CSE 564
Visualization & Visual Analytics

Visualizing High-Dimensional Data: non-Linear Methods

Klaus Mueller

Computer Science Department
Stony Brook University

Lecture	Topic	Projects
1	Intro and logistics	
2	Basic visualizations and tasks, data types, examples, ethical considerations	
3	Data preparation (cleaning, imputation, data set integration)	
4	AI-assisted coding for VIS applications (design, debugging, refactoring)	Project #1 out
5	Big data and data reduction (distance/sim metrics, intro to clustering)	
6	High-D data: concept, subspaces, dimension reduction, PCA	
7	Cluster analysis: hierarchical, density, model, embedding, temporal	
8	Perception and cognition (human visual system, color, contrast)	Project #2(a) out
9	Visual design and aesthetics	
10	Visualization of multivariate and high-D data: linear methods, projections	
11	Vis. of multivariate and high-D data: non-linear methods, embeddings	
12	Visualization and AI: mutual support and capabilities (VIS4AI, AI4VIS)	Project #2(b) out
13	Principles of interaction: drive what is visualized, analyzed & how (HCI4VIS)	
14	Visual analytics (VA), human-centered AI, mixed-initiative system	
15	Midterm #1 (tentative date)	
16	VA system design and evaluation, collaborative VA, uncertainty, provenance	
17	Midterm #1 discussion (tentative date)	Final proj. proposal call out
18	Visualization of hierarchical data	
19	Visualization of maps and data with geo-reference	
20	Visualization of graphs, networks (incl. derivation of causal networks)	Final project proposal due
21	Vis. of time-varying, time-series, streaming data, progressive visualization	
22	Visualization of text, LLMs, and semantic data	
23	Ed Tufte revisited: principles, critiques and limits, responsible visualization	
24	Design of effective infographics	Final proj. prelim report due
25	Foundations scientific and medical visualization, intro to volume rendering	
26	Scientific visualization	Bonus project out (Vol Ren)
27	Story telling with data, data journalism	
28	Midterm #2 (tentative date)	
Final	Final project demo on zoom (public)	All final proj. materials due

Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS)

MDS preserves similarity relationships, prevents ambiguity

- scattered points in high-dimensions (N-D)
- adjacency matrices

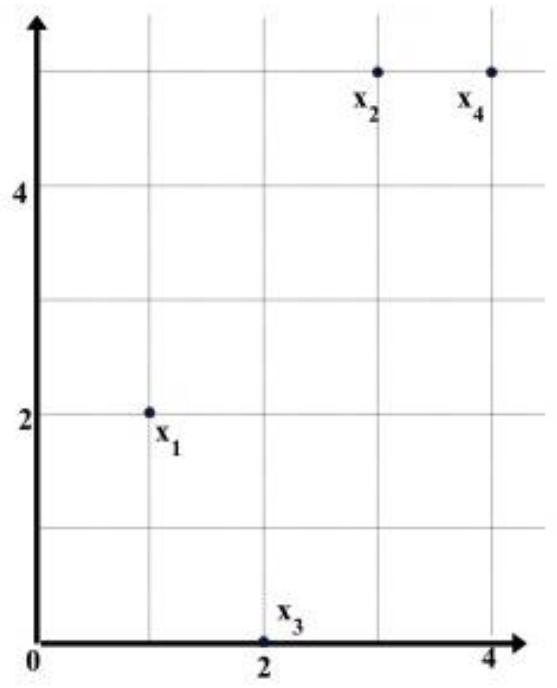
Maps the distances between observations from N-D into low-D (say 2D)

- 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 father apart

The Dissimilarity Matrix



Data Matrix

point	attribute1	attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Dissimilarity Matrix

(with Euclidean Distance)

	x1	*x2*	*x3*	*x4*
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Distance Matrix

MDS turns a distance matrix into a network or point cloud

- correlation, cosine, Euclidian, and so on

Suppose you know a matrix of distances among cities

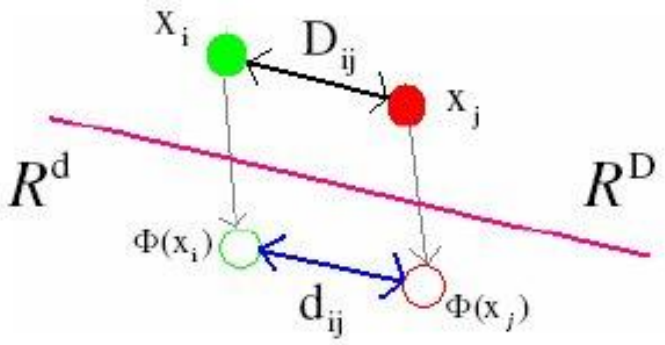
	Chicago	Raleigh	Boston	Seattle	S.F.	Austin	Orlando
Chicago	0						
Raleigh	641	0					
Boston	851	608	0				
Seattle	1733	2363	2488	0			
S.F.	1855	2406	2696	684	0		
Austin	972	1167	1691	1764	1495	0	
Orlando	994	520	1105	2565	2458	1015	0

Result of MDS

Compare with real Map

MDS Algorithm

- Task:
  - Find that configuration of image points whose pairwise distances are most similar to the original inter-point distances !!!
- Formally:
  - Define: $D_{ij} = \| x_i - x_j \|_D \qquad d_{ij} = \| y_i - y_j \|_d$
  - Claim: $D_{ij} \equiv d_{ij} \qquad \forall i, j \in [1, n]$
- In general: an exact solution is not possible !!!
- Inter Point distances → 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):

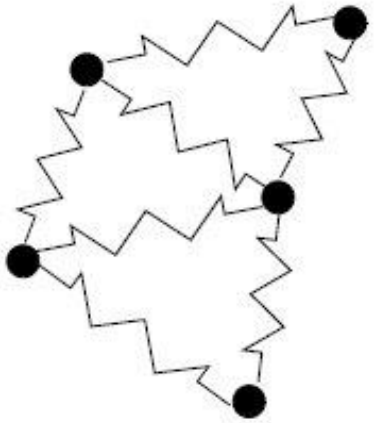
- iterative procedure to find a good configuration of image points
  - 1) Initialization
    → Begin with some (arbitrary) initial configuration
  - 2) Alter the image points and try to find a configuration of points that minimizes the following sum-of-squares error function:

$$E = \sum_{i<j}^{N} \left(D_{ij} - d_{ij}\right)^2$$

Force-Directed Algorithm

Spring-like system

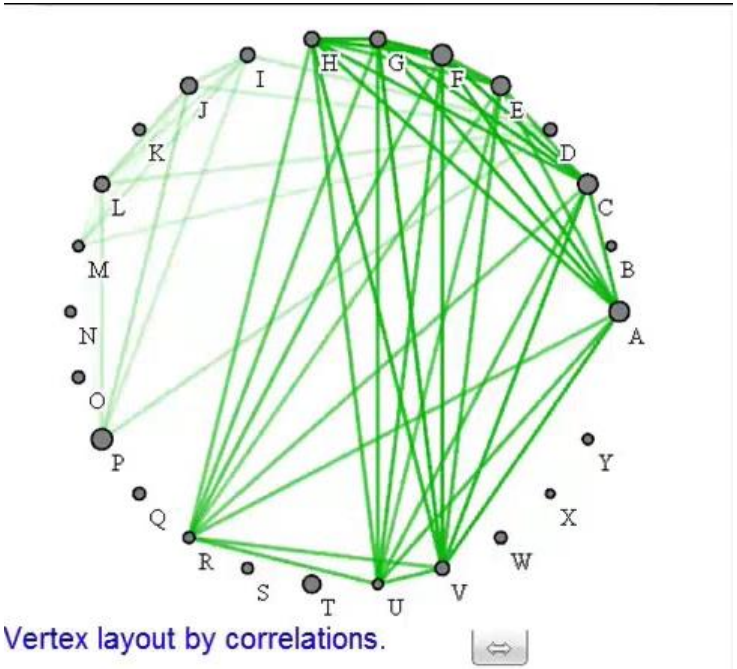
- insert springs within each node
- the length of the spring encodes the desired node distance
- start at an initial configuration
- iteratively move nodes until an energy minimum is reached



