Dimensionality Reduction

1

Why Reduce Dimensionality?

- Reduces time complexity: Less computation
- Reduces space complexity: Less parameters
- Saves the cost of observing the feature
- Simpler models are more robust on small datasets
- More interpretable; simpler explanation
- Data visualization (structure, groups, outliers, etc) if plotted in 2 or 3 dimensions

2

2

Feature Selection vs Extraction

- Feature selection: Choosing $k<d$ important features, ignoring the remaining $d - k$
  Subset selection algorithms
- Feature extraction: Project the
  original x_i , i =1,...,d dimensions to
  new $k<d$ dimensions, z_j , j =1,...,k
  Principal components analysis (PCA), linear discriminant analysis (LDA), factor analysis (FA)

3

3

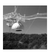
Subset Selection

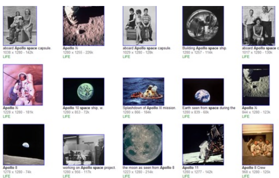
- There are 2^d subsets of d features
- Forward search: Add the best feature at each step
  - Set of features F initially $\emptyset$.
  - At each iteration, find the best new feature
    $j = \text{argmin}_i\, E(F \cup x_i)$
  - Add x_j to F if $E(F \cup x_j) < E(F)$
- Hill-climbing $O(d^2)$ algorithm
- Backward search: Start with all features and remove one at a time, if possible.
- Floating search (Add k, remove l)

4

4

Recall: Representing images



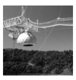


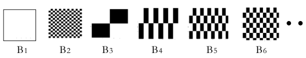
- The most common element in the image: background
  - Or rather large regions of relatively featureless shading
  - Uniform sequences of numbers

5

5

Adding more bases



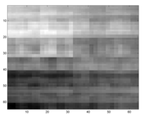


- Checkerboards with different variations

$$\text{Image} \approx w_1 B_1 + w_2 B_2 + w_3 B_3 + \ldots$$

$$W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ . \\ . \end{bmatrix} \qquad B = [B_1 \ B_2 \ B_3]$$

$$BW \approx \text{Image}$$
$$W = pinv(B)\,\text{Image}$$
$$PROJECTION = BW$$

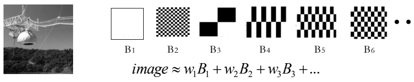


Getting closer at 625 bases!

6

6

"Bases"



$$image \approx w_1 B_1 + w_2 B_2 + w_3 B_3 + ...$$

- "Bases" are the "standard" units such that all instances can be expressed a weighted combinations of these units
- Ideal requirements: Bases must be orthogonal
- Checkerboards are one choice of bases
  - Orthogonal
  - But not "smooth"
- Other choices of bases: Complex exponentials, Wavelets, etc..

7

Data specific bases?

- **Issue**: **The bases we have considered so far are *data agnostic***
  - Checkerboards, Complex exponentials, Wavelets..
  - We use the same bases regardless of the data we analyze
    - Image of face vs. Image of a forest
    - Segment of speech vs. Seismic rumble
- How about data specific bases
  - Bases that consider the underlying data
    - E.g. is there something better than checkerboards to describe faces
    - Something better than complex exponentials to describe music?

8

Principal Components Analysis (PCA)

- Find a low-dimensional space such that when $\boldsymbol{x}$ is projected there, information loss is minimized.
- The projection of $\boldsymbol{x}$ on the direction of $\boldsymbol{w}$ is: $z = \boldsymbol{w}^T\boldsymbol{x}$
- Find $\boldsymbol{w}$ such that Var(z) is maximized

$$\begin{aligned}\mathrm{Var}(z) &= \mathrm{Var}(\boldsymbol{w}^T\boldsymbol{x}) = \mathrm{E}[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})^2] \\ &= \mathrm{E}[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})] \\ &= \mathrm{E}[\boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{w}] \\ &= \boldsymbol{w}^T\,\mathrm{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T]\boldsymbol{w} = \boldsymbol{w}^T \Sigma \boldsymbol{w}\end{aligned}$$

where $\mathrm{Var}(\boldsymbol{x}) = \mathrm{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T] = \Sigma$

