University of Southampton

COMP6237 Data Mining
Lecture 3: Covariance, EVD, PCA & SVD
(Dimensionality Reduction I)

Zhiwu Huang

Zhiwu.Huang@soton.ac.uk

Lecturer (Assistant Professor) @ VLC of ECS

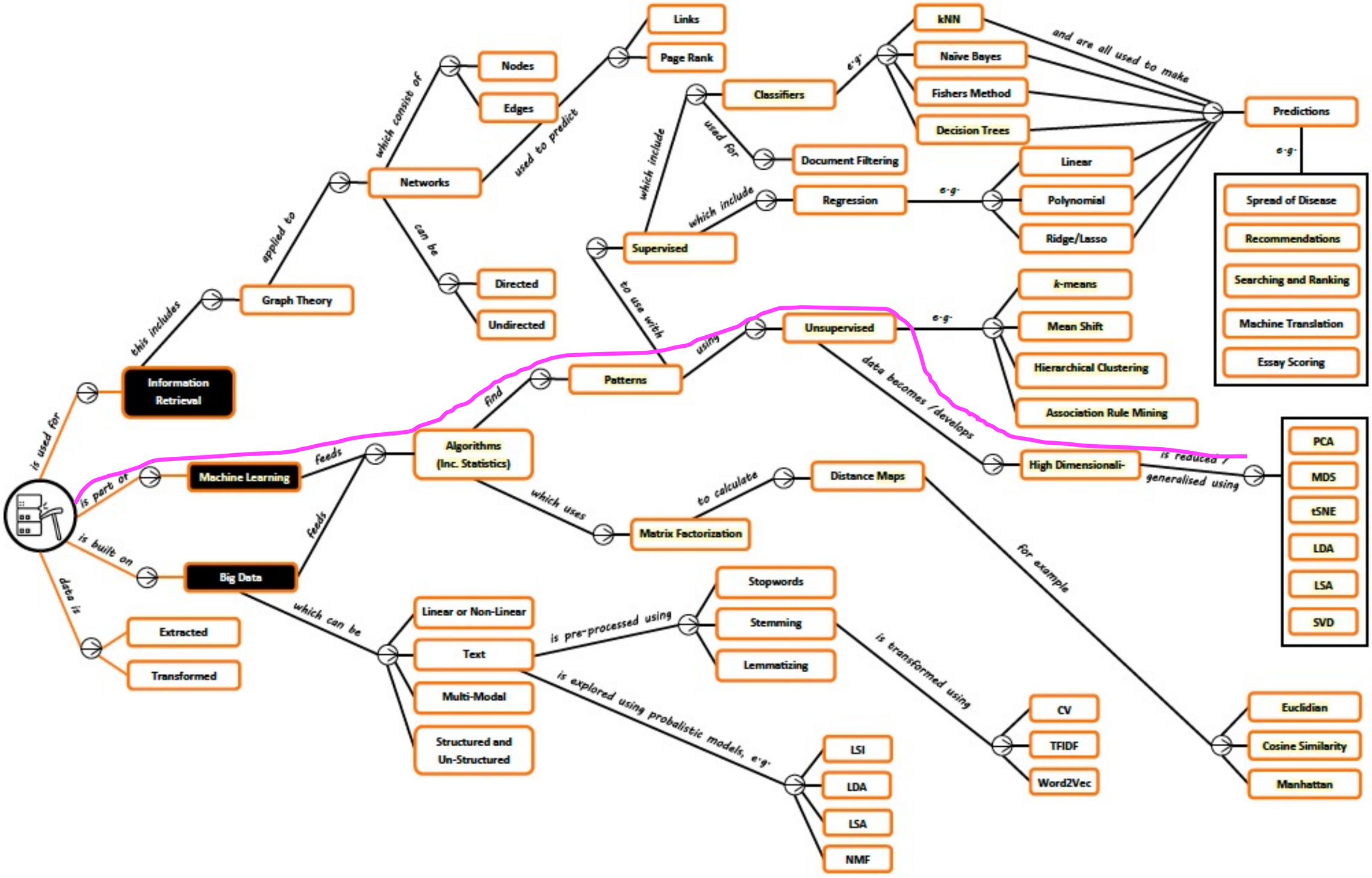
University of Southampton

Lecture slides available here:

http://comp6237.ecs.soton.ac.uk/jon.html

(Thanks to Prof. Jonathon Hare and Dr. Jo Grundy for providing the lecture materials used to develop the slides.)

Dimensionality Reduction I – **Roadmap**



Dimensionality Reduction I – **Textbook**

CHAPTER 7 Dimensionality Reduction

We saw in Chapter 6 that high-dimensional data has some peculiar characteristics, some of which are counterintuitive. For example, in high dimensions the center of the space is devoid of points, with most of the points being scattered along the surface of the space or in the corners. There is also an apparent proliferation of orthogonal axes. As a consequence high-dimensional data can cause problems for data mining and analysis, although in some cases high-dimensionality can help, for example, for nonlinear classification. Nevertheless, it is important to check whether the dimensionality can be reduced while preserving the essential properties of the full data matrix. This can aid data visualization as well as data mining. In this chapter we study methods that allow us to obtain optimal lower-dimensional projections of the data.

- Data Mining and Machine Learning: Fundamental Concepts and Algorithms M. L. Zaki and W. Meira

