Data mining

Dimensionality reduction

University of Szeged

The role of dimensionality reduction

- We can spare computational costs (or simply fit entire datasets into main memory) if we represent data in fewer dimensions
- Visualization of datasets (in 2 or 3 dimensions)
- Elimination of noise from data, feature selection
- Key idea: try to represent data points in lower dimensions
- Depending our objective function with respect the lower dimensional representation $\rightarrow$ PCA, LDA, SVD, ...

Principal Component Analysis

- Transform multidimensional data into lower dimensions in such a way that we lose as little proportion of the original variation of the data as possible
- Assumption: data points of the original m-dimensional space lie at (or at least very close to) an m'-dimensional subspace $\rightarrow$ we shall express data points with respect this subspace
- What that m'-dimensional subspace might be?
- We would like to minimize the reconstruction error

$$\sum_{i=1}^{n} \| (x_i - x_i') \|^2$$

, where x_i' is an approximation for point x_i

Covariance

Reminder

- Quantifies how much random variables Y and Z change together
- $cov(Y, Z) = \mathbb{E}[(Y - \mu_Y)(Z - \mu_Z)]$
  - $\mu_Y = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\mu_Z = \frac{1}{n}\sum_{i=1}^{n} z_i$

- Columns i, j of data matrix X (i.e. $X_{:,i}, X_{:,j}$) can be regarded as observations from two random variables

Scatter and covariance matrix

- Scatter matrix: $S = \sum_{k=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\intercal$
- (Un)biased covariance matrix: $\Sigma = \frac{1}{n} S$ $(\Sigma = \frac{1}{n-1} S)$

$$\Sigma = \begin{bmatrix} cov(X_{:,1}, X_{:,1}) & cov(X_{:,1}, X_{:,2}) & \dots & cov(X_{:,1}, X_{:,m}) \\ cov(X_{:,2}, X_{:,1}) & cov(X_{:,2}, X_{:,2}) & \dots & cov(X_{:,2}, X_{:,m}) \\ \vdots & \ddots & cov(X_{:,i}, X_{:,j}) & \vdots \\ cov(X_{:,m}, X_{:,1}) & \dots & \dots & cov(X_{:,m}, X_{:,m}) \end{bmatrix}$$

- $\Sigma_{i,j}$ is the covariance of variables i and j $(cov(X_{:,i}, X_{:,j}))$
- What values are included in the main diagonal?

Characteristics of scatter and covariance matrices

- Claim: matrices S and Σ are symmetric and positive definite

Bizonyítás.

$$S = \sum_{k=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\mathsf{T} = \left(\sum_{k=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\mathsf{T} \right)^\mathsf{T} = S^\mathsf{T}$$

$$\boldsymbol{a}^\mathsf{T} S \boldsymbol{a} = \sum_{k=1}^{n} (\boldsymbol{a}^\mathsf{T}(\boldsymbol{x}_i - \boldsymbol{\mu}))((\boldsymbol{x_i} - \boldsymbol{\mu})^\mathsf{T}\boldsymbol{a}) = \sum_{k=1}^{n} (\boldsymbol{a}^\mathsf{T}(\boldsymbol{x_i} - \boldsymbol{\mu}))^2 \geq 0 \qquad \square$$

- Consequence: the eigenvalues of S and Σ are $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0$
- The m'-dimensional projection which preserves most of the variation of the data can be obtained by projecting data points using the eigenvectors belonging to the m' highest eigenvalues of either matrix S (or Σ) (proof: see table)

Lagrange multipliers

- Provides a schema for solving (non-)linear optimization problems

$$f(\boldsymbol{x}) \rightarrow min/max$$
$$\text{such that } g_i(\boldsymbol{x}) = 0 \forall i \in \{1, \ldots, n\}$$

- Lagrange function:

$$L(\boldsymbol{x}, \lambda) = \mathrm{f(x)} - \sum_{\mathrm{i=1}}^{\mathrm{n}} \lambda_{\mathrm{i}} \mathrm{g_i(x)}$$

