Machine Learning and its Applications

Dimensionality Reduction: PCA & T-SNE

Urban Information Lab

Introduction

- Dimensionality reduction is the process of taking data in a high dimensional space and mapping it into a new space whose dimensionality is much smaller.
- This process is closely related to the concept of (lossy) compression in information theory.
- There are several reasons to reduce the dimensionality of the data.
- First, high dimensional data impose computational challenges.
- Moreover, in some situations high dimensionality might lead to poor generalization abilities of the learning algorithm.
- Finally, dimensionality reduction can be used for interpretability of the data, for finding meaningful structure of the data, and for illustration purposes.

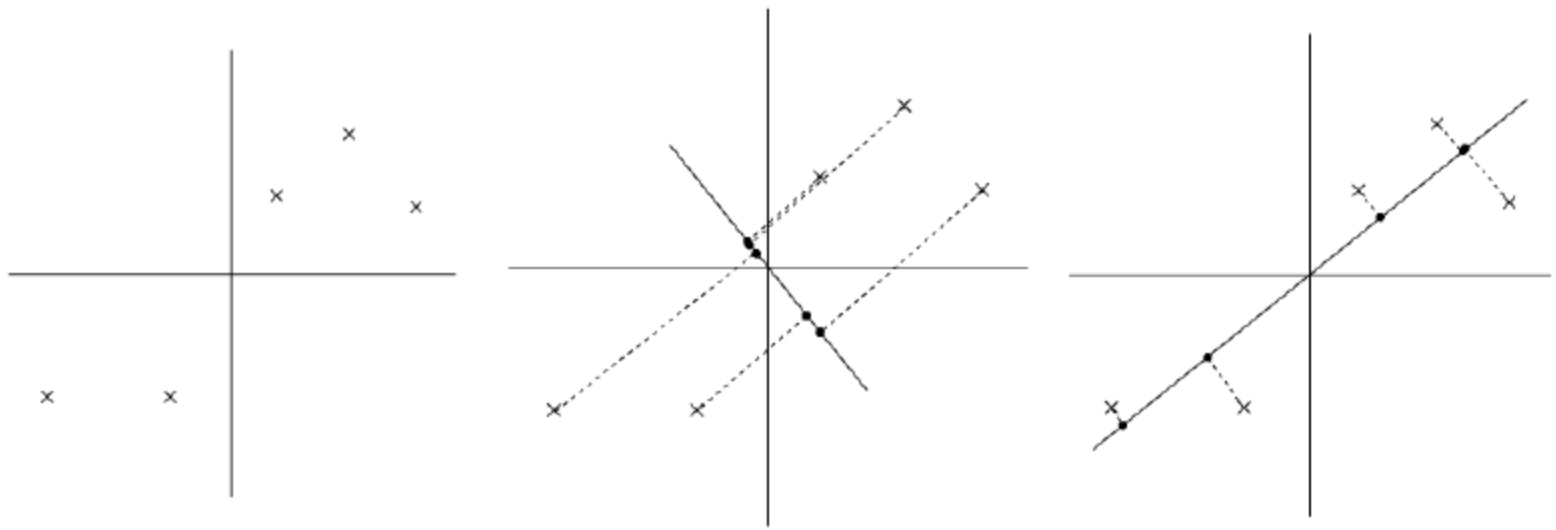
PCA (Principal Component Analysis)

- Both the compression and the recovery are performed by linear transformations.
- The method finds the linear transformations for which the differences between the recovered vectors and the original vectors are minimal in the least squared sense.

