COMP20008 Elements of Data Processing

Announcements

- Assignments (Phase 1-4) are available via LMS
- Answer to workshop 2 will be released next Monday - March 20th

Plan today

- Answer some questions
- Complete section of collaborative filtering
  - Item item similarity
  - Matrix factorisation
- Basic visualisation methods
  - Scatter plots, heat maps, parallel co-ordinates

Practice Example

User1	**12**	**2.5**	**20**	**-**	**17**	**-**	**3.5**
User2	**13**	**-**	**-**	**17**	**14**	**17.5**	**4.5**

SIM(User1,User2)=?

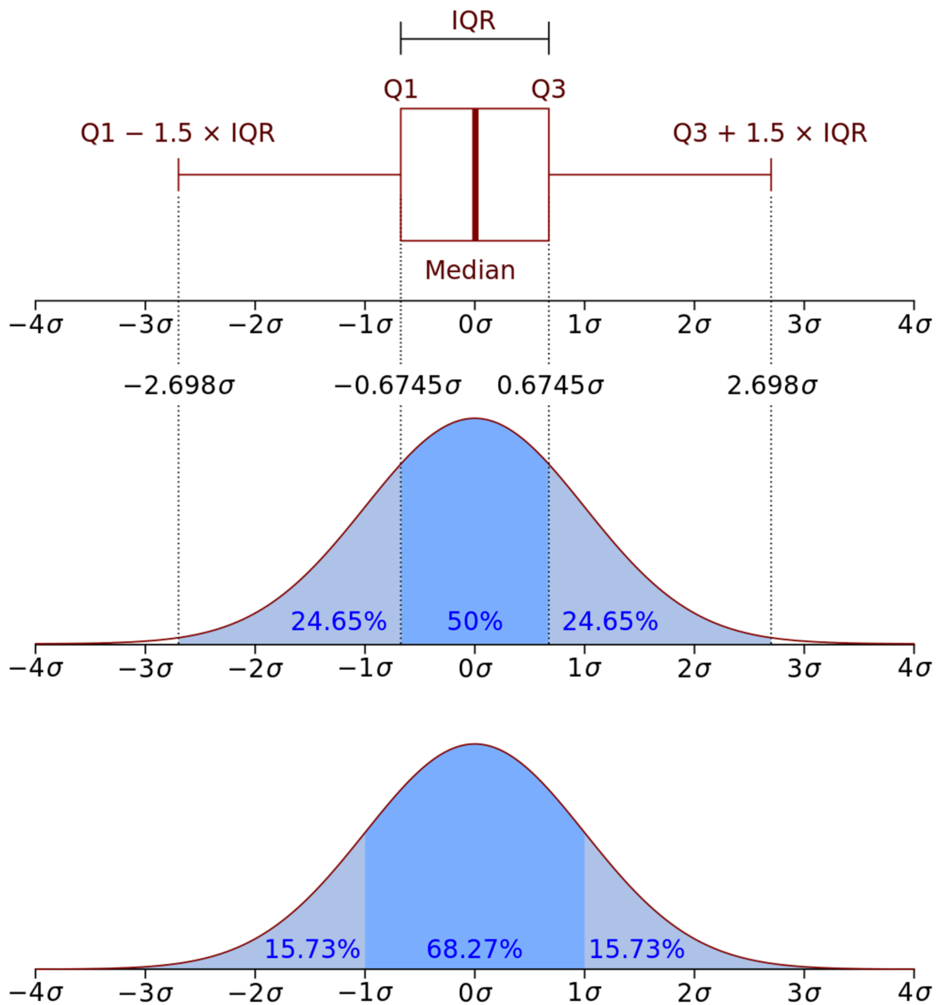
Practice Example

User1	**12**	**2.5**	**20**	**-**	**17**	**-**	**3.5**
User2	**13**	**-**	**-**	**17**	**14**	**17.5**	**4.5**

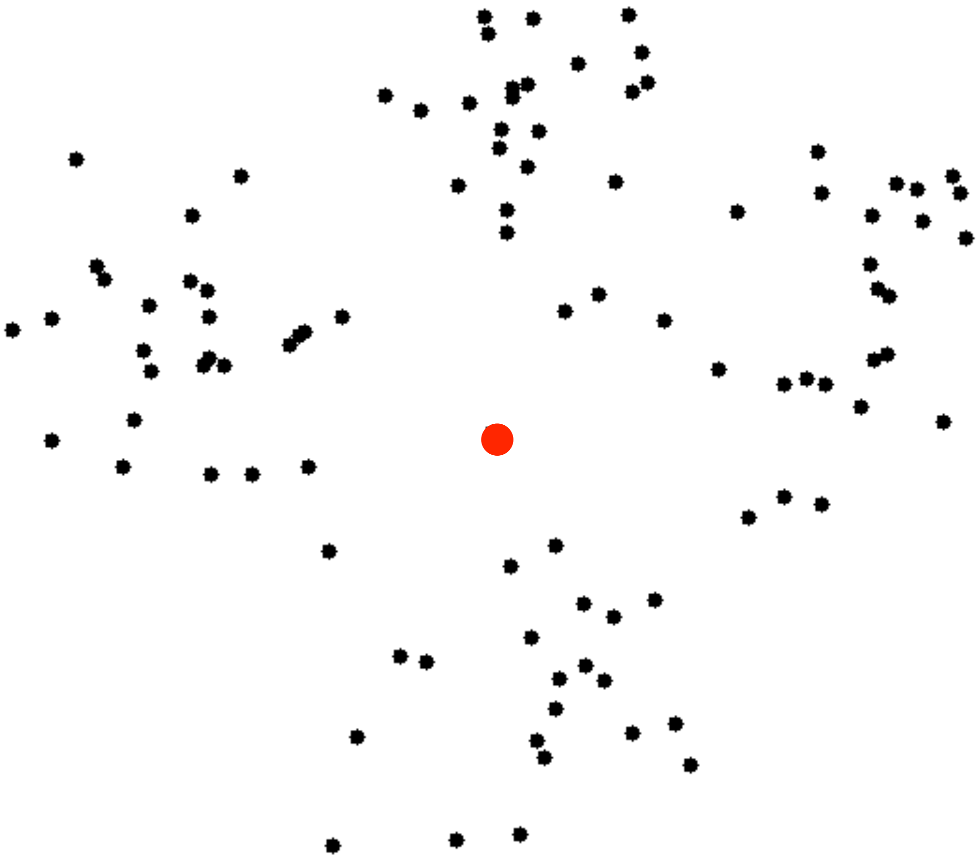
$$SIM(User_1, User_2) = \frac{7\, items}{3\, pairs}(|12-13|^2 + |17-14|^2 + |3.5-4.5|^2)$$

