


Distance Metrics & Dimensionality reduction

M. Vazirgiannis

 **DaScIM**, LIX, École Polytechnique

September, 2022

- **Distance Measures**
- Data Exploration and Preprocessing
- Dimensionality Reduction

- Machine Learning algorithms capitalize on similarity or distance measures between objects.
- Similarity or distance between data points can be expressed as:
 - Explicit similarity for each pair of objects
 - Similarity obtained indirectly based on data vector attributes
- A distance $d(i, j)$ is a **metric** iff
 - ① $d(i, j) \geq 0$ for all i, j and $d(i, j) = 0$ iff $i = j$
 - ② $d(i, j) = d(j, i)$ for all i and j
 - ③ $d(i, j) \leq d(i, k) + d(k, j)$ for all i, j and k
- It has to have the shuffling invariant property

- Notation: n objects with p attributes

$$x(i) = (x_1(i), x_2(i), \dots, x_p(i))$$

- Most common distance metric is *Euclidean* distance:

$$d_E(i, j) = \left(\sum (x_k(i) - x_k(j))^2 \right)^{1/2}$$

- Makes sense in the case where the different measurements are proportional; each variable measured in the same units.
- If the measurements are different, length and weight, it is not clear – need for standardization

- Finally, if we have some idea of the relative importance of each variable, we can weight them:

$$d_E(i, j) = \left(\sum w_k (x_k(i) - x_k(j))^2 \right)^{1/2}$$

- Minkowski or L_p metric:

$$d_E(i, j) = \left(\sum_{k=1}^p (x_k(i) - x_k(j))^\lambda \right)^{1/\lambda}$$

- Manhattan, city block or L_1 metric:

$$d_E(i, j) = \sum_{k=1}^p |x_k(i) - x_k(j)|$$

- Chebyshev L_∞

$$d_E(i, j) = \max_k |x_k(i) - x_k(j)|$$

- Sorensen

$$d_{\text{sor}}(i, j) = \frac{\sum_{k=1}^p |x_k(i) - x_k(j)|}{\sum_{k=1}^p |x_k(i) + x_k(j)|}$$

- Gowers

$$d_{\text{gow}}(i, j) = 1/p \sum_{k=1}^p |x_k(i) - x_k(j)|$$

- Lorentzian

$$d_{\text{lor}}(i, j) = \sum_{k=1}^p \ln(1 + |x_k(i) - x_k(j)|)$$

- Inner product

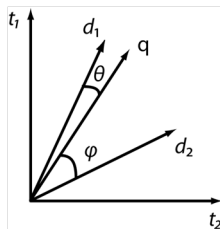
$$s_{IP}(i, j) = \sum_{k=1}^p x_k(i) x_k(j)$$

- Harmonic Mean

$$s_{HM}(i, j) = 2 \sum_{k=1}^p \frac{x_k(i) x_k(j)}{x_k(i) + x_k(j)}$$

- Cosine based similarity

$$\text{sim}(q, d) = \frac{q \cdot d}{|q||d|} = \frac{\sum_{k=1}^p w_{k,q} \cdot w_{k,d}}{\sqrt{\sum_{k=1}^p w_{k,q}^2} \cdot \sqrt{\sum_{k=1}^p w_{k,d}^2}}$$



Intersection family ¹

- Intersection

$$s_{IS}(i, j) = \sum_{k=1}^p \min(x_k(i), x_k(j))$$

- Czekanowski

$$s_{Cze}(i, j) = \frac{2 \sum_{k=1}^p \min(x_k(i), x_k(j))}{\sum_{k=1}^p (x_k(i) + x_k(j))}$$

- Jaccard

$$s_{Jac}(i, j) = \frac{\sum_{k=1}^p x_k(i)x_k(j)}{\sum_{k=1}^p x_k(i)^2 + \sum_{k=1}^p x_k(j)^2 - \sum_{k=1}^p x_k(i)x_k(j)}$$

- Dice

$$s_{Dice}(i, j) = \frac{2 \sum_{k=1}^p x_k(i)x_k(j)}{\sum_{k=1}^p x_k(i)^2 + \sum_{k=1}^p x_k(j)^2}$$

¹Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions Sung-Hyuk Cha, INT. J. OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES

- Squared Euclidean

$$d_{sqe}(i, j) = \sum_{k=1}^p (x_k(i) - x_k(j))^2$$

- Pearson χ^2

$$d_{pre}(i, j) = \frac{\sum_{k=1}^p (x_k(i) - x_k(j))^2}{x_k(j)}$$

- Divergence

$$d_{DIV}(i, j) = 2 \sum_{k=1}^p \frac{(x_k(i) - x_k(j))^2}{(x_k(i) + x_k(j))^2}$$

- Kullback Leibler

$$d_{KL}(i, j) = \sum_{k=1}^p x_k(i) \ln \frac{x_k(i)}{x_k(j)}$$

- Jeffreys

$$d_{JF}(i, j) = \sum_{k=1}^p (x_k(i) - x_k(j)) \ln \frac{x_k(i)}{x_k(j)}$$

- K-divergence

$$d_{kids}(i, j) = \sum_{k=1}^p x_k(i) \ln \frac{2x_k(i)}{x_k(i) + x_k(j)}$$

- Jensen Shannon

$$d_{JS}(i, j) = 1/2 \left[\sum_{k=1}^p x_k(i) \ln \frac{2x_k(i)}{x_k(i) + x_k(j)} + \sum_{k=1}^p x_k(j) \ln \frac{2x_k(j)}{x_k(i) + x_k(j)} \right]$$

