CSCI E-82
Advanced Machine Learning, Data Mining & Artificial Intelligence
Lecture 2

Peter V. Henstock
Fall 2018

Statistics vs. Machine Learning

- How are they the same?
- How are they different?

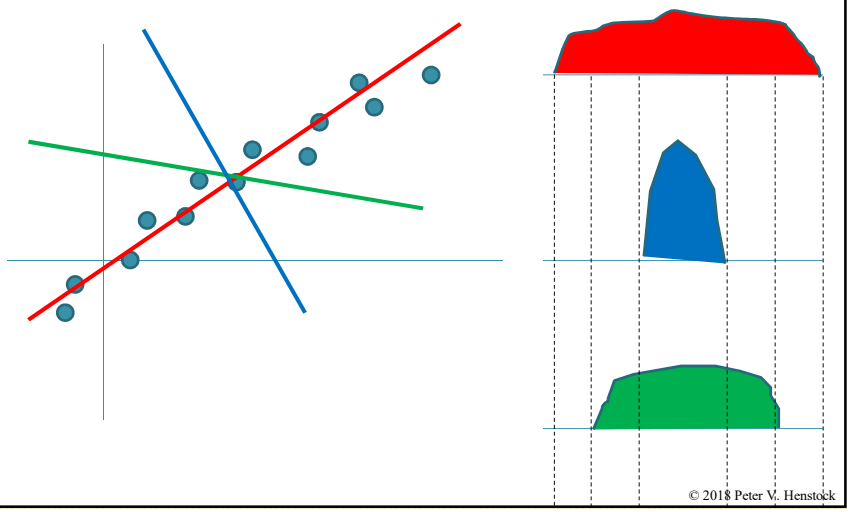
Statistics vs. Machine Learning

- "Data Science without statistics is possible, even desirable"
  - https://www.datasciencecentral.com/profiles/blogs/data-science-without-statistics-is-possible-even-desirable
- Statistics:
  - Make assumptions
  - Derive formulae
  - Assert properties like "significance"
- Machine learning:
  - Set up an optimal criteria
  - Figure out a computational way of solving it

Principal Component Analysis

1st Principal Component Axis

- Another Idea: maximize variance of the distribution projected onto the line



PCA Solution

- Trying to find axes of maximize variance
- New axes represent a "subspace"

- SVD
  - Breaks matrix into USV^T such that
    - S is a diagonal matrix of eigenvalues
    - Columns of U are eigenvectors
  - Eigenvectors of largest eigenvalues are the axes corresponding to maximum variance

Optimization of PCA

- A ~ USV^T where A is covariance matrix
- Minimizing $||A - USV^T||_F^2$
  - Frobenius Norm
  - Technically written $||A - suv^T||_F^2$
  - Unique minimum when $u^Tu=1$, $v^Tv=1$, $s>0$
- Achieve minimum using SVD for different numbers of eigenvalues (or eigenvectors)

Purge Eigenvalues

U

-0.5930	0.3879	-0.0341	-0.7048
-0.4779	-0.8362	-0.2654	-0.0453
-0.4527	-0.0221	0.8286	0.3287
0.4638	-0.3871	0.4918	-0.6271

S

0.1491	0	0	0
0	0.0820	0	0
0	0	0.0496	0
0	0	0	0.0128

V'

-0.5930	-0.4779	-0.4527	0.4638
0.3879	-0.8362	-0.0221	-0.3871
-0.0341	-0.2654	0.8286	0.4918
-0.7048	-0.0453	0.3287	-0.6271

0.0712	0.0165	0.0350	-0.0485
0.0165	0.0949	0.0227	-0.0126
0.0350	0.0227	0.0660	-0.0130
-0.0485	-0.0126	-0.0130	0.0614

Original Cov

0.0648	0.0161	0.0379	-0.0541
0.0161	0.0949	0.0229	-0.0130
0.0379	0.0229	0.0646	-0.0104
-0.0541	-0.0130	-0.0104	0.0563

Cov Approximation 1 eigenvalues

0.0648	0.0156	0.0393	-0.0533
0.0156	0.0914	0.0338	-0.0065
0.0393	0.0338	0.0306	-0.0306
-0.0533	-0.0065	-0.0306	0.0444

Cov with 2 eigenvalues

0.0524	0.0422	0.0400	-0.0410
0.0422	0.0340	0.0322	-0.0330
0.0400	0.0322	0.0305	-0.0313
-0.0410	-0.0330	-0.0313	0.0321

Cov with 3 eigenvalues

Process Check

X*U[,1:k] → projection
If k = 2, what dimensionality will we have?

X vectors with 4 features

0.0344	0.4387	0.3816	0.7655
0.7952	0.1869	0.4898	0.4456
0.6463	0.7094	0.7547	0.2760
0.6797	0.6551	0.1626	0.1190
0.4984	0.9597	0.3404	0.5853
0.2238	0.7513	0.2551	0.5060
0.6991	0.8909	0.9593	0.5472
0.1386	0.1493	0.2575	0.8407
0.2543	0.8143	0.2435	0.9293
0.3500	0.1966	0.2511	0.6160

SVD of (cov(A)) → U,S,V
U columns are eigenvectors

-0.5930	0.3879	-0.0341	-0.7048
-0.4779	-0.8362	-0.2654	-0.0453
-0.4527	-0.0221	0.8286	0.3287
0.4638	-0.3871	0.4918	-0.6271

X matrix 10 x 4 * U matrix is 4x2 after reduction

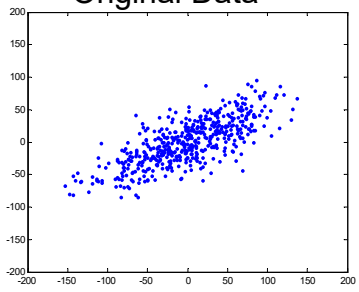
Principal Components

- Create a weighted sum of features
- Use all the features together in one shot
- Each feature' = $\Sigma_{featurej}$ weight(j) *feature(j)
- Weights are not equal
  - First several are much stronger than rest
  - Eliminate the lowest ranked transformations

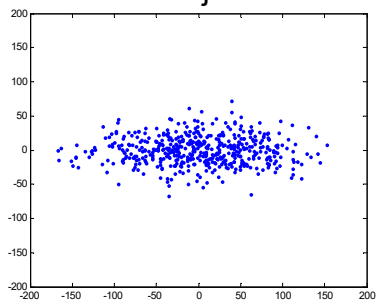
Projection

- Generated a random blob and rotated it by $\pi/6$ (Left)
- 1st PC X-axis; 2nd PC is Y-axis
- Dominant PC axis captures the variation

Original Data



PCA Projection



PCA for original feature selection?

- We have lots of features
- We can use a PCA to map to a lower dimensional space

- Can we use PCA for feature selection?
- Can we select high-value features?
- Can we purge low-value features?

PCA for original feature selection?

- We have lots of features
- We can use a PCA to map to a lower dimensional space
- Can we infer which of the original features are less meaningful?
- PCA1 = w11*feat1 + w12*feat2 + …
- PCA2 = w21*feat1 + w22*feat2 + ..
- PCA3 = w31*feat1 + w32*feat2 + …

Latent Semantic Indexing

- Document matrix: documents x words
- Consider query dot-product doc = similarity measure
- Use SVD to create low rank space
- Project query into PC space
- Assumption that need reduced space
- Document $\text{Query}_k = \Sigma_k U_k^T \text{query}$
  - Find similar documents
- Term $\text{Query}_k = \text{query}^T \Sigma_k V_k^T$
  - Find similar terms

What about outliers?

- Using mean and covariance
- Both are sensitive to outliers
- PCA focuses on preserves pairwise distances—particularly large ones

Robust PCA

- Instead of using mean and variance, it likely uses median and robust standard deviation
- Penalizes the PCA in a way that avoids the huge quadratic cost for outliers
- More complicated solution

Non-Negative Matrix Factorization

- Same idea as PCA but all weights >= 0
- Usually uses a different factorization
  - X = WH with same $||X-WH||_F^2$
- Face = FacialFeatures * FeatureWeight
- Document = Topic * TopicImportance
- Image = MemberSignal*MemberAbundance
- GeneExp = Specimens*SpecimenWeights
- BrainMeasurements = NeuroProc*Weights
- NP-Hard problem so no SVD

Independent Component Analysis

- Unsupervised learning
- $X_1 \ldots X_N$ are hidden variables
- $Y_1 \ldots Y_k$ are the observables (k >= N)
  - Y are independent Info(Yi, Yj) = 0
  - Maximize Y to X so Info(Yi,X) is large
  - Y is linear function of weighted X so Y = AX
- "Cocktail Party"
- "Blind Source Separation"
- Breaks world down into components
  - PCA of faces finds average faces
  - ICA of faces finds noses, ears, hair

Independent Component Analysis

- Two sources of ambiguity
  - Can't figure out which Xi is which
  - Can't figure out if it's Xi or –Xi
- Algorithm will not work if data is Gaussian
- Math is based on computing cumulative density distributions for the X

- 1967 Western
- Set in civil war
- Name has become an idiom

Good, Bad & Ugly of PCA

- Good?
- Bad?
- Ugly?

Good, Bad & Ugly of PCA

- Good
  - Single fast global transform—no parameters
  - Widely used so requires less explanation
  - Select the extent of the approximation (#eig)
- Bad
  - Often don't get the clear delineations since it is a global transform
- Ugly
  - Linear transform which sometimes doesn't do much beyond some kind of rotation depending on data
  - Ignores probabilistic distribution: variance only

Alternative Methods for Dimensionality Reduction

Multidimensional Scaling

- PCA is linear technique
- Linear transform of the points (rotation)
- Similar approach that is a linear transform

- Goal:
  - Map higher dimensional → lower dimensional
  - Find projection that preserves the pairwise distances in lower dimensional space

MDS

- Optimization function
  - $E(y) = \sum_{i \neq j} \frac{(dy_{ij} - dx_{ij})^2}{dy_{ij}}$
  - dy = pairwise difference of projection
  - dx = pairwise difference of original high dimensional points
- Note that researchers have come up with lots of different stress functions
- How to optimize this?

MDS

- Optimization function or “stress function”
  - $E(y) = \sum_{i \neq j} \frac{(dy_{ij} - dx_{ij})^2}{dy_{ij}}$
  - dy = pairwise difference of projection
  - dx = pairwise difference of original high dimensional points
- Note that researchers have come up with lots of different stress functions
- How to optimize this?

