Last time

LSH

1. Important technique for solving approx nearest neighbor queries.
2. Idea: construct hash functions that are likely to map similar items to same bucket.

Today

PCA

Principal Component Analysis

Data dependent dimension reduction.

data set $x_1, \ldots, x_m$

$x_i \in \mathbb{R}^n$
Table 1: Your friends' ratings of four different foods.

<table>
<thead>
<tr>
<th></th>
<th>kale</th>
<th>taco bell</th>
<th>sashimi</th>
<th>pop tarts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5.5</td>
<td>-3.5</td>
<td>-3</td>
<td>1.5</td>
</tr>
<tr>
<td>Bob</td>
<td>1.5</td>
<td>-3</td>
<td>-4</td>
<td>4.5</td>
</tr>
<tr>
<td>Carolyn</td>
<td>-3.5</td>
<td>4.5</td>
<td>2</td>
<td>-2.5</td>
</tr>
<tr>
<td>Dave</td>
<td>-2.5</td>
<td>1.5</td>
<td>5</td>
<td>-3.5</td>
</tr>
</tbody>
</table>

\[ \bar{x} = (5.5, 1.5, 4.5, 4.5) \]

![Figure 1: Visualizing 4-dimensional data in the plane.](image)

The goal of dimensionality reduction vs PCA:

- CARES ABOUT PRESERVING DISTANCES
- COORDINATES HAD NO MEANING
- CLANNY DIMENSIONS TO PRESERVE DISTANCES

JL dim reduction vs PCA:

- CARES ABOUT PRESERVING DISTANCES (1st goal)
- COORDINATES HAD NO MEANING (2nd goal)
Figure 4: The geometry of the inner product.

**Objective:** Choose $v$ so as to minimize

$$\minimize \frac{1}{m} \sum_{i=1}^{m} [\text{dist}(x_i: \text{to line defined by } v)]^2$$

Claim: this is the same thing as choosing $v$ of length $1$ to maximize

$$\maximize \frac{1}{m} \sum_{i=1}^{m} (x_i, v)^2$$

\[ \mathbf{X} = (x_i, v) \mathbf{w} \mathbf{w}^T \]
Figure 5: For the good line, the projection of the points onto the line keeps the two clusters separated, while the projection onto the bad line merges the two clusters.

- Larger $k$ objective
- Find $k$-dim subspace $S$ so as to max $\frac{1}{m} \sum_{i=1}^{m} (\text{length of } x_i \text{ projected to } S)^2$

Find set of $k$ orthonormal vectors.

- $||v_i||^2 = 1$
- $(v_i, v_j) = 0 \, \forall \, i \neq j$
- $(v_i, v_i) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

- Subspace spanned by all vectors that can be written as combo $\sum_{j=1}^{k} c_j v_j$

- $p = a_1 v_1 + a_2 v_2$
- $(x, p) = (x, v_1)$
- $(x, v_1) = (p, v_1) = (a_1 v_1 + a_2 v_2, v_1)$
- $= a_1 (v_1, v_1) + a_2 (v_2, v_1)$

- $a_1 = (x, v_1)$
- $\|p\|^2 = (a_1 v_1 + a_2 v_2, a_1 v_1 + a_2 v_2)$
- $= a_1^2 + a_2^2$
- $= (x, v_1)^2 + (x, v_2)^2$. 

\[ x \]
Objective: Find $v_1, \ldots, v_k$ orthogonal to maximize \[
\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} (x_i, v_j)^2
\]
Average sum of squared projected lengths.

Application: Visualization.

1. Perform PCA $\rightarrow v_1, \ldots, v_k$ take principal components
2. For $x_i$: define "$v_1$ coord" "$v_2$ coord" ... "$v_k$ coord"
3. Plot your pts $x_i = (\langle x_i, v_1 \rangle, \langle x_i, v_2 \rangle, \ldots, \langle x_i, v_k \rangle)$

- Look for clusters
- Look what are pts particularly large along $v_1$

Generic data of Europeans $\text{SNPs}$ $A, T, C, G$ 300,000 people

$k = 2$
Figure 1: The genetic map of Europe using PCA, with the geographic map of Europe for reference. Figure 2: The same map, but zoomed in on Switzerland. Swiss individuals tend to cluster with countries that speak the same language. (Courtesy: John Novembre, UCLA)

Application 2: Compression

Eigenvalues.
500 faces

2 examples of images after subtracting mean face.

first 4 principal components
Original Face Image

eigenfaces space
How PCA works

Find $v$ st. $\|v\|=1$ to maximize

$$\sum_{i=1}^{m} (x_{i}v)^2$$

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \Rightarrow Xv = \begin{pmatrix} (x_{1}v) \\ \vdots \\ (x_{m}v) \end{pmatrix}$$

$$A = XX^T$$

$$v^TAv = \sum_{i=1}^{m} (x_{i}v)^2$$

Find $v$ to max $v^TAv$.