- Nominal variables
 - Number of matches divided by number of dimensions

A	A	B	B	C	B	B	C	C	A
A	B	B	A	C	B	B	C	C	C

7/10

- Edit (Levenshtein) distance
 - kitten → sitten (substitution of "s" for "k")
 - sitten → sittin (substitution of "i" for "e")
 - sittin → sitting (insertion of "g" at the end)

- Methods not including formal statistical modeling and inference
 - Detection of mistakes
 - Checking of assumptions
 - Preliminary selection of appropriate models
 - Determining relationships among the explanatory variables, and
 - Assessing the direction and rough size of relationships between explanatory and outcome variables (i.e. demographics – purchase)
- Useful information about the data
 - Min and Max values
 - Mean Value
 - Standard Deviation
 - Number of instances per value (for nominal data)
 - Percentage of missing values
 - Data distribution

- 0-1 scaling:

- each variable V is recomputed as

$$V = (V - \min V) / (\max V - \min V)$$

- allows variables to have differing means and standard deviations but equal ranges.
- at least one value at the 0 and 1 endpoints.

- Dividing each value by the range:

- each variable V is recomputed as

$$V = V / (\max V - \min V)$$

- means, variances, and ranges of the variables are still different
- ranges are likely to be more similar.

- Z-score scaling:
 - each variable V is recomputed as $(V - \text{mean of } V)/s$,
 s standard deviation.
 - all variables have equal means (0) and standard deviations (1) but different ranges.
- Dividing each value by the standard deviation.
 - transformed variables with variances of 1
 - different means and ranges.

- Covariance and correlation measure linear dependence
- Assume variables X and Y and n objects taking on values $x(1), \dots, x(n)$ and $y(1), \dots, y(n)$.
- Sample **covariance** of X and Y is:

$$\text{Cov}(X, Y) = \frac{1}{N} \sum_1^n (x(i) - \bar{x})(y(i) - \bar{y})$$

- Covariance measures how X and Y vary together.
 - large and positive if large values of $X \Rightarrow$ large values of Y , and small $X \Rightarrow$ small Y

- Covariance depends on ranges of X and Y
- Standardize dividing with standard deviation
- Sample **correlation** coefficient

$$\rho(X, Y) = \frac{\sum_{i=1}^n (x(i) - \bar{x})(y(i) - \bar{y})}{(\sum_{i=1}^n (x(i) - \bar{x})^2 \sum_{i=1}^n (y(i) - \bar{y})^2)^{\frac{1}{2}}}$$

Dimensionality Reduction

- Some coordinates do not contribute to the data representation.
- Subsets of the dimensions may be highly correlated.
- Nearest neighbor is distorted in a high dimensional space
- Low dimension intuitions do not apply to high dimensions
- Empty space phenomenon

Empty space phenomenon

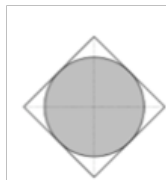
- Hyper-sphere (\mathbf{S}) within a hyper-rectangle (\mathbf{R})
- Respective volumes:

$$V(\mathbf{S}) = 2r^d \pi^{\frac{d}{2}} / d\Gamma(d/2), \quad V(\mathbf{R}) = (2r)^d$$

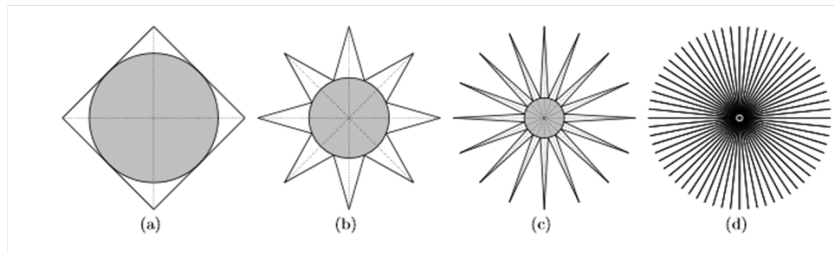
- Fraction of sphere within the rectangle becomes insignificant with d increasing:

$$\lim_{d \rightarrow \infty} \left(\pi^{\frac{d}{2}} / d 2^{d-1} \Gamma(d/2) \right) = 0$$

- the normal distribution in high dimensions
- longest/shortest distances converge
- clustering becomes infeasible



Inscription of hyper sphere in a hypercube²



- The radius of the inscribed circle accurately reflects the difference between the volume of the hypercube and the inscribed hypersphere in d -dimensions.