$$= \frac{7}{3}(1+9+1) = 25.66$$

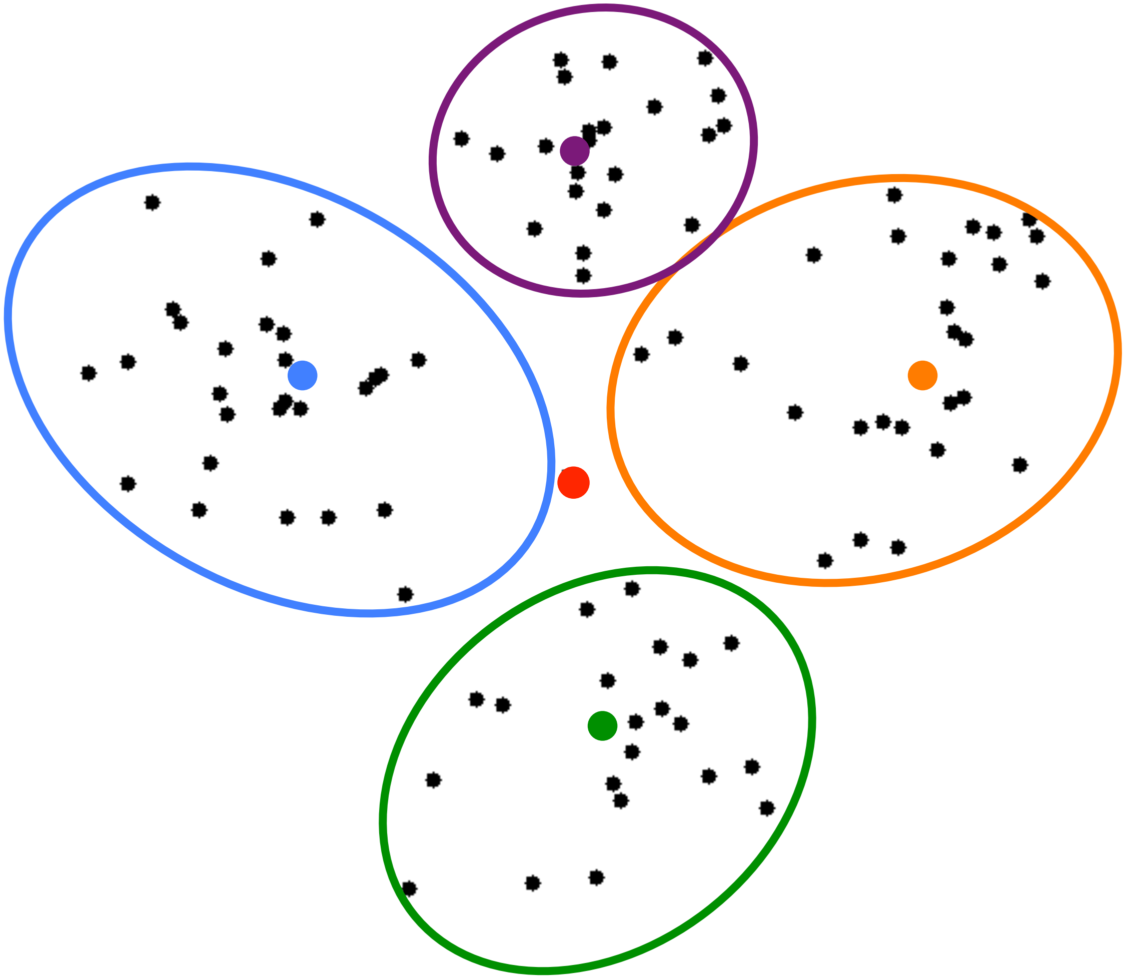
Boxplot – IQR Interpretation



User-user similarity



User-user similarity

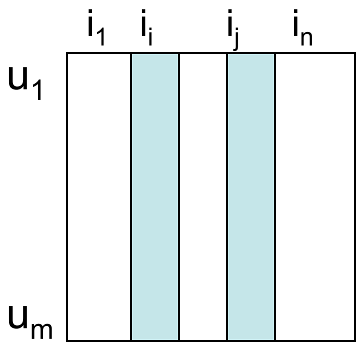


Item based methods: Intuition

- Search for similarities among items
- All computations can be done offline
- Item-Item similarity is more stable that user-user similarity
  - No need for frequent updates

Item Based Methods

- Same as in user-user similarity but on item vectors
  - Find similar items to the one whose rating is missing
  - E.g. For item i_i compute its similarity to each other item i_j



Item based similarity

- Offline phase. For each item
  - Determine its k-most similar items
  - Can use same type of similarity as for user-based
- Online phase:
  - Predict rating r_{aj} for a given user-item pair as a weighted sum over k-most similar items that they rated

$$r_{aj} = \frac{\sum_{i \in \text{k-similar items}} sim(i,j) \times r_{ai}}{\sum_{\in \text{k-similar items}} sim(i,j)}$$

User a	**8**		**r_{aj}**		**9**	**15**

Item j

Items

Users

	Item1	Item2	Item3	Item4	Item5	Item6
User1	17	-	20	18	17	18.5
User2	8	-	????	17	14	17.5
User3	-	-	17	18	18.5	17.5
User4	-	-	-	18	17.5	18
User5	17	-	18	19	15.5	-
User6	-	-	17.5	-	16	-
User7	15	17.5	-	17	-	17
User8	18	-	-	-	17	16.5
User9	18	17	-	-	18.5	17
User10	19	17	-	-	-	16.5
User11	17	18.5	19	19	-	-
User12	14	19	17	-	-	15.5
User13	-	16	-	-	17	-
User14	20	18.5	-	18	-	18

Matrix Based Techniques

- Treat the User-Item Rating table R as a matrix
  - Use matrix factorisation of this Rating Table

