COMP4030

DATA MODELLING AND ANALYSIS

Lecture 7: Data Pre-processing II

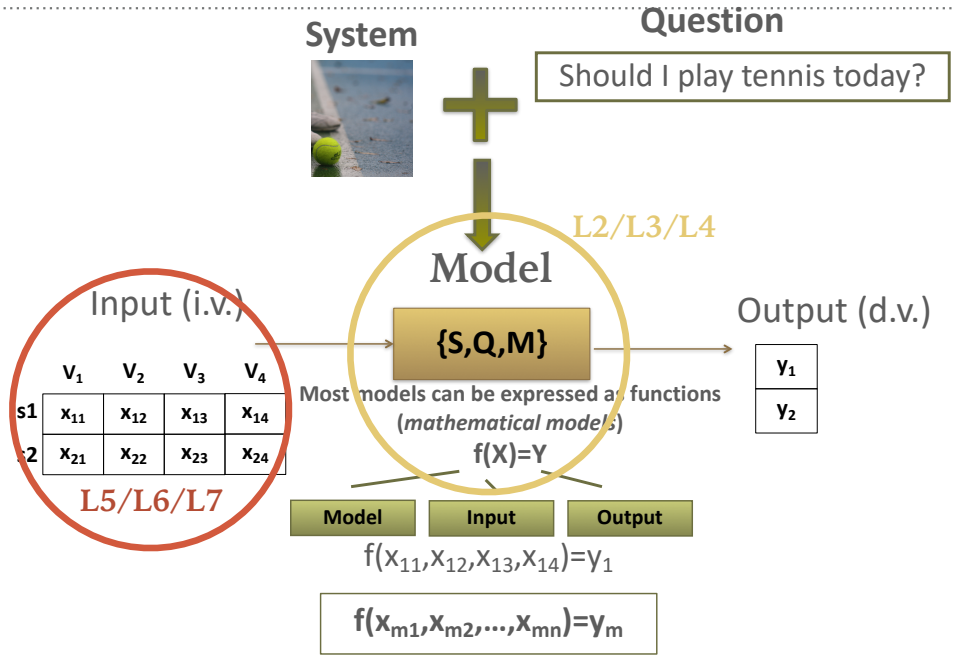
LECTURE OUTLINE

1. Lecture Outcomes
2. Previously in DMA
3. Reducing data
    1. Instance Reduction
        1. Deletion
        2. Sampling
    2. Attribute Reduction
        1. Deletion
        2. Attribute Selection
        3. Principal Component Analysis (for Transformation and Reduction