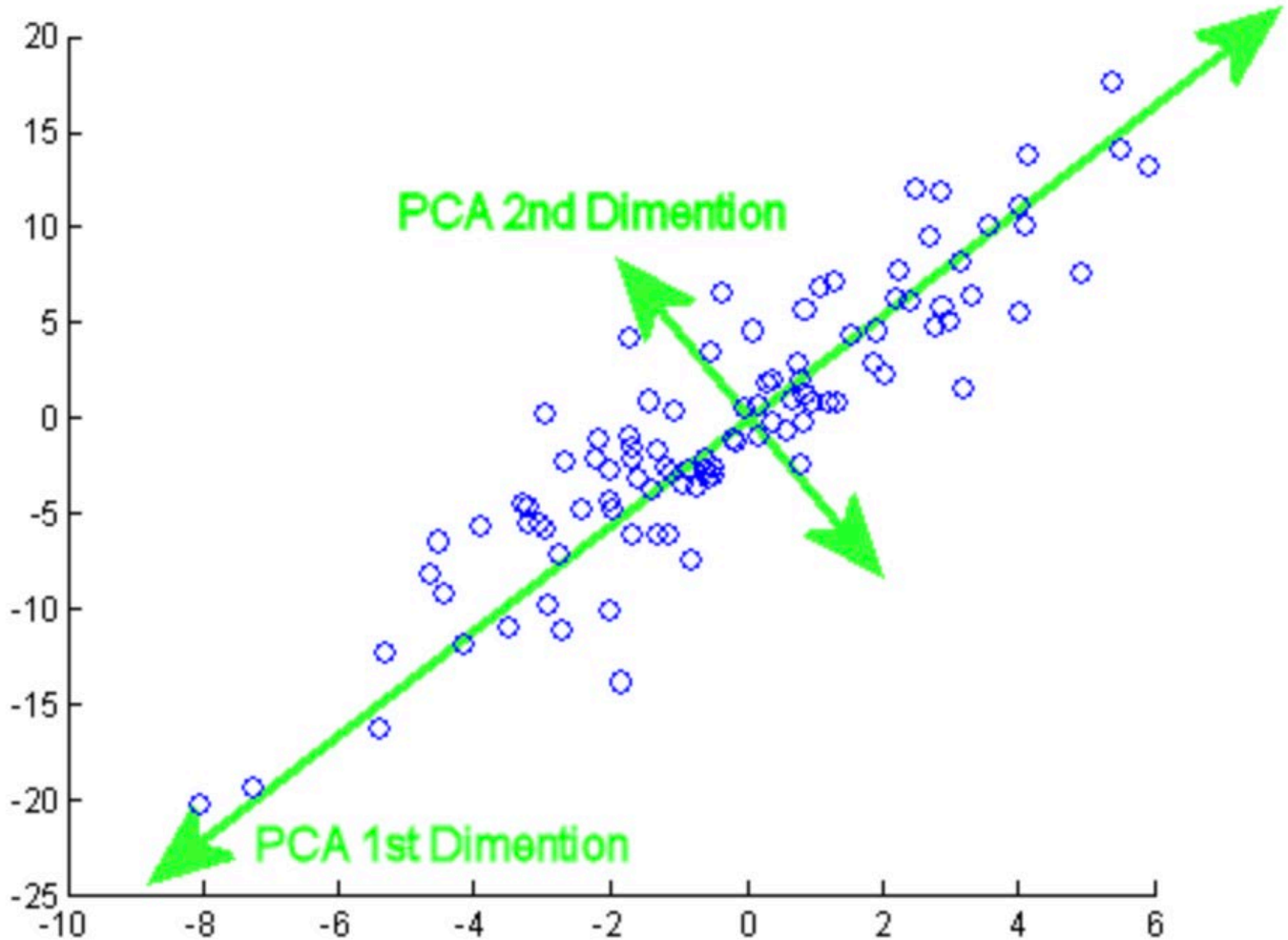
Lower dimension Projections

- Obtain a new feature vector by transforming the original features $x_1, x_2, ..., x_n$
- New features are linear combinalons of old ones
- Unsupervised Setting!

Which projection is better?



From notes by Andrew Ng


PCA 2nd Dimention
PCA 1st Dimention
20
15
10
5
0
-5
-10
-15
-20
-25
-10
-8
-6
-4
-2
0
2
4
6

Using a new basis for the data

- Project a point into a (lower dimensional) space:
  - **point**: $\mathbf{x} = (x_1,\ldots,x_n)$
  - **select a basis** – set of unit (length 1) basis vectors $(\mathbf{u}_1,\ldots,\mathbf{u}_k)$
    - we consider orthonormal basis:
      - $\mathbf{u}_j \bullet \mathbf{u}_j = 1$, and $\mathbf{u}_j \bullet \mathbf{u}_l = 0$ for $j \neq l$
  - **select a center** – $\bar{\mathbf{x}}$, defines offset of space
  - **best coordinates** in lower dimensional space defined by dot-products: $(z_1,\ldots,z_k)$, $z_j^i = (\mathbf{x}^i - \bar{\mathbf{x}}) \bullet \mathbf{u}_j$

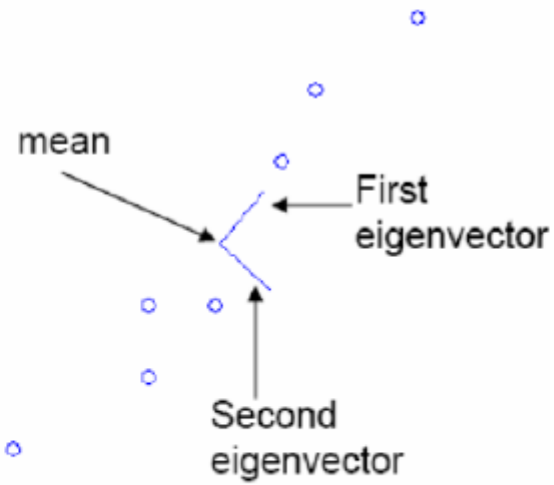
$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

PCA example

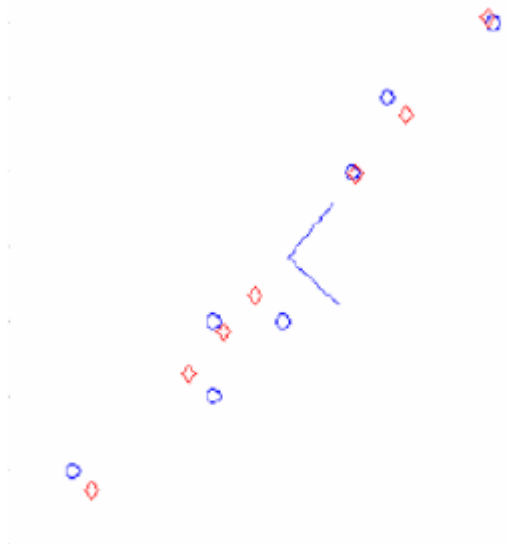
$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