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

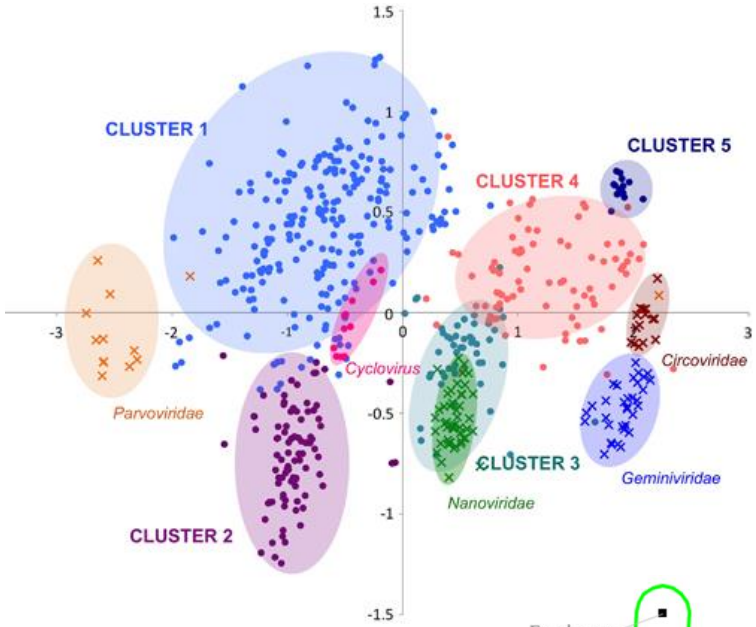


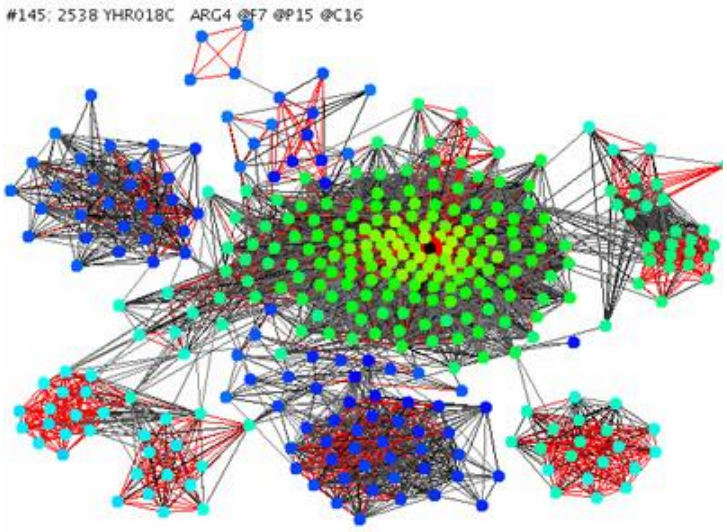
Uses of MDS

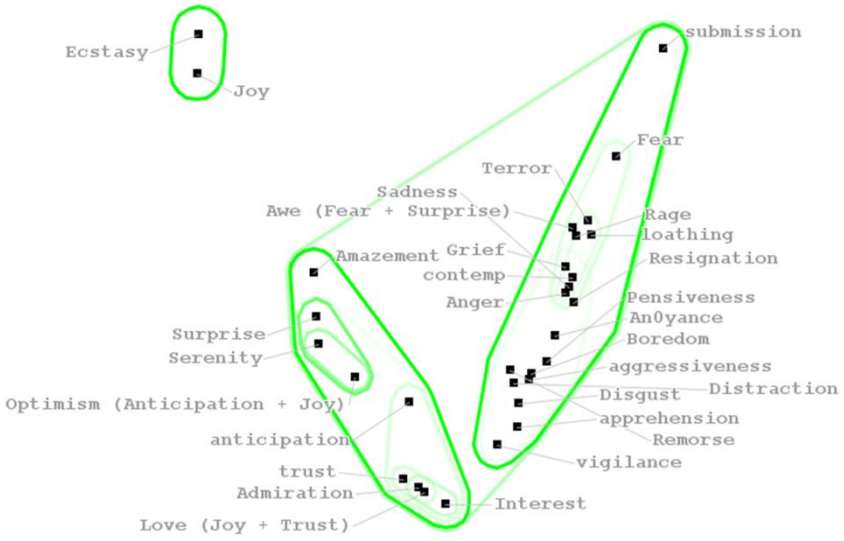
Distance (similarity) metric

- Euclidian distance (best for data)
- Cosine distance (best for data)
- 1-|correlation| distance (best for attributes)
- use | | if you do not care about positive or negative correlations
- leave off | | if you want positively correlated attribute points closer

MDS Examples







MDS in Scikit-learn

sklearn.manifold.MDS

class sklearn.manifold.**MDS**(*n_components=2, metric=True, n_init=4, max_iter=300, verbose=0, eps=0.001, n_jobs=1, random_state=None, dissimilarity='euclidean'*) [source]

sklearn.manifold.**MDS**(

n_components=2,

metric=True,

n_init=4, Number of time the smacof algorithm will be run with different initialisation.
The final results will be the best output of the n_init consecutive runs in terms of stress.

max_iter=300, Maximum number of iterations of the SMACOF algorithm for a single run

verbose=0,

eps=0.001, relative tolerance w.r.t stress to declare converge

n_jobs=1,

random_state=None,

dissimilarity='euclidean') Which dissimilarity measure to use. Supported are 'euclidean' and 'precomputed'.

The **SMACOF** (Scaling by MAjorizing a COmplicated Function) algorithm is a multidimensional scaling algorithm which minimizes an objective function (the *stress*) using a majorization technique.

Pros and Cons of MDS

Pros:

- preserve neighborhood relations of high-D points
- no projection ambiguities

Cons:

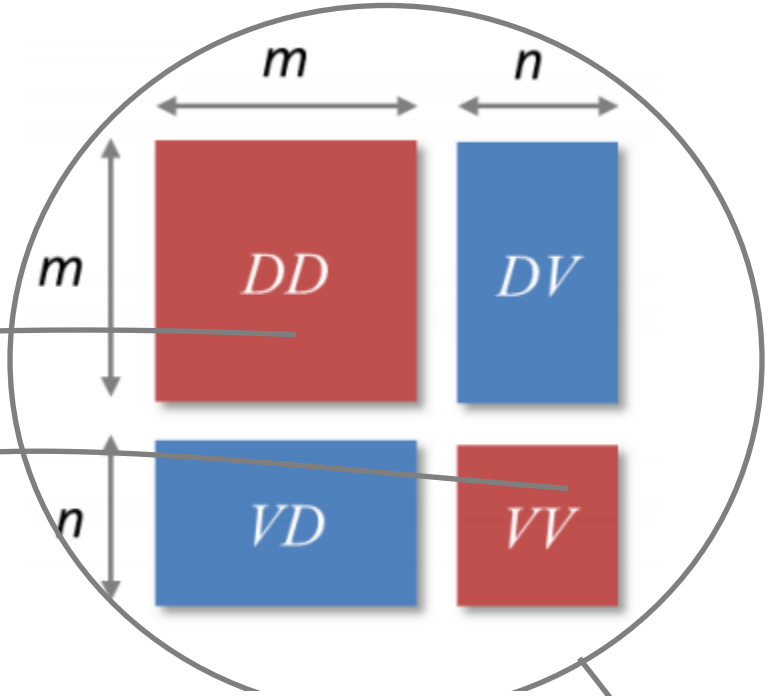
- relationship to data dimensions is lost
- context-less

Are there visualization paradigms that can overcome these problems?

- next slides

The Data Context Map

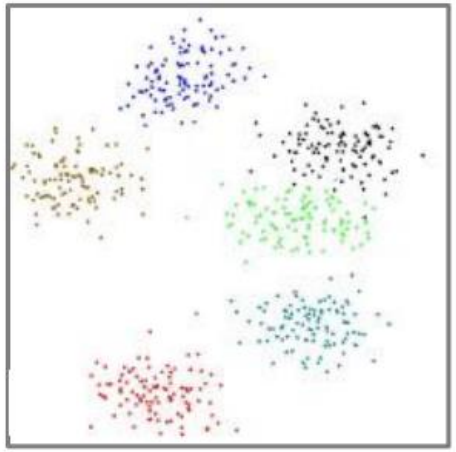
Uses MDS



data similarity matrix (DD)

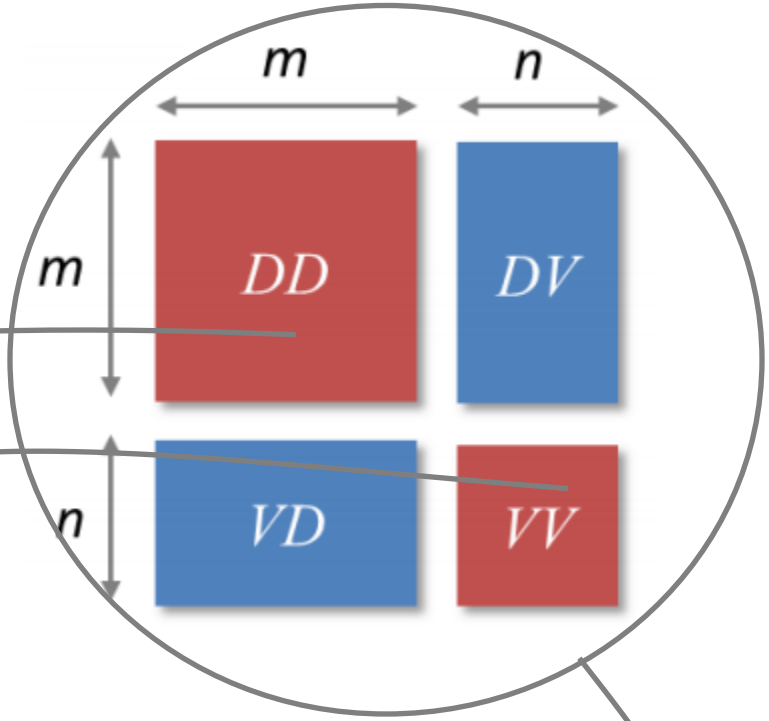
attribute similarity matrix (VV)

Data layout



+ data-attribute similarity matrix (VD, DV)

Uses MDS



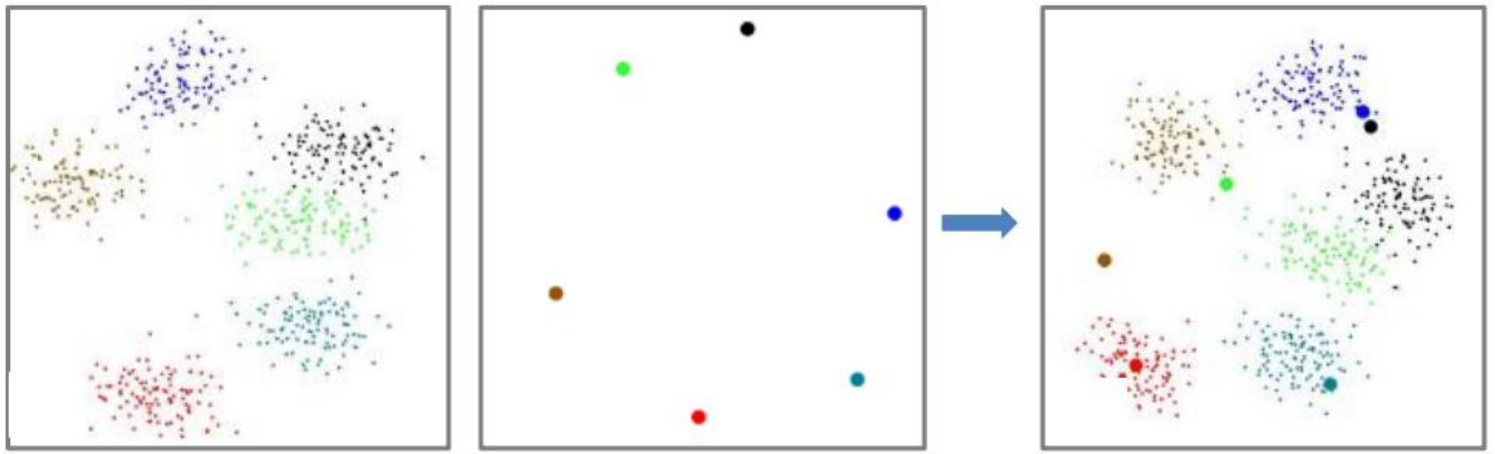
data similarity matrix (DD)

attribute similarity matrix (VV)

