Principal Component Analysis

Chetan Arora

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

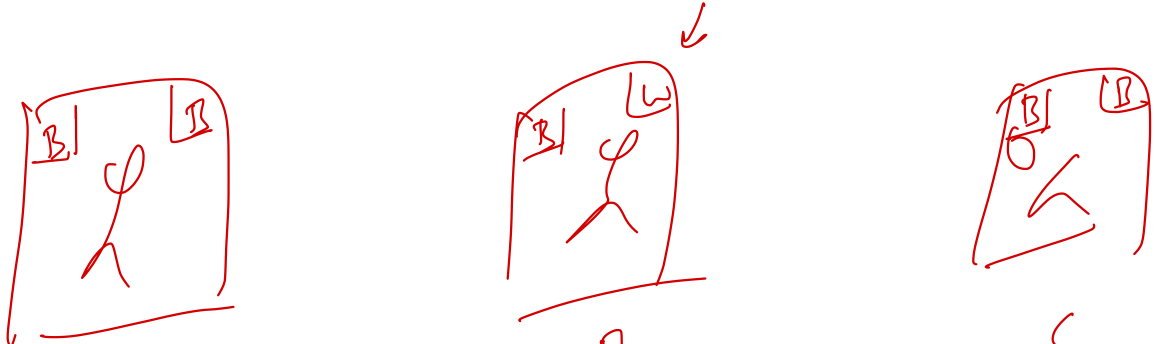
Covariance

- Variance and Covariance are a measure of the “spread” of a set of points around their center of mass (mean)

- **Variance** is a measure of the deviation from the mean for points in one dimension e.g. variance in the measuring the length of the same object by different people.

- **Covariance** is a measure of how much each of the dimensions vary from the mean with respect to each other.

x_i x_j



A B C

X_1 X_2

$$z_1 = \cos\theta \, x_1 + \sin\theta \, x_2$$

$$X = \{ x_1 \, \dots \, x_2 \, \dots \, x_d \}$$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Covariance

- Covariance is measured between 2 dimensions to see if there is a relationship between the 2 dimensions e.g. number of hours studied & marks obtained.

- The covariance between one dimension and itself is the variance

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

$$\sum_{k=1}^{N} \left(X_i^{(k)} - \bar{X}\right)\left(X_j^{(k)} - \bar{X}\right)$$

Covariance

X_1 X_2

$$\text{Covariance}(X, Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})\,(Y_i - \bar{Y})}{n}$$

- For a 3-dimensional data set (x, y, z), one can measure the covariance between:
  - x and y dimensions,
  - y and z dimensions, and
  - x and z dimensions.

ij ji

- Covariance is symmetrical: $\text{Cov}(X, Y) = \text{Cov}(Y, X)$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Covariance Matrix

- Covariance between various dimensions is typically represented as a 2D **covariance matrix (C)**. For a 3-dimensional data (X, Y, Z):

$$C = \begin{bmatrix} \mathrm{Cov}(X,X) & \mathrm{Cov}(X,Y) & \mathrm{Cov}(X,Z) \\ \mathrm{Cov}(Y,X) & \mathrm{Cov}(Y,Y) & \mathrm{Cov}(Y,\mathrm{Z}) \\ \mathrm{Cov}(Z,X) & \mathrm{Cov}(Z,Y) & \mathrm{Cov}(Z,Z) \end{bmatrix}$$

$d \times d$

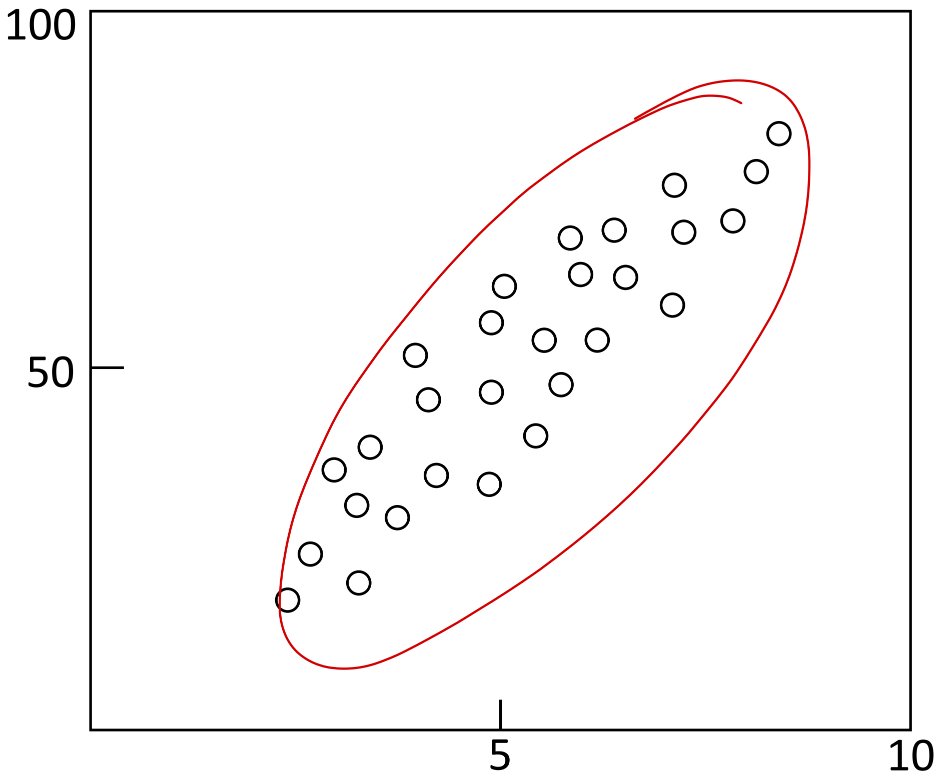
- Diagonal elements are the variances of X, Y and Z.

