CSE 564
Visualization & Visual Analytics

Cluster Analysis

Klaus Mueller

Computer Science Department
Stony Brook University and SUNY Korea
<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>Introduction to D3, basic vis techniques for non-spatial data</td>
<td>Project #1 out</td>
</tr>
<tr>
<td>4</td>
<td>Data assimilation and preparation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Bias in visualization</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Data reduction and dimension reduction</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>Data mining techniques: clusters, text, patterns, classifiers</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Computer graphics and volume rendering</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Techniques to visualize spatial (3D) data</td>
<td>Project #2 due</td>
</tr>
<tr>
<td>13</td>
<td>Scientific and medical visualization</td>
<td>Project #3 out</td>
</tr>
<tr>
<td>14</td>
<td>Scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Midterm #1</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>High-dimensional data, dimensionality reduction</td>
<td>Project #3 due</td>
</tr>
<tr>
<td>17</td>
<td>Big data: data reduction, summarization</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Correlation and causal modeling</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Principles of interaction</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>
When to Use Cluster Analysis

Data summarization
- data reduction
- cluster centers, shapes, and statistics

Customer segmentation
- collaborative filtering

Social network analysis
- find similar groups of friends (communities)

Precursor to other analyses
- use as a preprocessing step for classification and outlier detection
- use it for sampling and data reduction
With 1,000s of attributes (dimensions) which ones are relevant and which one are not?

Histogram of pairwise distances in N-D space

avoid

keep

(a) Uniform Data
(b) Clustered data
(c) Distance distribution (uniform)
(d) Distance distribution (clustered)
How to measure attribute “worthiness”

- use entropy

Entropy

- originates in thermodynamics
- measures lack of order or predictability

Entropy in statistics and information theory

- has a value of 1 for uniform distributions (not predictable)
- knowing the value has a lot of information (high surprise)
- has a value of 0 for a constant signal (fully predicable)
- knowing the value has zero information (low surprise)
Assume $m$ bins, $1 \leq i \leq m$: 

$$E = - \sum_{i=1}^{m} [p_i \log(p_i) + (1 - p_i) \log(1 - p_i)].$$

**Algorithm:**
- start with all attributes and compute distance entropy
- greedily eliminate attributes that reduce the entropy the most
- stop when entropy no longer reduces or even increases

**Binary source** (e.g. coin)
Two options for building the dendrogram on the left

- top down (divisive)
- bottom up (agglomerative)
Algorithm AgglomerativeMerge(Data: \( D \))
begin
    Initialize \( n \times n \) distance matrix \( M \) using \( D \);
    repeat
        Pick closest pair of clusters \( i \) and \( j \) using \( M \);
        Merge clusters \( i \) and \( j \);
        Delete rows/columns \( i \) and \( j \) from \( M \) and create a new row and column for newly merged cluster;
        Update the entries of new row and column of \( M \);
    until termination criterion;
    return current merged cluster set;
end

How to merge?
**Merge Criteria**

**Single (best-case) linkage**
- distance = minimum distance between all $m_i \cdot m_j$ pairs of objects
- joins the closest pair

**Complete (worst-case) linkage**
- distance = maximum distance between all $m_i \cdot m_j$ pairs of objects
- joins the pair furthest apart

**Group-average linkage**
- distance = average distance between all object pairs in the groups

**Other methods:**
- closest centroid, variance-minimization, Ward’s method
Centroid-based methods tend to merge large clusters

Single linkage method can merge chains of closely related points to discover clusters of arbitrary shape

- but can also (inappropriately) merge two unrelated clusters, when the chaining is caused by noisy points between two clusters

(a) Good case with no noise

(b) Bad case with noise
Complete (worst-case) linkage method tends to create spherical clusters with similar diameter

- will break up the larger odd-shaped clusters into smaller spheres
- also gives too much importance to data points at the noisy fringes of a cluster
The group average, variance, and Ward’s methods are more robust to noise due to the use of multiple linkages in the distance computation.

Hierarchical methods are sensitive to a small number of mistakes made during the merging process:

- can be due to noise
- no way to undo these mistakes
Highly-cited density-based hierarchical clustering algorithm (Ester et al. 1996)

- Clusters are defined as density-connected sets
- Epsilon-distance neighbor criterion ($Eps$)
  \[ N_{Eps}(p) = \{ q \in D \mid dist(p,q) \leq Eps \} \]
- Minimum point cluster membership and core point ($MinPts$)
  \[ |N_{Eps}(q)| \geq MinPts \]
- Notions of density-connected & density-reachable (direct, indirect)
- A point $p$ is directly density-reachable from a point $q$ wrt. $Eps$, $MinPts$ if
  \[ p \in N_{Eps}(q) \text{ and } |N_{Eps}(q)| \geq MinPts \text{ (core point condition)} \]
DBSCAN

- p: border point
- q: core point

(a) p density-reachable from q
- q not density-reachable from p

(b) p directly density-reachable from q
- q not directly density-reachable from p

(c) p density-reachable from q
- q not density-reachable from p

(d) p and q density-connected to each other by o
Probabilistic Extension to K-Means

