Data and Data Exploration

Xiaochun MAI

Shenzhen University

Outline

- Additional remarks on Principal component analysis
- Data Preprocessing

  5) Feature Subset Selection

  6) Feature Generation/Creation

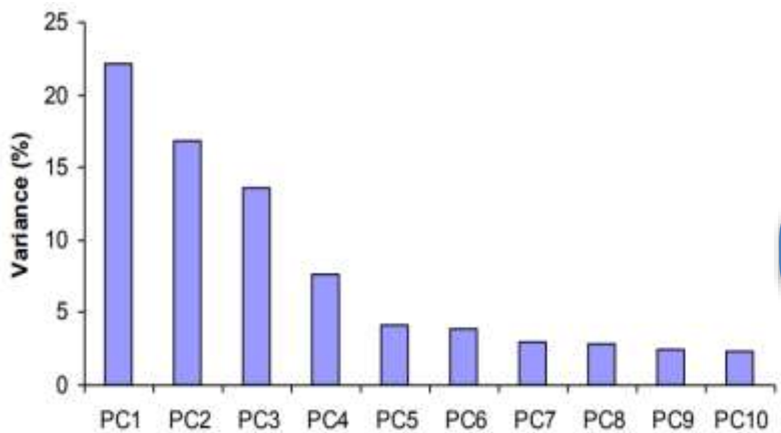
  7) Discretization and Binarization

  8) Attribute Transformation

- Measure of Similarity & Dissimilarity
- What is data exploration?

Additional remarks

- How many PCs?
  - We want to retain as much information as possible using these components.
  - We can compute each PC explains how much variance and then makes decision (still a parameter).



$$\frac{\lambda_k}{\sum_{i=1}^{N} \lambda_i}$$ Propotion of variance

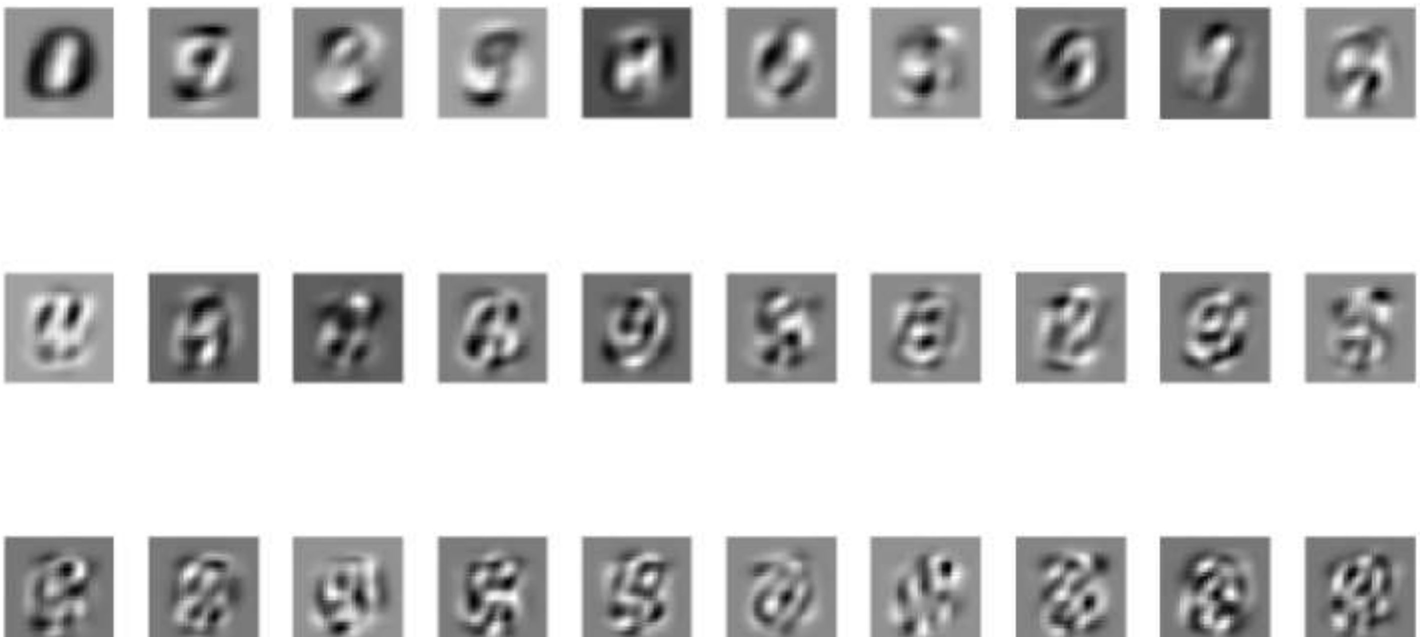
$$\frac{\sum_{k=1}^{d} \lambda_k}{\sum_{i=1}^{N} \lambda_i}$$ Cumulative propotion

Principal Component Analysis

MNIST  $= a_1 \underline{w^1} + a_2 \underline{w^2} + \cdots$

images

30 components:



Eigen-digits

Principal Component Analysis

Face

30 components:

http://www.cs.unc.edu/~lazebnik/research/spring08/assignment3.html

Eigen-face

Principal Component Analysis

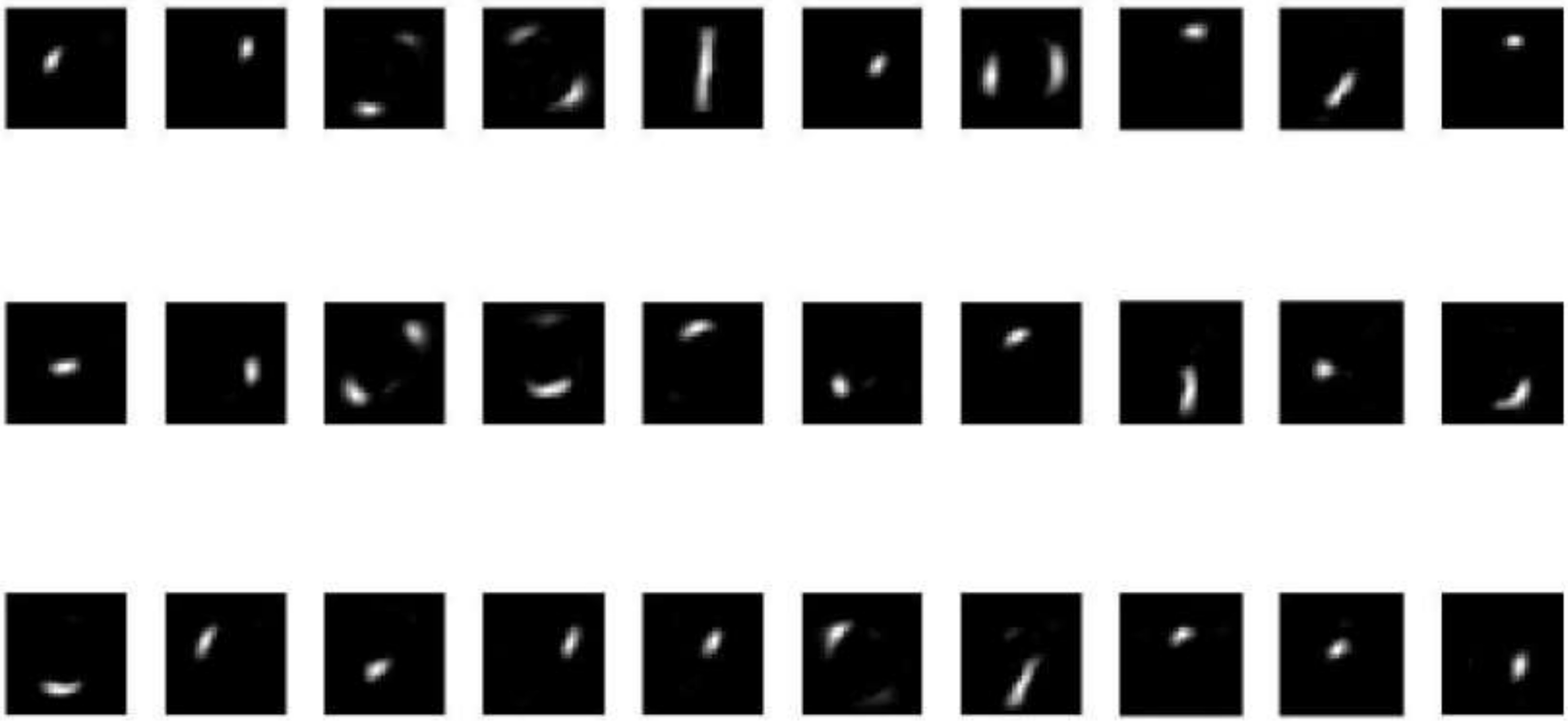
$$= a_1 w^1 + a_2 w^2 + \cdots$$

Can be any real number

- PCA involves adding up and subtracting some components (images)
  - Then the components may not be "parts of digits"
- Non-negative matrix factorization (NMF)
  - Forcing a_1, a_2 ...... be non-negative
    - additive combination
  - Forcing w^1, w^2...... be non-negative
    - More like "parts of digits"
- Ref: Daniel D. Lee and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." *Advances in neural information processing systems*. 2001.

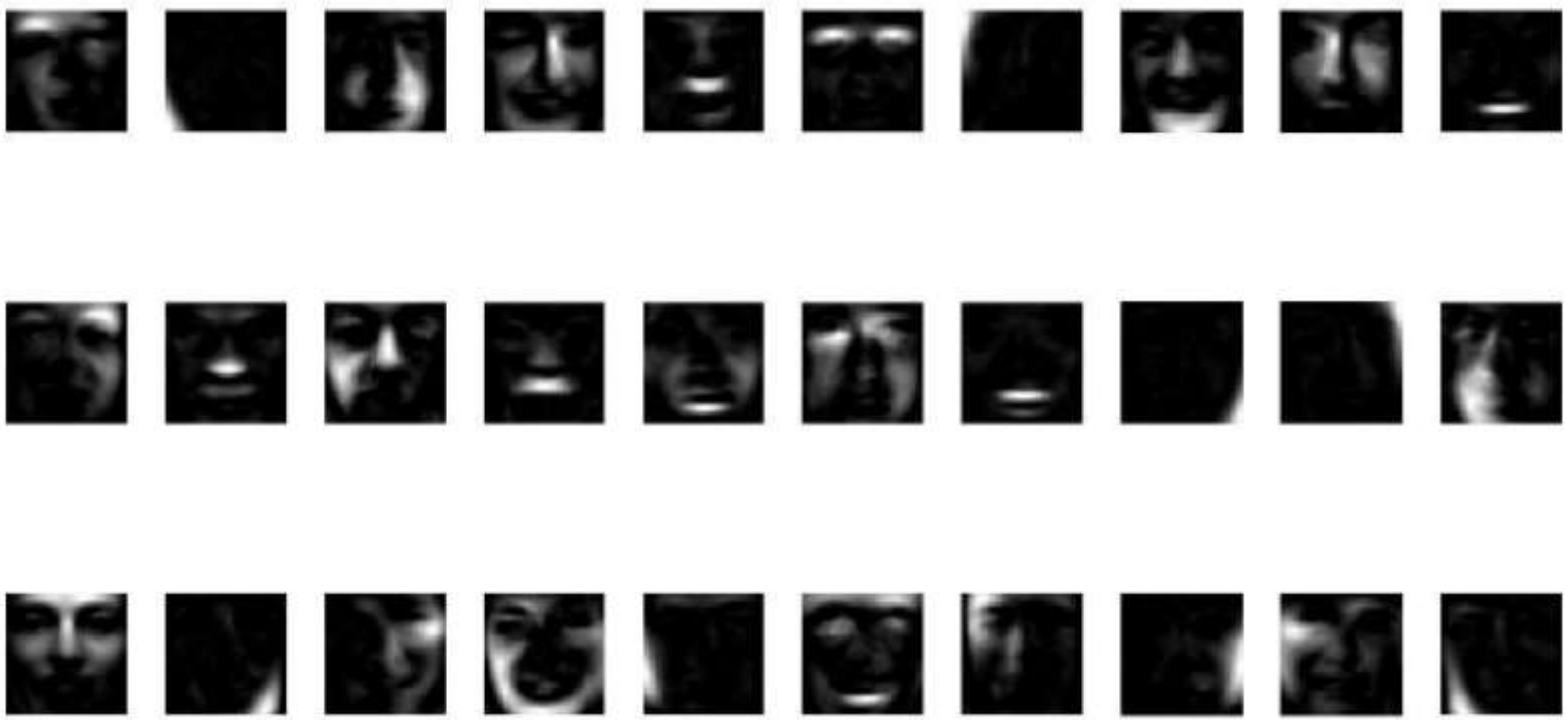
Principal Component Analysis

NMF on MNIST



Principal Component Analysis

NMF on Face



Outline

- Additional remarks on Principal component analysis
- Data Preprocessing

    5) Feature Subset Selection

    6) Feature Generation/Creation

    7) Discretization and Binarization

    8) Attribute Transformation

- Measure of Similarity & Dissimilarity
- What is data exploration?