Rating Table R

Items

Users

	Item1	Item2	Item3	Item4	Item5	Item6
User1	17	-	20	18	17	18.5
User2	8	-	-	17	14	17.5
User3	-	-	17	18	18.5	17.5
User4	-	-	-	18	17.5	18
User5	17	-	18	19	15.5	-
User6	-	-	17.5	-	16	-
User7	15	17.5	-	17	-	17
User8	18	-	-	-	17	16.5
User9	18	17	-	-	18.5	17
User10	19	17	-	-	-	16.5
User11	17	18.5	19	19	-	-
User12	14	19	17	-	-	15.5
User13	-	16	-	-	17	-
User14	20	18.5	-	18	-	18

Factorisation

- We are familiar with factorisation of numbers

  15=3*5

  99=3*33

  1000=10*100

  We can also do approximate factorisation

  17 ≈ 6*2.8 (RHS= 16.8, an error of 0.2)

  167 ≈ 17*9.8   (RHD=166.6, an error of 0.4)

Matrix Factorization

Given a matrix R, we can find matrices U and V such that when U and V are multiplied together

$$R \approx UV$$

- R is m*n, U is m*k and V is k*n
  - k is the "number of latent factors"

For example, suppose R is a 4*4 matrix

$$R = \begin{bmatrix} 5 & 2 & 3 & 6 \\ 4 & 4 & 6 & 11 \\ 3 & 19 & 2 & 7 \\ 3 & 8.5 & 4 & 2 \end{bmatrix}$$

Example: m=4, n=4, k=2

$$\begin{bmatrix} 5 & 2 & 3 & 6 \\ 4 & 4 & 6 & 11 \\ 3 & 19 & 2 & 7 \\ 3 & 8.5 & 4 & 2 \end{bmatrix} \approx \begin{bmatrix} 0.34776 & 1.97802 \\ 0.71609 & 3.13615 \\ 4.27876 & 0.58287 \\ 1.88074 & 0.56923 \end{bmatrix} \begin{bmatrix} 0.58367 & 4.40189 & 0.44605 & 1.04492 \\ 1.52915 & 0.26346 & 1.75046 & 3.09976 \end{bmatrix}$$

$$= \begin{bmatrix} 3.22769 & 2.05196 & 3.61758 & 6.49480 \\ 5.21363 & 3.97844 & 5.80912 & 10.46959 \\ 3.3887 & 18.98823 & 2.92886 & 6.27777 \\ 1.96819 & 8.42882 & 1.83534 & 3.72973 \end{bmatrix}$$

We can compute the error (squared distance between R and UV). The smaller it is, the better the fit of the factorisation.

$$(5 - 3.22769)^2 + (2 - 2.05196)^2 + (3 - 3.61758)^2 + \ldots$$
$$(4 - 1.83534)^2 + (2 - 3.72973)^2$$

How to factorise

- *Details of how to compute the matrix factorisation are beyond the scope of our study.*
- Intuitively, factorisation algorithms search over lots of choices for U and V, with the aim of making the error as low as possible
- If there are missing values in R, ignore these when computing the error.

Factorisation and missing values

$$\begin{bmatrix} 5 & - & - & 6 \\ - & 4 & 6 & 11 \\ - & 19 & 2 & 7 \\ 3 & 8.5 & - & - \end{bmatrix} \approx \begin{bmatrix} 1.51261 & 1.65457 \\ -0.0474 & 3.56317 \\ 3.88351 & 1.50482 \\ 1.76637 & 0.56005 \end{bmatrix} \begin{bmatrix} 1.07179 & 4.42771 & -0.13516 & 0.60378 \\ 2.01538 & 1.18272 & 1.67926 & 3.08647 \end{bmatrix}$$

$$= \begin{bmatrix} 4.95572 & 8.65430 & 2.57402 & 6.02008 \\ 7.13025 & 4.00394 & 5.98995 & 10.96899 \\ 7.19512 & 18.97488 & 2.00210 & 6.98942 \\ 3.02190 & 8.48338 & 0.70173 & 2.79509 \end{bmatrix}$$

$$\text{Error} = (5 - 4.95572)^2 + (6 - 6.02008)^2 + (4 - 4.00394)^2 + (6 - 5.98995)^2 + \ldots$$

The product of the two factors U and V, has no missing values. We can use this to predict our missing entries. E.g. R_{12}=8.65430

Using k=2 for factorisation

Items

Users

	Item1	Item2	Item3	Item4	Item5	Item6
User1	17	-	20	18	17	18.5
User2	8	-	**13.48**	17	14	17.5
User3	-	-	17	18	18.5	17.5
User4	-	-	-	18	17.5	18
User5	17	-	18	19	15.5	-
User6	-	-	17.5	-	16	-
User7	15	17.5	-	17	-	17
User8	18	-	-	-	17	16.5
User9	18	17	-	-	18.5	17
User10	19	17	-	-	-	16.5
User11	17	18.5	19	19	-	-
User12	14	19	17	-	-	15.5
User13	-	16	-	-	17	-
User14	20	18.5	-	18	-	18

- Real answer for (User 2, Item 3) is 13.5
  - Matrix technique predicts 13.48. Low error for this cell.
- Real answer for (User 13, Item 1) is 17.
  - Matrix technique predicts 15.3. Error is a little higher for this cell.
- In general, the prediction error varies across the cells, but taking all missing cells as a whole, the method aims to make predictions with low average error

Commerical Recommender Systems

- Commercial recommender systems (Netflix, Amazon) use variations of matrix factorisation.
- In 2009, Netflix offered a prize of $USD 1,000,000 in a competition to see which algorithms were most effective for predicting user-movie ratings.
  - Anonymised training data released to public: 100 million ratings by 480k users of 17.8k movies
  - Won by “BellKor’s Pragmatic Chaos” team
- *A followup competition was cancelled due to privacy concerns* … [We will elaborate when we get to topic on privacy]

Other issues

- Many challenging issues in deployment of recommendations
  - Interpretability of recommendations?
  - How to be fair to rare items?
  - How to avoid only recommending popular items?
  - How to handle new users?

References

- See
  - Matrix Factorization Techniques for Recommender Systems. Koren, Bell and Volinsky. IEEE Xplore, Vol 42, 2009. Available on the LMS in Week 3 section.

- Some slides based on “Data Mining Concepts and Techniques”, Han et al, 2nd edition 2006.