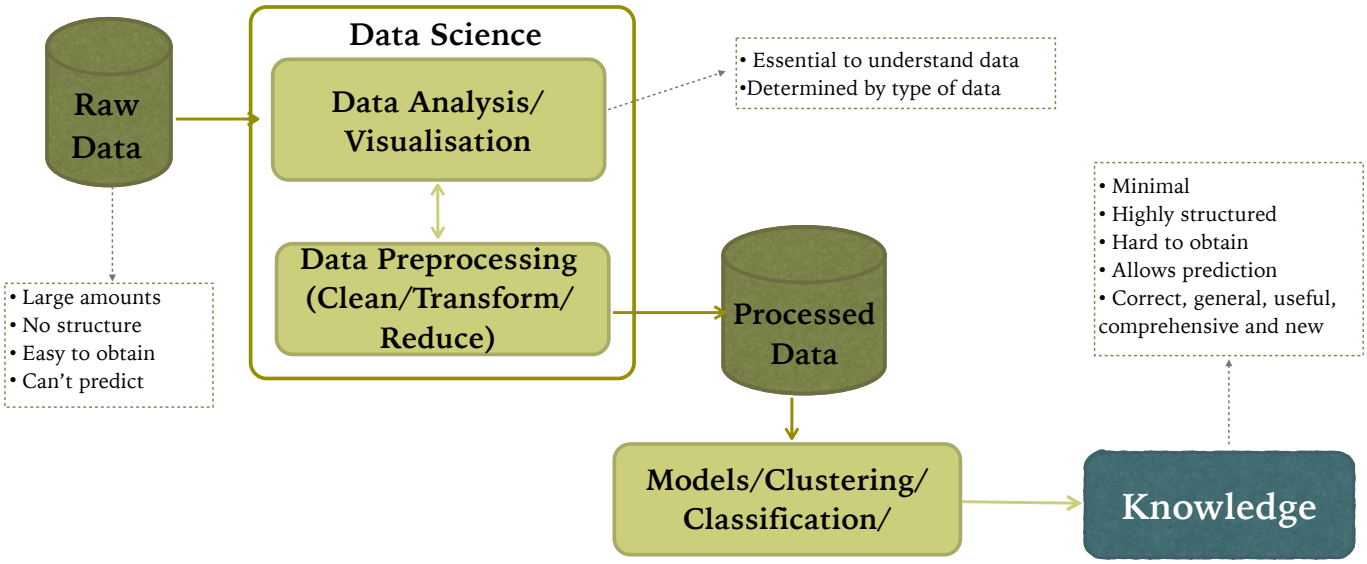
LECTURE OUTCOMES

- At the end of this lecture, you should be able to answer these questions:
  - What is dimensionality reduction?
  - What is Principal Component Analysis (PCA)?
  - How is PCA calculated? How is it calculated in R?
  - How can PCA be used to transform and to reduce your dataset?
  - What is attribute selection?

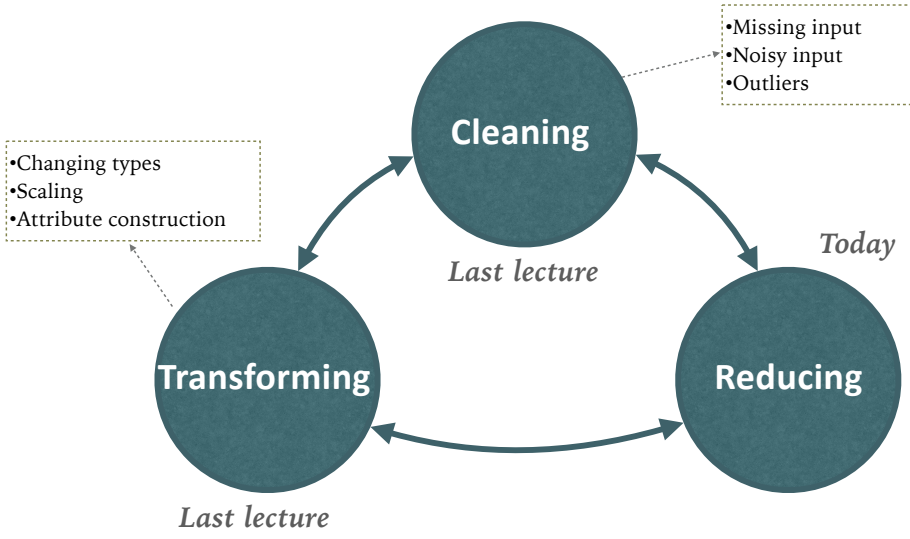
DMA: THE BIG PICTURE



PREVIOUSLY IN DMA: DATA VS KNOWLEDGE



THE PRE-PROCESSING CYCLE



WHY DO WE NEED DATA REDUCTION?

- Large no. of variables: difficult to study and interpret
- Too many pairwise correlations between the variables to consider.
- Graphical display of data: not helpful when data set is very large.
  - 12 variables: more than 200 three-dimensional scatterplots to be studied!
- Reducing data: helps to **interpret data in a more meaningful form**.
- Helps alleviate The Curse of Dimensionality

REDUCING DATA: THE CURSE OF DIMENSIONALITY

- When using high-dimensional data (hundreds/thousands dimensions)
- As dimensions increase: space volume increases so fast that the available data become sparse.
- Problematic for methods requiring statistical significance.
- To obtain a statistically reliable result: data needed often grows exponentially with the dimensionality.
- Distance functions lose their usefulness
- Searching data relies on detecting areas where objects groups:
  - Instances appear sparse and dissimilar in many ways.

REDUCING DATA: HOW DO WE DO DATA REDUCTION?