$A_{kl} = \sum_{i=1}^{m} x_{i}x_{i} |_{k,l}$

$A$ is symmetric.
Suppose rows of $X$ are documents, cols words.

\[
\max_{\|v\|=1} v^T A v
\]

Suppose $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$

\[
v = (v_1, v_2, v_3)
\]

\[
v^T A v \Rightarrow 2v_1^2 + 1v_2^2 + \frac{1}{2}v_3^2
\]

\[
v_1^2 + v_2^2 + v_3^2 = 1
\]

$v_1 = 1$

Every syndrome

\[
\begin{pmatrix}
2 & 0 \\
0 & \ddots \\
0 & \ddots & 2
\end{pmatrix}
\]

\[
\sum_{i=1}^{m} v_i^2 = 1
\]

\[
\sum_{i=1}^{m} v_i^2 = \sum_{i=1}^{m} v_i^2 = 1
\]
Figure 1: The point \((x, y)\) on the unit circle is mapped to \((2x, y)\).

Figure 2: The same scaling as Figure 1, but now rotated 45 degrees.

\[
\begin{pmatrix}
2 & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
\frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & -\frac{1}{2}
\end{pmatrix}
= \begin{pmatrix}
\frac{2}{\sqrt{2}} & \frac{0}{\sqrt{2}} \\
-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}
\end{pmatrix}
\]

\[
A = Q D Q^T
\]

\[
Q = \begin{pmatrix}
q_1 & q_2 & \ldots & q_n
\end{pmatrix}
\]

Orthogonal \iff \text{columns are orthonormal}

\[
\|q_i\|^2 = 1 \quad (q_i; q_i) = 0 \quad \forall i \neq j
\]

Special
\[
(Qv)^T Qv = v^T Q^T Q v = v^T v
\]

Orthogonal matrices preserve length.

\[
QQ^T = I
\quad Q^T Q = I
\]
\[ \max \ v^T A v = v^T \Phi D \Phi^T v \]

\[ y = \Phi^T v \]

\[ y_1 = 1 \Rightarrow y_i = 1 \quad i > 1 \]

\[ y_1 = 1 \quad y_i = 0 \quad i > 1 \]

\[ z_i > z_j > \ldots \]

\[ y_i = e_i \quad e_i = \Phi^T v \]

\[ \Phi e_i = \Phi \Phi^T v = v \]

\[ \lambda_1 e_1 = q_1 \quad (\text{first col of } Q) \]

\[ M_z = \begin{bmatrix} 2 \end{bmatrix} \]

\[ z_i \text{ is eigenvector of eigenvalue } \lambda \]

\[ Q e_i \text{ is eigenvector of matrix } A \]

\[ A e_i = \Phi D \Phi^T e_i = \Phi \Phi^T e_i = \Phi \begin{bmatrix} \lambda_i e_i \end{bmatrix} = \lambda_i \Phi e_i \]

\[ v^T A v \]

\[ \text{Solution for } k = 1 \text{ : longest eigenvector of } A = X^T X \]

\[ A = \Phi D \Phi^T \]
to do PCA up k = 1
find unit vector $v$ that maximizes $v^T X v$

principal eigenvector of $A$

$A = G D G^T$

$u_0$ random vector, eigenvectors $q_1, \ldots, q_n$ are a basis

$u_0 = \sum_{j=1}^{n} c_j q_j$

$A u_0 = \sum_{j} c_j A q_j = \sum_{j} c_j \lambda_j q_j$

$A \sum_{j} c_j q_j = \sum_{j} c_j \lambda_j q_j = \sum_{j} c_j \lambda_j^3 q_j$

$A \sum_{j} c_j q_j = \sum_{j} c_j \lambda_j^3 q_j$

$A^k u_0 = \sum_{j} c_j \lambda_j^k q_j$

$= \sum_{j} \lambda_j^k [c_j q_j]$

recall $q_1$
Algorithm 1
Power Iteration

Given matrix $A \equiv X^T X$:

- Select random unit vector $u_0$
- For $i = 1, 2, \ldots$, set $u_i = A u_{i-1}$. If $u_i / ||u_i||$ is normalized, then return $u_i / ||u_i||$. 

PageRank.
1. Find the top component, \( v_1 \), using power iteration.

2. Project the data matrix orthogonally to \( v_1 \):

\[
\begin{bmatrix}
\mathbf{X}_1 \\
\mathbf{X}_2 \\
\vdots \\
\mathbf{X}_n
\end{bmatrix} 
= \mathbf{X}_1 - \langle \mathbf{X}_1, v_1 \rangle v_1 \cdot v_1^t
\]

This corresponds to subtracting out the variance of dimension \( v_1 \) that is already explained by the first principal component \( v_1 \).

3. Recurse by finding the top \( k - 1 \) principal components of the new data matrix.

---

**Scree plot.** Principal components are ranked by the amount of variance they capture in the original dataset; a scree plot can provide some sense of how many components are needed.