CSE 664
Visualization & Visual Analytics

Data and Dimension Reduction

Klaus Mueller

Computer Science Department
Stony Brook University
<table>
<thead>
<tr>
<th>Lecture</th>
<th>Topic</th>
<th>Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro, schedule, and logistics</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Applications of visual analytics, basic tasks, data types</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Data sources and introduction to D3</td>
<td>Project #1 out</td>
</tr>
<tr>
<td>4</td>
<td>Data wrangling and preparation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Bias in visualization</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><strong>Data reduction and dimension reduction</strong></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Visual perception and cognition</td>
<td>Project #1 due</td>
</tr>
<tr>
<td>8</td>
<td>Visual design and aesthetics</td>
<td>Project #2 out</td>
</tr>
<tr>
<td>9</td>
<td>Python/Flask hands-on</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Cluster analysis: numerical data</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Cluster analysis: categorical data</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Foundations of scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Computer graphics and volume rendering</td>
<td>Project #2 due / Project #3 out</td>
</tr>
<tr>
<td>14</td>
<td>Scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Illustrative rendering</td>
<td>Project #3 due</td>
</tr>
<tr>
<td>16</td>
<td>High-dimensional data, dimensionality reduction</td>
<td>Final project proposal call out</td>
</tr>
<tr>
<td>17</td>
<td>Correlation visualization</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Principles of interaction</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Midterm #1</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Visual analytics and the visual sense making process</td>
<td>Final project proposal due</td>
</tr>
<tr>
<td>21</td>
<td>Evaluation and user studies</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Visualization of time-varying and time-series data</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Visualization of streaming data</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Visualization of graph data</td>
<td>Final Project preliminary report due</td>
</tr>
<tr>
<td>25</td>
<td>Visualization of text data</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Midterm #2</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Data journalism</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Final project presentations</td>
<td>Final Project slides and final report due</td>
</tr>
</tbody>
</table>
Data Reduction – How?

Reduce the number of data items (samples):
- random sampling
- stratified sampling

Reduce the number of attributes (dimensions):
- dimension reduction by transformation
- dimension reduction by elimination

Usually do both

Utmost goal
- keep the gist of the data
- only throw away what is redundant or superfluous
- it’s a one way street – once it’s gone, it’s gone
Data Reduction

Sampling
- random
- stratified

Data summarization
- binning (already discussed)
- clustering (see a future lecture)
- dimension reduction (see next lecture)
Because...

- need to reduce the data so they can be feasibly stored
- need to reduce the data so a mining algorithm can be feasibly run

What else could we do

- buy more storage
- buy more computers or faster ones
- develop more efficient algorithms (look beyond $O$-notation)

However, in practice, all of this is happening at the same time

- unfortunately, the growth of data and complexities is always faster
- and so, data reduction will always be important
Good candidates are *redundant* data

- how many cans of ravioli will you buy?
Sampling Principles

Keep a representative number of samples:

- pick one of each
- or maybe a few more depending on importance
You are faced with collections of many different data

- they are usually not nicely organized like this:

- but more like this:
Are all of these items pants?

- need a measure of similarity
- it’s a distance measure in high-dimensional feature space
We did not consider color, texture, size, etc...
- this would have brought more differentiation (blue vs. tan pants)
- the more features, the better the differentiation
How Many Features Do We Need?

Measuring similarity can be difficult
needs to be accurately measured

quantize each person into a vector
each vector element is a feature measurement
compare the vectors in terms of similarity
similarity is also called a distance function
Pant:
  <length, ornateness, color>

Food delivery customer:
  <type-pizza, type-salad, type-drink>

Examples:
- pants: <long, plain, tan>, <short, ornate, blue>, ...  
  expressed in numbers: <30\", 1, 2>, <15\", 2, 5>
- food: <pepperoni, tossed, none>, <pepperoni, tossed, coke>, ...  
  expressed in numbers: <1, 1, 0>, <1, 1, 3>
**Metric Distances**

**Manhattan distance**

\[
\text{dist}(a,b) = \|a - b\|_1 = \sum_i |a_i - b_i|
\]