- Karush-Kuhn-Tucker (KKT) conditions: necessity conditions for an optimum

$$\nabla L(\boldsymbol{x}, \lambda) = 0 \quad (1)$$
$$\lambda_i g_i(\boldsymbol{x}) = 0 \forall i \in \{1, \ldots, n\} \quad (2)$$
$$\lambda_i \geq 0 \quad (3)$$

Practical issues

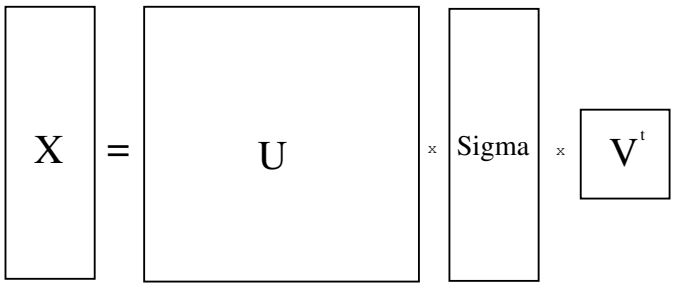
- Its worth handling all the features on similar scales
  - min-max normalization: $x_{i,j} = \frac{x_{i,j} - min(x_{*,j})}{max(x_{*,j}) - min(x_{*,j})}$
  - standardization: $x_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}$
- How to choose the reduced dimensionality (m')?

  $Hint : \left(\sum_{i=1}^{m} \lambda_i = \sum_{i=1}^{m} s_i^2 \right)$
  - $$m' = \arg \min_{1 \leq k \leq m} \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \geq t \text{ threshold}$$
  - $$m' = \arg \max_{1 \leq i \leq m} \left(\lambda_i > \frac{1}{m} \sum_{j=1}^{m} \lambda_j \right)$$
  - $$m' = \arg \max_{1 \leq i \leq m-1} (\lambda_i - \lambda_{i+1})$$

Summarizing PCA

- Subtract the mean vector from data X and also normalize it somehow
- Calculate the scatter/covariance matrix of the normalized data
- Calculate its eigenvalues
- Form projection matrix P from the eigenvectors corresponding to the m' largest eigenvalues
- $X' = XP$ gives the transformed data
- $X'P^{-1}$ gives an approximation on the original positions of the data points
- A useful tutorial on PCA

Singular Value Decomposition



- $X = U\Sigma V^\top = \sum_{i=1}^{rank(X)} \sigma_i u_i v_i^\top$
- $\|X\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}^2} = \sqrt{\sum_{i=1}^{rank(X)} \sigma_i^2}$

- Low(er) rank approximation of X is $\tilde{X} = U\tilde{\Sigma}V^\top$
- We rely on the top $m' < m$ largest singular value of Σ upon reconstructing $\tilde{X}$
- This is the best possible m'-dimensional approximation of X if we look for the approximation which minimizes $\|X - \tilde{X}\|_{Frobenius}$

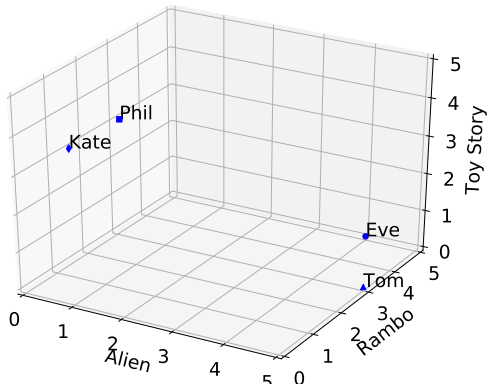
Example SVD

$$\begin{bmatrix} 5 & 3 & 0 \\ 4 & 5 & 0 \\ 1 & 0 & 4 \\ 2 & 0 & 5 \end{bmatrix} =$$