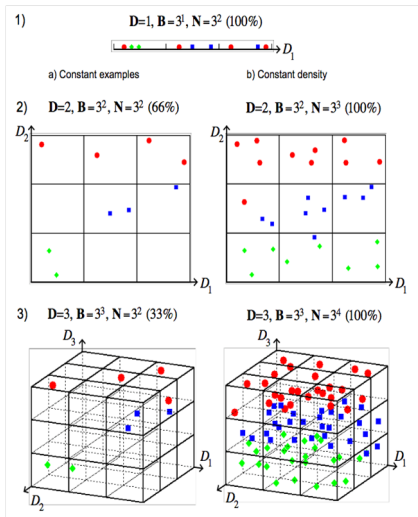
²<http://www.cs.rpi.edu/~zaki/www-new/uploads/Dmcourse/Main/chap6.pdf>

Curse of Dimensionality [Belmann 1961]

- Some coordinates do not contribute to the data representation.
- Subsets of the dimensions may be highly correlated.
- Nearest neighbor is distorted in a high dimensional space
- Low dimension intuitions do not apply to high dimensions

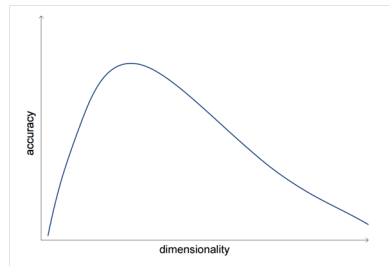
Curse of Dimensionality

- Assuming 3 classes (colors)
 - same number of points embedded in higher dimensions (**sparsity**)
 - need exponentially more points to maintain density in higher dimensions (**curse of dimensionality**)
 - Data tend to gather in extremely of small areas of the multidimensional space (**empty space phenomenon**)



Curse of Dimensionality

- Point queries
 - “as dimensionality increases, distance to the nearest data point approaches the distance to the farthest data point” - “When Is “Nearest Neighbor” Meaningful? “ - Beyer et al., [1999]
 - Increasing dimensionality may decrease of overall accuracy of system according to statistical learning theory approach [Vapnik, 1998].
 - for a given dataset, there is a maximum number of dimensions above which the quality of data analysis degrades when the number of training samples is small relative to dimensionality



- methods optimise an objective function
- does not contain any local optima - the solution space is convex [Boyd and Vandenberghe, 2004].
- has usually the form of solving an eigenvalue problem.
- final embedding space formed by eigenvectors which correspond to smallest or largest eigenvalues.

- Global methods: eigen-decomposition of a dense cost matrix
 - Methods: Principal Component Analysis, Multidimensional Scaling, Kernel Principal Component Analysis, Isomap, Maximum Variance Unfolding
- Local methods: eigen-decomposition of a sparse cost matrix
 - Methods: Locally Linear Embedding, Laplacian Eigenmaps etc.

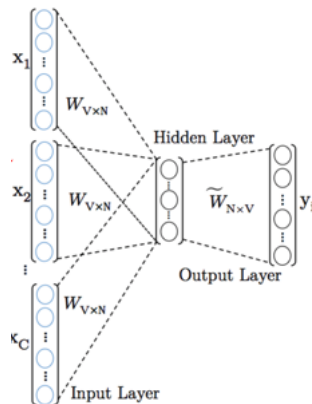
- *Matrix Factorization* methods
 - Principal Components Analysis (PCA)
 - Singular Value Decomposition (SVD)
 - Multidimensional Scaling (MDS)
 - Non negative Matrix Factorization (NMF)
 - Latent Semantic Indexing (LSI)

Dimensionality Reduction via Deep Learning

CBOW architecture

- Input C words
- Task: predict middle word
- Large dimensionality input $C \times V$
- Hidden layer dim: N
- Output $N \times V$

Data mapped to a $N \ll V$ dimensional space



Low Rank Approximation

- Data: $\mathbf{X} = \{\mathbf{x}_i \in \mathbf{R}^{m \times n} \mid \mathbf{x}_i \text{ columns of } \mathbf{X}\}$
- Goal: approximate $\mathbf{X} = \mathbf{U}\mathbf{V}^T$, $\mathbf{U} \in \mathbf{R}^{m \times r}$, $\mathbf{V} \in \mathbf{R}^{n \times r}$, $r \ll n$
 - each data vector $\mathbf{x}_i : \mathbf{x}_i \sim \mathbf{U}\mathbf{v}_i^T$, \mathbf{v}_i is the i -th column of \mathbf{V}
- Geometric interpretation:
 - each data vector $\mathbf{x}_i \in \mathbf{R}^m, \mathbf{x}_i \sim \mathbf{U}\mathbf{v}_i^T$, is approximated by its projection to an r -dimensional space spanned by the column vectors of \mathbf{U}
 - $\mathbf{Y} = \mathbf{U}\mathbf{V}^T$ the approximation matrix, max rank r

Evaluating the approximation

- Assuming a matrix $X_{m \times n}$ we need to define their similarity/distance.
- A popular matrix norm is the Frobenius (L_2 norm treated as a vector)

$$|X|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2} = \text{Tr}(XX^T)$$

- So assuming: $X = UV^T$
- the error approximation will be: $|X|_F - |UV^T|_F$

- A $n \times n$ matrix
 - eigenvalues $\lambda : |A - \lambda I| = 0$
 - Eigenvectors $x : Ax = \lambda x$
 - Matrix rank: # linearly independent rows or columns
 - A real symmetric table A $n \times n$ can be expressed as: $A = U\Lambda U^T$
 - U 's columns are A 's eigenvectors
 - Λ 's diagonal contains A 's eigenvalues
 - $A = U\Lambda U^T = \lambda_1 x_1 x_1^T + \lambda_2 x_2 x_2^T + \dots + \lambda_n x_n x_n^T$
 - $x_1 x_1^T$ represents projection via x_1 (λ_i eigenvalue, x_i eigenvector)
 - Interpretations XX^T vs. $X^T X$

Singular Value Decomposition (SVD)

Eigen decomposition applied to square matrices. For non square matrices we apply [Singular Value Decomposition](#).