**Euclidian distance**

\[
\text{dist}(a,b) = \|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}
\]
Pearson’s Correlation = correlation similarity

\[ s(x, y) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_{i=0}^{n-1} x_i y_i}{\sqrt{\sum_{i=0}^{n-1} (x_i)^2} \times \sqrt{\sum_{i=0}^{n-1} (y_i)^2}} \]

how is this related to correlation?

mean across all data values for attribute \( x, y \)

e.g. the “average looking” pair of pants in terms of attribute \( x, y \)
Correlation distance is invariant to addition of a constant

- subtracts out by construction
- green and blue curve have correlation of 1
- but cosine similarity is $< 1$
- correlated vectors just vary in the same way
- cosine similarity is stricter

Both correlation and cosine similarity are invariant to multiplication with a constant

- invariant to scaling

green = blue + 0.1
Distances can compare two attributes or two data items

- means and other stats are then measured correspondingly
- mean and std dev mileage and weight, resp. over all cars when computing correlation of weight and mileage
- mean of all attribute values for each car when computing the distance between two cars
What’s the Jaccard similarity of the two baskets A and B?
This process is called clustering

- and in contrast to a real store, we can make the computer do it for us
Note:
- in data mining similarity and distance are the same thing
- so we will use these terms interchangeably

Clustering = grouping of similar items (as determined by the distance function)
A cluster is a group of objects that are similar
- and dissimilar from other groups of objects at the same time

We need an objective function to capture this mathematically
- the computer will evaluate this function within an algorithm
- one such function is the mean-squared error (MSE)
- and the objective is to minimize the MSE

It’s not that easy in practice
- there is only one global minimum
- but often there are many local minima
- need to find the global minimum
In this case
- \( n = 12 \) (blue points)
- \( k = 2 \) (red points, the computed centroids)
- distance metric used: Euclidian
- minimization seems to be achieved
1. Decide on a value for $k$

2. Initialize the $k$ cluster centers (randomly, if necessary)

3. Decide the class memberships of the $N$ objects by assigning them to the nearest cluster center

4. Re-estimate the $k$ cluster centers, by assuming the memberships found above are correct

5. If none of the $N$ objects changed membership in the last iteration, exit. Otherwise goto 3
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance
K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance
K–Means Algorithm – Comments

Strengths:
- relatively efficient: $O(tkn)$, where $n$ is # objects, $k$ is # clusters, and $t$ is # iterations. Normally, $k, t << n$.
- simple to code

Weaknesses:
- need to specify $k$ in advance which is often unknown
- find the best $k$ by trying many different ones and picking the one with the lowest error
- often terminates at a local optimum
- the global optimum may be found by trying many times and using the best result
How Can We Find the Best K?

k=1, MSE=873.0

k=2, MSE=173.1

k=3, MSE=133.6
Is there a principled way we can know when to stop looking? Yes...

- we can plot the objective function values for k equals 1 to 6...
- then check for a flattening of the curve

the abrupt change at k = 2 is highly suggestive of two clusters
- this technique is known as “knee finding” or “elbow finding”
What is sampling?
- pick a **representative** subset of the data
- discard the remaining data
- pick as many you can afford to keep
- recall: once it’s gone, it’s gone
- be smart about it

Simplest: random sampling
- pick sample points at random
- will work if the points are distributed uniformly
- this is usually not the case
- outliers will likely be missed
- so the sample will not be representative
Pick the samples according to some knowledge of the data distribution

- cluster the data (outliers will form clusters as well)
- these clusters are also called *strata* (hence, stratified sampling)
- the size of each cluster represents its percentage in the population
- guides the number of samples – bigger clusters get more samples

sampling rate ~ cluster size
Eliminate redundant attributes

- eliminate correlated attributes
  - km vs. miles
  - \( a + b + c = d \) \( \rightarrow \) can eliminate ‘c’ (or ‘a’ or ‘b’)

Eliminate redundant data

- cluster the data with small ranges \( \varepsilon \)
- only keep the cluster centroids
- store size of clusters along to keep importance

question: how do we find a good \( \varepsilon \)?