Chapter 11

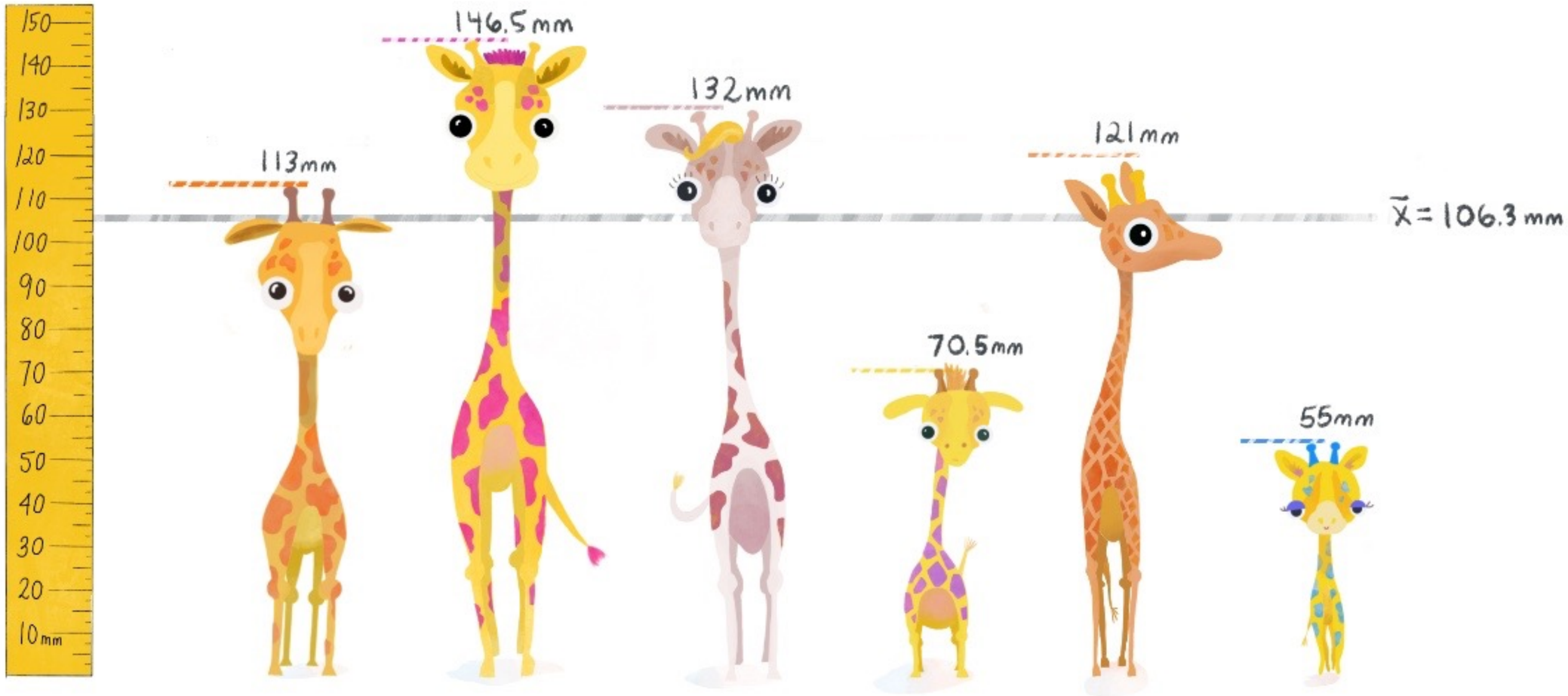
Dimensionality Reduction

There are many sources of data that can be viewed as a large matrix. We saw in Chapter 5 how the Web can be represented as a transition matrix. In Chapter 9, the utility matrix was a point of focus. And in Chapter 10 we

- Mining of Massive Datasets J. Leskovec *et al*

Dimensionality Reduction I – **Overview (1/3)**

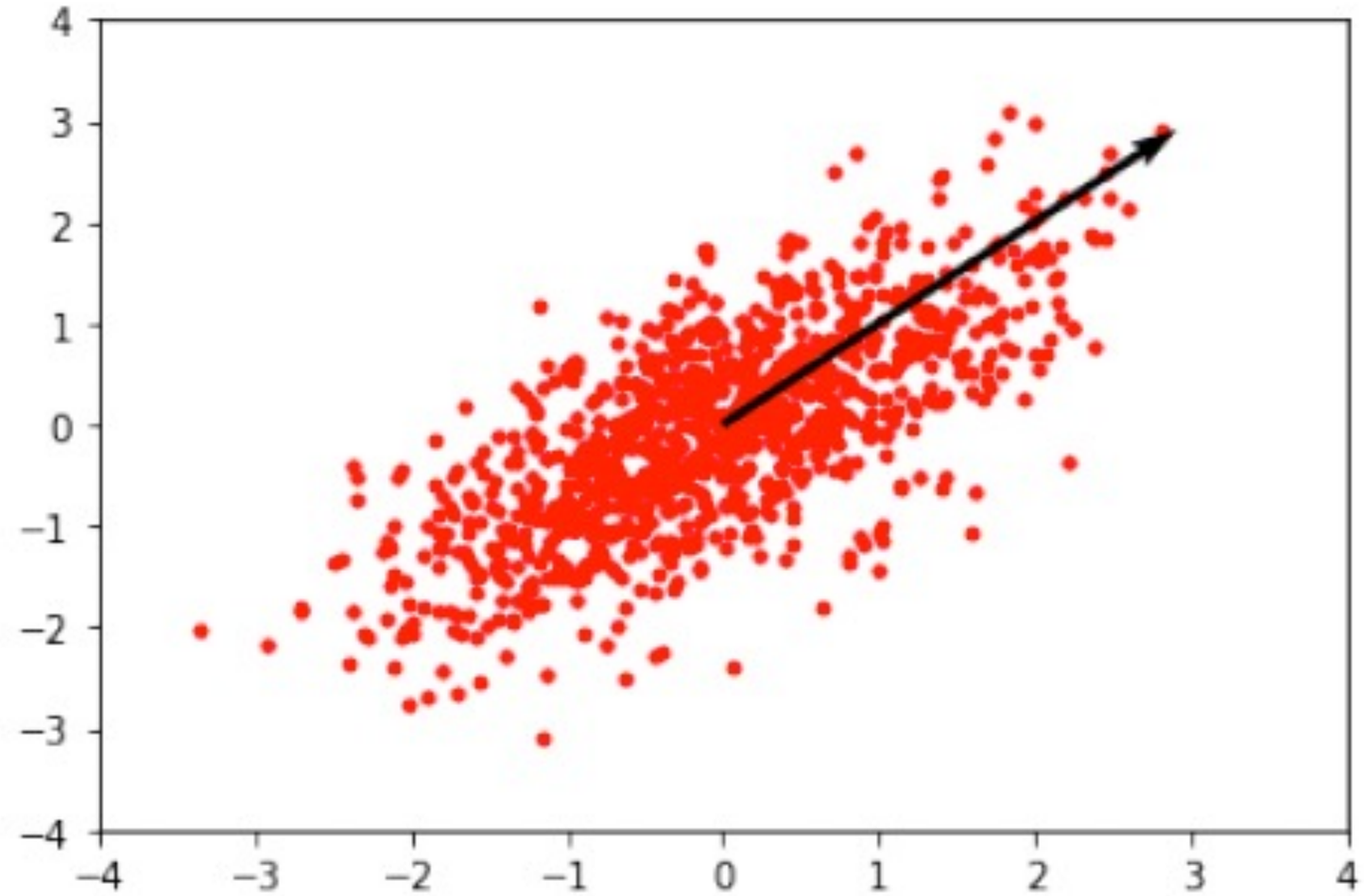
What is Variation/Variance?