SVD insight: treat the rows of $X_{n \times m}$ matrix as n points in a m -dimensional space

- Consider the problem of finding the best k -dimensional subspace with respect to the set of points ($k \ll m$).
- Best - best least squares fit: minimize the sum of the squares of the perpendicular distances of the points to the subspace.

Singular Value Decomposition (SVD) - I

- Let X a $n \times m$ table, $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
 - \mathbf{U} : orthogonal $m \times m$, its columns are the eigenvectors of $\mathbf{X}\mathbf{X}^T$
 - \mathbf{U}, \mathbf{V} define orthogonal basis $\mathbf{U}^T \mathbf{U} = \mathbf{V} \mathbf{V}^T = \mathbf{I}$
 - $\mathbf{\Sigma}$: $m \times n$ contains A 's singular values (square roots of $\mathbf{X}\mathbf{X}^T$ eigenvalues)
 - \mathbf{V} : $n \times n$, its columns are eigenvectors of $\mathbf{X}^T \mathbf{X}$
- k -dimensional matrix approximation $X_k = U_k \Sigma_k V_k^T$

Multidimensional Scaling (MDS)

- Application of SVD on the data distance matrix XX^T
- Aim to minimize the stress:

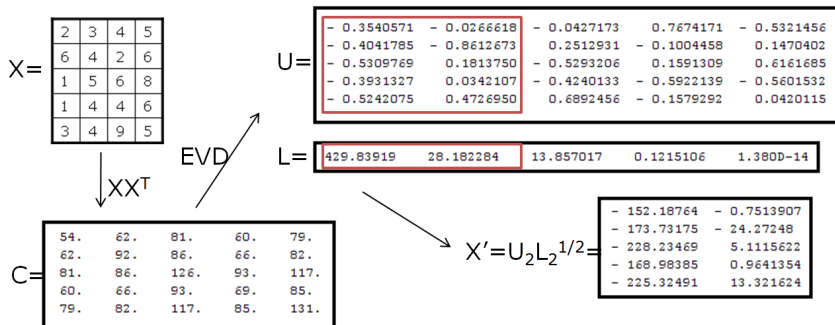
$$\text{stress}(X, X') = \frac{\sum_{ij} (d(i, j) - d'(i, j))^2}{\sum_{ij} (d(i, j))^2}$$

Complexity $O(N^3)$ (N : number of vectors)

- Result:
 - A new representation of the data in a lower dimensional space.
- Implement usually by:
 - Eigen decomposition of the inner product matrix
 - projection on the k eigenvectors corresponding to the k largest eigenvalues.

Multidimensional Scaling

- Data is given as rows in X
 - $C = XX^T$ (inner product of x_i with x_j)
 - Eigen decomposition of $C' = ULU^{-1}$
 - Eventually $X' = U_k L_k^{1/2}$, where k is the projection dimension



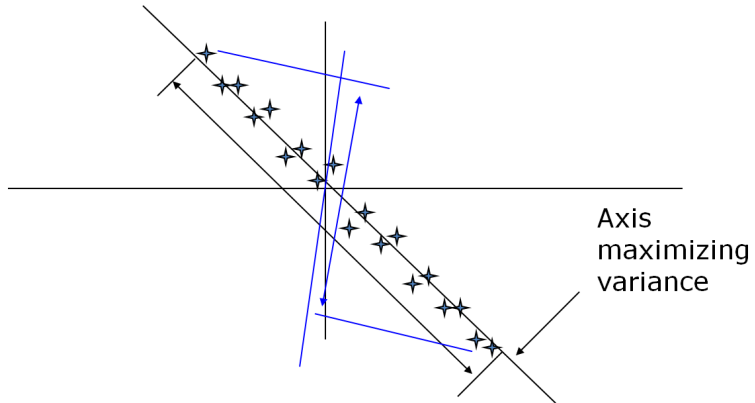
Principal Components Analysis (PCA)

- Main concept of **Principal Components Analysis**: dimensionality reduction, maintaining as much as possible data's variance.
- SVD on the data covariance matrix
- Data variance: $V(X) = \sigma^2 = E[(X - \mu)^2]$
- Let N objects, with mean value, m , it is approximated as:

$$\frac{1}{N} \sum_{i=1}^N (x_i - m)^2$$

Sample of N objects with unknown mean value: $\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$

Dimensionality reduction based on variance maintenance



- Let Matrix $X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$ where X_i vectors

- covariance matrix Σ is the matrix whose (i, j) entry is the covariance

$$\Sigma = \begin{bmatrix} E[(X_1 - \mu_1)(X_1 - \mu_1)] & E[(X_1 - \mu_1)(X_2 - \mu_2)] & \cdots & E[(X_1 - \mu_1)(X_n - \mu_n)] \\ E[(X_2 - \mu_2)(X_1 - \mu_1)] & E[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & E[(X_2 - \mu_2)(X_n - \mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ E[(X_n - \mu_n)(X_1 - \mu_1)] & E[(X_n - \mu_n)(X_2 - \mu_2)] & \cdots & E[(X_n - \mu_n)(X_n - \mu_n)] \end{bmatrix}$$

- Also: $\text{cov}(X) = [X']^T X'$, where $X' = X - M$

Principal Components Analysis (PCA)

- PCA intuition: maximization of the covariance.
 - Variance: Depicts the maximum deviation of a random variable from the mean.