answer: compute histogram of distances

choose a reasonable threshold from the left
Reservoir Sampling

/*
* S has items to sample, R will contain the result
*/
ReservoirSample(S[1..n], R[1..k])
  // fill the reservoir array
  for i = 1 to k
    R[i] := S[i]

  // replace elements with gradually decreasing probability
  for i = k+1 to n
    j := random(1, i) // important: inclusive range
    if j <= k
      R[j] := S[i]

Probabilities

- $k/i$ for the $i^{th}$ sample to go into the reservoir
- $1/k \cdot k/i = 1/i$ for the $j^{th}$ reservoir element to be replaced
- $k/n$ for all elements in the reservoir after $n$ has been reached
- can be shown via induction

A good algorithm to use for streaming data when $n$ is growing
Used in the CURE high-dimensional clustering algorithm


Algorithm

- initialize the point set \( S \) to empty
- pick the point farthest from the mean as the first point for \( S \)
- then iteratively pick points that are furthest from the points in \( S \) collected so far

Complexity is \( O(m \cdot n^2) \)

- \( n \) is the total number of points, \( m \) is the number of desired points
- can find arbitrarily shaped clusters and preserve outliers, too
- need some good data structures to run efficiently: kd-tree, heap
Dimension Reduction
Are there attributes that “go together”?

Can you name a few?
Physical attributes

- color
- number of doors
- number of wheels
- retractable roof
- height
- length
- frames around side windows

Which attributes are useful to distinguish SUVs from convertibles?

- number of doors (4 vs. 2) --> numerical, two levels
- retractable roof (no vs. yes) --> categorical, two levels
- frames around side windows (yes vs. no) --> categorical, two levels
- height (higher vs. lower) --> numerical, many levels
Which attributes are not so useful?

- number of wheels (constant 4) --> no discriminative power
- length (short and long SUVs, convertibles) --> confounding
- color (colors are seemingly random, or are they?)

Is color useful?

- the convertibles seem to have more vibrant colors (red, yellow, ...)
- so maybe we made a discovery
Need to consider more than two attributes

- *height* attribute would have distinguished the Range Rover from the convertibles and caused it to be an outlier
New classes are constantly evolving over time

- this is known as *cluster evolution*
- measuring more features will increase the chance of discovery
The more data (examples) the better
  - increases the chances to discover the rare specimen

- but some attributes are useless
- we can cull them away
- perform attribute reduction or \textit{dimension reduction}
By axis rotation

- determine a more efficient basis
- Principal Component Analysis (PCA)
- Singular value decomposition (SVD)
- Latent semantic analysis (LSA)

By type transformation

- determine a more efficient data type
- Fourier analysis and Wavelets for grids
- Multidimensional scaling (MSD) for graphs
- Locally Linear Embedding
- Isomap
- Self Organizing Maps (SOM)
- Linear Discriminant Analysis (LDA)
PRINCIPAL COMPONENT ANALYSIS (PCA)
Covariance

- measures how much two random variables change together

For N variable we have $N^2$ variable pairs

- we can write them in a matrix of size $N^2 \rightarrow$ the covariance matrix
- for two variables $X_1$ and $X_2$

$$
\text{Var}[X] = \begin{bmatrix}
\text{Var}[X_1] & \text{Cov}[X_1,X_2] \\
\text{Cov}[X_2,X_1] & \text{Var}[X_2]
\end{bmatrix}
$$
Covariance \( \text{cov}(X,Y) \)

\[
\text{COV}(X,Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1}
\]

Pearson’s correlation \( r \)
- is covariance normalized by the individual variances for \( X \) and \( Y \)

\[
R_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}
\]
Correlation rates between -1 and 1:

1.0  0.8  0.4  0.0  -0.4  -0.8  -1.0

Important to note:
- correlation is defined for linear relationships
- visualization can help
- none of these point distributions have correlations:
Analytical: \[ \text{Cov}(X, Y) = E[(X - \mu_x)(Y - \mu_y)] \]

Samples: \[ \sigma_{xy} = \text{cov}_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \]

An n-D dataset has \( n \) variables \( x_1, x_2, \ldots, x_n \)
- define pairwise covariance among all of these variables
- construct a covariance matrix

\[ \Sigma = \text{Cov}(X) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix} \]