Data:

Projection:



Reconstruction:



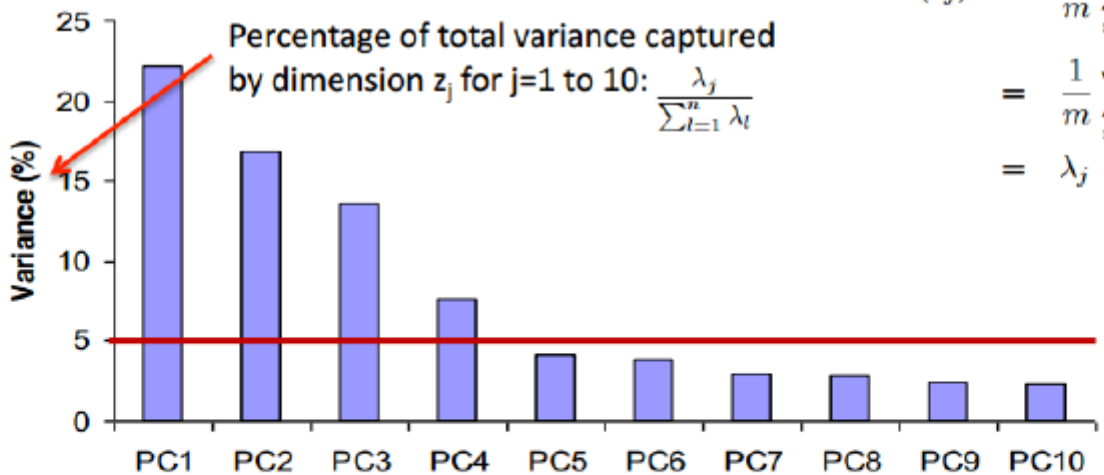
Dimensionality reduction with PCA

In high-dimensional problem, data usually lies near a linear subspace, as noise introduces small variability

Only keep data projections onto principal components with **large** eigenvalues

Can *ignore* the components of lesser significance.

$$\begin{aligned} \mathrm{var}(z_j) &= \frac{1}{m}\sum_{i=1}^{m}(z_j^i)^2 \\ &= \frac{1}{m}\sum_{i=1}^{m}(x^i \cdot u_j)^2 \\ &= \lambda_j \end{aligned}$$



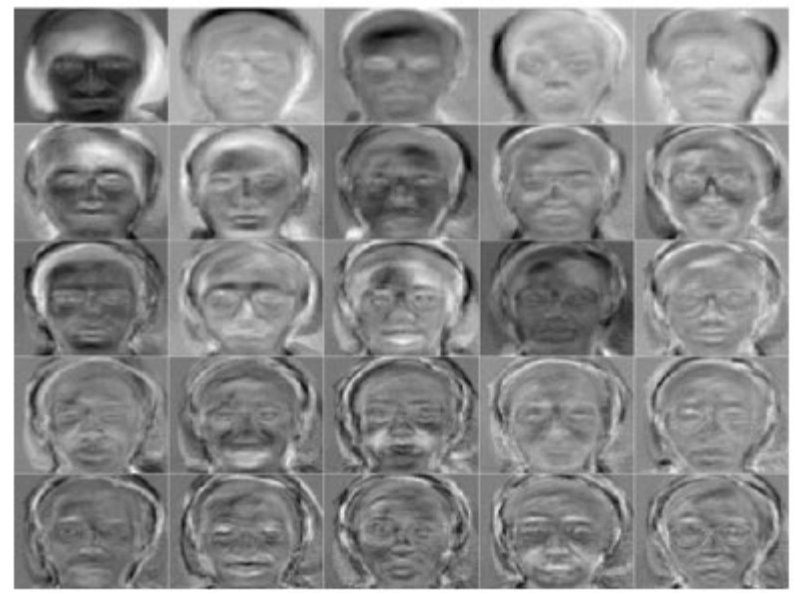
You might lose some information, but if the eigenvalues are small, you don't lose much

Slide from Aarti Singh

Eigenfaces [Turk, Pentland '91]

- Input images:

- Principal components:



Eigenfaces reconstruction

- Each image corresponds to adding together (weighted versions of) the principal components:

Scaling up

- Covariance matrix can be really big!
  - Σ is n by n
  - 10000 features can be common!
  - finding eigenvectors is very slow…
- Use singular value decomposition (SVD)
  - Finds k eigenvectors
  - great implementations available, e.g., Matlab svd

SVD

- Write $\mathbf{X} = \mathbf{Z}\,\mathbf{S}\,\mathbf{U}^T$
  - $\mathbf{X}$ ← data matrix, one row per datapoint
  - $\mathbf{S}$ ← singular value matrix, diagonal matrix with entries σ_i
    - Relationship between singular values of $\mathbf{X}$ and eigenvalues of Σ given by $\lambda_i = \sigma_i^2/m$
  - $\mathbf{Z}$ ← weight matrix, one row per datapoint
    - $\mathbf{Z}$ times $\mathbf{S}$ gives coordinate of x_i in eigenspace
  - $\mathbf{U}^T$ ← singular vector matrix
    - In our setting, each row is eigenvector $\mathbf{u}_j$