Data layout

Attribute layout

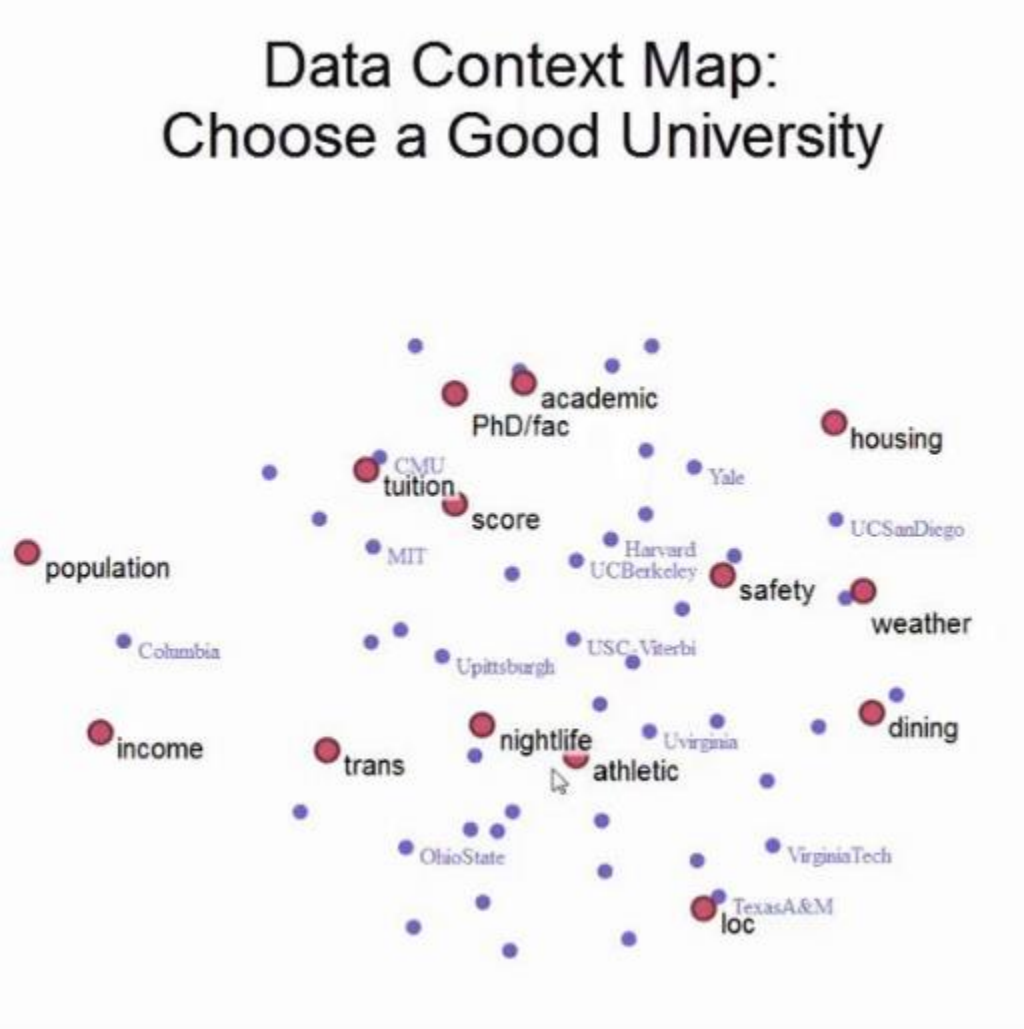
Combined layout



+ data-attribute similarity matrix (VD, DV)

Generates the Data Context Map

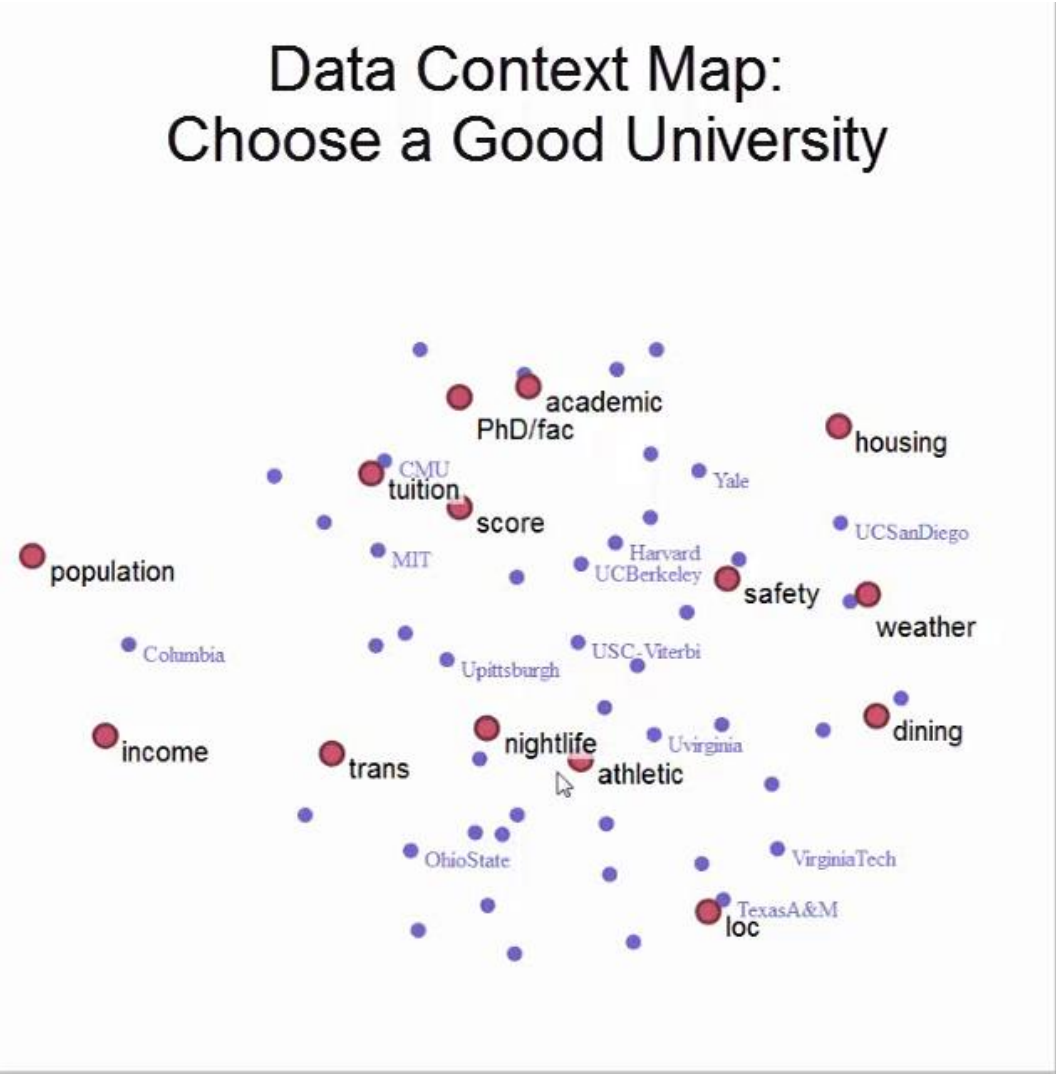
Data visualized in the context of the attributes

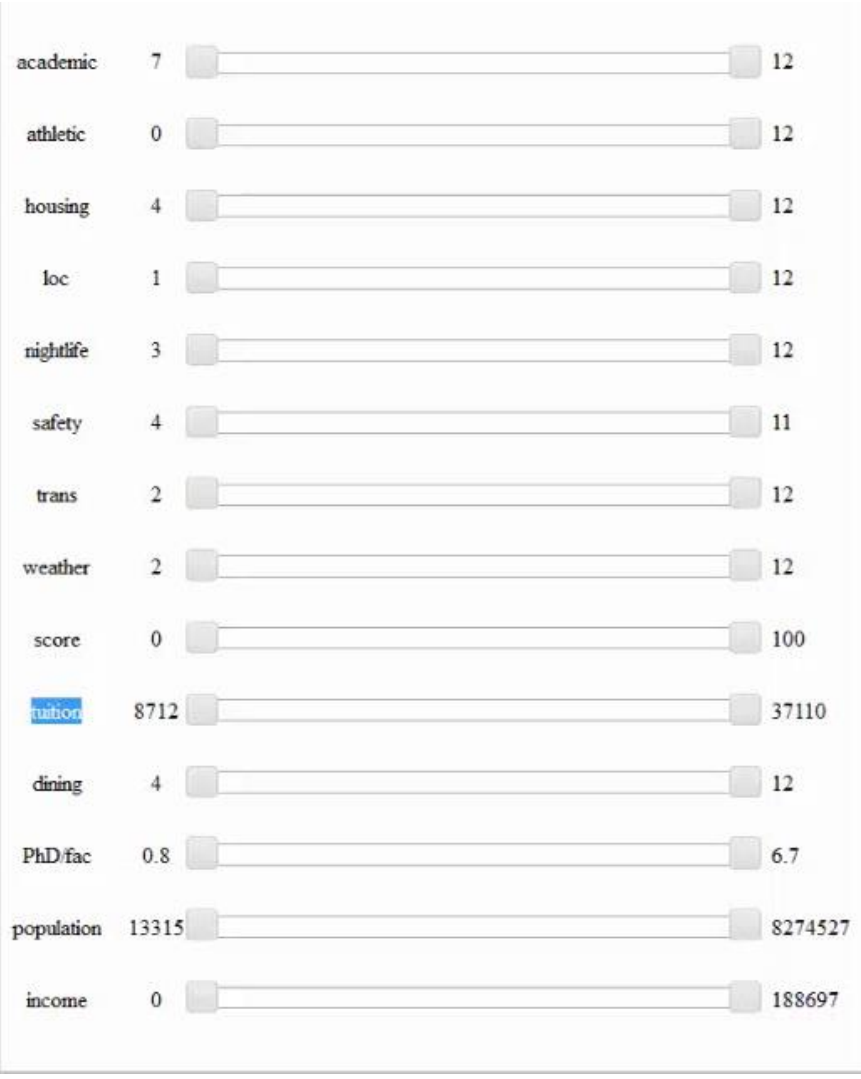


S. Cheng, K. Mueller, "The Data Context Map: Fusing Data and Attributes into a Unified Display," *IEEE Trans. on Visualization and Computer Graphics*, 22(1): 121-130, 2016.