- a correlation matrix would just list the correlations instead
# Correlation Matrix

<table>
<thead>
<tr>
<th></th>
<th>MO</th>
<th>FP</th>
<th>MP</th>
<th>IM</th>
<th>IC</th>
<th>FM</th>
<th>FE</th>
<th>FI</th>
<th>SPC</th>
<th>DSC</th>
<th>DST</th>
</tr>
</thead>
<tbody>
<tr>
<td>MO</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>0.31&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP</td>
<td>0.32&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.71&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IM</td>
<td>0.36&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.12&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.14&lt;sup&gt;c&lt;/sup&gt;</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IC</td>
<td>0.39&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.18&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0.21&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.62&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FM</td>
<td>0.26&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.21&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.14&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.30&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.27&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE</td>
<td>0.47&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.21&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.18&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0.38&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.28&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.24&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>0.53&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.26&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.22&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.36&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.37&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.29&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.47&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPC</td>
<td>0.32&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.22&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.31&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.51&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.47&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.32&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.37&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.35&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSC</td>
<td>−0.12&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.03&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.05&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.17&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0.08&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.18&lt;sup&gt;b&lt;/sup&gt;</td>
<td>−0.05&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.06&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.01&lt;sup&gt;c&lt;/sup&gt;</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>DST</td>
<td>−0.02&lt;sup&gt;c&lt;/sup&gt;</td>
<td>−0.01&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.05&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.24&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.14&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.05&lt;sup&gt;c&lt;/sup&gt;</td>
<td>−0.05&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.05&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.05&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.56&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.00</td>
</tr>
<tr>
<td>DM</td>
<td>0.05&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.144</td>
<td>0.136&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.199&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.169&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0.247&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.08&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.11&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.14&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0.46&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0.71&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

---

### Just Value

**Climatic Predictors**

<table>
<thead>
<tr>
<th>WetDays</th>
<th>TempJuly</th>
<th>TempJan</th>
<th>TempAnn</th>
<th>RHJuly</th>
</tr>
</thead>
</table>

---

### Distribution (Scatterplot Matrix)
Ultimate goal:

- find a coordinate system that can represent the variance in the data with as few axes as possible

- rank these axes by the amount of variance (blue, red)
- drop the axes that have the least variance (red)
Principal Components

2nd Principal Component, $y_2$

1st Principal Component, $y_1$
Find the principal components (factors) of a distribution

First characterize the distribution by

- covariance matrix Cov
- correlation matrix Corr
- lets call it C

- perform QR factorization or LU decomposition on that matrix to get

\[ C = QQ^{-1} \Lambda \]

- Q: matrix with Eigenvectors
- \( \Lambda \): diagonal matrix with Eigenvalues \( \lambda \)

- now order the Eigenvectors in terms of their Eigenvalues \( \lambda \)
$\lambda_1, \lambda_2$ are the Eigenvalues
- encode the length (and therefore significance) of the Eigenvectors
When to use what?

- use covariance matrix when the variable scales are similar
- use correlation matrix when the variables are on different scales
- the correlation matrix *standardizes* the data
- in general they give different results, especially when the scales are different
Before PCA

PC 1

PC 2

Variable $X_1$

Variable $X_2$
After PCA

- $\lambda_1 = 9.8783$  $\lambda_2 = 3.0308$  Trace = 12.9091
- PC 1 displays ("explains") $9.8783/12.9091 = 76.5\%$ of total variance
Create a *scree plot*

- plots a histogram of the Eigenvalues ordered by magnitude
- plots the explained variance as a curve

possible threshold (explain 75% of data variance)

Keep top 3 principal components → reduce dimensions by a factor of $4/7 = 57\%$
PCA APPLIED TO FACES

Some familiar faces...
We can reconstruct each face as a linear combination of “basis” faces, or Eigenfaces [M. Turk and A. Pentland (1991)]
90% variance is captured by the first 50 eigenvectors.

Reconstruct existing faces using only 50 basis images.