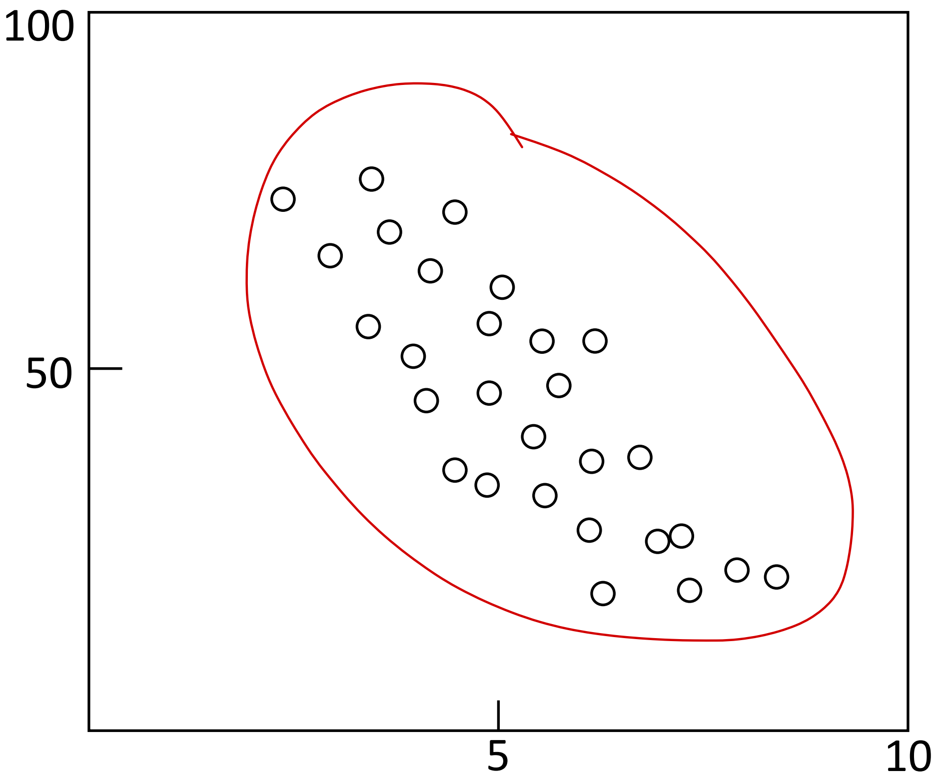
- The covariance matrix C is is symmetrical about the diagonal

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Interpreting Covariance

- **Example:** A 2-dimensional data set.
  - x: number of hours studied for a subject
  - y: marks obtained in that subject





Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Interpreting Covariance

- A positive value of covariance indicates both dimensions increase or decrease together e.g. as the number of hours studied increases, the marks in that subject increase.

- A negative value indicates while one increases the other decreases, or vice-versa e.g. number of hours on facebook vs performance in CS dept.

- If covariance is zero: the two dimensions are independent of each other e.g. heights of students vs the marks obtained in a subject

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Interpreting Covariance

Q. Why bother with calculating covariance when we could just plot the 2 values to see their relationship?

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Interpreting Covariance

Q. Why bother with calculating covariance when we could just plot the 2 values to see their relationship?

A. Covariance calculations are used to find relationships between dimensions in high dimensional data sets (usually greater than 3) where visualization is difficult.

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

PCA

- Principal components analysis (PCA) is a technique that can be used to simplify a dataset

- It is a linear transformation that chooses a new coordinate system for the data set such that greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component), the second greatest variance on the second axis, and so on.

- PCA can be used for reducing dimensionality by eliminating the later principal components.

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

PCA: Toy Example

- Consider following 3D points:

x_1	1	2	4	3	5	6
x_2	2	4	8	6	10	12
x_3	3	6	12	9	15	18

- Number of integers to be stored: 18

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

PCA: Toy Example

- How about the following representation?

Basis Vector

$$\begin{bmatrix}1\\2\\3\end{bmatrix} = 1 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}2\\4\\6\end{bmatrix} = 2 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}4\\8\\12\end{bmatrix} = 4 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

i
j
n

h

$$\begin{bmatrix}3\\6\\9\end{bmatrix} = 3 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}5\\10\\15\end{bmatrix} = 5 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}6\\12\\18\end{bmatrix} = 6 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

- Number of integers to be stored: 9

d

K Subspace

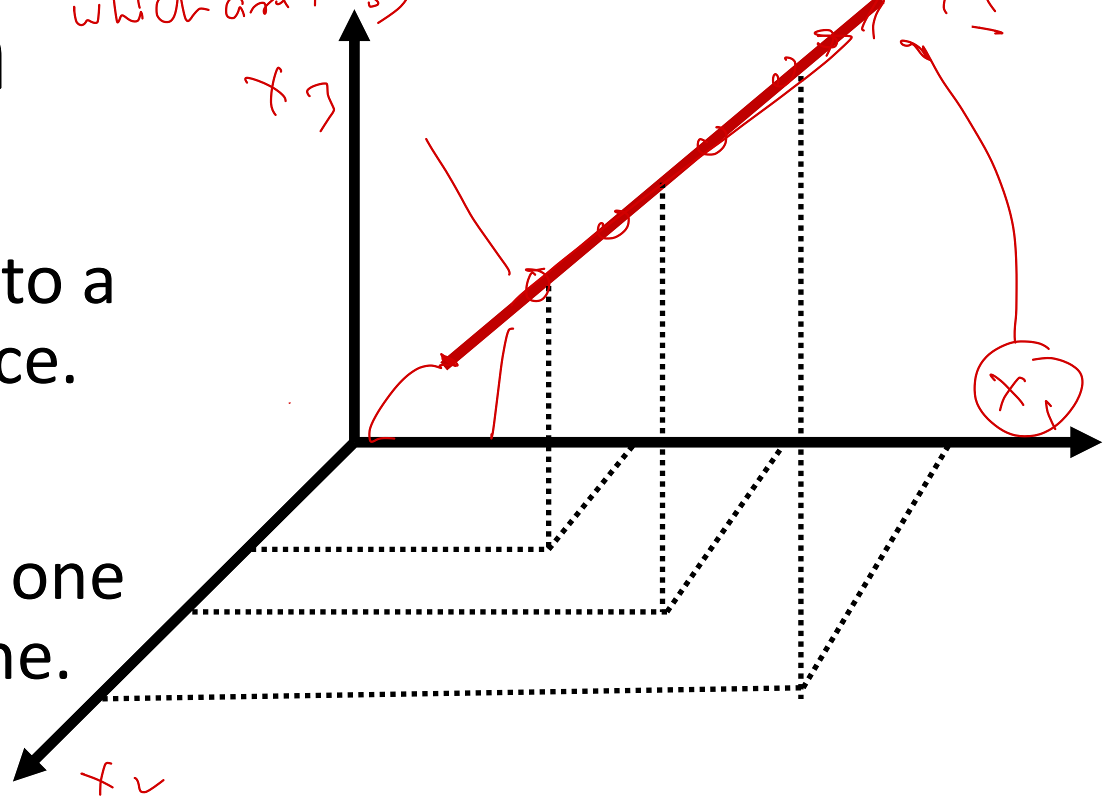
$K < d$

$<< d$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Geometrical Interpretation

Samples which are in 3D ⇒ 1D Subsp



- View each point in 3D space. But in this example, all the points happen to belong to a line: a 1D subspace of the original 3D space.
- Consider a new coordinate system where one of the axes is along the direction of the line.
- In this coordinate system, every point has only one non-zero coordinate.
- We only need to store the direction of the line (3 integers) and the non-zero coordinate for each of the 6 points (6 integers).