Feature Subset Selection (FSS)

- PCA maps data into different dimensions which is somewhat hard to explain.
- FSS is another way to reduce dimensionality of data.
- Redundant features
  - Example: **purchase price** of a product /services/dinner and the amount of **sales tax paid**
- Irrelevant features
  - Example: students' ID is often irrelevant to the task of predicting students' GPA

Feature Selection

- Feature Selection is a process that *chooses an optimal subset of features* according to a certain criterion.
- Why we need FS:
  - To reduce dimensionality, noise and complexity
  - To improve performance
  - To visualize the data for model selection
  - To improve the model understandability

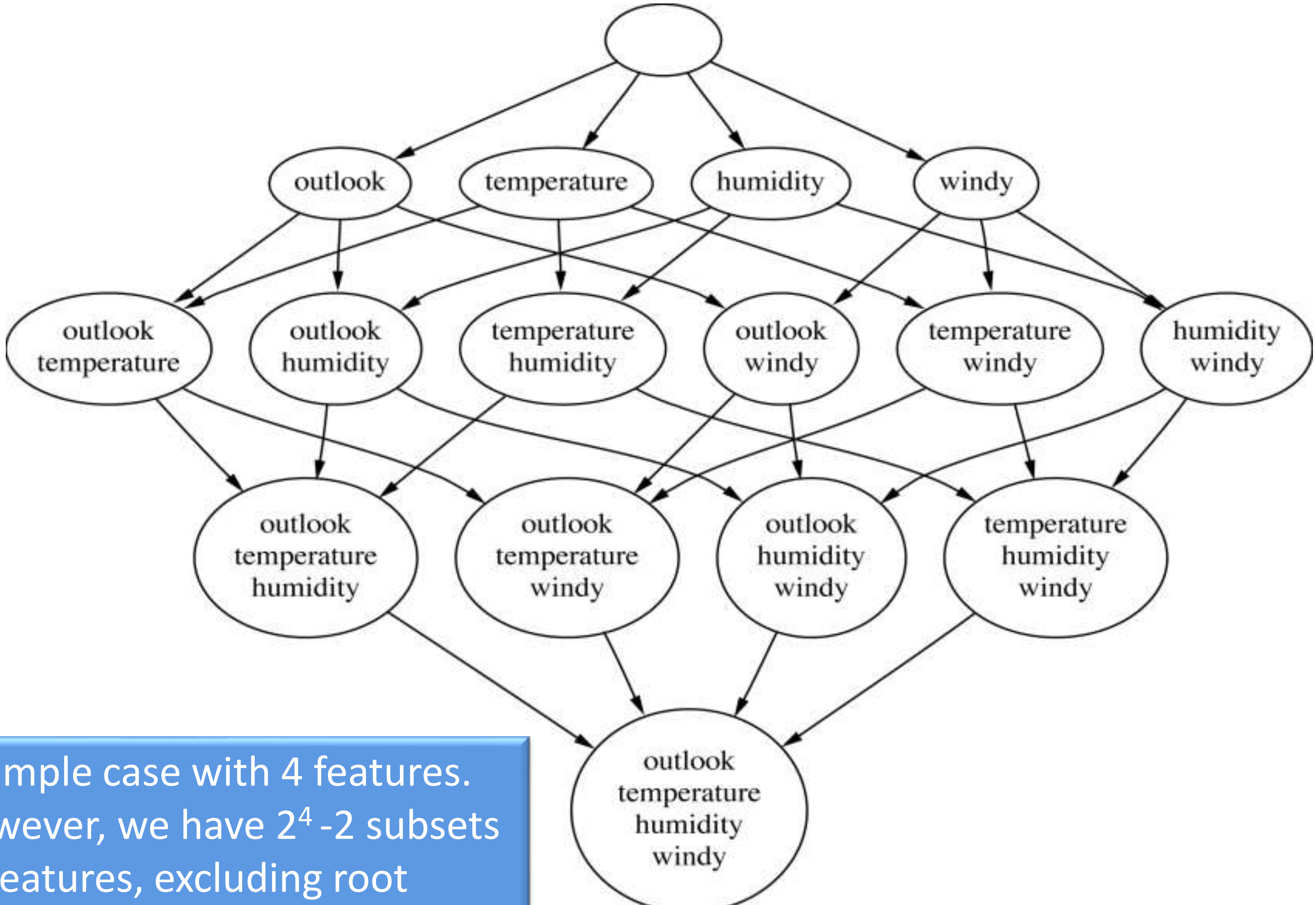
Feature Subset Selection

- Techniques:
  - Brute-force approach:
    - Try all possible feature subsets as input to machine learning algorithm. **Number of features could be huge**!

Weather Data

	outlook	temperature	humidity	windy	play
1	outlook	temperature	humidity	windy	play
2	sunny	hot	high	FALSE	no
3	sunny	hot	high	TRUE	no
4	overcast	hot	high	FALSE	yes
5	rainy	mild	high	FALSE	yes
6	rainy	cool	normal	FALSE	yes

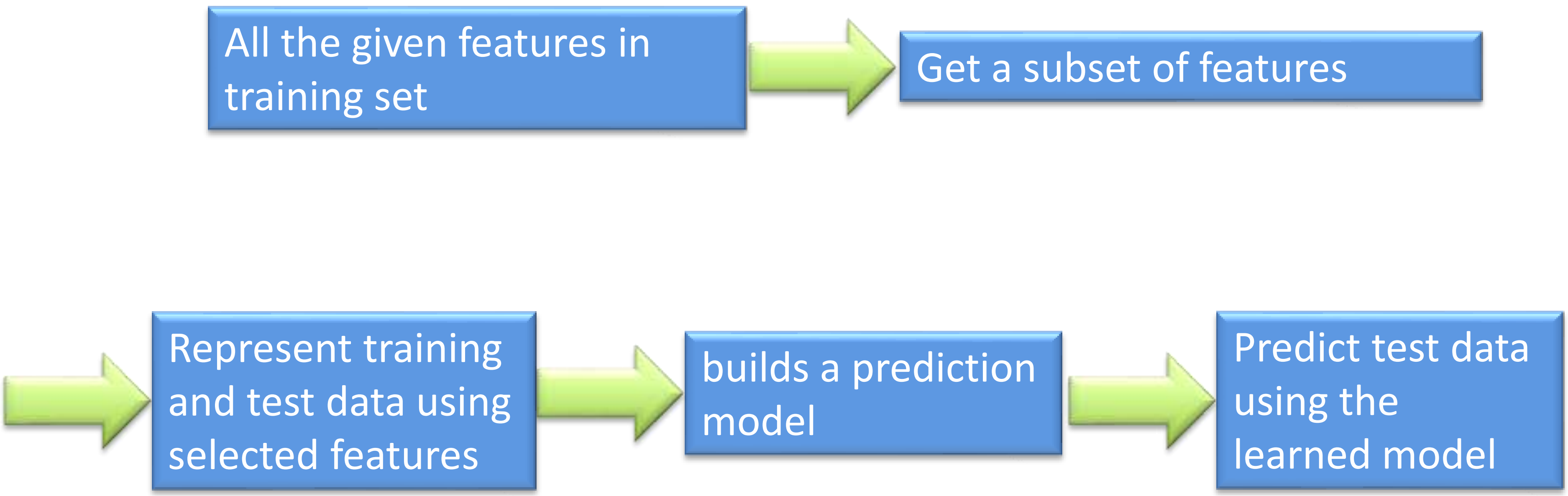
Attribute subsets for weather data



A simple case with 4 features. However, we have 2^4 -2 subsets of features, excluding root (empty set) and leave (full set)

Feature Subset Selection

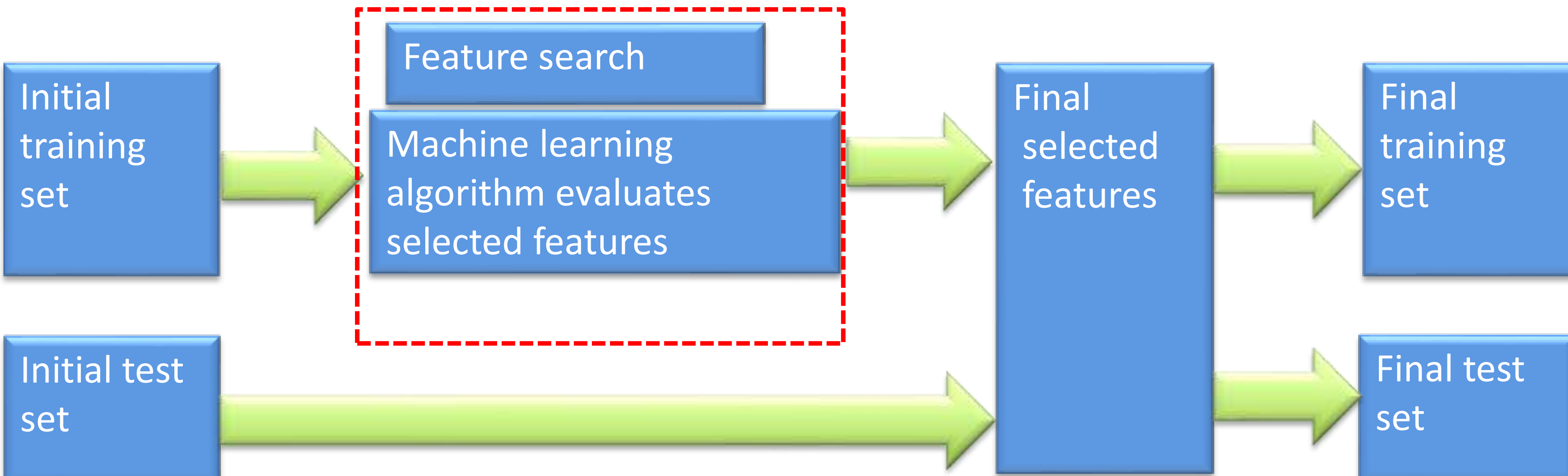
- Techniques:
  - Filter approaches:
    - Features are selected **before** machine learning algorithm is run



Feature Subset Selection

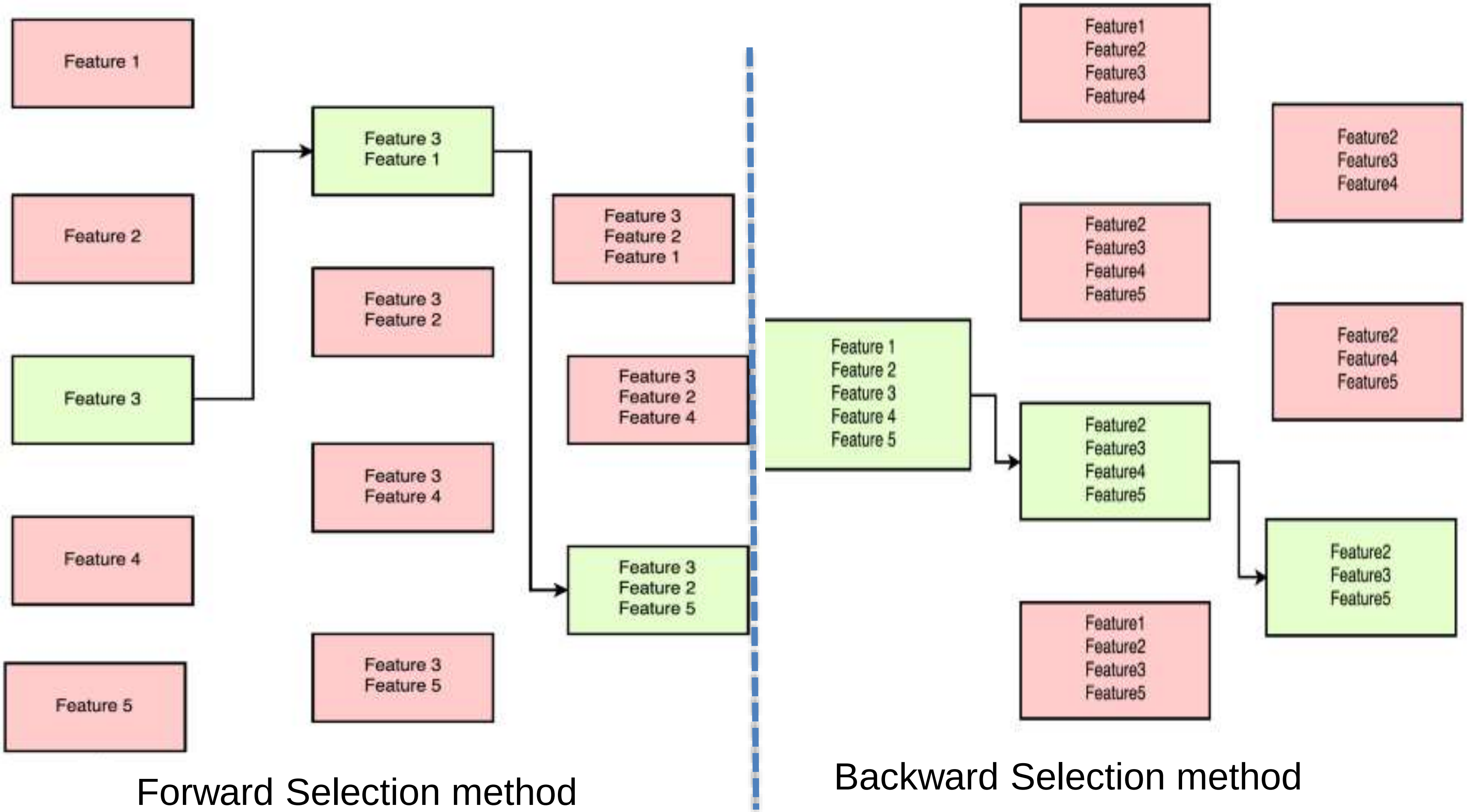
– Embedded approaches:
  - Feature selection occurs **naturally as part** of the machine learning algorithm, e.g. C4.5. We select best features (e.g. using information gain) to build a tree in top-down fashion.