https://tinystats.github.io/teacups-giraffes-and-statistics/04_variance.html

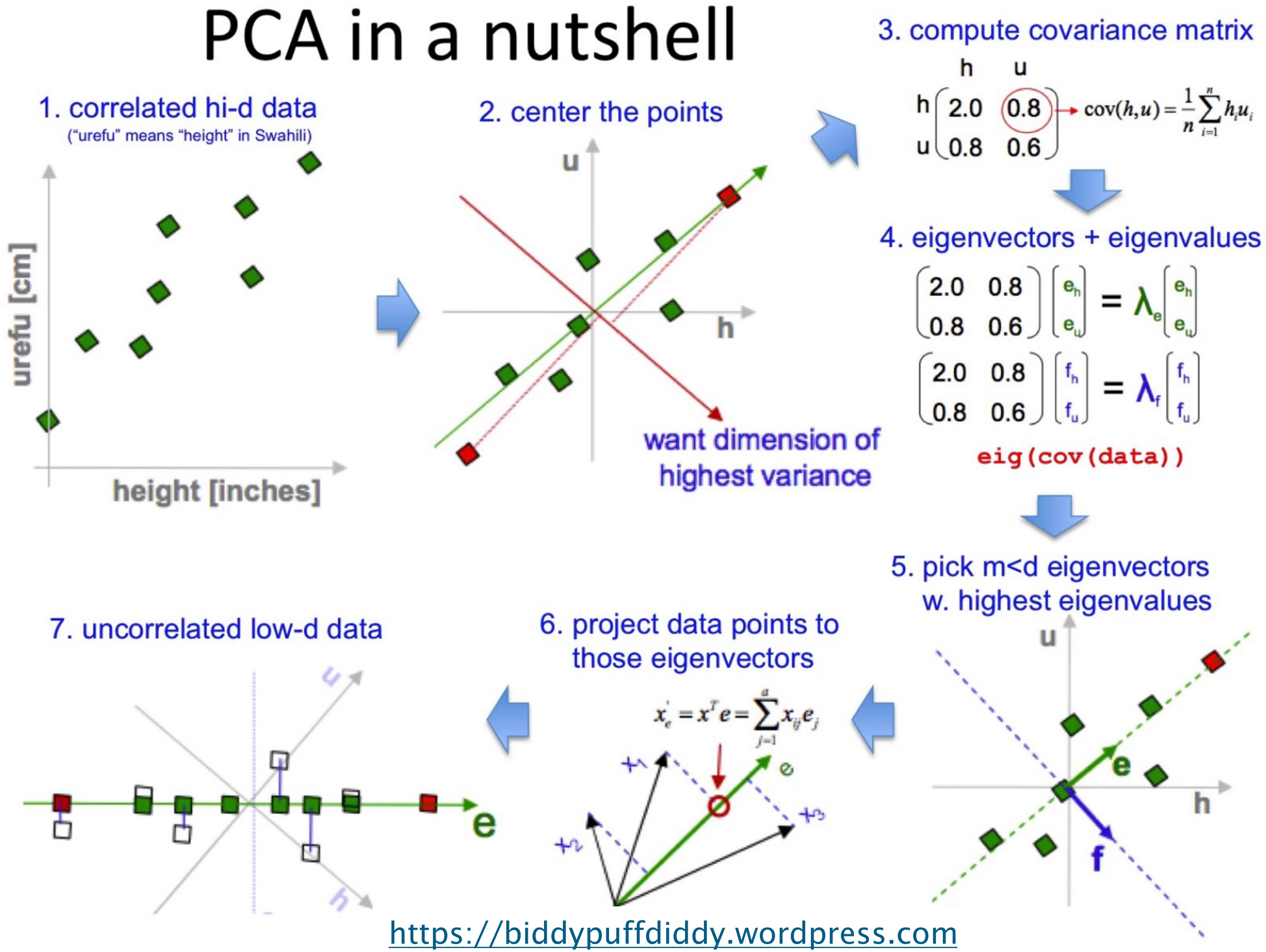
Dimensionality Reduction I – **Overview (2/3)**

Principal axes of variation:
1st principal axis: direction of greatest variance



Credit: Jo Grundy

Dimensionality Reduction I – **Overview (3/3)**



https://biddypuffdiddy.wordpress.com

Dimensionality Reduction I – **Learning Outcomes**

- **LO1**: Compute basic statistics such as variance and covariance over the given data (exam)
- **LO2**: Understand the key ideas and steps of PCA using EVD, SVD and truncated SVD (exam).
- **LO3**: Implement the PCA algorithm for dimensionality reduction (course work)

Assessment hints: Multi-choice Questions (single answer: concepts, calculation etc)

- *Textbook Exercises: textbooks (Programming + Mining)*
- *Other Exercises: https://www-users.cse.umn.edu/~kumar001/dmbook/sol.pdf*
- *ChatGPT or other AI-based techs*

Dimensionality Reduction I - Introduction

Recap :

- Expectation and Variance
- Covariance
- Basis set
- PCA
- SVD, truncated SVD

Credit: Jo Grundy

Dimensionality Reduction I - Expectation

A random variable takes on different values due to chance

The sample values from a single dimension of a featurespace can be considered to be a random variable

The expected value $E[X]$ is the most likely value a random variable will take.