We can also generate new faces by combining eigenvectors with different weights.
A More Challenging Example

• Data from research on habitat definition in the endangered Baw Baw frog

• 16 environmental and structural variables measured at each of 124 sites

• Correlation matrix used because variables have different units

Philoria frosti
# Eigenvalues

<table>
<thead>
<tr>
<th>Axis</th>
<th>Eigenvalue</th>
<th>% of Variance</th>
<th>Cumulative % of Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.855</td>
<td>36.60</td>
<td>36.60</td>
</tr>
<tr>
<td>2</td>
<td>3.420</td>
<td>21.38</td>
<td>57.97</td>
</tr>
<tr>
<td>3</td>
<td>1.122</td>
<td>7.01</td>
<td>64.98</td>
</tr>
<tr>
<td>4</td>
<td>1.116</td>
<td>6.97</td>
<td>71.95</td>
</tr>
<tr>
<td>5</td>
<td>0.982</td>
<td>6.14</td>
<td>78.09</td>
</tr>
<tr>
<td>6</td>
<td>0.725</td>
<td>4.53</td>
<td>82.62</td>
</tr>
<tr>
<td>7</td>
<td>0.563</td>
<td>3.52</td>
<td>86.14</td>
</tr>
<tr>
<td>8</td>
<td>0.529</td>
<td>3.31</td>
<td>89.45</td>
</tr>
<tr>
<td>9</td>
<td>0.476</td>
<td>2.98</td>
<td>92.42</td>
</tr>
<tr>
<td>10</td>
<td>0.375</td>
<td>2.35</td>
<td>94.77</td>
</tr>
</tbody>
</table>
How Many Axes Are Needed?

- Does the \((k+1)^{th}\) principal axis represent more variance than would be expected by chance?
- Several tests and rules have been proposed
- A common “rule of thumb” when PCA is based on correlations is that axes with eigenvalues > 1 are worth interpreting
- In our example 4 Eigenvectors fit this criterion (we shall keep 3 for simplicity)
Interpreting Eigenvectors

- Correlations between variables and the principal axes are known as loadings.
- Each element of the eigenvectors represents the contribution of a given variable to a component.
- The loadings of variables on the first three PCs are shown here.

<table>
<thead>
<tr>
<th>Variable</th>
<th>PC 1</th>
<th>PC 2</th>
<th>PC 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude</td>
<td>0.3842</td>
<td>0.0659</td>
<td>-0.1177</td>
</tr>
<tr>
<td>pH</td>
<td>-0.1159</td>
<td>0.1696</td>
<td>-0.5578</td>
</tr>
<tr>
<td>Cond</td>
<td>-0.2729</td>
<td>-0.1200</td>
<td>0.3636</td>
</tr>
<tr>
<td>TempSurf</td>
<td>0.0538</td>
<td>-0.2800</td>
<td>0.2621</td>
</tr>
<tr>
<td>Relief</td>
<td>-0.0765</td>
<td>0.3855</td>
<td>-0.1462</td>
</tr>
<tr>
<td>maxERht</td>
<td>0.0248</td>
<td>0.4879</td>
<td>0.2426</td>
</tr>
<tr>
<td>avERht</td>
<td>0.0599</td>
<td>0.4568</td>
<td>0.2497</td>
</tr>
<tr>
<td>%ER</td>
<td>0.0789</td>
<td>0.4223</td>
<td>0.2278</td>
</tr>
<tr>
<td>%VEG</td>
<td>0.3305</td>
<td>-0.2087</td>
<td>-0.0276</td>
</tr>
<tr>
<td>%LIT</td>
<td>-0.3053</td>
<td>0.1226</td>
<td>0.1145</td>
</tr>
<tr>
<td>%LOG</td>
<td>-0.3144</td>
<td>0.0402</td>
<td>-0.1067</td>
</tr>
<tr>
<td>%W</td>
<td>-0.0886</td>
<td>-0.0654</td>
<td>-0.1171</td>
</tr>
<tr>
<td>H1Moss</td>
<td>0.1364</td>
<td>-0.1262</td>
<td>0.4761</td>
</tr>
<tr>
<td>DistSWH</td>
<td>-0.3787</td>
<td>0.0101</td>
<td>0.0042</td>
</tr>
<tr>
<td>DistSW</td>
<td>-0.3494</td>
<td>-0.1283</td>
<td>0.1166</td>
</tr>
<tr>
<td>DistMF</td>
<td>0.3899</td>
<td>0.0586</td>
<td>-0.0175</td>
</tr>
</tbody>
</table>
What’s a “Loading”? 