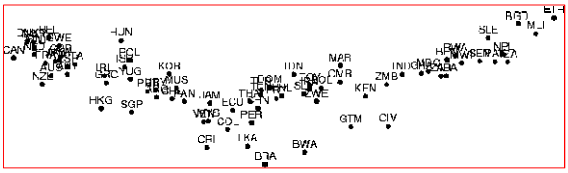
MDS

- Optimization function or "stress function"
  - $E(y) = \sum_{i \neq j} \frac{(dy_{ij} - dx_{ij})^2}{dy_{ij}}$
  - dy = pairwise difference of projection
  - dx = pairwise difference of original high dimensional points
- Note that researchers have come up with lots of different stress functions
- How to optimize this?
  - Gradient descent

Sammon's Algorithm

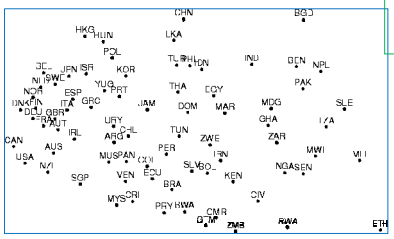
- Optimization function
  - $E(y) = \sum_{i \neq j} \frac{(dy_{ij} - dx_{ij})^2}{dx_{ij}}$ ⬅
  - dy = pairwise difference of projection
  - dx = pairwise difference between original high dimensional points
- MDS reminder
  - $E(y) = \sum_{i \neq j} \frac{(dy_{ij} - dx_{ij})^2}{dy_{ij}}$ ⬅

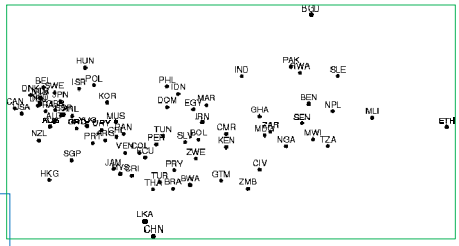
World Bank Data Comparison



PCA

Sammons
Preserves small distances



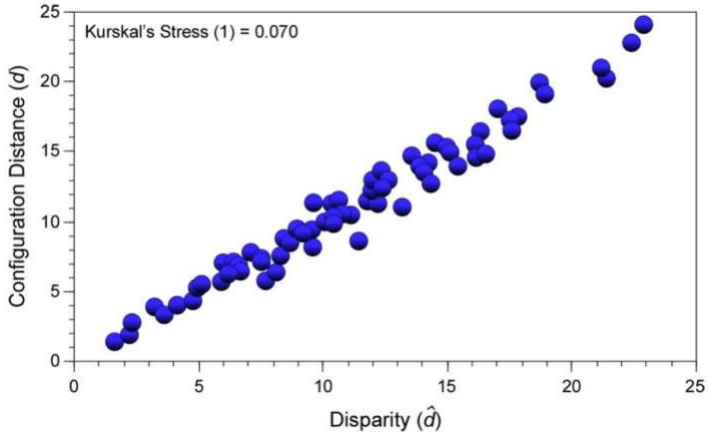


MDS

How would you assess these?

How would you assess these?

Shepard Diagram



- http://www.palass.org/modules.php?name=palaeo_math&page=20

Where do these fail?

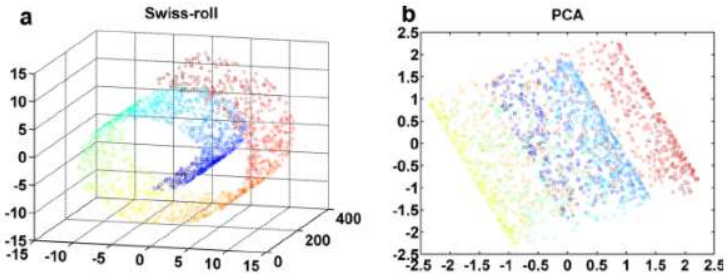
- PCA is linear technique
  - Linear transform of the points (rotation)
  - Removes noise on hyperplane
- MDS/Sammons focus on pairwise distances in space
- What if data doesn't follow a hyperplane?
  - Curve
  - Manifold
  - ?

Computational Speed?

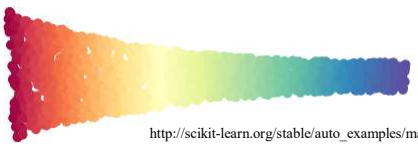
- $O(N^2)$
- Typically used for up to 1000s of points

- Computational speed is an issue
- Optimizing 1,000,000 pairwise distances is challenging

Nonlinear manifolds



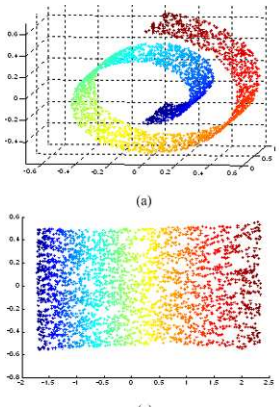
Luo, Lijia. (2014). Process Monitoring with Global–Local Preserving Projections. Industrial & Engineering Chemistry Research. 53. 7696–7705.



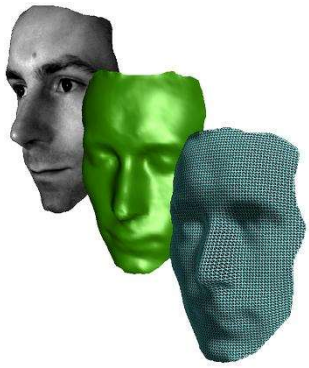
http://scikit-learn.org/stable/auto_examples/manifold/plot_swissroll.html

Isomap

- Tenenbaum, de Silva and Langford
- Science 290 (5500): 2319-2323, 22 Dec. 2000
- What if have nonlinear manifold?



(a)



https://www-users.cs.york.ac.uk/~nep/research/3Dface/tomh/

Isomap

- How do you construct distances?

Isomap

- How do you construct distances?

- Same process as the connectivity for the clustering
- Construct neighborhood graph
- Draw edge between all pairs of points with distance less than threshold
- Compute shortest distances along graph between all pairs of points
- Apply MDS

Floyd-Warshall Algorithm

- Computes shortest paths in [weighted] graphs with + and – edges
- Will not work for negative cycles
- $O(V^3)$

- Vertices 1 to N
- Trying to find shortest path from i to j
- Only use k vertices at a point in time
- Goal: find shortest path for k+1 vertices

Floyd-Warshall contd.

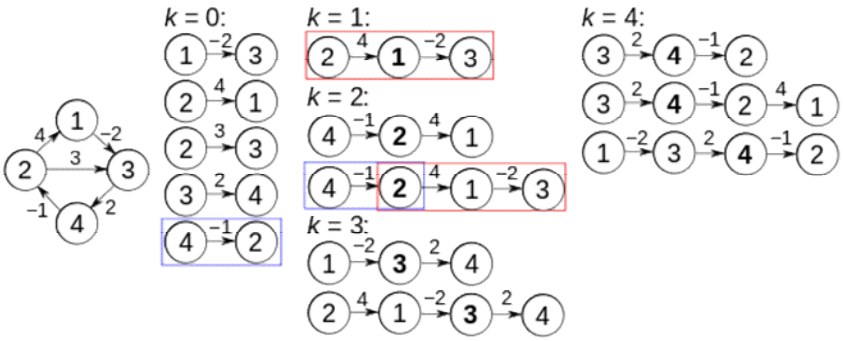
- For each pair of vertices i and j
- Two possibilities of shortest path

1) Uses vertices 1 to k
2) Using path from i →k+1→j

Algorithm:
initialize with edge weights
for k = 1 to |V|
  for each pair i,j
    dist(i,j)= min[dist(i,j), dist(i,k)+dist(k,j)]

Example from Wikipedia

- K = 0 is original edges
- K = 1 is paths through node {1}
- K = 2 is path through {1,2}



Local Linear Embedding

- Nonlinear Dimensionality Reduction by Locally Linear Embedding, Roweis & Saul 2000
- Embedding = mapping to lower dimensional space
- Motivation:
  - Most methods fail on nonlinear spaces
  - Take patches of local geometries together
  - Map patches to plane to preserve global geometry
- Assumptions:
  - Continuous manifold without holes and not noisy data
  - Other versions overcome these
- Approach:
  - Compute KNN of each point Xi (K=only parameter)
  - Minimize $\Sigma_i[X_i - \Sigma_j W_{ij} X_j]^2$ $W_{ij}>0$ for neighbors j; ensure $\Sigma_j W_{ij}=1$
  - Create $\Phi(Y) = \Sigma_i[Y_i - \Sigma_j W_{ij} Y_j]^2$ s.t. $\Sigma_i Y_i=0$ and $\Sigma_i Y_i Y_i^T=NI$

Local Linear Embedding

- Optimization of Y is equivalent to finding eigenvectors
- Space: NxK weight matrix
- Time: $O(DN^2)$ for D input dimensions
  - $O(DN^2)$ for KNN worst case
  - $O(DNK^3)$ to obtain W weight matrix
  - $O(DN^2)$ for the Y embedding

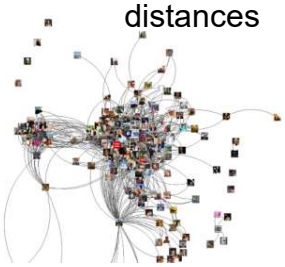
T-SNE

"Dandelions" & "Train Wrecks"