If we assume that the values an element of a feature can take are all equally likely then the expected value is just the mean value.

Credit: Jo Grundy

Dimensionality Reduction I - Variance

Variance = The expected squared difference from the mean

$$E[(X - E[X])^2]$$

i.e. the mean squared difference from the mean

$$\sigma^2(x) = \frac{1}{n} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

A measure of how spread out the data is

Credit: Jo Grundy

Dimensionality Reduction I - Covariance

Covariance = the product of the expected difference between each feature and its mean

$$E[(x - E[x])(y - E[y])]$$

i.e. it measures how two variables change together

$$\sigma(x, y) = \frac{1}{n}\sum_{i=1}^{n}(x - \mu_x)(y - \mu_y)$$

When both variables are the same, covariance = variance, as $\sigma(x, x) = \sigma^2(x)$
If $\sigma^2 = 0$ then the variables are uncorrelated

Credit: Jo Grundy

Dimensionality Reduction I - Covariance

A covariance matrix encodes how all features vary together
For two dimensions:

$$\begin{bmatrix} \sigma(x,x) & \sigma(x,y) \\ \sigma(y,x) & \sigma(y,y) \end{bmatrix}$$

For n dimensions:

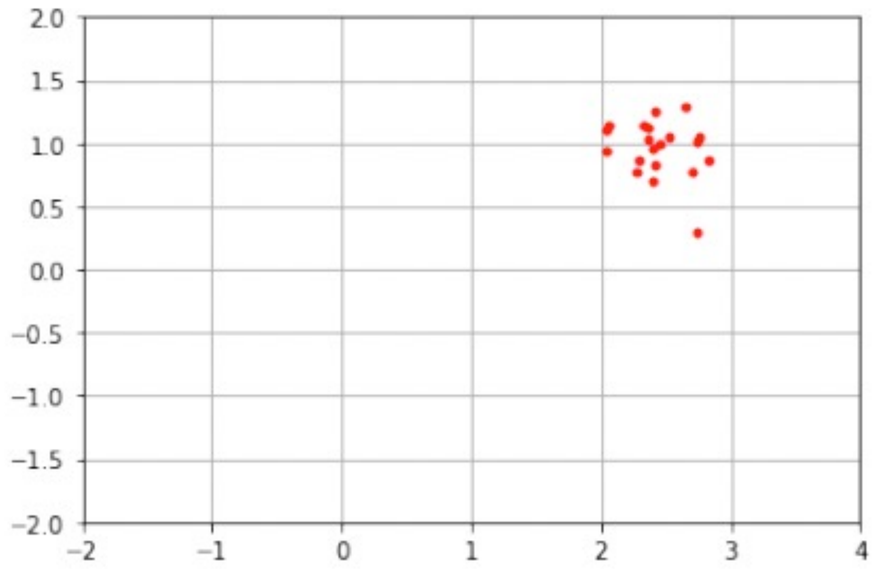
$$\begin{bmatrix} \sigma(x_1,x_1) & \sigma(x_1,x_2) & \dots & \sigma(x_1,x_n) \\ \sigma(x_2,x_1) & \sigma(x_2,x_2) & \dots & \sigma(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(x_n,x_1) & \sigma(x_n,x_2) & \dots & \sigma(x_n,x_n) \end{bmatrix}$$

This matrix must be square symmetric
(x cannot vary with y differently to how y varies with x!)
2d covariance demo

Credit: Jo Grundy

Dimensionality Reduction I - Covariance

Mean Centering = subtract the mean of all the vectors from each vector
This gives centered data, with mean at the origin

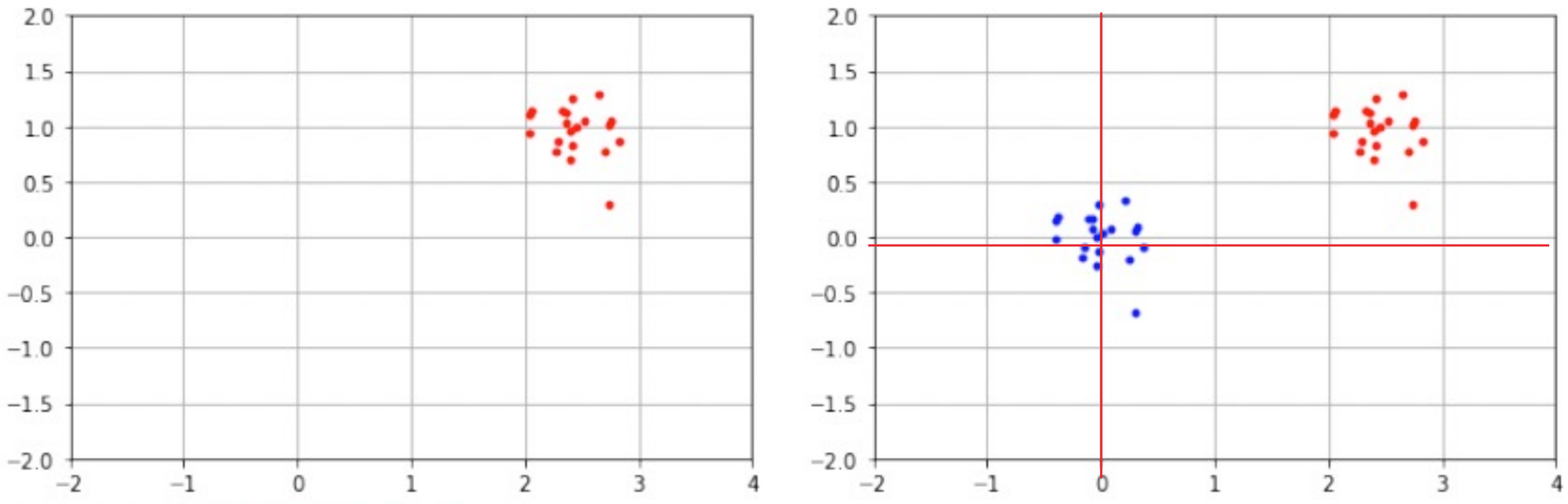


2D Covariance: $\sigma(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x - \mu_x)(y - \mu_y)$

Credit: Jo Grundy

Dimensionality Reduction I - Covariance

Mean Centering = subtract the mean of all the vectors from each vector
This gives centered data, with mean at the origin



ipynb mean centering demo https://github.com/zhiwu-huang/Data-Mining-Demo-Code-18-19

2D Covariance: $\sigma(x, y) = \frac{1}{n}\sum_{i=1}^{n}(x - \mu_x)(y - \mu_y)$

d-D Covariance: $C(X) = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{i=1}^{n}(X_i - \mu)^T(X_i - \mu)$, *where* $X_i = [X_{i1}, \ldots, X_{id}]$ is uncentred data, $\mu = [\mu_{i1}, \ldots, \mu_{id}]$.