9

Principal Components Analysis (PCA)

- Maximize Var(z) subject to $||\mathbf{w}||=1$

$$\max_{\mathbf{w}_1} \mathbf{w}_1^T \Sigma \mathbf{w}_1 - \alpha\left(\mathbf{w}_1^T \mathbf{w}_1 - 1\right)$$

$\Sigma \mathbf{w}_1 = \alpha \mathbf{w}_1$ that is, $\mathbf{w}_1$ is an eigenvector of Σ

Choose the one with the largest eigenvalue for Var(z) to be max

- Second principal component: Max Var(z_2), s.t., $||\mathbf{w}_2||=1$ and orthogonal to $\mathbf{w}_1$

$$\max_{\mathbf{w}_2} \mathbf{w}_2^T \Sigma \mathbf{w}_2 - \alpha\left(\mathbf{w}_2^T \mathbf{w}_2 - 1\right) - \beta\left(\mathbf{w}_2^T \mathbf{w}_1 - 0\right)$$

$\Sigma \mathbf{w}_2 = \alpha\, \mathbf{w}_2$ that is, $\mathbf{w}_2$ is another eigenvector of Σ

and so on.

10

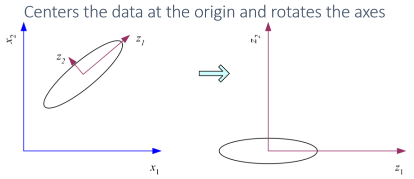
10

What PCA does

$$z = \mathrm{W}^T(x - m)$$

where the columns of W are the eigenvectors of Σ, and m is sample mean

Centers the data at the origin and rotates the axes



11

11

How to choose k ?

- Proportion of Variance (PoV) explained

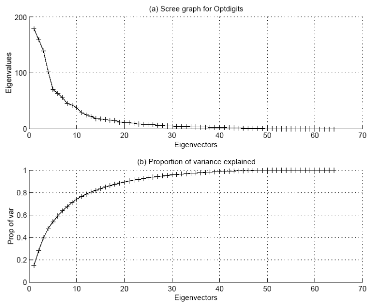
$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

when λ_i are sorted in descending order

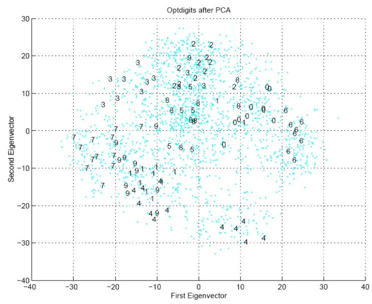
- Typically, stop at PoV>0.9
- Scree graph plots of PoV vs k, stop at "elbow"

12

12

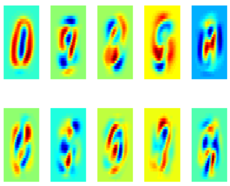

(a) Scree graph for Optdigits
Eigenvalues
Eigenvectors
(b) Proportion of variance explained
Prop of var
Eigenvectors

13


Optdigits after PCA
Second Eigenvector
First Eigenvector

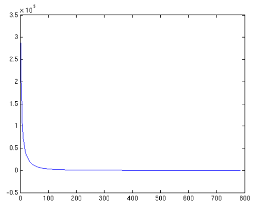
14

PCA



15

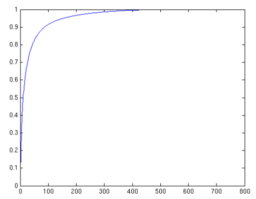
PCA : Eigen Values



16

16

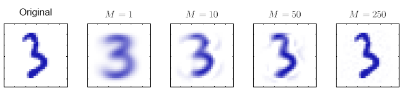
PCA : % information



17

17

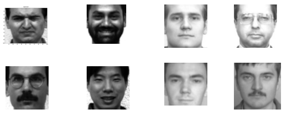
PCA: Reconstruction



18

18

Data-specific description of faces



- A collection of images
  - All normalized to 100x100 pixels
- What is common among all of them?
  - Do we have a common descriptor?

19

19

Eigen Faces!