Preserves local distances

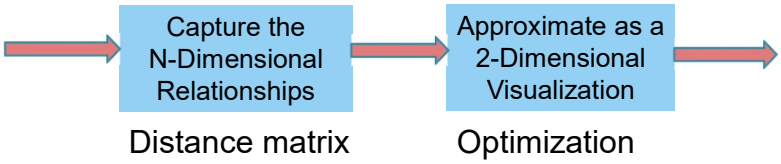
Preserves global distances

Facebook Connections

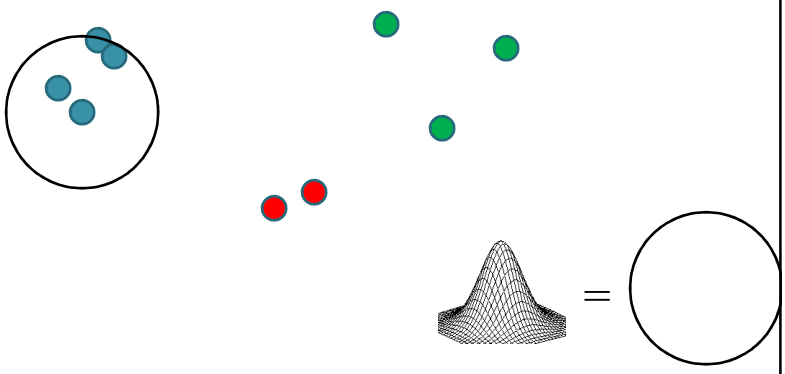


kipedia.org/wiki/Taraxacum
.roundsconsulting.com/2016/10/12/choosing-train-wreck-dumpster-fire/
.net/2014/04/facebook-data-collection-and-photo-network-visualization-with-gephi-and-r/#iLightbox[79713ef84a664b9d4dd]/0
http://kateto.net/2014/04/facebook-data-collection-and-photo-network-visualization-with-gephi-and-r/#iLightbox[d4dd84cf3af192994c5]/0

How t-SNE class of solutions works

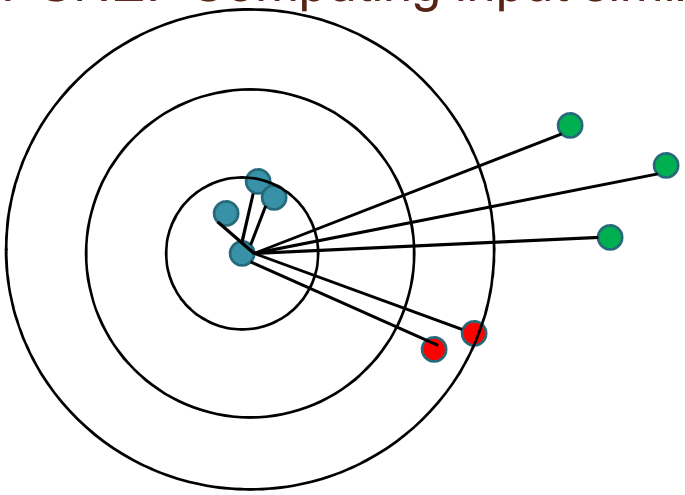


T-SNE Computing output similarity



- Compute square similarity matrix
- Similarities defined on Gaussian
- 1.0 is perfect match and 0.0 is distant

T-SNE: Computing input similarity



- After computing similarities
- Normalize scores to sum of all similarities
- Why? Balance spread on clusters

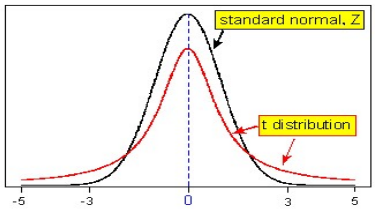
About this Gaussian variance

- Assumed that we had this magical distribution on the blue cluster
- Relates to the perplexity parameter
- Optimal calculation based on neighborhood density of each point
- Are all densities the same? No!
  - Sim(blue j, red k) != Sim(red k, blue j)
  - So…we average them
  - p(j,k) = Sim(j,k) / 2 + Sim(k,j) / 2

T-SNE Computing output similarity

- Trying to reproduce the distances on the output of course since want similar things to be close
- Setting up the same idea of similarity

- But to compute "q" output distance:
  - Use t-distribution instead of Normal



Now the optimization

- We can compute the input similarity
- We can compute the output similarity
- Now we need something to optimize

- Kullback-Leibler distance
  - KL(P|Q) = $\sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$
  - KL is based on an entropy based equation
  - Do we want this small or large?
- If p is similar then q should be….
- If p is distant then q should be …

Gradient Descent

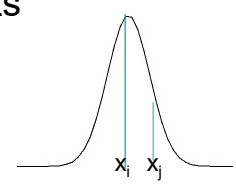
- What is it?

Gradient Descent

- What is it?
- How do we start?

t-SNE

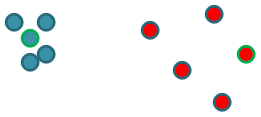
- T-Dist. Stochastic Neighbor Embedding
- Strategy:
  - Focus on local neighborhoods
  - Use normalized Gaussian probabilities of similarity between local points
  - $p_{ij} = \frac{\exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})}{\sum_k \sum_{l \neq k} \exp(-\frac{\|x_k - x_l\|^2}{2\sigma^2})}$



  - Note that the variance σ^2 is global here but that's not good enough

t-SNE

- $p_{ij} = \frac{\exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})}{\sum_k \sum_{l \neq k} \exp(-\frac{\|x_k - x_l\|^2}{2\sigma^2})}$
- Replace global variance with local variance
- $p_{j|i} = \frac{\exp(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2})}{\sum_{k \neq i} \exp(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2})}$



- $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$ kluge to restore symmetry
- $q_{ij} = \frac{(1+\|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1+\|y_k - y_l\|^2)^{-1}}$ in low dim space
- Gradient descent optimize Kullback-Leibler distance KL(P|Q) = $\sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

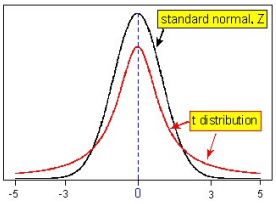
P & Q have different similarities?

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

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- $q_{ij} = \frac{(1+\|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1+\|y_k - y_l\|^2)^{-1}}$: Student-T

$$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$



http://election.princeton.edu/2014/09/18/how-our-predictions-work-continued/

- Heavy tails allow for distant points to be further apart as will usually happen

Where is the perplexity parameter?

- One of the more cryptic parts of t-SNE
- It's buried within the σ_i
- Create probability distribution P_i for given σ_i
- Perplexity(P_i) = $2^{-\sum_j p_{j|i} log_2(p_{j|i})}$
- Exponent part is the entropy of $p_{j|i}$
- t-SNE searches for a σ_i that achieves perplexity requested by the user

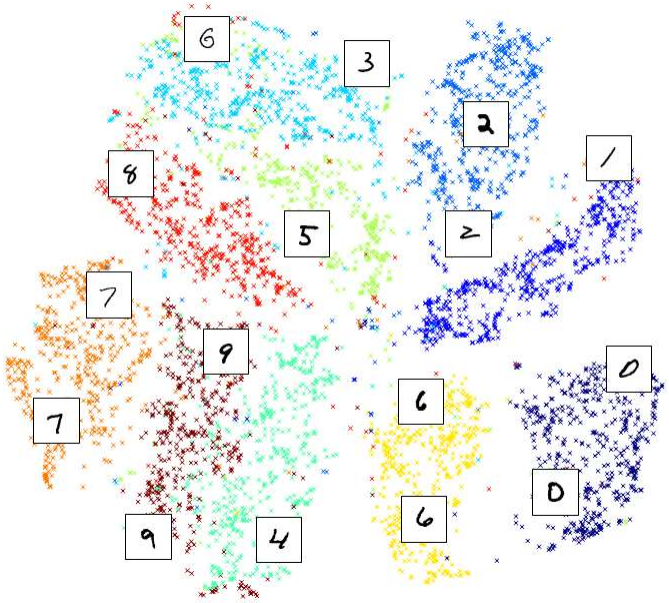
Comparison on MNIST

- http://colah.github.io/posts/2014-10-Visualizing-MNIST/
- 10K or 60K handwritten numbers