Credit: Jo Grundy

Dimensionality Reduction I - Covariance

If you have a set of mean centred data with d dimensions, where each row is your data point:

$$Z = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \end{bmatrix}, \text{ where } x_{ij} = X_{ij} - \mu_{ij}$$

Then its inner product is proportional to the covariance matrix

$$C \propto Z^T Z$$

d-D Covariance: $C(X) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^T(X_i - \mu)\,, where\ X_i = [X_{i1}, \dots, X_{id}]$ is uncentred data, $\mu = [\mu_{i1}, \dots, \mu_{id}]$.

ipynb mean centering demo https://github.com/zhiwu-huang/Data-Mining-Demo-Code-18-19

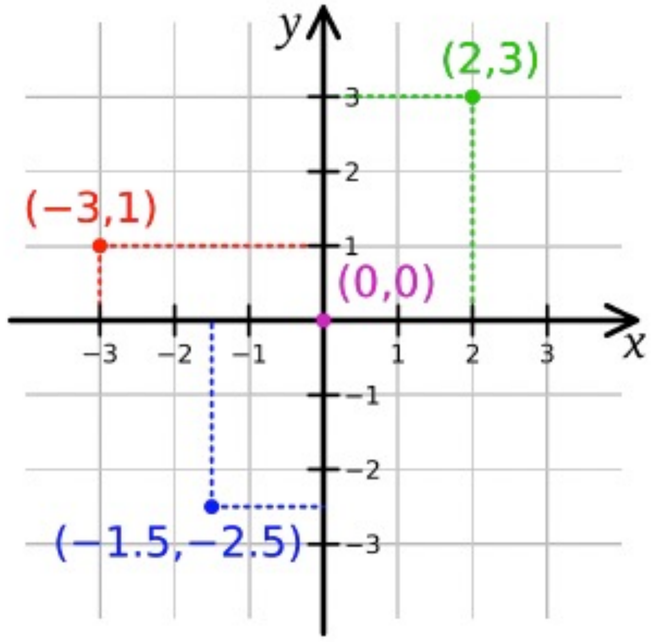
Credit: Jo Grundy

Dimensionality Reduction I – Basis Set

In linear algebra, a basis set is defined for a space with the properties:

- They are all linearly independent
- They span the whole space

Every vector in the space can be described as a combination of basis vectors



Using Cartesian coordinates, we describe every vector as a combination of x and y directions

Credit: Jo Grundy

Dimensionality Reduction I – Basis Set

Eigenvectors and eigenvalues

An eigenvector is a vector that when multiplied by a matrix $\boldsymbol{A}$ gives a value that is a multiple of itself, i.e.:

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$$

The eigenvalue λ is the multiple that it should be multiplied by.

eigen comes from German, meaning 'Characteristic' or 'Own'

for an $n \times n$ dimensional matrix $\boldsymbol{A}$ there are n eigenvector-eigenvalue pairs

Credit: Jo Grundy

Dimensionality Reduction I – Basis Set

So if $\boldsymbol{A}$ is a covariance matrix, then the eigenvectors are its principal axes.

The eigenvalues are proportional to the variance of the data along each eigenvector

The eigenvector corresponding to the largest eigenvalue is the first principal axis

Credit: Jo Grundy

Dimensionality Reduction I – Basis Set

To find eigenvectors and eigenvalues for smaller matrices, there are algebraic solutions, and all values can be found
For larger matrices, numerical solutions are found, using eigendecomposition.
Eigen - Value - Decomposition (EVD):

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}$$

Where $\boldsymbol{Q}$ is a matrix where the columns are the eigenvectors, and $\boldsymbol{\Lambda}$ is a matrix with eigenvalues along the corresponding diagonal

Covariance matrices are real symmetric, so $Q^{-1} = Q^T$ Therefore:

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\boldsymbol{T}}$$

This diagonalisation of a covariance matrix gives the principal axes and their relative magnitudes

Credit: Jo Grundy

Dimensionality Reduction I – Basis Set

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\boldsymbol{T}}$$

This diagonalisation of a covariance matrix gives the principal axes and their relative magnitudes

Usually the implementation will order the eigenvectors such that the eigenvalues are sorted in order of decreasing value.
Some solvers only find the top k eigenvalues and corresponding eigenvectors, rather than all of them.
Java demo: EVD and component analysis

Credit: Jo Grundy

Dimensionality Reduction I – PCA

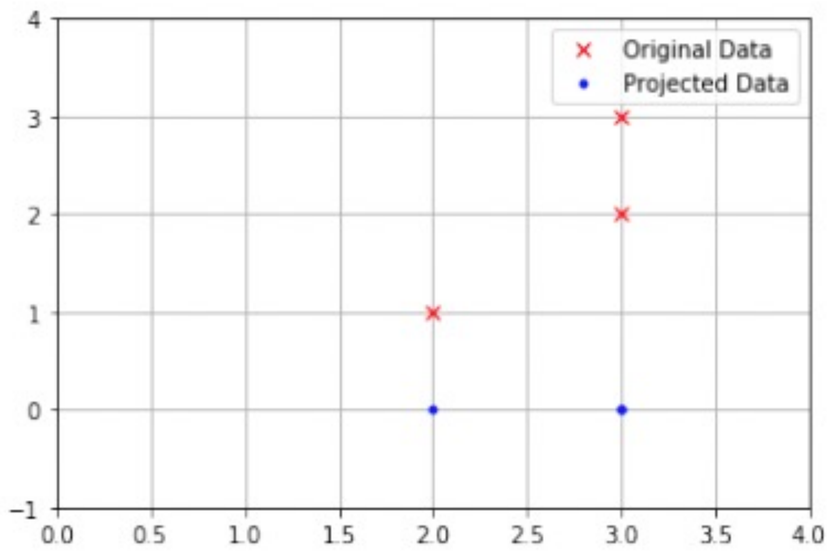
Principal Component Analysis - PCA

Projects the data in to a lower dimensional space, while keeping as much of the information as possible.

