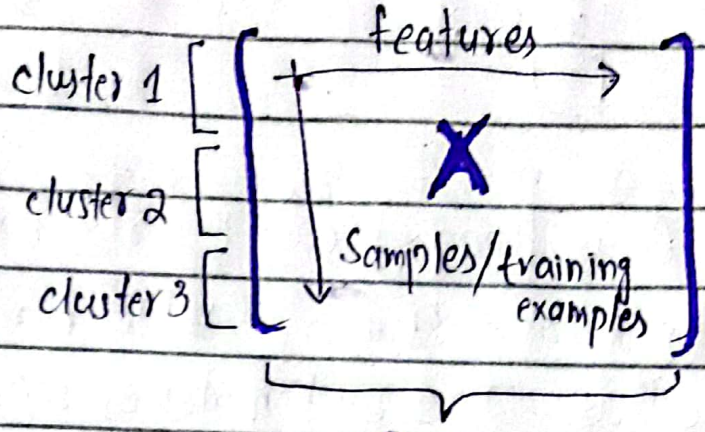
PCA

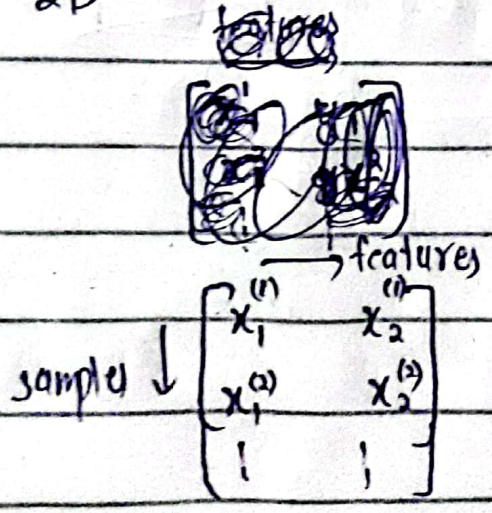
→ Method for dimensionality reduction

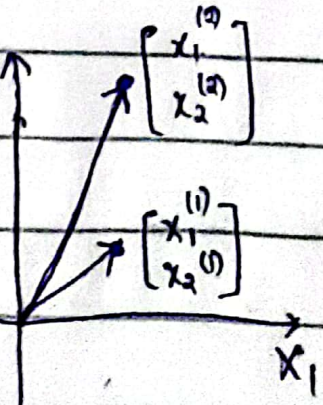


Reduce features by not discarding some features but transforming existing features to a smaller set of new features

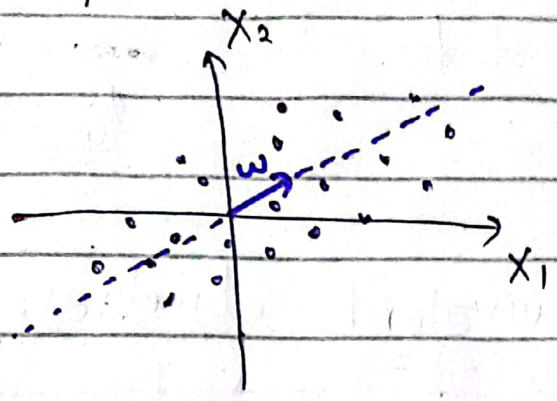
Lets start with the example of reducing 2 dimensions to one dimension.

In 2D





For many training examples:



If w is a unit vector i.e $\|w\| = 1$ then projections of sample vectors in X onto this unit vector are given by:

$$Xw$$

How to choose w?

