Aalto University
School of Electrical
Engineering

AGF-351:
Optical methods in auroral physics

DATA ANALYSIS: PATTERN RECOGNITION ET AL.

Mikko Syrjäsuo

School of Electrical Engineering

Aalto University

(email: firstname.lastname@aalto.fi)

Why am I lecturing these things to you?

- Worked on spaceflight instrumentation until 1995

- Worked on all-sky cameras at the Finnish Meteorological Institute (1995-2002)
  - MIRACLE network's optical imaging instruments
  - Doctoral degree in Computer Science
    - Computer vision, pattern recognition
- Worked on all-sky *imagers* at the University of Calgary (2002-2007)
  - Auroral image data processing
  - THEMIS and NORSTAR optical imaging instruments

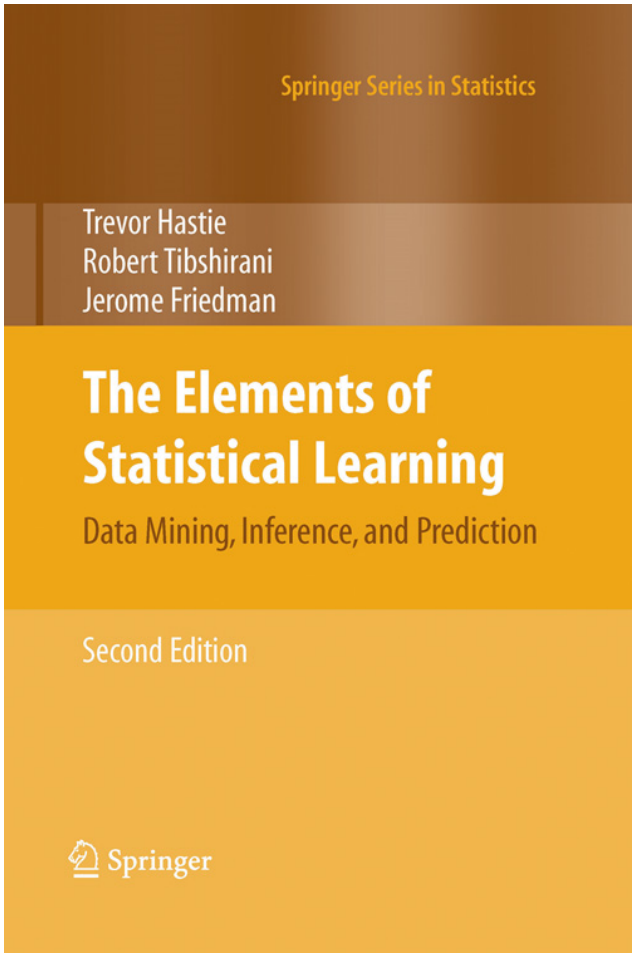
- Worked on spaceflight instrumentation (2007-2011)
- Technology Manager at Aalto University's School of Electrical Engineering since September, 2011

An important note on terminology!!

- The terminology that will be used is that of computer science and statistical analysis
  - This differs from what is used in physics, be warned
- For example, “model” refers to a mathematical model
  - In physics, you would call this a “fit”
- If something sounds weird, stop me and ask for explanations!!

If you don't have this book, get it…

- Hastie, Tibshirani & Friedmann: Elements of Statistical Learning, Springer, 2nd ed., 2009
  - A comprehensive collection of the "standard" methods
  - Fairly multidisciplinary and reading all previous chapters is usually not required to be able to use the tools
  - Almost all demonstrated methods are available in R and additional ones can be found on the associated website



Available for download, too…

http://www-stat.stanford.edu/~tibs/ElemStatLearn/

A slightly philosophical introduction

"We are actually not interested in numbers,
we are interested in what they represent."

(For proper conduction of scientific observations, we **need** to understand optical components, CCD-cameras and image acquisition etc. as they change the "numbers"…)

A less philosophical introduction

We are trying to understand how the universe behaves by observing natural phenomena such as aurora.

In physics, the observations lead to hypotheses whose correctness is then evaluated using data (from experiments).

Learning is accomplished by recognising patterns in data.

In computer science, we learn from data by using mathematical methods. We also evaluate our learning success by error analysis.

Topics

- **Overview**
  - Pattern recognition, data mining, computer vision
- **Terminology and tools**
  - Features, feature space and similarity metrics
  - Supervised and unsupervised classification
  - K-means clustering, KNN classification
- **Curse of dimensionality**
  - Dimensionality reduction

- **Pattern recognition with auroral images**
  - Ground-based and satellite imaging

What is a pattern?

- “An arrangement or sequence regularly found in comparable objects or events”
- “A regular and intelligible form or sequence discernible in certain actions or situations”

Examples:

- Human face
  - Eyes, nose, mouth etc.
- What patterns are there in aurora?

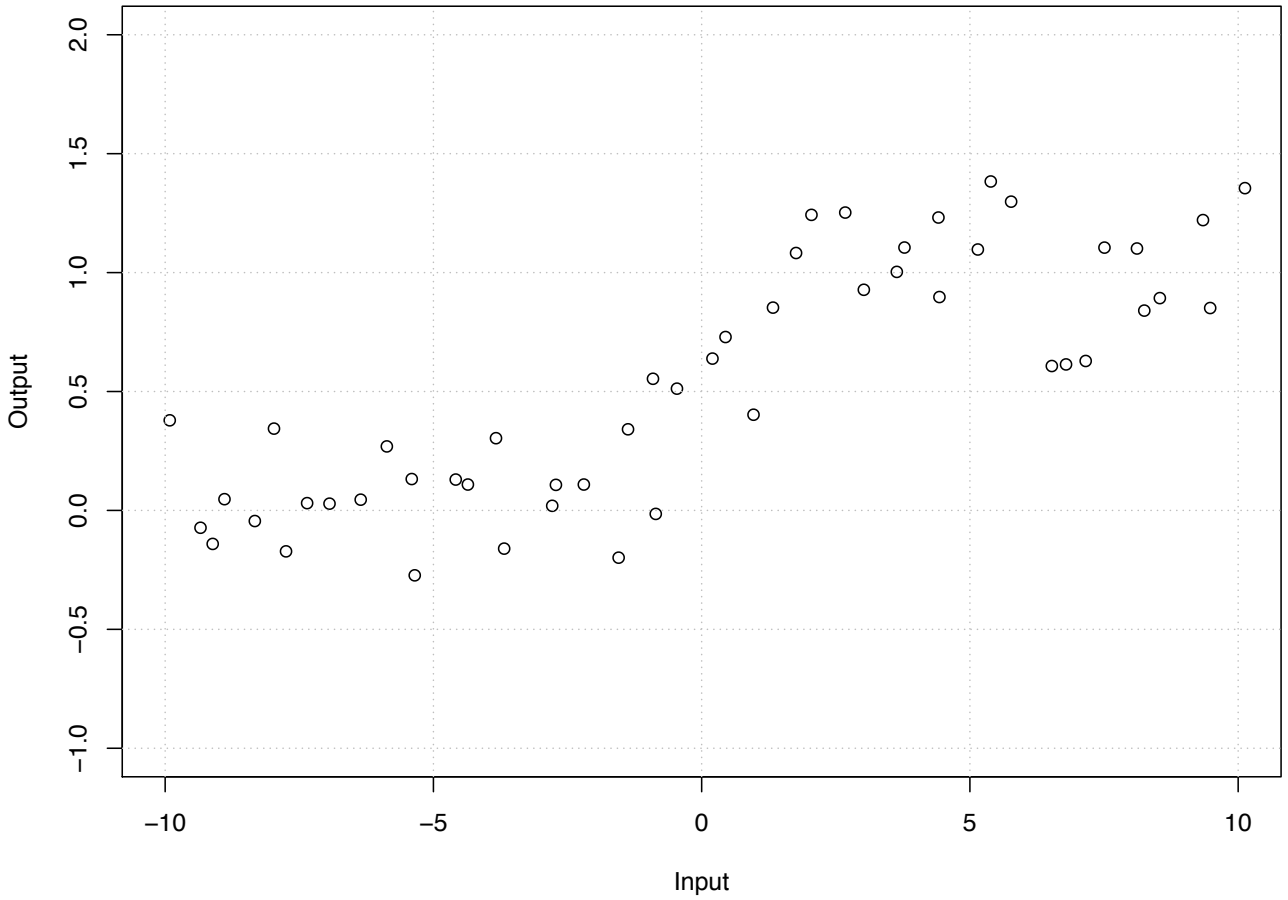
What is pattern recognition?

- Computer science → artificial intelligence → machine learning → pattern recognition
- Assignment of (some sort of) output value to (some sort of) input value
  - Deterministic: uses a specific algorithm
  - Predictor/learner: uses data to learn the assignments

Two main types:

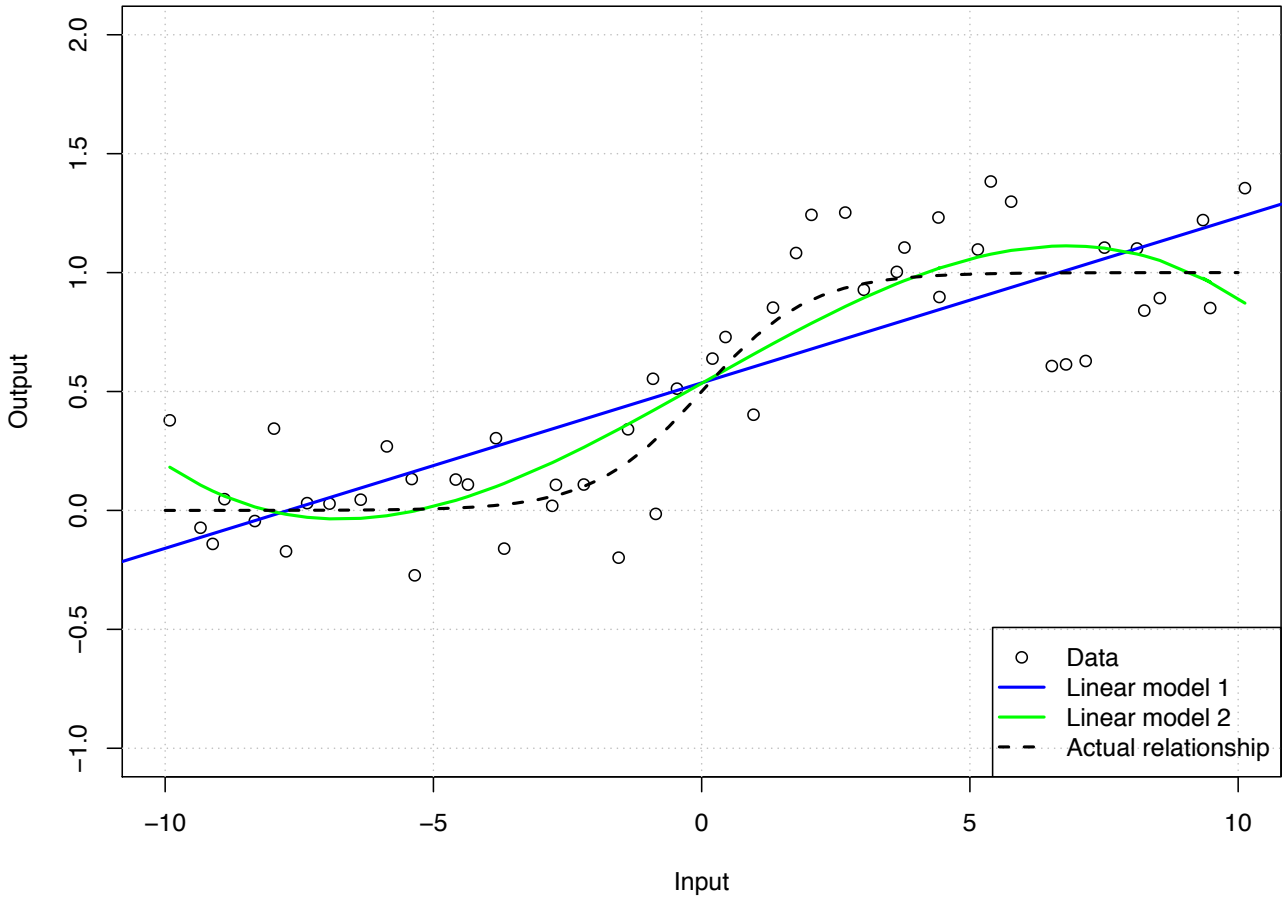
- Regression (output is quantitative)
- Classification (output is qualitative)

Regression — fitting a model to the data



In here, the objective (and the difficulty) is to recognise the pattern in data and choose a correct model…

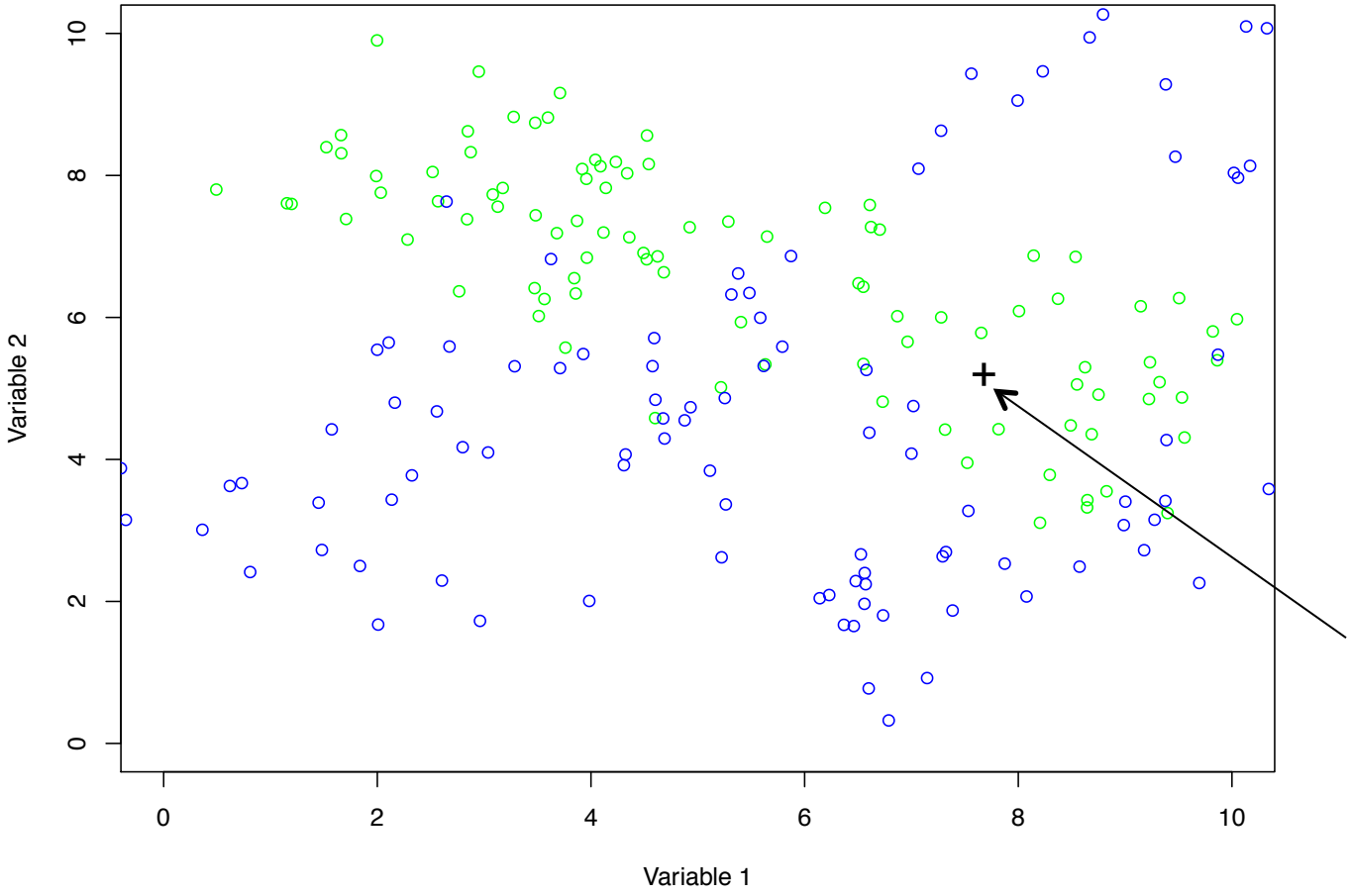
Regression — fitting a model to the data



Two linear models (polynomial fits) have been selected.

The relationship of the input and output values is actually different. Especially after adding some “measurement noise”

Classification — discrete labels



Depending on the values of two variables, each data point belongs into one of two different classes (GREEN/ BLUE)

What is the class of this new observation?

Data mining, computer vision etc.

- Pattern recognition is a fundamental tool used in many modern applications

- Data mining, knowledge discovery
  - Extraction of patterns is used in exploring the data
    - Clustering, automatic labelling of data
- Computer vision
  - Patterns are used in e.g. recognising and tracking objects in images
    - Two eyes, nose & mouth (in a certain arrangement) → human face

Observations and variables — examples

$$\begin{matrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{matrix}$$

- In auroral all-sky images:
  - Each image is an observation
  - Each pixel is a variable
- In time-series,
  - Each time point is one observation
  - Whatever you measured at that time point are the variables
    - E.g. temperature, pressure, humidity

Dimensionality = number of variables

Learning from data: data types

- Qualitative — nominal, categorical, discrete, factors
  - An unordered finite set of values
    - E.g. Ω={hammer, wrench, axe}
    - Typically represented numerically with "dummy variables"
      - Hammer ↔ [1,0,0]
      - Wrench ↔ [0,1,0]
      - Axe ↔ [0,0,1]
- Quantitative — ordinal, numeric, real-valued
  - An ordered set of (discrete) values
    - E.g. integer numbers, A/D-conversion results

Curse of dimensionality

- Reliable analysis requires a sufficient number of samples (observations)

- Sampling density is proportional to $N^{1/p}$, where p is the dimension of the input space and N the number of observations

Example: assume that for a 1-D data we have concluded that 20 observations are sufficient and we want the same accuracy in higher dimensions, too. For 2-D data, we need 400 observations etc.