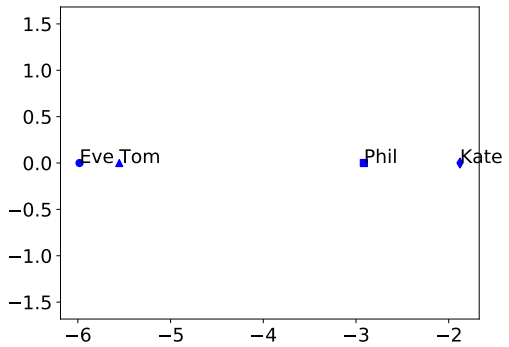
$$\begin{bmatrix} -0.63 & 0.22 & 0.73 \\ -0.67 & 0.33 & -0.65 \\ -0.21 & -0.58 & -0.19 \\ -0.33 & -0.72 & 0.08 \end{bmatrix} \begin{bmatrix} 8.87 & 0 & 0 \\ 0 & 6.33 & 0 \\ 0 & 0 & 1.52 \end{bmatrix} \begin{bmatrix} -0.75 & -0.59 & -0.28 \\ 0.06 & 0.36 & -0.93 \\ 0.65 & -0.72 & -0.24 \end{bmatrix}$$

Possible usage of SVD

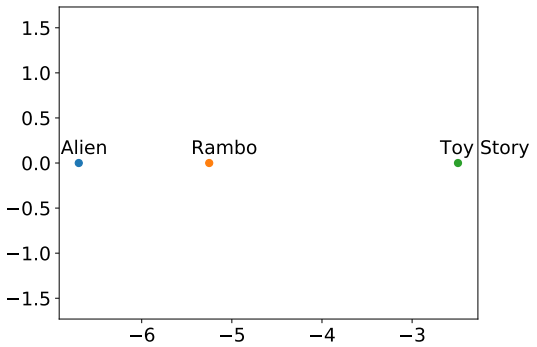
- We can construct a space of *latent topics* using singular vectors
- $X = U\Sigma V^\intercal$ implies $XV = U\Sigma$ and $U^\intercal X = \Sigma V^\intercal$
- We can „add" $\mathrm{x} \notin X$ to the latent space by calculating $\mathrm{x}^\intercal V$ and find similar data points in the latent space



(a) Original user–item ratings in 3D



(b) Rank 1 latent representation of users



(c) Rank 1 latent representation of items

Singular Value Decomposition and Eigendecomposition

Reminder

Any symmetric matrix A is decomposable as $A = X\Lambda X^{-1}$, where $X = [\boldsymbol{x}_1 \boldsymbol{x}_2 \dots \boldsymbol{x}_m]$ comprises of the orthogonal eigenvectors of A and $\Lambda = diag([\lambda_1 \lambda_2 \dots \lambda_m])$ containing the corresponding eigenvalues in its main diagonal. Why?

- Any $n \times m$ matrix X can be uniquely decomposed into the product of three matrices of the form $U\Sigma V^\intercal$ where
  - $U_{n\times n} = [\boldsymbol{u}_1 \boldsymbol{u}_2 \dots \boldsymbol{u}_n]$ is the orthonormal matrix consisting of the eigenvectors of $XX^\intercal$
  - $\Sigma_{n\times m} = diag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_m})$
  - $V_{m\times m} = [\boldsymbol{v}_1 \boldsymbol{v}_2 \dots \boldsymbol{v}_m]$ is the orthonormal matrix consisting of the eigenvectors of $X^\intercal X$ Why?
- Orthogonal matrices: $M^\intercal M = I$ (a transformation which preserves distance in the transformed space as well) Why?