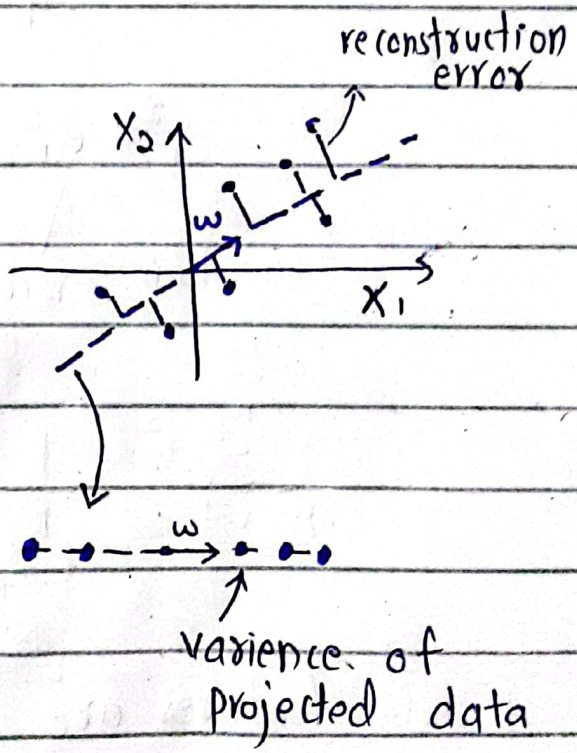
→ To minimize the reconstruction error

OR

→ To maximize the variance

↓

By maximizing variance, you ensure that variation in original data is preserved when projected onto lower dimensional representation. If variance is small then projected data points

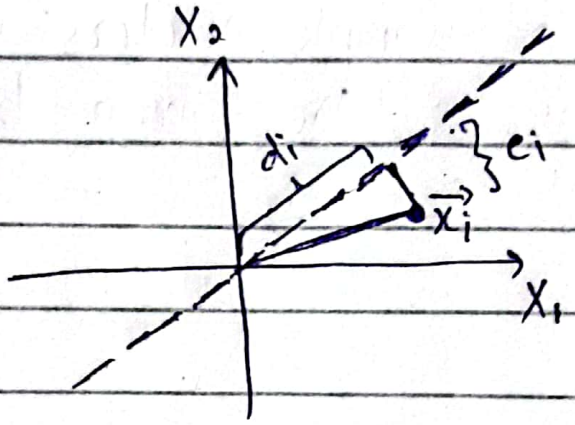


are very close, losing useful structure in original data.

small variance.

These are equivalent objectives:

Maximizing variance ⇔ minimizing error



$$d_i^2 + e_i^2 = \|x_i\|^2$$

projection — reconstruction error

$$\frac{1}{n}\sum_{i=1}^{n} d_i^2 + \frac{1}{n}\sum_{i=1}^{n} e_i^2 = \frac{1}{n}\sum_{i=1}^{n} \|x_i\|^2$$

MSE — const (doesn't depend on chosen axis)

if all features are centered (i.e subtracted their mean) then origin represent means of features and

$$\frac{1}{n}\sum_{i=1}^{n} d_i^2 = \frac{1}{n}\sum_{i=1}^{n} (d_i^2 - \underbrace{\text{mean}}_{=0}) = \text{Variance}$$

So

Variance + Mean Squared Error = const

Var. of projected data

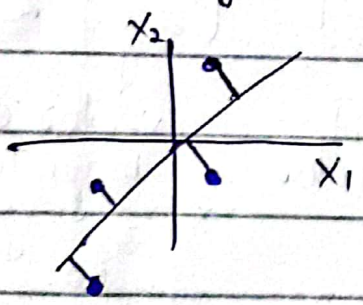
So either minimizing MSE maximizes variance or vice versa.

So Do either one of them.

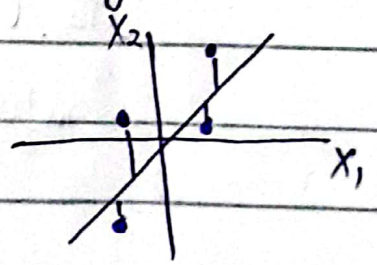
PCA vs regression:

In PCA, error is the perpendicular distance b/w data point and projection axis while

In regression, error is the vertical distance b/w predicted $\hat{y}$ and actual y



