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>1</td>
<td>Intro, schedule, and logistics</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Basic tasks and data types</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Data sources and what to look out for</td>
<td>Project 1 out</td>
</tr>
<tr>
<td>4</td>
<td>Data preparation and representation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Data reduction, notion of similarity and distance</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Dimension reduction</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Introduction to plotly and plotly Dash</td>
<td>Project 2 out</td>
</tr>
<tr>
<td>8</td>
<td>Visual perception and cognition</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Visual design and aesthetic</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Visual analytics tasks</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Cluster analysis</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>High-dimensional data, dimensionality reduction</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Visualization of spatial data: volume visualization intro</td>
<td>Project 3 out</td>
</tr>
<tr>
<td>14</td>
<td>Introduction to GPU programming</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Visualization of spatial data: raycasting, transfer functions</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Illumination and isosurface rendering</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Midterm</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Scientific visualization</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Non-photorealistic and illustrative rendering</td>
<td>Project 4 out</td>
</tr>
<tr>
<td>20</td>
<td>Midterm discussion</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Principles of interaction</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Visual analytics and the visual sense making process</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Visualization of graphs and hierarchies</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Visualization of time-varying and streaming data</td>
<td>Project 5 out</td>
</tr>
<tr>
<td>25</td>
<td>Maps</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Memorable visualizations, visual embellishments</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Evaluation and user studies</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Narrative visualization, storytelling, data journalism, XAI</td>
<td></td>
</tr>
</tbody>
</table>
Recall: The Rectangular Dataset

The variables → the attributes or properties we measured

The data items → the samples (observations) we obtained from the population of all instances
Each data item is an N-dimensional vector (N variables)
- recall 2D and 3D vectors in 2D and 3D space, respectively

Now we have N-D attribute space
- now the data axes extend into more than 3 orthogonal directions
- hard to imagine?
- that’s why need good visualization methods (will see some soon...)
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
Data Vectors

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**

$$\text{dist}(a, b) = \|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$
Cosine Similarity

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

how is this related to correlation?
What is *correlation*?

- Correlation is a statistical measure that indicates the extent to which two or more variables fluctuate together.
- A **positive correlation** indicates the extent to which those variables increase or decrease in parallel.
- A **negative correlation** indicates the extent to which one variable increases as the other decreases.
Pearson’s Correlation = correlation similarity

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

how is this related to correlation?

mean across all 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

\[ \text{green} = \text{blue} + 0.1 \]
Can you find a pair of correlated and uncorrelated data items and explain why they are (un)correlated?

The data items → the samples (observations) we obtained from the population of all instances.

The variables → the attributes or properties we measured.