– Wrapper approaches:
  - Use a machine learning algorithm as a **black box** (compute accuracy) to find best subset of attributes



Feature Search

Common greedy approaches



Forward Selection method

Backward Selection method

Two figures are from https://medium.com/@sagar.rawale3/feature-selection-methods-in-machine-learning-eaeef12019cc

One Example of Feature/Signal Selection

- Given a sample space of p dimensions

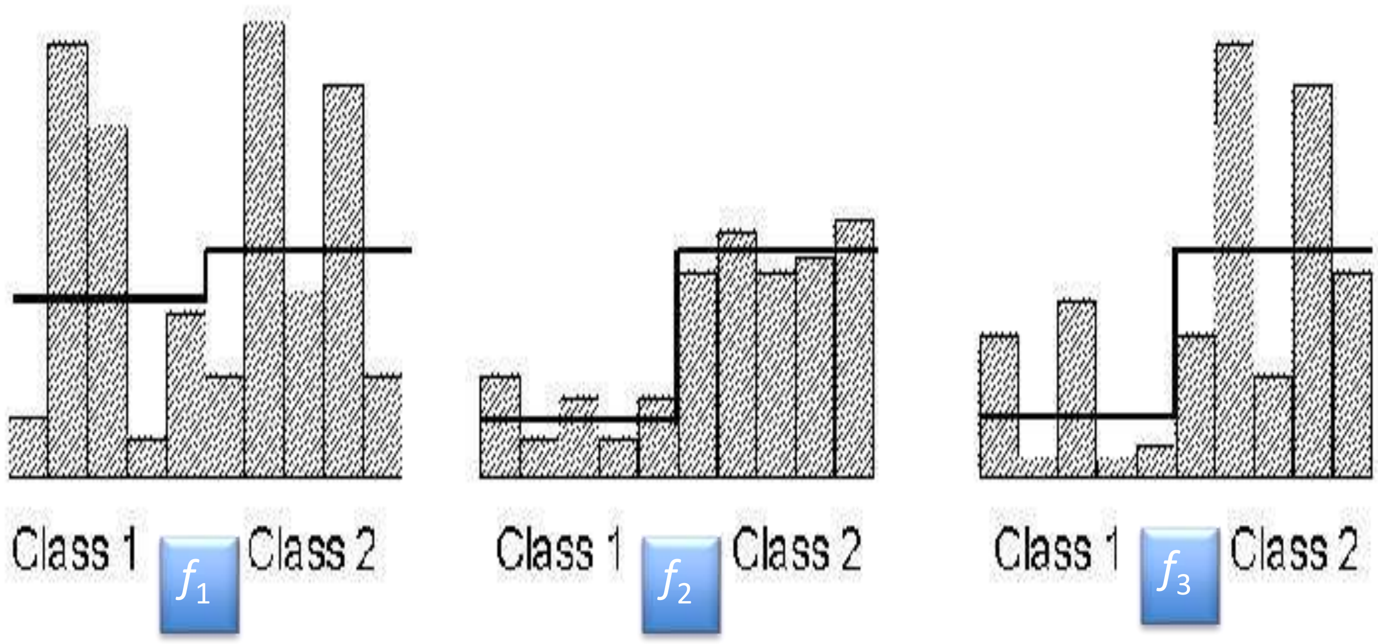
- It is possible that some dimensions are irrelevant or less important.

- Need to find ways to separate those dimensions that are relevant from those that are irrelevant

Signal Selection (Basic Idea)

- Choose a feature with low *intra-class distance* (*variance* is smaller)
- Choose a feature with high *inter-class distance*

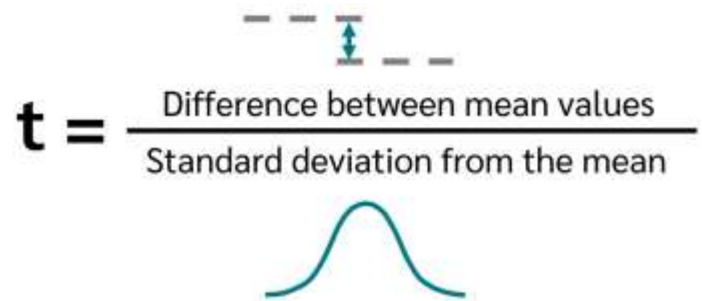
  (*mean* difference is bigger)
- Given features f_1, f_2 and f_3 for binary classification task (Class 1 and 2), which feature is the best?



Signal Selection (*t*-statistics/ *t*-test)

$$t = \frac{\text{signal}}{\text{noise}} = \frac{difference\ b.t.w\ group\ means}{variability\ of\ groups} = \frac{|\overline{X_1} - \overline{X_2}|}{\sqrt{S_1^2/n_1 + S_2^2/n_2}}$$

- t-statistics/t-test can be used for signal selection.
- The term "t-statistic" is abbreviated from "hypothesis test statistic".
- Determine whether there is a statistically significant difference between the means of two groups
- Uses
  - One-sample Student's t-test
  - Independent (unpaired) samples
  - Paired samples



https://en.wikipedia.org/wiki/Student%27s_t-test

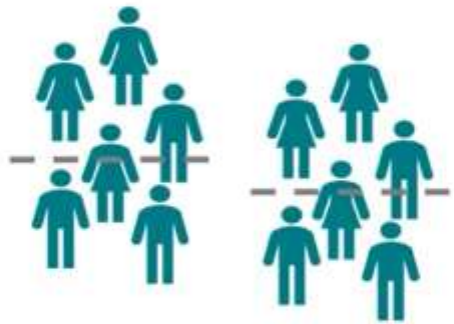
Signal Selection (*t*-statistics/ *t*-test)

1.

One sample
t-test

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

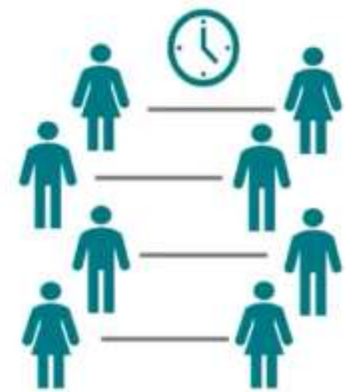
2.



Independent
samples t-test

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

3.



Paired
samples t-test

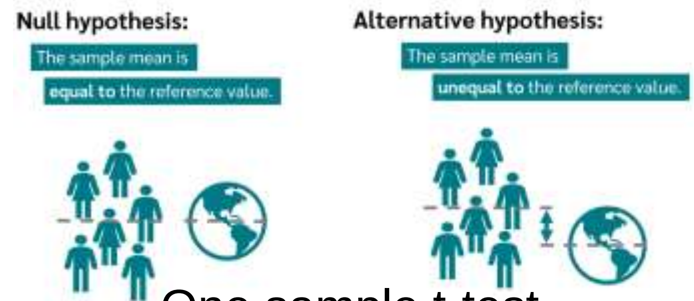
$$t = \frac{\overline{x_d} - 0}{\frac{s}{\sqrt{n}}}$$

[t-Test - Everything you need to know]

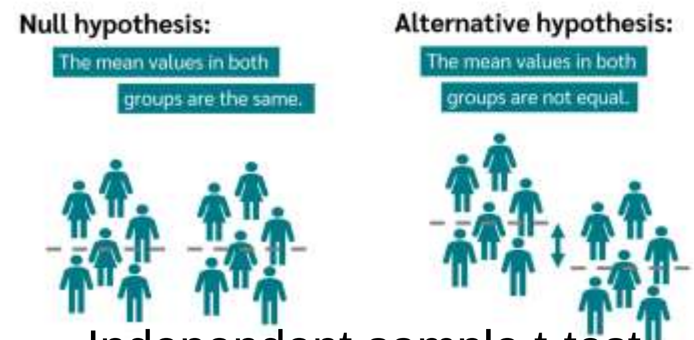
Signal Selection (*t*-statistics/ *t*-test)

- Assumptions for the t-test of two independent samples
  - ✓ The means of the two populations being compared should follow [normal distributions]().
  - ✓ The data used to carry out the test should either be sampled independently from the two populations being compared or be fully paired.
  - ✓ If using Student's original definition of the t-test, the two populations being compared should have the same variance.

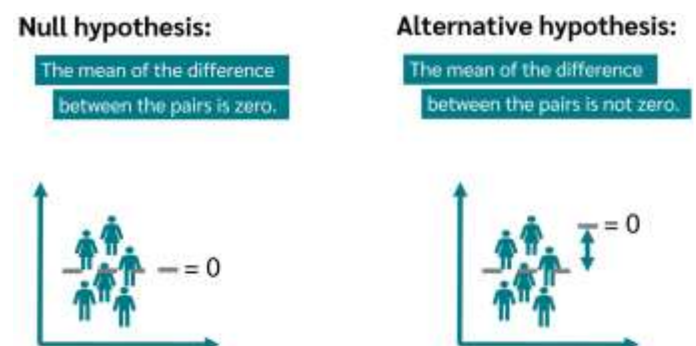
- Hypotheses



One sample t-test



Independent sample t-test



Paired samples t-test

Signal Selection (*t*-statistics/ *t*-test)

Controlled via sample size (=1-Power of test)

Typically restrict to a 5% Risk = level of significance

	Study reports **NO** difference (Do not reject H_0)	Study reports **IS** a difference (Reject H_0)
H_0 is true Difference Does **NOT** exist in population	✓	X Type I Error
H_1 is true Difference **DOES** exist in population	X Type II Error	✓

Prob of this = Power of test

Signal Selection (*t*-statistics/ *t*-test)

How to calculate a t-test?

Independent two-sample *t*-test

Given $X_1 \sim N(\mu_1, \sigma_1^2), X_2 \sim N(\mu_2, \sigma_2^2)$

Collect $\{X_{11}, X_{12}, \cdots, X_{1n_1}\}$ $\{X_{21}, X_{22}, \cdots, X_{2n_2}\}$

Calculate $\bar{X}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1} X_{1i}, \quad \bar{X}_2 = \frac{1}{n_2}\sum_{i=1}^{n_2} X_{2i}$

$$s_1 = \sqrt{\frac{1}{n_1 - 1}\sum_{i=1}^{n_1}(X_{1i} - \bar{X}_1)^2}, \quad s_2 = \sqrt{\frac{1}{n_2 - 1}\sum_{i=1}^{n_2}(X_{2i} - \bar{X}_2)^2}$$

Case 1: for unknown $\sigma^2 = \sigma_1^2 = \sigma_2^2$.

Equal or unequal sample sizes, similar variances

$$\bar{X}_1 - \bar{X}_2 \sim N\left(\mu_1 - \mu_2, (\frac{1}{n_1} + \frac{1}{n_2})\sigma^2\right)$$

$$\rightarrow \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sigma\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim N(0,1)$$

Signal Selection (*t*-statistics/ *t*-test)

$$\frac{(n_1-1)s_1^2}{\sigma^2}+\frac{(n_2-1)s_2^2}{\sigma^2}\sim\chi^2(n_1+n_2-2)$$

$$\frac{(\bar{X}_1-\bar{X}_2)-(\mu_1-\mu_2)}{s_p\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}\sim t(n_1+n_2-2)$$

$$s_p=\sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}=\sqrt{\frac{\sum_{i=1}^{n_1}(X_{1i}-\bar{X}_1)^2+\sum_{i=1}^{n_2}(X_{2i}-\bar{X}_2)^2}{n_1+n_2-2}}$$

n_1+n_2-2: degree of freedom

Case 2: for completely unknown.
Equal or unequal sample sizes, unequal variances

$$t=\frac{\bar{X}_1-\bar{X}_2}{\sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}}$$

Signal Selection (*t*-statistics/ *t*-test)

The t-stats of a signal is defined as

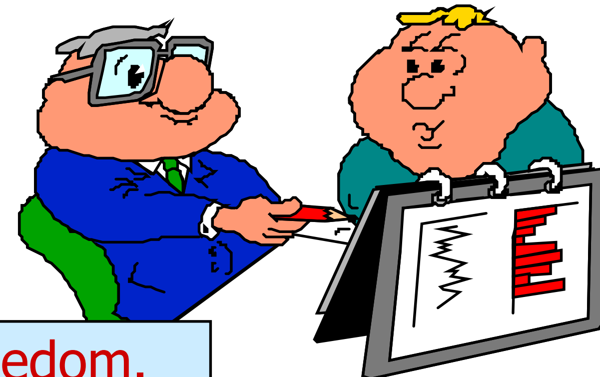
$$t = \frac{|\mu_1 - \mu_2|}{\sqrt{(\sigma_1^2/n_1) + (\sigma_2^2/n_2)}}$$

where σ_i^2 is the variance of that signal in class i, μ_i is the mean of that signal in class i, and n_i is the size of class i.