PCA



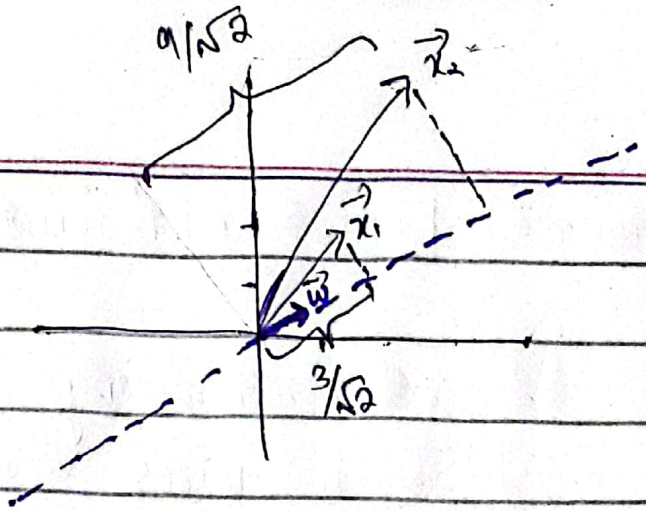
regression

PCA loss function:

1) **Minimizing reconstruction error:**

$X\omega$ gives projections of data vectors onto ω:

$$\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{3}{\sqrt{2}} \\ \frac{9}{\sqrt{2}} \end{bmatrix}$$

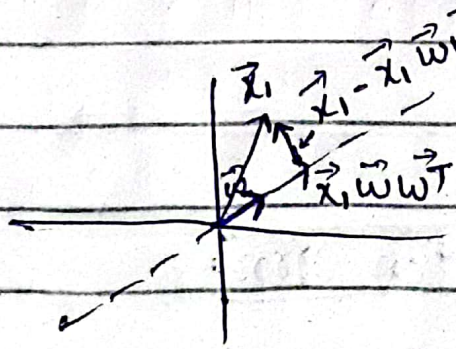


$X w w^T$ will give vectors of these projections

$$\underbrace{\begin{bmatrix} 3/\sqrt{2} \\ 9/\sqrt{2} \end{bmatrix}}_{Xw} \underbrace{\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}}_{w^T} = \underbrace{\begin{bmatrix} 3/2 & 3/2 \\ 9/2 & 9/2 \end{bmatrix}}_{Xww^T}$$

$$\frac{3}{\sqrt{2}} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 3/2 & 3/2 \end{bmatrix}$$

↑ magnitude ↑ unit vector ↑ vector of magnitude $3/\sqrt{2}$ along the unit vector.



$$\vec{e}_2 = \vec{x}_i - \vec{x}_i \vec{w} \vec{w}_T$$

l_2 norm gives length of error vector

$$\|\vec{e}_i\| = \|\vec{x}_i - \vec{x}_i \vec{w} \vec{w}_T\|_2$$

$$\sum_i |\vec{e}_i|^2 = \sum_i \|\vec{x}_i - \vec{x}_i \vec{w} \vec{w}_T\|_2^2$$

Sum of squared errors $= \sum_i \|\vec{x}_i - \vec{x}_i \vec{w} \vec{w}_T\|_2^2$

$$= \left(\sqrt{\sum_i \|\vec{x}_i - \vec{x}_i \vec{w} \vec{w}_T\|_2^2} \right)^2$$

$$SSE = \left(\|X - X w w^T\|_F \right)^2$$

$$L = \|X - X w w^T\|_F^2$$

l_2 of forbeniuous norm of matrix $=$ l_2 norm (l_2 norm of each row vector or column vector in matrix)

forbenius norm

$$= \sqrt{\sum_i \sum_j |x_{ij}|}$$ ← sum of absolute values

$$= \sqrt{\sum_i \|r_i\|_2^2}$$ ← sum of square of length of row vectors

$$= \sqrt{\sum_j \|c_j\|_2^2}$$ ← Sum of square of length of column vectors

2) Maximizing variance:

$$-L = \frac{1}{n} \underbrace{(Xw)^T (Xw)}_{\text{Variance}}$$

Sum of square of projections

$$-L = \frac{1}{n} w^T X^T X w$$

if $\frac{1}{n} X^T X = C$ ← covariance matrix

$$\boxed{-L = w^T C w, \text{ s.t. } \|w\|^2 = 1}$$

→ These equations are true when 1) data is centered with mean = 0

2) w is a unit vector making Xw the projection of X.

$$\vec{X}\cdot\vec{w} = |\vec{w}||\vec{X}|\cos\theta$$

$$\vec{X}\cdot\vec{w} = |\vec{X}|\cos\theta \quad (|\vec{w}| = 1)$$

$\vec{X}\vec{w}$ = projection.