Dimension	Required number of observations
1	20
2	$20^2 = 400$
3	$20^3 = 8{,}000$
10	$20^{10} \approx 10^{13}$

Remember that in star calibration, you wanted to have stars evenly distributed over the whole field-of-view. This is what is meant with "sampling density"

Features — compact representation

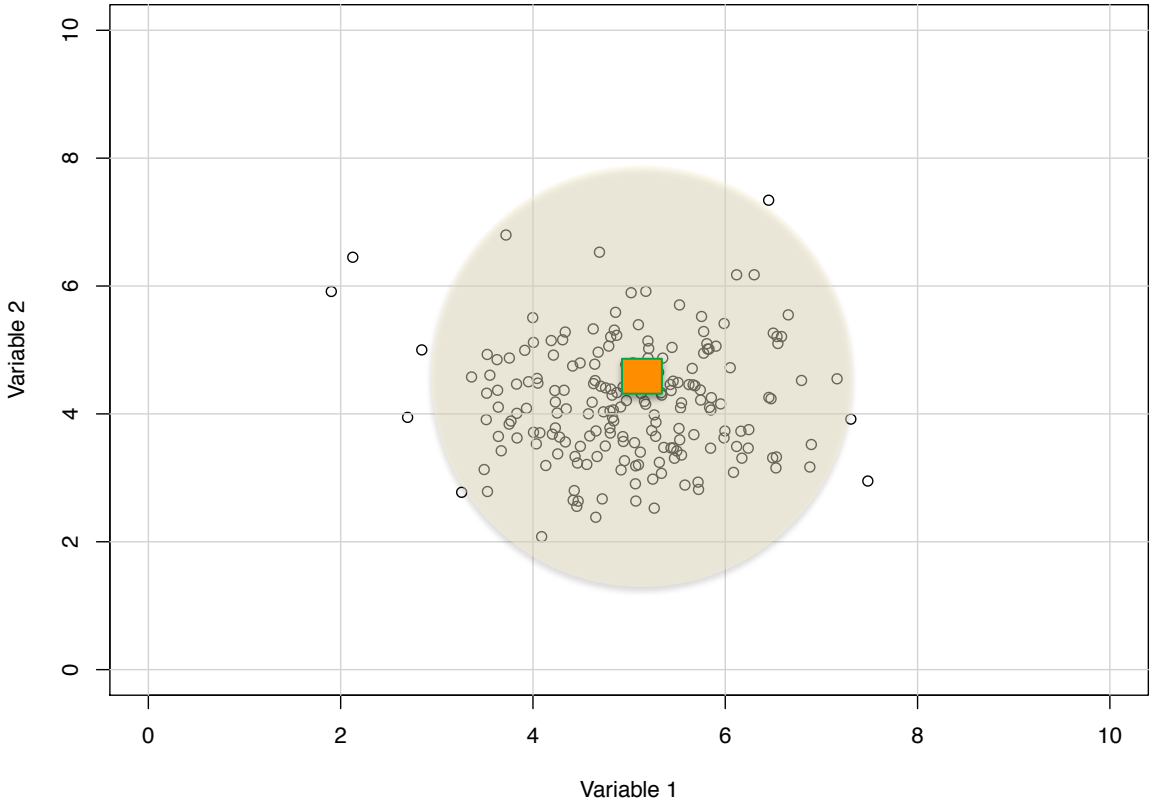
The dimensionality of a 512×512 auroral image is 262144 ☺

- Numeric features are extracted from data to facilitate analysis
  - Obtaining compact numeric representations of complex data
  - In some cases, the variables can be the feature (e.g. magnetic field components as a 3-element vector)
- **APPLICATION DEPENDENT!!!**
  - Computer scientists still looking for a perfect general-use feature...
  - Domain expertise can help if you already know what you are looking for

An example of a compact feature — normal distribution parameters

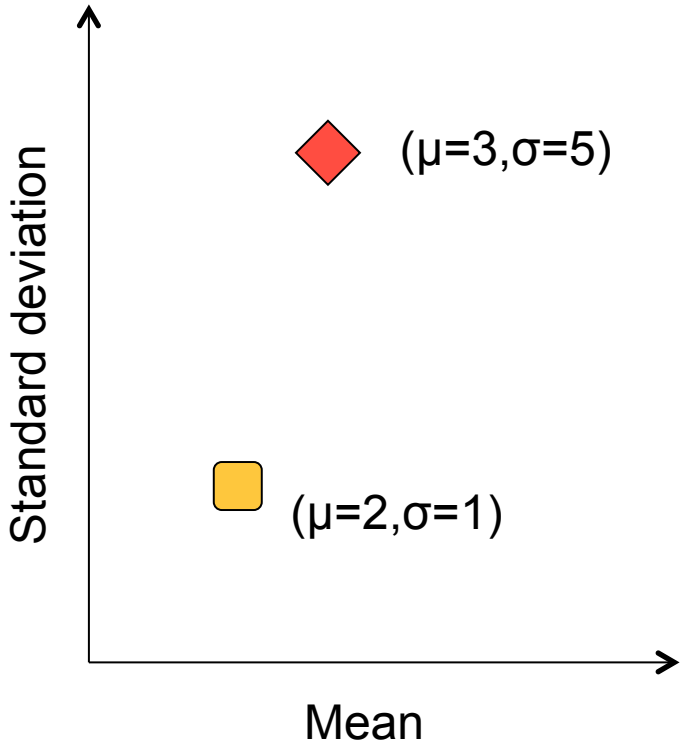
All data points can be described as random samples from a population.

Rather than providing a large number of observations, we just give the mean and standard deviation.



The observations are points in the feature space

- The features form the basis of the feature space
- Following the previous example, the space is two-dimensional
  - One axis is the mean value
  - The other axis is the standard deviation
- Each observation is a point in this feature space

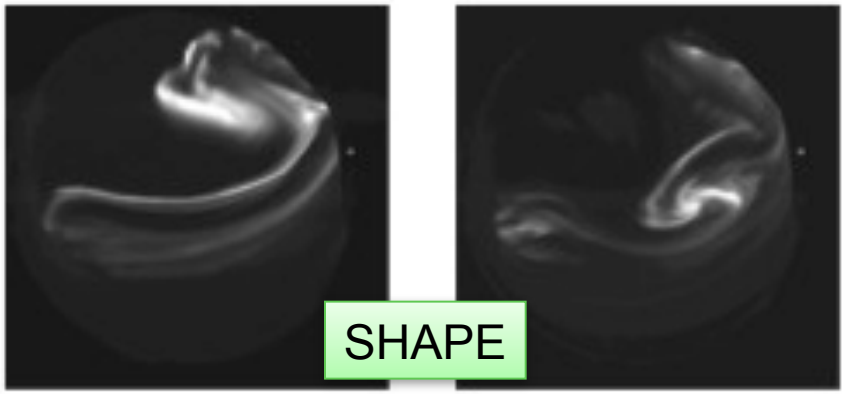


The semantic gap — low and high-level features

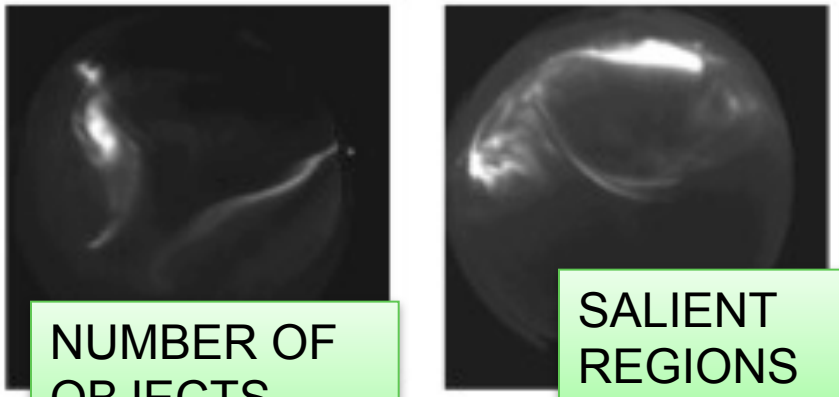
- The human concepts need to be transformed into something we can process with computers
  - One of the grand challenges in Computer Science…
  - How do you quantify similarity? This knowledge is required for any computation requiring ordering etc.

- Low-level (primitive) features
  - Colour, texture, shape, location (e.g. “bright”, “green”)
- Medium-level features/semantics
  - Objects/targets identified by low-level features (e.g. “auroral oval”)
- High-level semantics
  - Abstract concepts (e.g. “substorm”)

Possible features in an all-sky image



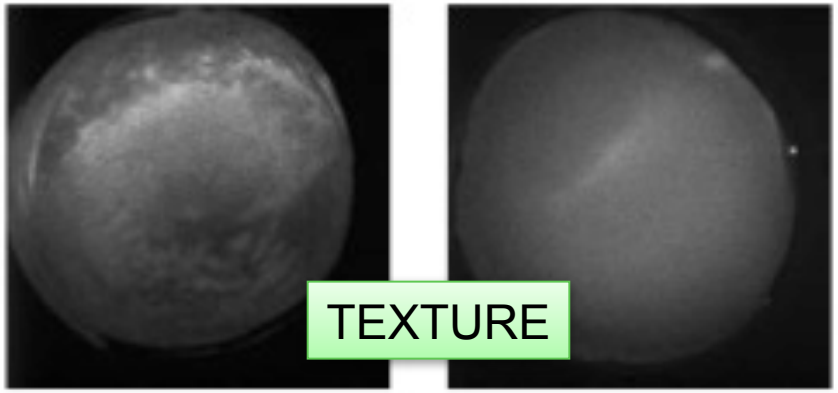
SHAPE



NUMBER OF OBJECTS

SALIENT REGIONS

INTENSITY VALUES



TEXTURE

META-DATA

Preprocessing (1):

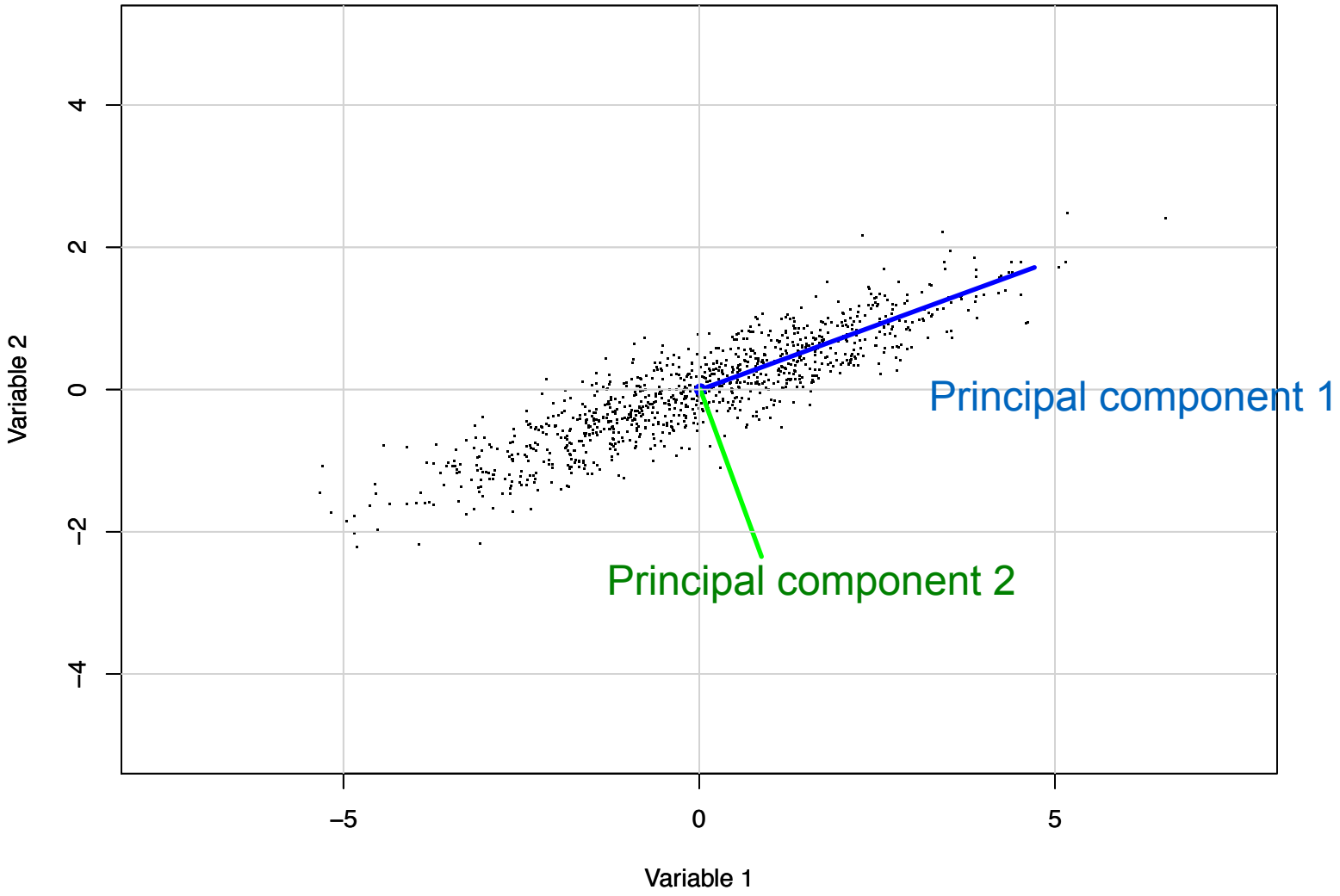
- The data are represented as a matrix
  - Images (matrices) are reshaped into vectors (1×D)
  - Usually each row is an observation and the variables are in columns
- Handling of missing/bad values. Some possibilities:
  1) Remove/ignore the whole observation if one variable is missing
  2) Compute the missing value by, e.g., averaging
  3) Use NaNs (often works in Matlab and R)
     - **NEVER** replace with 0 or 999999 or -1 etc.!!
       - If this is how you got your data, replace these special values with NaN before calculations…

Preprocessing (2):

- Normalising variable ranges (subtract μ, divide by σ)
  - Needed if the variables are not comparable, of the same type or in the same range
    - E.g., the variables are velocity (m/s), electron density ($1/m^3$), Kp-index (0–9)

- Dealing with correlated variables → dimensionality reduction
    - E.g. the 262144 variables in an all-sky image are obviously not uncorrelated!!

Illustration of variance in 2-D data



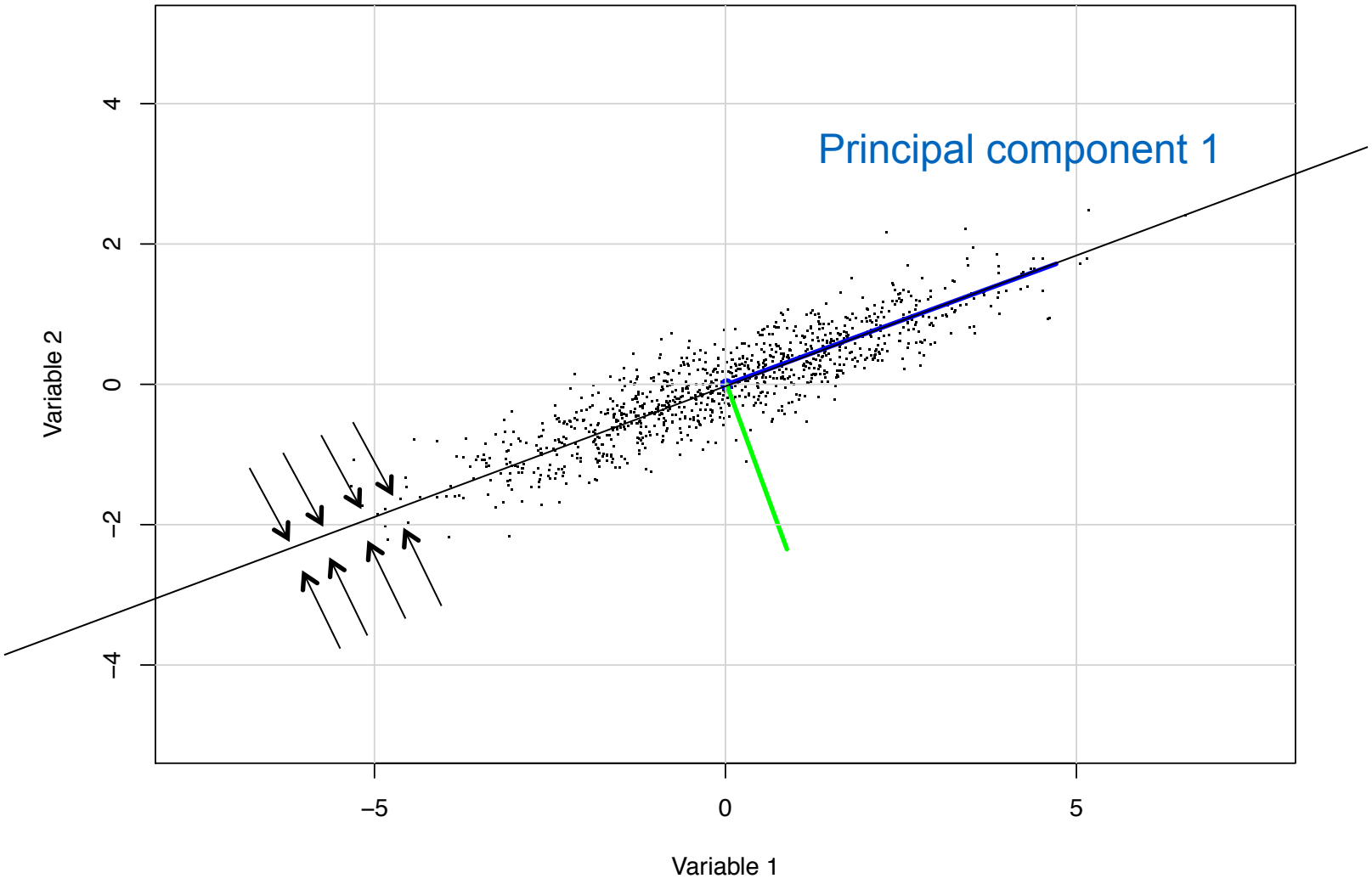
NOTE: PCA is computationally expensive and requires careful preprocessing to avoid “weird results”…

Dimensionality reduction

Standard tools exist, here are the most common ones

- Principal Component Analysis (PCA)
  - Projection matrix **rotates** the data
  - The principal components are usually ordered based on their significance in explaining the variation in the data
  - In R's MASS-library: `pca <-prcomp(x)`
- Linear Discriminant Analysis (LDA)
  - "Relative" of PCA
  - Explicit attempt to model the differences between the classes of data
  - In R's MASS-library: `lda <-lda(x,classes_of_x)`