PCA using SVD algorithm

- Start from m by n data matrix **X**
- **Recenter**: subtract mean from each row of **X**
  - $\mathbf{X}_c \leftarrow \mathbf{X} - \overline{\mathbf{X}}$
- **Call SVD** algorithm on $\mathbf{X}_c$ – ask for k singular vectors
- **Principal components:** k singular vectors with highest singular values (rows of $\mathbf{U}^T$)
  - **Coefficients:** project each point onto the new vectors

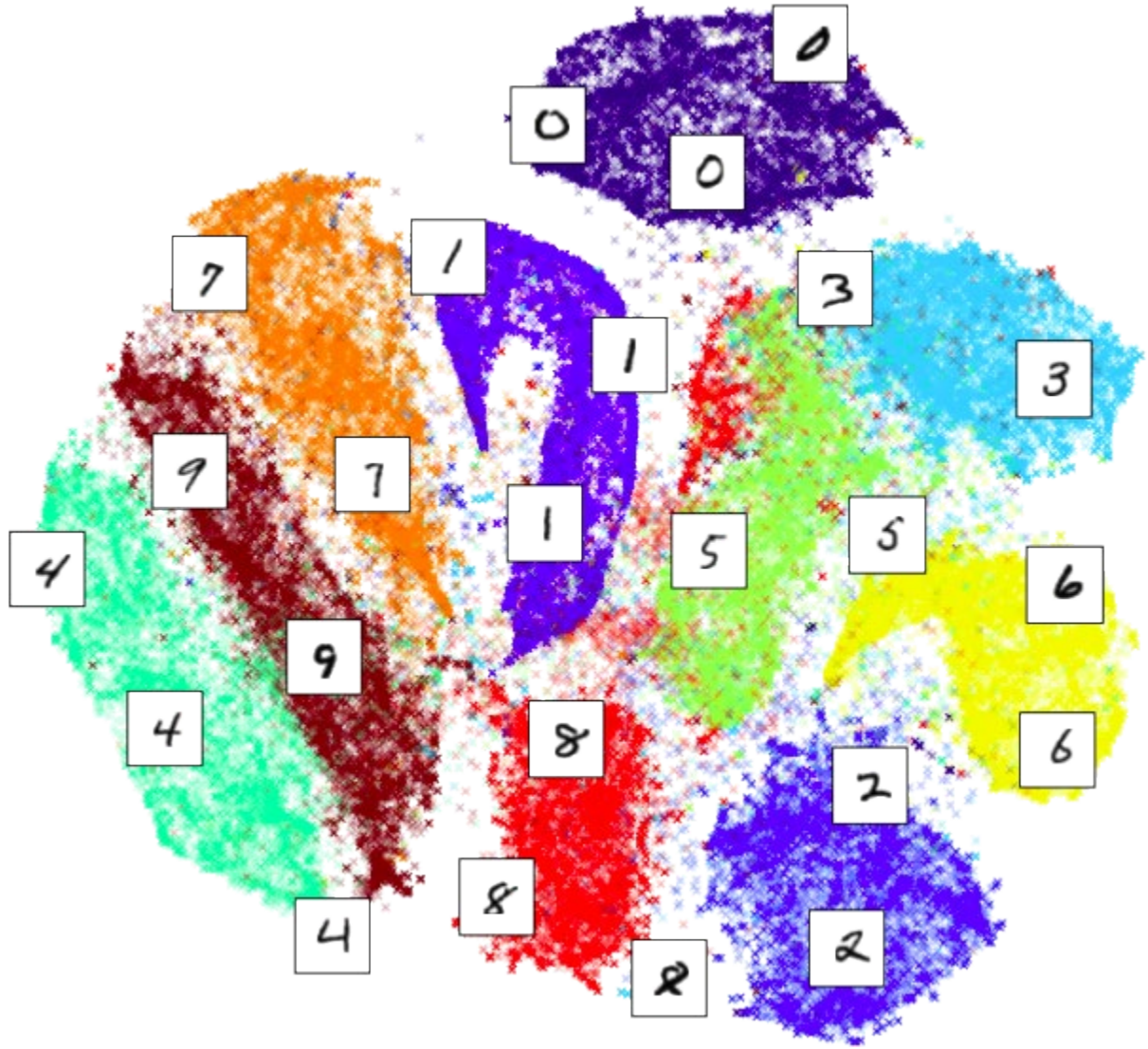
SUMMARY

- In PCA, the main idea to re-express the available dataset to extract the relevant information by reducing the redundancy and minimize the noise.
- We didn't care about whether this dataset represent features from one or more classes, i.e. the discrimination power was not taken into consideration while we were talking about PCA .
- In PCA, we had a dataset matrix $\mathbf{X}$ with dimensions mxn, where columns represent different data samples.
- We first started by subtracting the mean to have a zero mean dataset, then we computed the covariance matrix $\mathbf{S} = \mathbf{X}\mathbf{X}^T$.
- Eigen values and eigen vectors were then computed for $\mathbf{S}$. Hence the new basis vectors are those eigen vectors with highest eigen values, where the number of those vectors was our choice.