COMP20008 Elements of Data Processing

New topic: Visualisation and data clustering

Motivation for visualisation

- Converting data into a visual format
  - Reveals characteristics of the data, relationships between objects or relationships between features
  - Simplifies the data

- Humans are very good at analysing information in a visual format
  - Spot trends, patterns, outliers
  - Visualisation can help show data quality

- Visualisation helps tell a story ….

Visualisations we have already encountered

- Boxplots
  - Median, quartiles, outliers

- Scatter plots
  - Plotting points in 2D or 3D space, using colours to indicate classes/segments

Running Example: Iris Flower Dataset

- Well known dataset introduced by statistican Ronald Fisher with 150 objects
  - https://en.wikipedia.org/wiki/Iris_flower_data_set
- Three flower types (classes):
  - Setosa
  - Virginica
  - Versicolour
- Four features
  - Sepal width and length
  - Petal width and length

Virginica. Robert H. Mohlenbrock. USDA NRCS. 1995. Northeast wetland flora: Field office guide to plant species. Northeast National Technical Center, Chester, PA. Courtesy of USDA NRCS Wetland Science Institute.

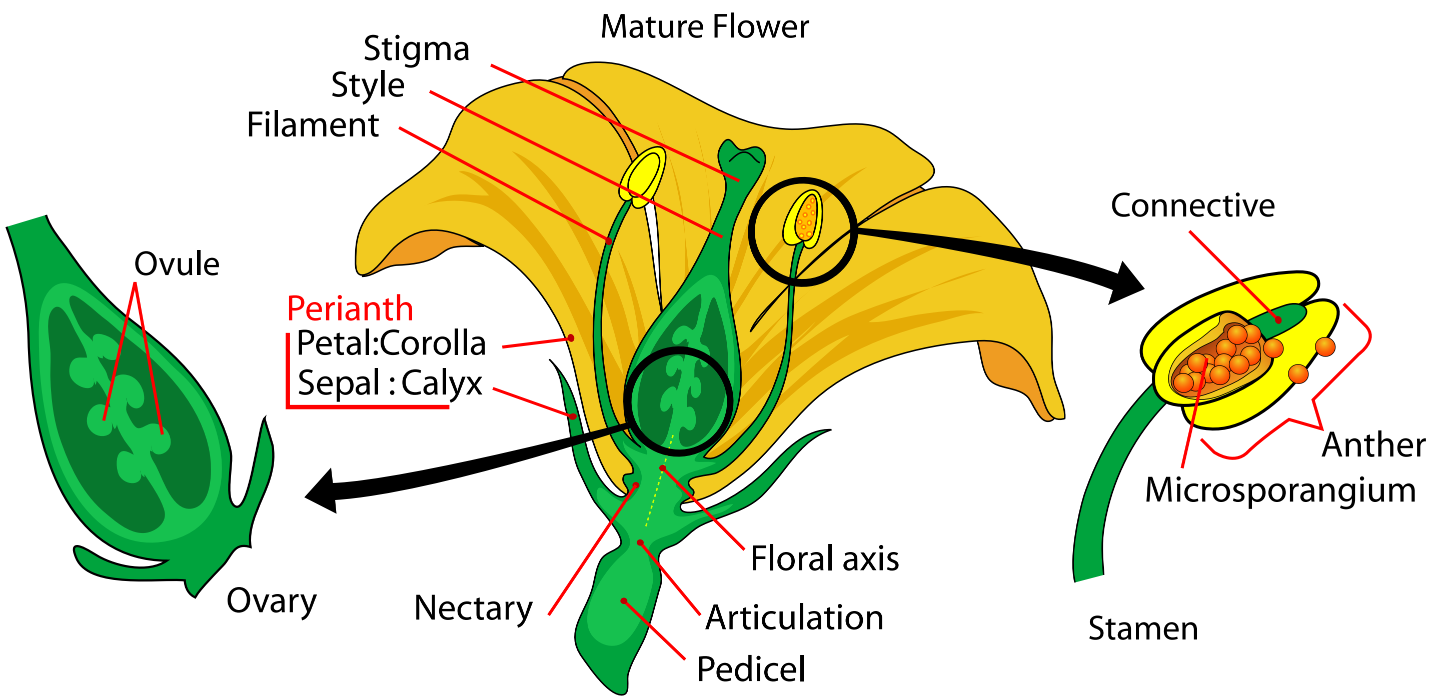
Iris dataset running example

- Extract of Iris data from Wikipedia

Fisher's *Iris* Data

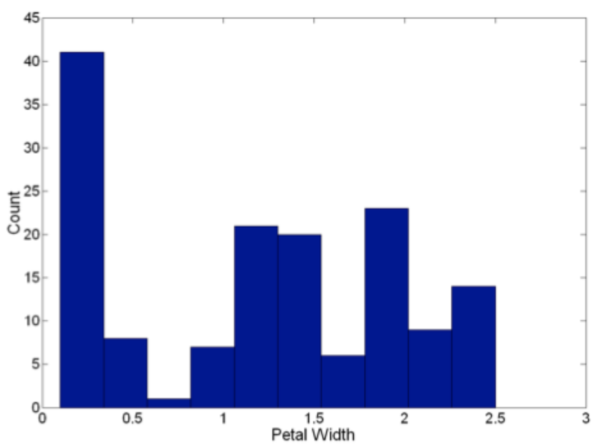
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.2	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
5.0	3.4	1.5	0.2	*I. setosa*

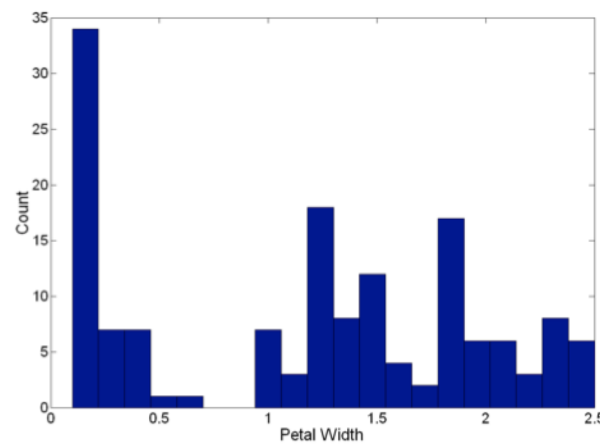
Flower diagram: https://en.wikipedia.org/wiki/Sepal