Project 2D-data into 1D



Random projection

A computationally very light method

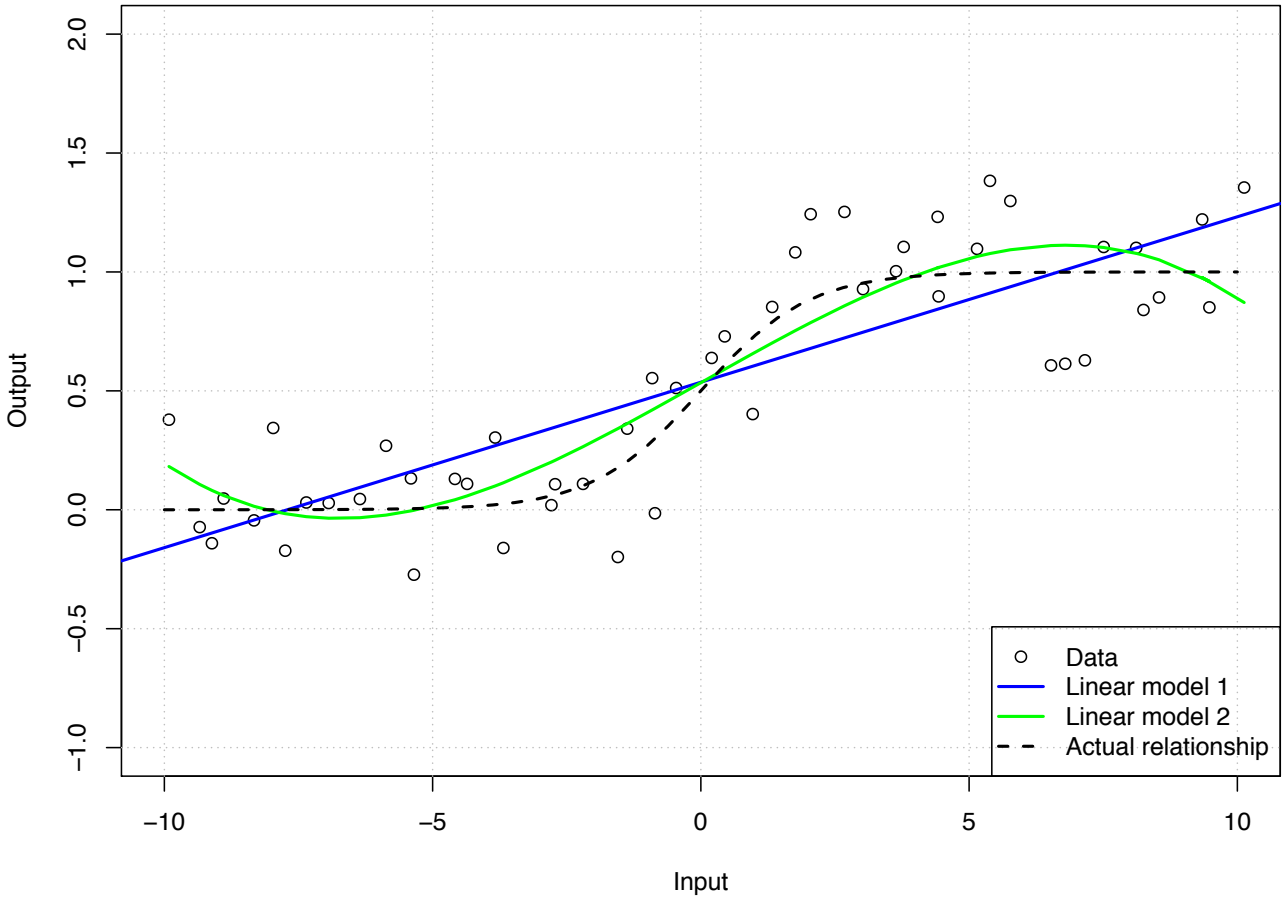
- Reduce the dimensionality by using a random projection matrix
    - In high dimensions, randomly selected unit vectors are *approximately* orthogonal (Johnson-Lindenstrauss Lemma)
- Given a D-dimensional data vector **X** (size D×1), reduce the dimension from D to K, where K<<D
    - Generate a random **R** matrix (size K×D)
    - Normalise the rows of **R** into unit vectors (1×D)
    - Project the data to a lower dimension by $\mathbf{X}_{LOW} = \mathbf{R}\,\mathbf{X}$

Computing with observations — metrics

- In order to compare observations, one usually computes the distances between data points in the feature space
  - A goodness measure or metric needs to be chosen
    - The squared error and the Euclidian distance (L2-norm) are commonly used
    - …but may not be optimal (e.g. histogram comparisons!)
  - Normalisation → all variables (dimensions) become equally important

- Similarity is associated with distance
  - Smaller distances mean more similar data (observations)
  - Good features and other dimensionality reduction methods preserve similarity

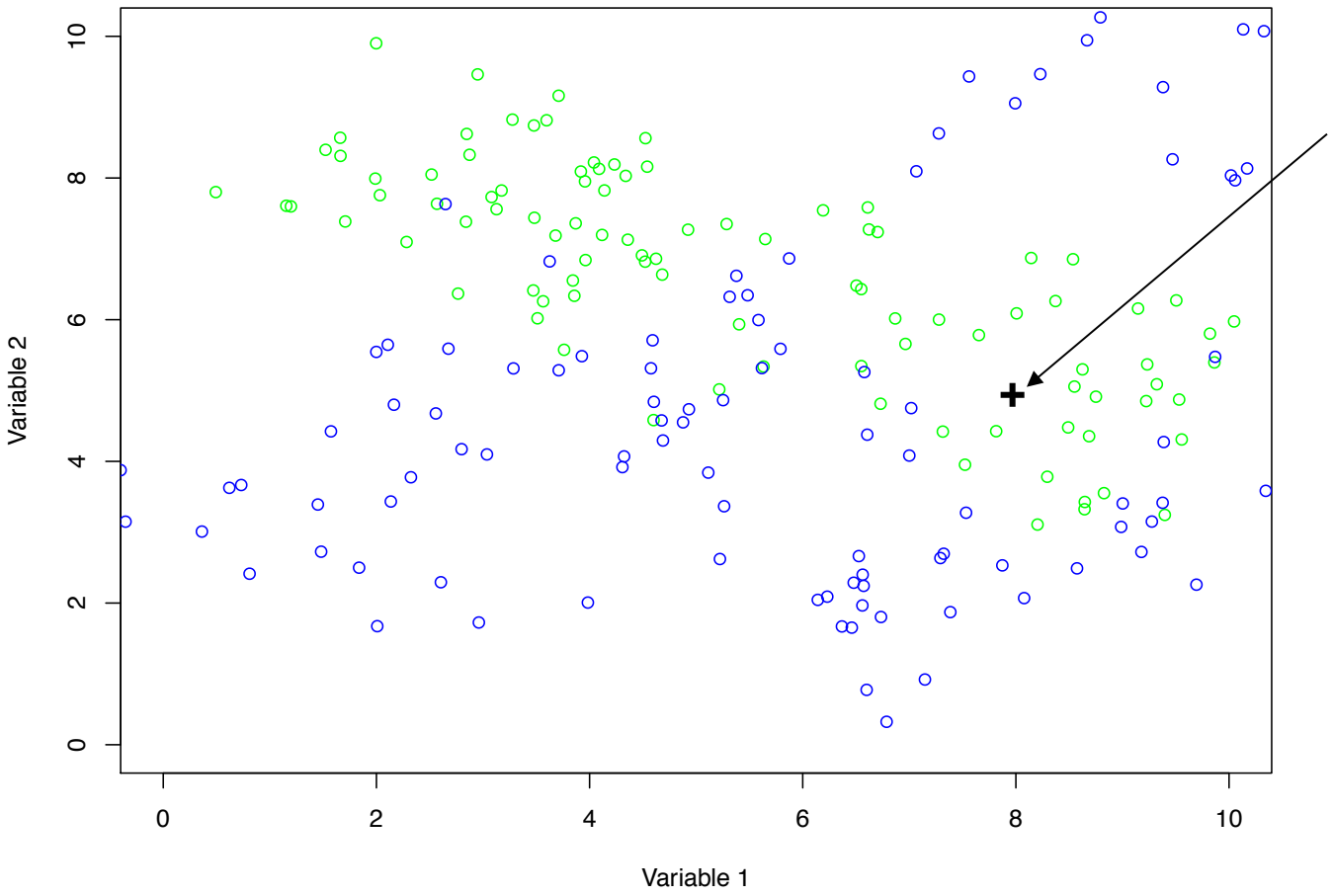
Regression — using metrics in validation



The residual sum of squares (RSS) of the Linear model 2 is smaller than that of the model 1.

Hence, the model 2 is better **according to this metric**.

Classification — using distance



When analysing the data visually, this previously unseen data point is clearly "inside" GREEN data. So, it likely belongs to the same class.

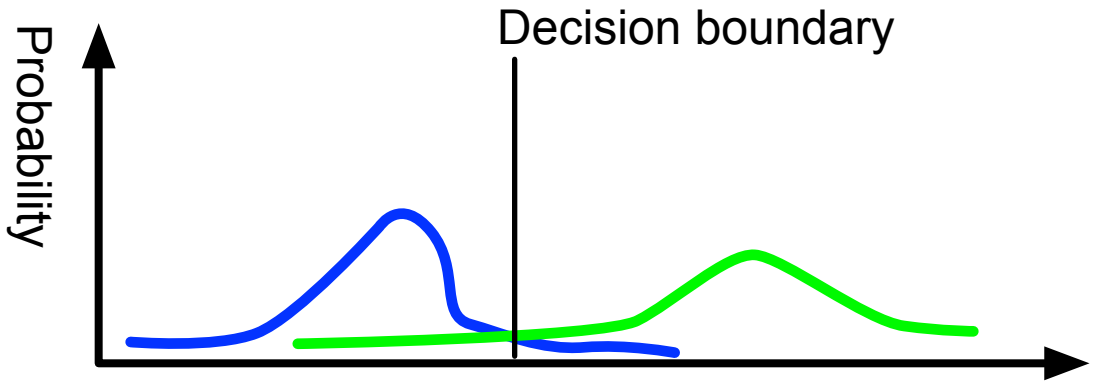
We can easily determine that the nearest neighbours are all GREEN when **using Euclidian distance as metric**.

Decision boundary — a trip to probability

- Given N different classes, the probability that a sample x belongs to class G_i is

$$p_i = \Pr(G_i \mid X{=}x)$$

- It is reasonable to classify the sample to the most probable class — this is known as the *Bayes classifier*
- Equally well one can classify the sample to the class of the closest known samples → density estimation!!



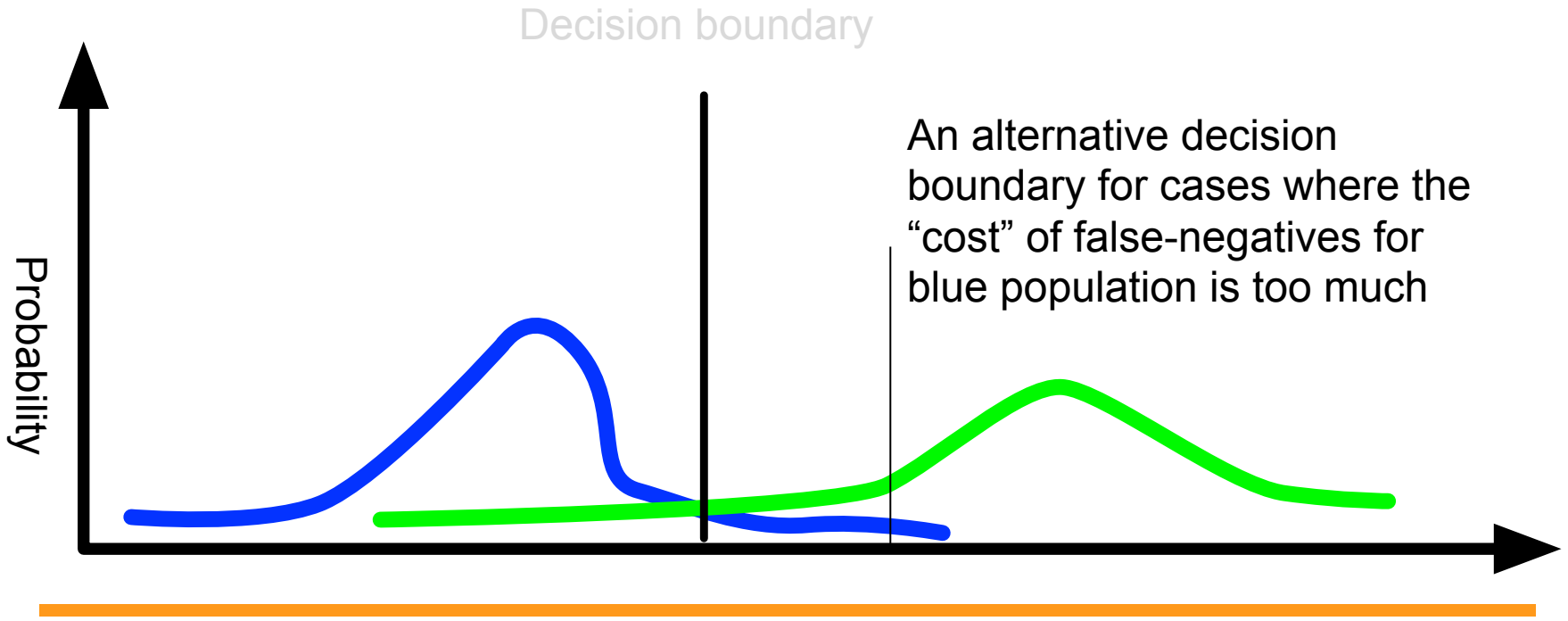
Classification errors — false positive/negative

	Auroras visible in the image	**No aurora visible in the image**
Computer says “aurora in image”	Correct	False positive (Type I error)
Computer says “no aurora in image”	False negative (Type II error)	Correct

Type III error: Correctly rejecting the null hypothesis for wrong reasons; Getting a correct answer to a wrong question…
— ill-defined problems with insufficient statistics!

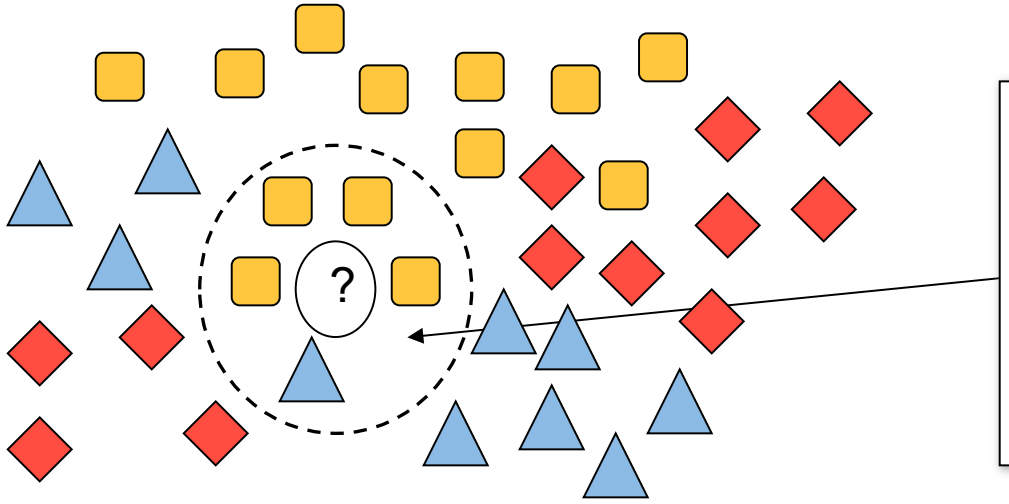
Risk of misclassification

The decision boundary need not be at the location where the class probabilities are equal!!



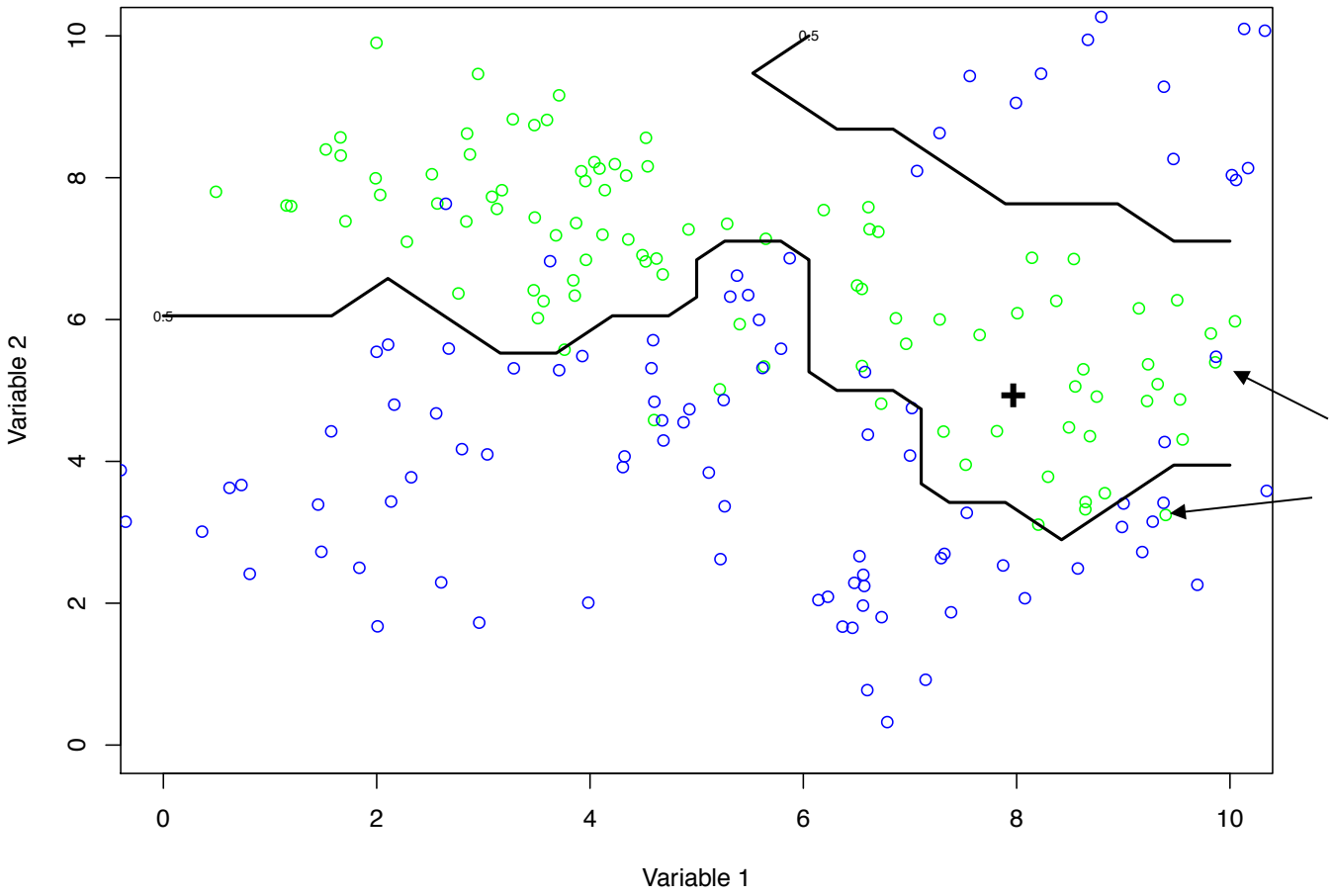
K-nearest classifier (KNN)

- Given a new observation, find the K nearest observations
- Based on the (known) classes of these K neighbours, compute the class for the new observation
- SIMPLE AND WORKS WELL!!
- Approximates the Bayes classifier
- In R's class-library: `results <- knn(train,test,classes,k=K)`
- A good value for K is needed — usually the choice is not critical ☺



The previously unseen sample is classified to the class of the majority of the neighbours: here

Classification — decision boundary



The decision boundaries determined using KNN with K=5.

The new sample is inside the GREEN region.

NOTE: there are classification errors!!