youtube

Data Context Map in Action





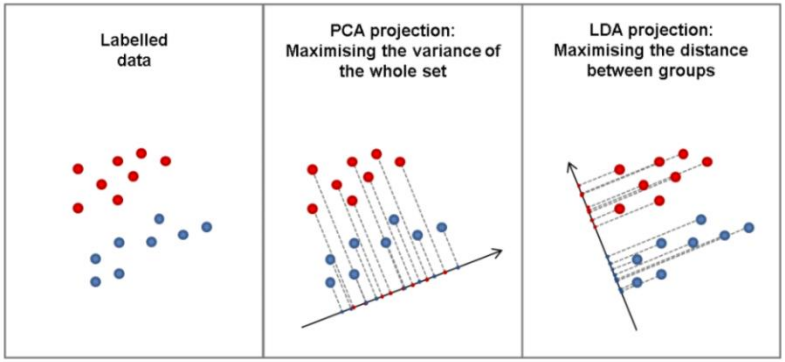
Embeddings

Linear Discriminate Analysis (LDA)

Procedure

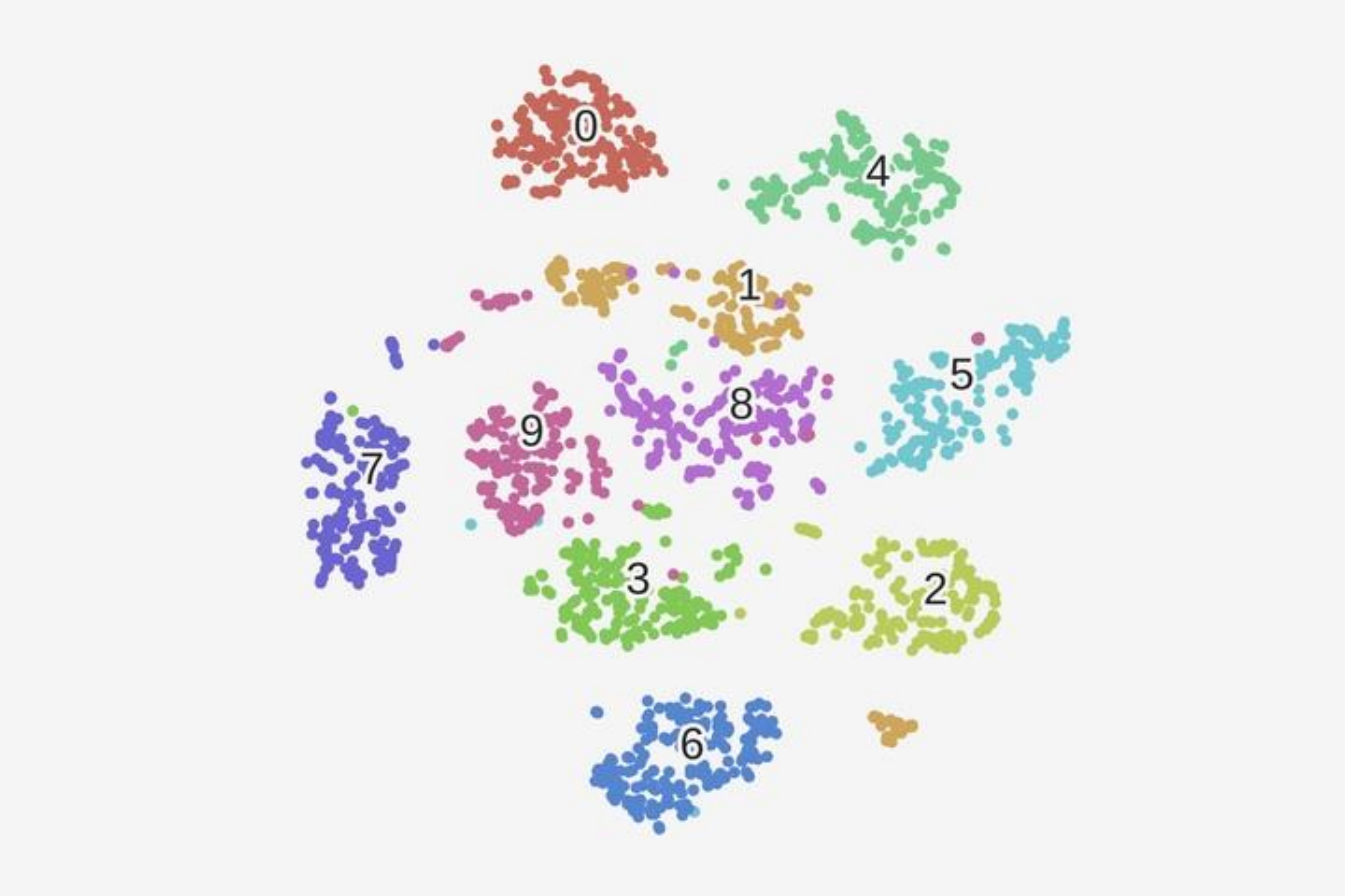
- maximize inter-class variance $S_b = \sum_{i=1}^{g} N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$
- minimize intra-class variance $S_w = \sum_{i=1}^{g} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$
- using this ratio $P_{lda} = \arg\max_P \frac{\left| P^T S_b P \right|}{\left| P^T S_w P \right|}$ ← Fisher Criterion

  P is low-Dim projection
- can be solved using Eigenvector decomposition
- finds a basis that maximally separates the classes
- Dim(P) is the # of classes *g*



T-SNE

t-distributed stochastic neighbor embedding



T-SNE Distance Metric

Uses the following density-based (probabilistic) distance metric

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

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

- 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

T-SNE Implementation

Use a symmetrized version of the conditional similarity:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

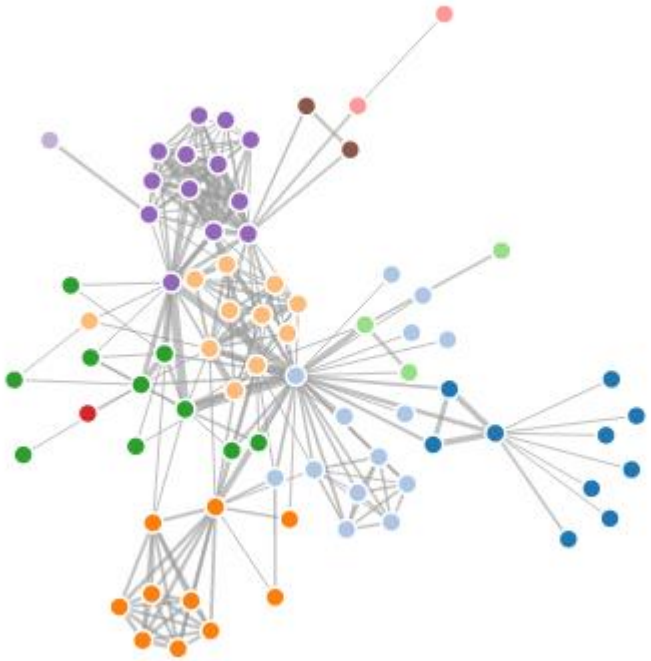
Similarity (distance) metric for mapped points:

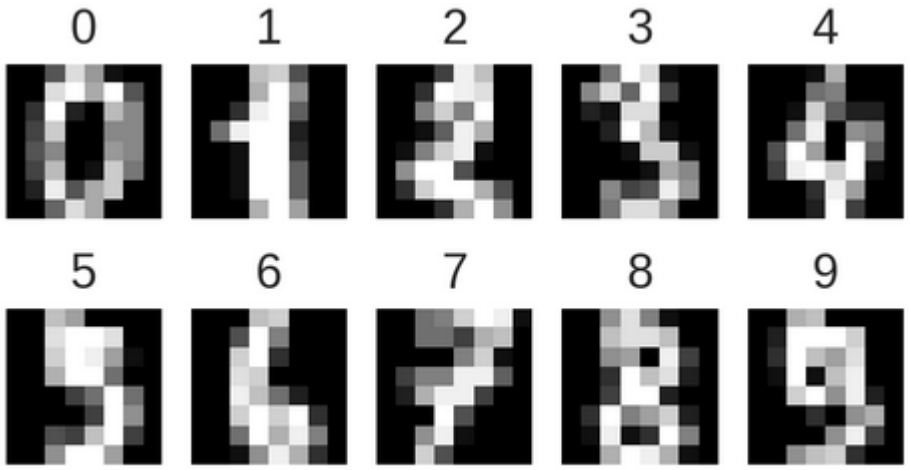
$$q_{ij} = \frac{f(|x_i - x_j|)}{\sum_{k \neq i} f(|x_i - x_k|)} \quad \text{with} \quad f(z) = \frac{1}{1+z^2}$$

This uses the t-student distribution with one degree of freedom, or Cauchy distribution, instead of a Gaussian distribution

Layout

Can use mass-spring system enforcing minimum of $|p_{ij} - q_{ij}|$





The classic *handwritten digits* datasets. It contains 1,797 images with 8∗8=64 pixels each.