A feature f_2 can be considered better than a feature f_1 if $t(f_2, C_1, C_2) > t(f_1, C_1, C_2)$. Thus given a collection of candidate features in samples of C_1 and C_2, we simply sort them by their *t*-test statistical measure, and pick those with the largest *t*-test statistical measures.

compute mean and variance http://www.mathsisfun.com/data/standard-deviation.html

Signal Selection (*t*-statistics/ *t*-test)



Formulate H_0 and H_1

↓

Select Appropriate Test

↓

Choose Level of Significance

↓

Calculate Test Statistic. Given degree of freedom.

Determine Prob Assoc with Test Stat (Calculate p-value)	Determine Critical Value of Test Stat (Read the table of t-values)
↓	↓
Compare with Level of Significance, α	Determine if Test Stat falls into (Non) Rejection Region

↓

Reject/Do not Reject H_0

↓

Draw Conclusion

https://en.wikipedia.org/wiki/Student%27s_t-distribution#Table_of_selected_values

Self-fulfilling oracle

a) Construct artificial dataset with 100 samples, each with 100,000 randomly generated features and randomly assigned binary class labels

b) Select 20 features with the t-statistics method (or other methods)

c) Evaluate accuracy by cross validation using the 20 selected features

d) The resulting accuracy can be **~90%.**

e) But the true accuracy should be 50%, as the data were derived randomly.

What went wrong?

- The 20 features were selected from the whole dataset.
- Information in the held-out testing samples has thus been “leaked” to the training process.
- The correct way is to re-select the 20 features at 9 folds (training data) and then to construct test set from the remaining 1 fold by keeping the selected 20 features only.

Outline

- Data Preprocessing

  5) Feature Subset Selection

  6) Feature Generation/Creation

  7) Discretization and Binarization

  8) Attribute Transformation
- Measure of Similarity & Dissimilarity
- What is data exploration?

6). Feature Generation/Creation

- Create new attributes that can capture the important information in a data set much more efficiently/effectively than the original attributes
- Three general methodologies:
  - Feature Extraction (image analysis using deep learning)
  - Mapping Data to New Space (PCA; Fourier transform or Wavelet transform, relatively easy to identify differences, e.g. sensor data analytics – normal, fault classes)
  - Feature Construction (very powerful in practice -winning formula for many competitions), e.g.
    - Combining features (Insurance: |agent age-customer age|), ratio of total cholesterol/LDL [heart disease prediction]
    - #transactions per hour (total, mean, average, sd), #home country (1 or 0)? [fraud detection]
    - Width*Length [property price prediction]

Feature creation and selection

Predict which passengers survived the tragedy

Feature generation: generate new features from the original raw feature, e.g.
Title,
Family Size,
Gender
……

1	Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. Owen Har	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mrs. John Bra	female	38	1	0	PC 17599	71.28	C85	C
4	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2.	7.925		S
5	4	1	1	Futrelle, Mrs. Jacques H	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. William Henr	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr. James	male		0	0	330877	8.458		Q
8	7	0	1	McCarthy, Mr. Timothy	male	54	0	0	17463	51.86	E46	S
9	8	0	3	Palsson, Master. Gosta	male	2	3	1	349909	21.08		S
10	9	1	3	Johnson, Mrs. Oscar W	female	27	0	2	347742	11.13		S

9 existing features

Sibsp: #sibling or spouses; **Parch:** # parents or children on board

Outline

- Data Preprocessing

  5) Feature Subset Selection

  6) Feature Generation/Creation

  7) Discretization and Binarization

  8) Attribute Transformation

- Measure of Similarity & Dissimilarity

- What is data exploration?

7). Discretization and Binarization

- Some data mining algorithms, e.g. association rule mining, require the data to be in the form of categorical attributes or binary attributes
- Discretization (attribute-type change)
  - Transform a continuous attribute into a categorical attribute (e.g. age, blood pressure)
- Binarization (1 to multiple)
  - Transform either a continuous attribute or a categorical attribute into one or more binary attributes

Discretization of Continuous Attribute

- **Transformation of a continuous attribute to a categorical attribute involve two subtasks**
  - (1) **Decide how many *categories***
    - After the values are sorted, they are then divided into ***n*** intervals by specifying ***n*-1 split points**.
  - (2) **Determine how to map the values of the continuous attribute to these *categories***
    - All the values in one interval are mapped to the *same* categorical value
  - The key issue is **how many split points** to choose and **where to place them** [equal intervals, equal frequency]
- **Discretization methods**
  - Unsupervised vs. Supervised discretization (need training data)

Binarization

- Given *m* categorical values, assign each original value to an integer in the interval [0, *m*-1].
- Convert each of these *m* integers into a binary number
- Require *n* = ceiling(log2(*m*)) binary digits to represent these integers

Binarization

Example:

A categorical variable with 5 values {Awful, Poor, OK, Good, Great}, require **3 binary attributes (**x1, x2, x3) , i.e. log2(5)=2.3->3 (ceiling)

Categorical Value	Integer value	X1	X2	X3
Awful	**0**	0	0	0
Poor	**1**	0	0	1
OK	**2**	0	1	0
Good	**3**	0	1	1
Great	**4**	1	0	0

One hot encoding

- Generate one Boolean column for each *category*. The number of the *categories* will be the number of columns.
- Only one of these columns could take on the value 1 for each sample in your training/test data. Hence, the term one hot encoding. The main drawback is its size could be very big when we handle features when many *categories.*
- It is used in NLP to represent document, and some classification models, such as Xgboost (which only accept numeric values)

Categorical Value	Integer value	X1	X2	X3	X4	X5
Awful	**0**	1	0	0	0	0
Poor	**1**	0	1	0	0	0
OK	**2**	0	0	1	0	0
Good	**3**	0	0	0	1	0
Great	**4**	0	0	0	0	1

https://www.quora.com/What-is-one-hot-encoding-and-when-is-it-used-in-data-science

https://www.quora.com/What-are-good-ways-to-handle-discrete-and-continuous-inputs-together

37

When we need binarization or one hot encoding

- This works very well with most machine learning algorithms that need *all* the features/attributes are continuous.
- Some algorithms, like random forests, association rule mining, can handle categorical features natively. Then, binarization and one hot encoding are not necessary. However, other algorithms do need this preprocessing step to change attribute types into numeric/continuous before we can build machine learning models (NN, Xgboost etc) .

Outline

- Additional remarks on Principal component analysis
- Data Preprocessing

  5) Feature Subset Selection

  6) Feature Generation/Creation

  7) Discretization and Binarization

  8) Attribute Transformation
- Measure of Similarity & Dissimilarity
- What is data exploration?

Attribute/Variable Transformation

- **A function** that maps the *entire set of values* of a given attribute to **a new set** of replacement values via certain math functions (an original value as input to generate a new value)
- Simple math functions: v^k, $\log(v)$, e^v, $|v|$, $1/v$, $\sin v$
  - Could be scale down/up
  - Normalization (or Standardization)

Normalization (frequently used)

- Min-max normalization:
  - $[min_A, max_A]$ - - -> $[new\_min_A, new\_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - Example:

    Annual income range [12,000, 300,000] normalized to [0.0, 1.0]. Then 73,000 is mapped to

$$\frac{\boxed{73{,}000} - 12{,}000}{300{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.21$$

$$\frac{\boxed{12{,}000} - 12{,}000}{300{,}000 - 12{,}000}(1.0 - 0) + 0 = 0 \qquad \frac{\boxed{300{,}000} - 12{,}000}{300{,}000 - 12{,}000}(1.0 - 0) + 0 = 1$$

Normalization (cont)

- Z-score normalization

  (μ_A: mean, σ_A: standard deviation): $v' = \dfrac{v - \mu_A}{\sigma_A}$

  – Example: Consider a value v=73,000,

  – Let μ_A = 54,000, σ_A= 16,000. Then $\dfrac{73{,}000 - 54{,}000}{16{,}000} = 1.225$

- Normalization by Decimal Scaling

$$v' = \frac{v}{10^j}$$

here *j* is the smallest integer such that Max(|v'|) < =1

1, 10, 100, 1000 ->1/10^3, 10/10^3 , 100/10^3 , 1000/10^3 (Here j=3; If we use j=4, then it will not be the smallest integer)

Quantile normalization in statistics

- QN is a technique for making two distributions identical in statistical properties (apple to apple)
- To quantile normalize two or more distributions to each other, we sort, then set to the average of the distributions.
  - The highest value in all cases becomes the mean of the highest values; the second highest value becomes the mean of the second highest values, and so on.
- Quantile normalization is frequently used in microarray data analysis in computational biology or bioinformatics

How to perform quantize normalization?

Array/Variable 1, 2, …, *n*

Observations/ Genes 1, 2, …, *p*

	1	**2**	**…**	***n***
1	0.8	0.7		
2				
3				
…..				
P				

Sort each column to give X_{sort}

Take means across rows of X_{sort} *and assign this* mean to each element in the row to get X'_{sort}

Get $X_{normalized}$ *by arranging each column of* X'_{sort} to have same ordering as X

Exercise

- http://en.wikipedia.org/wiki/Quantile_normalization
- Arrays 1 to 3, genes A to D

	Array 1	Array 2	Array 3
A	5	4	3
B	2	1	4
C	3	4	6
D	4	2	8

How to perform quantile normalization?

Rank->Average-> Replace (same order)

Quantile normalization (rank array)

- Arrays 1 to 3, genes A to D

	Array 1	Array 2	Array 3
A	5	4	3
B	2	1	4
C	3	4	6
D	4	2	8

- For each column determine a rank from lowest to highest and assign number i-iv

A	iv	iii	i
B	i	i	ii
C	ii	iii	iii
D	iii	ii	iv

These rank values are set aside to use later. We will convert the ranks into actual values.

Quantile normalization

(<u>Average</u> genes' rank values across array)

- Go back to the first set of data. Rearrange that first set of column values so each column is in order going lowest to highest value. The result is:

A	5	4	3
B	2	1	4
C	3	4	6
D	4	2	8

Smallest Values

A	2	1	3
B	3	2	4
C	4	4	6
D	5	4	8

Largest Values

- Now find the mean for each row to determine the ranks

A (2 1 3)/3 = 2.00 = rank i

B (3 2 4)/3 = 3.00 = rank ii

C (4 4 6)/3 = 4.67 = rank iii

D (5 4 8)/3 = 5.67 = rank iv

Quantile Normalization (explanation)

- Go back to the first set of data. Rearrange that first set of column values so each column is in order going lowest to highest value. The result is:

A	5	4	3
B	2	1	4
C	3	4	6
D	4	2	8

→

A	2	1	3
B	3	2	4
C	4	4	6
D	5	4	8

- Now find the mean for each row to determine the ranks

A (2 1 3)/3 = 2.00 = rank i — Average of the smallest

B (3 2 4)/3 = 3.00 = rank ii — Average of the second smallest

C (4 4 6)/3 = 4.67 = rank iii — Average of the second largest

D (5 4 8)/3 = 5.67 = rank iv — Average of the largest

Quantile Normalization (Replace)

2.00 = rank i, 3.00 = rank ii , 4.67 = rank iii , 5.67 = rank iv

- Now take the ranking order and substitute in new values

A	iv	iii	i
B	i	i	ii
C	ii	iii	iii
D	iii	ii	iv

A	5.67	4.67	2.00
B	2.00	2.00	3.00
C	3.00	4.67	4.67
D	4.67	3.00	5.67

Original Data

A	5	4	3
B	2	1	4
C	3	4	6
D	4	2	8

Outline

- Additional remarks on Principal component analysis
- Data Preprocessing

  5) Feature Subset Selection

  6) Feature Generation/Creation

  7) Discretization and Binarization

  8) Attribute Transformation
- Measure of Similarity & Dissimilarity
- What is data exploration?

Measure of Similarity & Dissimilarity

- Similarity and dissimilarity/distance are important and fundermental as they are used by many data mining techniques.
- In some cases, the initial data set is not needed once these similarities or dissimilarities/distances have been computed.
- For convenience, the term "**proximity**" is used to refer to either similarity or dissimilarity/distance.

Similarity and Dissimilarity

- Similarity
  - Numerical measure of how **alike** two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- Dissimilarity
  - Numerical measure of how **different** are two data objects
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0 (e.g. same objects)
  - Upper limit varies

Similarity/Dissimilarity for Simple Attributes

- Similarity/Dissimilarity between p and q. p and q are the attribute **values** for two data objects (use *single feature* value for illustration)
- Object 1: p (e.g. p=male, p=young, or p=23)
- Object 2: q (e.g. q=female, p=old, or p=40)

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Similarity and dissimilarity for simple attributes

Common Properties of a Similarity

- Similarities have some well-known properties.
- Let us denote by $s(p, q)$ the similarity between two data objects (points) p and q.

