CSE 332
INTRODUCTION TO VISUALIZATION

DATA REDUCTION & SIMILARITY METRICS

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></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Intro, schedule, and logistics</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Applications of visual analytics, data, and basic tasks</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Data preparation and representation</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Data reduction, notion of similarity and distance</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Introduction to D3, basic vis techniques for non-spatial data</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Visual perception and cognition</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Visual design and aesthetics</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Statistics foundations</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Data mining techniques: clusters, text, patterns, classifiers</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Data mining techniques: clusters, text, patterns, classifiers</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>High-dimensional data, dimensionality reduction</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Computer graphics and volume rendering</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Techniques to visualize spatial (3D) data</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Non-photorealistic rendering</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Midterm</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Principles of interaction</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Visual analytics and the visual sense making process</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Correlation and causal modeling</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Big data: data reduction, summarization</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Visualization of graphs and hierarchies</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Visualization of text data</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Visualization of time-varying and time-series data</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Memorable visualizations, visual embellishments</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Evaluation and user studies</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Narrative visualization and storytelling</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Data journalism</td>
<td></td>
</tr>
</tbody>
</table>
What can we do to see through the mess of lines?
### How would you estimate the missing value?

<table>
<thead>
<tr>
<th>Age</th>
<th>Income</th>
<th>Team</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>24,200</td>
<td>Mets</td>
<td>M</td>
</tr>
<tr>
<td>39</td>
<td>50,245</td>
<td>Yankees</td>
<td>F</td>
</tr>
<tr>
<td>45</td>
<td>45,390</td>
<td>Yankees</td>
<td>F</td>
</tr>
<tr>
<td>22</td>
<td>32,300</td>
<td>Mets</td>
<td>M</td>
</tr>
<tr>
<td>52</td>
<td></td>
<td>Yankees</td>
<td>F</td>
</tr>
<tr>
<td>27</td>
<td>28,300</td>
<td>Mets</td>
<td>F</td>
</tr>
<tr>
<td>48</td>
<td>53,100</td>
<td>Yankees</td>
<td>M</td>
</tr>
</tbody>
</table>
Today’s Theme

Data Reduction
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
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
Good candidates are *redundant* data

- how many cans of ravioli will you buy?
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
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

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

Euclidean distance

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

$$r = r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

mean across all variable values for data items $x, y$

e.g. the “average looking” pair of pants or shoes
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

$$green = blue + 0.1$$
\[ J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}. \]

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
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.
**Objective – Minimize Squared Error**

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 Clustering Diagram](image)
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

**Better: Adaptive Sampling**

**Heights of Black Cherry Trees**

```
sampling rate ~ bin height
```

```
sampling rate ~ cluster size
```