- Arrange your input data into a matrix $\mathbf{X}$
- Compute the correlation $\mathbf{R} = \mathbf{X}\mathbf{X}^T$
- Solve the Eigen decomposition: $\mathbf{R}\mathbf{V} = \Lambda\mathbf{V}$
- The Eigen vectors corresponding to the K largest eigen values are our optimal bases
- We will refer to these as *eigen faces*.

20

20

Class specificity

- Eigen bases are class specific

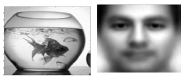
- Composing a fishbowl from Eigenfaces

21

21

Class specificity

- Eigen bases are class specific



- Composing a fishbowl from Eigenfaces
- With 1 basis

$$f = w_1 \mathbf{v}_1$$

22

22

Class specificity

- Eigen bases are class specific

- Composing a fishbowl from Eigenfaces
- With 10 bases

$$f = w_1 \mathbf{v}_1 + w_2 \mathbf{v}_2 + \ldots + w_{10} \mathbf{v}_{10}$$

23

23

Class specificity

- Eigen bases are class specific

- Composing a fishbowl from Eigenfaces
- With 30 bases

$$f = w_1 \mathbf{v}_1 + w_2 \mathbf{v}_2 + \ldots + w_{10} \mathbf{v}_{10} + \ldots + w_{30} \mathbf{v}_{30}$$

24

24

Class specificity

- Eigen bases are class specific

- Composing a fishbowl from Eigenfaces
- With 100 bases

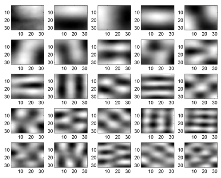
$$f = w_1\mathbf{v}_1 + w_2\mathbf{v}_2 + \ldots + w_{10}\mathbf{v}_{10} + \ldots + w_{30}\mathbf{v}_{30} + \ldots + w_{100}\mathbf{v}_{100}$$

25

25

Universal bases

- Universal bases..



- End up looking a lot like *discrete cosine transforms!!!!*
- *DCTs are the best "universal" bases*
  - *If you don't know what your data are, use the DCT*

26

26

Factor Analysis

- Find a small number of factors $\boldsymbol{z}$, which when combined generate $\boldsymbol{x}$:

  $x_i - \mu_i = v_{i1}z_1 + v_{i2}z_2 + \ldots + v_{ik}z_k + \varepsilon_i$

  where z_j, $j = 1,\ldots,k$ are the latent factors with

  $E[z_j]=0$, $Var(z_j)=1$, $Cov(z_i, z_j)=0$, $i \neq j$,

  ε_i are the noise sources

  $E[\varepsilon_i] = \psi_i$, $Cov(\varepsilon_i, \varepsilon_j) = 0$, $i \neq j$, $Cov(\varepsilon_i, z_j) = 0$,

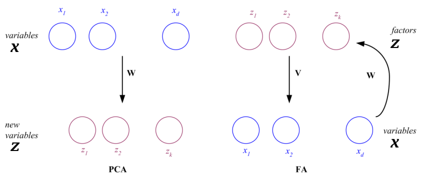
  and v_{ij} are the factor loadings

27

27

PCA vs FA

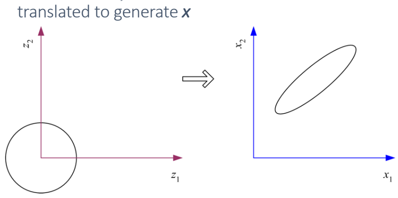
- PCA From $\boldsymbol{x}$ to $\boldsymbol{z}$ $\quad\boldsymbol{z} = \mathbf{W}^T(\boldsymbol{x} - \boldsymbol{\mu})$
- FA $\quad$ From $\boldsymbol{z}$ to $\boldsymbol{x}$ $\quad\boldsymbol{x} - \boldsymbol{\mu} = \mathbf{V}\boldsymbol{z} + \boldsymbol{\varepsilon}$



28

Factor Analysis

- In FA, factors z_j are stretched, rotated and translated to generate $\boldsymbol{x}$



29

29

Multidimensional Scaling

- Given pairwise distances between N points,
  d_{ij}, $i,j = 1,\dots,N$
  place on a low-dim map s.t. distances are preserved.
- $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x} \mid \vartheta)$ $\quad$ Find ϑ that min Sammon stress