Supervised learning — learning with a teacher

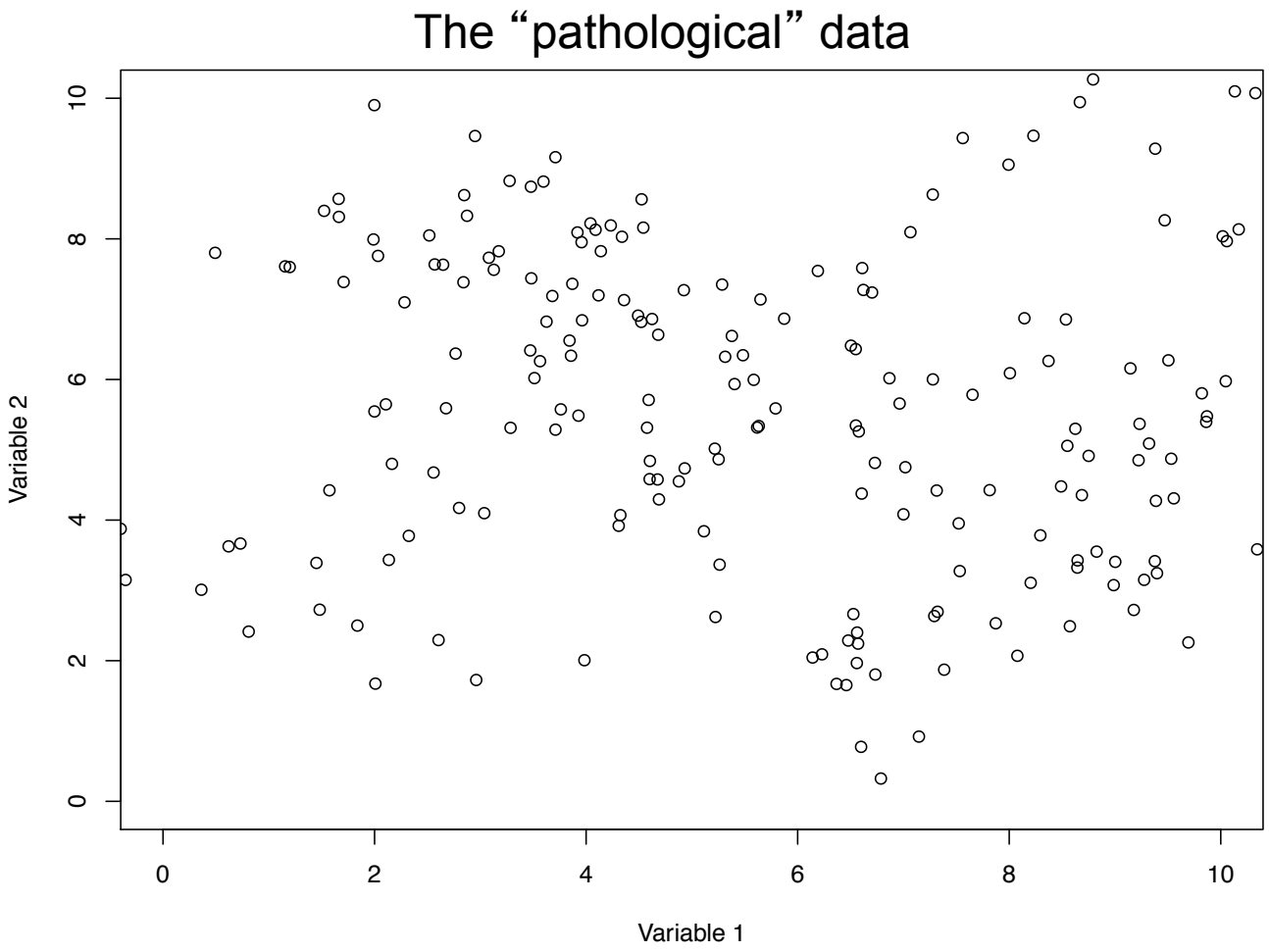
- If you have a number of samples (observations) whose class you know, this can be used in training a classifier
  - In mathematical terms this is an “optimisation problem”
    - You are searching for an optimal solution minimising a cost function
  - In practice, you need to find any minimum that provides a **sufficiently** good performance
- Popular classifiers:
  - K-nearest neighbours
  - Support vector machines

Unsupervised learning

- You have data but you do not know its internal structure — or whether it has an internal structure…
  - In mathematical terms this is also an "optimisation problem" but
    - You **do not know** the correct answers (cost function)
    - You **do not know** how many classes there are…
  - Problem approached by grouping observations based on their similarity and analysing the results
    - The challenge can be anything from trivial to impossible…

- Popular methods:
  - Association rules, principal and independent component analysis
  - Clustering: K-means (K-medoids), hierarchical trees

Unsupervised learning



What is the internal structure here??

How many different groups (classes) are there?

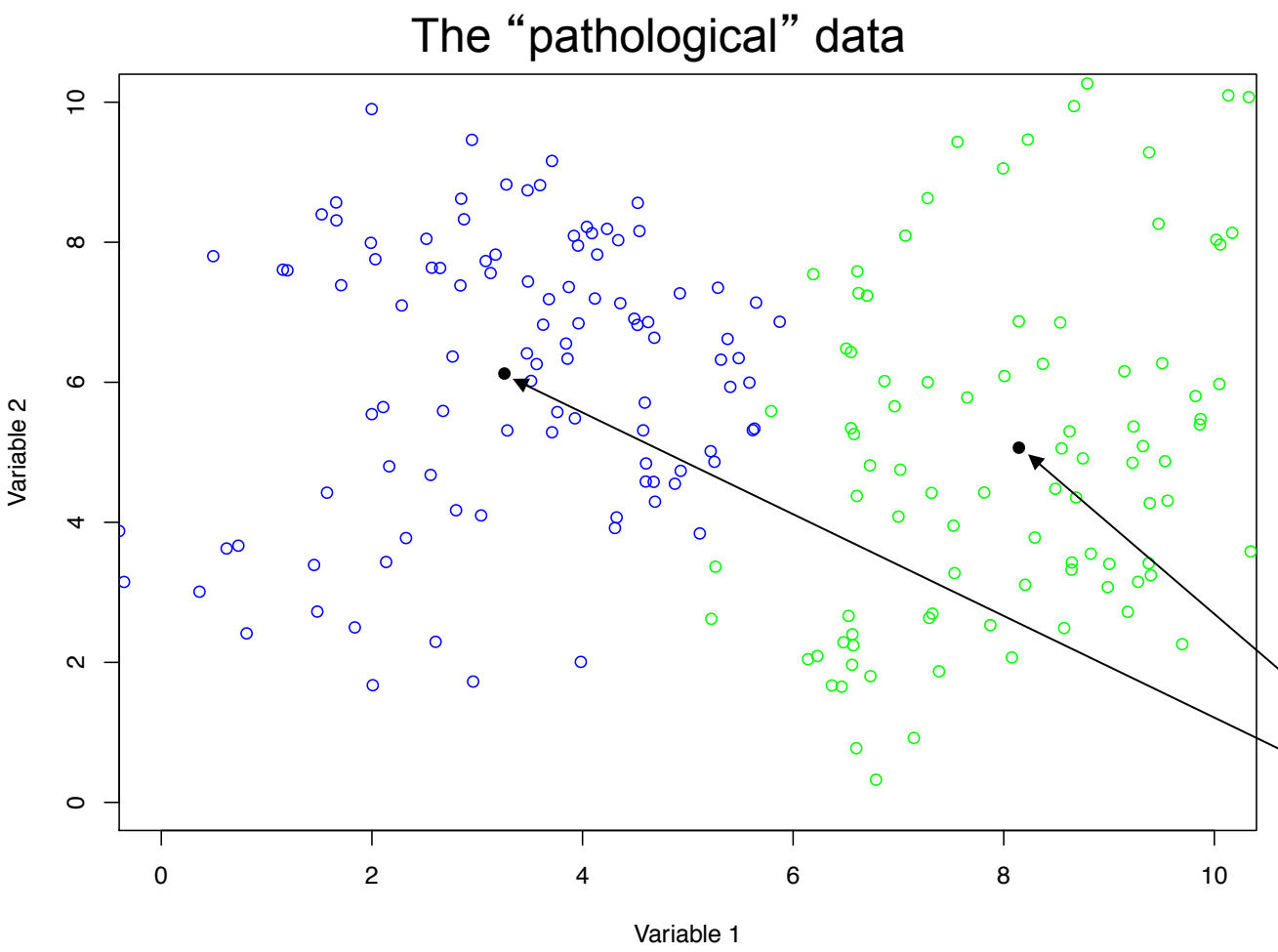
Are there different groups in the first place?

K-means clustering

- Most popular clustering method
  - YOU need to provide the number of clusters K
  - Standard in R: `km<-kmeans(x,K)`
  - For Matlab, free versions + Statistics Toolbox

- **Outputs K cluster centres and cluster assignments for data**
  - Minimises the "within cluster sum of squares" (WCSS)
  - Uses Euclidian distance
    - K-medoids works also if a proper metric cannot be defined but a value for similarity/dissimilarity can be determined (in R's cluster-library)
- Iterative
  - The initial cluster centres are usually selected randomly
  - Several runs → pick the "best" clustering with the smallest WCSS

K-means with K=2

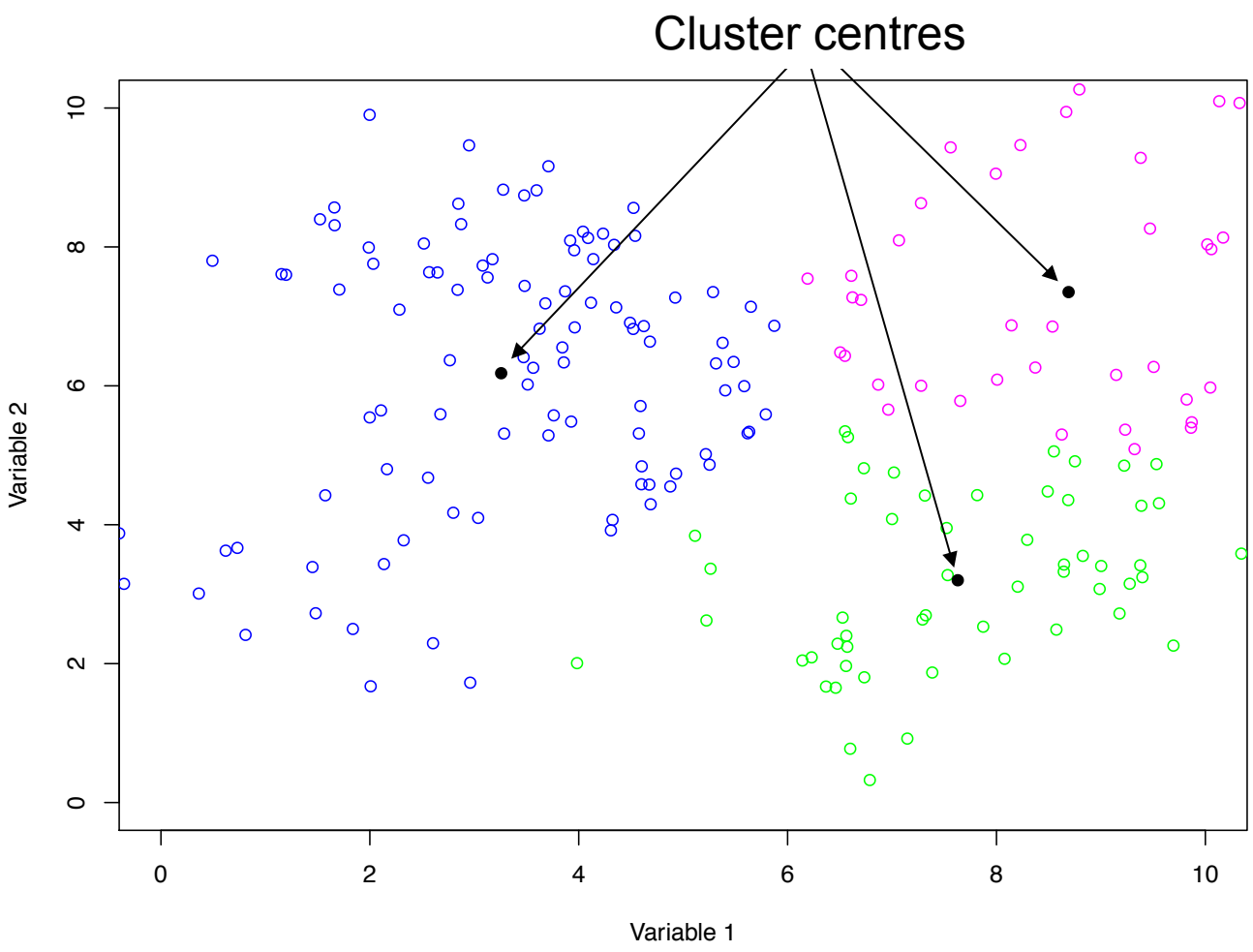


The “true” classes are obviously not visible here!!

Average WCSS≈730

Cluster centres

K-means with K=3

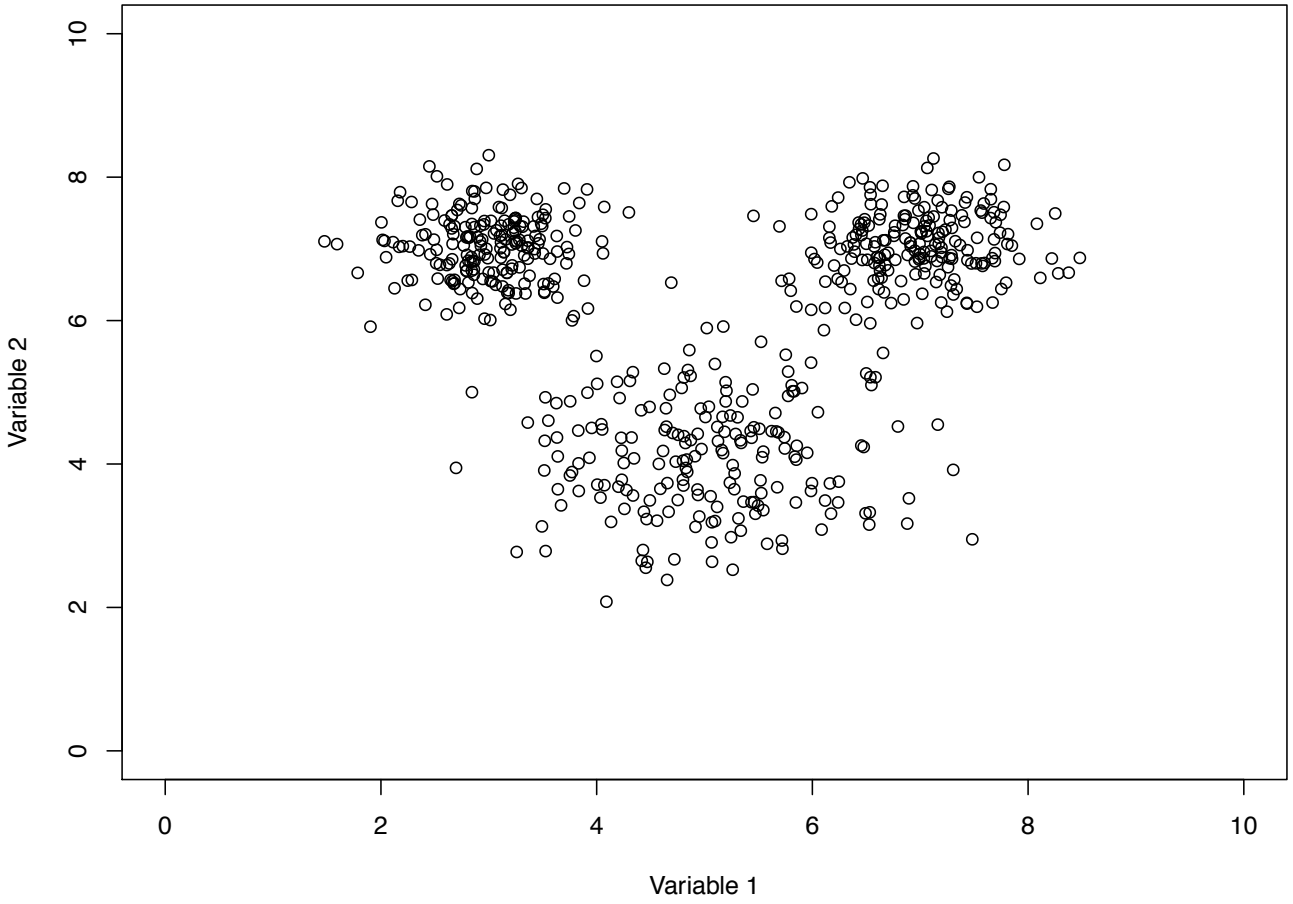


The “true” classes are obviously not visible here either!!

Average WCSS≈330

Note: the WCSS decreases as we increase the number of clusters. Why?

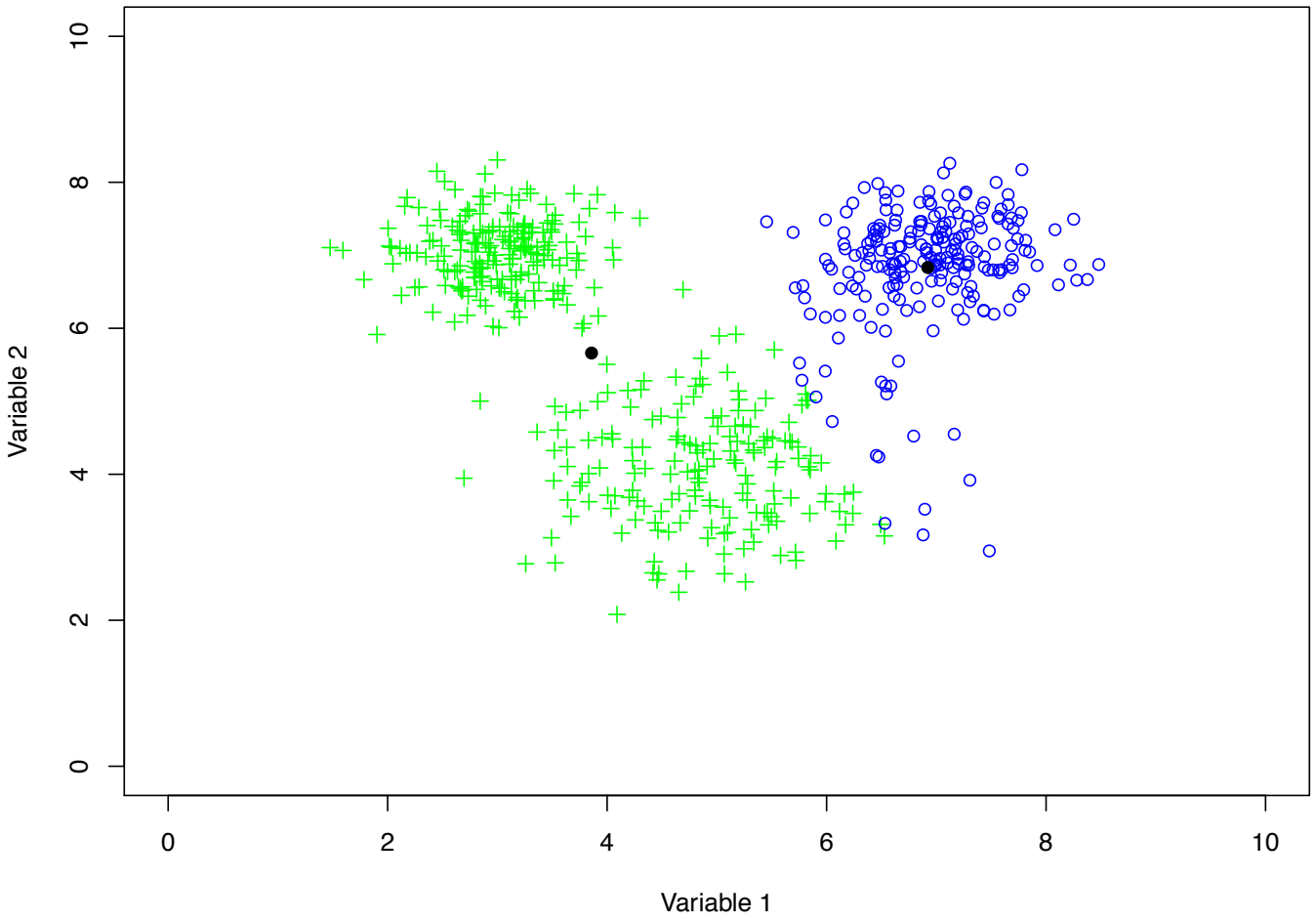
Clustering — a less pathological example



The original observations contain three obvious data clusters.