First a comparison:

Different cluster analysis results on "mouse" data set:
Original Data  k-Means Clustering  EM Clustering
The distance between a point $X$ and a distribution $D$

- measures how many standard deviations $X$ is away from the mean $\mu$ of $D$
- $S$ is the covariance matrix of the distribution $D$
- the Mahalanobis distance $D_M$ of a point $x$ to a cluster center $\mu$ is

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}.$$ 

- $x$ and $\mu$ are N-dimensional vectors
- $S$ is the $N \times N$ covariance matrix
- the outcome $D_M(x)$ is a single-dimensional number
Is a better match for point distributions

- overlapping clusters are now possible
- better match with real world?
- Gaussian mixtures

Need a probabilistic algorithm

- Expectation-Maximization
EM Algorithm (Mixture Model)

- Initialize K cluster centers
- Iterate between two steps
  - **Expectation step**: assign points to \( m \) clusters/classes
    \[
    P(d_i \in c_k) = w_k \Pr(d_i | c_k) \big/ \sum_j w_j \Pr(d_i | c_j)
    \]
    \[
    w_k = \frac{\sum_{i} \Pr(d_i \in c_k)}{N} = \text{probability of class } c_k
    \]
  - **Maximation step**: estimate model parameters
    do similar also for covariance matrix \( S \)
    \[
    \mu_k = \frac{1}{m} \sum_{i=1}^{m} \frac{d_i P(d_i \in c_k)}{\sum_k P(d_i \in c_j)}
    \]

probability that data point \( d_i \) is in class \( c_j \)
\( (= \text{Mahalanobis distance of } d_i \text{ to } c_j) \)
Iteration 1

The cluster means are randomly assigned
Iteration 2

Mean Likelihood = -12.501313295068318

0.233925249561192798

0.4801878834773863

0.231996096904228
Iteration 5

Mean Likelihood = -11.879896828880106
Iteration 25

Mean Likelihood = -11.13452288716779

0.21511520737329874

0.591678275692965

0.3004821586057919
t-distributed stochastic neighbor embedding
T-SNE Distance Metric

Uses the following density-based (probabilistic) distance metric

$$P_{ji} = \frac{\exp\left(-\frac{|x_i - x_j|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{|x_i - x_k|^2}{2\sigma_i^2}\right)}$$

Measures how (relatively) close $x_j$ is from $x_i$, considering a Gaussian distribution around $x_i$ with a given variance $\sigma_i^2$.

- this variance is different for every point
- $t$ is chosen such that points in dense areas are given a smaller variance than points in sparse areas
Use a symmetrized version of the conditional similarity:

\[ p_{ij} = \frac{p_{ji} + p_{ij}}{2N} \]

Similarity (distance) metric for mapped points:

\[ q_{ij} = \frac{f(|x_i - x_j|)}{\sum_{k \neq i} f(|x_i - x_k|)} \quad \text{with} \quad f(z) = \frac{1}{1+z^2} \]

This uses the t-student distribution with one degree of freedom, or Cauchy distribution, instead of a Gaussian distribution.
Can use mass-spring system enforcing minimum of $|p_{ij} - q_{ij}|$

The classic *handwritten digits* datasets. It contains 1,797 images with $8 \times 8 = 64$ pixels each.
See [this webpage](#)
Rectangular data set with a temporal component

- assume you have these data for each year
- how to handle that, you might ask?
Assume for now we have:
- two attributes (burglary, theft)
- both observed over time

Can visualize:
- but each point is a time series!
Similarity Measures

Needed it for clustering
  - recall Euclidean, correlation, cosine distances

<table>
<thead>
<tr>
<th>State</th>
<th>Burglary</th>
<th>Larceny-theft</th>
<th>Motor Vehicle Theft</th>
<th>Arson2</th>
<th>Violent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALIFORNIA</td>
<td>2.616</td>
<td>10.49</td>
<td>6,298</td>
<td>3,344</td>
<td>71</td>
</tr>
<tr>
<td>MICHIGAN</td>
<td>1.049</td>
<td>1,049</td>
<td>979</td>
<td>154</td>
<td>72</td>
</tr>
<tr>
<td>TENNESSEE</td>
<td>5,604</td>
<td>8,431</td>
<td>12,141</td>
<td>3,828</td>
<td>290</td>
</tr>
<tr>
<td>MISSOURI</td>
<td>1,952</td>
<td>6,432</td>
<td>1,376</td>
<td>177</td>
<td>50</td>
</tr>
<tr>
<td>MARYLAND</td>
<td>1,372</td>
<td>8,761</td>
<td>1,376</td>
<td>137</td>
<td>50</td>
</tr>
<tr>
<td>ALABAMA</td>
<td>3,964</td>
<td>4,375</td>
<td>551</td>
<td>458</td>
<td>21</td>
</tr>
<tr>
<td>OHIO</td>
<td>1.759</td>
<td>2,424</td>
<td>196</td>
<td>148</td>
<td>18</td>
</tr>
<tr>
<td>ILLINOIS</td>
<td>875</td>
<td>1,960</td>
<td>1,376</td>
<td>177</td>
<td>50</td>
</tr>
<tr>
<td>ARKANSAS</td>
<td>1,852</td>
<td>5,012</td>
<td>1,376</td>
<td>137</td>
<td>50</td>
</tr>
<tr>
<td>CALIFORNIA</td>
<td>2,616</td>
<td>5,604</td>
<td>12,141</td>
<td>3,828</td>
<td>290</td>
</tr>
<tr>
<td>WISCONSIN</td>
<td>2,619</td>
<td>7,622</td>
<td>1,719</td>
<td>120</td>
<td>34</td>
</tr>
</tbody>
</table>