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Principal Component Analysis

- Given a set of points, how do we know if they can be compressed like in the previous example?
- The answer is to look into the correlation between the points
- The tool for doing this is called PCA

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Principal Component Analysis

- By finding the eigenvalues and eigenvectors of the covariance matrix, we find that the eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset.

- This is the principal component.

- PCA is a useful statistical technique that has found applications in:
  - fields such as face recognition and image compression
  - finding patterns in data of high dimension.

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Vector

Column vector

PCA Theorem

- Let $x_1, x_2, \ldots, x_n$ be a set of n, $d \times 1$ vectors and let $\bar{x}$ be their average:

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix} \qquad \bar{x} = \frac{1}{n}\sum_{i=1}^{i=n} \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix}$$

- Let X be the $d \times n$ matrix with columns: $[x_1 - \bar{x} \quad x_2 - \bar{x} \quad \ldots \quad x_n - \bar{x}]$

$$X = \begin{bmatrix} x_{i1} & \cdots \\ x_{i2} & \cdots \\ \vdots & \\ x_{id} & \end{bmatrix}$$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Covariance Matrix

- Let $Q = XX^T$ be the $d \times d$ covariance matrix:

$$Q = [x_1 - \bar{x} \quad x_2 - \bar{x} \quad \dots \quad x_n - \bar{x}] \begin{bmatrix} (x_1 - \bar{x})^T \\ (x_2 - \bar{x})^T \\ \vdots \\ (x_n - \bar{x})^T \end{bmatrix}$$

$Cov(X_2, X_1)$

- Also called **Scatter Matrix** in the context of PCA.

- Q is square as well as symmetric

$d \times d$

- Q can be very large (in vision, d is often the number of pixels in an image!)

$$\begin{bmatrix} x_{11} & x_{i1} \\ x_{12} & x_{i2} \quad x_{n2} \\ x_{13} & \vdots \\ \vdots & \vdots \\ x_{1d} & x_{id} \end{bmatrix}
\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ \vdots & & & \\ x_{i1} & x_{i2} & & x_{id} \\ & & & \\ x_{n1} & & & \end{bmatrix}$$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

32x32

1024

d x d

d eigen vectors

d

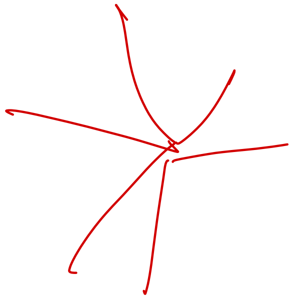
PCA Theorem

Theorem:

Each x_j can be written as: $x_j = \bar{x} + \sum_{i=1}^{i=d} g_{ji} e_i$, where e_i are the d eigenvectors of Q with non-zero eigenvalues.

e_i

- The eigenvectors $e_1, e_2, \ldots, e_d$ span an **eigenspace**.



- $e_1, e_2, \ldots, e_d$ are $d \times 1$ orthonormal vectors.

- The scalars g_{ji} are the coordinates of x_j in the eigenspace:

$$g_{ji} = (x_j - \bar{x}) . e_i$$

$\lambda_1 > \lambda_2 \ldots \lambda_d$

e_1

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Using PCA to Compress Data

- Expressing x in terms of $e_1, \ldots e_d$ has not changed the size of the data

- If the points are highly correlated many of the coordinates of x will be zero or close to zero (if they indeed lie in a lower-dimensional linear subspace)

- Sort the eigenvectors e_i according to their eigenvalue: $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_\mathrm{d}$

- Assuming $\lambda \approx 0$, if $i > k$: $x_j \approx \bar{x} + \sum_{i=1}^{i=k} g_{ji} e_i$