- We are going to see different methods:
  1. Instance Reduction
     1. Sampling instances
     2. Balancing instances
  2. Attribute Reduction
     1. Deleting attributes
     2. Attribute Selection
     3. Principal Component Analysis (PCA)

REDUCING DATA: REDUCING INSTANCES

- Check for missing values and/or duplicate rows
  - delete? CARE (duplicate rows may be meaningful)!
- Dealing with duplicate rows in R:
  - duplicated(c(1,2,3,1)) #FALSE FALSE FALSE TRUE
  - df = data.frame(name=c("John","Peter","John","Peter"), age=c(23,23,21,30))
  - duplicated(df$name) #FALSE FALSE TRUE TRUE
  - duplicated(df$age) #FALSE TRUE FALSE FALSE
  - df1 = df[!duplicated(df$name),]
  - df2 = df[!duplicated(df$age),]

REDUCING DATA: SAMPLING INSTANCES

- Random sampling
  - fixed number of instances, or fixed proportion
    - first *x* instances
      - may cause problems, e.g. if data sorted by key attribute
    - 10%: every 10th instance
      - may cause problems, e.g. time ordered & contiguous required
- Stratified sampling
  - split the data into chunks
    - use all in model creation and combine the models later
    - keep separate as training, testing & validation sets
  - split the data according to a specific attribute
    - e.g. split men / women in census data?

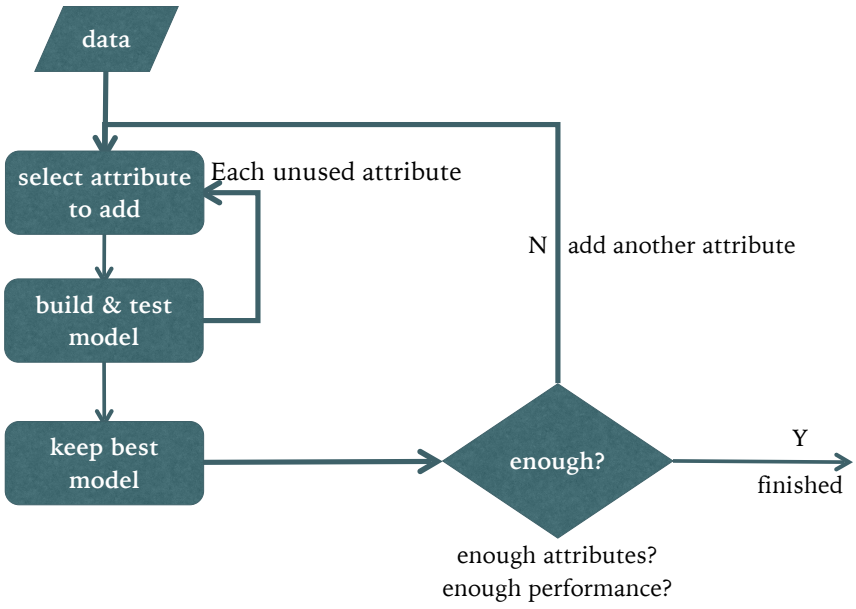
REDUCING DATA: BALANCING INSTANCES

- Balanced classes
  - a set of data with a *balanced* class is one in which the distribution of class values is uniform
    - Adult census data: *High* and *Low* earners
      - High: 25%; Low: 75% ⇒ *unbalanced*
- Many machine learning algorithms require a balanced data set to operate 'optimally'
  - extremely unbalanced (e.g. High: 5%; Low: 95%) means that a prediction of Low (for everything) achieves 95%
    - hard to beat the *default classifier*
- Balanced attributes
  - the same principles may apply to attributes, e.g. gender
- Undersampling (oversampling)