PCA MDS Sammon's

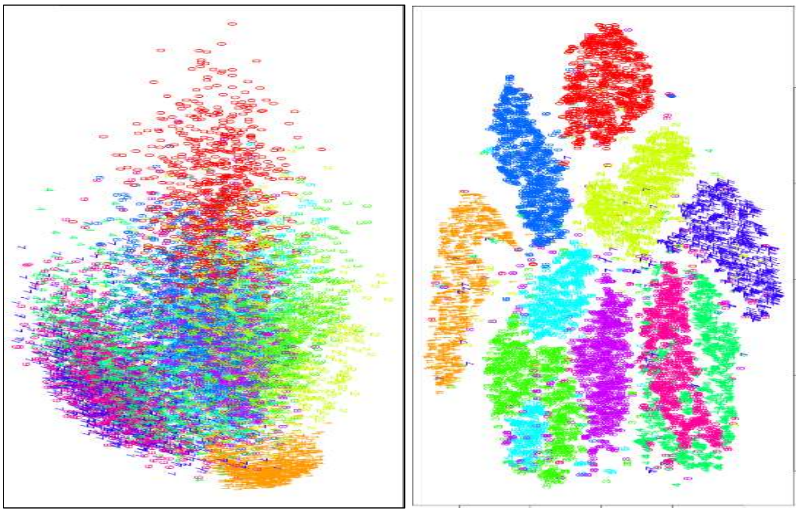
T-SNE on MNIST from

- http://alexanderfabisch.github.io/t-sne-in-scikit-learn.html



MNIST Handwritten Digits Data Set

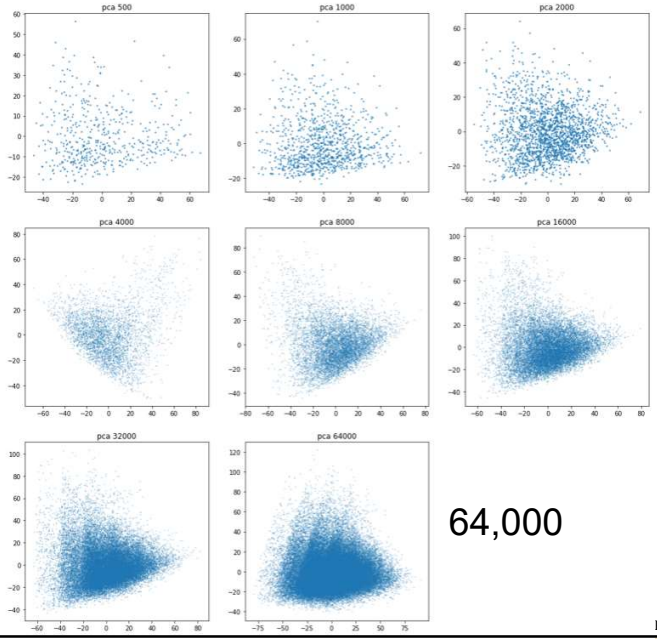
PCA t-SNE



https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/

Ugi Data Set using PCA

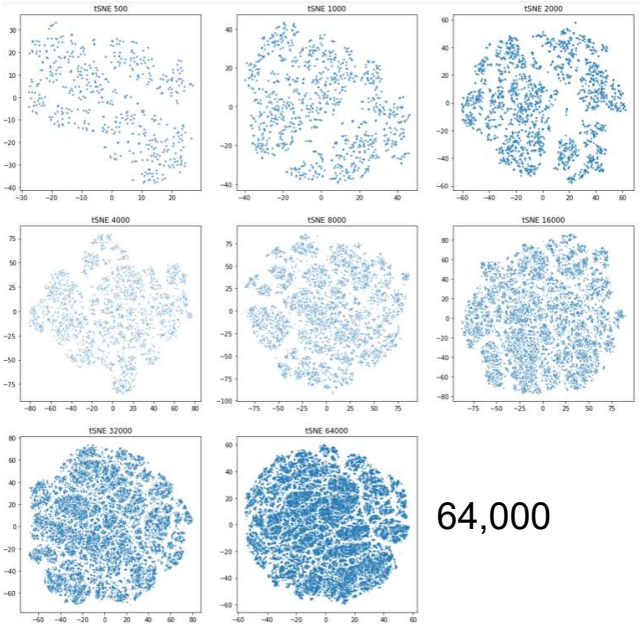
500



64,000

Ugi Data Set using t-SNE

500



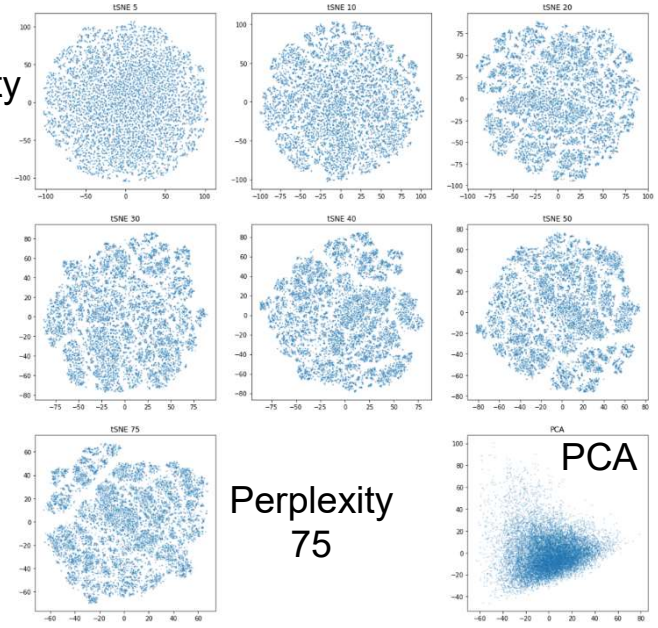
64,000

62

Peter V. Henstock

Ugi Series Data – t-SNE Perplexity

Perplexity 5

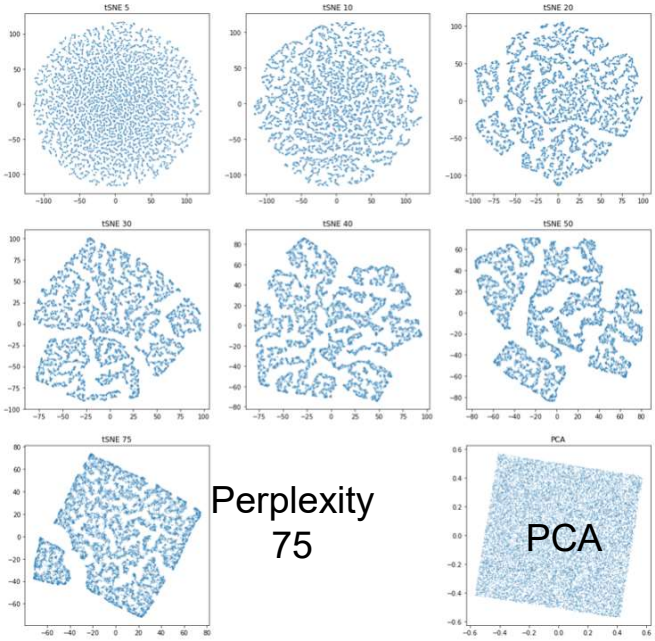


Perplexity 75

PCA

Random Data – t-SNE Perplexity

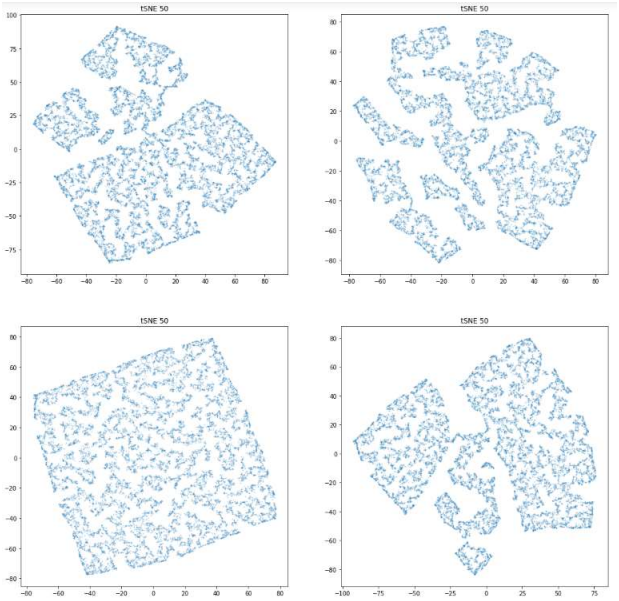
Perplexity 5



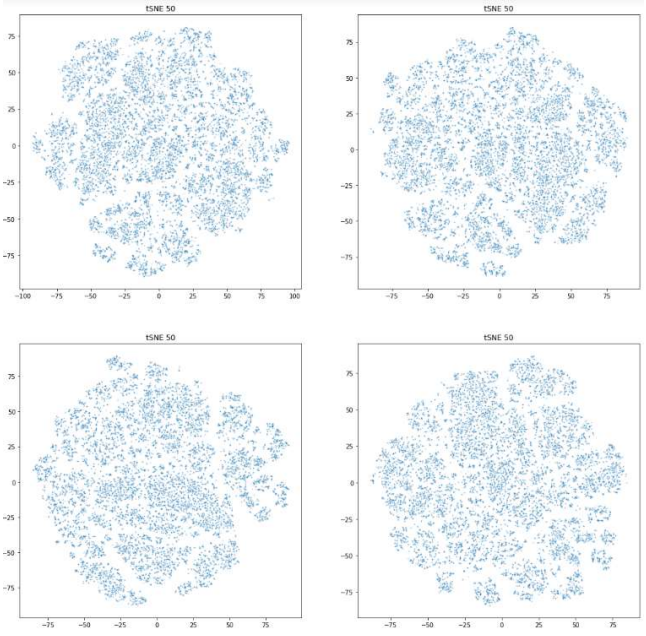
Perplexity 75

PCA

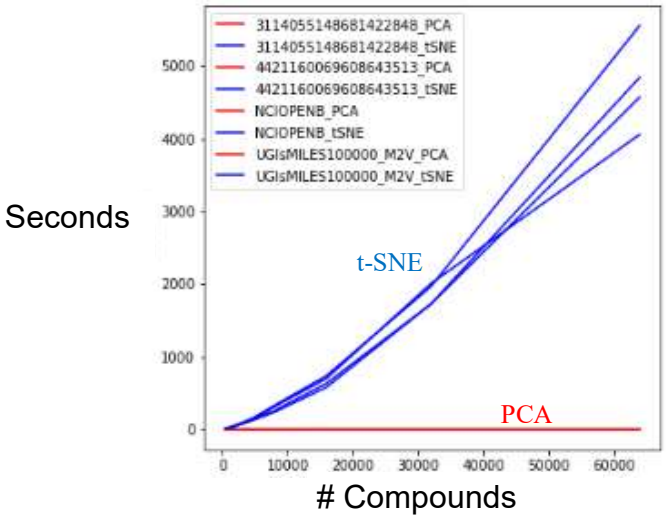
Repeatability for t-SNE (Random)



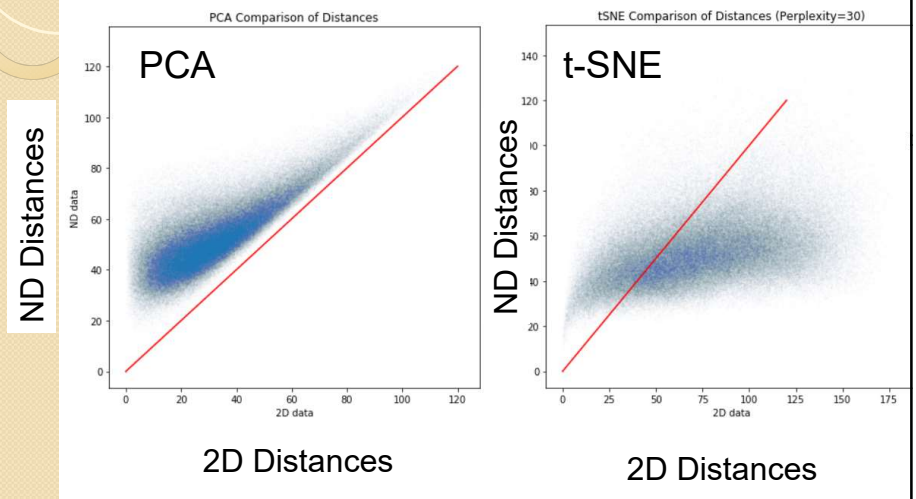
Repeatability for t-SNE (Ugi)