Animated layout

9

More information

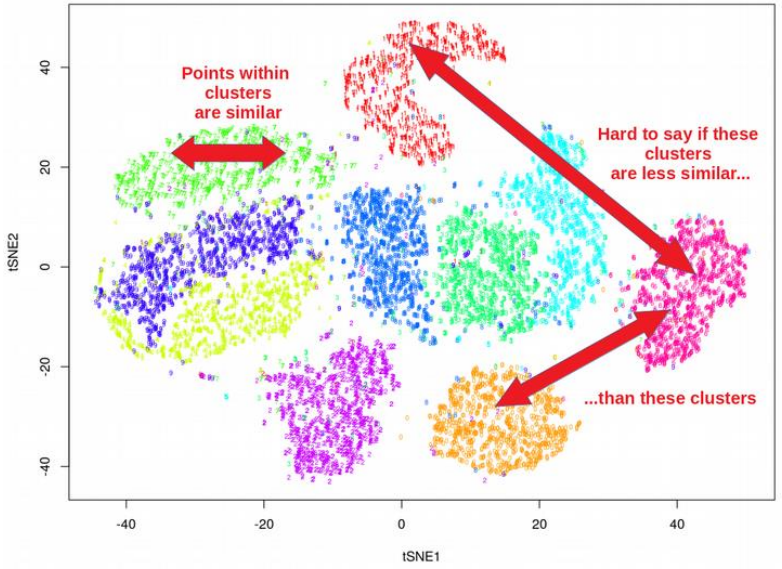
See this webpage

Shortcomings of t-SNE

t-SNE does not preserve global data structure

- only within cluster distances are meaningful
- between cluster similarities are not guaranteed

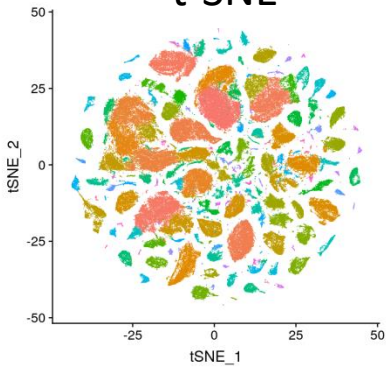
t-SNE MINST



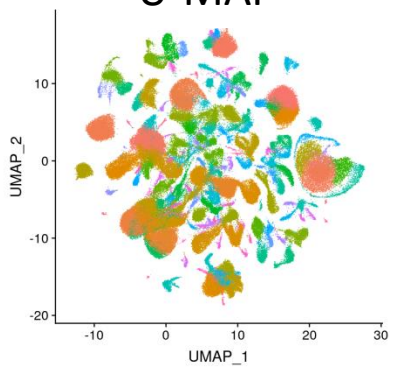
More recently introduced: U-MAP

- follows the philosophy of t-SNE
- but introduces many improvements
- more info, for example, here

t-SNE



U-MAP



Embedding of Categorical data

Text Processing

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?

Singular Value Decomposition (SVD)

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

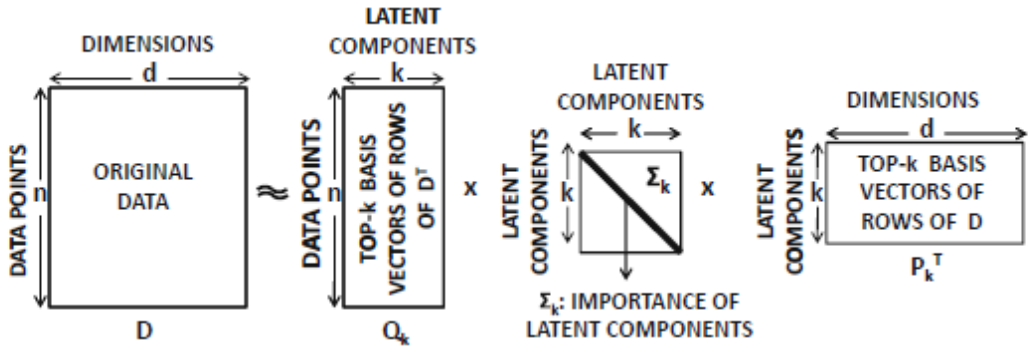
SVD often used for text analysis

- values close to zero are frequent and should not affect the analysis

Singular Value Decomposition (SVD)

Decomposes *C* into the matrix:

$$Q_k \Sigma_k P_k^T$$



q_i and p_i are two column vectors with significance σ_i

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

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

SVD Computation

Find the matrices **U**, **D,** and **V** such that:

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

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

Latent Semantic Analysis

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)* = simplest form is frequency of *t* in *d* = *f(t,d)*

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

Latent Semantic Analysis

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)* = simplest form is frequency of *t* in *d* = *f(t,d)*

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

- *N* = 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

A

M	D_1	D_2	D_3	D_4	D_5	D_6	$\cdots$	D_n
T_1	0.00060	0.00012	0.00003	0.00003	0.00333	0.00048	$\cdots$	a_{1n}
T_2	0	0	0	0	0	0	$\cdots$	a_{2n}
T_3	0	2.98862	0	0	0	1.49431	$\cdots$	a_{3n}
T_4	0	0	0	13.32555	0	0	$\cdots$	a_{4n}
T_5	0	0	0	0	0	0	$\cdots$	a_{5n}
T_6	1.03442	1.03442	0	0	0	3.10326	$\cdots$	a_{6n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
T_m	a_{m1}	a_{m2}	a_{m3}	a_{m4}	a_{m5}	a_{m6}	$\cdots$	a_{mn}

A

M	D_1	D_2	D_3	D_4	D_5	D_6	⋯	D_n
T_1	0.00060	0.00012	0.00003	0.00003	0.00333	0.00048	⋯	a_{1n}
T_2	0	0	0	0	0	0	⋯	a_{2n}
T_3	0	2.98862	0	0	0	1.49431	⋯	a_{3n}
T_4	0	0	0	13.32555	0	0	⋯	a_{4n}
T_5	0	0	0	0	0	0	⋯	a_{5n}
T_6	1.03442	1.03442	0	0	0	3.10326	⋯	a_{6n}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
T_m	a_{m1}	a_{m2}	a_{m3}	a_{m4}	a_{m5}	a_{m6}	⋯	a_{mn}

B

U = term-concept matrix

concept = latent (hidden) *topic*

sort and keep the k most significant rows/columns

V = concept-document matrix

$$U = U_k = \begin{array}{c|cccc} & C_1 & C_2 & C_3 & \cdots & C_m \\ T_1 & a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ T_2 & a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ T_3 & a_{31} & a_{32} & a_{33} & \cdots & a_{3m} \\ T_4 & a_{41} & a_{42} & a_{43} & \cdots & a_{4m} \\ T_5 & a_{51} & a_{52} & a_{53} & \cdots & a_{5m} \\ T_6 & a_{61} & a_{62} & a_{63} & \cdots & a_{6m} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ T_m & a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mm} \end{array}$$

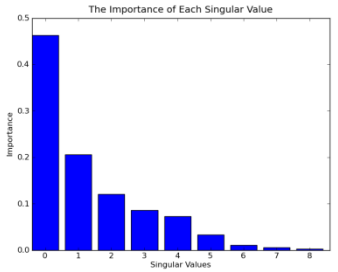
$$\Sigma = \Sigma_k = \begin{array}{c|cccc} & D_1 & D_2 & D_3 & \cdots & D_n \\ T_1 & a_{11} & 0 & 0 & \cdots & 0 \\ T_2 & 0 & a_{22} & 0 & \cdots & 0 \\ T_3 & 0 & 0 & a_{33} & \cdots & 0 \\ T_4 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ T_m & 0 & 0 & 0 & \cdots & a_{mm} \end{array}$$

$$V^T = V_k^T = \begin{array}{c|ccccc} & D_1 & D_2 & D_3 & \cdots & D_n \\ C_1 & a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ C_2 & a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ C_3 & a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ C_4 & a_{41} & a_{42} & a_{43} & \cdots & a_{4n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ C_n & a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{array}$$