$\| x_j - \tilde{x}_j \|$

$\tilde{x}_j = \bar{x} + \sum$

Reconstruction error

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

L_2 error and PCA dim

32x32

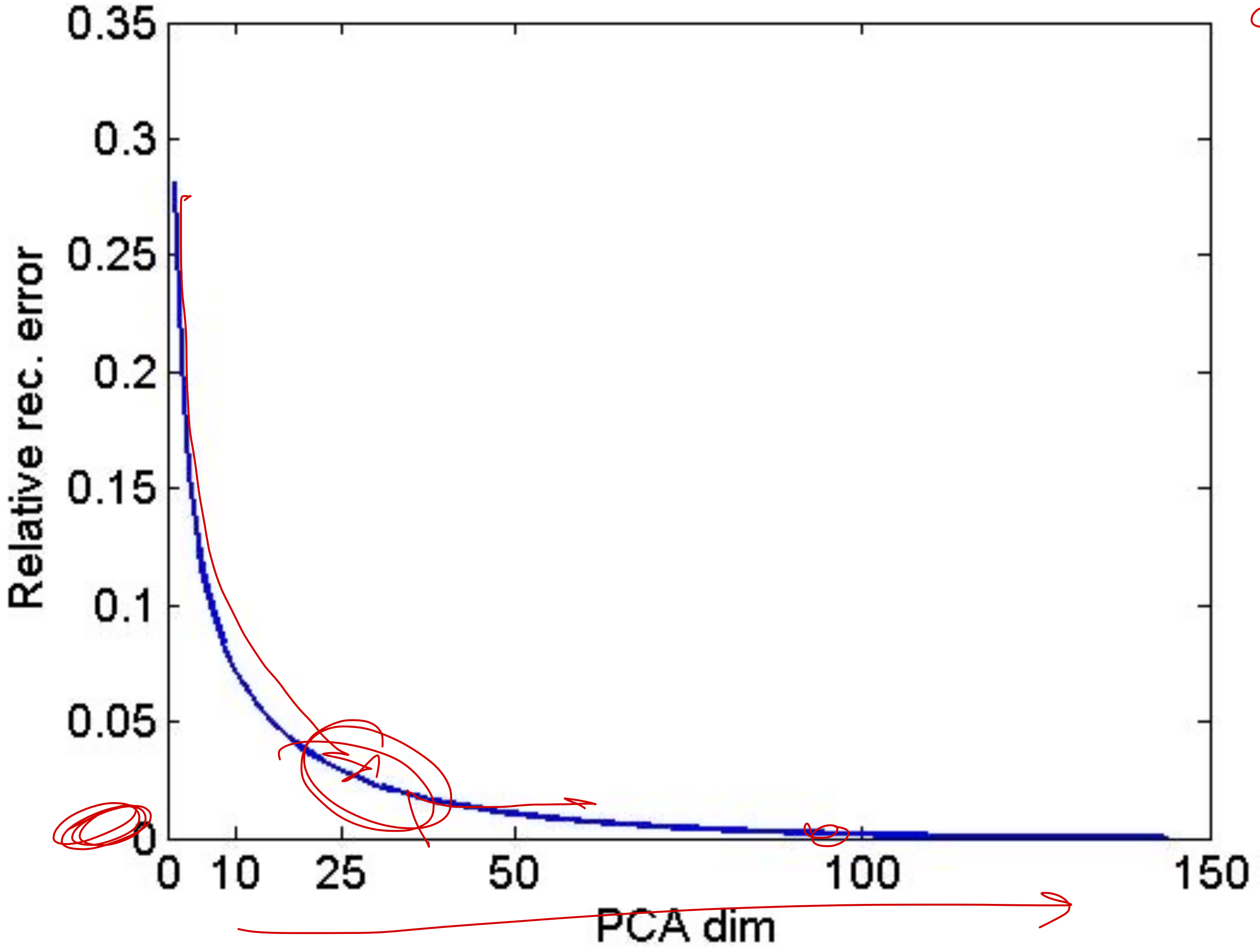
1024

eigenvectors

d = 1024

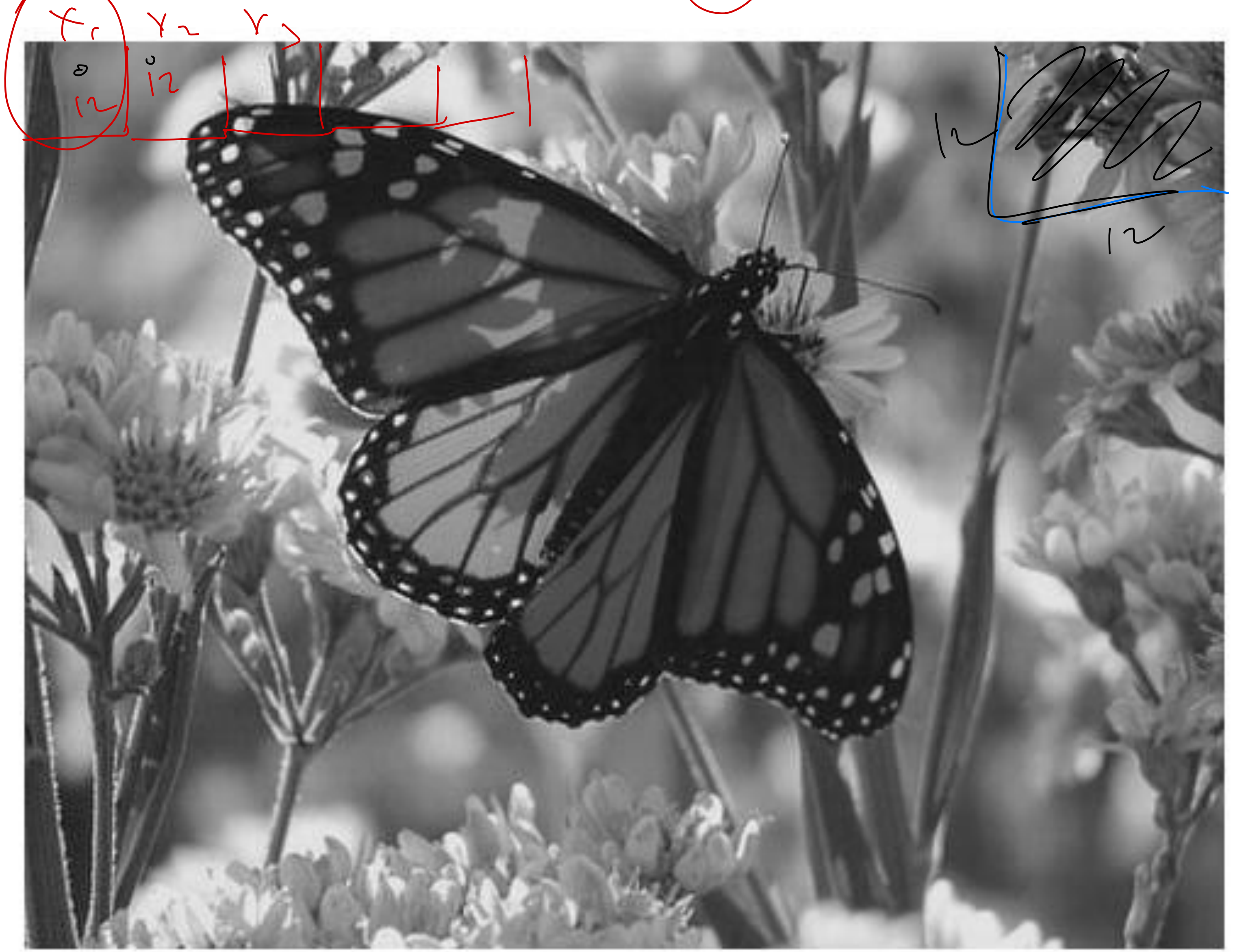
100

coefficients



Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

PCA compression: 144D → 60D

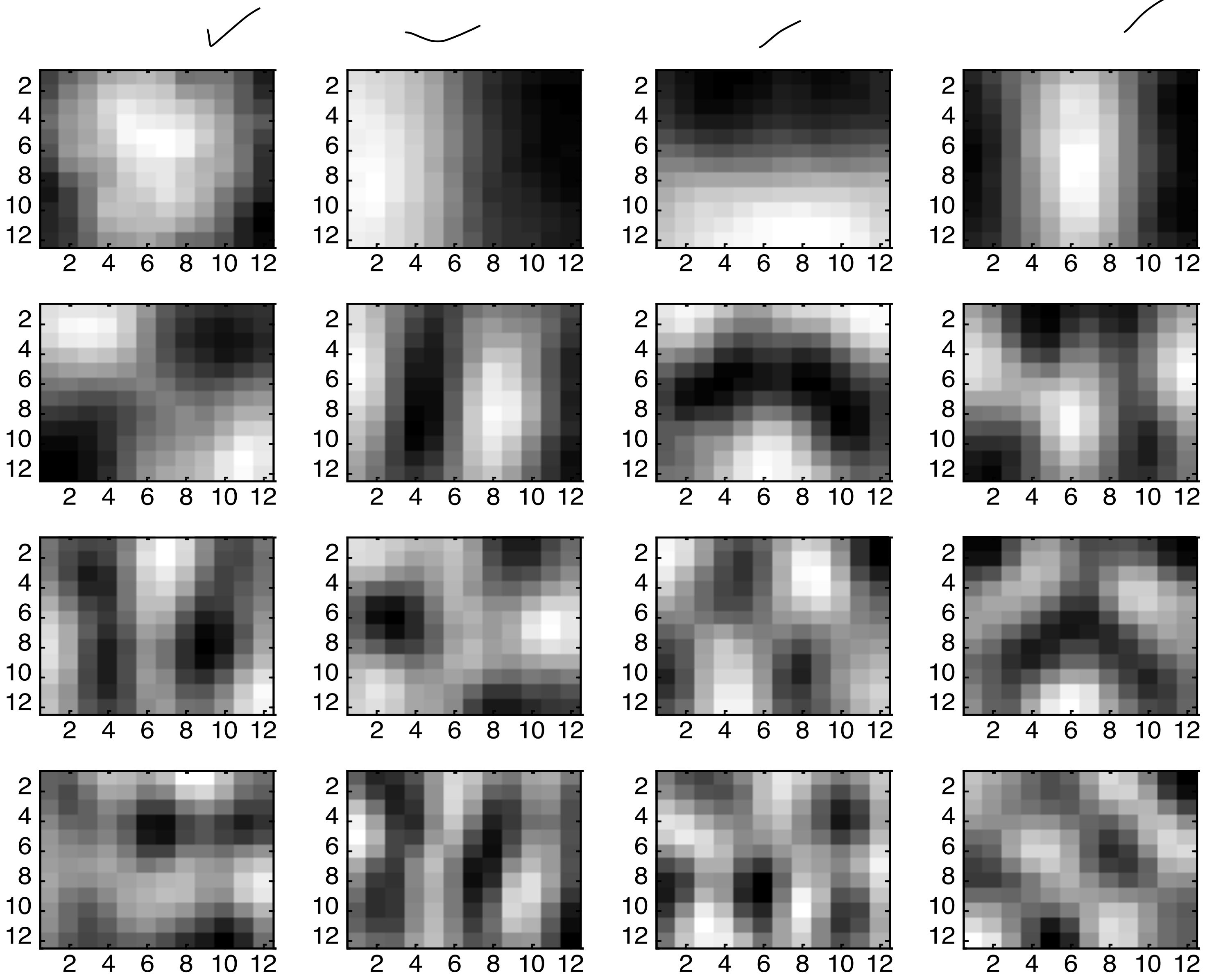


Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