Relation between SVD and Frobenius-norm

- Suppose $M = P \times Q \times R$, i.e. $m_{ij} = \sum_k \sum_l p_{ik} q_{kl} r_{lj}$

$$M = \begin{bmatrix} -58 & 87 \\ -32 & 48 \\ -28 & 42 \end{bmatrix} = \begin{bmatrix} 6 & 3 & 1 \\ 2 & 4 & 2 \\ 3 & 2 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} -2 & 3 \end{bmatrix} \Rightarrow m_{32} = 3*4*3 + 2*1*3 + 0$$

- Then $\|M\|_F^2 = \sum_i \sum_j (m_{ij})^2 = \sum_i \sum_j \left(\sum_k \sum_l p_{ik} q_{kl} r_{lj} \right)^2$
- Also $\left(\sum_k \sum_l p_{ik} q_{kl} r_{lj} \right)^2 = \sum_k \sum_l \sum_m \sum_n p_{ik} q_{kl} r_{lj} p_{in} q_{nm} r_{mj}$
- From where $\|M\|_F^2 = \sum_i \sum_j \sum_k \sum_l \sum_m \sum_n p_{ik} q_{kl} r_{lj} p_{in} q_{nm} r_{mj}$
- Given than matrices P, Q, R originate from an SVD decomposition,
  $\|M\|_F^2 = \sum_{i,j,k,n} p_{ik} q_{kk} r_{kj} p_{in} q_{nn} r_{nj} = \sum_{j,k} q_{kk} r_{kj} q_{kk} r_{kj} = \sum_k (q_{kk})^2.$
- The error of the approximating X by $\tilde{X} = U\tilde{\Sigma}V^\intercal$ is
  $\|X - \tilde{X}\|_F^2 = \|U(\Sigma - \tilde{\Sigma})V^\intercal\|_F^2 = \sum_k (\sigma_{kk} - \tilde{\sigma}_{kk})^2$

CUR

- The drawback of SVD is that a typically sparse matrix is decomposed into a products of dense matrices (i.e. U and V)
- One alternative is to use CUR decomposition
  - This time only matrix U happens to be dense
  - Matrices C and R are composed of the rows and columns of the matrix X, thus they preserve the sparsity of X
  - SVD is unique, unlike CUR

CUR decomposition – producing C and R

- Choose k columns from the data matrix with replacement
  - Potentially, a column can be selected more than once into C
  - The probability of selecting a column should be proportional to the sum of squared elements in it
  - Elements in the selected columns can be scaled by $1/\sqrt{kp_i}$ (kp_i is the expected number of times column i gets selected)
- Construction of R is totally analogous but relies on rows instead of columns

CUR decomposition – producing U

- $U = C^\dagger X R^\dagger$ with † denoting the pseudoinverse operation, hence $CUR = C(C^\dagger X R^\dagger)R = (CC^\dagger)X(R^\dagger R) \approx X$
  - Pseudoinverse is a generalization of „regular" matrix inverse for non-square and/or invertible matrices
    - $MM^\dagger M = M$
    - Given that M is square&invertible $M^{-1} = M$
  - Relation to SVD: $M = U\Sigma V^\intercal \Rightarrow M^\dagger = (U\Sigma V^\intercal)^\dagger = V\Sigma^\dagger U^\intercal$
    - Diagonal matrices are easily invertible

$$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -0.2 \\ 0 & 0 & 0 \end{bmatrix}^\dagger = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -5 & 0 \end{bmatrix}$$

    - It suffices to transpose and take the reciprocal of its nonzero entries

CUR decomposition – example

	Alien	Rambo	Toy Story	P
Tom	5	3	0	$\frac{34}{121}$
Eve	4	5	0	$\frac{41}{121}$
Kate	1	0	4	$\frac{17}{121}$
Phil	2	0	5	$\frac{29}{121}$
P	$\frac{46}{121}$	$\frac{34}{121}$	$\frac{41}{121}$	

$$C = \begin{bmatrix} 5.734 & 0 \\ 4.587 & 0 \\ 1.147 & 4.859 \\ 2.294 & 6.074 \end{bmatrix}$$