$$X = \begin{bmatrix} 2 & 1 \\ 3 & 2 \\ 3 & 3 \end{bmatrix}$$

For example: data set X can be transformed so only information from the x dimension is retained using a projection matrix P:

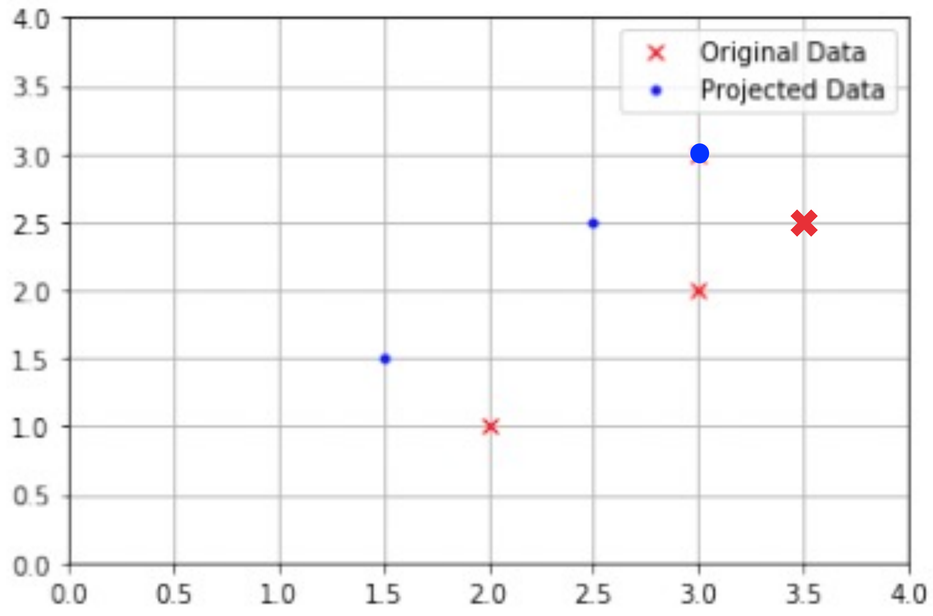
$X_p = XP$



Credit: Jo Grundy

Dimensionality Reduction I – PCA

However if a different line is chosen, more information can be retained.

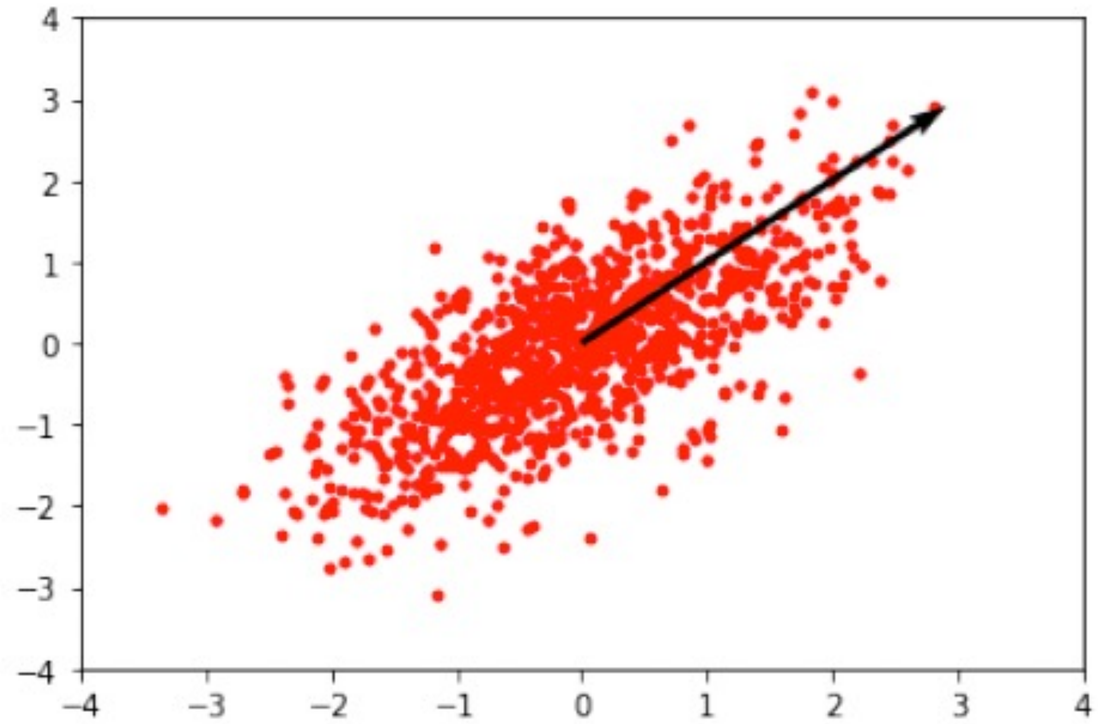


This process can be reversible, (Using $\hat{X} = X_p P^{-1}$) but this is lossy if the dimensionality has been changed.