Also otherwise:

for example:

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} x_1^2 + x_2^2 & x_1 y_1 + x_2 y_2 \\ x_1 y_1 + x_2 y_2 & y_1^2 + y_2^2 \end{bmatrix}$$

if mean of each feature x, y is zero then

$$\frac{1}{n} X^T X = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{xy} & C_{yy} \end{bmatrix}$$

- $L = w^T C w$ can be increased to infinity using very large ($\to \infty$) w

Solution:

Maximizing $w^T C w$:

We will use 'Lagrange' multiplier to solve this constrained optimization problem

Constraint:- $\|w\|_2^2 = 1$

$$w^T w = 1$$

$$-L = w^T C w - \lambda(w^T w - 1)$$

Setting $\frac{\partial L}{\partial w} = 0$,

$$\boxed{Cw = \lambda w}$$

w should be the eigenvector of C

To maximize variance. $w^T C w = \lambda w^T w = \lambda$,

choose eigenvector with the largest eigenvalue λ.

Spectral theorem:

for p features

$C = \frac{1}{n} X^T X$ is a symmetric $p \times p$ matrix.

It has p eigenvectors orthogonal to each other.

$$V = \begin{bmatrix} | & | & & | \\ \vec{\omega}_1 & \vec{\omega}_2 & \cdots & \vec{\omega}_p \\ | & | & & | \end{bmatrix}$$

$p \times p$ size

Projecting data on multiple eigenvectors of C:-

Rotated data: XV

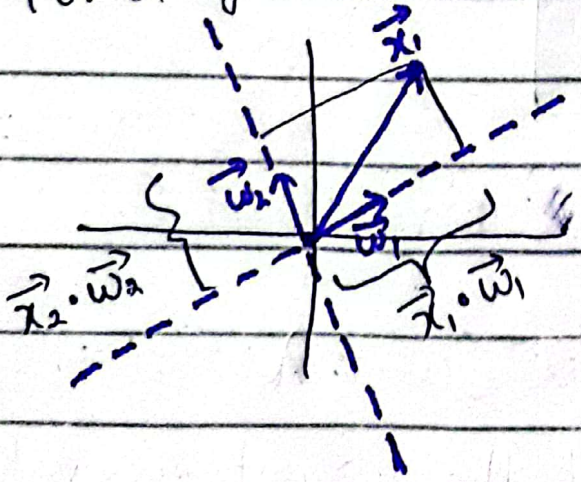
vector of first data point in standard basis

$$\begin{bmatrix} \vec{x}_1 \\ \vdots \\ \vec{x}_n \end{bmatrix} * \begin{bmatrix} \vec{\omega}_1 & \vec{\omega}_2 & \cdots & \vec{\omega}_p \end{bmatrix}$$

row representing vector of first data point in eigenbasis.

$$\begin{bmatrix} \vec{x}_1 \cdot \vec{\omega}_1 & \vec{x}_1 \cdot \vec{\omega}_2 & \cdots & \vec{x}_1 \cdot \vec{\omega}_p \\ & \vdots & & \\ & n & & \end{bmatrix}$$

For 2 eigenvectors:



The covariance (which was variance when using only one eigenvector) in the eigen basis is found as

$$\text{Covariance after projection on eigen basis} = \frac{1}{n}(XV)^T XV$$

$$= \frac{1}{n} V^T X^T X V$$

$$= V^T C V$$

↑ Covariance of original data in standard basis

where $CV = V\Lambda$ ← Diagonal matrix of eigen-values

$$= \overbrace{V^T V}^{I \text{ as } V \text{ is othogonal}} \Lambda \Rightarrow \Lambda \qquad \Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_p \end{bmatrix}$$

Covariance matrix after projection on eigen basis $= \Lambda$ → New Data projected on eigenbasis has variance of λ_i along ith axis and covariance btw all axis is 0 ★

→ There is no correlation b/w the features of transformed data, avoiding any redundent feature.