Computational Speed t-SNE vs PCA



Comparing Distances: ND vs. 2D

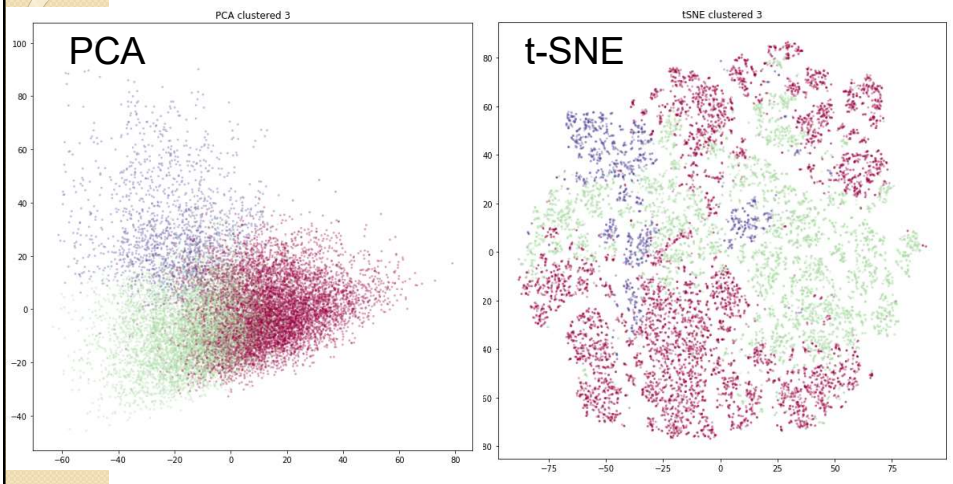


ND Distances

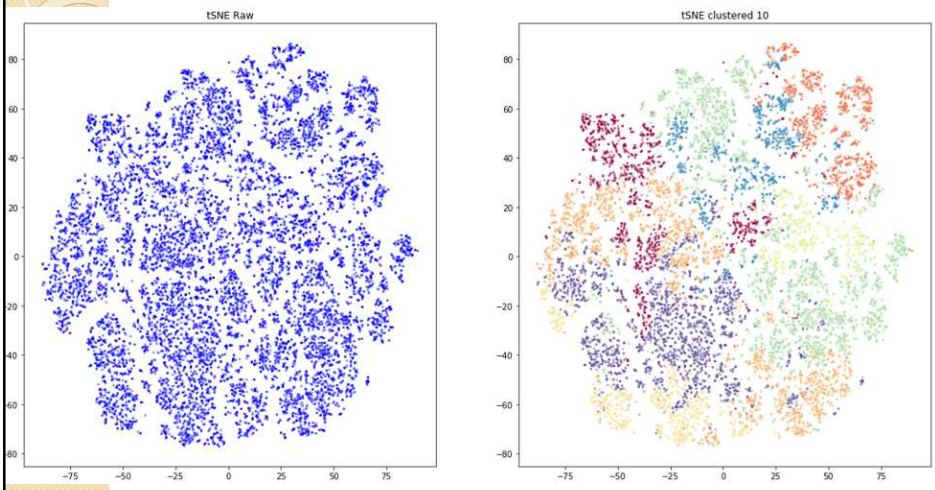
2D Distances

2D Distances

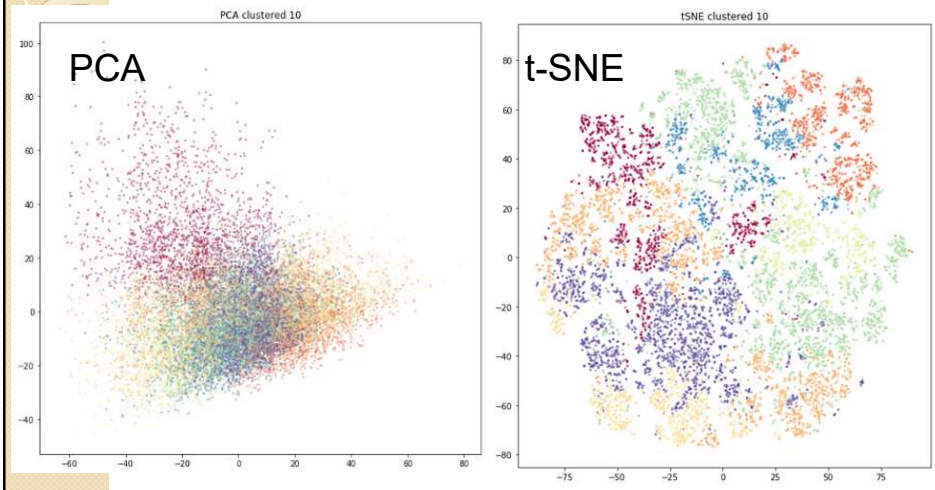
Clustering Data with 3 clusters



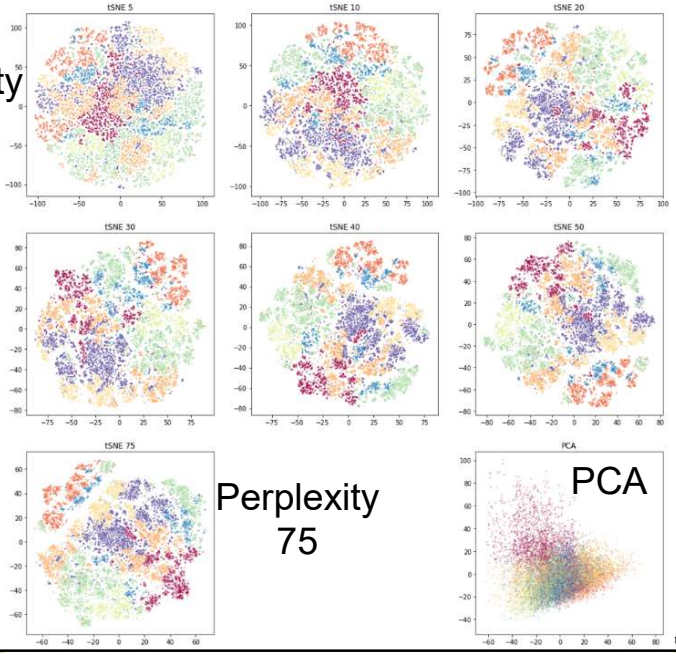
t-SNE with10 clusters



10 clusters



Can Perplexity Solve the Problem?

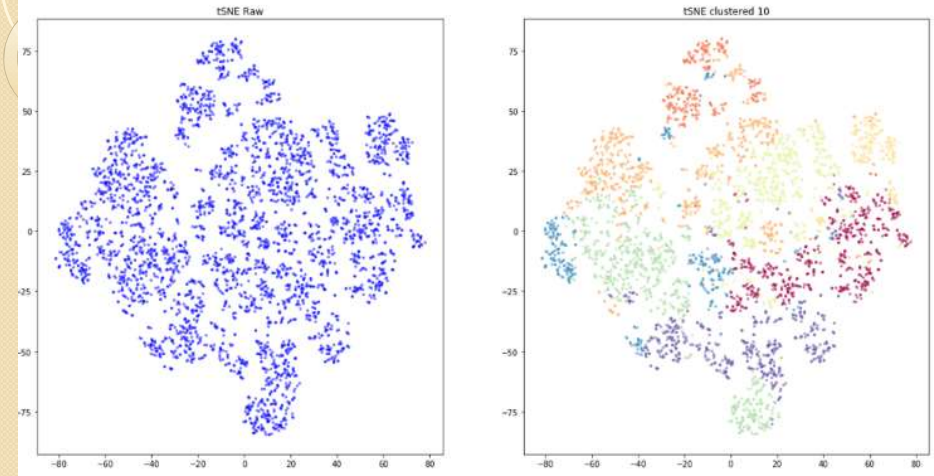


Perplexity 5

Perplexity 75

PCA

Is Scale the Problem: t-SNE with 4K

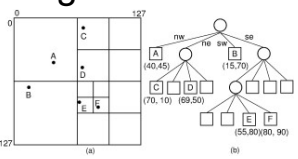


Previously looking at 16K compounds

Barnes-Hut

- Computational time for t-SNE
  - Compute the distance matrix so N^2
  - If have 1 million points → zzz
- Barnes-Hut approximation:

  - Distance between red and blue are all fairly similar so approximate them together
  - Implementation is quadtree



  - Run time?

LargeVis

- "Visualizing Large-Scale and High-Dimensional Data" Tang et al. 2016
- Approach is O(sMN) s=2,3 M=#neg samp
  - Computes k-NN graph & maps
  - Layouts graph to lower dimensional space

From Tang et al.

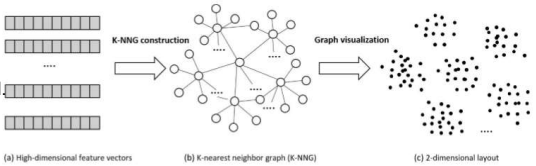


Figure 1: A typical pipeline of data visualization by first constructing a K-nearest neighbor graph and then projecting the graph into a low-dimensional space.

- Speed 30x > t-SNE in KNN, 7x for layout
- Better data structures & fewer parameters

Fast KNN graph

- Typical is too slow $O(N^2d)$
- Builds tree by randomly selecting points and dividing up space evenly
- Points in same node: NN candidates
- Standard for high accuracy: many trees
  - LargeVis avoids bottleneck: uses a few trees
  - Takes neighbors of neighbors for KNN
- Uses same t-SNE equation for pij

Embedding

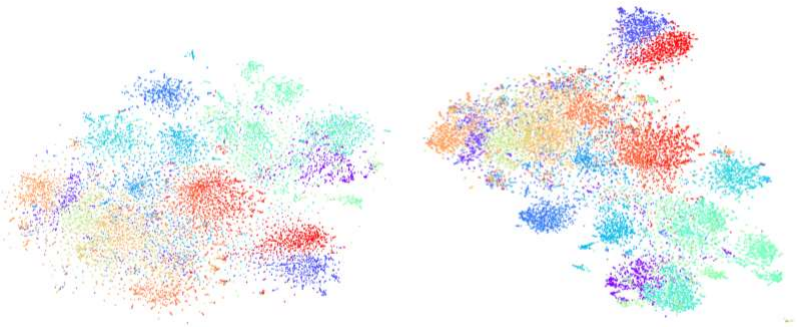
- P(edge from i to j) = $P(e_{ij}=1)=f(||y_i-y_j||)$
  - High when edges are close
- P(weighted edge i-j)= $P(e_{ij}=w_{ij})= P(e_{ij}=1)^{wij}$
- P(graph) = $\Pi\ P(e_{ij}=1)^{wij}\ \Pi\ (1-P(e_{ij}=1))^{\gamma}$
  - γ = weight for negative (i.e. no) edges
  - $O(N^2)$ operation so avoid this

$$\sum_{ij\in E} w_{ij} \log p(e_{ij} is\ 1) + \sum_{k=1}^{M} E_{jk\sim P_n(j)}\ \gamma \log(1 - p(e_{ij} = 1))$$

  - Randomly sample vertices and estimate Pn(j) as negative edges
- Asynchronous gradient descent on samples

T-SNE vs. LargeVis

- 20K documents from 20 different newsgroups
- From LargeVis paper



(a) 20NG (t-SNE)

(b) 20NG (LargeVis)

Good, Bad & Ugly of t-SNE

- Good?
- Bad?
- Ugly?