1. Self-Similarity

s(p, q) = 1 (or maximum similarity) only if $p = q$.

2. Symmetry

$s(p, q) = s(q, p)$ for all p and q.

Similarity does not necessarily preserve the triangle inequality, like distance.

Similarity Between **Binary Vectors**

could be n-dimensional vectors

- Consider two objects, p and q, having only binary attributes

  $p = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$
  $q = 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$

- Compute similarities using the following quantities
  M_{01} = the number of attributes where p was 0 and q was 1
  M_{10} = the number of attributes where p was 1 and q was 0
  M_{00} = the number of attributes where p was 0 and q was 0
  M_{11} = the number of attributes where p was 1 and q was 1

- **Simple Matching Coefficient (SMC)**
  SMC = number of matches / number of attributes
  $= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

- **Jaccard Coefficient (J)**
  J = number of 11 matches / number of not-both-zero attributes values
  $= M_{11} / (M_{01} + M_{10} + M_{11})$

SMC versus Jaccard: Example

$p =$ 1 0 0 0 0 0 0 0 0 0
$q =$ 0 0 0 0 0 0 1 0 0 1

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)
$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)
$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)
$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

SMC $= (M_{11} + M_{00})/(M_{01} + M_{10} + M_{11} + M_{00}) = (0+7) / (2+1+0+7) = 0.7$

J $= (M_{11}) / (M_{01} + M_{10} + M_{11}) = 0 / (2 + 1 + 0) = 0$

In what cases, SMC or Jaccard similarity is useful?

Cosine Similarity

- If d_1 and d_2 are two vectors (e.g. document vectors), then

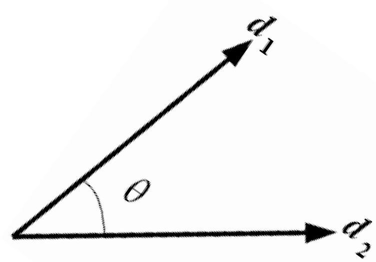
$$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1||\ ||d_2||$$

where $\bullet$ indicates vector dot product and $|| d ||$ is the length of vector d.

- It is a measure of the *cosine* of the angle between the two vectors.
- Example:

d_1 = **3 2 0 5 0 0 0 2 0 0**

d_2 = **1 0 0 0 0 0 0 1 0 2**



$d_1 \bullet d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$

$||d_1|| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.4807$

$||d_2|| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5} = (6)^{0.5} = 2.4495$

$$\cos(d_1, d_2) = 5/(6.4807*2.4495) = 0.3150$$

Questions: Does the cosine similarity depend on the number of shared 0 values (0-0 matches) between two vectors?

Euclidean Distance

- Euclidean Distance between two n-dimensional vectors (objects) $\boldsymbol{p}$ and $\boldsymbol{q}$

$$\boldsymbol{dist} = \sqrt{\sum_{\boldsymbol{k}=1}^{\boldsymbol{n}} (\boldsymbol{p_k} - \boldsymbol{q_k})^2}$$

- where $\boldsymbol{p}=\{p_1, p_2, \ldots, p_k, \ldots, p_n\}$,
- $\boldsymbol{q}=\{q_1, q_2, \ldots, q_k, \ldots, q_n\}$.
- n is the number of dimensions (attributes) and p_k and q_k are the k^{th} attributes of data objects ***p*** and ***q***, respectively.
- Feature normalization is usually necessary if scales are different.

Scaling issues

- Attributes may have to be scaled or normalized to prevent distance measures from being dominated by one of the attributes.
- Example:
  - F1: height of a person may vary from 1.2m to 2.4m
  - F2: weight of a person may vary from 35kg to 442kg
  - F3: Annual income of a person may vary from 10K to 50,000K

$p = (p_1\, p_2\, p_3) = (1.64,\ 48,\ 6000)$

$q = (q_1\, q_2\, q_3) = (1.82,\ 75,\ 10000)$

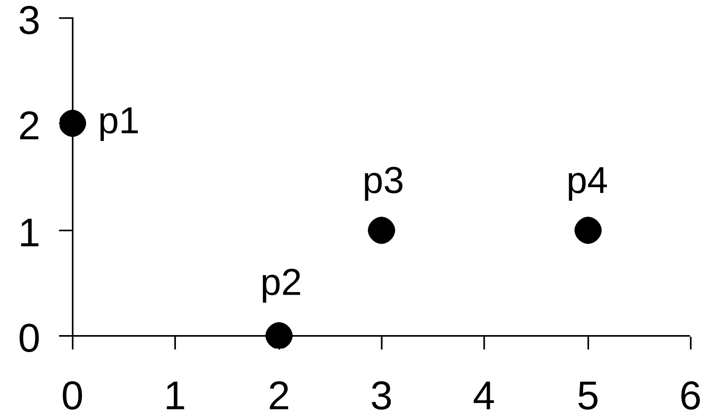
F3 dominates the calculation of Euclidean

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2} = \sqrt{(1.65-1.82)^2 + (48-75)^2 + (6000-10000)^2}$$

Euclidean Distance in 2D

- Example:

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1



	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Euclidean Distance Matrix

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

Where r is a parameter, n is the number of dimensions (attributes) and p_k and q_k are the k-th attributes (components) of data objects p and q respectively.

Minkowski Distance: Special Cases

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

(applied to any vectors)

- $r = 1$:

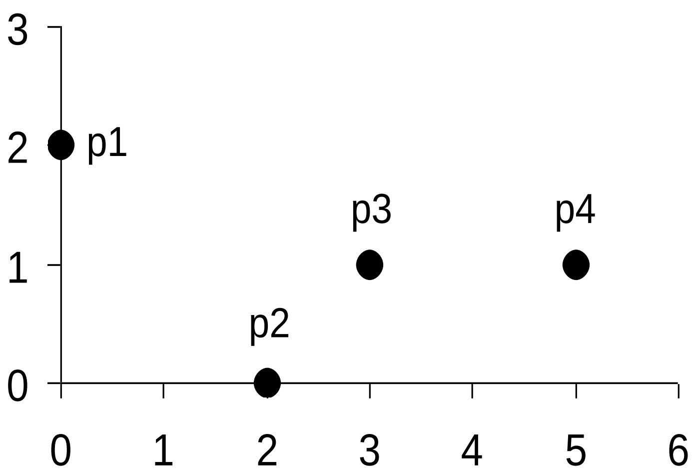
  City block (Manhattan, taxicab, **L_1 norm**) distance.

  – A common example of this is the **Hamming distance**, which is just the number of bits that are different between two binary vectors (**Hamming distance** is only applied to binary vectors)

- $r = 2$:

  Euclidean distance (**L_2 norm**)

point	x	y
p1	0	2
p2	2	0



L_1 norm: dist (p1,p2)=|0-2|+|2-0| = 4

L_2 norm:

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Minkowski Distance: Special Cases

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

- $r = 1$:

  City block (Manhattan, taxicab, **L_1 norm**) distance.
- $r = 2$:

  Euclidean distance (**L_2 norm**)
- $r \rightarrow \infty$:

  "supremum" (**L_{max} norm**, L_∞ norm) distance.

  – The **maximum difference** between any component of the two vectors: $\max(|p_1 - q_1|, \ldots, |p_n - q_n|)$

Do not confuse parameter *r* with dimensionality *n*, i.e., all these distances are defined for all the dimensions.

Minkowski Distance

Distance Matrix

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

City block

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Euclidean

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Supremum

An Example

Distance between P1 and P3

- **r=1, L_1 norm, City block distance**

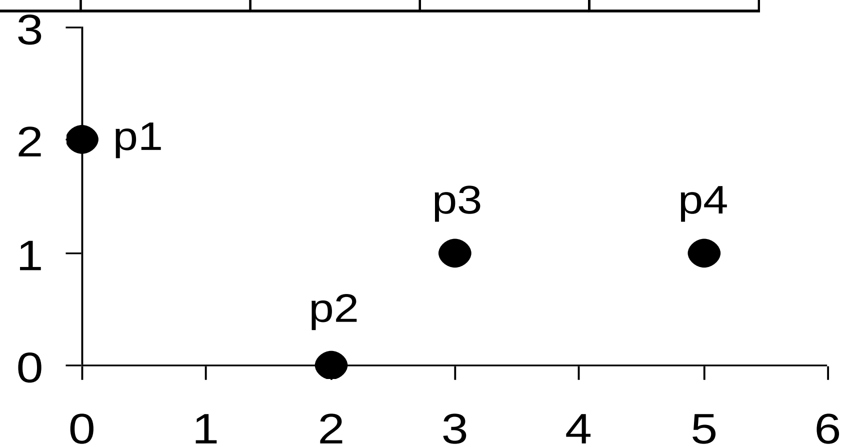
|0-3|+|2-1|=4

- **r=2, L_2 norm, Euclidean distance**

$$\sqrt{(0-3)^2 + (2-1)^2} = \sqrt{10} = 3.162$$

- **$r \to \infty$, L_∞ norm, supremum distance**

Max(|0-3|,|2-1|) =Max (3,1) =3



Common Properties of a Distance

Distances, such as the Euclidean distance, have some well known properties. Let us denote by ***d*(*p*, *q*)** is the distance (dissimilarity) between points (data objects) ***p*** and **q**.

1. **Positive Definiteness**

   $d(p, q) \geq 0$ for all p and q

   $d(p, q) = 0$ if only if $p = q$.

2. **Symmetry**

   $d(p, q) = d(q, p)$ for all p and q

3. **Triangle Inequality**

   $\mathrm{d}(p, r) \leq d(p, q) + d(q, r)$ for all points p, q, and r.

A distance satisfying all the above three properties is a metric.

Correlation

- In statistics, the **Pearson correlation coefficient** (typically denoted by *r*) is a measure of the correlation (linear dependence) between two variables *X* and *Y*.
- The values of r are between +1 and −1 inclusive.
- It is widely used in the sciences as a measure of the strength of linear dependence between two variables.

Formula - Pearson's correlation coefficient

- Pearson's correlation coefficient between two variables is defined as the covariance of the two variables divided by the product of their standard deviations:

$$r = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y},$$

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

Easy to compute

Example: Visually Evaluating Correlation

Scatter plots showing the correlation from –1 to 1.

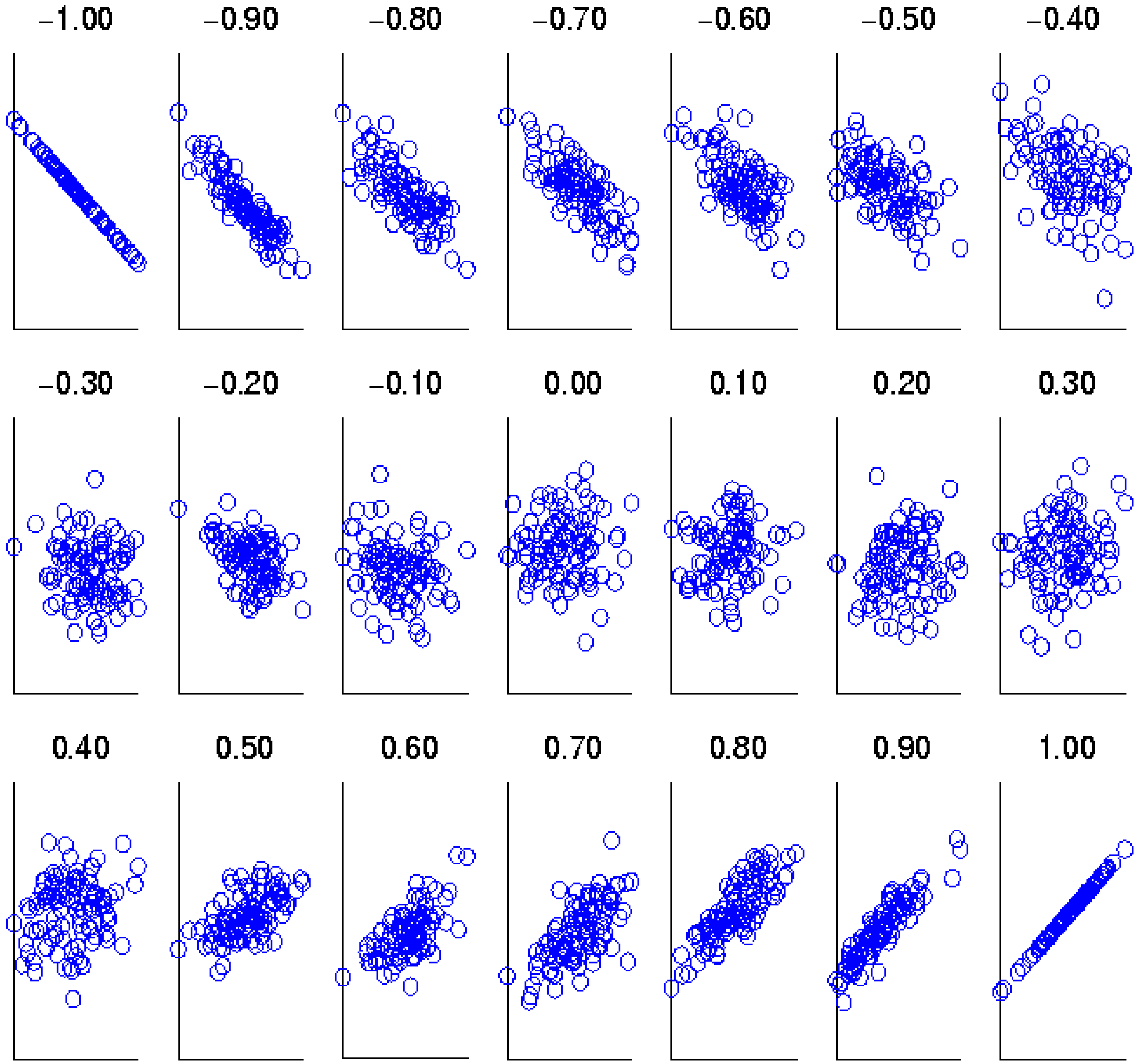


Figure 5.11. Scatter plots illustrating correlations from -1 to 1.