T-SNE

Motivation

- t-SNE : t-distributed Stochastic Neighbor Embedding
- allows us to separate data that cannot be separated by any straight line!


0
0
0
1
1
7
3
3
9
7
1
5
5
6
4
9
8
6
4
2
8
8
2

- measure similarities between points in the high dimensional space.
  - For each data point (xi) we'll center a Gaussian distribution over that point.
  - Then we measure the density of all points (xj) under that Gaussian distribution.
  - Then renormalize for all points. This gives us a set of probabilities (Pij) for all points. Those probabilities are proportional to the similarities. All that means is, if data points x1 and x2 have equal values under this gaussian circle then their proportions and similarities are equal and hence you have local similarities in the structure of this high-dimensional space.
  - The Gaussian distribution or circle can be manipulated using what's called perplexity, which influences the variance of the distribution (circle size) and essentially the number of nearest neighbors.

T-SNE

- PCA is a linear dimension reduction technique that seeks to maximize variance and preserves large pairwise distances. In other words, things that are different end up far apart. This can lead to poor visualization especially when dealing with non-linear manifold structures. Think of a manifold structure as any geometric shape like: cylinder, ball, curve, etc.
- t-SNE differs from PCA by preserving only small pairwise distances or local similarities whereas PCA is concerned with preserving large pairwise distances to maximize variance.
- The t-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function.

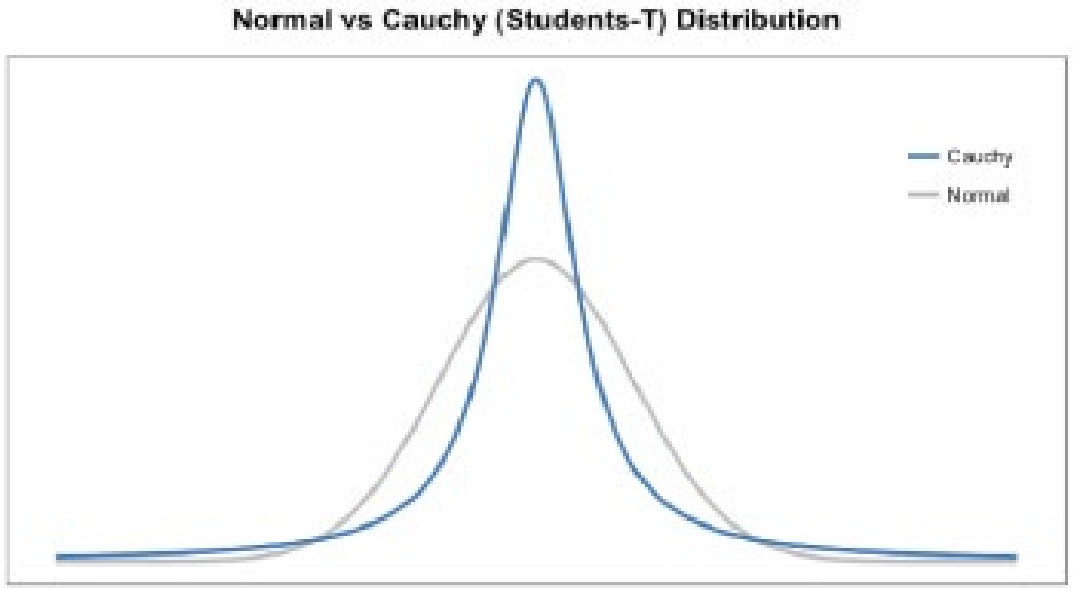


Figure 3 — Normal vs Student t-distribution

- measure similarities between points in the high dimensional space.
  - instead of using a Gaussian distribution you use a Student t-distribution with one degree of freedom, which is also known as the Cauchy distribution.
  - This gives us a second set of probabilities (Qij) in the low dimensional space.
  - As you can see the Student t-distribution has heavier tails than the normal distribution.
  - The heavy tails allow for better modeling of far apart distances.

https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1

- The last step is that we want these set of probabilities from the low-dimensional space (Qij) to reflect those of the high dimensional space (Pij) as best as possible.
- We want the two map structures to be similar. We measure the difference between the probability distributions of the two-dimensional spaces using Kullback-Liebler divergence (KL). It is an asymmetrical approach that efficiently compares large Pij and Qij values.
- Finally, we use gradient descent to minimize our KL cost function.

https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1

Thank You!