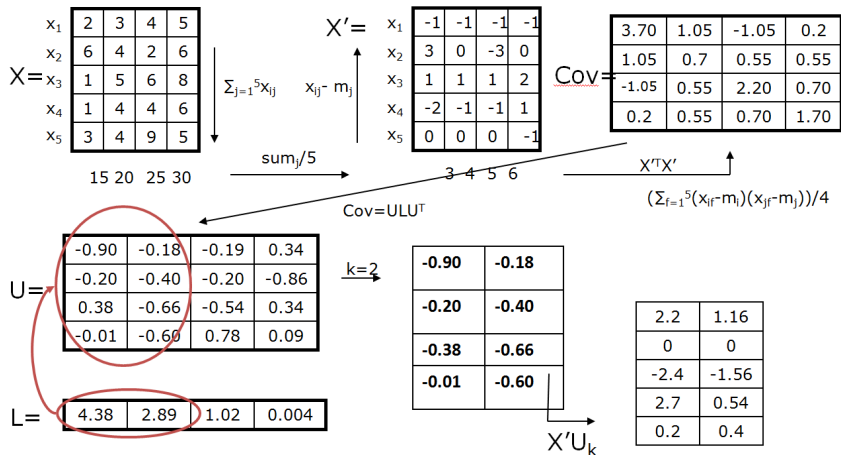
$$\sigma^2 = \sum_{i=1}^n (x_i - \mu_i)^2 / n$$

- Method:
 - Data feature p variables and contained as rows in matrix $X_{p \times n}$
 - Covariance matrix $W = [X']^T X'$, $X' = X - M$
- Calculate eigenvalues and eigenvectors of

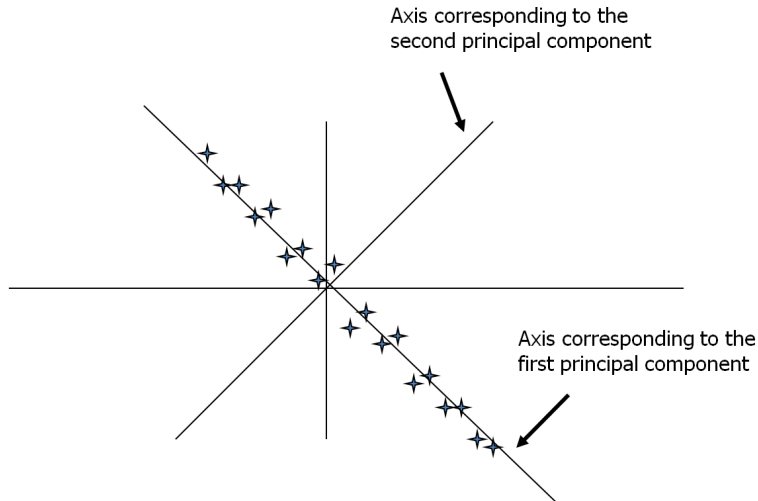
$$W = U \Lambda U^T$$

- Retain k largest eigenvalues and eigenvectors
 - k is estimated by $\sum_{j=k+1}^p \lambda_j / \sum_{j=1}^p \lambda_j > 85\%$
- Projection: $X'_k = X' U_k$

Principal Components Analysis



PCA, example



- Preprocessing step preceding the application of data mining algorithms (such as clustering).
- Data Visualization & Noise reduction.
- It is a dimensionality reduction method
- Nominal complexity $O(np^2 + p^3)$
 - n : number of data points
 - p : number of initial space dimensions
- The new space maintains sufficiently the data variance.

Explaining data by factorization

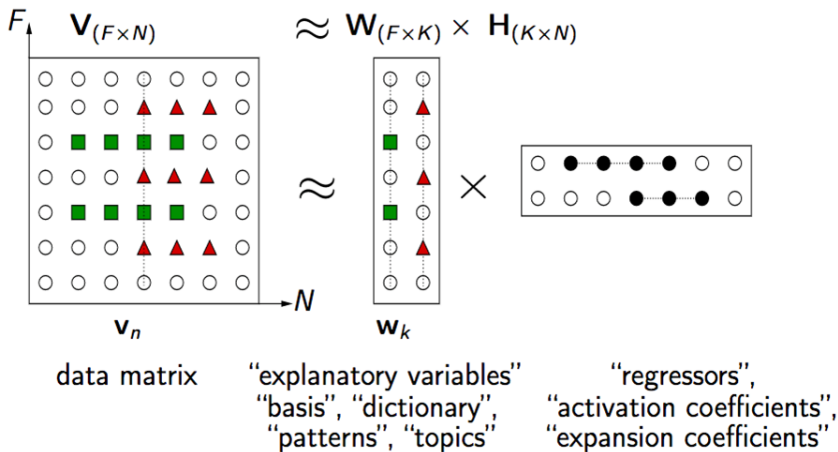


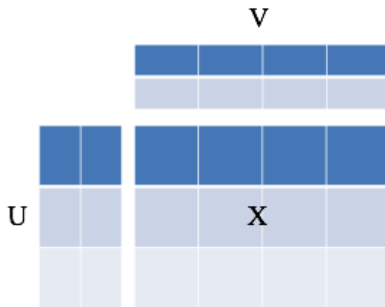
Illustration by C. Févotte

Non Negative Matrix factorization (NMF)