PCA compression: 144D → 16D

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

16 most important eigenvectors

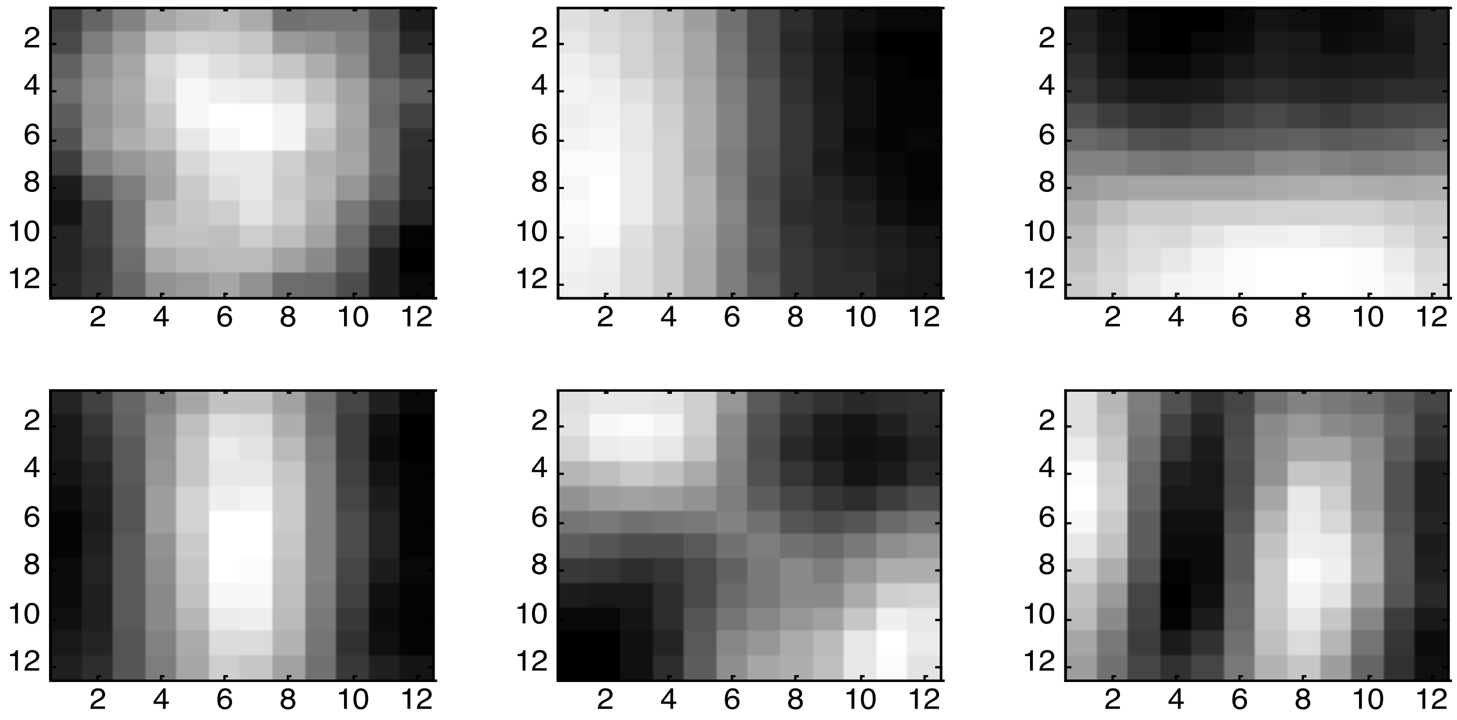


Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

PCA compression: 144D → 6D

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

6 most important eigenvectors

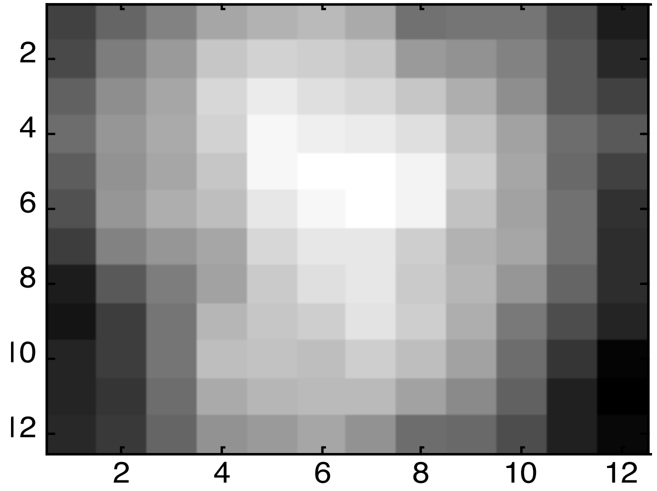


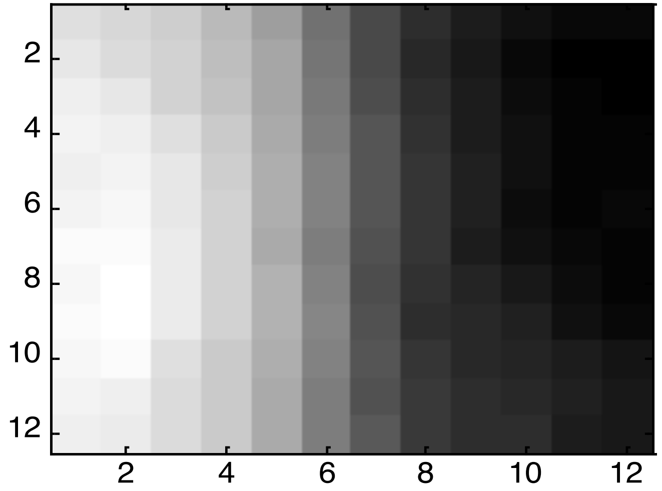
Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