Similarity of two states

Similarity of two crimes
in a given state over time

Similarity of two crimes
for a given crime over time

Two time series

Time

Euclidean
CLUSTERING

What can be clustered with these measures?

- crimes (averaged over time)
- states (averaged over time)
- crimes in a given state (taking time series into account)
- states for a given crime (taking time series into account)

Can we get more inclusive?

- cluster crimes but including the time series characteristics
- cluster states but including the time series characteristics

Capture more information about their time series when you compare two data points

- compute the similarity of two crimes by summing the times-series similarities for each state
- compute the similarity of two states by summing the times-series similarities for each crime
Time-series aware similarity (distance) $S_{tsa}$ for a pair of states

for a given pair of states $i, j$

for each crime $c$

compute the time series similarity $\rightarrow \text{sim}_t(c)$

sum all $\text{sim}_t(c)$ $\rightarrow S_{tsa}(i,j)$

$S_{tsa}$ for a pair of crimes

for each pair of crimes $i, j$

for each state $s$

compute the time series similarity $\rightarrow \text{sim}_t(s)$

sum all $\text{sim}_t(s)$ $\rightarrow S_{tsa}(i,j)$

If the time series are aligned for all states then the $S_{tsa}$ will be high and the two crimes have very similar time behaviors nationwide.

$\text{similarity}$ could be some measure of correlation of the two time series vectors
The time series might not be aligned
- one crime might cause another
- can apply dynamic time warping (see next)

You may (also) have a geospatial component in your data
- can use them as a regular attribute (encoded by an ID)
- can you make them more continuous and linearly ordered?
- use a space filling curve (see next)

You may want to just keep time instances as separate entities
- that will work too
- then you might discover clusters that are sensitive to time
- or you can see how the years relate to another along a trajectory
- as a general rule, when you visualize multivariate data first decide what you will put into the rectangular data matrix (samples, attributes)
Standard pairwise distance

$$\text{Dist}(X, Y) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Shortcomings:
- designed for time series of equal length
- cannot address distortions on the temporal (contextual) attributes
Can better accommodate local mismatches

Three constraints
- no skipping of beginning or ends of either sequence
- continuity – no jumps
- monotonicity – can’t go back in time
DTW – Find The Minimum Cost Path

Euclidian

DTW
DTW – Find The Minimum Cost Path

DTW

Compute using dynamic programming

Available in python
Linearizing Map Locations

Convert a geographical map into a grid map

Linearize using a space-filling curve (Hilbert curve)
Plot two variables

But what if you have more than two variables?
Problem:
- Multivariate relationships are scattered across the tiles
Biplots

Plots data points and dimension axes into a single visualization

- uses first two PCA vectors as the basis to project into
- find plot coordinates \([x] [y]\)
  for data points: \([\text{PCA}_1 \cdot \text{data vector}] [\text{PCA}_2 \cdot \text{data vector}]\)
  for dimension axes: \([\text{PCA}_1[\text{dimension}]] [\text{PCA}_2[\text{dimension}]]\)
Biplots in Practice

See data distributions into the context of their attributes
Biplots in Practice

See data points into the context of their attributes
Biplots – A Word of Caution

Do be aware that the projections may not be fully accurate

- you are projecting N-D into 2D by a linear transformation
- if there are more than 2 significant PCA vectors then some variability will be lost and won’t be visualized
- remote data points might project into nearby plot locations suggesting false relationships
- leads to projection ambiguity
MDS preserves similarity relationships, prevents ambiguity
  - scattered points in high-dimensions (N-D)
  - adjacency matrices

Maps the distances between observations from N-D into low-D (say 2D)
  - attempts to ensure that differences between pairs of points in this reduced space match as closely as possible

The input to MDS is a distance (similarity) matrix
  - actually, you use the dissimilarity matrix because you want similar points mapped closely
  - dissimilar point pairs will have greater values and map farther apart
The Dissimilarity Matrix

![Graph with points x1, x2, x3, x4 on a Cartesian plane.]