- Data is often nonnegative by nature
 - pixel intensities; occurrence counts; food or energy consumption; user scores; stock market values;
- Interpretability of the results, optimal processing of nonnegative data may call for processing under Nonnegativity constraints
- Applying SVD results in factorized matrices with positive and negative elements may contradict the physical meaning of the result.
 - Nonnegative matrix factorization (NMF)
 - find the reduced rank nonnegative factors to approximate a given nonnegative data matrix.

NMF model

- $X \simeq UV^T$
 - $U = [u_{fk}], w_{fk} \geq 0$
 - $V = [v_{kn}], h_{kn} \geq 0$
 - $k \ll f, n$



- Assume X ($m \times n$) data matrix and $r \ll m, n$
- NMF aims to find non negative matrices

$$U \in R^{m \times r}, V \in R^{r \times n} : X \approx UV^T$$

- To find U, V , optimization problem:

$$\min_{(U,V)} \|X - UV^T\|_2$$

- Alternative error function:

$$\begin{aligned} \min_{U,V} f(U, V) &= \sum_{i=1}^m \sum_{j=1}^n \left(X_{ij} \log \frac{X_{ij}}{(UV^T)_{ij}} - X_{ij} + (UV^T)_{ij} \right) \\ \text{s.t. } U_{ia} &\geq 0, V_{jb} \geq 0, \forall i, a, b, j \end{aligned}$$

Alternating Least squares

- ① Suppose we know U , with V unknown.

for each j minimize $\|X_{\cdot j} - UV_{\cdot j}^T\|_2$

- find $V_{\cdot j}$ that minimizes with $X_{\cdot j}$ and U known.
- Frobenius norm: sum of squares,
 - minimization is a least-squares problem, i.e. linear regression
 - “predicting” $X_{\cdot j}$ from W .

$$V_{\cdot j} = (U^T U)^{-1} U^T X_{\cdot j}$$

- repeat for all columns $V_{\cdot j}$

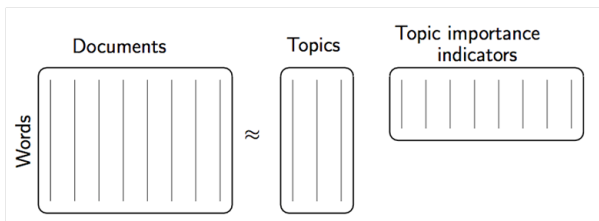
- ② assume V , with U unknown: $X^T = VU^T$

- Interchange roles of U , V in the above optimization
- Compute a row of U , repeat for all rows

- Putting all this together
 - random initialization of U and V
 - alternate:
 - Compute U assuming V known
 - Compute V based on that new U
 - ...
 - may generate negative values: truncate to 0

NMF issues, applications

- choice of NMF dimensionality
- $U_{m \times r}$, r (rank) choice: via SVD ...
- Applications
 - Topic detection
 - Source separation (music, speech)
 - Clustering
 - Recommendations



- based on the "Stochastic Neighbor Embedding" - Hinton, 2002.
- Stochastic Neighbor Embedding (SNE): map high-dimensional Euclidean point distances to conditional probabilities representing similarities.
- similarity between x_j , x_i - conditional probability, $p_{j|i}$ x_j , pick x_i as its neighbor, using a probability density under a Gaussian centered at x_i
- For nearby datapoints, $p_{j|i}$ is high vs. for widely separated datapoints, $p_{j|i}$ will be very low

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

For low-dimensional representations y_i , y_j resp. conditional probability $q_{j|i}$:

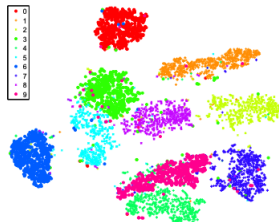
$$q_{j|i} = \frac{\exp(-|y_i - y_j|^2 / 2\sigma_i^2)}{\sum_{k \neq i} (\exp(-|y_i - y_k|^2 / 2\sigma_i^2))}$$

- For correct mapping: $p_{j|i} = q_{j|i}$, thus SNE finds a low-dimensional data representation minimising cost as Kullback Leibler divergence with gradient descent
- The cost function C:

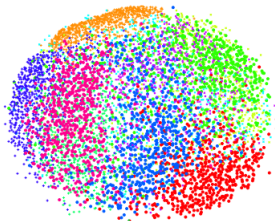
$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

The gradient of symmetric SNE: $\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ji})(y_i - y_j)$

- Data Sets:
MNIST data set, Olivetti faces, COIL-20 data set, the word-features data set, Netflix data set
- Baselines:
ISOMAP, t-SNE, Sammon mapping, Isomap, LLE

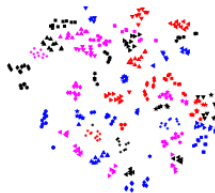


(a) Visualization by t-SNE.