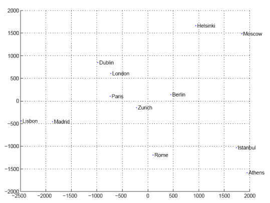
$$E(\theta \mid \mathcal{X}) = \sum_{r,s} \frac{\left(\|\mathbf{z}^r - \mathbf{z}^s\| - \|\mathbf{x}^r - \mathbf{x}^s\|\right)^2}{\|\mathbf{x}^r - \mathbf{x}^s\|^2}$$

$$= \sum_{r,s} \frac{\left(\|\mathbf{g}(\mathbf{x}^r \mid \theta) - \mathbf{g}(\mathbf{x}^s \mid \theta)\| - \|\mathbf{x}^r - \mathbf{x}^s\|\right)^2}{\|\mathbf{x}^r - \mathbf{x}^s\|^2}$$

30

30

Map of Europe by MDS



31

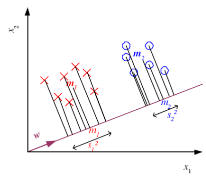
31

Linear Discriminant Analysis

- Find a low-dimensional space such that when $\boldsymbol{x}$ is projected, classes are well-separated.
- Find $\boldsymbol{w}$ that maximizes

$$J(\mathbf{w})=\frac{(m_1-m_2)^2}{s_1^2+s_2^2}$$

$$m_1=\frac{\sum_t \mathbf{w}^T\mathbf{x}^t r^t}{\sum_t r^t} \qquad s_1^2=\sum_t\left(\mathbf{w}^T\mathbf{x}^t-m_1\right)^2 r^t$$



32

32

Linear Discriminant Analysis

- Between-class scatter:

$$(m_1-m_2)^2=\left(\mathbf{w}^T\mathbf{m}_1-\mathbf{w}^T\mathbf{m}_2\right)^2$$
$$=\mathbf{w}^T(\mathbf{m}_1-\mathbf{m}_2)(\mathbf{m}_1-\mathbf{m}_2)^T\mathbf{w}$$
$$=\mathbf{w}^T\mathbf{S}_B\mathbf{w} \text{ where } \mathbf{S}_B=(\mathbf{m}_1-\mathbf{m}_2)(\mathbf{m}_1-\mathbf{m}_2)^T$$

- Within-class scatter:

$$s_1^2=\sum_t\left(\mathbf{w}^T\mathbf{x}^t-m_1\right)^2 r^t$$
$$=\sum_t \mathbf{w}^T\left(\mathbf{x}^t-\mathbf{m}_1\right)\left(\mathbf{x}^t-\mathbf{m}_1\right)^T\mathbf{w}r^t=\mathbf{w}^T\mathbf{S}_1\mathbf{w}$$
$$\text{where } \mathbf{S}_1=\sum_t\left(\mathbf{x}^t-\mathbf{m}_1\right)\left(\mathbf{x}^t-\mathbf{m}_1\right)^T r^t$$
$$s_1^2+s_1^2=\mathbf{w}^T\mathbf{S}_W\mathbf{w} \text{ where } \mathbf{S}_W=\mathbf{S}_1+\mathbf{S}_2$$

33

33

Fisher's Linear Discriminant

- Find $\mathbf{w}$ that max

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \frac{\left|\mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)\right|^2}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

- LDA soln:

$$\mathbf{w} = c \cdot \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

- Parametric soln:

$$\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2)$$

when $p(\mathbf{x} \mid C_i) \sim \mathcal{N}(\mu_i, \Sigma)$

34

34

K>2 Classes

- Within-class scatter:

$$\mathbf{S}_W = \sum_{i=1}^{K} \mathbf{S}_i \qquad \mathbf{S}_i = \sum_t r_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T$$

- Between-class scatter:

$$\mathbf{S}_B = \sum_{i=1}^{K} N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \qquad \mathbf{m} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{m}_i$$