Machine Learning and its Applications

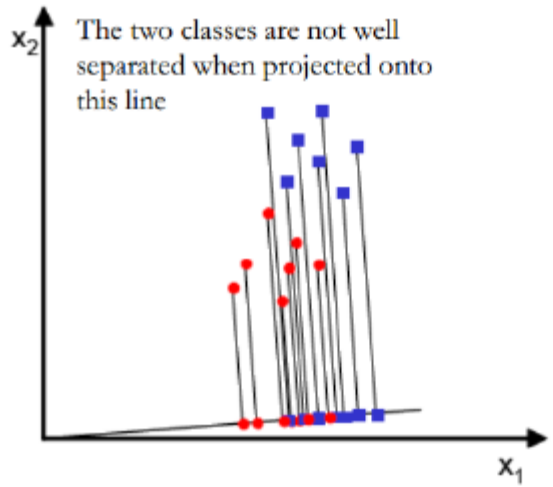
Dimensionality Reduction: Linear Discriminant Analysis (LDA)

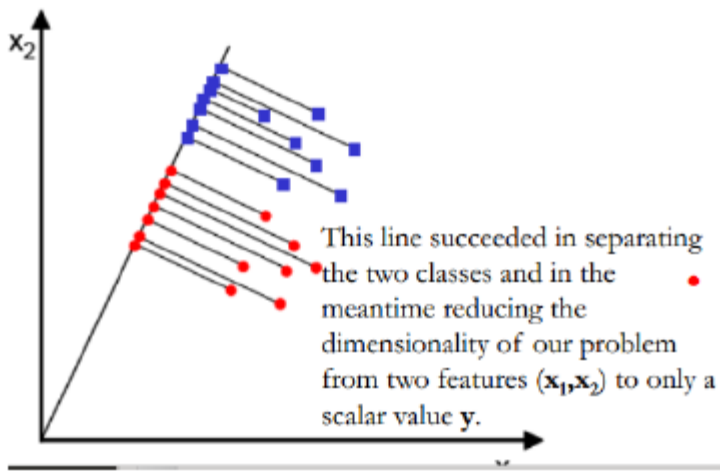
Urban Information Lab

The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible

- Consider a pattern classification problem, where we have C-classes, e.g. seabass, tuna, salmon …
- Each class has $\mathbf{N}_i$ *m*-dimensional samples, where $i = 1,2, \ldots, C$.
- Hence we have a set of *m*-dimensional samples $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{Ni}\}$ belong to class $\boldsymbol{\omega}_i$.
- Stacking these samples from different classes into one big fat matrix **X** such that each column represents one sample.
- **We seek to obtain a transformation of X to Y through projecting the samples in X onto a hyperplane with dimension *C-1*.**
- **Let's see what does this mean?**

LDA … Two Classes





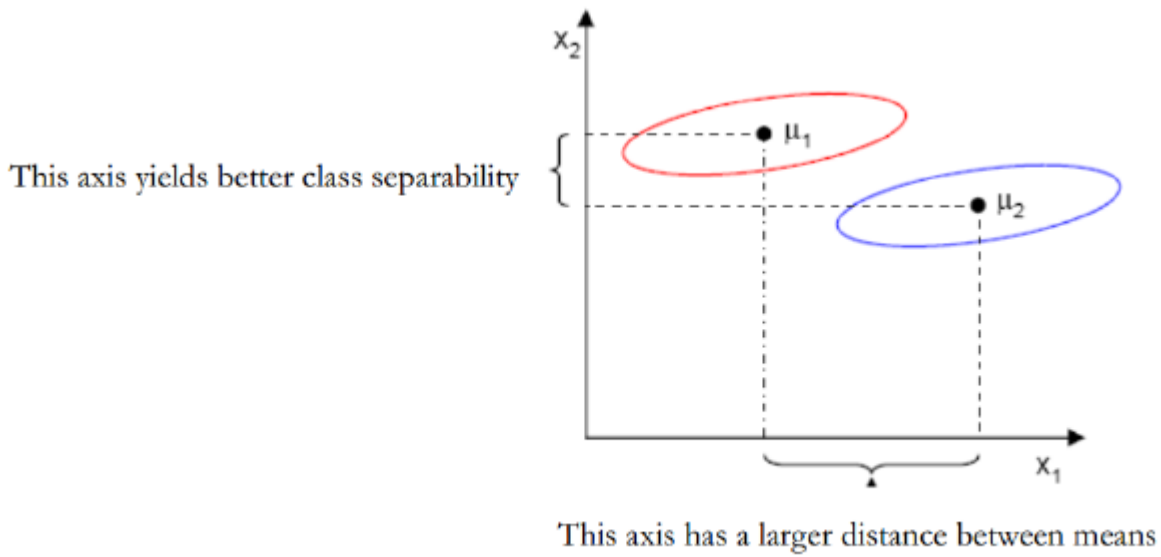
- Assume we have *m*-dimensional samples $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$, N_1 of which belong to ω_1 and N_2 belong to ω_2.
- We seek to obtain a scalar **y** by projecting the samples **x** onto a line (C-1 space, C = 2).

$$y = w^T x \quad where \quad x = \begin{bmatrix} x_1 \\ . \\ . \\ x_m \end{bmatrix} \quad and \quad w = \begin{bmatrix} w_1 \\ . \\ . \\ w_m \end{bmatrix}$$

  - where **w** is the projection vectors used to project **x** to **y**.
- **Of all the possible lines we would like to select the one that maximizes the separability of the scalars.**

LDA … Two Classes

- However, the distance between the projected means is not a very good measure since it does not take into account the standard deviation within the classes.



LDA … Two Classes

- The solution proposed by Fisher is to maximize a function that represents the difference between the means, normalized by a measure of the within-class variability, or the so-called *scatter*.
- For each class we define the **scatter**, an equivalent of the variance, as; (sum of square differences between the projected samples and their class mean).