**Data Matrix**

<table>
<thead>
<tr>
<th>point</th>
<th>attribute1</th>
<th>attribute2</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>x2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>x3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>x4</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**Dissimilarity Matrix (with Euclidean Distance)**

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>3.61</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>2.24</td>
<td>5.1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td>4.24</td>
<td>1</td>
<td>5.39</td>
<td>0</td>
</tr>
</tbody>
</table>
MDS turns a distance matrix into a network or point cloud

- correlation, cosine, Euclidean, and so on

Suppose you know a matrix of distances among cities

<table>
<thead>
<tr>
<th></th>
<th>Chicago</th>
<th>Raleigh</th>
<th>Boston</th>
<th>Seattle</th>
<th>S.F.</th>
<th>Austin</th>
<th>Orlando</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chicago</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Raleigh</td>
<td>641</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boston</td>
<td>851</td>
<td>608</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seattle</td>
<td>1733</td>
<td>2363</td>
<td>2488</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.F.</td>
<td>1855</td>
<td>2406</td>
<td>2696</td>
<td>684</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Austin</td>
<td>972</td>
<td>1167</td>
<td>1691</td>
<td>1764</td>
<td>1495</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Orlando</td>
<td>994</td>
<td>520</td>
<td>1105</td>
<td>2565</td>
<td>2458</td>
<td>1015</td>
<td>0</td>
</tr>
</tbody>
</table>
RESULT OF MDS
MDS Algorithm

- **Task:**
  - Find that configuration of image points whose pairwise distances are most similar to the original inter-point distances !!!

- **Formally:**
  - Define: \( D_{ij} = \| x_i - x_j \|_D \) \quad \( d_{ij} = \| y_i - y_j \|_D \)
  - Claim: \( D_{ij} \equiv d_{ij} \quad \forall i, j \in [1, n] \)

- **In general:** an exact solution is not possible !!!

- **Inter Point distances \( \rightarrow \) invariance features**
MDS Algorithm

Strategy (of metric MDS):

- iterative procedure to find a good configuration of image points
  
  1) Initialization
     - Begin with some (arbitrary) initial configuration
  
  2) Alter the image points and try to find a configuration of points that minimizes the following sum-of-squares error function:
MDS Algorithm

Strategy (of metric MDS):

- iterative procedure to find a good configuration of image points
  
- 1) Initialization
     → Begin with some (arbitrary) initial configuration
  
- 2) Alter the image points and try to find a configuration of points that minimizes the following sum-of-squares error function:

\[
E = \sum_{i<j}^{N} (D_{ij} - d_{ij})^2
\]
Spring-like system

- insert springs within each node
- the length of the spring encodes the desired node distance
- start at an initial configuration
- iteratively move nodes until an energy minimum is reached
**Force-Directed Algorithm**

Spring-like system

- insert springs within each node
- the length of the spring encodes the desired node distance
- start at an initial configuration
- iteratively move nodes until an energy minimum is reached

Vertex layout by correlations.
Distance (similarity) metric

- Euclidean distance (best for data)
- Cosine distance (best for data)
- $|1$-correlation$|$ distance (best for attributes)
- Use $1$-correlation to move correlated attribute points closer
- Use $||$ if you do not care about positive or negative correlations
MDS Examples
Let’s look at application in text processing

Assume you are given a large corpus of documents and you wish to get an overview about what they contain

What can you do?
The same as PCA when the mean of each attribute is zero

SVD does not subtract the mean

- appropriate if values close to zero should not be influential
- PCA puts them at in the extreme negative side

SVD often used for text analysis

- values close to zero are frequent and should not affect the analysis
Decomposes $C$ into the matrix:

$$Q_k \Sigma_k P_k^T$$

$q_i$ and $p_i$ are two column vectors with significance $\sigma_i$

$$Q_k \Sigma_k P_k^T = \sum_{i=1}^{k} q_i \sigma_i p_i^T = \sum_{i=1}^{k} \sigma_i (q_i p_i^T)$$

Example: in a user-item ratings matrix we wish to determine:

- a reduced representation of the users
- a reduced representation of the items
- SVD has the basis vectors for both of these reductions
Find the matrices $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ such that:

$$\mathbf{C} = \mathbf{U} \mathbf{D} \mathbf{V}^T$$

$\mathbf{U}$ are the Eigenvectors of $\mathbf{C} \mathbf{C}^T$
$\mathbf{V}$ are the Eigenvectors of $\mathbf{C}^T \mathbf{C}$
$\mathbf{D}$ a diagonal matrix of $\sqrt{\lambda_k}$ where $\lambda_k$ are Eigenvalues of $\mathbf{C} \mathbf{C}^T$
$k = \text{Rank}(\mathbf{C}) < \text{Min}(r-1,c-1)$
Create an occurrence matrix (term-document matrix)

- words (terms \( t \)) are the rows
- paragraphs (documents \( d \)) are the columns
- uses the term frequency–inverse document frequency (tf-idf) metric
  \[ tf(t,d) = \text{simplest form is frequency of } t \text{ in } d = f(t,d) \]

<table>
<thead>
<tr>
<th>Index Words</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>T9</th>
</tr>
</thead>
<tbody>
<tr>
<td>book</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dads</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dummies</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>estate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>guide</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>investing</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>market</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>real</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>rich</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>stock</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Create an occurrence matrix (term-document matrix)