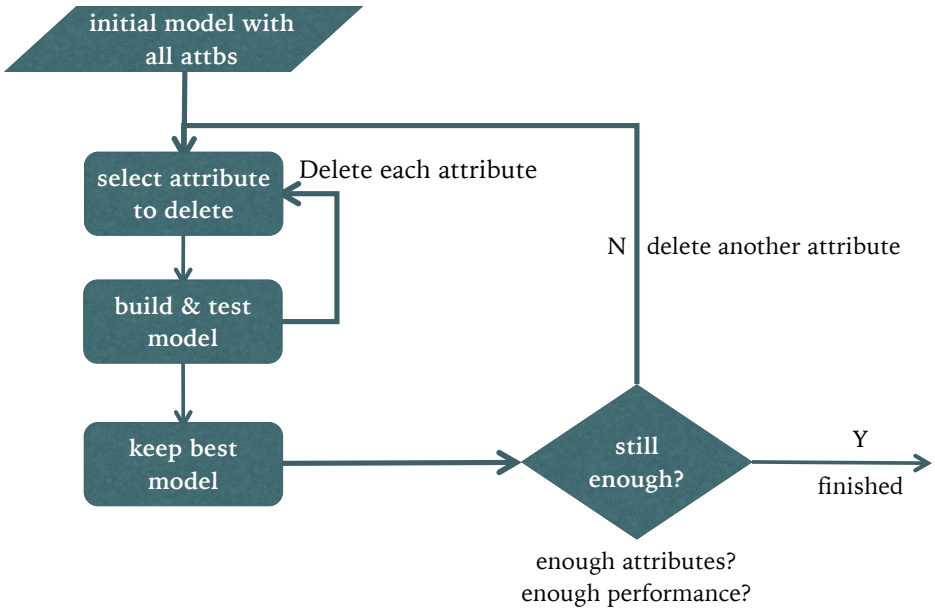
REDUCING DATA: REDUCING ATTRIBUTES

- Check for missing values
  - delete? CARE!
- Check for duplicate columns in R:
  - it is uncommon to have completely identical columns
  - if two columns are suspected identical, use
    - `a==b`, and filter the column to look for `FALSE`
- A much harder problem is columns that contain very similar information
  - numerical: use pair-wise correlations
  - categorical: use filtering
    - use pair-wise chi-squared (advanced!)

REDUCING DATA: ATTRIBUTE FORWARD SELECTION

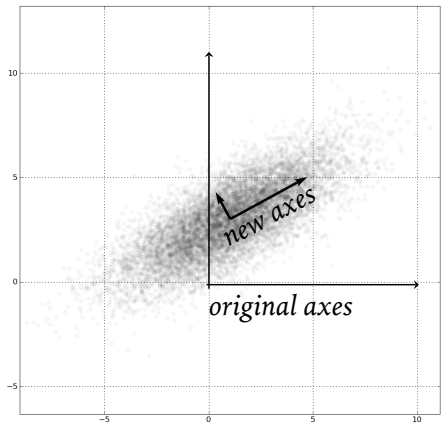


REDUCING DATA: ATTRIBUTE BACKWARDS SELECTION



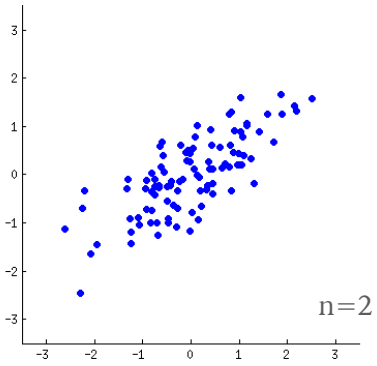
REDUCING DATA: PCA – OVERVIEW

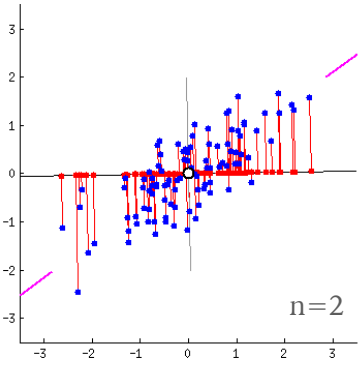
- PCA can be used for data transformation and data reduction.
- Let's first consider the case of **data transformation**
- Consider an n-dimensional data set
- Find the first axis, along which the data (after projection onto that axis) has the largest variance.
- Find the next axis (perpendicular to the first), that maximises the variance of the data.
- Continue in this way until all n axes have been found



n=2

REDUCING DATA: PCA – VISUAL OVERVIEW





- Red dots are projections of the blue dots to each of the new pair of axes.
- The “spread” of the red points is linked to the variance. We are looking for “maximal spread”.
- Maximum variance is found when large axis matches the magenta line.