As $\underbrace{V^T C V}_{\text{transformation}} = \Lambda$

C in the perspective of eigen vectors of C is just scaling transformation Λ.

$$V(V^T C V)V^T = V(\Lambda)V^T$$

spectral theorem → $C = V\Lambda V^T$

eigen decomposition

$C = V\Lambda V^{-1}$
for symmetric C,
$C = V\Lambda V^T$
which is spectral theorem

scaling transformation Λ of eigen vectors in the perspective of standard basis is just C

→ Each eigenvector in V is the <u>principal component</u> sorted in the descending order of λ in Λ.

Eigendecomposition is $C = V\Lambda V^{-1}$ which do not exist when V^{-1} do not exist. For any symmetric matrix, it simplifies to $C = V\Lambda V^T$ and exist. This is spectral theorem

Relationship to SVD:

SVD is given as

$$X = USV^T$$

- S: Diagonal matrix of singular values
- V^T: right singular vectors of X ↔ eigenvectors of A^TA
- U: left singular vectors of X ↔ eigenvectors of AA^T

if X represent original data which is centered then covariance matrix is

$$C = \frac{1}{n} X^T X$$

$$= \frac{1}{n} VSU^T USV^T$$

$$= \frac{1}{n} VS^2V^T$$

For diagonal matrix S:
$\rightarrow S^T = S$
$\rightarrow \underbrace{SSS\ldots}_{n \text{ times}} = S^n$

$$C = \frac{VS^2V^T}{n}$$

$$C = V\Lambda V^T$$

This is the same form as found using eigen decomposition in previous section.

Orthogonal ↓ Diagonal ↓ Orthogonal

$$C = \begin{bmatrix} \text{right singular vectors of } X \end{bmatrix} \begin{bmatrix} \text{square of singular values of } X \text{ divided by } n \end{bmatrix} \begin{bmatrix} \text{right singular vectors of } X \end{bmatrix}^T$$ } From SVD

which is equivalent to

$$C = \begin{bmatrix} \text{eigen vectors of } C \end{bmatrix} \begin{bmatrix} \text{eigen values of } C \end{bmatrix} \begin{bmatrix} \text{eigen vectors of } C \end{bmatrix}^T$$ } From eigen decomposition

This means

eigen vectors of covariance matrix C = right singular vectors of data matrix X

eigen values of covariance matrix C = Square of singular values of data matrix X divided by sample size n.

→ Principal components of data X is its right singular vectors with variance of square of their singular values divided by n.

Total variance:

We know from eigendecomposition of symmetric matrix **A** (spectral theorem) that

$$A = V \Lambda V^T$$

or

$$\Lambda = V^T A V$$

- transformation A in the perspective of eigen basis
- Λ: eigen values of A

trace of matrix is sum of its diagonal entries

From a property,

$$tr(A) = tr(\Lambda)$$

For covariance of original data **C** and covariance of transformed data (to its eigen basis) Λ

$$\boxed{tr(C) = tr(\Lambda)}$$

Λ: eigen values of C $\leftrightarrow$ Covariance of transformed data

→

Sum of variance of all the original features = Sum of variance explained by all the principal components

→ Explained variance ratio by PC i is $\lambda_i / tr(C)$

→ Using PCA, we reconstruct the data in few new axes, called principal components, which explain most of the variance while remaining axes are discarded as representing (noise)