$$U = \begin{bmatrix} 0.134 & -0.001 \\ -0.047 & 0.113 \end{bmatrix}$$

$$R = \begin{bmatrix} 6.670 & 5.363 & 0 \\ 2.889 & 0 & 7.222 \end{bmatrix}$$

Linear Discriminant Analysis

- Transform data points into lower dimensions in such a way that points of the same class have as little dispersion as possible whereas points of different classes mix as little as possible
- How should we choose w, i.e. the direction of the projection?
- $\tilde{\mu_c} = \boldsymbol{w}^\mathsf{T}\mu_c \Rightarrow |\tilde{\mu_1} - \tilde{\mu_2}| = |\boldsymbol{w}^\mathsf{T}(\mu_1 - \mu_2)|$
-

$$\tilde{s}_c^2 = \sum_{\{(\boldsymbol{x}_i, y_i)|y_i=c\}} (\boldsymbol{w}^\mathsf{T}\boldsymbol{x} - \tilde{\mu_c})^2$$

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \arg\max_{\boldsymbol{w}} \frac{|\tilde{\mu_1} - \tilde{\mu_2}|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \qquad (1)$$

LDA – Within and outer scatter matrices

- The within-scatter matrix of points for class c:

$$S_c = \sum_{\{(\boldsymbol{x}_i, y_i)|y_i=c\}} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^\mathsf{T}$$

- Aggregated within scatter matrix: $S_W = S_1 + S_2$
- Scatter of the points for class c:

$$\tilde{s}_c^2 = \sum_{\{(\boldsymbol{x}_i, y_i)|y_i=c\}} (\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i - \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu}_c)^2 =$$
$$= \sum_{\{(\boldsymbol{x}_i, y_i)|y_i=c\}} \boldsymbol{w}^\mathsf{T}(\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^\mathsf{T}\boldsymbol{w} = \boldsymbol{w}^\mathsf{T} S_c \boldsymbol{w}$$

- Scatter matrix of the points between different classes:
  $S_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\mathsf{T}$
- Scatter of the points between different classes:

$$(\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_2)^2 = (\boldsymbol{w}^\mathsf{T}\boldsymbol{\mu}_1 - \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu}_2)^2 = \boldsymbol{w}^\mathsf{T} S_B \boldsymbol{w}$$