REDUCING DATA: PCA – PROPERTIES

- PCA decorrelates the data.
- Prior to PCA: columns of the data matrix may have some correlation with each other
- PCA provides a linear transformation of the data:
  - after which, the columns of the data are uncorrelated with one another
  - In a new orthogonal space

REDUCING DATA: PCA – PROPERTIES

- BUT
  - attributes lose their original ‘meaning’
- **Important:** for PCA to completely uncorrelate the data, data must be standardized.

REDUCING DATA: PCA – MANUAL CALCULATION

- Manual calculation:
  1. Manual PCA: involves calculating the **eigenvectors and eigenvalues** of the **covariance matrix** of the **standardised version of the original matrix**.
  2. Steps: Given A, a *mxn* matrix:
  1. Compute the mean μ
  2. Build the standard matrix B, and compute covariance matrix S.
  3. Find the eigenvalues $\lambda 1, \ldots, \lambda m$ of S (decreasing order), and orthogonal set of eigenvectors $u1, \ldots, um$

REDUCING DATA: PCA – USING R

- Using R:
  - **p <- prcomp(standard_database,scale=T,...)**
  - p <- princomp(standard_database,...)

REDUCING DATA: PCA – CHARACTERISTICS

- PCA is, first of all, a linear transformation:
  - n attributes → n principal components (PCs)
  - each one of the *n* PCs is a **linear combination** of all the original attributes.
  - Useful for visualisation:
    - plot points as scatter of PC1 v PC2
- **But**, it can also be used for data reduction:
  - Projects original data into PCs

PCA VISUALISATION

- We can visualise PCs to identify relationships within the data.
- PCs=2: only the first two PCs.
- Finds the 2D plane through the high-dimensional dataset in which data **is most spread out.**
- If clusters: these too may be most spread out.
  - Most visible to be plotted out in a two-dimensional diagram;
- If two of the original variables are chosen at random:
  - Clusters might overlay each other, making them indistinguishable.

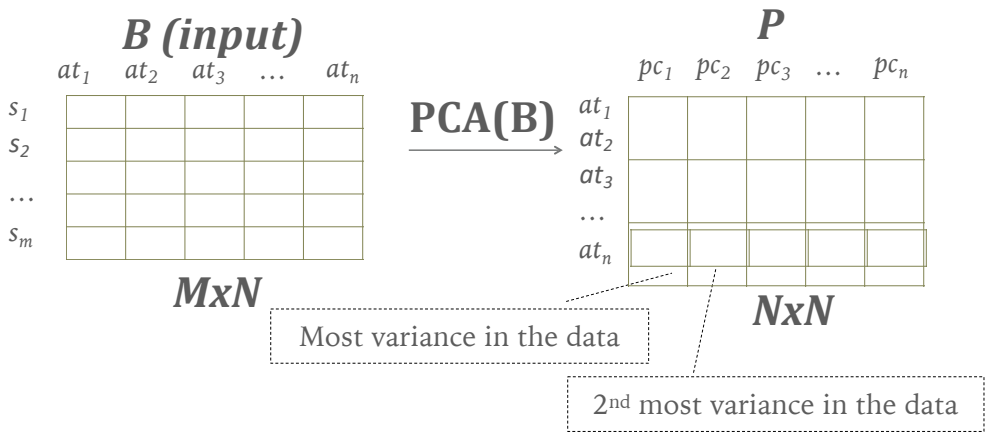
PCA TO TRANSFORM (DECORRELATE) DIMENSIONS

- PCA: For a matrix of *m* x *n* dimensions:
  - n attributes → n principal components (PCs)
  - PCs are ordered
    - PC1: PC containing most variation (most important!?)
    - PC2: next most variation
    - …
    - PCN: least variaton
    - CARE: most variation ≠ most informative attribute
- So, we can simply **project** original data into new space given **by all of the of PCs.**
  - this is common strategy in high-dimensional numeric data sets (i.e. numeric data with many columns.)