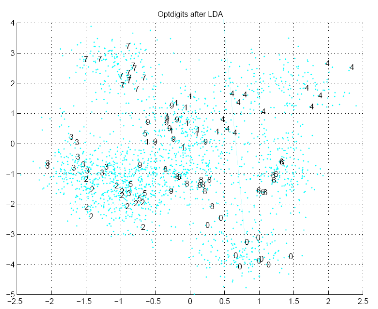
- Find $\mathbf{W}$ that max

$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|}$$

The largest eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$
Maximum rank of K-1

35

35



36

36

Neighborhood Components Analysis

- NCA learns a Mahalanobis distance metric for the KNN classifier by maximizing the leave-one-out cross validation.
  - Let p_{ij} be the probability that point x_j is selected as point x_i's neighbour.
  - The probability that points are correctly classified when x_i is used as the reference is:

$$P_i = \sum_{j \in C_i} p_{ij}$$
$$C_i = \{x_j | \text{class}(x_j) = \text{class}(x_i)\}$$

  - p_{ij} is related to the distance between x_i and x_j

$$p_{ij} = \frac{e^{-d_{ij}}}{\sum_{k \neq i} e^{-d_{ik}}}, \, p_{ii} = 0$$

  - The expected number of correctly classification points:

$$f(A) = \sum_i P_i$$

37

Neighborhood Components Analysis

- How do we define d_{ij}?
- Limit the distance measure within Mahalanobis (quadratic) distance:

$$d(x, y) = (x - y)^T Q (x - y)$$
$$Q = A^T A$$
$$d(x, y) = (Ax - Ay)^T (Ax - Ay)$$

- That is to say, we project the original feature vectors x into another vector space with transformation matrix A.

Distance Learning ⇔ Finding the Best matrix A

Réda DEHAK 38

38

Neighborhood Components Analysis

- Subtitute d_{ij} in p_{ij}

$$p_{ij} = \frac{\exp\left(-\|Ax_i - Ax_j\|^2\right)}{\sum_{k \neq i} \exp(-\|Ax_i - Ax_k\|^2)}$$

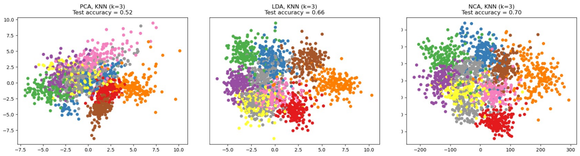
- The **objective function**:

$$f(A) = \sum_i P_i = \sum_i \sum_{j \in C_i} p_{ij}$$

Réda DEHAK 39

39

Neighborhood Components Analysis



Réda DEHAK 40

40

t-SNE

- SNE - Stochastic Neighbor Embedding
- t-SNE
- KL divergence

Réda DEHAK 41

41

Local Embedding

- t-SNE is an alternative dimensionality reduction algorithm.
- PCA tries to find a **global** structure
  - Low dimensional subspace
  - Can lead to local inconsistencies
    - Far away point can become nearest neighbors
- t-SNE tries to preserve **local** structure
  - Low dimensional neighborhood should be the same as original neighborhood.
- Unlike PCA almost only used for visualization
  - No easy way to embed new points

42

Stochastic Neighbor Embedding (SNE)

- ”Encode” high dimensional neighborhood information as a distribution
- Intuition: Random walk between data points.
  - High probability to jump to a close point
- Find low dimensional points such that their neighborhood distribution is similar
- How do you measure distance between distributions?
  - Most common measure: KL divergence

43

Neighborhood Distributions

- Consider the neighborhood around an input data point $x_i \in \mathbb{R}^d$
- Imagine that we have a Gaussian distribution centered around x_i
- Then the probability that x_i chooses some other datapoint x_j as its neighbor is in proportion with the density under this Gaussian
- A point closer to x_i will be more likely than one further away
- $P_{j|i}$ probability (similar to NCA), is the probability that point x_i chooses x_j as its neighbor:

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

44

Neighborhood Distributions

- $P_{j|i}$ probability (similar to NCA), is the probability that point x_i chooses x_j as its neighbor:

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

With $P_{i|i} = 0$

- The parameter σ_i sets the size of the neighborhood
  - Very low σ_i: all the probability is in the nearest neighbor.
  - Very high σ_i: Uniform weights.
- We set σ_i differently for each data point
- Results depend heavily on σ_i:
  - it defines the neighborhoods we are trying to preserve.
- Final distribution over pairs is symmetrized: $P_{ij} = \frac{1}{2N}\left(P_{i|j} + P_{j|i}\right)$
- Random Walk:
  - Pick i uniformly and then ”jump” to j according to $P_{j|i}$

45

SNE objective

- Given $x_1, x_2, \dots, x_N \in \mathbb{R}^d$, we define the distribution P_{ij}
- Goal: Find good embedding $y_1, y_2, \dots, y_N \in \mathbb{R}^p$ for some $p \ll d$ (normally 2 or 3 for visualization)
- How do we measure an embedding quality?
- For points $y_1, y_2, \dots, y_N \in \mathbb{R}^p$, we can define distribution Q similarly the same (notice no σ_i^2 and not symmetric)

$$Q_{ij} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}$$

- Optimize Q to be close to P
  - Minimize KL-divergence
- The embeddings $y_1, y_2, \dots, y_N \in \mathbb{R}^p$ are the parameters we are optimizing.
  - How do you embed a new point? No embedding function!

46

SNE objective

- Given $x_1, x_2, \dots, x_N \in \mathbb{R}^d$, we define the distribution P_{ij}
- Goal: Find good embedding $y_1, y_2, \dots, y_N \in \mathbb{R}^p$ for some $p \ll d$ (normally 2 or 3 for visualization)
- How do we measure an embedding quality?
- For points $y_1, y_2, \dots, y_N \in \mathbb{R}^p$, we can define distribution Q similarly the same (notice no σ_i^2 and not symmetric)

$$Q_{ij} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}$$

- Optimize Q to be close to P
  - Minimize KL-divergence
- The embeddings $y_1, y_2, \dots, y_N \in \mathbb{R}^p$ are the parameters we are optimizing.
  - How do you embed a new point? No embedding function!

47

SNE algorithm

- We have P, and are looking for $y_1, y_2, \dots, y_N \in \mathbb{R}^p$ such that the distribution Q we infer will minimize $\mathbb{C}(Q) = \mathrm{KL}(P\|Q)$(notice Q on right)

$$\mathrm{KL}(P\|Q) = \sum_{ij} P_{ij} \log\left(\frac{P_{ij}}{Q_{ij}}\right) = -\sum_{ij} P_{ij} \log(Q_{ij}) + const$$

- Not a convex problem! can use multiple restarts.
- Main issue: crowding problem

48

Crowding Problem: Neighborhood

- In high dimension we have more room, points can have a lot of different neighbors
- In 2D a point can have a few neighbors at distance one all far from each other
  - what happens when we embed in 1D?
- Crowding problem:
  - We don't have enough room to accommodate all neighbors.
  - One of the biggest problems with SNE.
- t-SNE solution: Change the Gaussian in Q to a heavy tailed distribution.
- if Q changes slower, we have more wiggle room to place points at.

49

T-SNE: t-Distributed Stochastic Neighbor Embedding

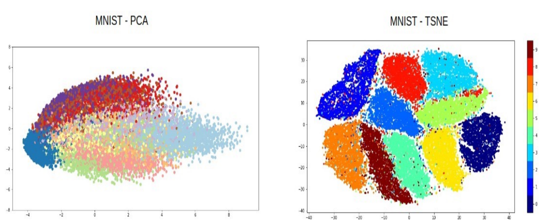
- Student-t Probability density $p(x) \propto \left(1 + \frac{x^2}{v}\right)^{-\frac{v+1}{2}}$
  - For $v = 1$ we get $p(x) \propto \frac{1}{1+x^2}$
- Probability goes to zero much slower than a Gaussian
- We can now redefine Q_{ij} as

$$Q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|y_k - y_l\|^2)^{-1}}$$

- We leave P_{ij} as is

50

PCA vs t-SNE



Réda DEHAK

51

51

Conclusions

- A lot of tools can be used for dimensionality reduction linear and not linear.
- Helpful for visualization
- Helps to understand ”black-box” algorithms
- t-SNE is a great way to visualize data

Réda DEHAK 52

52