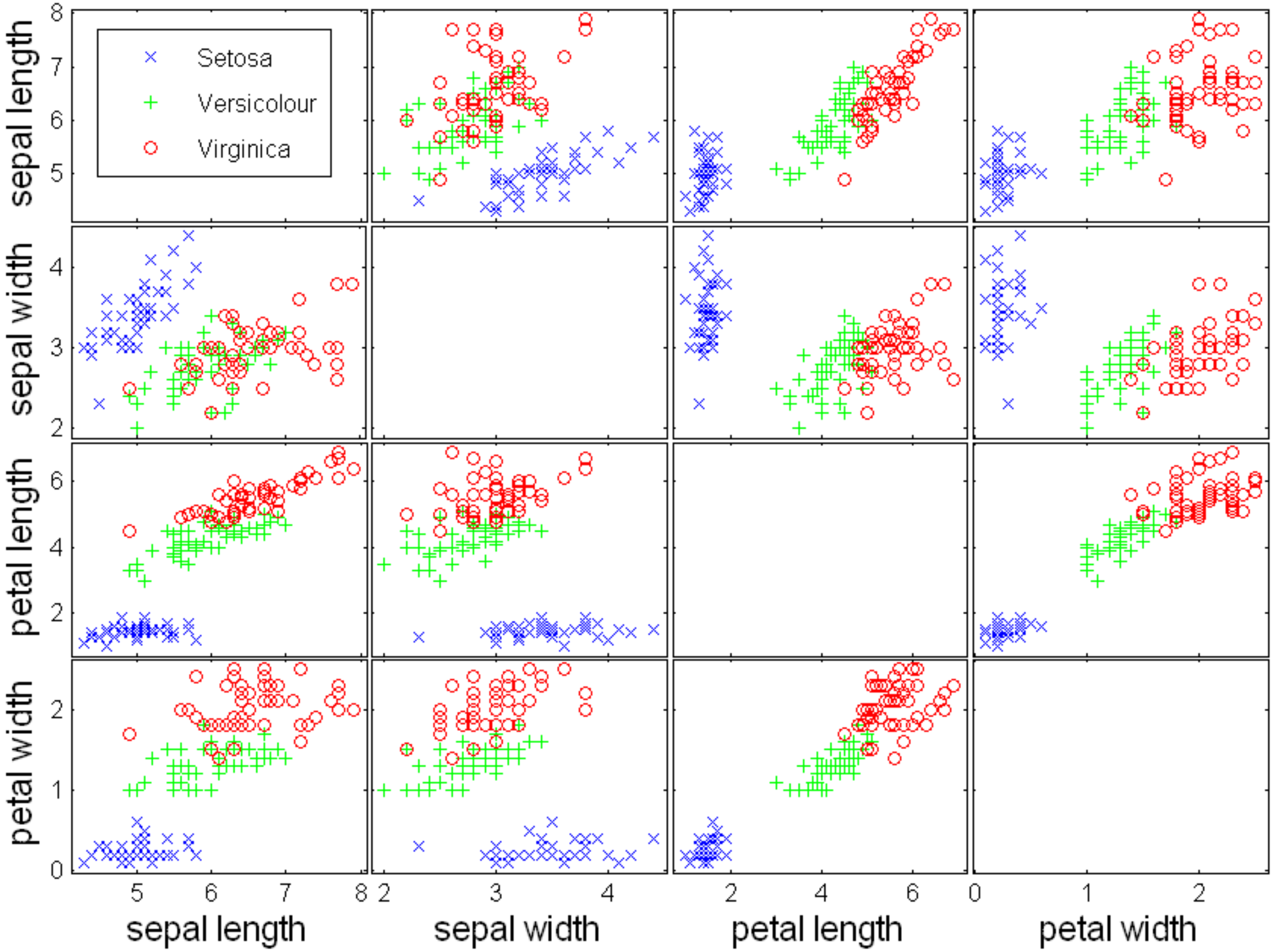
Basic Visualisations: Histograms

- Histogram
  - Usually shows the distribution of values of a single variable
  - Divide the values into bins and show a bar plot of the number of objects in each bin.
  - The height of each bar indicates the number of objects
  - Shape of histogram depends on the number of bins
- Example: Petal Width (10 and 20 bins, respectively)