REDUCING DATA: PCA TO REDUCE DIMENSIONS

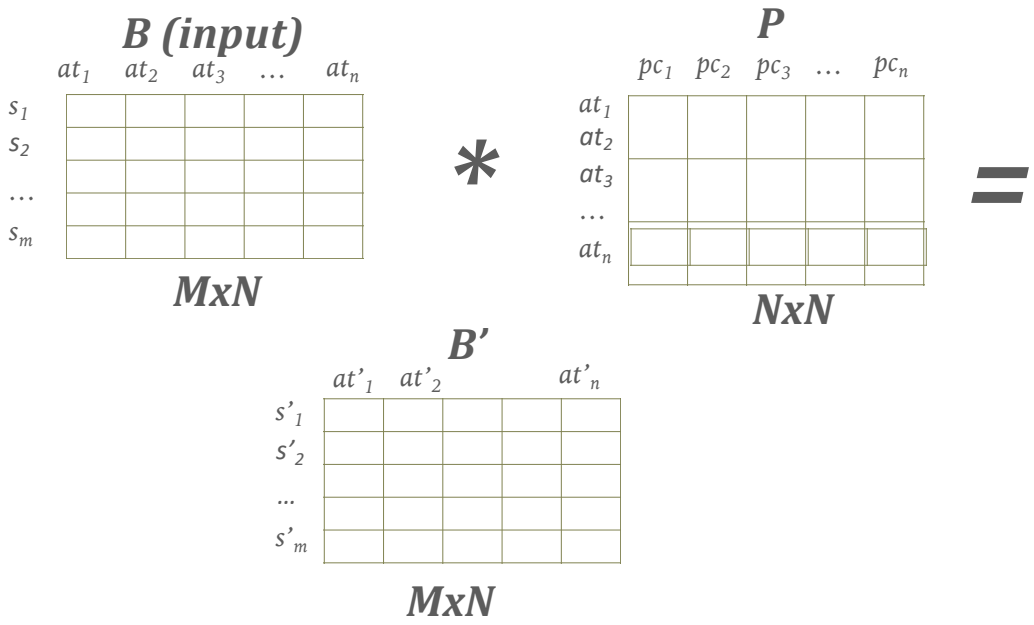
- A: a *mxn* matrix, the original matrix (your database)
- B: a *mxn* matrix, obtained by standardising A.
- P: a *nxn* matrix, the result of PCA(B)
  - P= prcomp(B,scale=T)
  - Chose as many PCs as you need.
  - The % of variance will tell you how many PCs to choose.
  - In our case, for example: 2.
  - reduced_P= P[,1:2], a *n*x2 matrix
- Reduce dimensions in B by projecting it into P
  - C= B*reduced_P
- C is a *m*x2 matrix which can be used as input for models.

PCA TO TRANSFORM (DECORRELATE) DIMENSIONS



You need to choose how many PCs are necessary for the transformed data to be representative enough. Normally: 95% to 99% cumulative variance

PCA TO TRANSFORM (DECORRELATE) DIMENSIONS



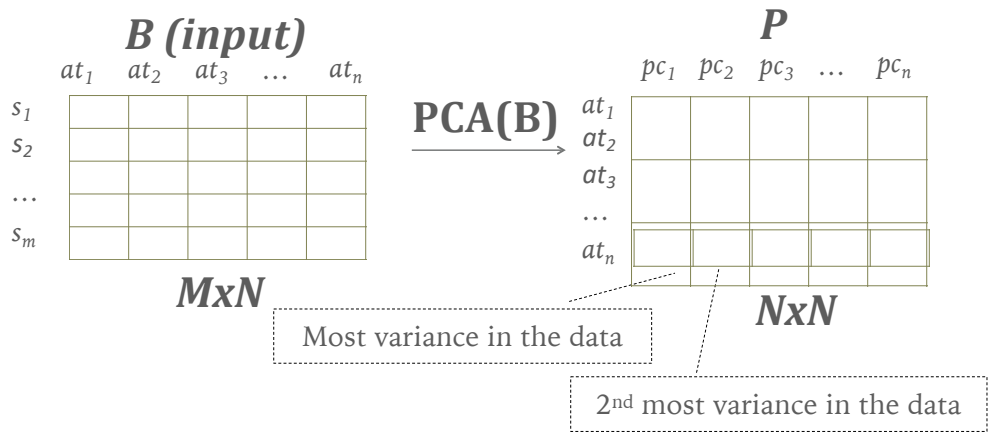
REDUCING DATA: PCA TO REDUCE DIMENSIONS

- PCA: For a matrix of *m* x *n* dimensions:
  - n attributes → n principal components (PCs)
  - PCs are ordered
    - PC1: PC containing most variation (most important!?)
    - PC2: next most variation
    - CARE: most variation ≠ most informative attribute
- So, we can simply **project** original data into new space given **by a selection of PCs**.
  - this is common strategy in high-dimensional numeric data sets (i.e. numeric data with many columns.)