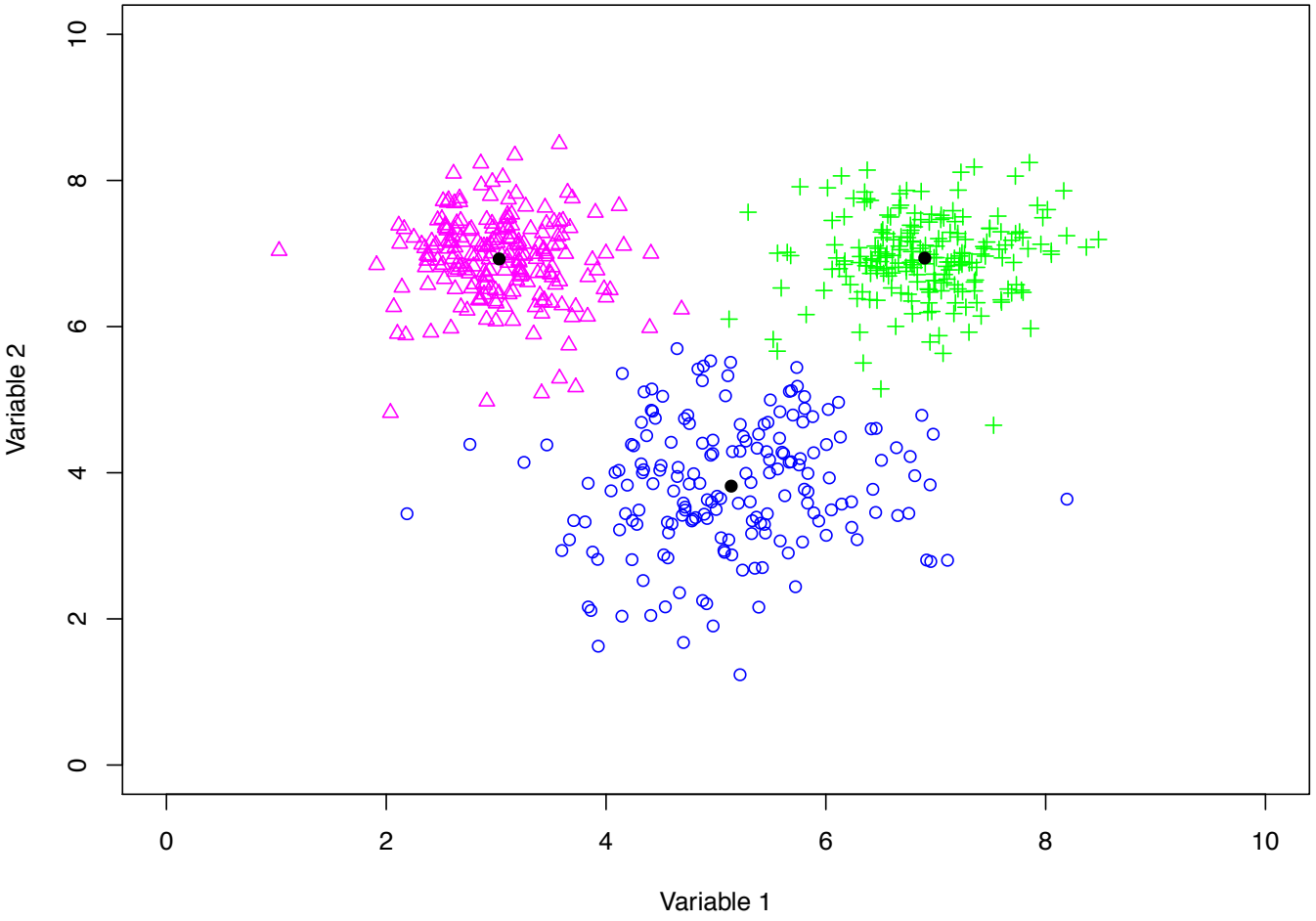
Clustering with K=2

WCSS≈2090



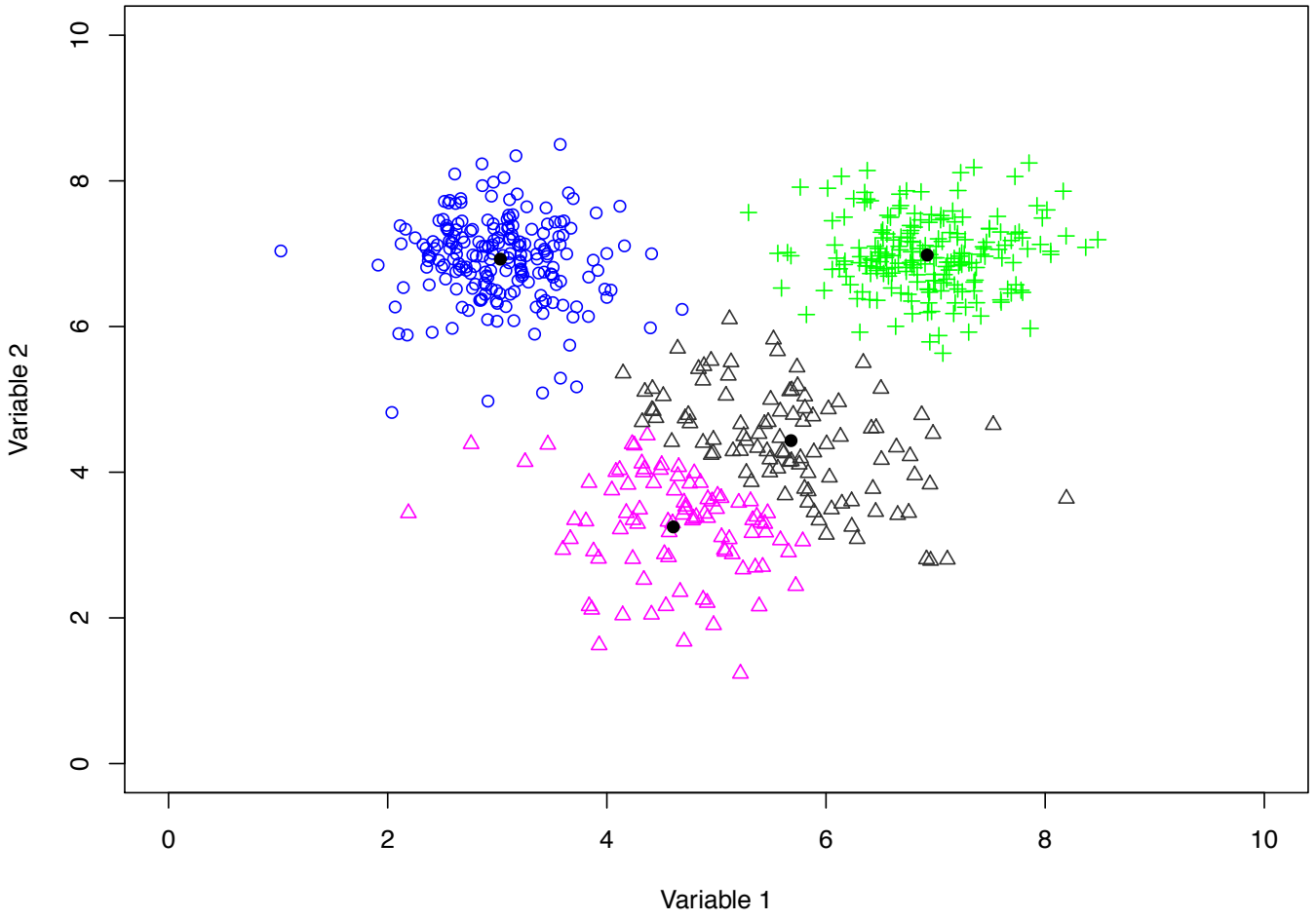
Clustering with K=3

WCSS≈540



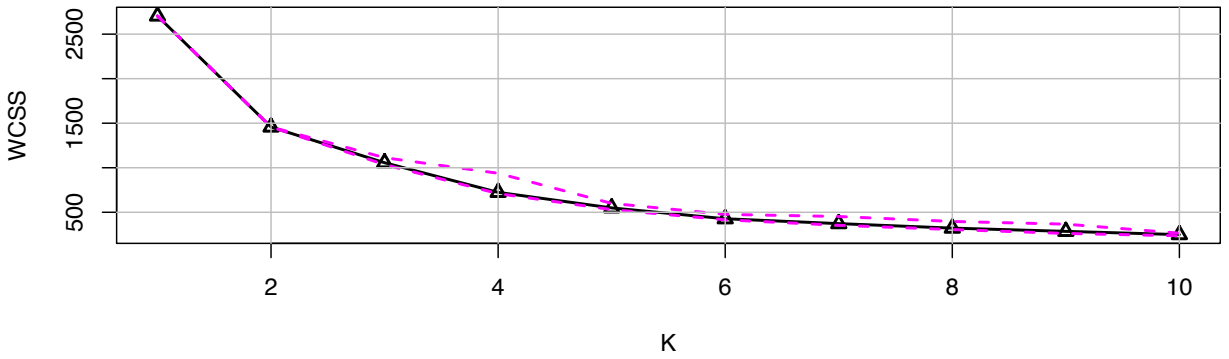
Clustering with K=4

WCSS≈425

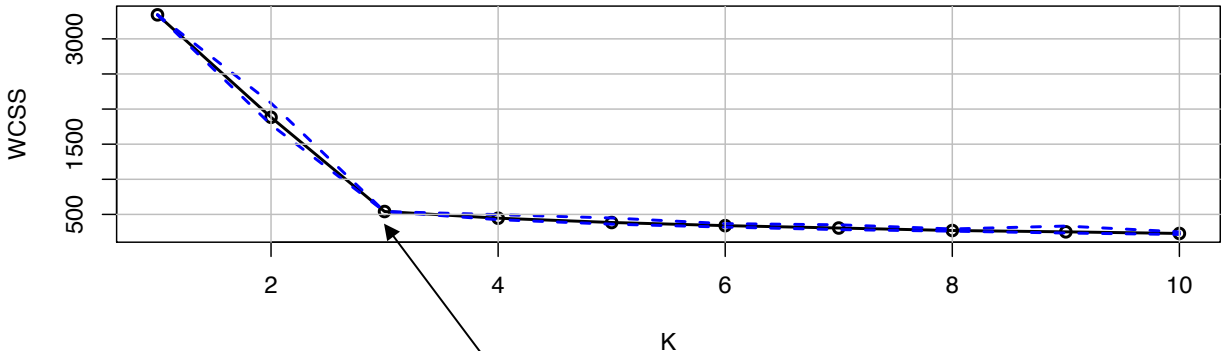


K-means — using WCSS to choose K

The "pathological" data



The data with three clear clusters



There is a clear "kink" at K=3

Many (heuristic) algorithms for choosing K can be found in the literature.

Intuitively, if K is too small, each cluster contains observations from multiple "true" clusters.
If K is too large, each "true" cluster is divided into sub-clusters. WCSS decreases more slowly.

In R: `wcss<-kmeans(x,K)$tot.withinss`

Supervised learning — error estimation

There are NO EXCUSES for not conducting an error analysis in any scientific research!!

Error estimation — “ideal case”

Training set

~50% of data

- Fitting models
- Training classifiers

Validation set

~25%

- Prediction error
- Model selection
- Choosing parameters

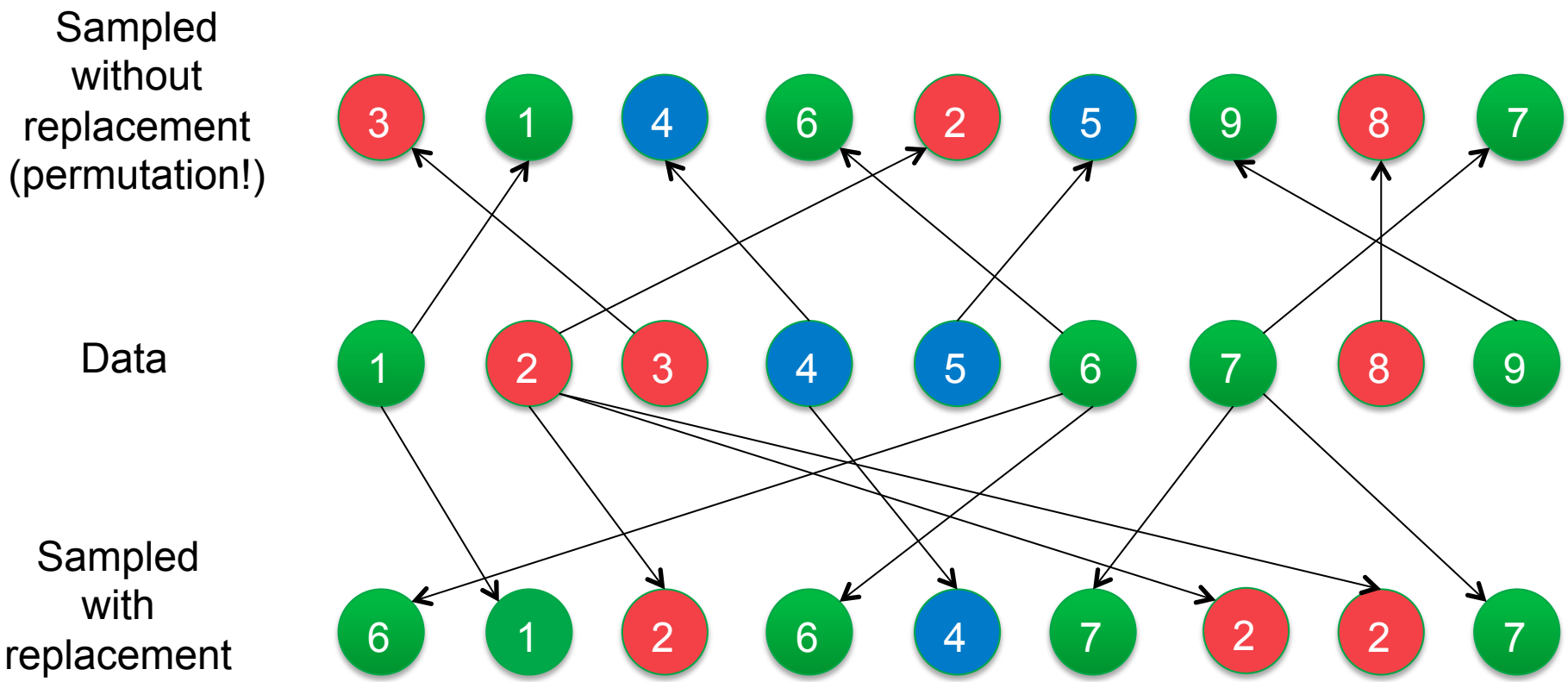
Test set

~25%

- Final error estimate
- Generalisation error

Set aside and use only at the end of the data analysis

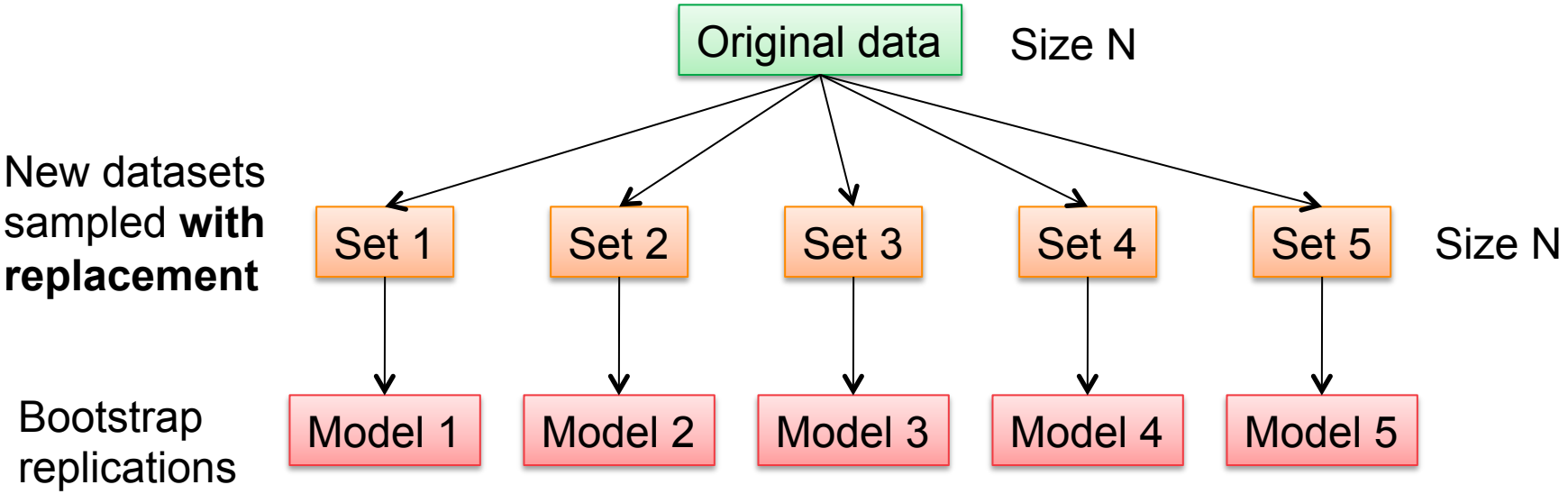
Sampling with replacement



In R:

```
newOne <- sample(x,size=N, replace=FALSE)
newOne <- sample(x,size=N, replace=TRUE)
```