Visualizing The Concept Space

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



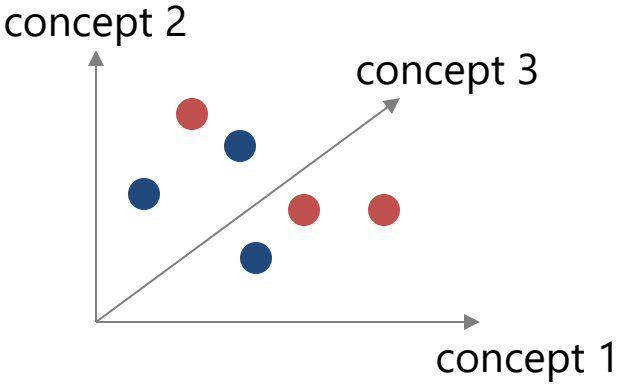
Throw out the 1st dimension in U and V

- 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

document

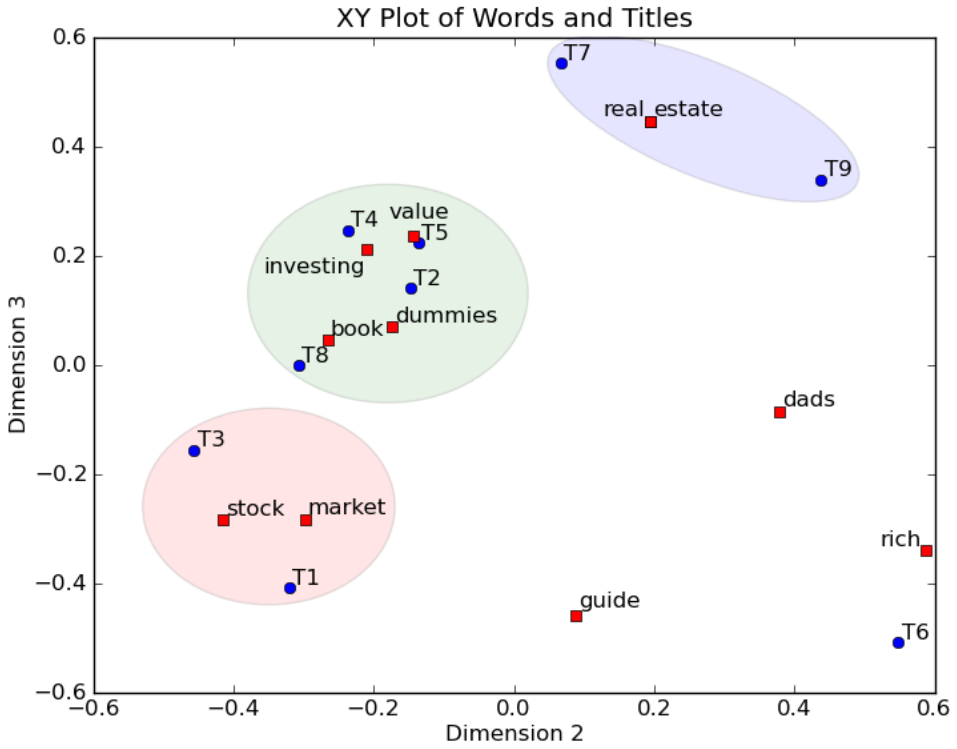
term



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 Disadvantages

LSA assumes a Gaussian distribution and Frobenius norm

- this may not fit all problems

LSA cannot handle polysemy effectively

- need LDA (Latent Dirichlet Allocation) for this

LSA depends heavily on SVD

- computationally intensive
- hard to update as new documents appear
- but faster algorithms have emerged recently

What About Categorical Variables?

You will need to use correspondence analysis (CA)

- CA is PCA for categorical variables
- related to factor analysis

Makes use of the χ^2 test

- what's χ^2 ?

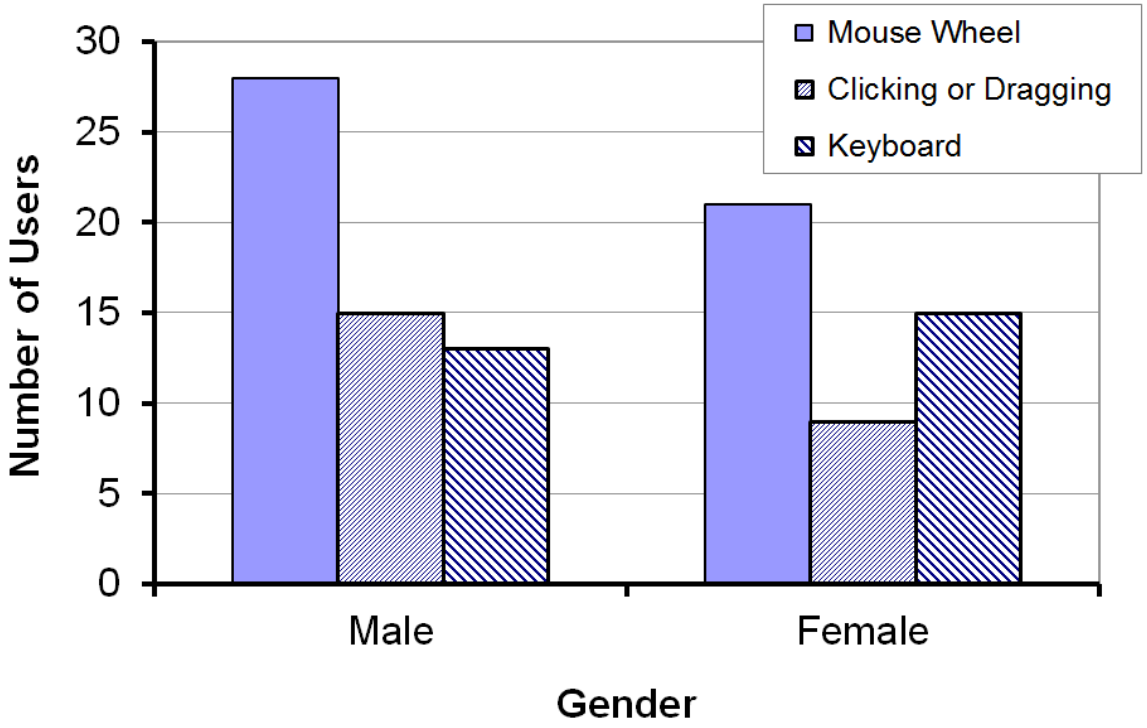
Chi-square Test (Nominal Data)

- 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

Observed Number of Users				
Gender	Scrolling Method			Total
	MW	CD	KB	
Male	28	15	13	56
Female	21	9	15	45
Total	49	24	28	101

MW = mouse wheel
CD = clicking, dragging
KB = keyboard



Chi-square – Example #1

56.0·49.0/101=27.2

Expected Number of Users				
Gender	Scrolling Method			Total
	MW	CD	KB	
Male	27.2	13.3	15.5	56.0
Female	21.8	10.7	12.5	45.0
Total	49.0	24.0	28.0	101

$(\text{Expected-Observed})^2/\text{Expected} = (28-27.2)^2/27.2$

Chi Squares				
Gender	Scrolling Method			Total
	MW	CD	KB	
Male	0.025	0.215	0.411	0.651
Female	0.032	0.268	0.511	0.811
Total	0.057	0.483	0.922	**1.462**

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 = number of rows, c = number of columns

Significance Threshold (α)	Degrees of Freedom							
	1	2	3	4	5	6	7	8
.1	2.71	4.61	6.25	7.78	9.24	10.65	12.02	13.36
.05	3.84	5.99	7.82	9.49	11.07	12.59	14.07	15.51
.01	6.64	9.21	11.35	13.28	15.09	16.81	18.48	20.09
.001	10.83	13.82	16.27	18.47	20.52	22.46	24.32	26.13

$\chi^2 = 1.462$ (< 5.99 ∴ not significant)

Correspondence Analysis (CA)

Example:

more info

	Smoking Category				
Staff Group	(1) None	(2) Light	(3) Medium	(4) Heavy	Row Totals
(1) Senior Managers	4	2	3	2	11
(2) Junior Managers	4	3	7	4	18
(3) Senior Employees	25	10	12	4	51
(4) Junior Employees	18	24	33	13	88
(5) Secretaries	10	6	7	2	25
Column Totals	61	45	62	25	193

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 ?

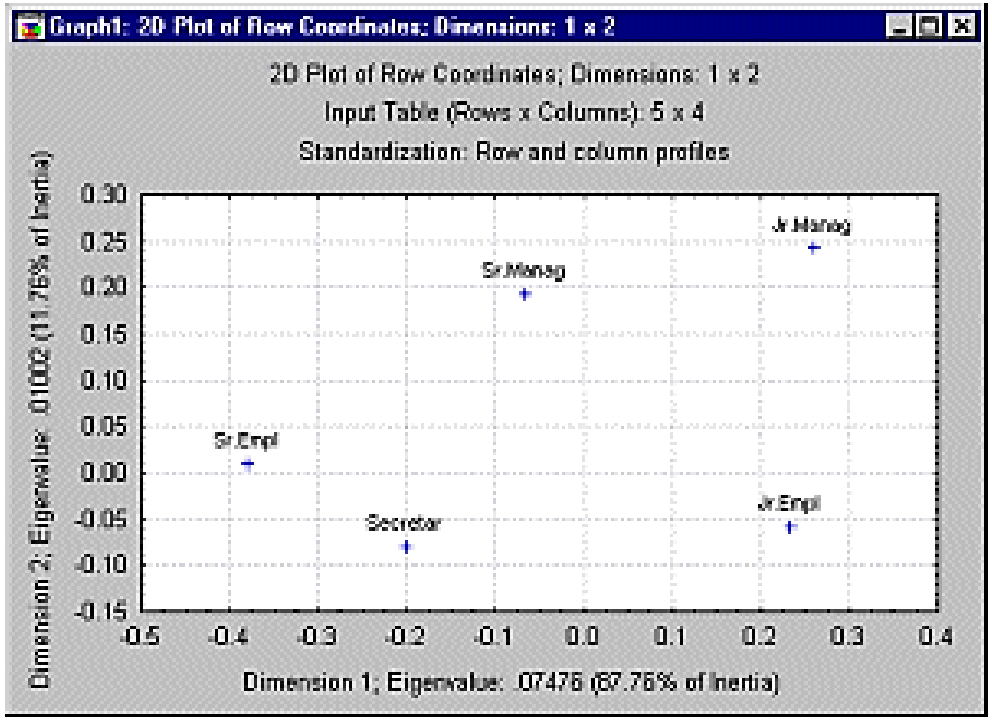
- this occurs when the χ^2 statistics of the table is insignificant

CA Eigen Analysis

Staff Group	Smoking Category (1) None	(2) Light	(3) Medium	(4) Heavy	Row Totals
(1) Senior Managers	4	2	3	2	11
(2) Junior Managers	4	3	7	4	18
(3) Senior Employees	25	10	12	4	51
(4) Junior Employees	18	24	33	13	88
(5) Secretaries	10	6	7	2	25
Column Totals	61	45	62	25	193

Let's do some plotting

- compute distance matrix of the rows $\mathbf{CC}^T$
- 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

Eigenvalues and Inertia for all Dimensions Input Table (Rows x Columns): 5 x 4 Total Inertia = .08519 Chi^2 = 16.442					
No. of Dims	Singular Values	Eigen-Values	Perc. of Inertia	Cumulatv Percent	Chi Squares
1	.273421	.074759	87.75587	87.7559	14.42851
2	.100086	.010017	11.75865	99.5145	1.93332
3	.020337	.000414	.48547	100.0000	.07982