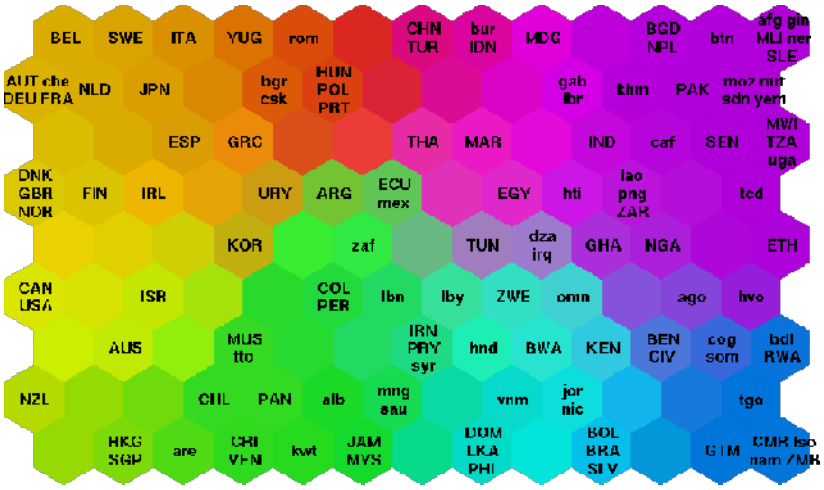
Good, Bad & Ugly of t-SNE

- Good
  - Currently best in separating high dimensional classes
  - Gaining traction as a valuable tool
- Bad
  - Stochastic part → often can't get same output
  - Not a perfect clustering even without Barnes-Hut
- Ugly
  - Perplexity over perplexity: artifacts and tweaking
  - Control allowed with balance of local vs. global

An Introduction to

Self-Organizing Maps

Self-Organizing Map



- World Poverty Map
- http://www.cis.hut.fi/research/som-research/worldmap.html

Temporal SOM for HIV

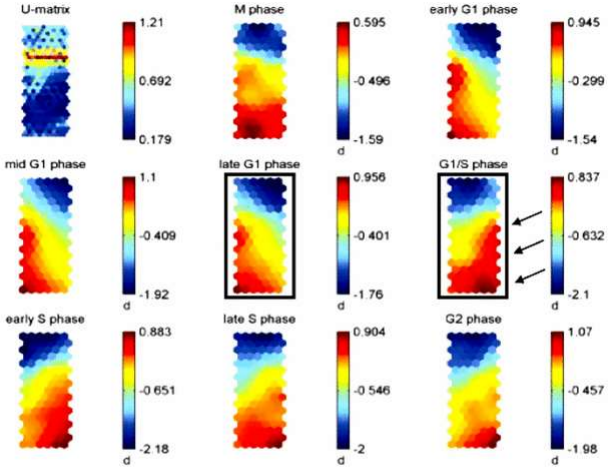


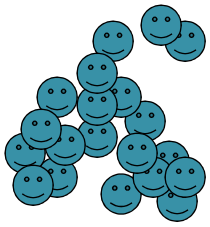
Figure 4
Temporal SOM analysis of HIV-1 Tat-induced cellular genes in synchronized Tat cells. 3 separate filters were applied to remove genes that did not display at least a 1.5, 2, or 3-fold change at each time point analyzed in the 16 eTat chips (see Methods); each filter produced a discrete dataset that was applied to SOM analysis. The third and most restrictive dataset is shown here. Genes that were significantly up (red) and down-regulated (blue) are shown. The U-matrix identifies which genes are similar to each other in terms of expression profile (blue) separated by a "boundary" (red). This SOM graph contains 17 rows and 6 columns of neurons, represented as coordinates. The arrows adjacent to the G_2/S SOM indicate those genes significantly up-regulated during this transition and S phase, and those that show decreased expression in the G_1 phase.

- Liang et al. Therapeutic target for HIV-1 infection in the host proteome Retrovirology 2005.

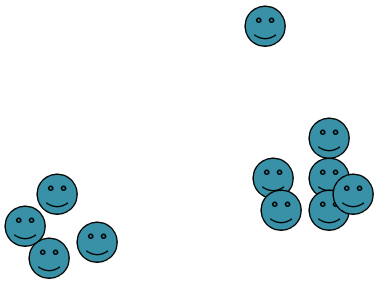
Competitive Learning & Clustering

- Class of clustering algorithms based around neural networks
- Notion is that clusters compete to represent a given data point
- Winning cluster definition is updated or "learns" to better represent that point in the future
- Examples:
  - Adaptive Resonance Theory
    - (Grossman @Boston Univ)
  - Self-Organizing Map
    - (Kohonen @Helsinki Univ., Finland)

Visual Competitive Learning



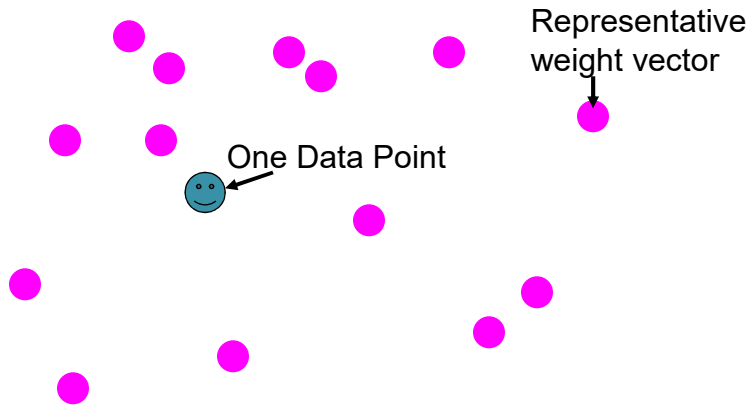
We start with a data set...



- Each  is a data point.
  - Here, they are points in a 2D space.
- We want to find the best representatives.

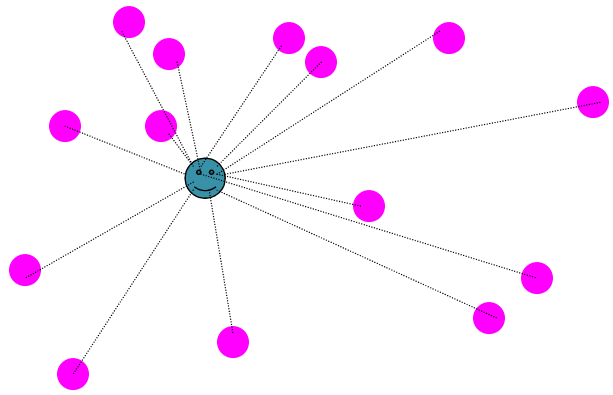
Visual Competitive Learning

We guess some representatives...



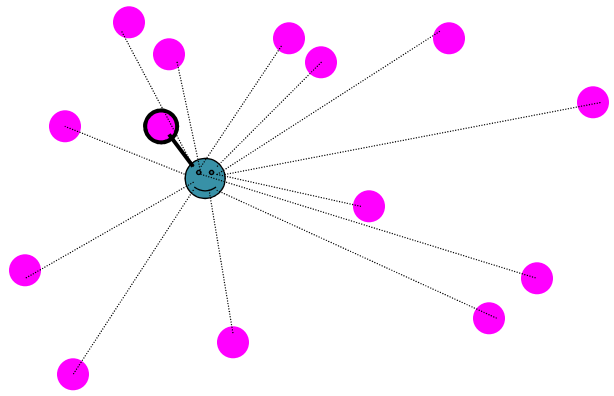
- Each data point is processed sequentially, independent from other data points

Visual Competitive Learning



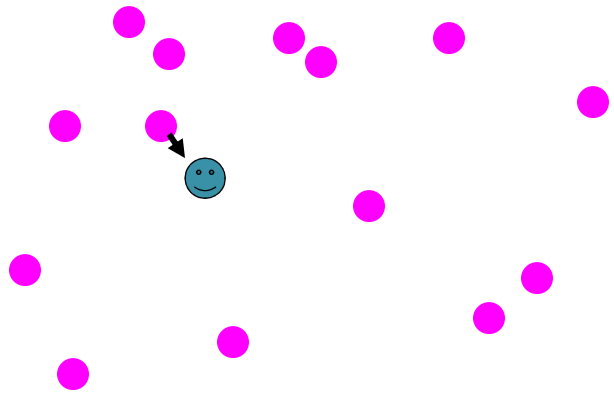
- Calculate distances to all weight/reps

Visual Competitive Learning



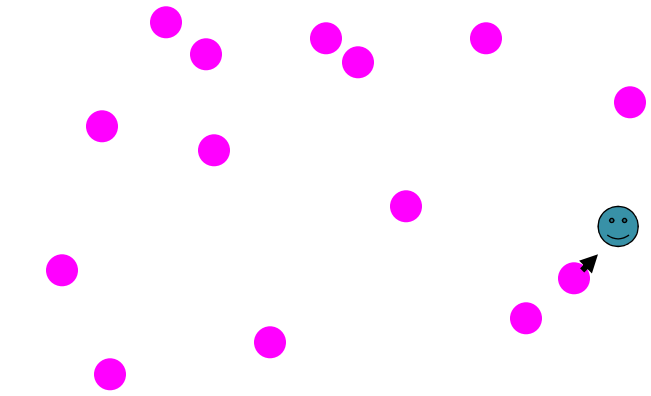
- Find the closest weight
  - = best representative