Basic Visualisations: Scatter plots

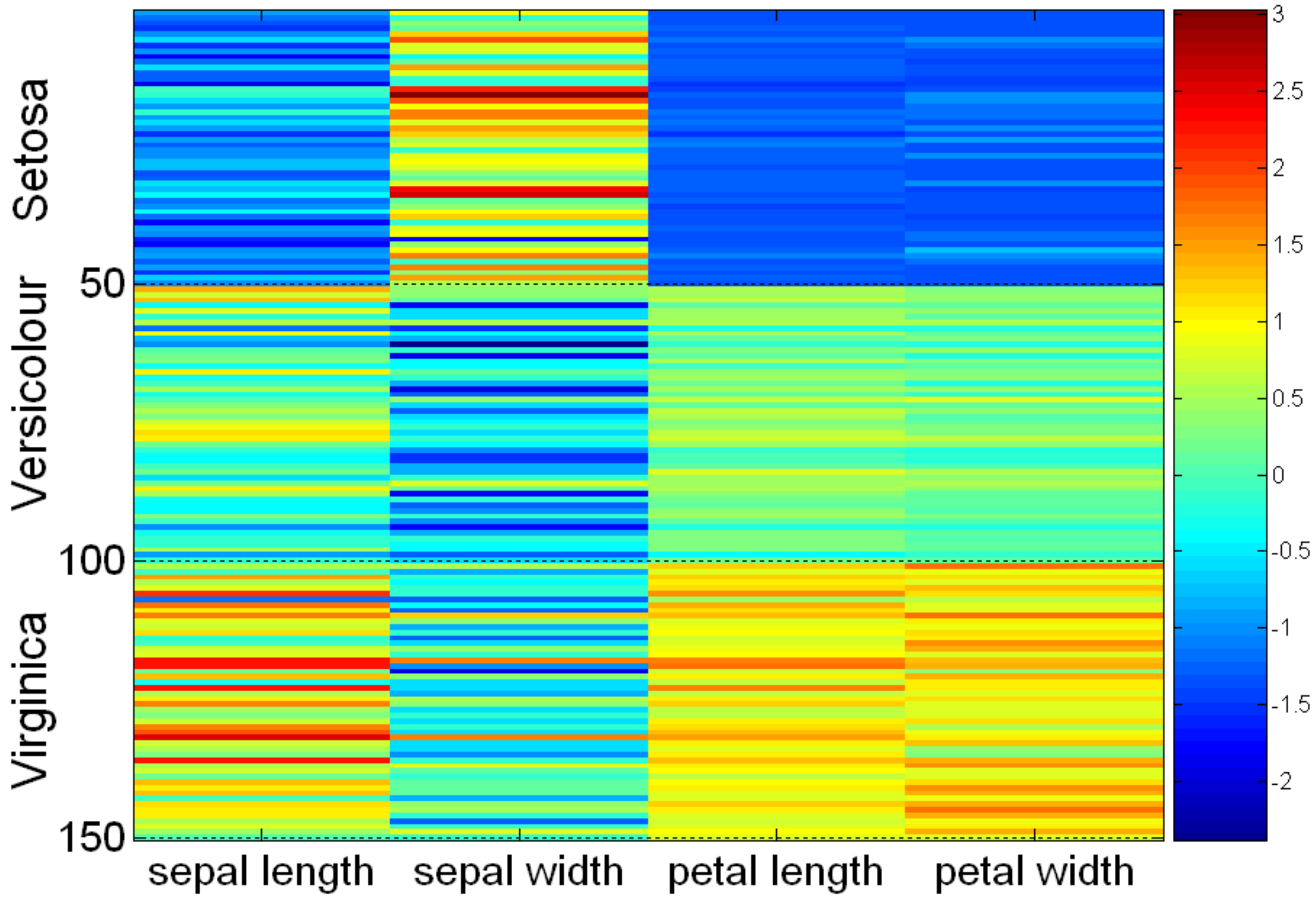


Scatter plots for iris dataset

Heat maps

- Heat maps
  - Plot the data matrix
  - This can be useful when objects are sorted according to class
  - Typically, features are normalized to prevent one attribute from dominating the plot

Visualization of the (normalised) Iris Data Matrix



[Columns have been standardized to have a mean of zero and standard deviation of 1]

Parallel Coordinates

- Parallel Coordinates
  - Used to plot the feature values of high-dimensional data
  - Instead of using perpendicular axes, use a set of parallel axes
  - The feature values of each object are plotted as a point on each corresponding coordinate axis and the points are connected by a line
  - Thus, each object is represented as a line
  - Often, the lines representing a distinct class of objects group together, at least for some features
  - Ordering of attributes is important in seeing such groupings