- words (terms $t$) are the rows
- paragraphs (documents $d$) are the columns
- uses the term frequency–inverse document frequency (tf-idf) metric
- $tf(t,d) = \text{simplest form is frequency of } t \text{ in } d = f(t,d)$

- $idf(t,d) = \text{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$

- $N = \text{number of docs } = |D|$, $D$ is the corpus of documents
- $idf$ is a measure of term rareness, it’s 0 when term occurs in all of $D$
- important terms get a higher tf-idf

Use SVD to reduce the number of rows

- preserves similarity of columns
Co-Occurrence TF-IDT matrix
$U = \text{term-concept matrix}$

concept = latent (hidden) topic

sort and keep the $k$
most significant rows/columns

$V = \text{concept-document matrix}$
How many concepts to use when approximating the matrix?

- if too few, important patterns are left out
- if too many, noise caused by random word choices will creep in
- can use the elbow method in the scree plot

Throw out the 1\textsuperscript{st} dimension in U and V

- in U it is correlated with document length
- in V it correlates with the number of times a term was mentioned

Now we have a k-D concept space shared by both terms and documents
Visualizing The Concept Space

Project the k-D concept space into 2D and visualize as a map

- can cluster the map
- the cluster of documents are then labeled by the terms
- provides map semantics
LSA assumptions and disadvantages:

- Assumes a Gaussian distribution and Frobenius norm.
  - This may not fit all problems.

- Cannot handle polysemy effectively.
  - Need LDA (Latent Dirichlet Allocation) for this.

- Depends heavily on SVD.
  - Computationally intensive.
  - Hard to update as new documents appear.
  - But faster algorithms have emerged recently.
You will need to use correspondence analysis (CA)
  - CA is PCA for categorical variables
  - related to factor analysis

Makes use of the $\chi^2$ test
  - what’s $\chi^2$?
Chi-square Test (Nominal Data)

• A *chi-square test* is used to investigate relationships
• Relationships between categorical, or nominal-scale, variables representing attributes of people, interaction techniques, systems, etc.
• Data organized in a *contingency table* – cross tabulation containing counts (frequency data) for number of observations in each category
• A chi-square test compares the *observed values* against *expected values*
• Expected values assume “no difference”
• Research question:
  – *Do males and females differ in their method of scrolling on desktop systems?* (next slide)
Chi-square – Example #1

<table>
<thead>
<tr>
<th>Gender</th>
<th>Scrolling Method</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MW</td>
<td>CD</td>
</tr>
<tr>
<td>Male</td>
<td>28</td>
<td>15</td>
</tr>
<tr>
<td>Female</td>
<td>21</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>49</td>
<td>24</td>
</tr>
</tbody>
</table>

MW = mouse wheel  
CD = clicking, dragging  
KB = keyboard
Chi-square – Example #1

56.0 - 49.0 / 101 = 27.2

<table>
<thead>
<tr>
<th>Gender</th>
<th>Scrolling Method</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MW</td>
<td>CD</td>
</tr>
<tr>
<td>Male</td>
<td>27.2</td>
<td>13.3</td>
</tr>
<tr>
<td>Female</td>
<td>21.8</td>
<td>10.7</td>
</tr>
<tr>
<td>Total</td>
<td>49.0</td>
<td>24.0</td>
</tr>
</tbody>
</table>

(Expected-Observed)^2 / Expected = (28 - 27.2)^2 / 27.2

<table>
<thead>
<tr>
<th>Gender</th>
<th>Scrolling Method</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MW</td>
<td>CD</td>
</tr>
<tr>
<td>Male</td>
<td>0.025</td>
<td>0.215</td>
</tr>
<tr>
<td>Female</td>
<td>0.032</td>
<td>0.268</td>
</tr>
<tr>
<td>Total</td>
<td>0.057</td>
<td>0.483</td>
</tr>
</tbody>
</table>

Significant if it exceeds critical value (next slide)

\( \chi^2 = 1.462 \)
Chi-square Critical Values

- Decide in advance on *alpha* (typically .05)
- Degrees of freedom
  - $df = (r - 1)(c - 1) = (2 - 1)(3 - 1) = 2$
  - $r = \text{number of rows}, \ c = \text{number of columns}$

<table>
<thead>
<tr>
<th>Significance Threshold ($\alpha$)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1</td>
<td>2.71</td>
<td>4.61</td>
<td>6.25</td>
<td>7.78</td>
<td>9.24</td>
<td>10.65</td>
<td>12.02</td>
<td>13.36</td>
</tr>
<tr>
<td>.05</td>
<td>3.84</td>
<td><strong>5.99</strong></td>
<td>7.82</td>
<td>9.49</td>
<td>11.07</td>
<td>12.59</td>
<td>14.07</td>
<td>15.51</td>
</tr>
<tr>
<td>.01</td>
<td>6.64</td>
<td>9.21</td>
<td>11.35</td>
<td>13.28</td>
<td>15.09</td>
<td>16.81</td>
<td>18.48</td>
<td>20.09</td>
</tr>
<tr>
<td>.001</td>
<td>10.83</td>
<td>13.82</td>
<td>16.27</td>
<td>18.47</td>
<td>20.52</td>
<td>22.46</td>
<td>24.32</td>
<td>26.13</td>
</tr>
</tbody>
</table>