Credit: Jo Grundy

Dimensionality Reduction I – PCA

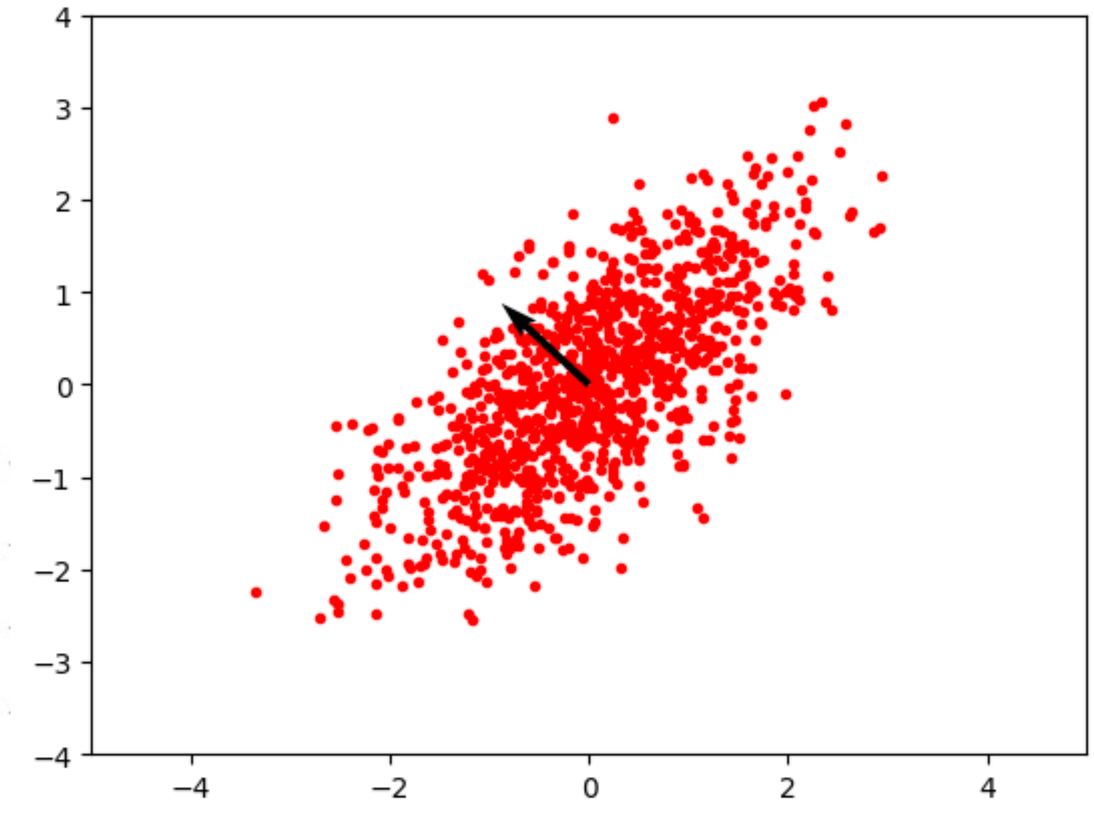
Principal axes of variation:
1st principal axis: direction of greatest variance



Credit: Jo Grundy

Dimensionality Reduction I - PCA

2nd Principal axis: direction orthogonal to 1st principal axis in direction of greatest variance



Credit: Jo Grundy

Dimensionality Reduction I – PCA

In PCA, a line is chosen that minimises the orthogonal distances of the data points from the projected space.
It does this by keeping the dimensions where it has the most variation, i.e. using the directions provided by the eigenvectors corresponding to the largest eigenvalues of the estimated covariance matrix
It uses the mean centred data to give the matrix proportional to the covariance matrix

(ipynb projection demo) https://github.com/zhiwu-huang/Data-Mining-Demo-Code-18-19

Credit: Jo Grundy

Dimensionality Reduction I – PCA

Algorithm 1: PCA algorithm using EVD

Data: N data points with feature vectors X_i $i = 1 \ldots N$

$Z = \text{meanCentre}(X)$;

$eigVals, eigVects = \text{eigendecomposition}(Z^T Z)$;

take k $eigVects$ corresponding to k largest $eigVals$;

make projection matrix P;

Project data $Xp = ZP$ in to lower dimensional space;

(Java PCA demo)

Credit: Jo Grundy

Dimensionality Reduction I – SVD

Eigenvalue Decomposition, EVD, $A = Q\Lambda Q^T$ only works for symmetric matrices.
Singular value decomposition - SVD

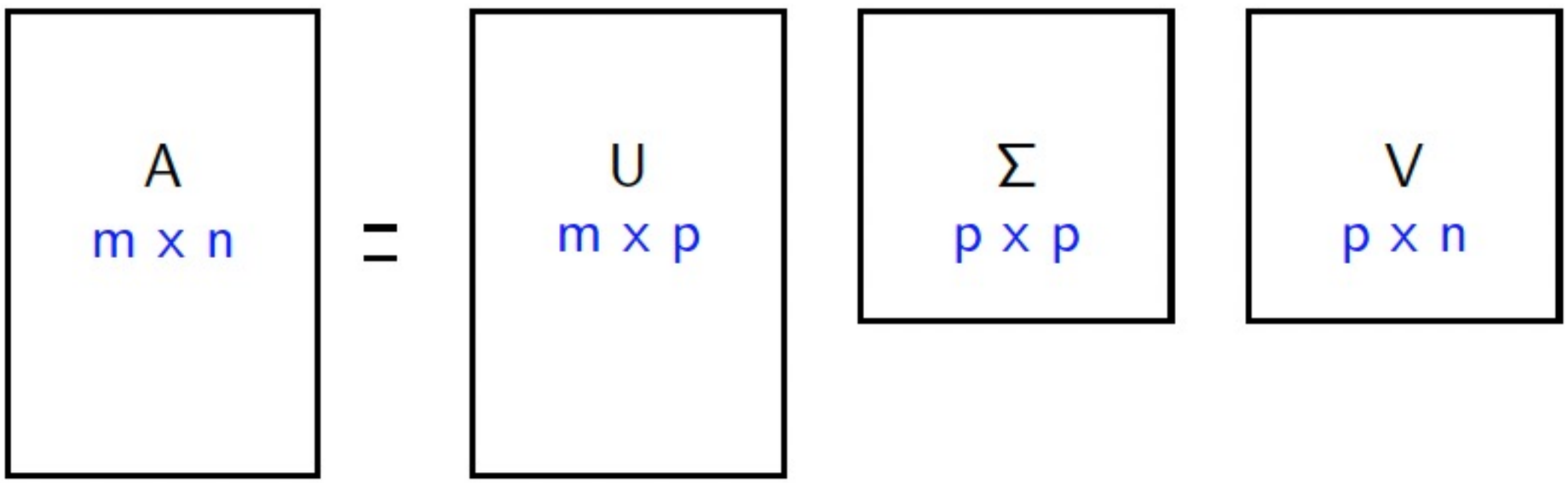
$$A = U\Sigma V^T$$

where U and V are both different orthogonal matrices, and Σ is a diagonal matrix
Any matrix can be factorised this way.