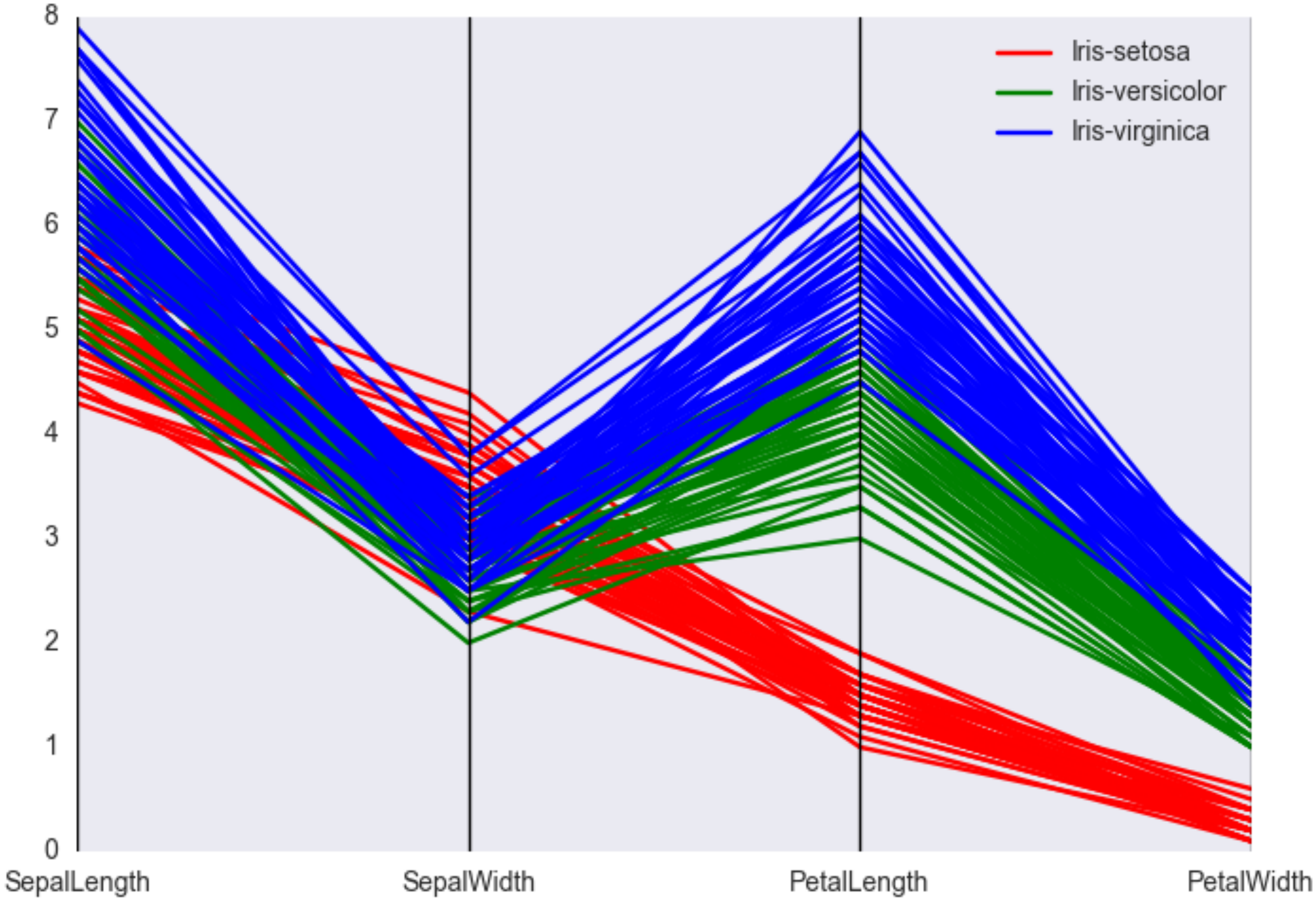
Iris dataset

- Extract of Iris data from Wikipedia

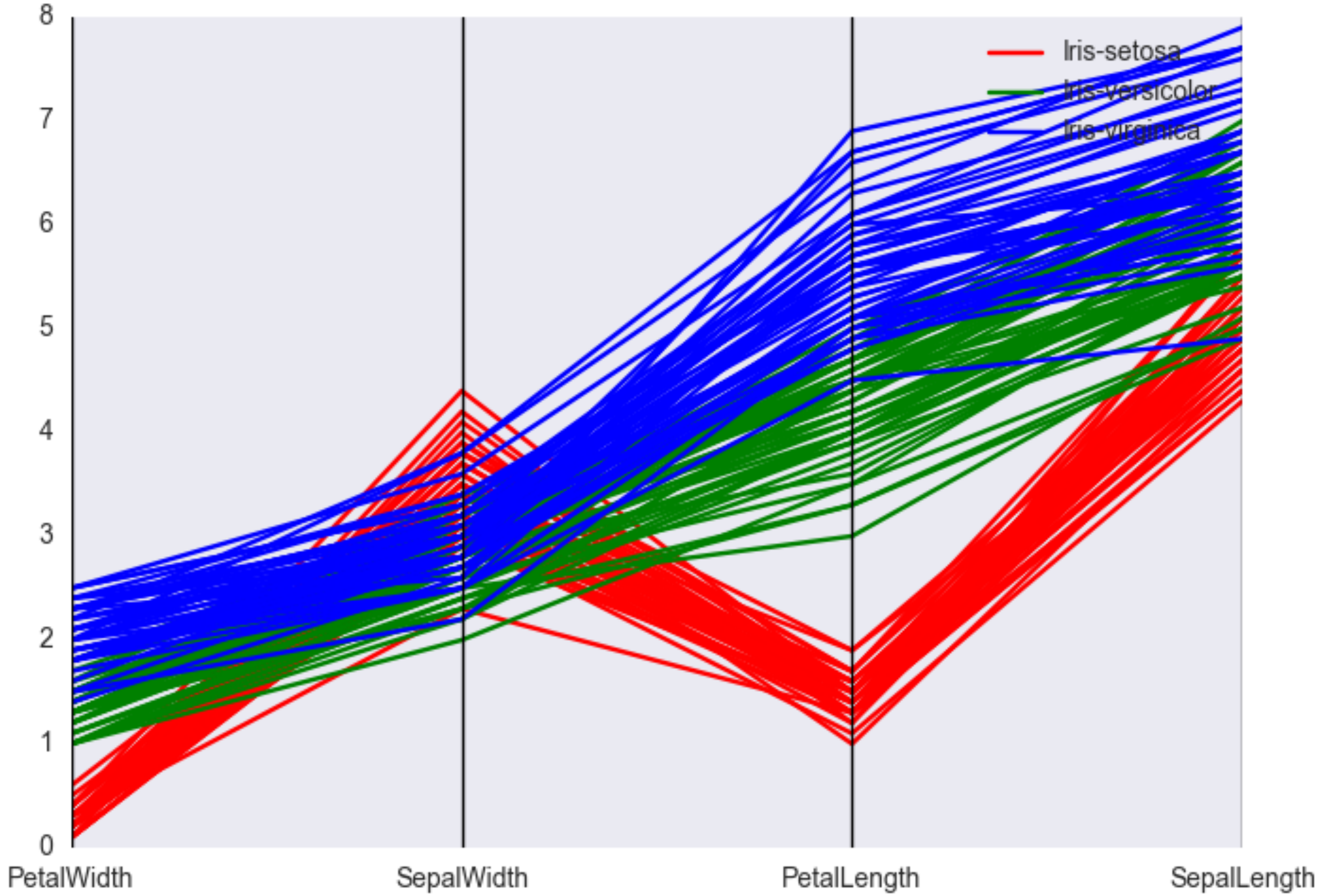
Fisher's *Iris* Data

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.2	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
5.0	3.4	1.5	0.2	*I. setosa*