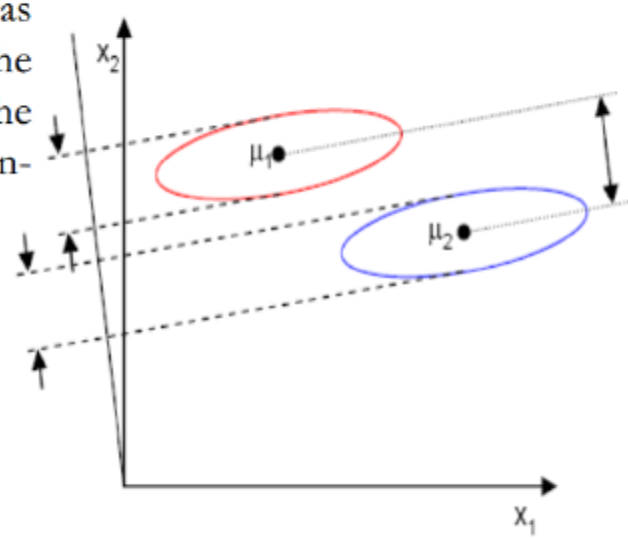
$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2$$

- $\tilde{s}_i^2$ measures the variability within class ω_i after projecting it on the y-space.
- Thus $\tilde{s}_1^2 + \tilde{s}_2^2$ measures the variability within the two classes at hand after projection, hence it is called *within-class scatter* of the projected samples.

LDA … Two Classes

- The Fisher linear discriminant is defined as the linear function $\mathbf{w}^T\mathbf{x}$ that maximizes the criterion function: (the distance between the projected means normalized by the within-class scatter of the projected samples.

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$



- Therefore, we will be looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as farther apart as possible

LDA … Two Classes

- In order to find the optimum projection w^*, we need to express $J(w)$ as an explicit function of w.
- We will define a measure of the scatter in multivariate feature space **x** which are denoted as *scatter matrices*;

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

$$S_w = S_1 + S_2$$

$$J(w) = \frac{\left| \tilde{\mu}_1 - \tilde{\mu}_2 \right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- Where $\mathbf{S}_i$ is the covariance matrix of class $\boldsymbol{\omega}_i$, and $\mathbf{S}_w$ is called the *within-class scatter matrix*.

LDA … Two Classes

- Similarly, the difference between the projected means (in y-space) can be expressed in terms of the means in the original feature space (x-space).

$$\left(\tilde{\mu}_1 - \tilde{\mu}_2\right)^2 = \left(w^T\mu_1 - w^T\mu_2\right)^2$$

$$= w^T\underbrace{\left(\mu_1 - \mu_2\right)\left(\mu_1 - \mu_2\right)^T}_{S_B}w$$

$$= w^T S_B w = \tilde{S}_B$$

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- The matrix $\mathbf{S_B}$ is called the *between-class scatter* of the original samples/feature vectors, while $\tilde{S}_B$ is the between-class scatter of the projected samples **y**.
- Since $\mathbf{S_B}$ is the outer product of two vectors, its rank is at most one.

LDA … Two Classes

- We can finally express the Fisher criterion in terms of $\mathbf{S_W}$ and $\mathbf{S_B}$ as:

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{w^T S_B w}{w^T S_W w}$$

- Hence ***J(w)*** **is a measure of the difference between class means (encoded in the between-class scatter matrix) normalized by a measure of the within-class scatter matrix.**

LDA … Two Classes

- To find the maximum of $J(w)$, we differentiate and equate to zero.

$$\frac{d}{dw}J(w)=\frac{d}{dw}\left(\frac{w^T S_B w}{w^T S_W w}\right)=0$$

$$\Rightarrow \left(w^T S_W w\right)\frac{d}{dw}\left(w^T S_B w\right)-\left(w^T S_B w\right)\frac{d}{dw}\left(w^T S_W w\right)=0$$

$$\Rightarrow \left(w^T S_W w\right)2S_B w-\left(w^T S_B w\right)2S_W w=0$$

Dividing by $2w^T S_W w$:

$$\Rightarrow \left(\frac{w^T S_W w}{w^T S_W w}\right)S_B w-\left(\frac{w^T S_B w}{w^T S_W w}\right)S_W w=0$$