(b) Visualization by Sammon mapping.

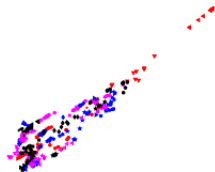
- Data Sets: MNIST data set, **Olivetti faces**, COIL-20 data set, the word-features data set, Netflix data set
- Baselines: ISOMAP, t-SNE, Sammon mapping, Isomap, LLE



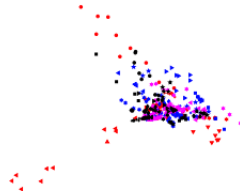
(a) Visualization by t-SNE.



(b) Visualization by Sammon mapping.



(c) Visualization by Isomap.



(d) Visualization by LLE.

SVD application - Latent Structure in documents³

- Documents are represented based on the Vector Space Model
- Vector space model consists of the keywords contained in a document.
- In many cases baseline keyword based performs poorly – not able to detect synonyms.
- Therefore document clustering is problematic
- Example where of keyword matching with the query: “IDF in computer-based information look-up”

	access	document	retrieval	information	theory	database	indexing	computer
Doc1	x	x	x			x	x	
Doc2				x	x			x
Doc3			x	x				x

³Indexing by Latent Semantic Analysis (1990) Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman, Journal of the American Society of Information Science

- Finding similarity with exact keyword matching is problematic.
- Using SVD we process the initial document-term document.
- Then we choose the k larger singular values. The resulting matrix is of order k and is the most similar to the original one based on the Frobenius norm than any other k -order matrix.

- The initial matrix is SVD decomposed as: $A = ULV^T$
- Choosing the top- k singular values from L we have:

$$A_k = U_k L_k V_k^T$$

- L_k is square $k \times k$ containing the top- k singular values of the diagonal in matrix L ,
- U_k , the $m \times k$ matrix containing the first k columns in U (left singular vectors)
- V_k^T , the $k \times n$ matrix containing the first k lines of V^T (right singular vectors)
Typical values for $k \sim 200-300$ (empirically chosen based on experiments appearing in the bibliography)

- Term to term similarity: $A_k A_k^T = U_k L_k^2 U_k^T$, $A_k = U_k L_k V_t$
- Document-document similarity: $A_k^T A_k = V_k L_k^2 V_k^T$
- Term document similarity (as an element of the transformed – document matrix)
- Extended query capabilities transforming initial query q to q_n :
 $q_n = q^T U_k L_k^{-1}$
- Thus q_n can be regarded a line in matrix V_k

- LSI application on a term – document matrix
 - C1: Human machine Interface for Lab ABC computer application
 - C2: A survey of user opinion of computer system response time
 - C3: The EPS user interface management system
 - C4: System and human system engineering testing of EPS
 - C5: Relation of user-perceived response time to error measurements
 - M1: The generation of random, binary unordered trees
 - M2: The intersection graph of path in trees
 - M3: Graph minors IV: Widths of trees and well-quasi-ordering
 - M4: Graph minors: A survey
- The dataset consists of 2 classes, 1st: “human – computer interaction” (c1-c5) 2nd: related to graph (m1-m4). After feature extraction the titles are represented as follows.

LSI – an example

	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	1	0	0	1	0	0	0	0	0
Interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	0	1	1	2	0	0	0	0	0
Response	0	1	0	0	1	0	0	0	0
Time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
Survey	0	1	0	0	0	0	0	0	1
Trees	0	0	0	0	0	1	1	1	0
Graph	0	0	0	0	0	0	1	1	1
Minors	0	0	0	0	0	0	0	1	1

LSI – an example

- $A = ULV^T$
- $A =$

1	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	2	0	0	0	0	0
0	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	0	0	0	0	0	1
0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	1	1

LSI – an example

- $A = ULV^T$

- $U =$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41	0	0	0
0.2	-0.07	0.14	-0.55	0.28	0.5	-0.07	-0.01	-0.11	0	0	0
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.3	0.06	0.49	0	0	0
0.4	0.06	-0.34	0.1	0.33	0.38	0	0	0.01	0	0	0
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27	0	0	0
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05	0	0	0
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05	0	0	0
0.3	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17	0	0	0
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58	0	0	0
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23	0	0	0
0.04	0.62	0.22	0	-0.07	0.11	0.16	-0.68	0.23	0	0	0
0.03	0.45	0.14	-0.01	-0.3	0.28	0.34	0.68	0.18	0	0	0

LSI – an example

- $A = ULV^T$
- $L =$

3.34	0	0	0	0	0	0	0	0
0	2.54	0	0	0	0	0	0	0
0	0	2.35	0	0	0	0	0	0
0	0	0	1.64	0	0	0	0	0
0	0	0	0	1.5	0	0	0	0
0	0	0	0	0	1.31	0	0	0
0	0	0	0	0	0	0.85	0	0
0	0	0	0	0	0	0	0.56	0
0	0	0	0	0	0	0	0	0.36
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