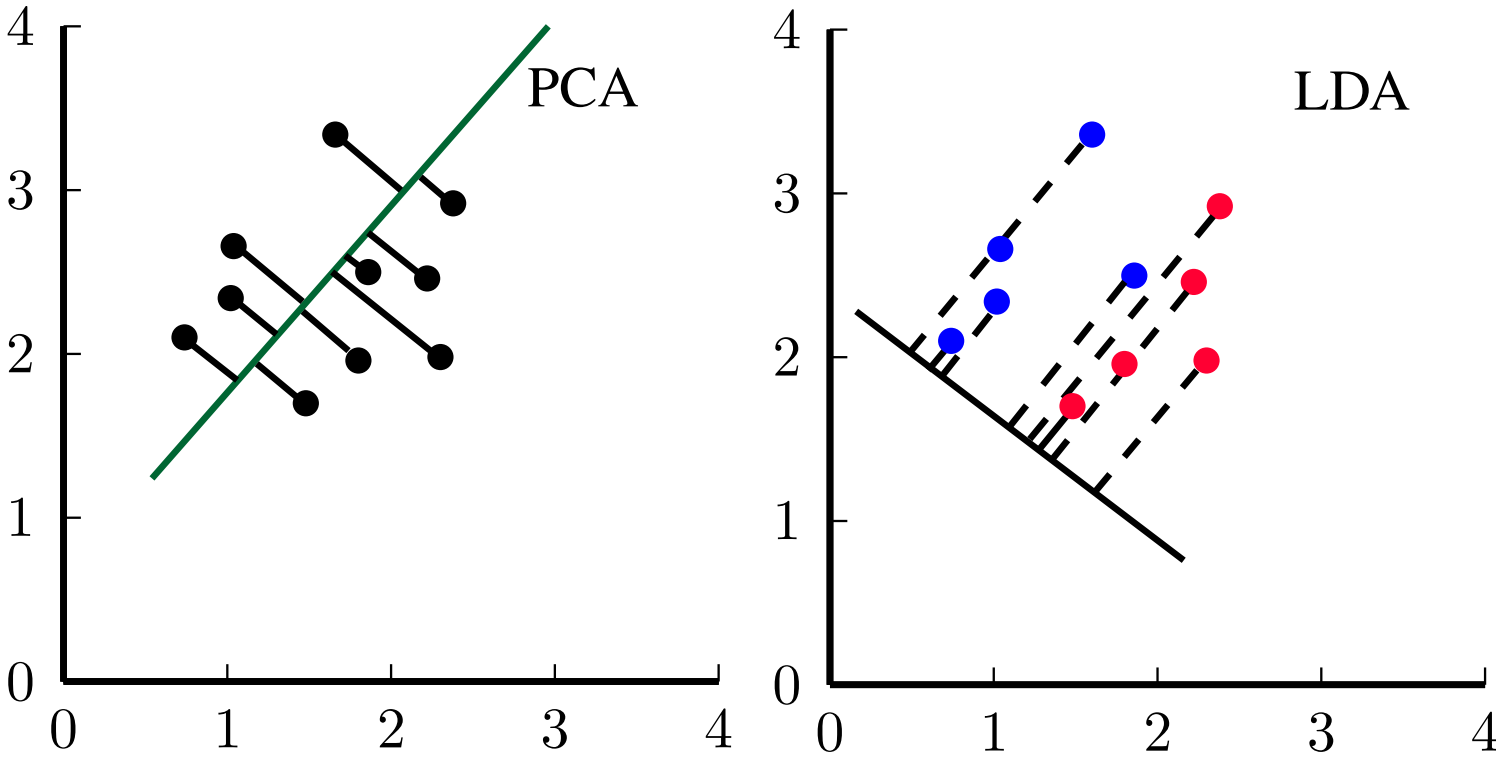
LDA

- An equivalent objective with Eq. (1) is
  $\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \arg\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^\mathsf{T} S_B \boldsymbol{w}}{\boldsymbol{w}^\mathsf{T} S_W \boldsymbol{w}}$
  - $\frac{\boldsymbol{w}^\mathsf{T} S_B \boldsymbol{w}}{\boldsymbol{w}^\mathsf{T} S_W \boldsymbol{w}}$ is the so-called generalized Rayleigh-coefficient
- $J(\boldsymbol{w})$ is maximal $\Rightarrow \nabla \frac{\boldsymbol{w}^\mathsf{T} S_B \boldsymbol{w}}{\boldsymbol{w}^\mathsf{T} S_W \boldsymbol{w}} = 0 \Leftrightarrow S_B \boldsymbol{w} = \lambda S_W \boldsymbol{w} \Leftrightarrow$
  $S_W^{-1} S_B \boldsymbol{w} = \lambda \boldsymbol{w} \Leftrightarrow \boldsymbol{w} = S_W^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$

Reminder

- $\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{g^2(x)}$
- $\nabla_{\boldsymbol{x}} \boldsymbol{x}^\mathsf{T} A \boldsymbol{x} = 2A\boldsymbol{x}$, given that $A = A^\mathsf{T}$
- $\boldsymbol{x}\boldsymbol{x}^\mathsf{T} y = \left(\sum_{i=1}^{n} x_i y_i\right) \boldsymbol{x}$ (i.e. a vector pointing in the direction of $\boldsymbol{x}$)