$$\Rightarrow S_B w-J(w)S_W w=0$$

$$\Rightarrow S_W^{-1}S_B w-J(w)w=0$$

LDA … Two Classes

- Solving the generalized eigen value problem

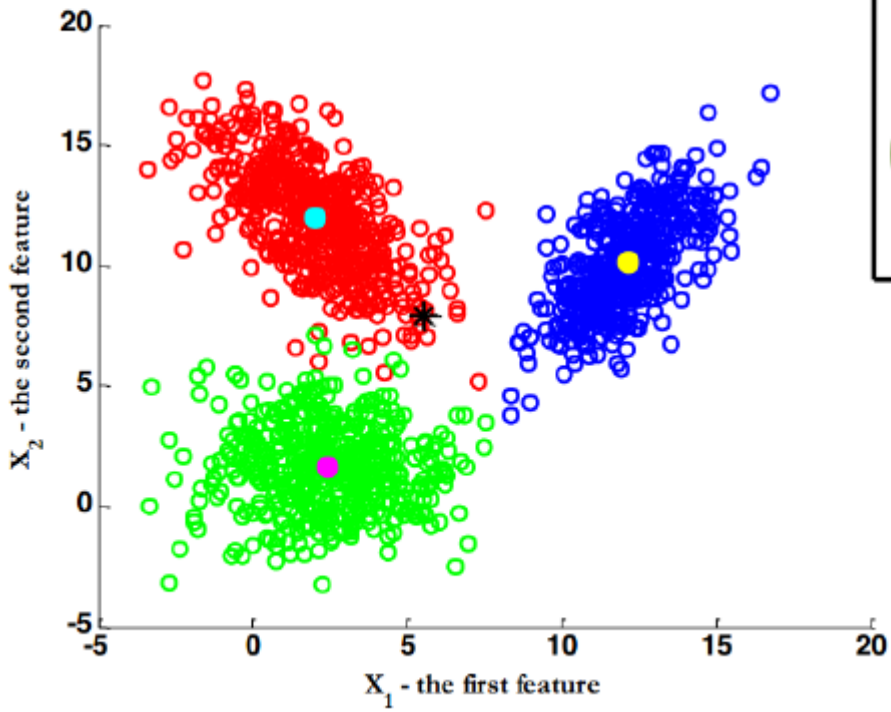
$$S_W^{-1} S_B w = \lambda w \qquad where \quad \lambda = J(w) = scalar$$

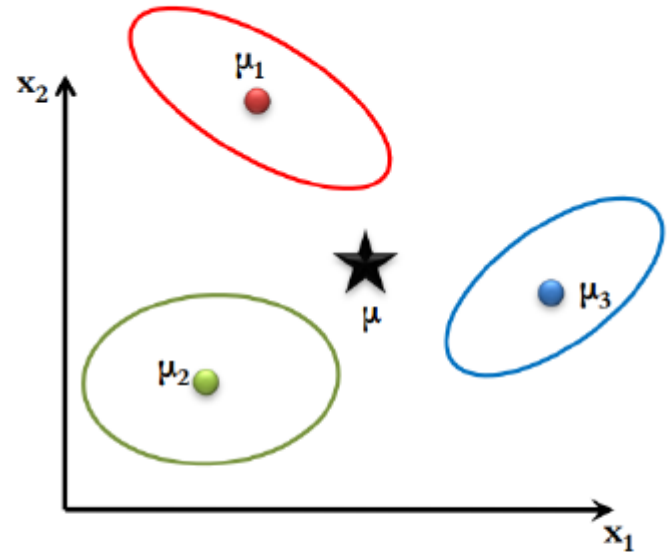
yields

$$w^* = \arg\max_w J(w) = \arg\max_w \left(\frac{w^T S_B w}{w^T S_W w} \right) = S_W^{-1} (\mu_1 - \mu_2)$$

- This is known as Fisher's Linear Discriminant, although it is not a discriminant but rather a specific choice of direction for the projection of the data down to one dimension.
- Using the same notation as PCA, **the solution will be the eigen vector(s) of** $S_X = S_W^{-1} S_B$

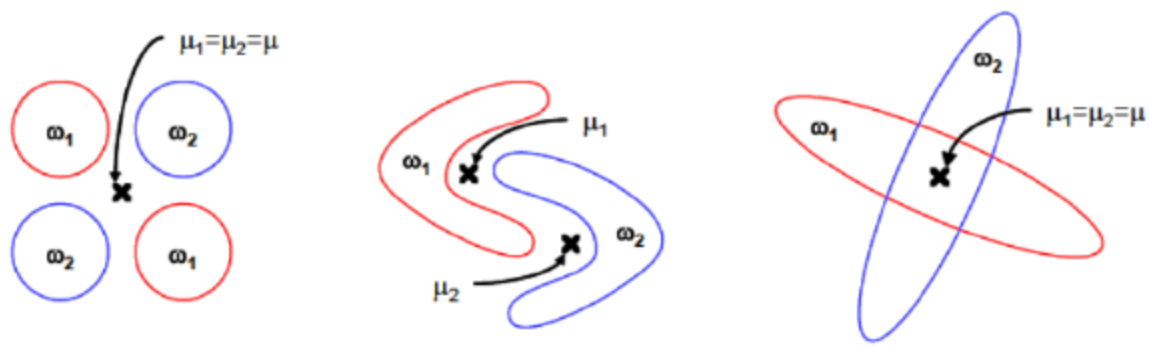
It's Working … ☺


20
15
10
5
0
-5
-5
0
5
10
15
20
X2 - the second feature
X1 - the first feature


x2
μ1
μ
μ3
μ2
x1

Limitations of LDA ☹

- **LDA produces at most C-1 feature projections**
  - If the classification error estimates establish that more features are needed, some other method must be employed to provide those additional features
- **LDA is a parametric method since it assumes unimodal Gaussian likelihoods**
  - If the distributions are significantly non-Gaussian, the LDA projections will not be able to preserve any complex structure of the data, which may be needed for classification.



Thank You!