Example of Correlation

- Perfect Correlation
  - Correlation is always in the range -1 and 1.
    A correlation of value 1 (-1) means that p and q have a perfect positive (negative) linear relationship, i.e.,
    y = a* **x** + b, where a and b are constants.
  - The follow two sets of **x** and **y** indicate two cases of correlation -1 and +1, respectively.

x=(-3, 6, 0, 3, -6)	**x**= (3, 6, 0, 3, 6)
y = (1, -2, 0, -1, 2)	**y**=(1, 2, 0, 1, 2)
corr(**x**, **y**) = -1	corr(**x**, **y**) = 1
y=-1/3*x	**y=1/3*x**

General Approach for Combining Similarities

Given two vectors: p={p_1 , p_2 , p_k , …, p_n },
q={q_1 , q_2 , q_k , …, q_n }.

- Sometimes attributes are of many different types, but an **overall** similarity is needed.
- The following approach computes similarities of ***heterogeneous objects*** (with different types of attributes)
  1. For the *k*-th attribute, compute a similarity s_k
  2. Define an indicator variable $\boldsymbol{\delta_k}$ for the *k*-th attribute as follows

$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have} \\ & \text{a value of 0, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$

3. Compute the overall similarity between the two objects using the following formula:

$$similarity(\mathbf{p}, \mathbf{q}) = \frac{\sum_{k=1}^{n} \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

δ_k: Should we consider *k*-th attribute?
We will skip missing values.
We will skip all zero attributes

Using Weights to Combine Similarities

Given two vectors: p={p_1 , p_2 , p_k , …, p_n },
q={q_1 , q_2 , q_k , …, q_n }.

- May not want to treat all attributes the same.
  - Use weights w_k which are between 0 and 1 and sum to 1.

$$similarity(p,q) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

$$\sum_{k=1}^{n} w_k = 1$$

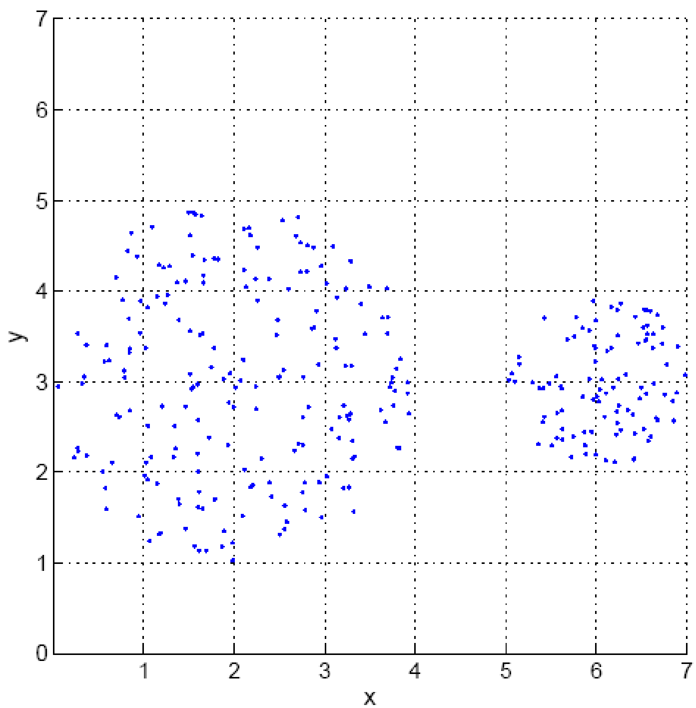
$$distance(p,q) = \left(\sum_{k=1}^{n} w_k |p_k - q_k|^r \right)^{1/r}.$$

Density

- Density-based **clustering** requires a notion of **density.**
- Examples:
  - **Euclidean density**
    - Euclidean density = number of points per unit volume
  - **Probability density**
    - Distribution measures such as covariance
  - **Graph-based density**
    - #internal links
    - #external links

Euclidean Density – Cell-based

- Simple approach
  - Divide region into a number of rectangular cells of equal volume
  - Define density as # of points the cell contains



Cell-based density.

0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

Point counts for each grid cell.

Euclidean Density – Center-based

- Euclidean density is the number of points within a specified radius of the point.

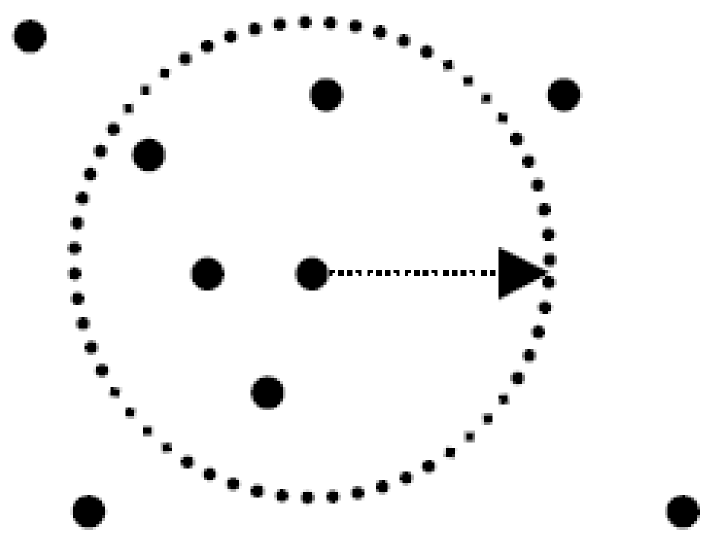


Illustration of center-based density.

Outline

- Additional remarks on Principal component analysis
- Data Preprocessing

  5) Feature Subset Selection

  6) Feature Generation/Creation

  7) Discretization and Binarization

  8) Attribute Transformation
- Measure of Similarity & Dissimilarity
- What is data exploration?

What is data exploration?

- **A preliminary exploration of the data to better understand its characteristics.**
- **In our discussion of data exploration, we focus on**
  - Summary statistics
    - Summarize the properties of the data
  - Visualization
    - Making use of humans' abilities to recognize patterns

Example of a data: Iris Flower Data Set

- Many of the exploratory data techniques are illustrated with the famous ***Iris Flower*** data set (a.k.a. ``**Iris**”).
  - Available at the UCI Machine Learning Repository http://www.ics.uci.edu/~mlearn/MLRepository.html
  - From the statistician R.A. Fisher
  - Three flower types (classes):
    - Iris Setosa
    - Iris Versicolour
    - Iris Virginica
  - Four (non-class) attributes
    - Sepal width
    - Sepal length
    - Petal width
    - Petal length
  - Total number Instances = 150


Iris setosa


Iris versicolor


Iris virginica

1. Summary Statistics

- Summary statistics are numbers that summarize properties of the data.
  - Summarized properties include
    - ***Frequency***, ***location***, and ***spread***
    - Examples: Location – mean / median
      Spread – standard deviation
  - Most summary statistics can be calculated in a single pass through the data.

Frequency and Mode

- The ***frequency*** of an attribute value is the ***percentage of*** **time** the value occurs in the data set.
  - For example, given the attribute “gender” and a representative population of people, the gender “female” occurs about 50% of the time (but this could be changed in different locations, age groups)
- The ***mode*** of an attribute is the most **frequent attribute *value.***
- The notions of ***frequency*** and ***mode*** are typically used with categorical data.

Measures of Location: Mean and Median

- Suppose I have data x_1 x_2, …, x_m
- The *mean* is the most common measure of the location of a set of points.

$$\mathrm{mean}(x) = \overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

- However, the *mean* is very sensitive to outliers.
- Thus, the *median* or a *trimmed* mean is also used:

$$\mathrm{median}(x) = \begin{cases} x_{(r+1)} & \text{if } m \text{ is odd, i.e., } m = 2r+1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{if } m \text{ is even, i.e., } m = 2r \end{cases}$$

Measures of Spread: Range and Variance

- *Range* is the difference between the **max** and **min.**
- The *variance* or *standard deviation* is the most common measure of the **spread** of a set of points.

$$\text{variance}(x) = s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x})^2$$

- However, this is also sensitive to outliers, so that other measures are often used.

Average absolute deviation

$$\text{AAD}(x) = \frac{1}{m}\sum_{i=1}^{m}|x_i - \overline{x}|$$

Median absolute deviation

$$\text{MAD}(x) = median\Big(\{|x_1 - \overline{x}|, \ldots, |x_m - \overline{x}|\}\Big)$$

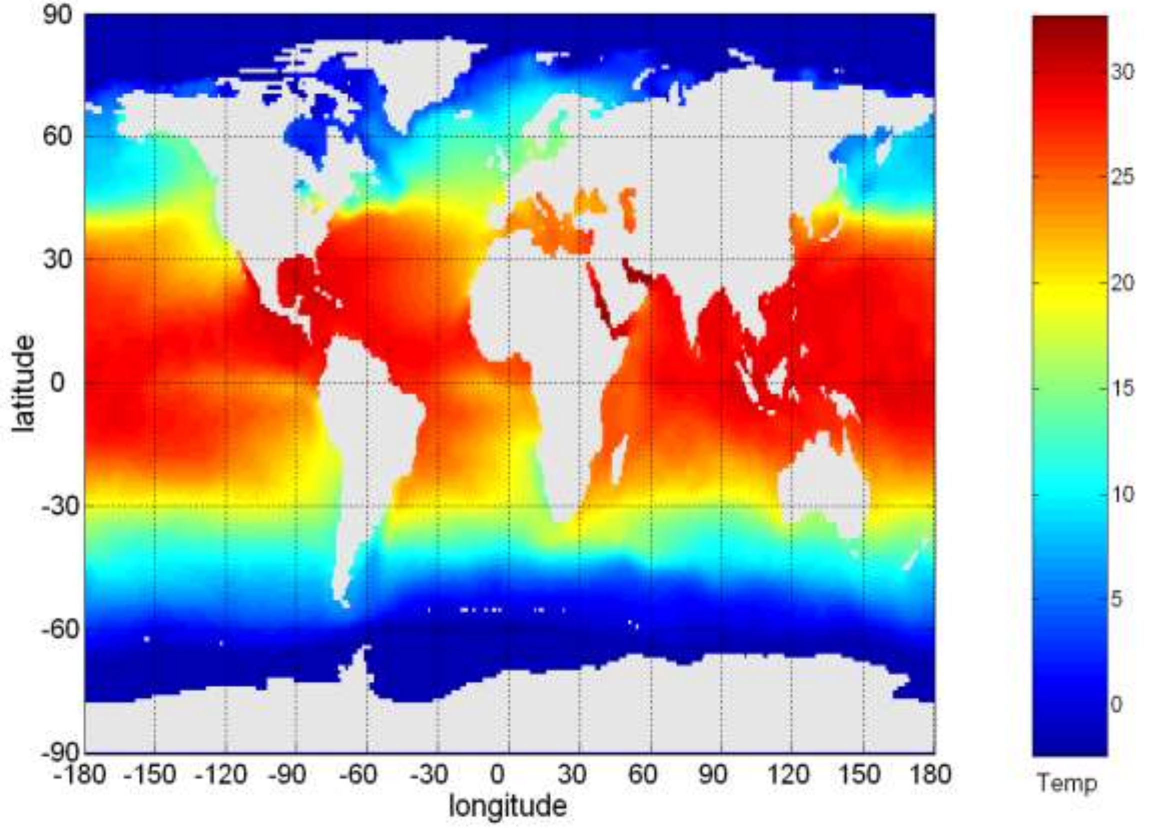
IQR $\text{interquartile range}(x) = x_{75\%} - x_{25\%}$

2 Visualization

- **Visualization** is the conversion of data into a visual or tabular **format** so that the characteristics of the data and the relationships among data items or attributes can be analyzed or reported.
- Visualization of data is one of the most powerful and appealing techniques for data exploration.
  - Humans have a well developed ability to analyze large amounts of information that is presented visually.
  - Can detect general patterns and trends.
  - Can detect outliers and unusual patterns.