$\chi^2 = 1.462 \ (< 5.99 : \text{not significant})$
Correspondence Analysis (CA)

Example:

<table>
<thead>
<tr>
<th>Staff Group</th>
<th>Smoking Category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1) None</td>
</tr>
<tr>
<td>(1) Senior Managers</td>
<td>4</td>
</tr>
<tr>
<td>(2) Junior Managers</td>
<td>4</td>
</tr>
<tr>
<td>(3) Senior Employees</td>
<td>25</td>
</tr>
<tr>
<td>(4) Junior Employees</td>
<td>18</td>
</tr>
<tr>
<td>(5) Secretaries</td>
<td>10</td>
</tr>
<tr>
<td>Column Totals</td>
<td>61</td>
</tr>
</tbody>
</table>

There are two high-D spaces
- 4D (column) space spanned by smoking habits – plot staff group
- 5D (row) space spanned by staff group – plot smoking habits

Are these two spaces (the rows and columns) independent?
- this occurs when the $\chi^2$ statistics of the table is insignificant
Let’s do some plotting

- compute distance matrix of the rows \( CC^T \)
- compute Eigenvector matrix \( U \) and the Eigenvalue matrix \( D \)
- sort eigenvectors by values, pick two major vectors, create 2D plot

-- senior employees most similar to secretaries
Next:

- compute distance matrix of the columns \( C^T C \)
- compute Eigenvector matrix \( V \) (gives the same Eigenvalue matrix \( D \))
- sort eigenvectors by value
- pick two major vectors
- create 2D plot of smoking categories

Following (next slide):

- combine the plots of \( U \) and \( V \)
- if the \( \chi^2 \) statistics was significant we should see some dependencies
Interpretation sample (using the $\chi^2$ frequentist mindset)

- *relatively speaking*, there are more non-smoking senior employees
Plot would now show 193 cases and 9 variables
Multiple Correspondence Analysis

Extension where there are more than 2 categorical variables

<table>
<thead>
<tr>
<th>Case No.</th>
<th>SURVIVAL</th>
<th>AGE</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NO</td>
<td>YES</td>
<td>LESST50</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>...</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>...</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>762</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>763</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>764</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Let’s call it matrix X
Compute $X'X$ to get the Burt Table

Compute Eigenvectors and Eigenvalues

- keep top two Eigenvectors/values
- visualize the attribute loadings of these two Eigenvectors into the Burt table plot (the loadings are the coordinates)
Results of a survey of car owners and car attributes