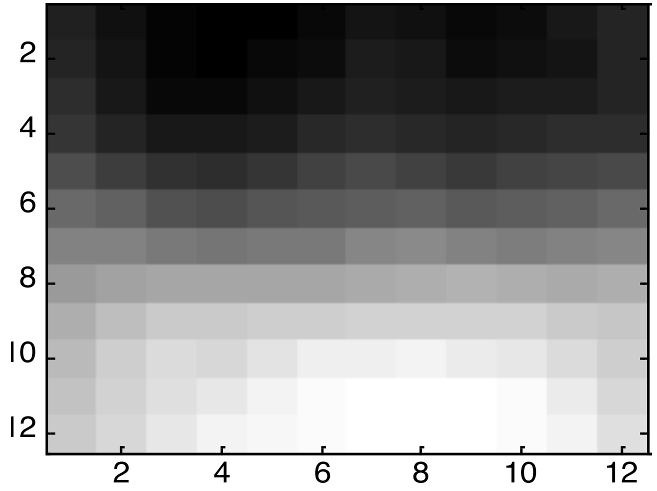
PCA compression: 144D → 3D

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

3 most important eigenvectors

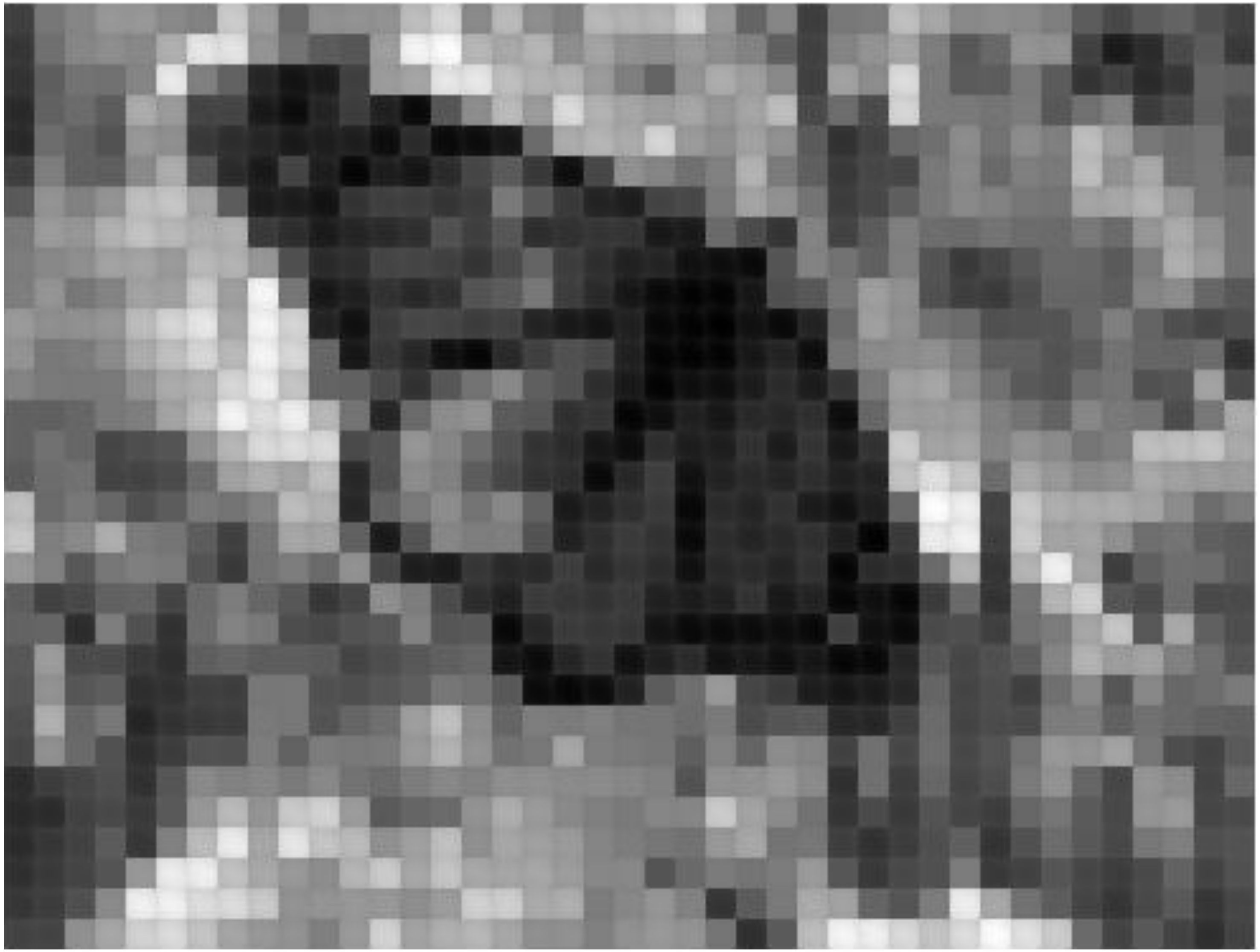






Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

PCA compression: 144D → 1D



Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

PCA in Matrix Form

$U \quad U^T x$

z

$$x_j \approx \bar{x} + [e_1 \quad e_2 \quad \cdots \quad e_d] \begin{bmatrix} e_1{}^T \\ e_d{}^T \\ \vdots \\ e_d{}^T \end{bmatrix} (x_j - \bar{x})$$