Example: Sea Surface Temperature Data->picture->story

- Below shows the Sea Surface Temperature (SST) for July 1982. Tens of thousands of data points are summarized in a single figure.



Summarizes information from approximately 250,000 numbers and is readily interpreted in a few seconds.

Representation

- The first step of visualization: the mapping of **information** to a **visual** format
- Data objects, their attributes, and the relationships among data objects are translated into graphical elements such as points, lines, shapes, and colors.
- Example:
  - Objects are often represented as points.
  - Their attribute values can be represented as the position of the points or the characteristics of the points, e.g., color, size, and shape.
  - If position is used, then the relationships of points, i.e., whether they form groups or a point is an outlier, is easily perceived.

Arrangement

- Is the placement of visual elements within a display
- Can make a large difference in how easy it is to understand the data? Example:

	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	1	0	1	0	0	1
5	0	1	0	1	1	0
6	1	0	1	0	0	1
7	0	1	0	1	1	0
8	1	0	1	0	0	1
9	0	1	0	1	1	0

	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

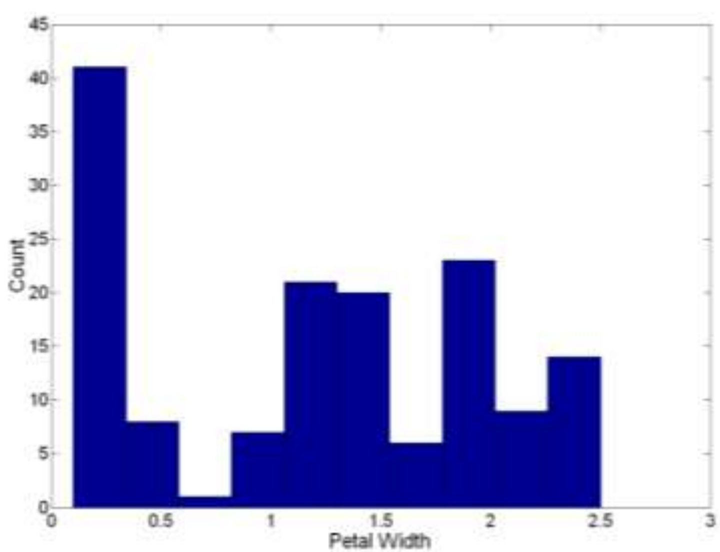
Same data
Re-arrange the sequence of rows and columns

Selection

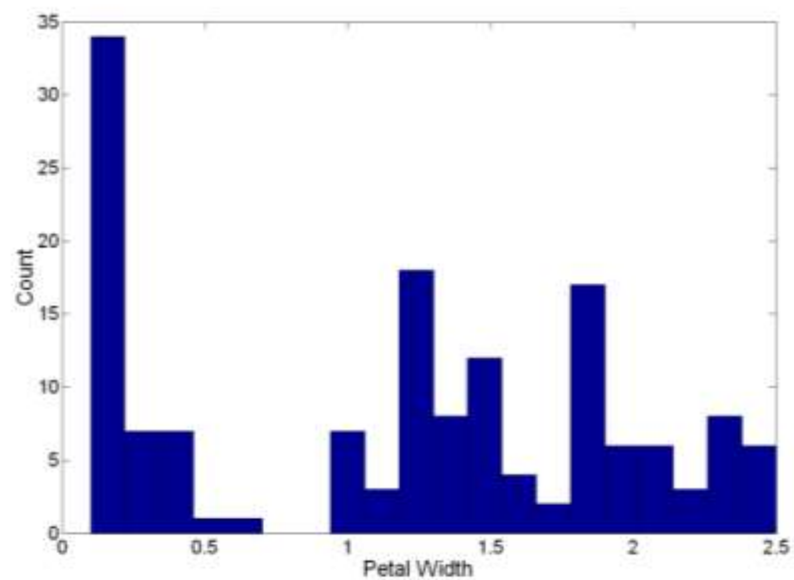
- Selection is the **elimination** or the **de-emphasis** of certain objects and attributes.
- Selection may involve choosing a **subset of attributes**
  - Commonly, pairs of attributes are considered.
  - Sophisticatedly, **dimensionality reduction** is often used to reduce the number of dimensions to *two or three.*
- Selection may also involve choosing a **subset of objects**
  - A region of the screen can only show so many points
  - Can sample, but want to preserve points in sparse areas

Visualization Techniques: 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.
  - The shape of a histogram depends on the number of bins.
- **Example:** Iris data set - **Petal Width** (10 and 20 bins, respectively)



large bin (10 bins)

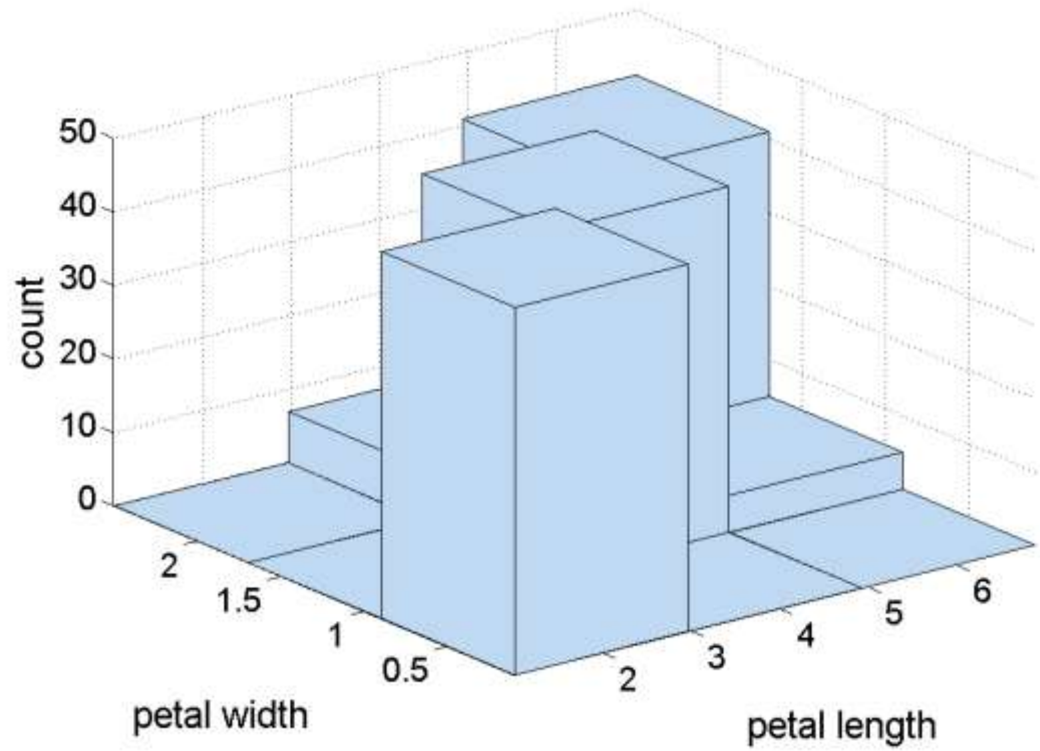


Small bin (20 bins)

Two-Dimensional Histograms

- Show the joint distribution of the values of two attributes
- Example:
  - Petal width and Petal length

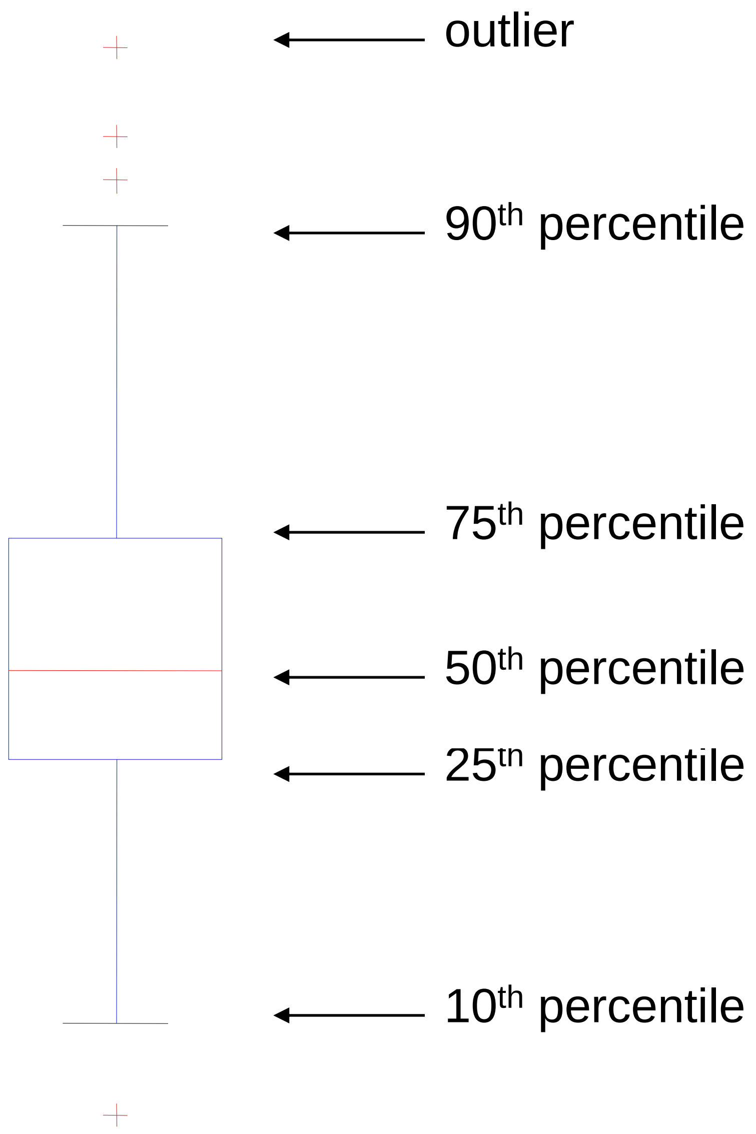
- What does this tell us?



http://en.wikipedia.org/wiki/Iris_flower_data_set

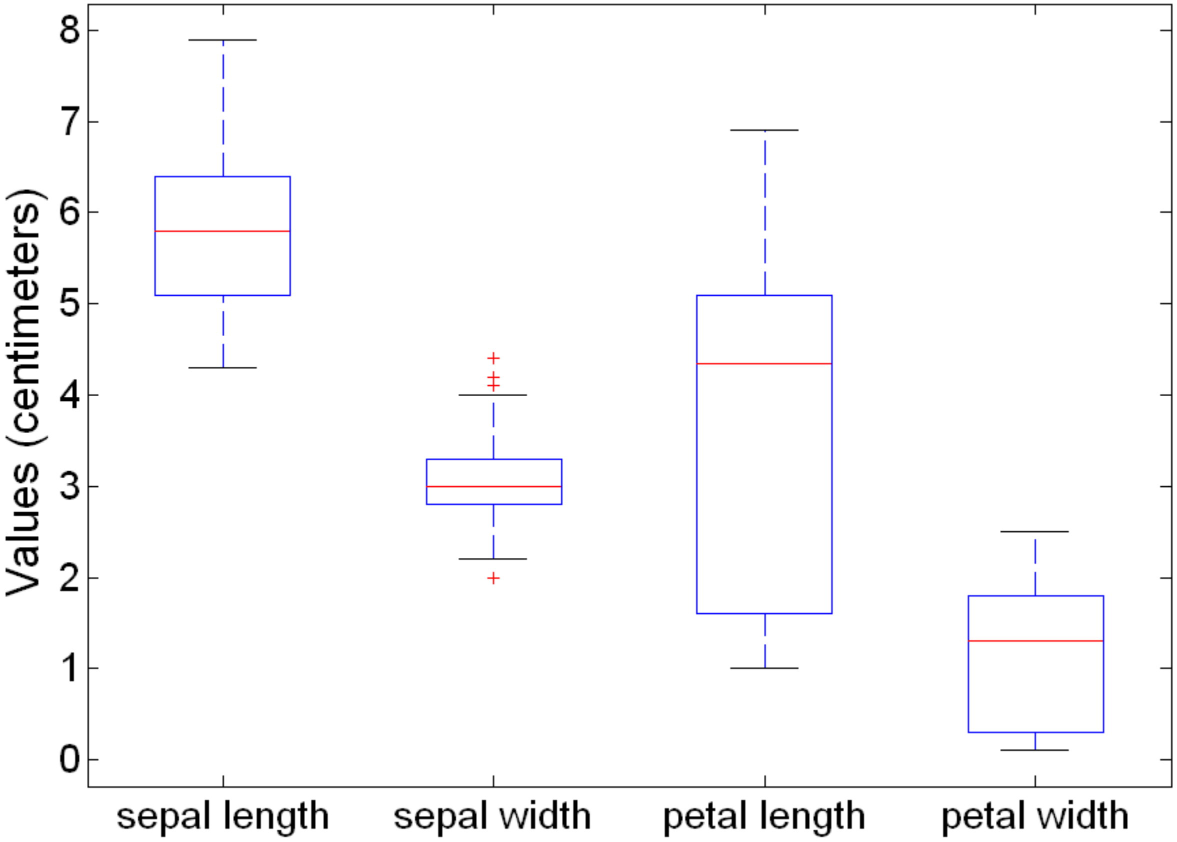
Visualization Techniques: Box Plots

- Box Plots
  - Invented by J. Tukey
  - Another way of displaying the distribution of data
  - The figure shows the basic part of a box plot