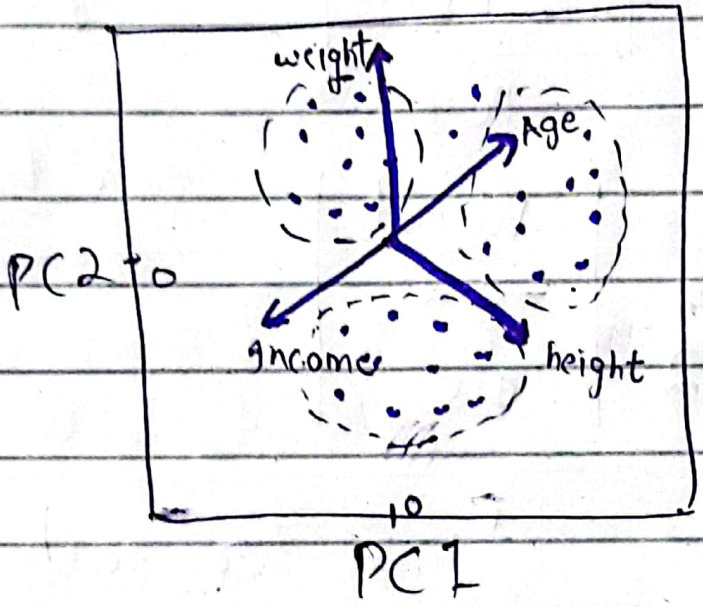
→ Advantage of PCA:

1) Minimizing noise by discarding PCs with small variance
2) Removing redundent features by creating new uncorelated features

PCA is used for:
1) Data Exploration
2) Data preprocessing

PCA for data exploration:

Biplot is created with datapoints in the coordinates of PC1 and PC2.

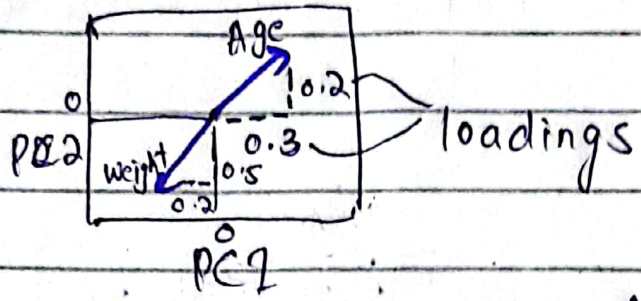


It contains feature vectors. The angle b/w the feature vectors represent the correlation b/w them.

90° → No correlation
greater than 90° → Negative correlation
less than 90° → Positive correlation.

Most importantly, the projection of each

feature vector on PC1 and PC2 axis gives the correlation of that feature with PC1 and PC2 respectively. These are called **loadings** for that feature.



Components:-

	Age	Weight	other features
PC1	0.3	-0.2	- - - -
PC2	0.2	0.5	

principal components vectors

loadings for each feature

Intuition:

When projecting data point (x, y) on $\overrightarrow{PC} : [v_1 \; v_2]^T$, we have projection of

$$\text{projection}_{\text{on } \overrightarrow{PC}} = x v_1 + y v_2$$

v_1 and v_2, which are cordinates of PC vector, act as weights of feature x and y for the projection.

These weights are loadings and show correlation b/w feature and PC.
For example if

$v_1 = 1$

then $v_2 = 0$ as $\|\vec{PC}\|_2 = 1$

and

projection on $\vec{PC}$ $= x$

$\vec{PC} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ← which is along the x axis.

So PC and x feature are highly correlated.

→ Max value of loading is 1 when using unit eigenvector representing PC

→ Before PCA, standardize the data to have same unit variance for each feature otherwise the features with high variance will dominate the PC. These will be selected as PC; the axes explaining max variance.

How many PCs to choose?

1) Heuristic of choosing number of PCs that will capture 90% of total variance.

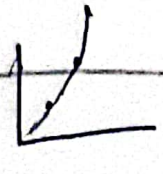
2) Shuffling the features.