Output

input

$$\tilde{x}_j = \cancel{\bar{x}} + UU^T (x_j - \cancel{\bar{x}})$$

$\to$ Autoencoders

$\sim$ NN

A

Low rank approximation

$$= \bar{x} + U \boxed{z_j}$$

Low dimensional representation of x_j

$\|\tilde{x} - x\|_2$

reconstruction

$\tilde{x} \approx x$

$U \Leftarrow U^T \quad x$

z

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

PCA as Minimizing Reconstruction Error

- It can be shown that PCA minimizes the reconstruction error:

$$\min_U \sum_{j=1}^{N} \left\| x_j - \tilde{x}_j \right\|^2 = \sum_{j=1}^{N} \left\| x_j - U z_j \right\|^2$$

Assuming zero mean

$$\min_U \| X - UZ \|_F$$

$\|A\|_F$ shows Frobenius norm, sum of elements-wise squares

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Matrix Completion/Recommender Systems

- The Netflix problem
  - Movie recommendation: Users watch movies and rate them as good/bad.
  - Because users only rate a few items, one would like to infer their preference for unrated items

User	Movie	Rating
	Thor	★ ☆ ☆ ☆ ☆
	Chained	★ ★ ☆ ☆ ☆
	Frozen	★ ★ ★ ☆ ☆
	Chained	★ ★ ★ ★ ☆
	Bambi	★ ★ ★ ★ ★
	Titanic	★ ★ ★ ☆ ☆
	Goodfellas	★ ★ ★ ★ ★
	Dumbo	★ ★ ★ ★ ★
	Twilight	★ ★ ☆ ☆ ☆
	Frozen	★ ★ ★ ★ ★
	Tangled	★ ☆ ☆ ☆ ☆

1 crore

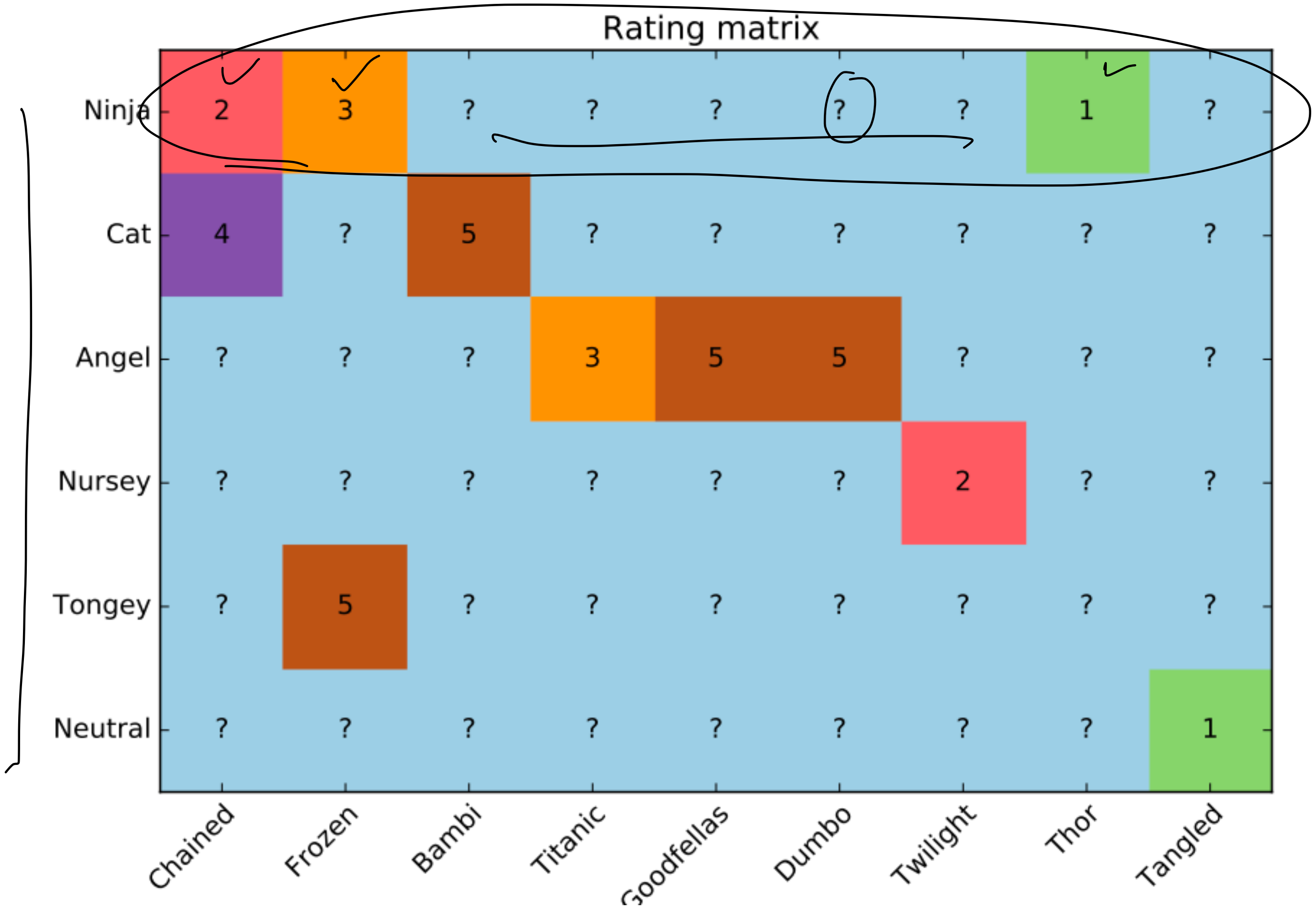
100k

10k × 10k

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Matrix Completion

- Matrix completion problem: Transform the table into a N users by M movies matrix called R



Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Matrix Completion

- Matrix completion problem: Transform the table into a N users by M movies matrix called R

- **Data:** Users rate some movies. R matrix is very sparse.

- **Task:** Predict missing entries, i.e. how a user would rate a movie they haven't previously rated

- **Evaluation Metric:** Squared error (used by Netflix Competition)

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Matrix Completion

$\|R - UZ\|_F$

Sum of element wise squares

- Let the representation of user i in the K-dimensional space be u_i and the representation of movie j be z_j
  - Intuition: maybe the first entry of u_i says how much the user likes horror films, and the first entry of z_j says how much movie j is a horror film.