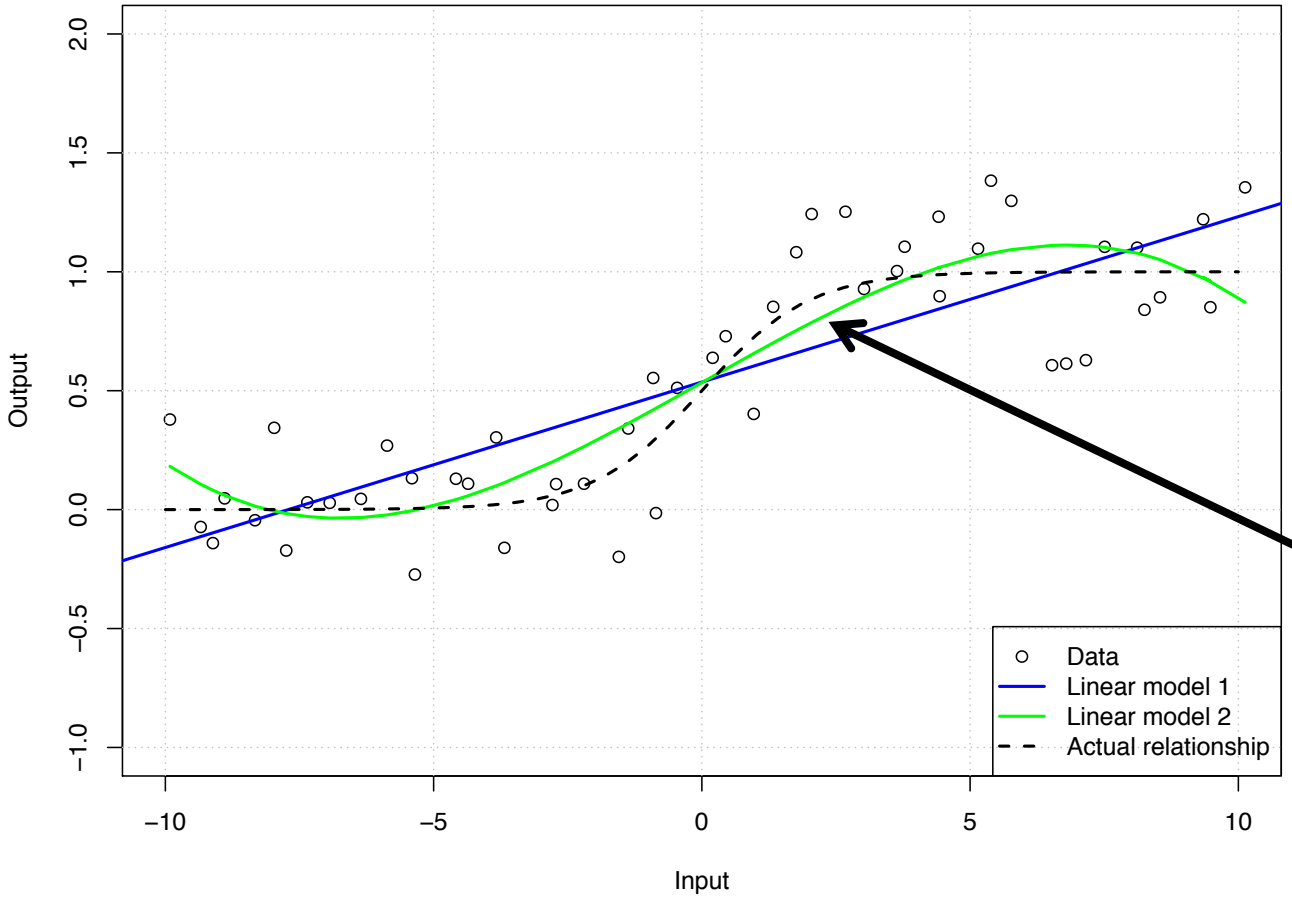
Error estimation — bootstrap



→ examine the behaviour of the replications

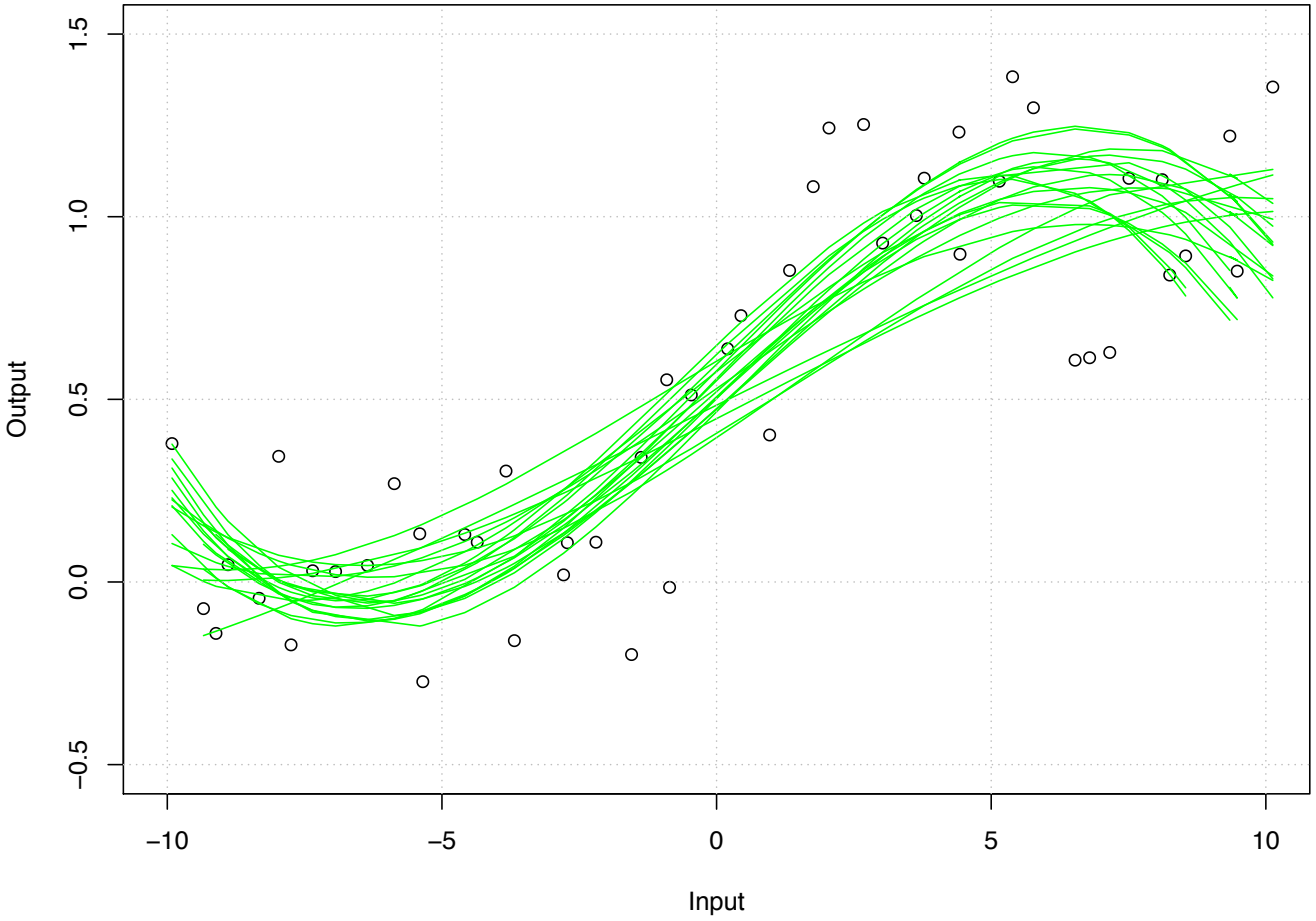
NOTE: it's not this simple in practice as there is a bias in the estimated error — don't re-invent the wheel but use existing software tools (e.g. R's boot-library)

Now, let's revisit the regression plot



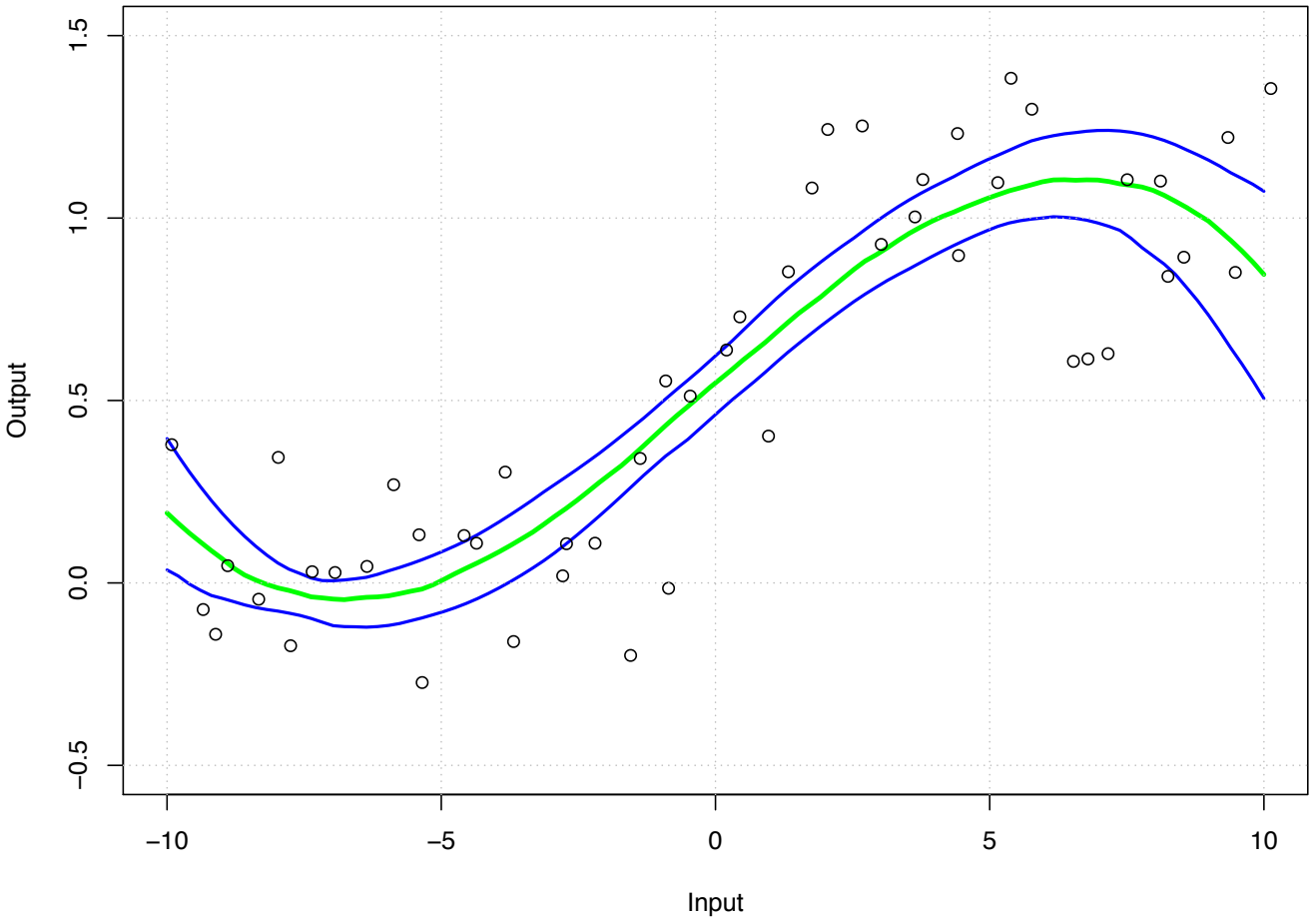
We choose the linear model 2 (green curve) and estimate the fitting error

Error estimation — bootstrap



Linear model 2 fitted with 20 replications of the observations.

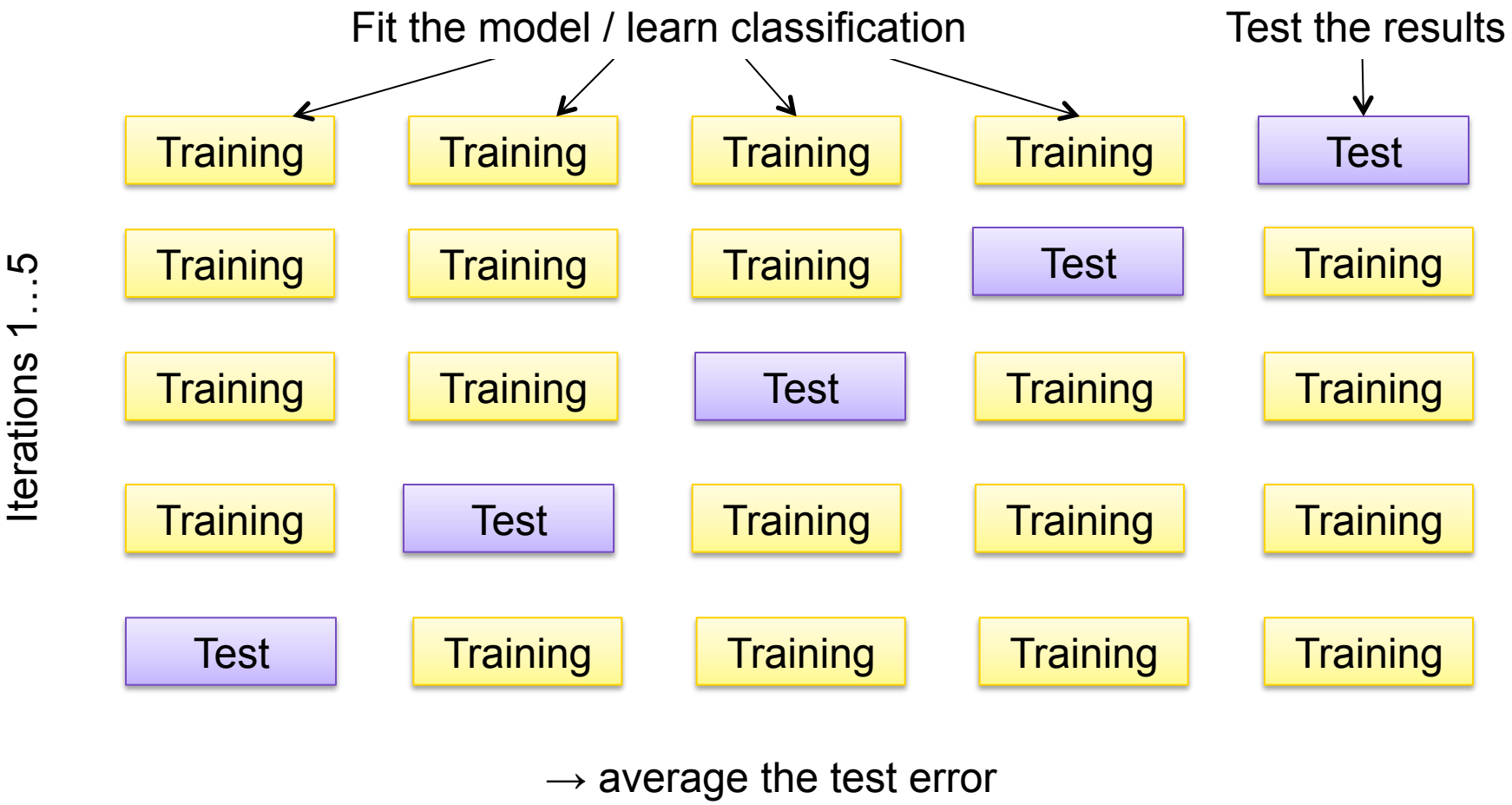
Error estimation — bootstrap



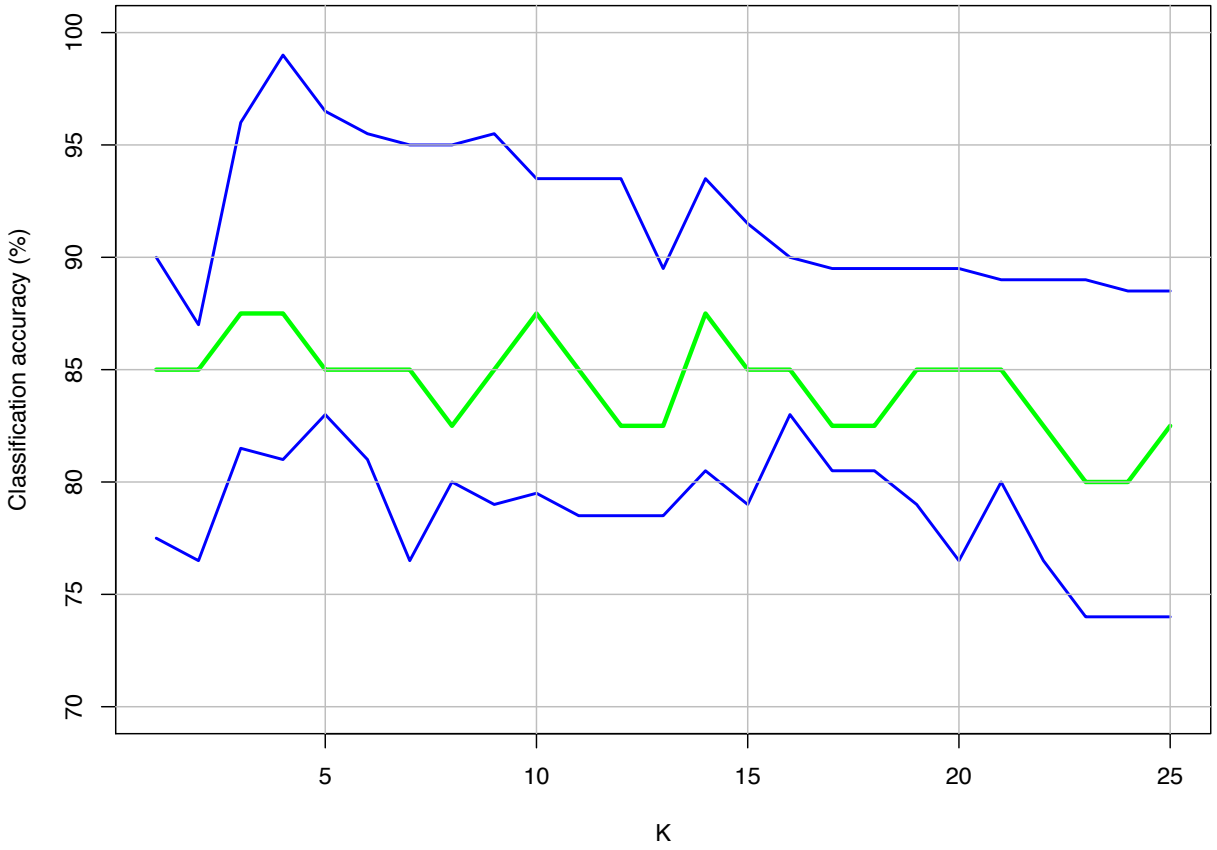
Linear model 2 fitted to 50 replications of the observations.

The 5% and 95% confidence levels and the median value are shown.

Error estimation — cross-validation



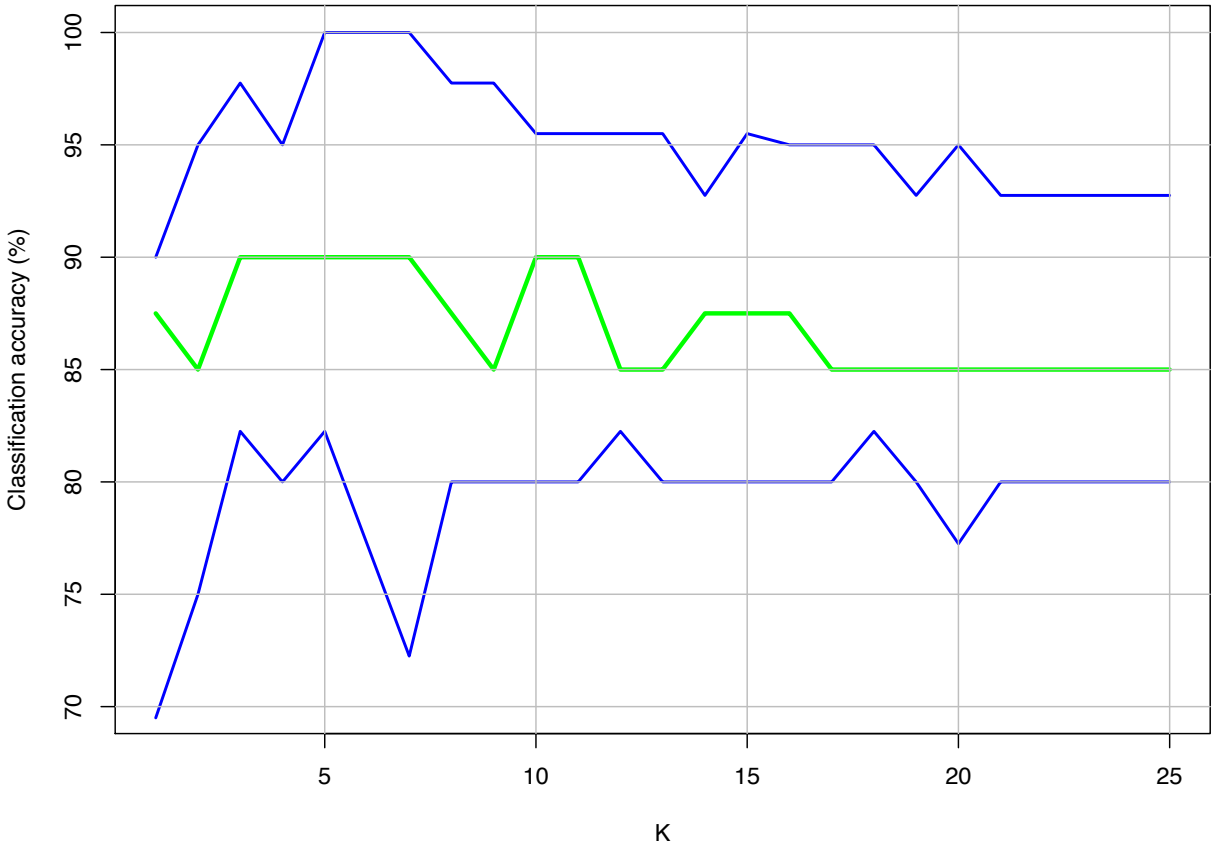
5-fold cross-validation



Pathological data classification with KNN.