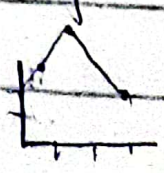
It will remove existing relationships b/w features and make data a random noise while preserving the variance of each feature.

f_1	f_2
1	2
2	4
3	6

→ variance remain same

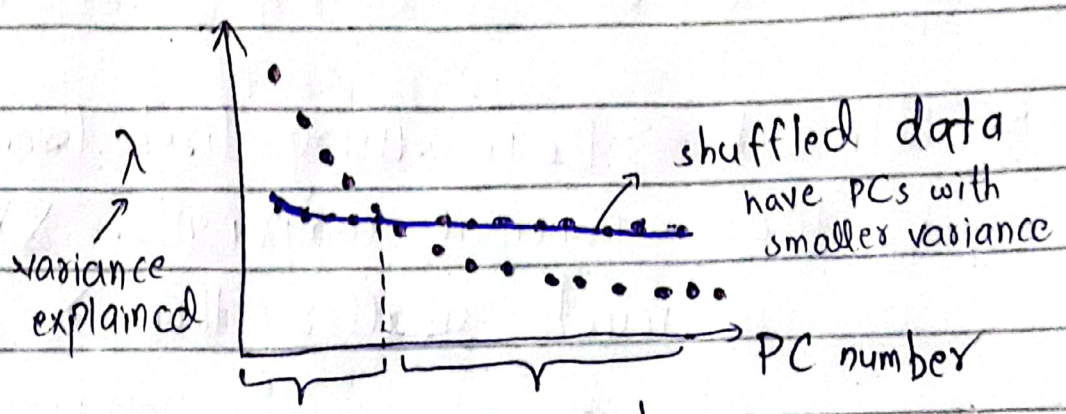
f_1	f_2
3	2
2	6
1	4





After shuffling

Find the eigenvalues and compare them with eigenvalues of original unshuffled data.



Number of PCs that should be selected. They explain more variance than random noise does.

Discard as noise

PCA for preprocessing:

Reduce X with p features to a small number k of PCs using XV_k where V_k is a $p \times k$ matrix of unit norm eigenvectors with the largest eigenvalues. Then use XV_k for downstream processing.

$$\underset{n \times p}{[X]} \underset{\substack{V_k \\ p \times k}}{[\downarrow \downarrow]} = \underset{n \times k}{[XV_k]} \quad \text{where } k \leq p$$

Advantage of XV_k over X:

1) XV_k has correlation b/w k features be equal to 0.

2) with $k << p$, i.e using first few significant principal components, XV_k size is much smaller than X.

3) No small singular values / eigenvalues left

Summary of PCA

PCA is a dimensionality reduction technique used to transform data containing correlated features and noise to a smaller set of uncorrelated features, called principal components, which capture most of the variance in the original data while discarding remaining variance as caused by noise. PCA is used for data exploration and preprocessing.

- These principal components are found to be the eigenvectors of covariance matrix **C** of original data, sorted in the descending order of their eigenvalues.

- Fist principal component captures the maximum variance, the second principal component captures the second most, and so on.

- The covariance matrix of projected data on these eigenvectors (principal components) will be a diagonal matrix Λ containing eigenvalues in diagonal.

This shows that: 1) Principal components are uncorrelated due to 0s in off diagonal. 2) Variance explained by a principal component i is λ_i (eigen value)

- In relationship to SVD, principal components for data X is the right singular vectors of X and their variance is square of singular values of X divided by sample size n.

- From trace property; $tr(C) = tr(\Lambda)$ where Λ is the diagonal matrix of eigen values of covariance C of original data, which is also the covariance matrix of transformed data. It shows that sum of variance of all the original features is equal to the sum of variance explained by all the principal components.

- Explained variance ratio by PC i is $\lambda_i / tr(C)$

- Biplot is used to visualize the projection of each vector, representing a training example in n feature dimensions, onto the PC1 and PC2. The feature vectors are plotted with dimension of loadings. The loading of a feature f for PCi is the weight given to that feature in finding the projections of training examples on PCi. It shows the correlation b/w the features and principal components. The angle b/w feature vectors shows the correlation b/w them. greater than 90, equal to 90 and less than 90 angle corresponds to negative, zero and positive correlation respectively.

- PCA can be used to preprocess data to a transformed data of uncorrelated features and much smaller dimensionality