### The Rectangular Dataset

<table>
<thead>
<tr>
<th>Name</th>
<th>Country</th>
<th>Miles Per Gallon</th>
<th>Acceleration</th>
<th>Horsepower</th>
<th>weight</th>
<th>cylinders</th>
<th>year</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volkswagen Rabbit DI</td>
<td>Germany</td>
<td>43.1</td>
<td>21.5</td>
<td>48</td>
<td>1985</td>
<td>4</td>
<td>78</td>
<td>2400</td>
</tr>
<tr>
<td>Ford Fiesta</td>
<td>Germany</td>
<td>36.1</td>
<td>14.4</td>
<td>66</td>
<td>1800</td>
<td>4</td>
<td>78</td>
<td>1900</td>
</tr>
<tr>
<td>Mazda GLC Deluxe</td>
<td>Japan</td>
<td>32.8</td>
<td>19.4</td>
<td>52</td>
<td>1985</td>
<td>4</td>
<td>78</td>
<td>2200</td>
</tr>
<tr>
<td>Datsun B210 GX</td>
<td>Japan</td>
<td>39.4</td>
<td>18.6</td>
<td>70</td>
<td>2070</td>
<td>4</td>
<td>78</td>
<td>2725</td>
</tr>
<tr>
<td>Honda Civic CVCC</td>
<td>Japan</td>
<td>36.1</td>
<td>16.4</td>
<td>60</td>
<td>1800</td>
<td>4</td>
<td>78</td>
<td>2250</td>
</tr>
<tr>
<td>Oldsmobile Cutlass</td>
<td>USA</td>
<td>19.9</td>
<td>15.5</td>
<td>110</td>
<td>3365</td>
<td>8</td>
<td>78</td>
<td>3300</td>
</tr>
<tr>
<td>Dodge Diplomat</td>
<td>USA</td>
<td>19.4</td>
<td>13.2</td>
<td>140</td>
<td>3735</td>
<td>8</td>
<td>78</td>
<td>3125</td>
</tr>
<tr>
<td>Mercury Monarch</td>
<td>USA</td>
<td>20.2</td>
<td>12.8</td>
<td>139</td>
<td>3570</td>
<td>8</td>
<td>78</td>
<td>2850</td>
</tr>
<tr>
<td>Pontiac Phoenix</td>
<td>USA</td>
<td>19.2</td>
<td>19.2</td>
<td>105</td>
<td>3535</td>
<td>6</td>
<td>78</td>
<td>2800</td>
</tr>
<tr>
<td>Chevrolet Malibu</td>
<td>USA</td>
<td>20.5</td>
<td>18.2</td>
<td>95</td>
<td>3155</td>
<td>6</td>
<td>78</td>
<td>3275</td>
</tr>
<tr>
<td>Ford Fairmont A</td>
<td>USA</td>
<td>20.2</td>
<td>15.8</td>
<td>85</td>
<td>2965</td>
<td>6</td>
<td>78</td>
<td>2375</td>
</tr>
<tr>
<td>Ford Fairmont M</td>
<td>USA</td>
<td>25.1</td>
<td>15.4</td>
<td>88</td>
<td>2720</td>
<td>6</td>
<td>78</td>
<td>2275</td>
</tr>
<tr>
<td>Plymouth Volare</td>
<td>USA</td>
<td>20.5</td>
<td>17.2</td>
<td>100</td>
<td>3430</td>
<td>4</td>
<td>78</td>
<td>2700</td>
</tr>
<tr>
<td>AMC Concord</td>
<td>USA</td>
<td>19.4</td>
<td>17.2</td>
<td>90</td>
<td>3210</td>
<td>6</td>
<td>78</td>
<td>2300</td>
</tr>
<tr>
<td>Buick Century</td>
<td>USA</td>
<td>20.6</td>
<td>15.8</td>
<td>105</td>
<td>3380</td>
<td>6</td>
<td>78</td>
<td>3300</td>
</tr>
<tr>
<td>Mercury Zephyr</td>
<td>USA</td>
<td>20.8</td>
<td>16.7</td>
<td>85</td>
<td>3070</td>
<td>6</td>
<td>78</td>
<td>2425</td>
</tr>
<tr>
<td>Dodge Aspen</td>
<td>USA</td>
<td>18.6</td>
<td>18.7</td>
<td>110</td>
<td>3620</td>
<td>8</td>
<td>78</td>
<td>2700</td>
</tr>
<tr>
<td>AMC Concord D1</td>
<td>USA</td>
<td>18.1</td>
<td>15.1</td>
<td>120</td>
<td>3410</td>
<td>6</td>
<td>78</td>
<td>2425</td>
</tr>
<tr>
<td>Chevrolet MonteCarlo</td>
<td>USA</td>
<td>19.2</td>
<td>13.2</td>
<td>145</td>
<td>3425</td>
<td>8</td>
<td>78</td>
<td>3900</td>
</tr>
<tr>
<td>Buick RegalTurbo</td>
<td>USA</td>
<td>17.7</td>
<td>13.4</td>
<td>165</td>
<td>3445</td>
<td>8</td>
<td>78</td>
<td>4400</td>
</tr>
<tr>
<td>Ford Futura</td>
<td>Germany</td>
<td>18.1</td>
<td>11.2</td>
<td>139</td>
<td>3205</td>
<td>8</td>
<td>78</td>
<td>2525</td>
</tr>
<tr>
<td>Dodge Magnum XE</td>
<td>USA</td>
<td>17.5</td>
<td>13.7</td>
<td>140</td>
<td>4080</td>
<td>8</td>
<td>78</td>
<td>3000</td>
</tr>
<tr>
<td>Chevrolet Chevette</td>
<td>USA</td>
<td>30</td>
<td>16.5</td>
<td>68</td>
<td>2155</td>
<td>4</td>
<td>78</td>
<td>2100</td>
</tr>
</tbody>
</table>
What’s the Jaccard similarity of the two baskets A and B?

3/13 = 0.23
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

The last slide and the next 8 slides contain figures courtesy of Eamonn Keogh, UC Riverside
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 $K$?

$k=1, \text{MSE}=873.0$

$k=2, \text{MSE}=173.1$

$k=3, \text{MSE}=133.6$
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 ~ bin height

sampling rate ~ cluster size
Good candidates for elimination are redundant data

- how many cans of ravioli will you buy?
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
- only keep the cluster centroids
- store size of clusters along to keep importance
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
- can get really expensive when the dimensionality $d$ is large because each pairwise distance has $O(d)$
Learned about

- distance metrics to evaluate similarity among data points
  - correlation, cosine, Euclidian, Jacquard, Manhattan distance
- used it for clustering that can identify groups in data
- these groups can be used for unbiased data reduction and augmentation
- the k-means algorithm as a simple yet effective clustering scheme
- the elbow method to pick a good \( k = \) number of clusters
- advanced sampling methods: well-scattered points, Reservoir sampling for streaming data