CA Eigen Analysis

	Smoking Category				
Staff Group	(1) None	(2) Light	(3) Medium	(4) Heavy	Row Totals
(1) Senior Managers	4	2	3	2	11
(2) Junior Managers	4	3	7	4	18
(3) Senior Employees	25	10	12	4	51
(4) Junior Employees	18	24	33	13	88
(5) Secretaries	10	6	7	2	25
Column Totals	61	45	62	25	193

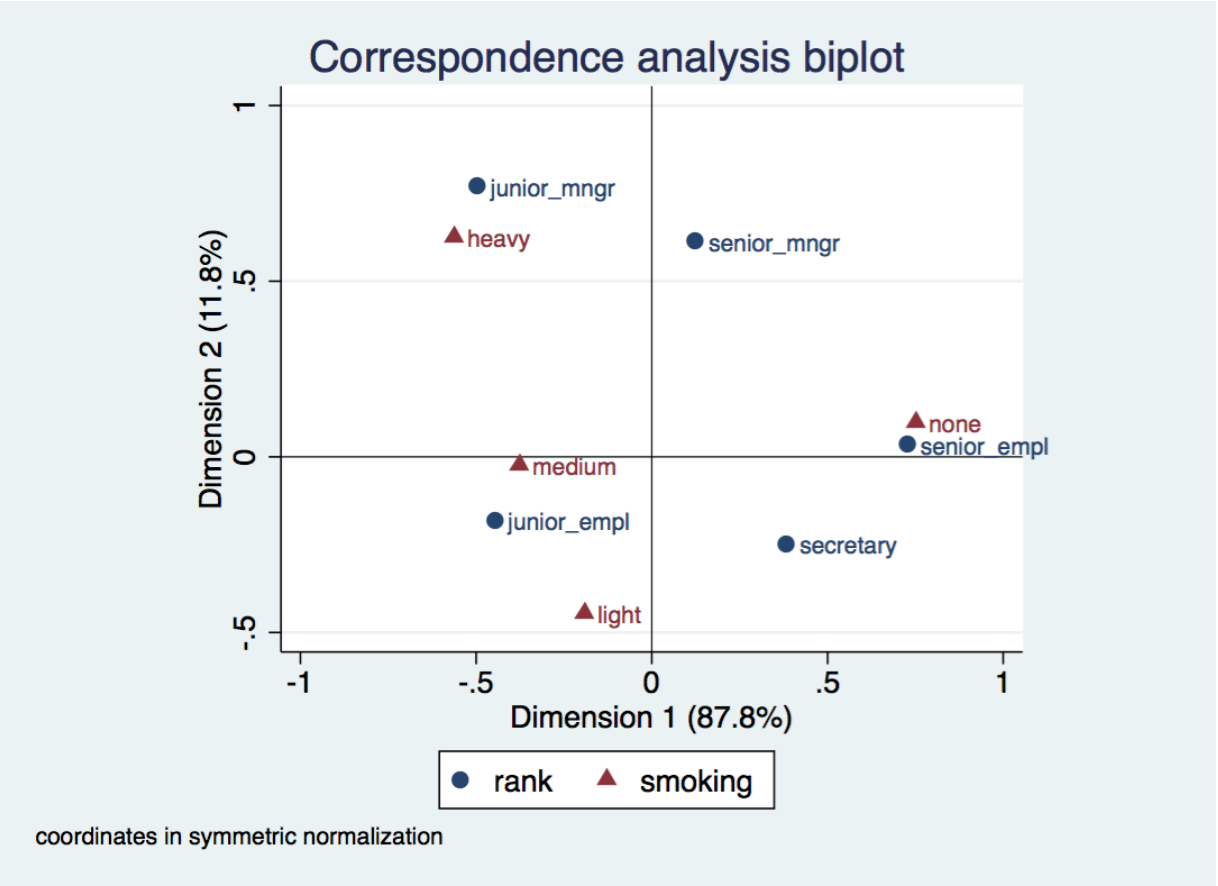
Next:

- compute distance matrix of the columns $\mathbf{C}^T\mathbf{C}$
- 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 χ^2 statistics was significant we should see some dependencies

Combined CA Plot



Interpretation sample (using the χ^2 frequentist mindset)

- *relatively speaking,* there are more non-smoking senior employees

Extending to Cases

Case Number	Senior Manager	Junior Manager	Senior Employee	Junior Employee	Secretary	None	Light	Medium	Heavy
1	1	0	0	0	0	1	0	0	0
2	1	0	0	0	0	1	0	0	0
3	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	1	0	0	0
5	1	0	0	0	0	0	1	0	0
...	.	.	.	.	.	.	.	.	.
...	.	.	.	.	.	.	.	.	.
...	.	.	.	.	.	.	.	.	.
191	0	0	0	0	1	0	0	1	0
192	0	0	0	0	1	0	0	0	1
193	0	0	0	0	1	0	0	0	1

Plot would now show 193 cases and 9 variables

Multiple Correspondence Analysis

Extension where there are more than 2 categorical variables

	SURVIVAL		AGE			LOCATION		
Case No.	NO	YES	LESST50	A50TO69	OVER69	TOKYO	BOSTON	GLAMORGN
1	0	1	0	1	0	0	0	1
2	1	0	1	0	0	1	0	0
3	0	1	0	1	0	0	1	0
4	0	1	0	0	1	0	0	1
...	.	.	.	.	.	.	.	.
...	.	.	.	.	.	.	.	.
...	.	.	.	.	.	.	.	.
762	1	0	0	1	0	1	0	0
763	0	1	1	0	0	0	1	0
764	0	1	0	1	0	0	0	1

Let's call it matrix X

Multiple Correspondence Analysis

Compute X'X to get the Burt Table

	SURVIVAL		AGE			LOCATION		
	NO	YES	<50	50-69	69+	TOKYO	BOSTON	GLAMORGN
SURVIVAL:NO	210	0	68	93	49	60	82	68
SURVIVAL:YES	0	554	212	258	84	230	171	153
AGE:UNDER_50	68	212	280	0	0	151	58	71
AGE:A_50TO69	93	258	0	351	0	120	122	109
AGE:OVER_69	49	84	0	0	133	19	73	41
LOCATION:TOKYO	60	230	151	120	19	290	0	0
LOCATION:BOSTON	82	171	58	122	73	0	253	0
LOCATION:GLAMORGN	68	153	71	109	41	0	0	221

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)

Larger MCA Example

Results of a survey of car owners and car attributes

Burt Table

	American	European	Japanese	Large	Medium	Small	Family	Sporty	Work	1 Income	2 Incomes	Own	Rent	Married	Married with Kids	Single	Single with Kids	Female	Male
American	125	0	0	36	60	29	81	24	20	58	67	93	32	37	50	32	6	58	67
European	0	44	0	4	20	20	17	23	4	18	26	38	6	13	15	15	1	21	23
Japanese	0	0	165	2	61	102	76	59	30	74	91	111	54	51	44	62	8	70	95
Large	36	4	2	42	0	0	30	1	11	20	22	35	7	9	21	11	1	17	25
Medium	60	20	61	0	141	0	89	39	13	57	84	106	35	42	51	40	8	70	71
Small	29	20	102	0	0	151	55	66	30	73	78	101	50	50	37	58	6	62	89
Family	81	17	76	30	89	55	174	0	0	69	105	130	44	50	79	35	10	83	91
Sporty	24	23	59	1	39	66	0	106	0	55	51	71	35	35	12	57	2	44	62
Work	20	4	30	11	13	30	0	0	54	26	28	41	13	16	18	17	3	22	32
1 Income	58	18	74	20	57	73	69	55	26	150	0	80	70	10	27	99	14	47	103
2 Incomes	67	26	91	22	84	78	105	51	28	0	184	162	22	91	82	10	1	102	82
Own	93	38	111	35	106	101	130	71	41	80	162	242	0	76	106	52	8	114	128
Rent	32	6	54	7	35	50	44	35	13	70	22	0	92	25	3	57	7	35	57
Married	37	13	51	9	42	50	50	35	16	10	91	76	25	101	0	0	0	53	48
Married with Kids	50	15	44	21	51	37	79	12	18	27	82	106	3	0	109	0	0	48	61
Single	32	15	62	11	40	58	35	57	17	99	10	52	57	0	0	109	0	35	74
Single with Kids	6	1	8	1	8	6	10	2	3	14	1	8	7	0	0	0	15	13	2
Female	58	21	70	17	70	62	83	44	22	47	102	114	35	53	48	35	13	149	0
Male	67	23	95	25	71	89	91	62	32	103	82	128	57	48	61	74	2	0	185

more info see here

MCA Example (2)

Summary table:

Inertia and Chi-Square Decomposition

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent	4 8 12 16 20 ----+----+----+----+----+---
0.56934	0.32415	970.77	18.91	18.91	**********
0.48352	0.23380	700.17	13.64	32.55	*******
0.42716	0.18247	546.45	10.64	43.19	******
0.41215	0.16987	508.73	9.91	53.10	******
0.38773	0.15033	450.22	8.77	61.87	*****
0.38520	0.14838	444.35	8.66	70.52	*****
0.34066	0.11605	347.55	6.77	77.29	****
0.32983	0.10879	325.79	6.35	83.64	****
0.31517	0.09933	297.47	5.79	89.43	***
0.28069	0.07879	235.95	4.60	94.03	***
0.26115	0.06820	204.24	3.98	98.01	**
0.18477	0.03414	102.24	1.99	100.00	*
Total	1.71429	5133.92	100.00		