- The amount of weight a data dimension has on a principal component
  - petal length/width have a high loading on PC1
  - sepal width has a high loading on PC2

- Another observation
  - projection into PC basis can also bring out clusters better
  - since spread is maximized
Significance of Variables

- We can compute the significance of the variables as the sum of squared loadings on to the most significant Eigenvectors we selected (3 in our example).

- The next slide shows the table of the last slide expanded with these squared loadings.

- We can then sort the table by the squared loadings and make a scree plot.

- The most significant variables are those above some chosen cutoff, for example 0.4 (marked in yellow in the table).
## Significance of Variables

<table>
<thead>
<tr>
<th></th>
<th>PC 1</th>
<th>PC 2</th>
<th>PC 3</th>
<th>sum of squared loadings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude</td>
<td>0.3842</td>
<td>0.0659</td>
<td>-0.1177</td>
<td>0.41</td>
</tr>
<tr>
<td>pH</td>
<td>-0.1159</td>
<td>0.1696</td>
<td>-0.5578</td>
<td>0.59</td>
</tr>
<tr>
<td>Cond</td>
<td>-0.2729</td>
<td>-0.1200</td>
<td>0.3636</td>
<td>0.47</td>
</tr>
<tr>
<td>TempSurf</td>
<td>0.0538</td>
<td>-0.2800</td>
<td>0.2621</td>
<td>0.39</td>
</tr>
<tr>
<td>Relief</td>
<td>-0.0765</td>
<td>0.3855</td>
<td>-0.1462</td>
<td>0.42</td>
</tr>
<tr>
<td>maxERht</td>
<td>0.0248</td>
<td>0.4879</td>
<td>0.2426</td>
<td>0.55</td>
</tr>
<tr>
<td>avERht</td>
<td>0.0599</td>
<td>0.4568</td>
<td>0.2497</td>
<td>0.52</td>
</tr>
<tr>
<td>%ER</td>
<td>0.0789</td>
<td>0.4223</td>
<td>0.2278</td>
<td>0.49</td>
</tr>
<tr>
<td>%VEG</td>
<td>0.3305</td>
<td>-0.2087</td>
<td>-0.0276</td>
<td>0.39</td>
</tr>
<tr>
<td>%LIT</td>
<td>-0.3053</td>
<td>0.1226</td>
<td>0.1145</td>
<td>0.35</td>
</tr>
<tr>
<td>%LOG</td>
<td>-0.3144</td>
<td>0.0402</td>
<td>-0.1067</td>
<td>0.33</td>
</tr>
<tr>
<td>%W</td>
<td>-0.0886</td>
<td>-0.0654</td>
<td>-0.1171</td>
<td>0.16</td>
</tr>
<tr>
<td>H1Moss</td>
<td>0.1364</td>
<td>-0.1262</td>
<td>0.4761</td>
<td>0.51</td>
</tr>
<tr>
<td>DistSWH</td>
<td>-0.3787</td>
<td>0.0101</td>
<td>0.0042</td>
<td>0.38</td>
</tr>
<tr>
<td>DistSW</td>
<td>-0.3494</td>
<td>-0.1283</td>
<td>0.1166</td>
<td>0.39</td>
</tr>
<tr>
<td>DistMF</td>
<td>0.3899</td>
<td>0.0586</td>
<td>-0.0175</td>
<td>0.39</td>
</tr>
</tbody>
</table>
Significance of Variables

• Scree plot

variables considered significant
more aggressive reduction of variables
only eliminate very weak variables
Summary

Data reduction
- notions of similarity and distance in high-D data spaces
- clustering (k-means) and how to pick optimal k
- sampling

Dimension reduction
- important vs. irrelevant dimensions
- notion of principal components and Eigenvectors
- scree plots to visualize explained variance and threshold it
- principal component analysis (PCA)
- using PCA loadings to find most important data dimensions