<table>
<thead>
<tr>
<th></th>
<th>American</th>
<th>European</th>
<th>Japanese</th>
<th>Large</th>
<th>Medium</th>
<th>Small</th>
<th>Family</th>
<th>Sporty</th>
<th>Work</th>
<th>1 Income</th>
<th>2 Incomes</th>
<th>Own</th>
<th>Rent</th>
<th>Married</th>
<th>Married with Kids</th>
<th>Single</th>
<th>Single with Kids</th>
<th>Female</th>
<th>Male</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>125</td>
<td>0</td>
<td>0</td>
<td>36</td>
<td>50</td>
<td>29</td>
<td>81</td>
<td>24</td>
<td>20</td>
<td>58</td>
<td>67</td>
<td>93</td>
<td>32</td>
<td>37</td>
<td>50</td>
<td>32</td>
<td>6</td>
<td>58</td>
<td>67</td>
</tr>
<tr>
<td>European</td>
<td>0</td>
<td>44</td>
<td>0</td>
<td>4</td>
<td>20</td>
<td>20</td>
<td>17</td>
<td>23</td>
<td>4</td>
<td>18</td>
<td>26</td>
<td>38</td>
<td>6</td>
<td>13</td>
<td>15</td>
<td>15</td>
<td>1</td>
<td>21</td>
<td>23</td>
</tr>
<tr>
<td>Japanese</td>
<td>0</td>
<td>0</td>
<td>165</td>
<td>2</td>
<td>61</td>
<td>102</td>
<td>76</td>
<td>59</td>
<td>30</td>
<td>74</td>
<td>91</td>
<td>111</td>
<td>54</td>
<td>51</td>
<td>44</td>
<td>62</td>
<td>8</td>
<td>70</td>
<td>92</td>
</tr>
<tr>
<td>Large</td>
<td>36</td>
<td>4</td>
<td>2</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>1</td>
<td>11</td>
<td>20</td>
<td>22</td>
<td>35</td>
<td>7</td>
<td>9</td>
<td>21</td>
<td>11</td>
<td>1</td>
<td>17</td>
<td>25</td>
</tr>
<tr>
<td>Medium</td>
<td>60</td>
<td>20</td>
<td>61</td>
<td>0</td>
<td>141</td>
<td>0</td>
<td>09</td>
<td>39</td>
<td>13</td>
<td>57</td>
<td>04</td>
<td>106</td>
<td>35</td>
<td>42</td>
<td>51</td>
<td>40</td>
<td>0</td>
<td>70</td>
<td>71</td>
</tr>
<tr>
<td>Small</td>
<td>29</td>
<td>20</td>
<td>102</td>
<td>0</td>
<td>0</td>
<td>151</td>
<td>55</td>
<td>66</td>
<td>30</td>
<td>73</td>
<td>78</td>
<td>101</td>
<td>50</td>
<td>50</td>
<td>37</td>
<td>58</td>
<td>6</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>Family</td>
<td>81</td>
<td>17</td>
<td>76</td>
<td>30</td>
<td>89</td>
<td>55</td>
<td>174</td>
<td>0</td>
<td>0</td>
<td>69</td>
<td>105</td>
<td>130</td>
<td>44</td>
<td>50</td>
<td>79</td>
<td>35</td>
<td>10</td>
<td>83</td>
<td>91</td>
</tr>
<tr>
<td>Sporty</td>
<td>24</td>
<td>23</td>
<td>59</td>
<td>1</td>
<td>39</td>
<td>66</td>
<td>0</td>
<td>106</td>
<td>0</td>
<td>55</td>
<td>51</td>
<td>71</td>
<td>35</td>
<td>35</td>
<td>12</td>
<td>57</td>
<td>2</td>
<td>44</td>
<td>62</td>
</tr>
<tr>
<td>Work</td>
<td>20</td>
<td>4</td>
<td>30</td>
<td>11</td>
<td>13</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>54</td>
<td>26</td>
<td>28</td>
<td>41</td>
<td>13</td>
<td>16</td>
<td>18</td>
<td>17</td>
<td>3</td>
<td>22</td>
<td>32</td>
</tr>
<tr>
<td>1 Income</td>
<td>58</td>
<td>18</td>
<td>74</td>
<td>20</td>
<td>57</td>
<td>73</td>
<td>69</td>
<td>53</td>
<td>26</td>
<td>150</td>
<td>0</td>
<td>80</td>
<td>70</td>
<td>10</td>
<td>27</td>
<td>99</td>
<td>14</td>
<td>47</td>
<td>103</td>
</tr>
<tr>
<td>2 Incomes</td>
<td>67</td>
<td>26</td>
<td>91</td>
<td>22</td>
<td>94</td>
<td>78</td>
<td>105</td>
<td>51</td>
<td>20</td>
<td>0</td>
<td>104</td>
<td>162</td>
<td>22</td>
<td>91</td>
<td>02</td>
<td>10</td>
<td>1</td>
<td>102</td>
<td>62</td>
</tr>
<tr>
<td>Own</td>
<td>93</td>
<td>38</td>
<td>111</td>
<td>35</td>
<td>106</td>
<td>101</td>
<td>130</td>
<td>71</td>
<td>41</td>
<td>80</td>
<td>162</td>
<td>242</td>
<td>0</td>
<td>76</td>
<td>106</td>
<td>52</td>
<td>8</td>
<td>114</td>
<td>126</td>
</tr>
<tr>
<td>Rent</td>
<td>32</td>
<td>6</td>
<td>54</td>
<td>7</td>
<td>35</td>
<td>50</td>
<td>44</td>
<td>35</td>
<td>13</td>
<td>70</td>
<td>22</td>
<td>92</td>
<td>25</td>
<td>3</td>
<td>3</td>
<td>57</td>
<td>7</td>
<td>35</td>
<td>57</td>
</tr>
<tr>
<td>Married</td>
<td>37</td>
<td>13</td>
<td>51</td>
<td>9</td>
<td>42</td>
<td>50</td>
<td>50</td>
<td>35</td>
<td>16</td>
<td>10</td>
<td>91</td>
<td>76</td>
<td>25</td>
<td>101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>53</td>
<td>48</td>
</tr>
<tr>
<td>Married with Kids</td>
<td>50</td>
<td>15</td>
<td>44</td>
<td>21</td>
<td>51</td>
<td>37</td>
<td>79</td>
<td>12</td>
<td>13</td>
<td>27</td>
<td>82</td>
<td>106</td>
<td>3</td>
<td>0</td>
<td>109</td>
<td>0</td>
<td>0</td>
<td>48</td>
<td>61</td>
</tr>
<tr>
<td>Single</td>
<td>32</td>
<td>15</td>
<td>62</td>
<td>11</td>
<td>40</td>
<td>58</td>
<td>33</td>
<td>57</td>
<td>17</td>
<td>99</td>
<td>10</td>
<td>52</td>
<td>57</td>
<td>0</td>
<td>0</td>
<td>109</td>
<td>0</td>
<td>35</td>
<td>74</td>
</tr>
<tr>
<td>Single with Kids</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>Female</td>
<td>58</td>
<td>21</td>
<td>70</td>
<td>17</td>
<td>70</td>
<td>62</td>
<td>83</td>
<td>44</td>
<td>22</td>
<td>47</td>
<td>102</td>
<td>114</td>
<td>35</td>
<td>53</td>
<td>46</td>
<td>35</td>
<td>13</td>
<td>149</td>
<td>0</td>
</tr>
<tr>
<td>Male</td>
<td>67</td>
<td>23</td>
<td>95</td>
<td>25</td>
<td>71</td>
<td>89</td>
<td>91</td>
<td>62</td>
<td>32</td>
<td>103</td>
<td>82</td>
<td>128</td>
<td>57</td>
<td>48</td>
<td>61</td>
<td>74</td>
<td>2</td>
<td>0</td>
<td>185</td>
</tr>
</tbody>
</table>

more info see [here](#)
MCA Example (2)

