to recover absolute positions in $k$-dimensional space
**Multidimensional Scaling**

- Example: given pairwise distances between cities

<table>
<thead>
<tr>
<th></th>
<th>Atl</th>
<th>Chi</th>
<th>Den</th>
<th>Hou</th>
<th>LA</th>
<th>Mia</th>
<th>NYC</th>
<th>SF</th>
<th>Sea</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chicago</td>
<td>587</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Denver</td>
<td>1212</td>
<td>920</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Houston</td>
<td>701</td>
<td>940</td>
<td>879</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LA</td>
<td>1936</td>
<td>1745</td>
<td>831</td>
<td>1374</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miami</td>
<td>604</td>
<td>1188</td>
<td>1726</td>
<td>968</td>
<td>2339</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NYC</td>
<td>748</td>
<td>713</td>
<td>1631</td>
<td>1420</td>
<td>2451</td>
<td>1092</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SF</td>
<td>2139</td>
<td>1858</td>
<td>949</td>
<td>1645</td>
<td>347</td>
<td>2594</td>
<td>2571</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seattle</td>
<td>2182</td>
<td>1737</td>
<td>1021</td>
<td>1891</td>
<td>959</td>
<td>2734</td>
<td>2406</td>
<td>678</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>543</td>
<td>597</td>
<td>1494</td>
<td>1220</td>
<td>2300</td>
<td>923</td>
<td>205</td>
<td>2442</td>
<td>2329</td>
<td>0</td>
</tr>
</tbody>
</table>

Want to recover \((x, y)\) locations

[Pellacini et al.]
Euclidean MDS

• Formally, let’s say we have $n \times n$ matrix $D$ consisting of squared distances $d_{ij} = ||x_i - x_j||^2$

• Want to recover $n \times k$ matrix $X$ of positions in $k$-dimensional space

$$D = \begin{pmatrix} 0 & (x_1 - x_2)^2 & (x_1 - x_3)^2 \\ (x_1 - x_2)^2 & 0 & (x_2 - x_3)^2 \\ (x_1 - x_3)^2 & (x_2 - x_3)^2 & 0 \\ & & \ddots \end{pmatrix}$$

$$X = \begin{pmatrix} \cdots x_1 \cdots \\ \cdots x_2 \cdots \\ \vdots \end{pmatrix}$$
Euclidean MDS

- Observe that

\[ d_{ij}^2 = (x_i - x_j)^2 = x_i^2 - 2x_i x_j + x_j^2 \]

- Strategy: convert matrix \( D \) of \( d_{ij}^2 \) into matrix \( B \) of \( x_i x_j \)
  - “Centered” distance matrix
  - Then decompose \( B = XX^T \)
Euclidean MDS

• Centering:
  – Sum of row $i$ of $D = \text{sum of column } i \text{ of } D =$
    \[
s_i = \sum_j d_{ij}^2 = \sum_j x_i^2 - 2x_i x_j + x_j^2
    \]
    \[
    = nx_i^2 - 2x_i \sum_j x_j + \sum_j x_j^2
    \]
  – Sum of all entries in $D =$
    \[
s = \sum_i s_i = 2n \sum_i x_i^2 - 2 \left( \sum_i x_i \right)^2
    \]
Euclidean MDS

• Choose $\Sigma x_i = 0$
  – Solution will have average position at origin

\[ s_i = nx_i^2 + \sum_j x_j^2, \quad s = 2n \sum_j x_j^2 \]

  – Then,

\[ d_{ij}^2 - \frac{1}{n}s_i - \frac{1}{n}s_j + \frac{1}{n^2}s = -2x_ix_j \]

• So, to get $B$:
  – compute row (or column) sums
  – compute sum of sums
  – apply above formula to each entry of $D$
  – Divide by $-2$
Factoring $B = XX^T$ using SVD

- Now have $B$, want to factor into $XX^T$
- If $X$ is $n \times k$, $B$ must have rank $k$
- Take SVD, set all but top $k$ singular values to 0
  - Eliminate corresponding columns of $U$ and $V$
  - Have $B' = U'W'V'^T$
  - $B'$ is square and symmetric, so $U' = V'$
  - Take $X = U'$ times square root of $W'$
Multidimensional Scaling

- Result \((k = 2)\):

[Map of the United States showing cities like Seattle, SF, LA, Denver, Chicago, DC, NYC, Atlanta, Houston, Miami]
Another application

Figure 2 (a) RMDS of children’s similarity judgments about 15 body parts: (b) RMDS of adults’ similarity judgments about 15 body parts.

From Young 1985 / Jacobowitz 1973
Perceptual Mapping for Marketing
Multidimensional Scaling

- Caveat: actual axes, center not necessarily what you want (can’t recover them!)
- This is “classical” or “Euclidean” MDS [Torgerson 52]
  - Distance matrix assumed to be actual Euclidean distance
- More sophisticated versions available
  - “Non-metric MDS”: not Euclidean distance, sometimes just inequalities
  - Replicated MDS: for multiple data sources (e.g. people)
  - “Weighted MDS”: account for observer bias