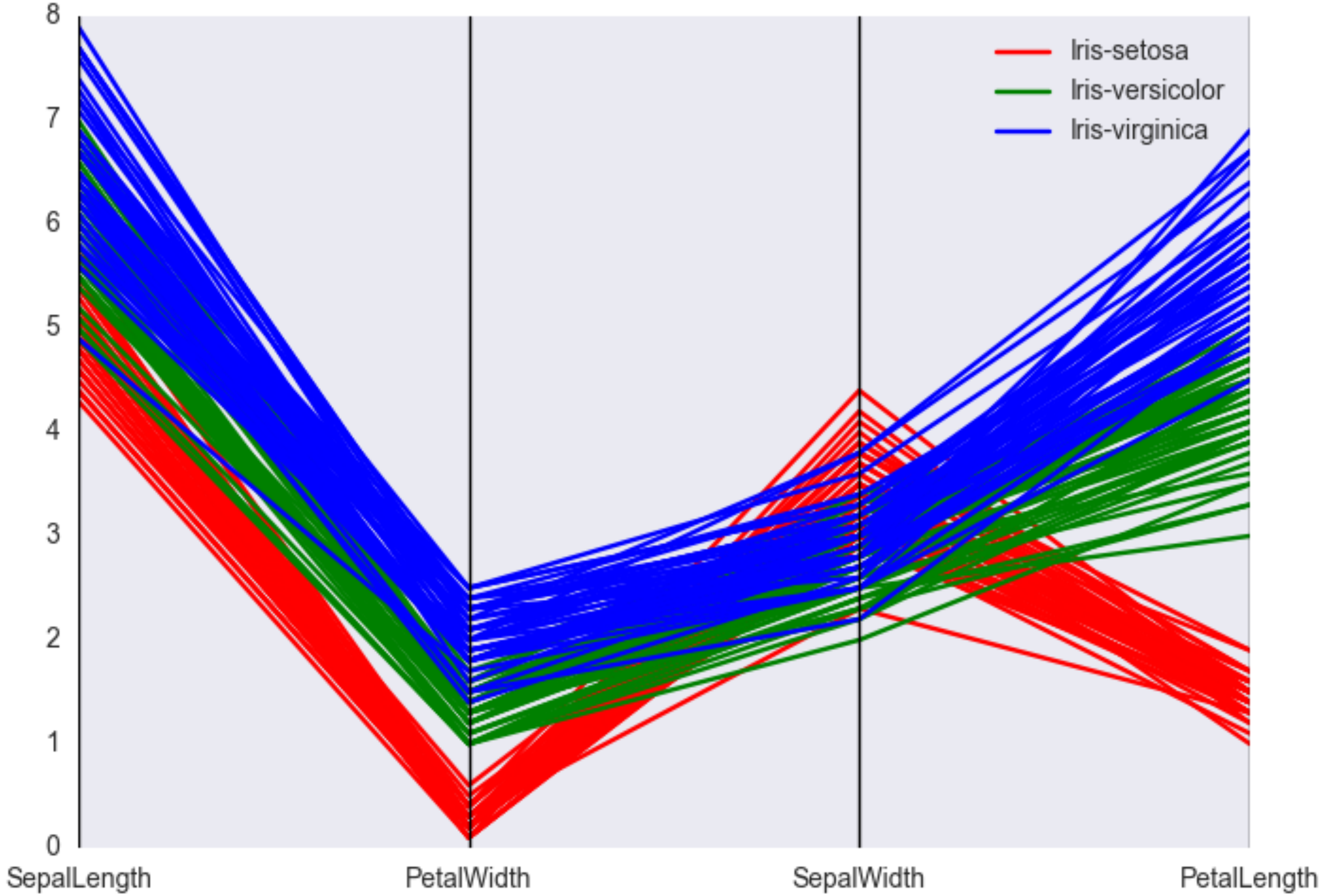
Parallel coordinates: ordering 1



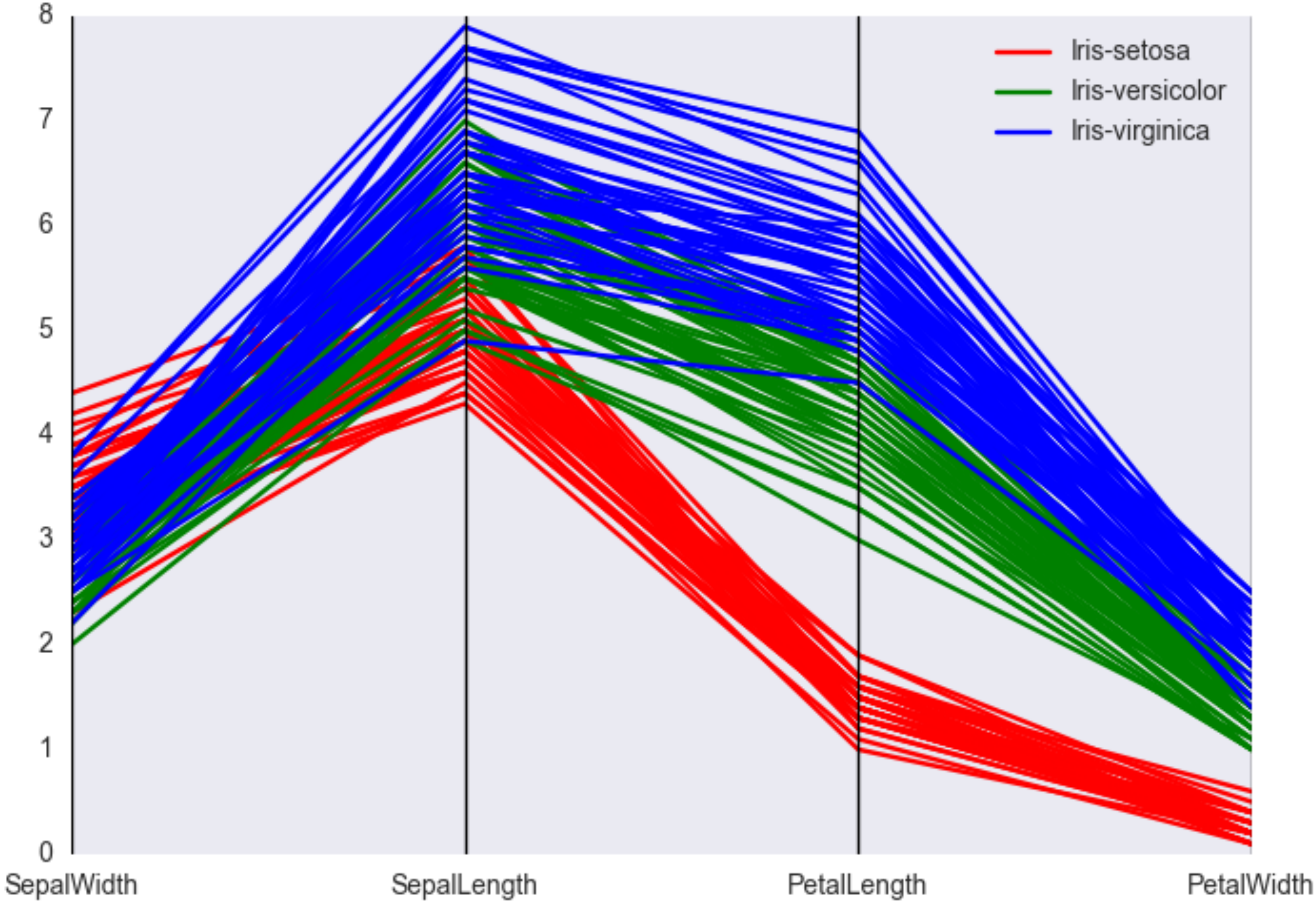
Parallel coordinates: ordering 2



Parallel coordinates: ordering 3



Parallel coordinates: ordering 4



Key issues in using parallel coordinates

- Scaling axes
  - Affects the visualisation. May choose to scale all features into the range [0,1] via a pre-processing step

- Ordering of axes
  - Influences the relationships that can be seen. Correlations between pairs of features may only be visible in certain orderings

Parallel co-ordinates code

- Python code
  - *parallel_coordinates* in *pandas.tools.plotting*
  - Will practice in workshop

Acknowledgements

- Material partly adapted from
  - “Data Mining Concepts and Techniques”, Han et al, 2nd edition 2006.
  - “Introduction to Data Mining”, Tan et al 2005.