observed

Frobenius

observed

- Assume the rating user i gives to movie j is given by a dot product:

$$R_{ij} = u_i . z_j$$

$\sum_{ij} R_{ij}^2$

10M, 10K

user i

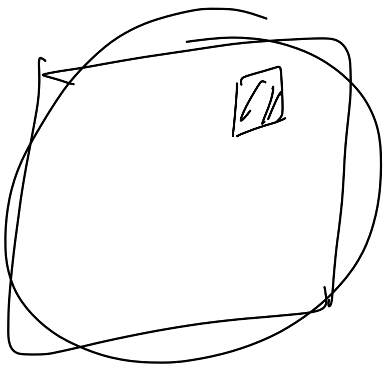
movie j

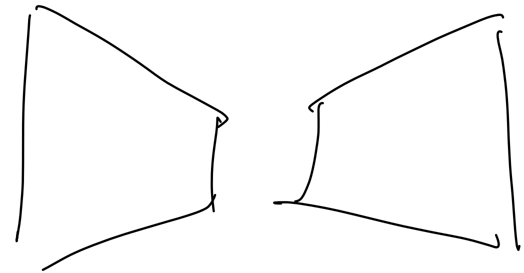
$\|R - UZ\|_F$

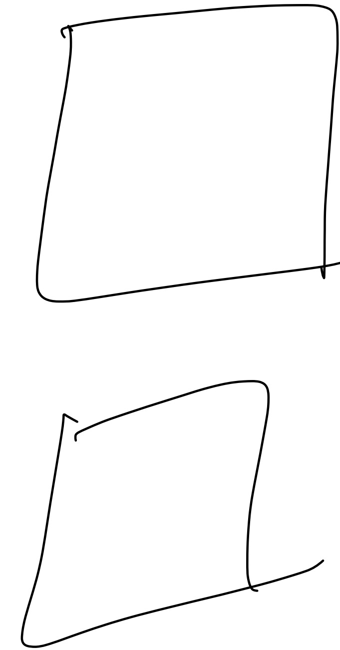
10M × 10

10 × 10M 20K

Pseudo Labels







Masked AE

Im → Encoder

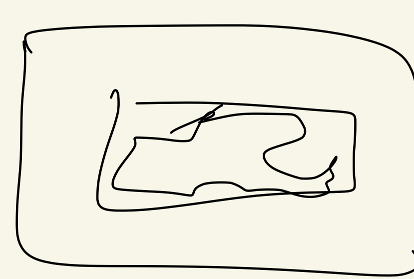
↓

1024

→ Supervised

→ Un- Supervised

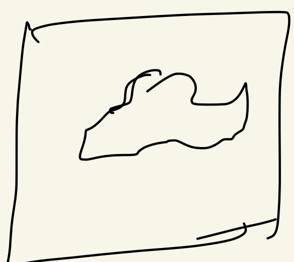
weaky Supervised →



Reinforcement Learning

Semi - supervised

Labeled + unlabeled



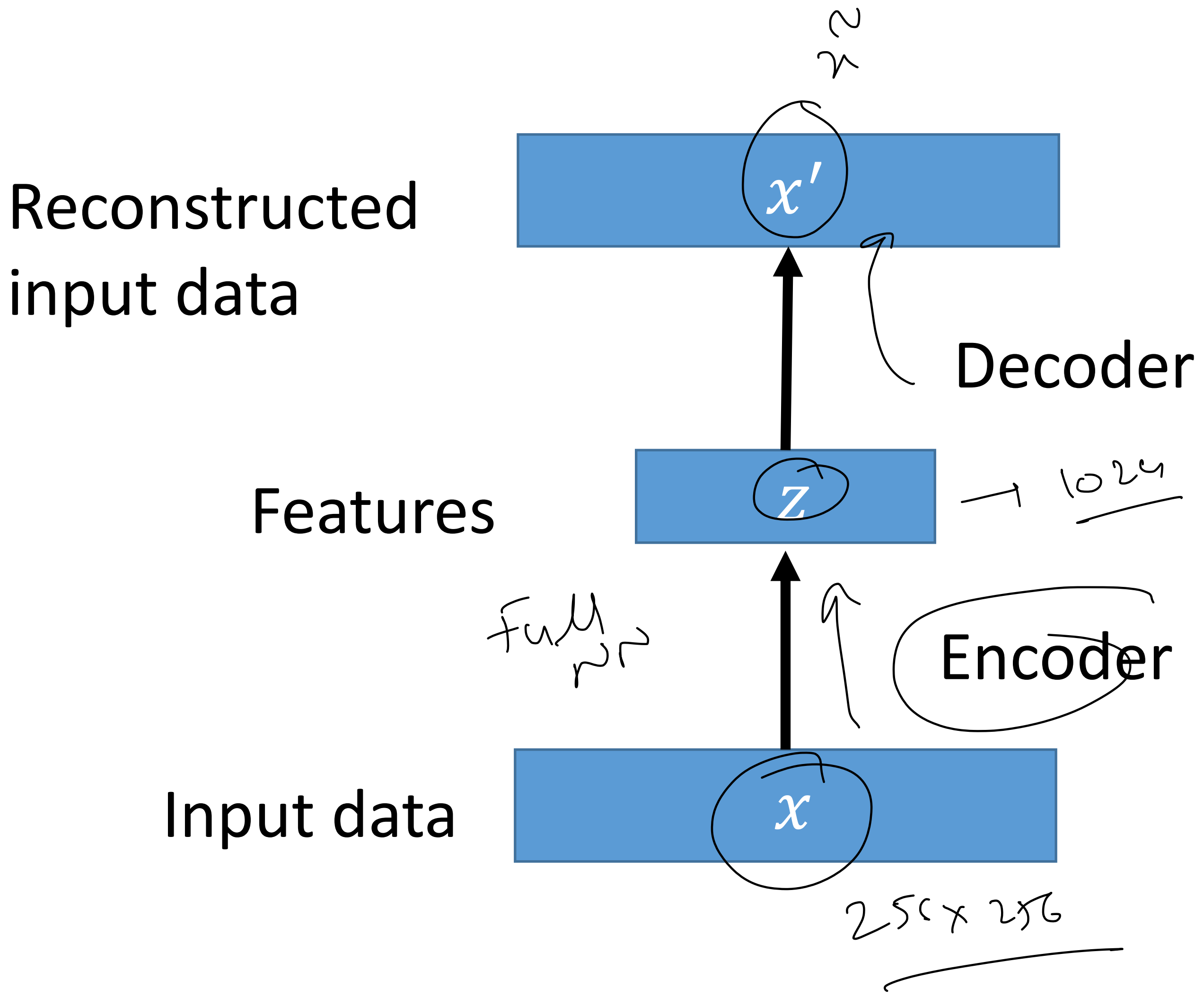
Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Matrix Completion

- In matrix form: $U^T = [u_1 \quad u_2 \quad \dots \quad u_N]$
- In matrix form: $Z^T = [z_1 \quad z_2 \quad \dots \quad z_M]$
- $R = UZ^T$
- **Matrix completion problem:** $\min_{U,Z} \|R - UZ^T\|_F$
- Frobenius norm computed over only observed values.

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Autoencoders as Non-Linear Factorization



U

Autoencoders can reconstruct data, and can learn features to initialize a supervised model

Features capture factors of variation in training data. Can we generate new images from an autoencoder?

$\|x - x'\|_2$

$\tilde{x} = U \underbrace{U^T x}_{z}$