REDUCING DATA: PCA TO REDUCE DIMENSIONS

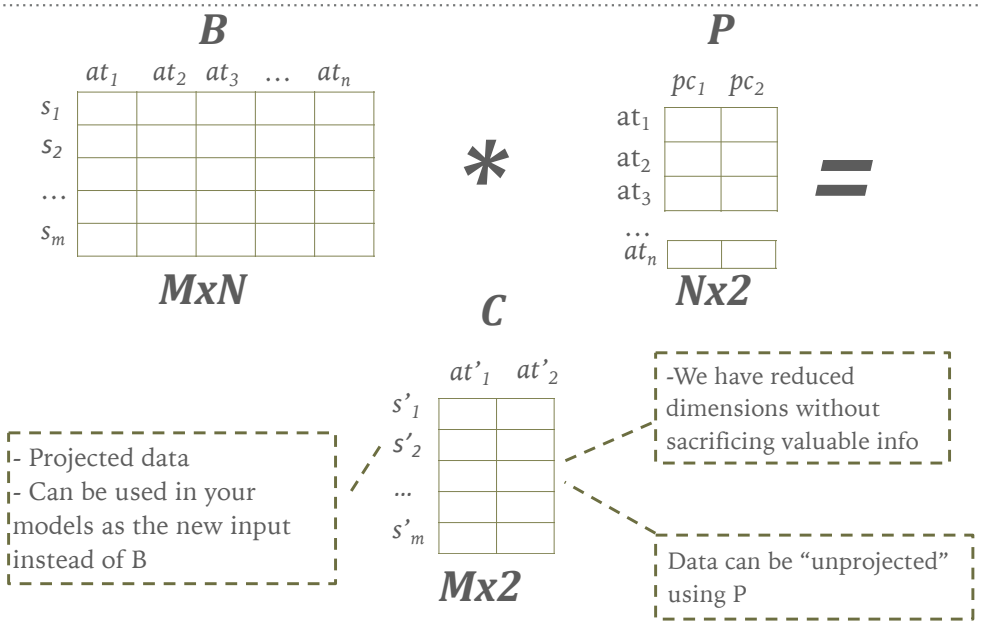
- A: a *mxn* matrix, the original matrix (your database)
- B: a *mxn* matrix, obtained by standardising A.
- P: a *nxn* matrix, the result of PCA(B)
  - P= prcomp(B,scale=T)
  - Chose as many PCs as you need.
  - The % of variance will tell you how many PCs to choose.
  - In our case, for example: 2.
  - reduced_P= P[,1:2], a *n*x2 matrix
- Reduce dimensions in B by projecting it into P
  - C= B*reduced_P
- C is a *m*x2 matrix which can be used as input for models.

REDUCING DATA: PCA TO REDUCE DIMENSIONS



You need to choose how many PCs are necessary for the transformed data to be representative enough. Normally: 95% to 99% cumulative variance

REDUCING DATA: PCA TO REDUCE DIMENSIONS – PROJECT DATA



ADDITIONAL RESOURCES

- If you are interested in the algebra behind PCA, you might want to look at:
  - Pattern Recognition and Machine Learning (Bishop, 2006).
  - Principal component analysis with linear algebra (Jauregui, 2012):http://www.math.union.edu/~jaureguj/PCA.pdf

REDUCING DATA: OTHER METHODS

- There are many other methods for Data Reduction (a.k.a. Dimensionality Reduction):
  - High Correlation Filter
  - Linear Discriminant Analysis
  - Independent Component Analysis
  - Kernel Principal Component Analysis
  - t-Distributed Stochastic Neighbour Embedding (t-SNE)
  - …

DEMO TIME

LECTURE OUTCOMES

- Now, you should be able to answer these questions:
  - What is dimensionality reduction?
  - What is PCA?
  - How is PCA calculated? How is it calculated in R?
  - How can PCA be used to transform and to reduce your dataset?
  - What is attribute selection?
  - How can it be done?

THE END

Questions