Degrees of Freedom = 324

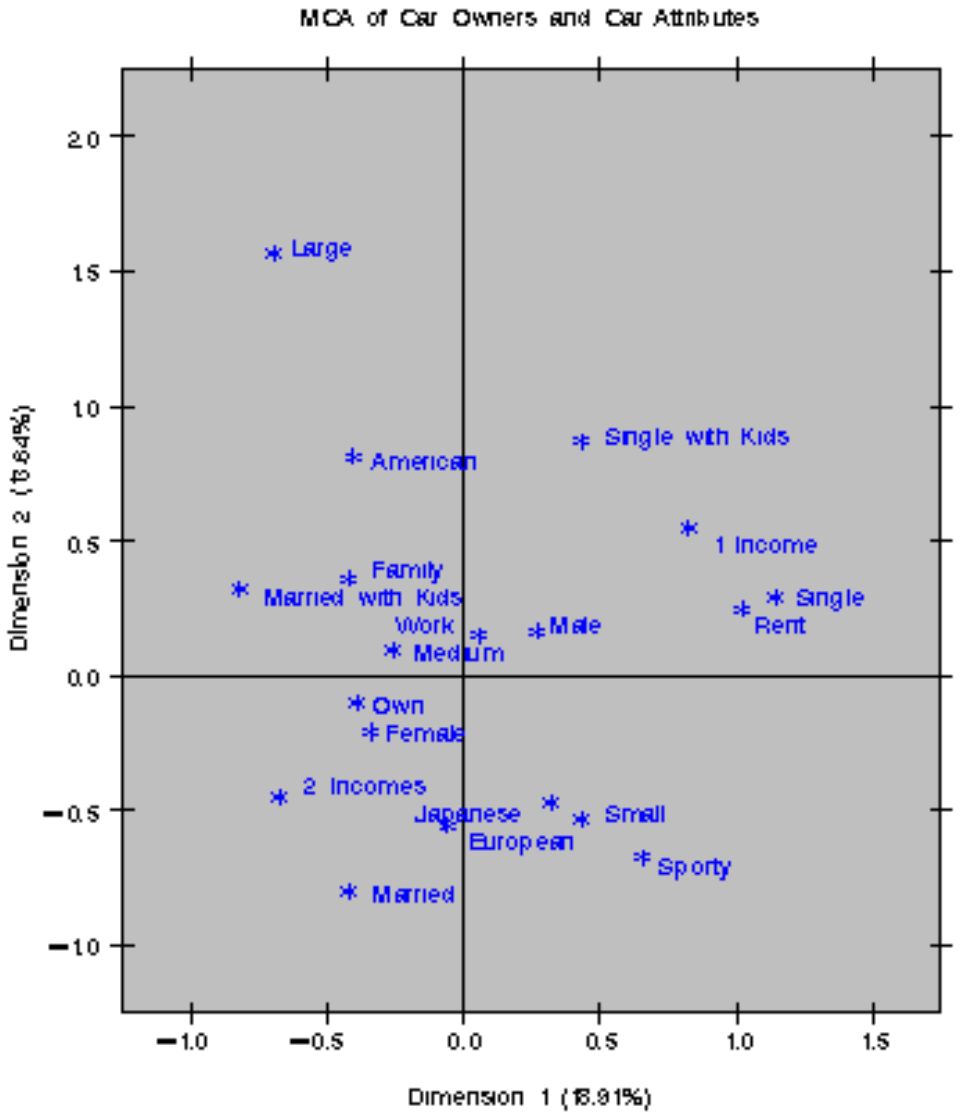
MCA Example (3)

Most influential column points (loadings):

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

MCA Example (4)

Burt table plot:



Plot Observations

Top-right quadrant:

- categories single, single with kids, 1 income, and renting a home are associated

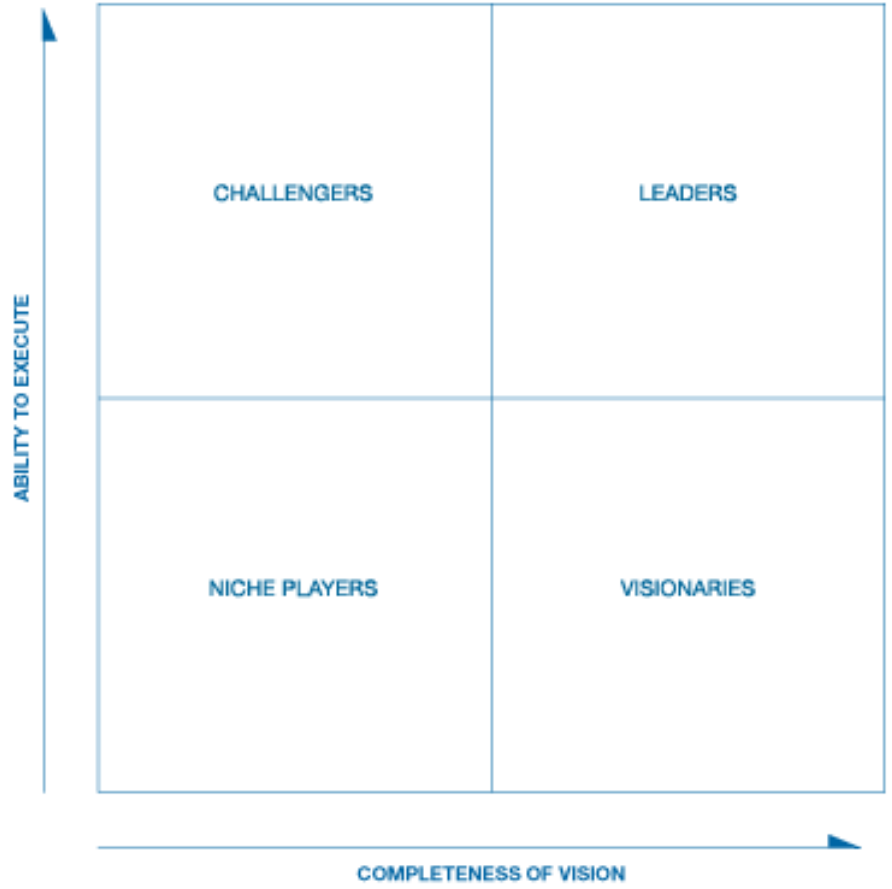
Proceeding clockwise:

- 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

Gartner Magic Quadrant

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

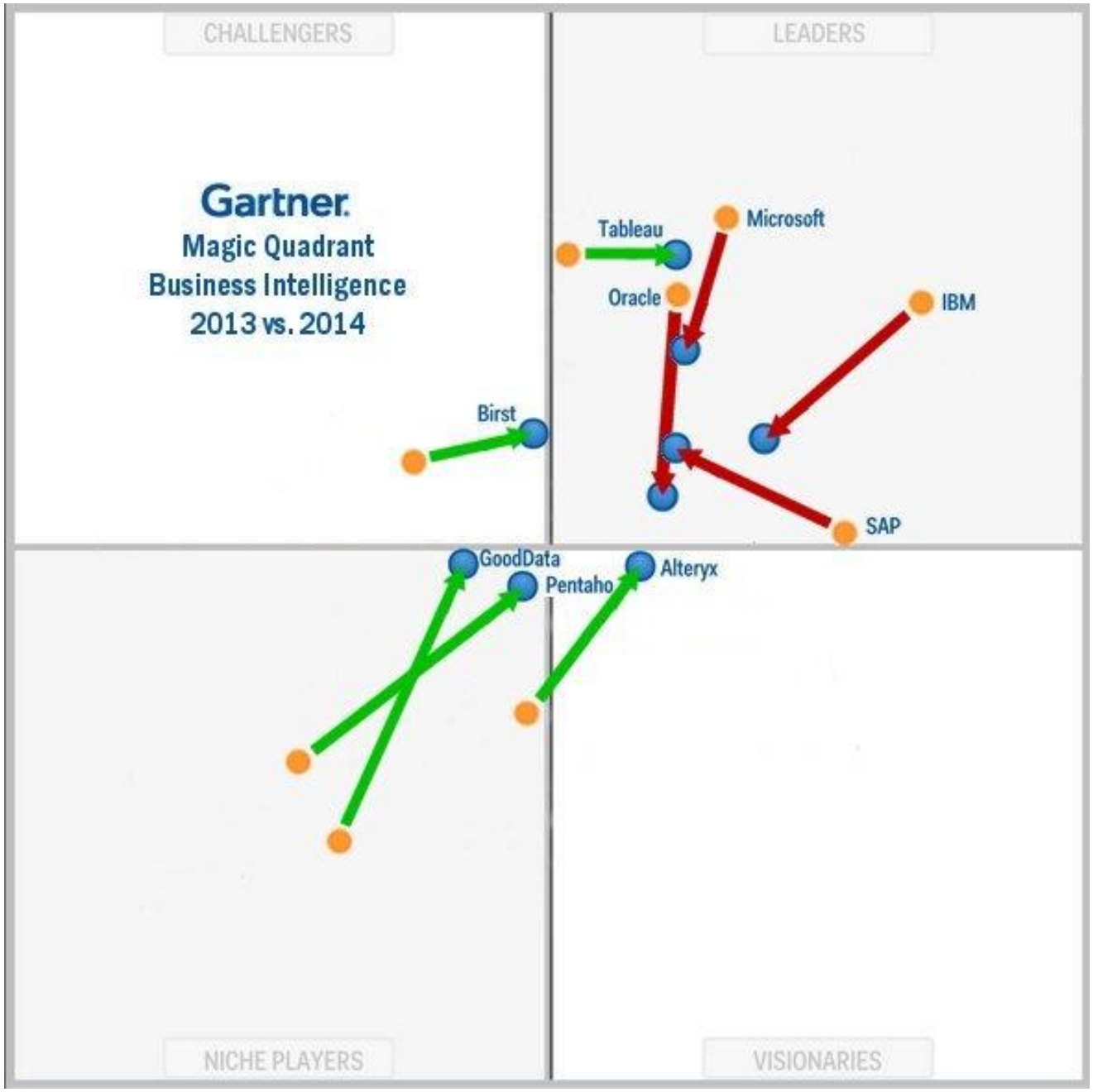


This concept can be used for other dimension pairs as well

- 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)


CHALLENGERS
LEADERS
Gartner
Magic Quadrant
Business Intelligence
2013 vs. 2014
Tableau
Microsoft
Oracle
IBM
Birst
SAP
GoodData
Pentaho
Alteryx
NICHE PLAYERS
VISIONARIES