The 5% and 95% confidence levels and the median value are shown.

10-fold cross-validation



Pathological data classification with KNN.

The 5% and 95% confidence levels and the median value are shown.

On the importance of features

No statistical analysis works if the input is garbage…

- Extract meaningful features from your data to
  - Reduce noise, complexity and dimensionality
  - Separate natural groupings, emphasise differences
- Computer scientists generally agree that
  - If features are well-chosen, the classifier choice is usually not critical *from the accuracy point of view*
  - It is useful to have many different features: an optimal subset is selected for any particular application
    - This is also a learning problem…

A review of automated analysis of auroral images

Aalto University
School of Electrical
Engineering

Literature review — general observations

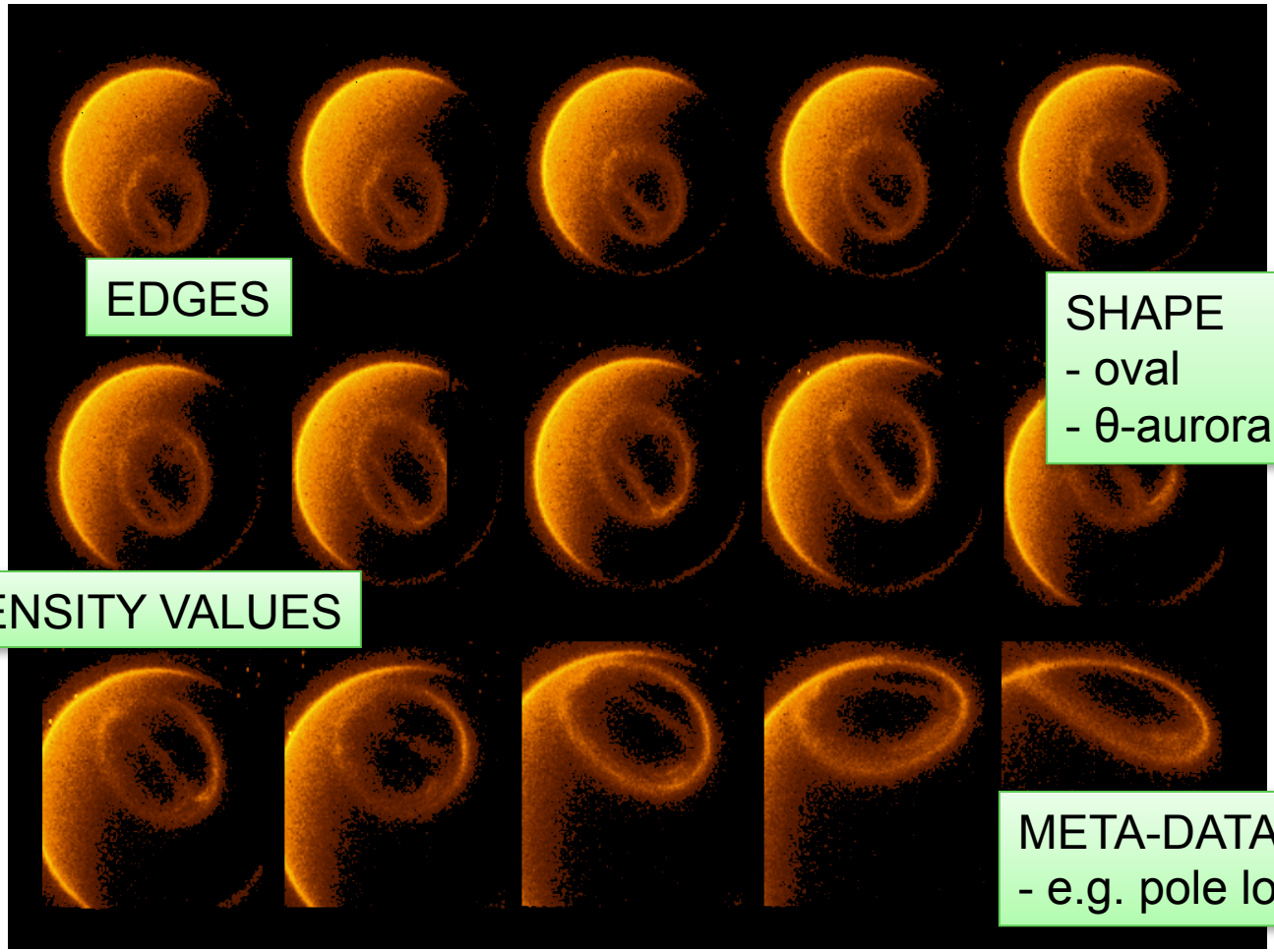
- We found 38 research articles (1989–2011) applying computer science methods to auroral image data
  - Practically all studies were motivated by the excess amounts of data
  - Neural networks were “fashionable” in the 1990’s ☺

- The early studies often relied on heuristics in feature extraction, error analysis and validation
  - In more recent papers, rigorous methodology from Computer Science is used ← peer-reviewed by computer scientists ☺

- There were no data mining or exploration studies
  - Supervised classification is still the norm
  - Manual labelling of data is still required

⬅ error analysis possible!!

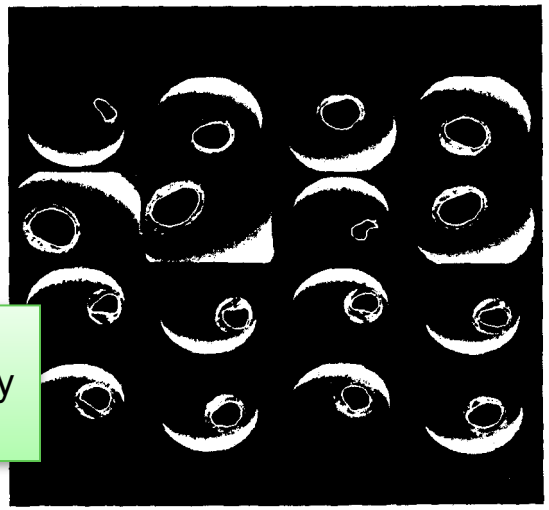
Possible features in a satellite image



Dynamics Explorer - 1 (DE-1) *8 Nov 1981 (81/312) 14:12–17:02 UT*

Auroral oval — boundaries and shape

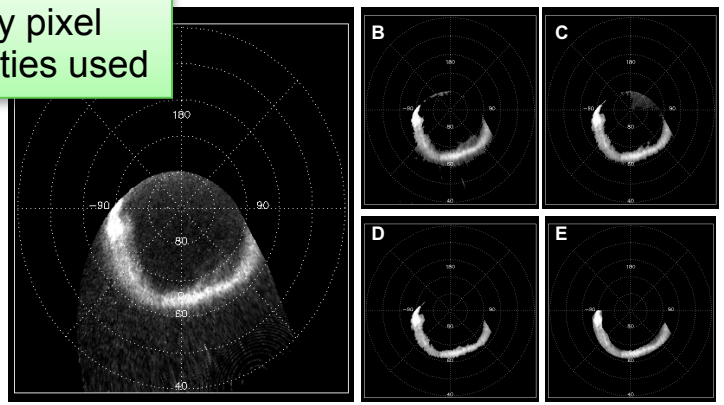
- The main objective is to segment the image into the foreground (aurora) and background (everything else including dayglow)
- Most successful approach: fitting a shape to data



Inner boundary only!

Mihovilovic et al., “Evaluation of an elastic curve technique for automatically finding the auroral oval from satellite images”, *Proc. Int. Geoscience and Remote Sensing Symposium*, Vol. 3, pp. 1268–1272, 1989.

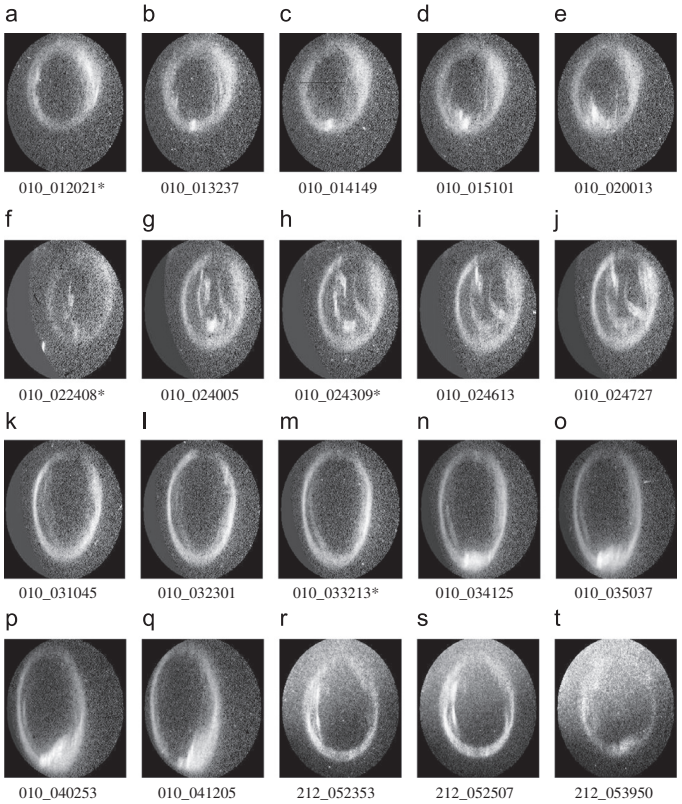
Only pixel intensities used



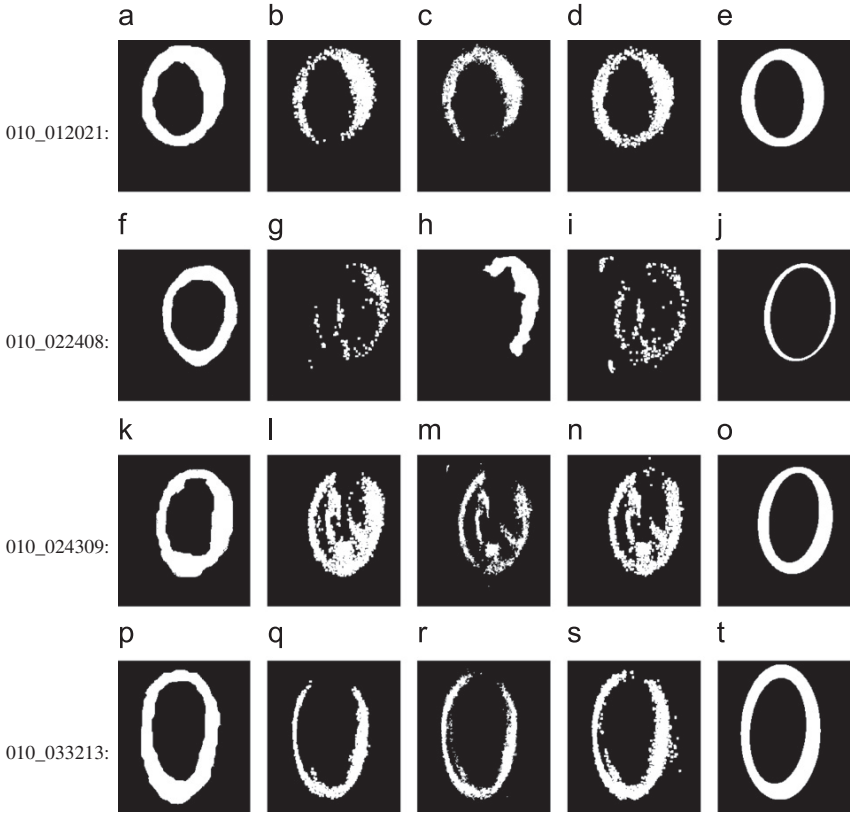
Li et al., “Comparing different thresholding algorithms for segmenting auroras”, *Int. Conf. Information Technology: Coding and Computing*, Vol. 2, pp. 594–601, 2004.

Auroral oval detection — state-of-the-art

Cao et al., “New shape-based auroral oval segmentation driven by LLS-RHT”, *Pattern Recognition*, Vol. 42, 607–618, 2009.



NASA Polar UVI



Manual

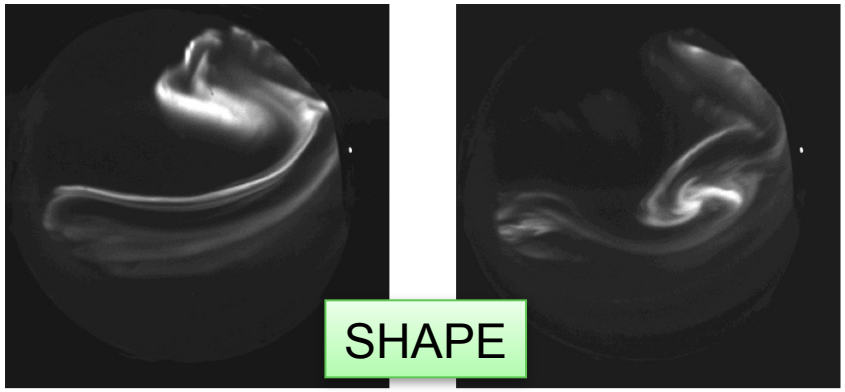
HKM
Hung & Germany, 2003

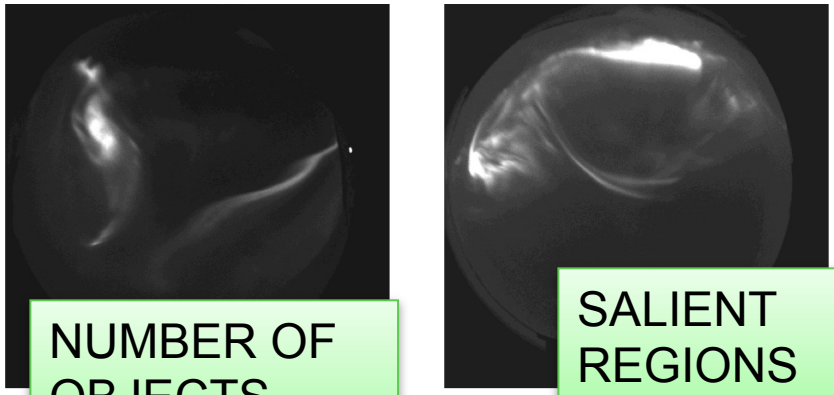
PCNN
Germany et al., 1997

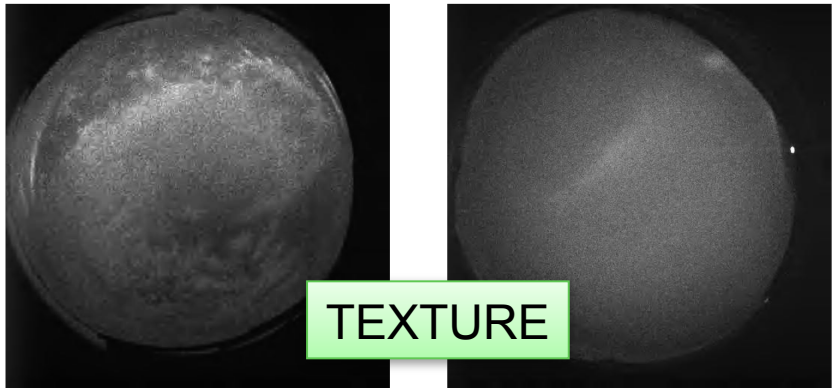
AMET
Li et al., 2004

LLS-RHT
(Linear Least Squares Randomised Hough Transform)

Possible features in an all-sky image







FMI emCCD ASC in Kilpisjärvi, 2009-2011.

Ground-based imaging — seminal studies

- Syrjäsuo and Donovan, “Diurnal auroral occurrence statistics obtained via machine vision”, *Annales Geophysicae*, Vol. 22, pp. 1103–1113, 2004.
  - Several features including texture were used to classify single images into multiple classes
  - 258 labelled images, 350000 images classified
  - Automatic pruning of images without aurora

- Wang et al., “Spatial texture based automatic classification of dayside aurora in all-sky image”, *Journal of Atmospheric and Solar-Terrestrial Physics*, Vol. 72, pp. 498–508, 2010.
  - Local Binary Pattern texture feature used for the first time in auroral research
  - 38044 labelled images, 71595 images classified
  - Manual pruning of cloudy and bad images

Category ambiguity

Syrjäsuo and Donovan, 2004

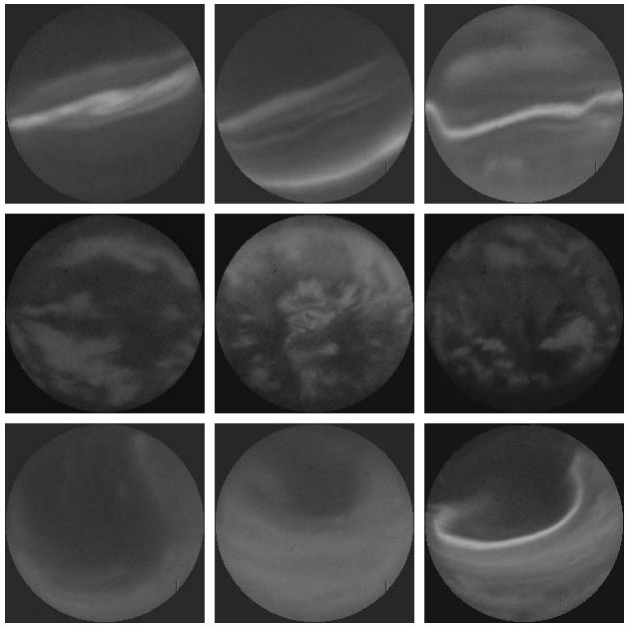
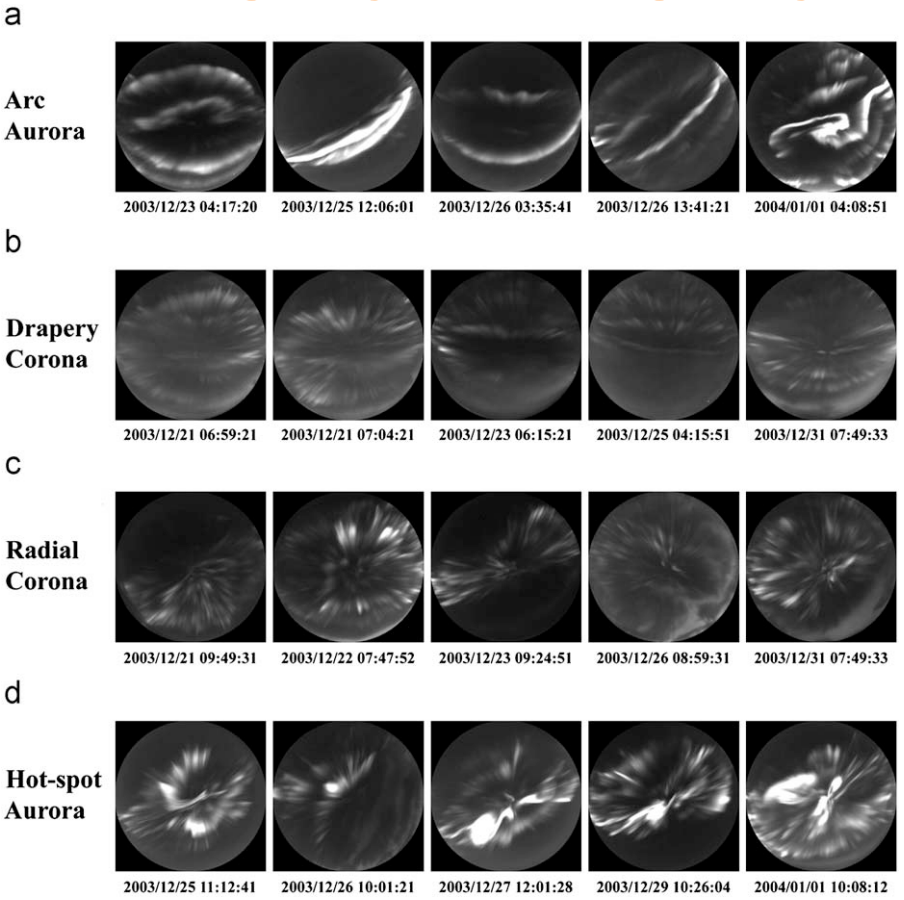
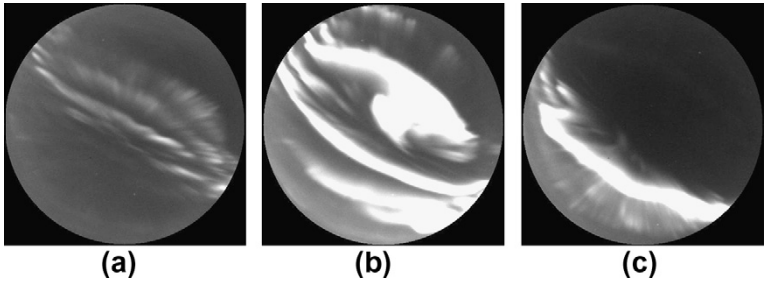


Fig. 4. Sample images from the three main auroral shape categories. Rows from top to bottom: auroral arcs, patchy auroras and Omega-bands.