Summary table:

<table>
<thead>
<tr>
<th>Singular Value</th>
<th>Principal Inertia</th>
<th>Chi-Square</th>
<th>Percent</th>
<th>Cumulative Percent</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>16</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.56934</td>
<td>0.32415</td>
<td>970.77</td>
<td>18.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.48352</td>
<td>0.23380</td>
<td>700.17</td>
<td>13.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.42716</td>
<td>0.18247</td>
<td>546.45</td>
<td>10.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.41215</td>
<td>0.16987</td>
<td>508.73</td>
<td>9.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.38773</td>
<td>0.15033</td>
<td>450.22</td>
<td>8.77</td>
<td>61.87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.38520</td>
<td>0.14838</td>
<td>444.35</td>
<td>8.66</td>
<td>70.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.34066</td>
<td>0.11605</td>
<td>347.55</td>
<td>6.77</td>
<td>77.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.32983</td>
<td>0.10879</td>
<td>325.79</td>
<td>6.35</td>
<td>83.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.31517</td>
<td>0.09933</td>
<td>297.47</td>
<td>5.79</td>
<td>89.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.28069</td>
<td>0.07879</td>
<td>235.95</td>
<td>4.60</td>
<td>94.03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.26115</td>
<td>0.06820</td>
<td>204.24</td>
<td>3.98</td>
<td>98.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.18477</td>
<td>0.03414</td>
<td>102.24</td>
<td>1.99</td>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total 1.71429 5133.92 100.00

Degrees of Freedom = 324
Most influential column points (loadings):

<table>
<thead>
<tr>
<th>Column</th>
<th>Dim1</th>
<th>Dim2</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>-0.4035</td>
<td>0.8129</td>
</tr>
<tr>
<td>European</td>
<td>-0.0568</td>
<td>-0.5552</td>
</tr>
<tr>
<td>Japanese</td>
<td>0.3208</td>
<td>-0.4678</td>
</tr>
<tr>
<td>Large</td>
<td>-0.6949</td>
<td>1.5666</td>
</tr>
<tr>
<td>Medium</td>
<td>-0.2562</td>
<td>0.0965</td>
</tr>
<tr>
<td>Small</td>
<td>0.4326</td>
<td>-0.5258</td>
</tr>
<tr>
<td>Family</td>
<td>-0.4201</td>
<td>0.3602</td>
</tr>
<tr>
<td>Sporty</td>
<td>0.6604</td>
<td>-0.6696</td>
</tr>
<tr>
<td>Work</td>
<td>0.0575</td>
<td>0.1539</td>
</tr>
<tr>
<td>1 Income</td>
<td>0.8251</td>
<td>0.5472</td>
</tr>
<tr>
<td>2 Incomes</td>
<td>-0.6727</td>
<td>-0.4461</td>
</tr>
<tr>
<td>Own</td>
<td>-0.3887</td>
<td>-0.0943</td>
</tr>
<tr>
<td>Rent</td>
<td>1.0225</td>
<td>0.2480</td>
</tr>
<tr>
<td>Married</td>
<td>-0.4169</td>
<td>-0.7954</td>
</tr>
<tr>
<td>Married with Kids</td>
<td>-0.8200</td>
<td>0.3237</td>
</tr>
<tr>
<td>Single</td>
<td>1.1461</td>
<td>0.2930</td>
</tr>
<tr>
<td>Single with Kids</td>
<td>0.4373</td>
<td>0.8736</td>
</tr>
<tr>
<td>Female</td>
<td>-0.3365</td>
<td>-0.2057</td>
</tr>
<tr>
<td>Male</td>
<td>0.2710</td>
<td>0.1656</td>
</tr>
</tbody>
</table>
MCA Example (4)

Burt table plot:
Plot Observations

Top-right quadrant:
- categories single, single with kids, 1 income, and renting a home are associated

Proceeding clockwise:
- the categories sporty, small, and Japanese are associated
- being married, owning your own home, and having two incomes are associated
- having children is associated with owning a large American family car

Such information could be used in market research to identify target audiences for advertisements
A Gartner Magic Quadrant is a culmination of research in a specific market, providing a wide-angle view of the relative positions of the market's competitors.

This concept can be used for other dimension pairs as well:
- essentially require to think of a segmentation of the 4 quadrants.
Figure 1. Magic Quadrant for Business Intelligence and Analytics Platforms

Source: Gartner (February 2014)