Visual Competitive Learning



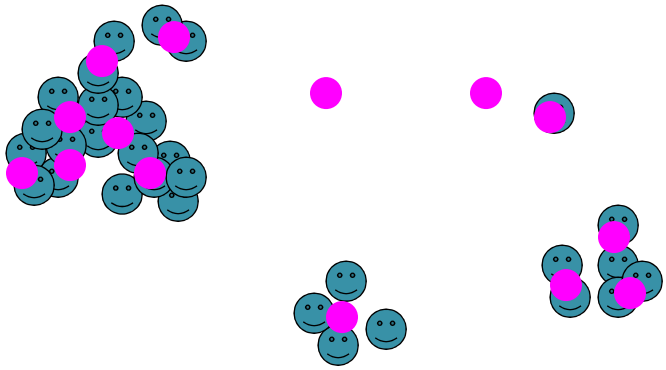
- Move weight α distance toward data point

Visual Competitive Learning



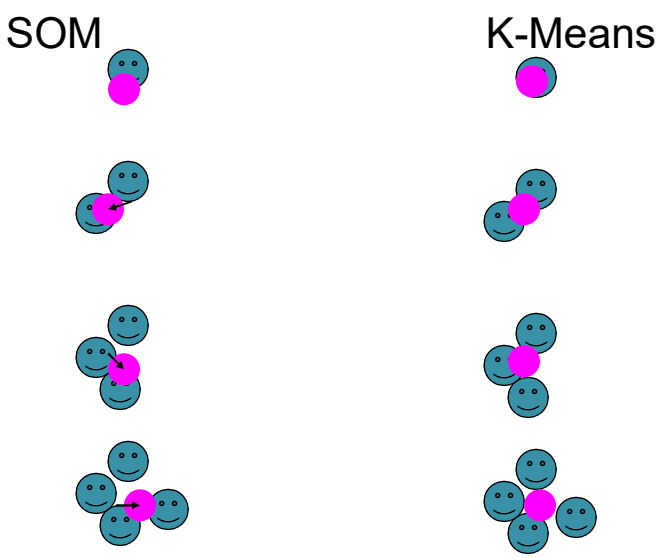
- Repeat for next data point
- Repeat for N epochs over set of all data points

Clustering with Competitive Learning



- Weights migrate to represent clusters
  - More data points—more representatives
- Some weights might be loners

Comparison with K-Means



Maintaining a Topology

- Up until now, we have explained competitive learning as a clustering tool
- Very similar to K-means or other approaches

- Why are SOMs different?
  - Ability to represent N-dimensional data in 2D
  - Similar areas of the high dimensional space map to similar areas of the 2D space

Maintaining a Topology

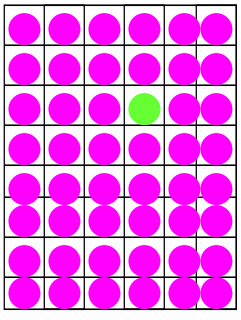
- Create a 2D array of weight vectors (representatives)
- Initialize each representative to a random location in the compound space
- For i=1 to E epochs {
  - For J = 1 to D data points {
    - Find closest representative to dataPoint[J]
    - Move representative a distance α towards dataPoint[J]
    - Move the K neighbors of the closest representative *in 2D array* a distance α in direction of dataPoint[J]

  }

  Reduce learning rate and neighborhood.
- }

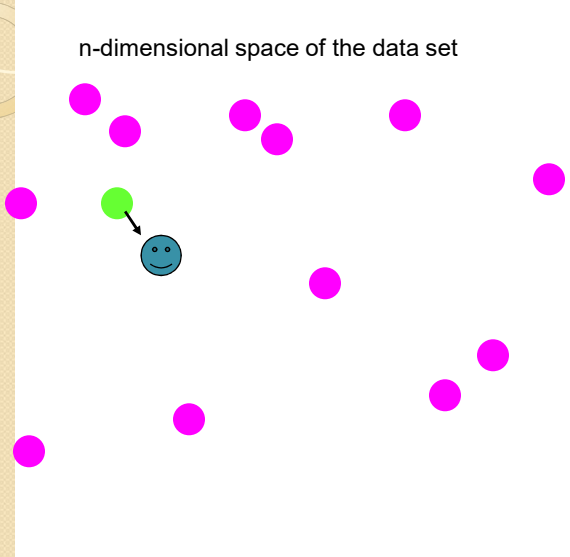
2D SOM Map (often hexagons)

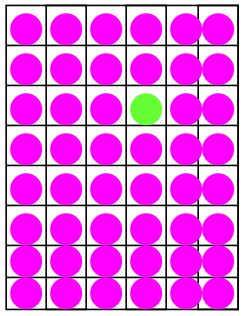
2D map space



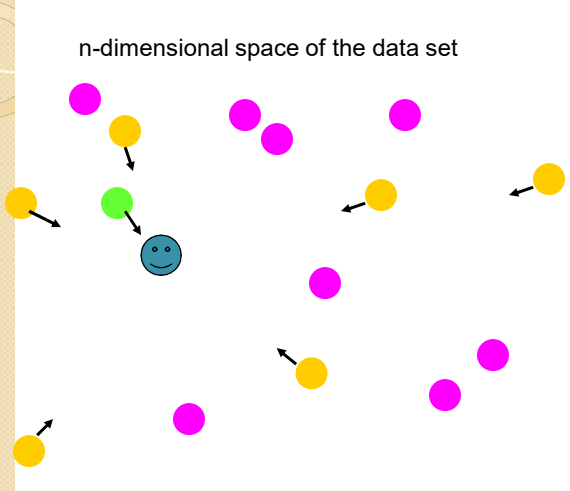
- Representative weights are our 2D visualization of the space
- It's a 2D array of pointers
- Each circle is a pointer to a location in ND space
- The pointer in ND is located at the representative of the cluster
- The representative moves
- The displayed position in the grid does not move…ever

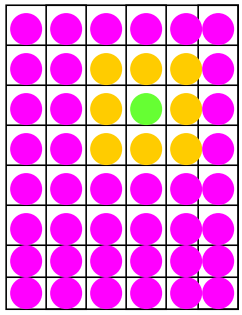
Updating the Neighbors





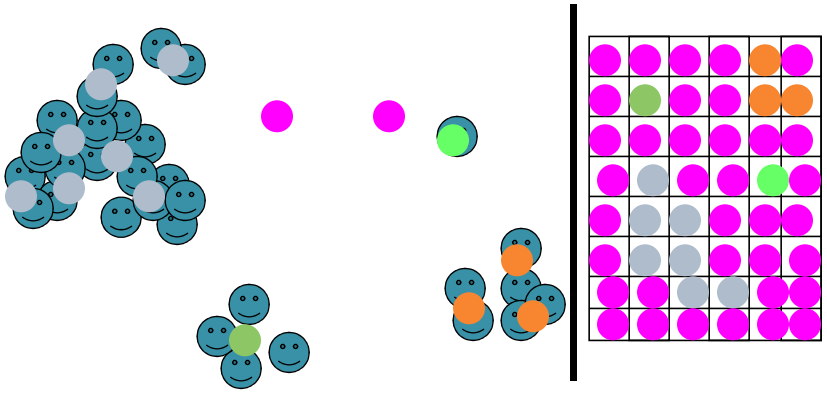
Updating the Neighbors





- Neighbors of best weight in 2D array learn

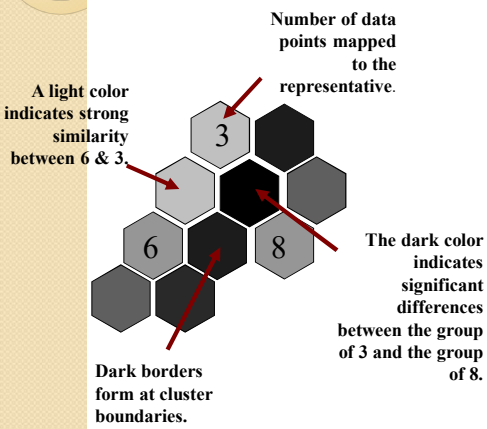
Effect of Preserving Topology



- **Similar representatives map to similar areas**
- Relative distances between clusters will be represented
  - if you are careful about neighborhood parameters
  - We are limited by the reduction of dimensionality

U-Matrix Coloring Algorithm

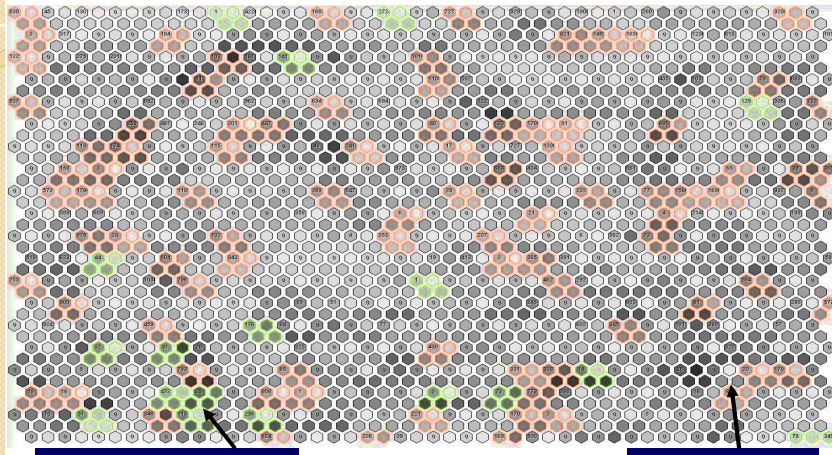
Hexagons with numbers in them are the representatives.



- Hexagons are convenient to represent pairwise distances
- Create histogram of all pairwise distances and map to color table
- Dark = Dissimilar
  - Far distance in n-dimensional space
- Light = Similar
  - Short distance in n-dimensional space

Scalable Self-Organizing Map

- Only known tool that can visualize a large virtual library of 100,000+ compounds and show ADME



Good, Bad & Ugly of SOM

- Good?
- Bad?
- Ugly?

Good, Bad & Ugly of SOM

- Good
  - Scalable and accurate with relatively few parameters
  - Provides a visualization along with the clustering
- Bad
  - Difficult to explain to fellow researchers and customers
  - No implicit definition of cluster
  - Sometimes issues with local minima between similar clusters
- Ugly
  - Appreciating the visualization requires time

Text Mining

Records discovery could solve 50-year-old Birmingham civil rights era bombing case

By Joseph D. Bryant | jbryant@al.com
Email the author | Follow on Twitter
on September 19, 2012 at 7:30 PM, updated September 19, 2012 at 11:39 PM

Print

Birmingham city records analyst, Bruce Wright, looks over boxes of 50-year-old police reports and documents at City Hall Wednesday. The latest discovery of a package of documents might solve a cold case civil right era bombing. (Birmingham News Tamika Moore)

BIRMIINGHAM, Alabama -- A discov among Birmingham civil rights era police records and documents could solve a 50-year-old bombing cold case.