LDA vs. PCA

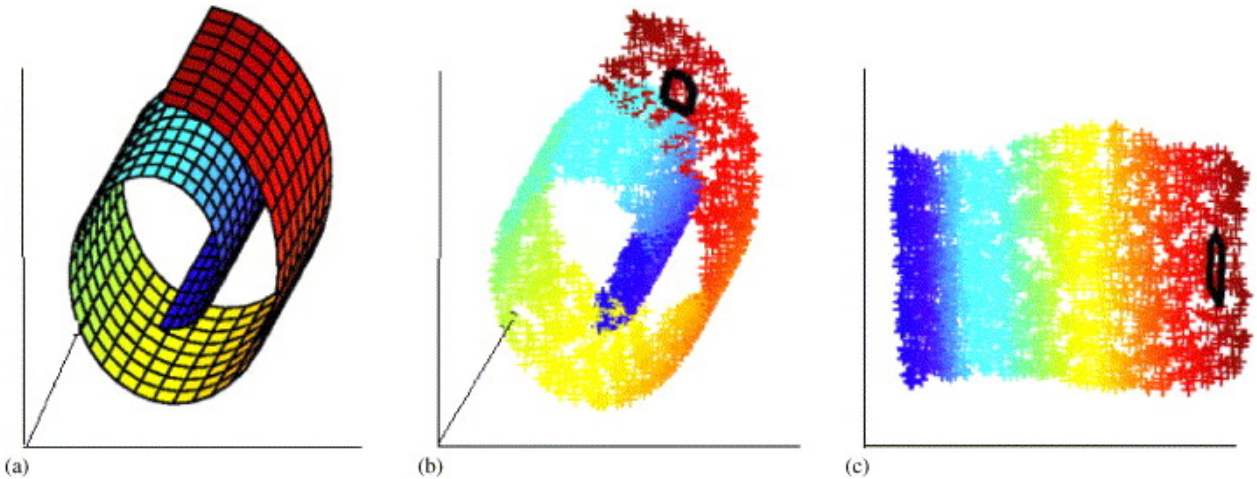


Multi-Dimensional Scaling (MDS)

- Goal: given Δ containing pair-wise cost/distances of points find the positions of the points for which $\| x_i - x_j \| \approx \delta_{ij}$
- Transforms multidimensional points into lower dimensions such that the pairwise distances get preserved as much as possible

Locally Linear Embedding (LLE)

- PCA, SVD and LDA all assume linear relationship between variables
- Non-linear dimensionality reduction technique
- Idea: define the nearest neighbors for all points and define them as their linear combination
- $J(W) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \sum_{j=1}^{n} W_{ij}\boldsymbol{x_j}\|^2$, such that $\sum_{j=1}^{n} W_{ij} = 1$ and $w_{ij} > 0 \Leftrightarrow x_j \in neighbors(x_i)$



Canonical Correlation Analysis (CCA)

- Our data points have two distinct representations (coordinate systems)
- Goal: find a common coordinate system (with reduced dimensionality) such that the correlation between the transformed points get maximized
- $\rho = \frac{\mathbb{E}[xy]}{\sqrt{\mathbb{E}[x^2]\,\mathbb{E}[y^2]}} = \frac{\mathbb{E}[w_x^\mathsf{T} x y^\mathsf{T} w_y]}{\sqrt{\mathbb{E}[w_x^\mathsf{T} x x^\mathsf{T} w_x^\mathsf{T}]\,\mathbb{E}[w_y^\mathsf{T} y y^\mathsf{T} w_y^\mathsf{T}]}} = \frac{w_x^\mathsf{T} C_{xy} w_y}{\sqrt{w_x^\mathsf{T} C_{xx} w_x\, w_y^\mathsf{T} C_{yy} w_y}}$
- $\arg\max \rho$ is independent from the length of w_x and $w_y \Rightarrow \arg\max \rho = \arg\max w_x^\mathsf{T} C_{xy} w_y$
- $\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} = \mathbb{E}\left[\begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}^\mathsf{T} \right]$