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>Intro continued</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Applications of visual analytics, data, and basic tasks</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Data preparation and reduction</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>Data reduction and similarity metrics</strong></td>
<td>Project 1 out</td>
</tr>
<tr>
<td>6</td>
<td>Dimension reduction</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Introduction to D3</td>
<td>Project 2 out</td>
</tr>
<tr>
<td>8</td>
<td>Bias in visualization</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Perception and cognition</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Visual design and aesthetics</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Cluster and pattern analysis</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>High-Dimensional data visualization: linear methods</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>High-D data vis.: non-linear methods, categorical data</td>
<td>Project 3 out</td>
</tr>
<tr>
<td>14</td>
<td>Principles of interaction</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Visual analytics and the visual sense making process</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>VA design and evaluation</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Visualization of graphs and hierarchies</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Visualization of time-varying and time-series data</td>
<td>Project 4 out</td>
</tr>
<tr>
<td>19</td>
<td>Midterm</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Maps and geo-vis</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Computer graphics and volume rendering</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Techniques to visualize spatial (3D) data</td>
<td>Project 4 halfway report due</td>
</tr>
<tr>
<td>23</td>
<td>Scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Non-photorealistic rendering</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Memorable visualizations, visual embellishments</td>
<td>Project 5 out</td>
</tr>
<tr>
<td>27</td>
<td>Infographics design</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Projects Hall of Fame demos</td>
<td></td>
</tr>
</tbody>
</table>
Recall: The Rectangular Dataset

- **One data item**
- **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...)
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?
Sampling Principles

Keep a representative number of samples:

- pick one of each
- or maybe a few more depending on importance
How to Pick?

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
Back To Similarity Functions

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
Pants:
<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

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

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.

[Diagram showing positive, negative, and no correlation]

**spatial proximity representation**
Cosine 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?

Pearson’s Correlation = correlation similarity

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

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

Correlations: $(5 \times 5 - 5)/2 = 10$ pairs

- positively correlated: shoes 1 and 2, shoes 4 and 5
- negatively correlated: shoes 1 and 4, 1 and 5, 2 and 4, 2 and 5
- fairly uncorrelated: shoe 3 with all others 1, 2, 4, 5
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

\[ \text{green} = \text{blue} + 0.1 \]
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

What is Clustering?

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
The K-Means Clustering Algorithm

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, 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
Good candidates for elimination are *redundant* data

- how many cans of ravioli will you buy?
Eliminate redundant attributes

- eliminate correlated attributes
  - km vs. miles
  - \( a + b + c = d \) → 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
Sampling of Well-Scattered Points

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 $k = \text{number of clusters}$
- advanced sampling methods: well-scattered points, Reservoir sampling for streaming data