Orthogonal matrices are where each column is a vector pointing in an othogonal direction to each other, $U^T U = UU^T = I$

Credit: Jo Grundy

Dimensionality Reduction I – SVD



Where p is rank of matrix A

U called *left singular vectors*, contains the eigenvectors of AA^T,
V called *right singular vectors*, contains the eigenvectors of $A^T A$
Σ contains square roots of eigenvalues of AA^T and $A^T A$

If A is matrix of mean centred featurevectors, V contains principal components of the covariance matrix

Credit: Jo Grundy

Dimensionality Reduction I – SVD

Algorithm 2: PCA algorithm using SVD

Data: N data points with feature vectors X_i $i = 1 \ldots N$

$Z = \text{meanCentre}(X)$;

$U, \Sigma, V = \text{SVD}(Z)$;

take k columns of V corresponding to the largest k values of Σ;

make projection matrix P;

Project data $Xp = ZP$ in to lower dimensional space;

Better than using EVD of $Z^T Z$ as:

- has better numerical stability
- can be faster

(ipynb projection demo) https://github.com/zhiwu-huang/Data-Mining-Demo-Code-18-19

Credit: Jo Grundy

Dimensionality Reduction I – SVD

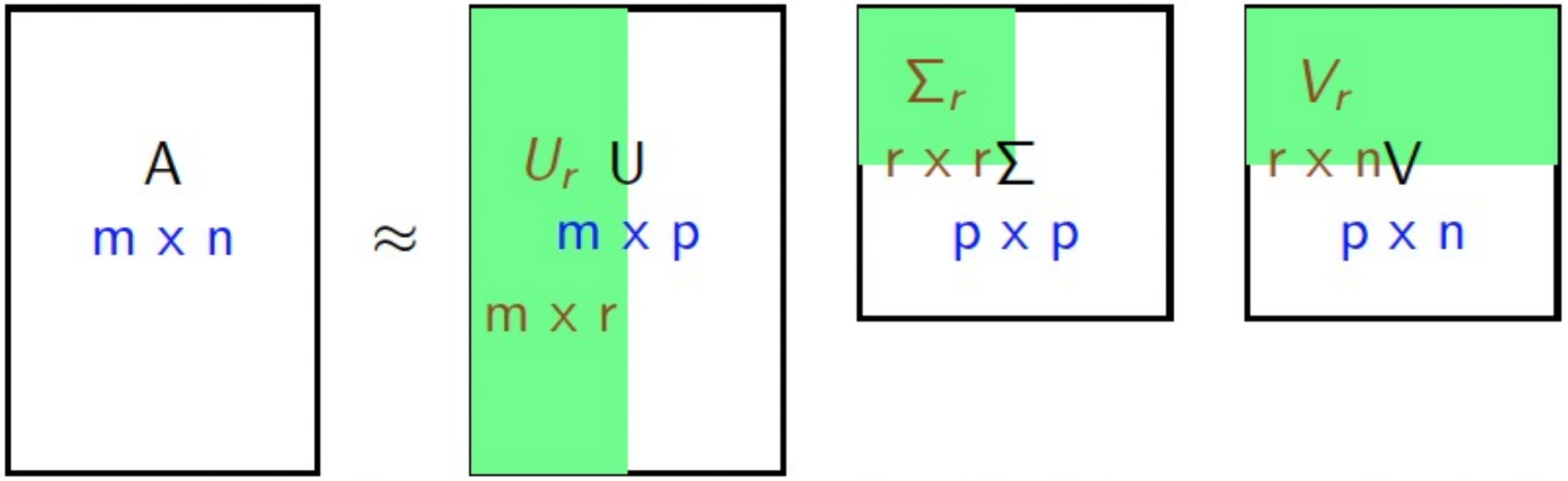
SVD has better numerical stability:
E.g. Läuchli matrix:

$$X^T X = \begin{bmatrix} 1 & \epsilon & 0 & 0 \\ 1 & 0 & \epsilon & 0 \\ 1 & 0 & 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix} = \begin{bmatrix} 1+\epsilon^2 & 1 & 1 \\ 1 & 1+\epsilon^2 & 1 \\ 1 & 1 & 1+\epsilon^2 \end{bmatrix}$$

If ϵ is very small, $1 + \epsilon^2$ will be counted as 1, so information is lost

Credit: Jo Grundy

Dimensionality Reduction I – Truncated SVD



Uses only the largest r singular values (and corresponding left and right vectors)

This can give a *low rank approximation* of A, $\tilde{A} = U_r \Sigma_r V_r$

has the effect of minimising the Frobenius norm of the difference between A and $\tilde{A}$

Credit: Jo Grundy

Dimensionality Reduction I – Truncated SVD

SVD can be used to give a Pseudoinverse:

$$A^{+} = V\Sigma^{-1}U^{T}$$

This is used to solve $\boldsymbol{A}x = b$ for x where $||\boldsymbol{A}x - b||_2$ is minimised i.e. in least squares regression $x = \boldsymbol{A}^{+}b$

Also useful in solving homogenous linear equations $\boldsymbol{A}x = 0$

SVD has also found application in:

- model based CF recommender systems
- latent factors
- image compression
- and much more..

Credit: Jo Grundy

Dimensionality Reduction I – Summary

Covariance measures how different dimensions change together:

- Represented by a matrix
- Eigenvalue decomposition gives eigenvalue - eigenvector pairs
- The dominant eigenvector gives the principal axis
- The Eigenvalue is proportional to the variance along that axis
- The principal axes give a basis set, describing the directions of greatest variance

PCA: aligns data with its principal axes, allows dimensional reduction losing the least information by discounting axes with low variance

SVD: a general matrix factorisation tool with many uses.

Credit: Jo Grundy