Wang et al, 2010

Aalto University
School of Electric Engineering



Gao et al, 2011

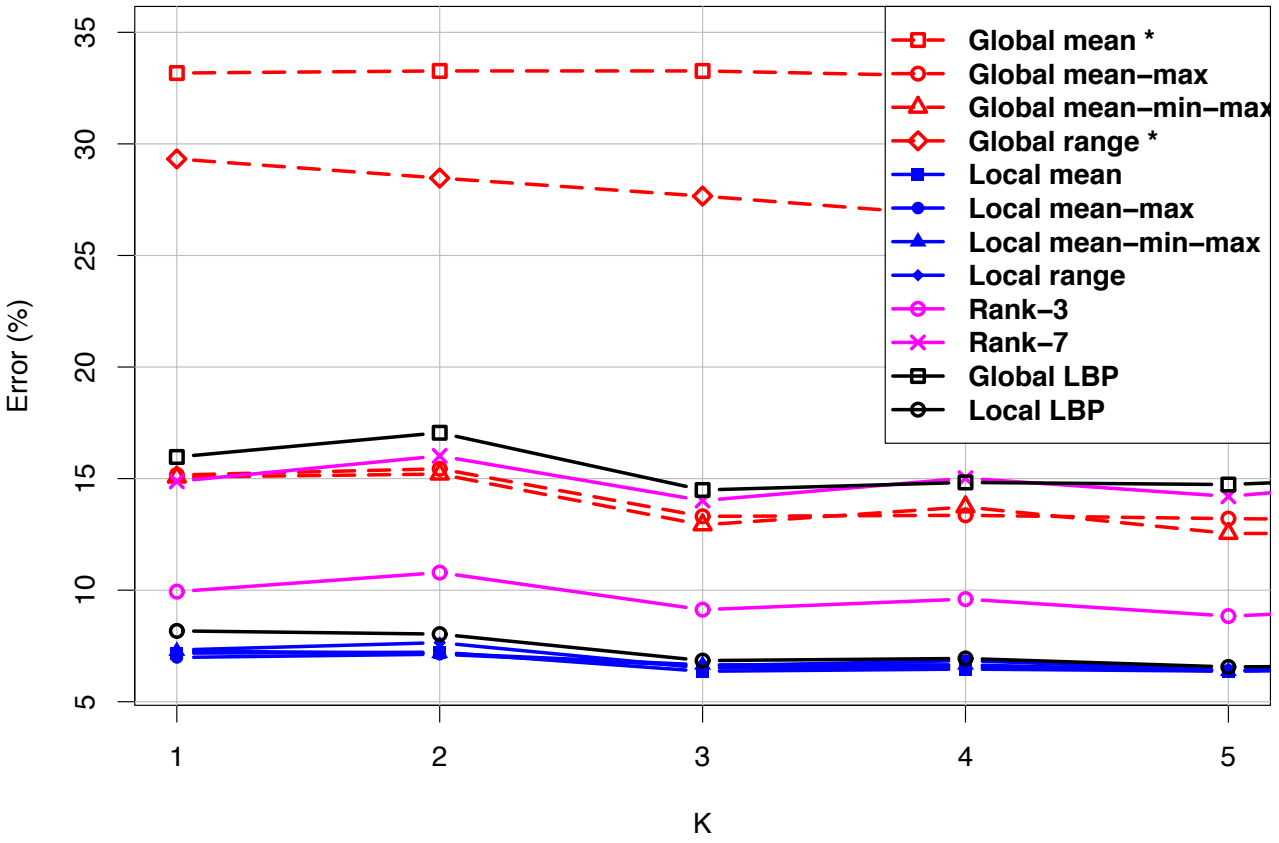
Fig. 1. Three distinct categories of auroras: (a) patchy aurora, (b) arcs, and (c) Omega-bands.

Comparison of different features

Aurora vs. no aurora

- Data from emCCD at Kilpisjärvi
- 30526 manually labelled images
- Moon visible in images
- 6…8% test error
- Error analysis via cross-validation

Syrjäsuo and Partamies, "Numeric image features for detection of aurora", *IEEE Geoscience and Remote Sensing Letters*, 2011 (doi: 10.1109/LGRS.2011.2163616)



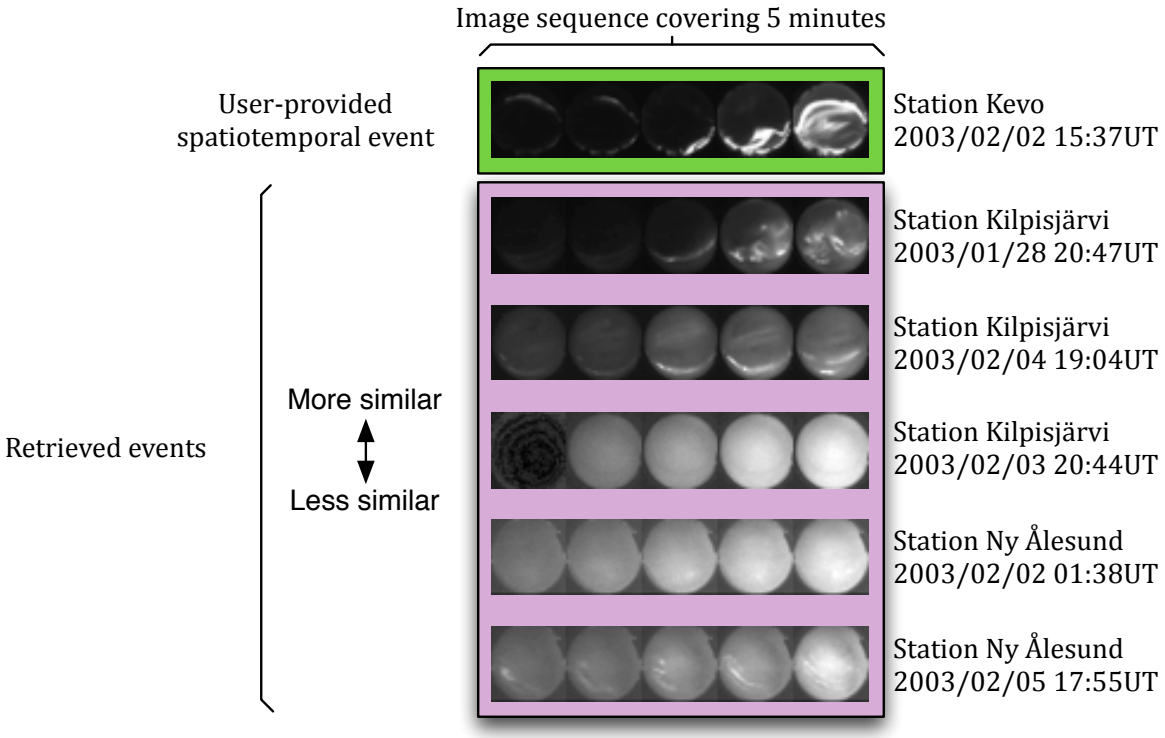
Directions for future research

- Shape-based analysis is popular in other applications, too
  - Active contours, probabilistic shape models
- Texture-based methods have been successful but are they sufficient to be used without any other features?
  - Evaluate a large number of different features and their suitability for clustering similar images or classifying images correctly → feature selection
  - Determine a compact set of computationally efficient features for large scale automated analysis → search for frequent patterns
- Experiments with existing video retrieval and activity learning methods
  - The aurora is a dynamic natural process — the temporal auroral activity has not really been studied using methods from Computer Science
  - These topics are of interest to computer scientists who are interested in difficult data and participating in major discoveries…

Case-study on temporal analysis

"Retrieval of 5-min events"

Instead of using Singular Value Decomposition, we used Random Projection to reduce the feature vector dimensionality from 5120 to 50. In other words, no need to determine the eigenvalues of a 5120×5120 matrix ☺



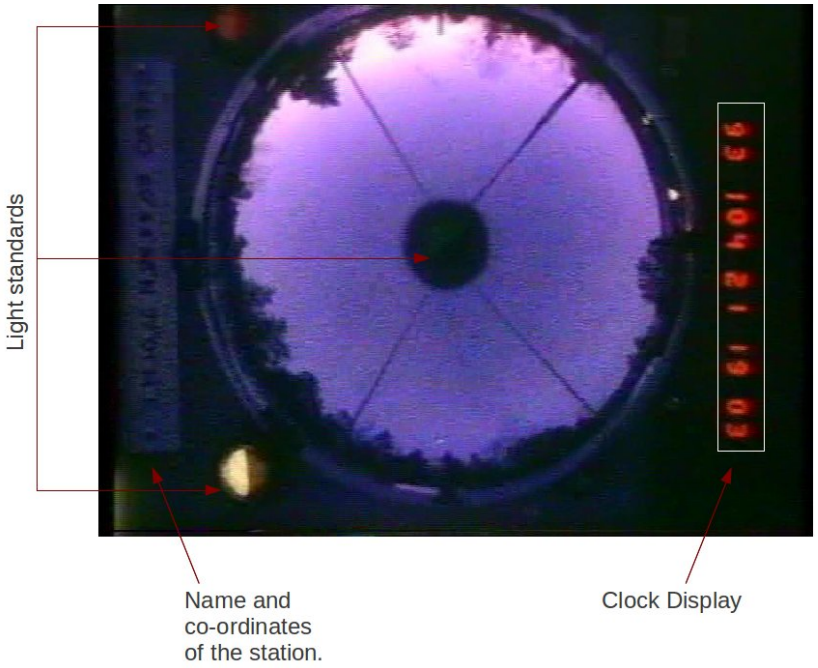
Syrjäsuo and Hollmén, *unpublished*…

Summary

- Error estimation is a must
  - For measured data, this is actually easy to do, so don't skip it!
- Reduce data dimensionality and complexity by using compact numeric features
- Features used in the analysis are more important than the classification/ clustering algorithms
  - Form a set of (uncorrelated) features from the same data
  - Select a subset for trial runs
  - Use KNN or K-means/medoids first to get an estimate of the performance

LET THE COMPUTER DO THIS WORK…

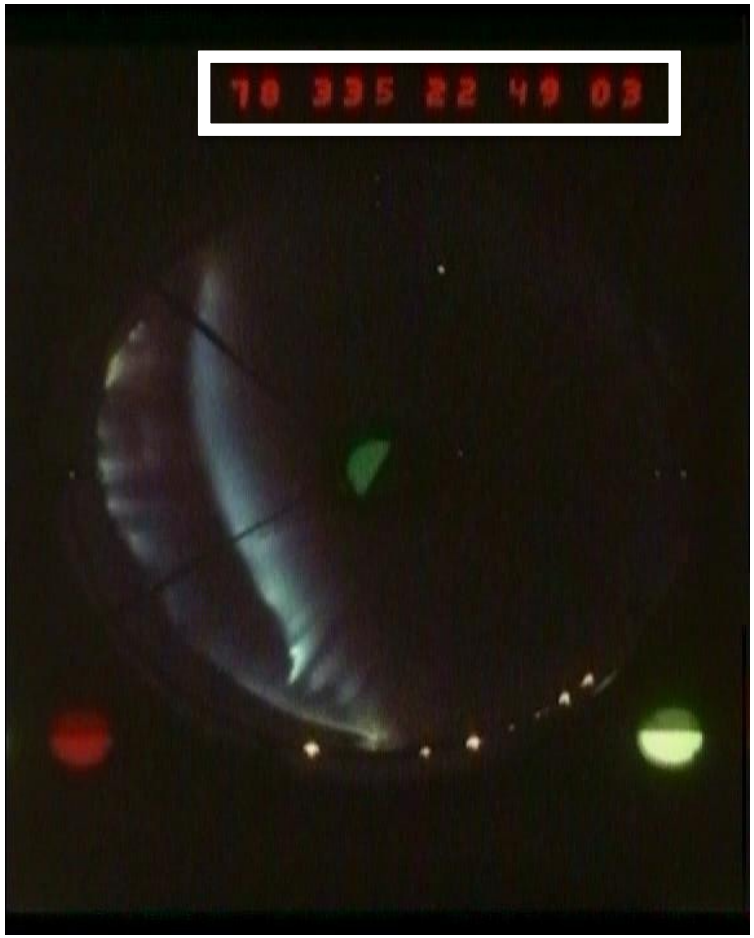
Exercise #1 — unsupervised learning (1/2)



- In 1973–1995, several film-based all-sky cameras were operated in Finland
- The films have finally been digitised into DVDs in 2011

- We want to automatically extract the date-time information captured in each frame to allow convenient data processing with computers

Exercise #1 — unsupervised learning (2/2)

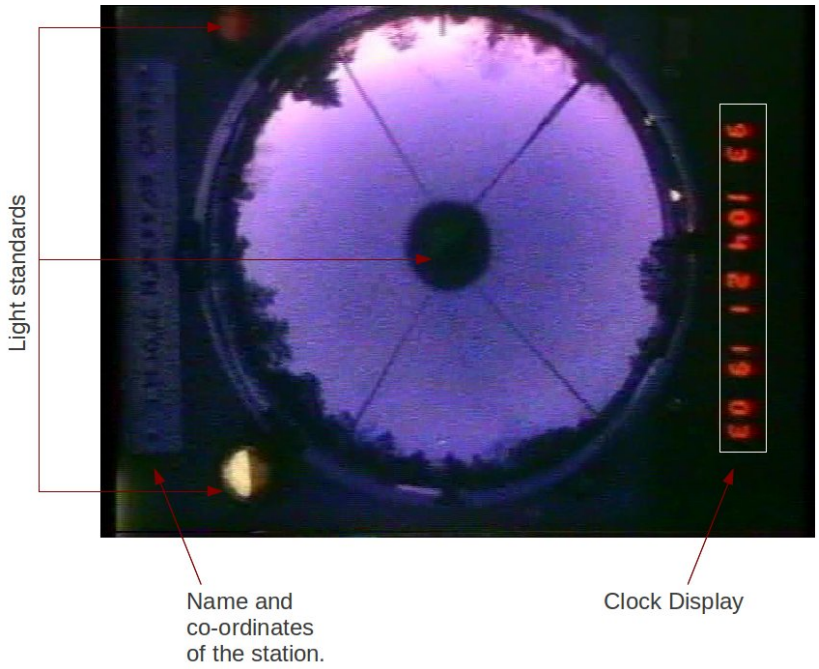


- Individual video frames have already been processed
  - Detection of the clock display
  - Separation of individual digits
  - Extraction of features
- There are 9328 observations of 94-dimensional data
- Use K-means to find the best number of clusters

Exercise #2 — supervised learning

- There are 9328 observations of 94-dimensional data
- There are also manually determined correct classes for each observation

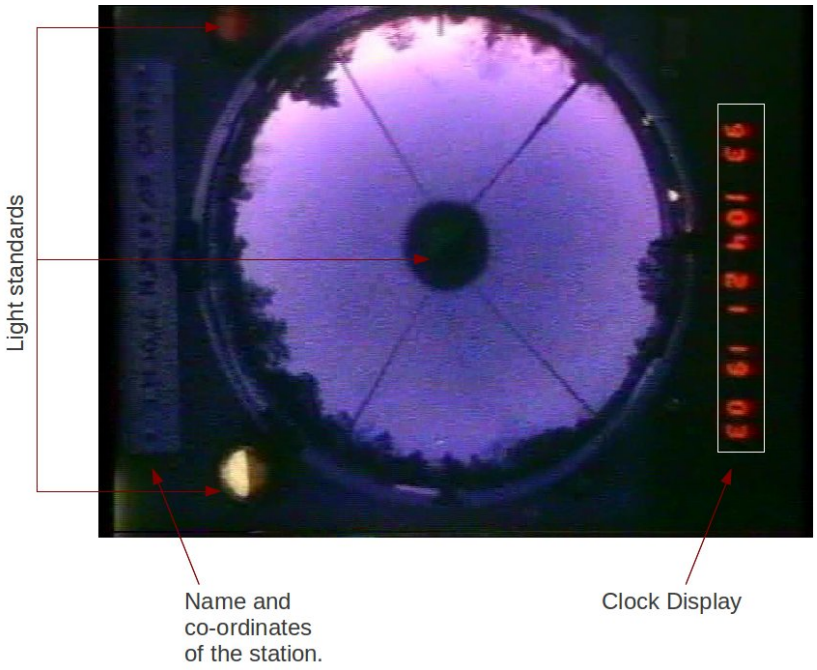
- Use K-nearest neighbours and estimate the classification accuracy (and its error)



Exercise #3 — supervised learning

- There are 9328 observations of 94-dimensional data
- There are also manually determined correct classes for each observation

- Use PCA to reduce the dimensionality to D<<94
- Use K-nearest neighbours and estimate the classification accuracy (and its error)



Practical details

http://www.R-project.org/

- Half of the students should work on Exercise #1 and the other half on Exercise #2 (first)

- There are two R-scripts for testing the methods
  - Installing R on your own laptop is highly recommended, but you can use any software you prefer, too!
  - From the command line, use the command `source` to run the scripts
  - Edit the coefficients and parameters in the script
  - `exercise1_clustering.R` for K-means clustering
  - `exercise2_knn.R` for K-nearest neighbour classification
  - `exercise3_knn_pca.R` is as above but with dimensionality reduction via PCA
    - What is the lowest dimensionality that still "works ok"?

QUICK HELP IN USING R
http://www.statmethods.net/