LSI – an example

- $A = ULV^T$
- $V =$

0.2	-0.06	0.11	-0.95	0.05	-0.08	0.18	-0.01	-0.06
0.61	0.17	-0.5	-0.03	-0.21	-0.26	-0.43	0.05	0.24
0.46	-0.13	0.21	0.04	0.38	0.72	-0.24	0.01	0.02
0.54	-0.23	0.57	0.27	-0.21	-0.37	0.26	-0.02	-0.08
0.28	0.11	-0.51	0.15	0.33	0.03	0.67	-0.06	-0.26
0	0.19	0.1	0.02	0.39	-0.3	-0.34	0.45	-0.62
0.01	0.44	0.19	0.02	0.35	-0.21	-0.15	-0.76	0.02
0.02	0.62	0.25	0.01	0.15	0	0.25	0.45	0.52
0.08	0.53	0.08	-0.03	-0.6	0.36	0.04	-0.07	-0.45

LSI – an example

- Choosing the 2 largest singular values we have

$$U_k =$$

0.22	-0.11
0.2	-0.07
0.24	0.04
0.4	0.06
0.64	-0.17
0.27	0.11
0.27	0.11
0.3	-0.14
0.21	0.27
0.01	0.49
0.04	0.62
0.03	0.45

$$L_k =$$

3.34	0
0	2.54

$$V_k^T =$$

0.2	0.61	0.46	0.54	0.28	0	0.02	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53

LSI (2 singular values)

• $A_k =$

	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	0.16	0.4	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
Interface	0.14	0.37	0.33	0.4	0.16	-0.03	-0.07	-0.1	-0.04
Computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
User	0.26	0.84	0.61	0.7	0.39	0.03	0.08	0.12	0.19
System	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
Response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
Time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.2	-0.11
Survey	0.1	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
Trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
Graph	-0.06	0.34	-0.15	-0.3	0.2	0.31	0.69	0.98	0.85
Minors	-0.04	0.25	-0.1	-0.21	0.15	0.22	0.5	0.71	0.62

- Query: “human computer interaction” retrieves documents: c1, c2, c4 but *not* c3 and c5.
- If we submit the same query (based on the transformation shown before) to the transformed matrix we retrieve (using cosine similarity) all c1-c5 even if c3 and c5 have no common keyword to the query.
- According to the transformation for the queries we have:

Query transformation

	query
human	1
Interface	0
computer	1
User	0
System	0
Response	0
Time	0
EPS	0
Survey	0
Trees	0
Graph	0
Minors	0

$q =$

1
0
1
0
0
0
0
0
0
0
0
0

Query transformation

$$q^T =$$

1	0	1	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

$$U_k =$$

0.22	-0.11
0.2	-0.07
0.24	0.04
0.4	0.06
0.64	-0.17
0.27	0.11
0.27	0.11
0.3	-0.14
0.21	0.27
0.01	0.49
0.04	0.62
0.03	0.45

$$L_k =$$

0.3	0
0	0.39

$$q_n = q^T U_k L_k =$$

0.138	-0.0273
-------	---------

Query transformation

Map docs to the 2 dim space $V_k L_k =$

0.2	-0.06
0.61	0.17
0.46	-0.13
0.54	-0.23
0.28	0.11
0	0.19
0.01	0.44
0.02	0.62
0.08	0.53

3.34	0
0	2.54

0.67	-0.15
2.04	0.43
1.54	-0.33
1.8	-0.58
0.94	0.28
0	0.48
0.03	1.12
0.07	1.57
0.27	1.35

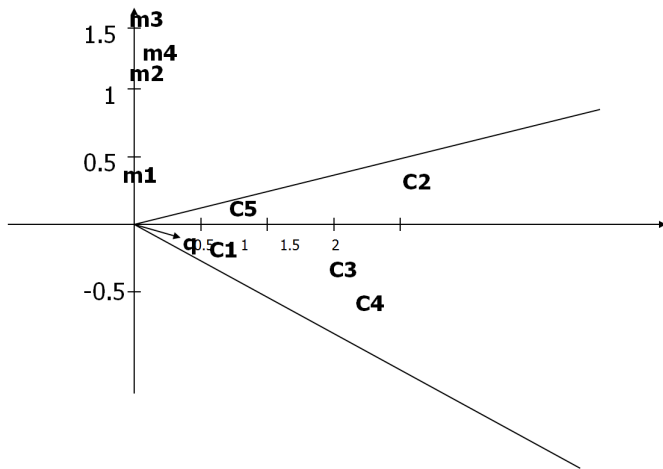
$q_n L_k =$

0.138	-0.0273
-------	---------

3.34	0
0	2.54

0.46	-0.069
------	--------

Query transformation



- Comparison of the transformed query to the new document vectors based on cosine similarity, where the similarity is computed as:

$$\text{Cos}(x, y) = \langle x, y \rangle / \|x\| \cdot \|y\|$$

Where $x = (x_1, \dots, x_n)$, $y = (y_1, \dots, y_n)$

$$\langle x, y \rangle = x_1 * y_1 + \dots + x_n * y_n$$

Query transformation

- The cosine similarity matrix of query vector to the documents is:

	query
C1	0.99
C2	0.94
C3	0.99
C4	0.99
C5	0.9
M1	-0.14
M2	-0.13
M3	-0.11
M4	0.05