Birmingham lawyer Doug Jones, the former U.S. attorney who successfully prosecuted two men in the 1963 Sixteenth Street Baptist Church bombing, said papers stuffed in an envelope and long forgotten in city files might provide details needed to finally close the case.

Jones, in an interview with The Birmingham News this afternoon, would not name the case, but expressed excitement over the finding. Jones was recently called to City Hall to see the historic files.

"The city had just pulled out all these old police records," he said. "I just looked at the file and said, 'We may be able to solve it.'"

http://blog.al.com/spotnews/2012/09/records_discovery_could_solve.html

Enron

- Dot-com energy company
- Executives fudged the accounting
- Bankruptcy

- Email evidence used to convict senior leadership for securities fraud
- Now a public data set

Power of Language

- Horse
- The horse

Power of Language

- Horse
- The horse
- The horse raced.

Power of Language

- Horse
- The horse
- The horse raced.
- The horse raced past.

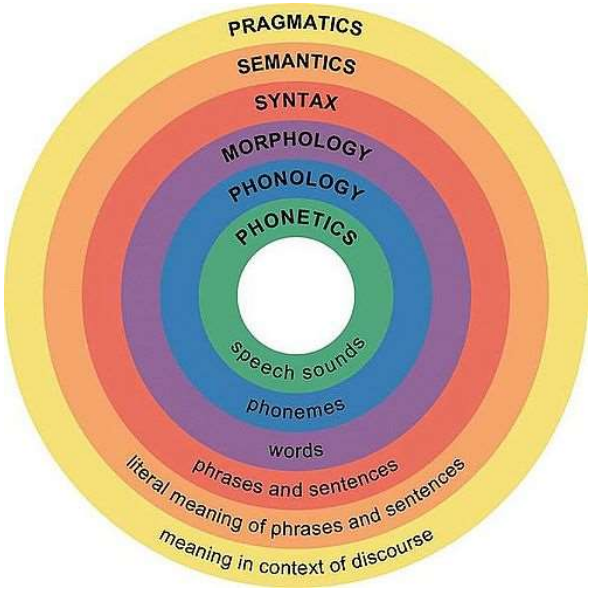
Power of Language

- Horse
- The horse
- The horse raced.
- The horse raced past.
- The horse raced past the barn.

Power of Language

- Horse
- The horse
- The horse raced.
- The horse raced past.
- The horse raced past the barn.
- The horse raced past the barn fell.

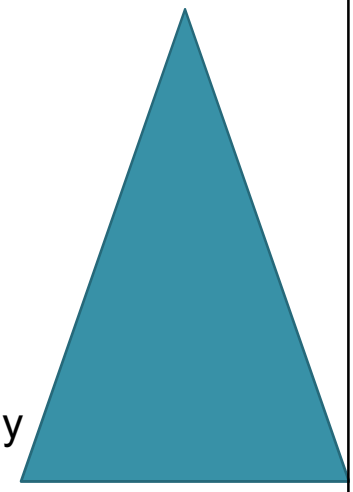
Linguistics Hierarchy



http://www.sanjaymeena.net/introduction-to-linguistics.html

Linguistics Pyramid

- Pragmatics
- Sociolinguistics
- Semantics
- Syntax
- Morphology
- Lexical
- Phonology / Orthography



Linguistics Pyramid

- Pragmatics
- Sociolinguistics
- Semantics
- Syntax
- Morphology
- Lexical
- Phonology

Levels of Linguistics

thehorseracedpastthebarn

The horse raced past the barn.

article noun verb adverb article noun

Noun phrase

Noun phrase

Adverbial phrase

Predicate

sentence







animal movement location

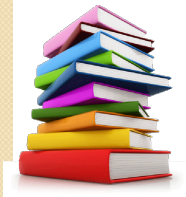
Move(agent:animal, where: past barn)

Introduction to Text Mining

Emails

Forms

?



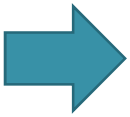
Tweets

- http://www.turnafrownaround.org/volunteer/documents/
- http://www.thebooksamaritan.com/
- http://onclickads.net/?auction_id=74e2bffd191e9427&zoneid=302895&pbk2=6060e265141b149e46c75dc98df186da620104441004039363 2&r=%2Foc%2Fhan 2Ftomb

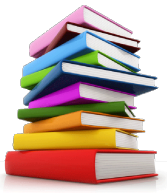
Introduction to Text Mining

Emails

Forms



Features



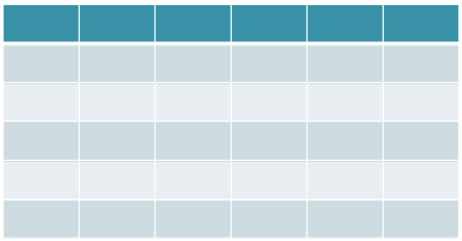
Tweets

- http://www.turnafrownaround.org/volunteer/documents/
- http://www.thebooksamaritan.com/
- http://onclickads.net/?auction_id=74e2bffd191e9427&zoneid=302895&pbk2=6060e265141b149e46c75dc98df186da620104441004039363 2&r=%2Foc%2Fhan 2Ftomb

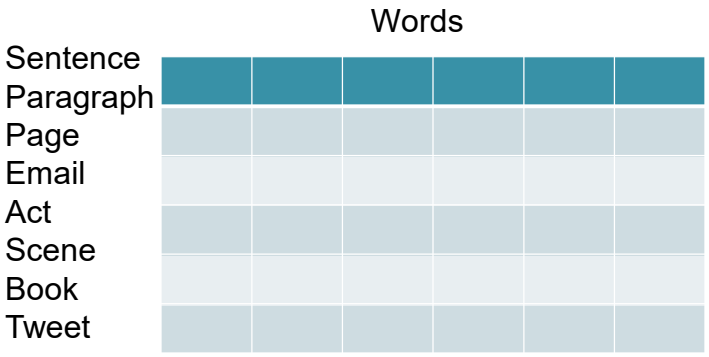
What are the rows and columns?

Data feature

Data instance



What are the rows and columns?



Approaches to Text Mining

- Bag of words
- N-gram
- Shallow parsing
- Full parsing

Approaches to Text Mining

- Bag of words
  - Randomly ordered collection of words
  - {hello, my, name, is, peter}
- N-gram
  - Order within each set of N adjacent units but random N-grams
  - {hello my, my name, name is, is peter}
- Shallow parsing
  - Recognize subset of subject-object-verb or other relationships
    - Subject=Name
    - Verb=is
    - Predicate Nominative=Peter

Full parsing

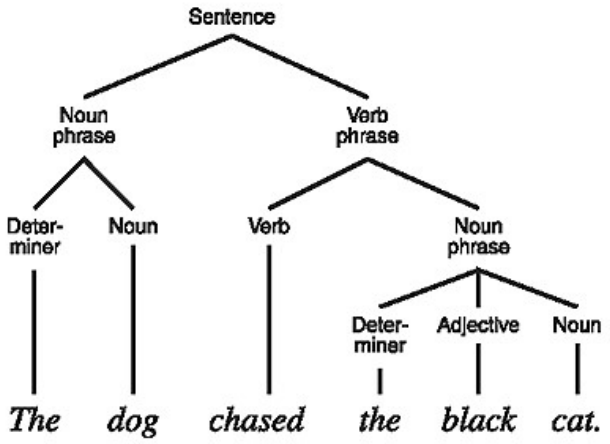


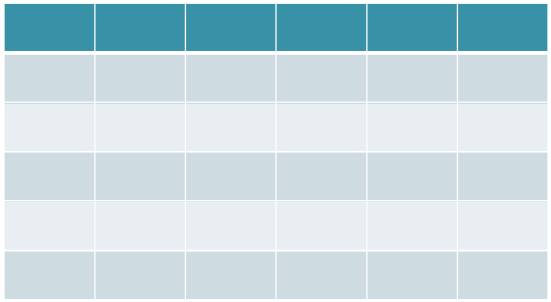
FIGURE 192. Parsing: structure of a sentence

- http://www.suggestkeyword.com/cGFyc2luZyAgbWVhbmluZw/

What are the rows and columns?

Words

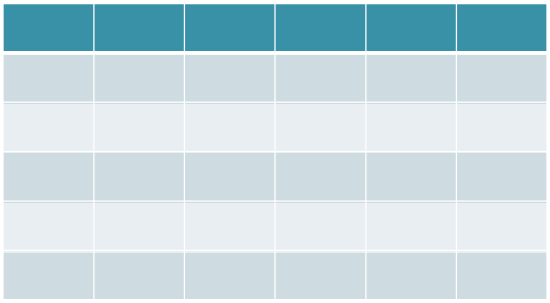
Sentence
Paragraph
Page
Email
Act
Scene
Book
Tweet



Thinking about clustering text…

Words

Sentence
Paragraph
Page
Email
Act
Scene
Book
Tweet



- How would you characterize the distance between two sentences?

Workflow for Bag of Words

- Extract the sentence/paragraph/page
- Remove the punctuation
  - Remove capitalization?
- Extract the words
- Apply [Brill] Part of Speech tagger?
- Remove morphological endings
- Remove stop words

Brill Tagger

- Developed Eric Brill in 1992
- Supervised learning approach to label each word with a part of speech
- Uses a dictionary
- Uses the word structure to guess if it's not in the dictionary
- Set of rules
  - IN NN WDPREVTAG DT while
  - IN=preposition, NN = noun DT=determiner
  - Word "while" should be changed from preposition to noun if previous word tag is DT
  - "in a while" should have while be noun

Stemmers or Lemmatization

- Morphology = modification of a word
- Swim → swims, swimming, swam, etc.
- Peach→ peachy, peaches
- Amylase → β-amylase

- Goal: swimming, swims → swim
- Porter Stemmer is one of most common
  - Arguably pretty bad

- Some stemmers require part of speech
  - Noun plurals are good for nouns only

All words are not created equal

- Clustering documents are not likely to be influenced by filler words like “a”, “the”, “in”, etc.
- Useless words are usually removed
- Called “stop words”