Example of Box Plots

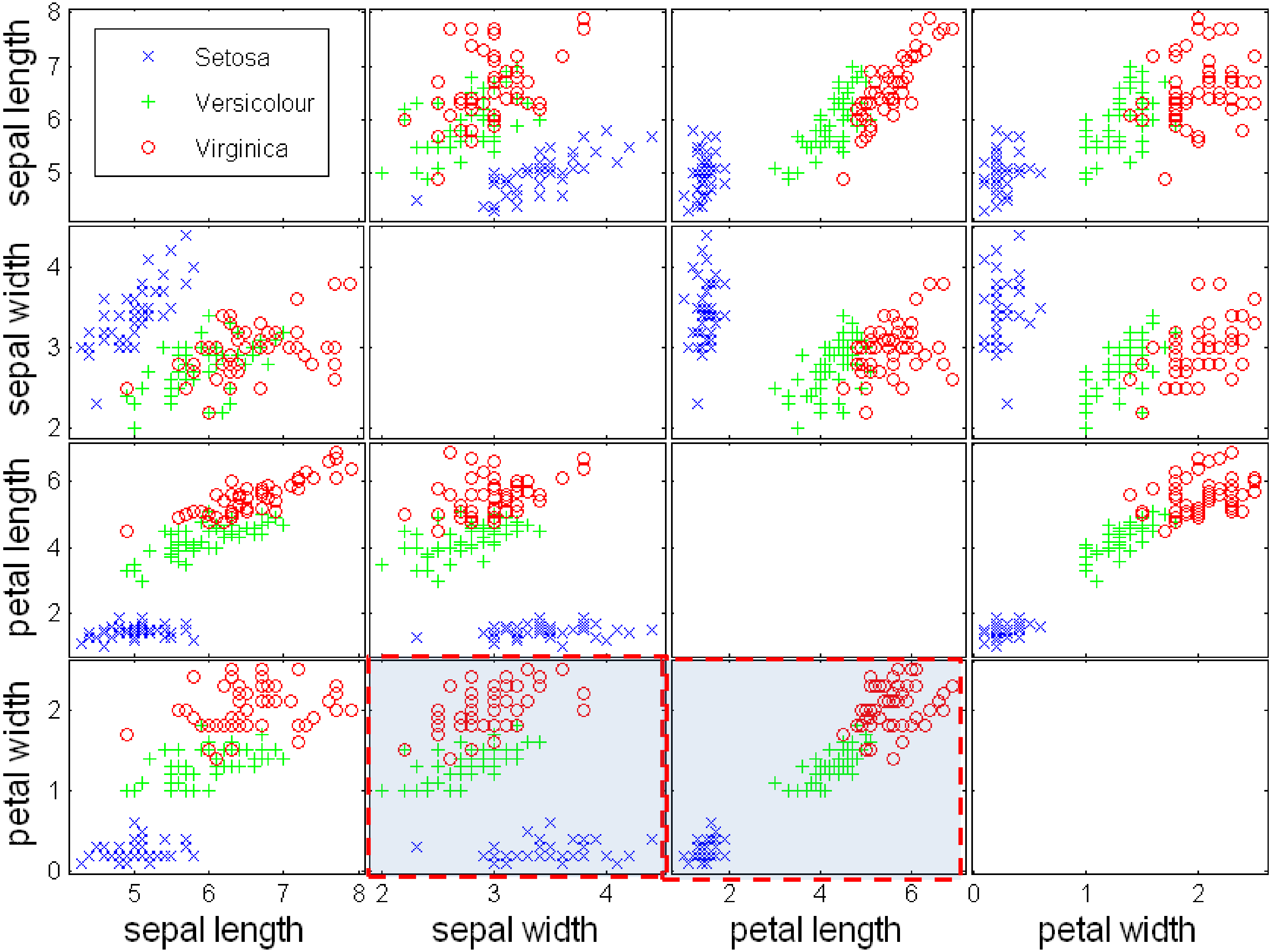
- Box plots can be used to compare attributes.



Visualization Techniques: Scatter Plots

- Scatter plots
  - Attributes values determine the position.
  - Two-dimensional scatter plots are most common, but we can have three-dimensional scatter plots.
  - Additional attributes often can be displayed by using the *size*, *shape*, and *color* of the markers that represent the objects.
  - It is useful to have arrays of scatter plots that can compactly summarize the relationships of several pairs of attributes.
    - See example on the next slide

Scatter Plot Array of Iris Attributes



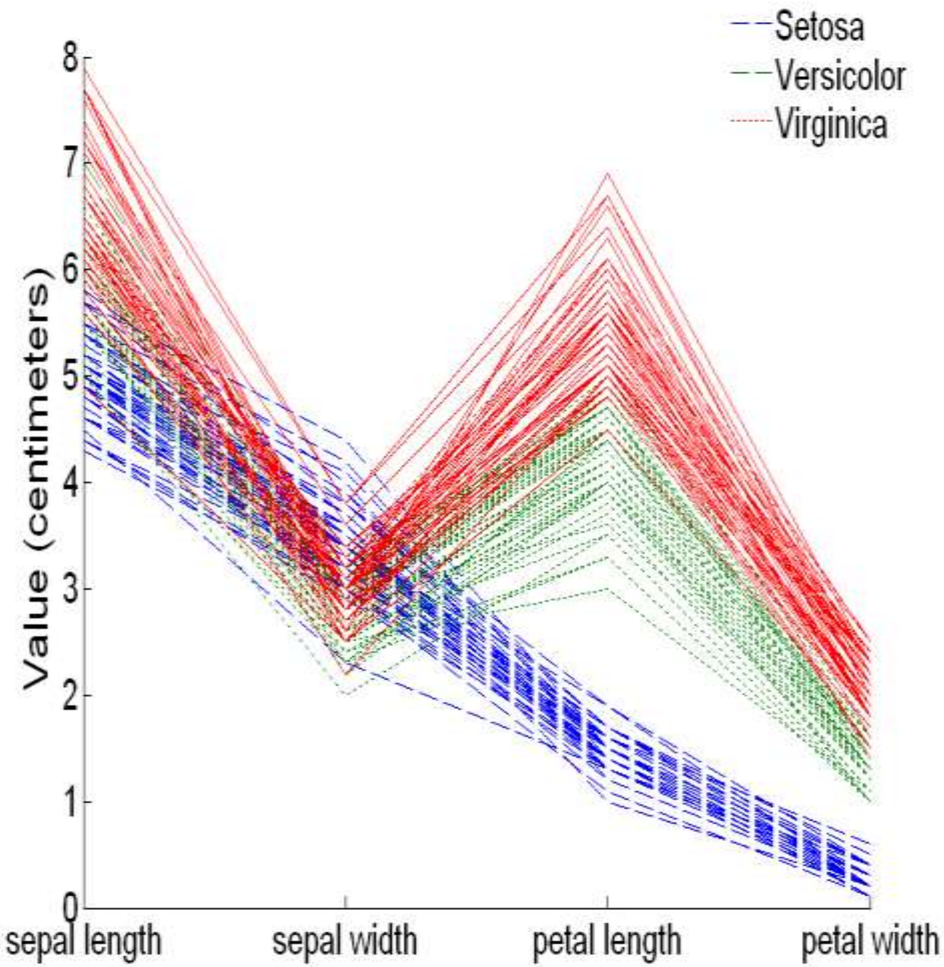
1. Correlations
2. Class distribution

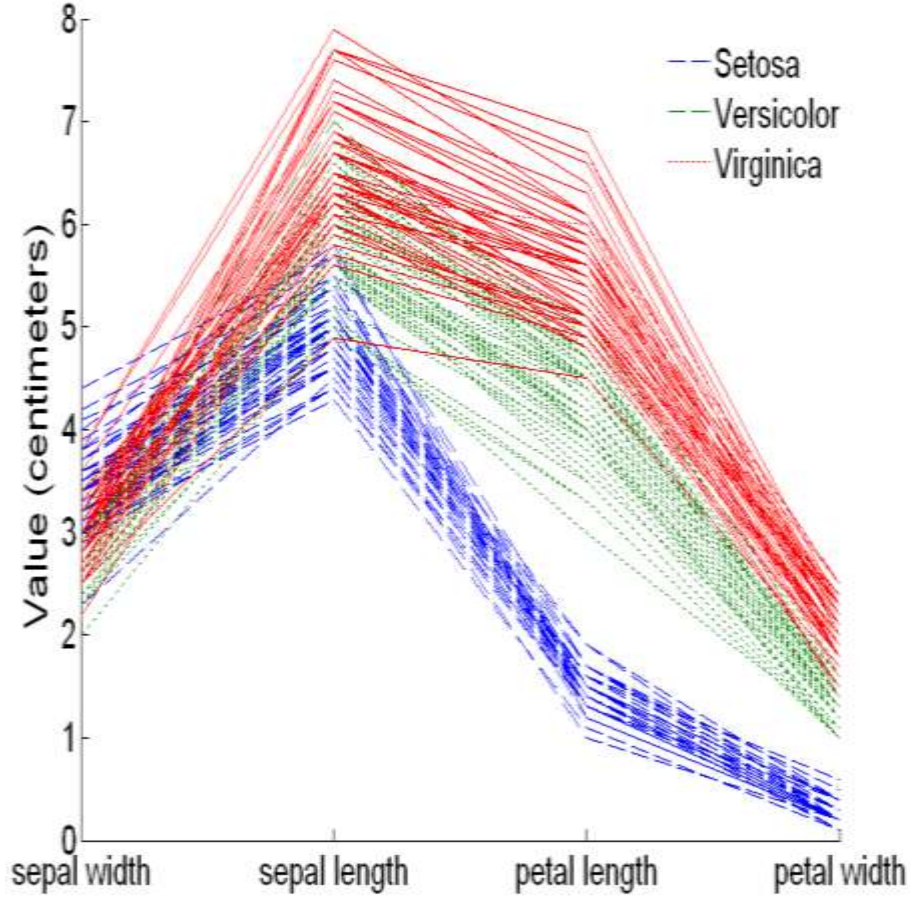
Visualization Techniques: Matrix Plots

- Matrix plots
  - Can plot the data matrix (all the data).
  - This can be useful when objects are sorted according to class.
  - Typically, **the attributes are normalized** to prevent one attribute from dominating the plot.
  - Plots of similarity or distance matrices can also be useful for visualizing the relationships between objects.

Parallel Coordinates Plots for Iris Data

Visualize all the150 data records

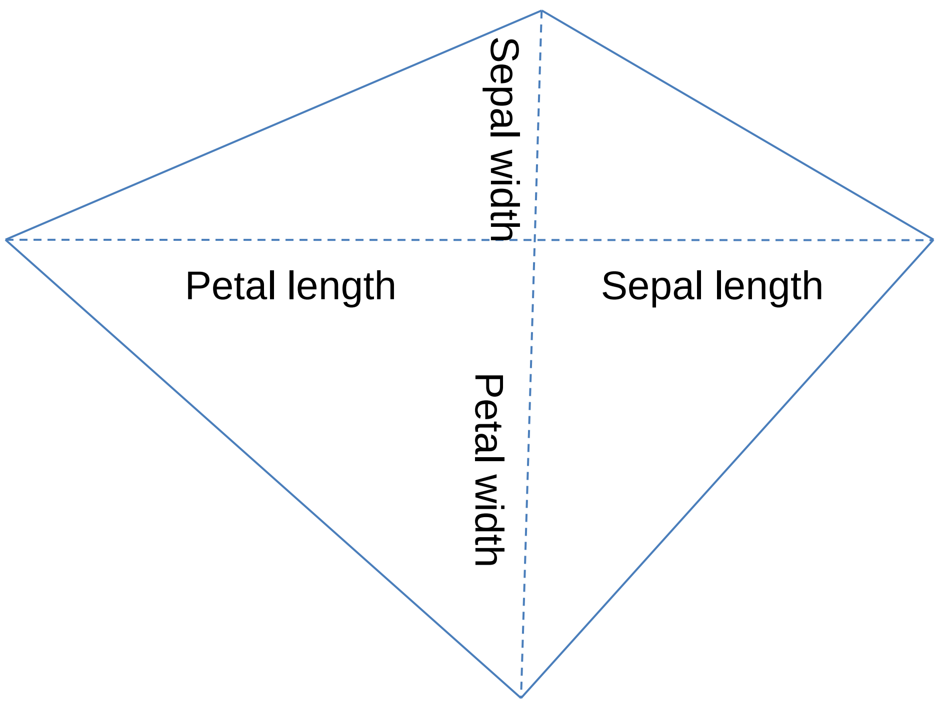




Change the sequence of the first two features

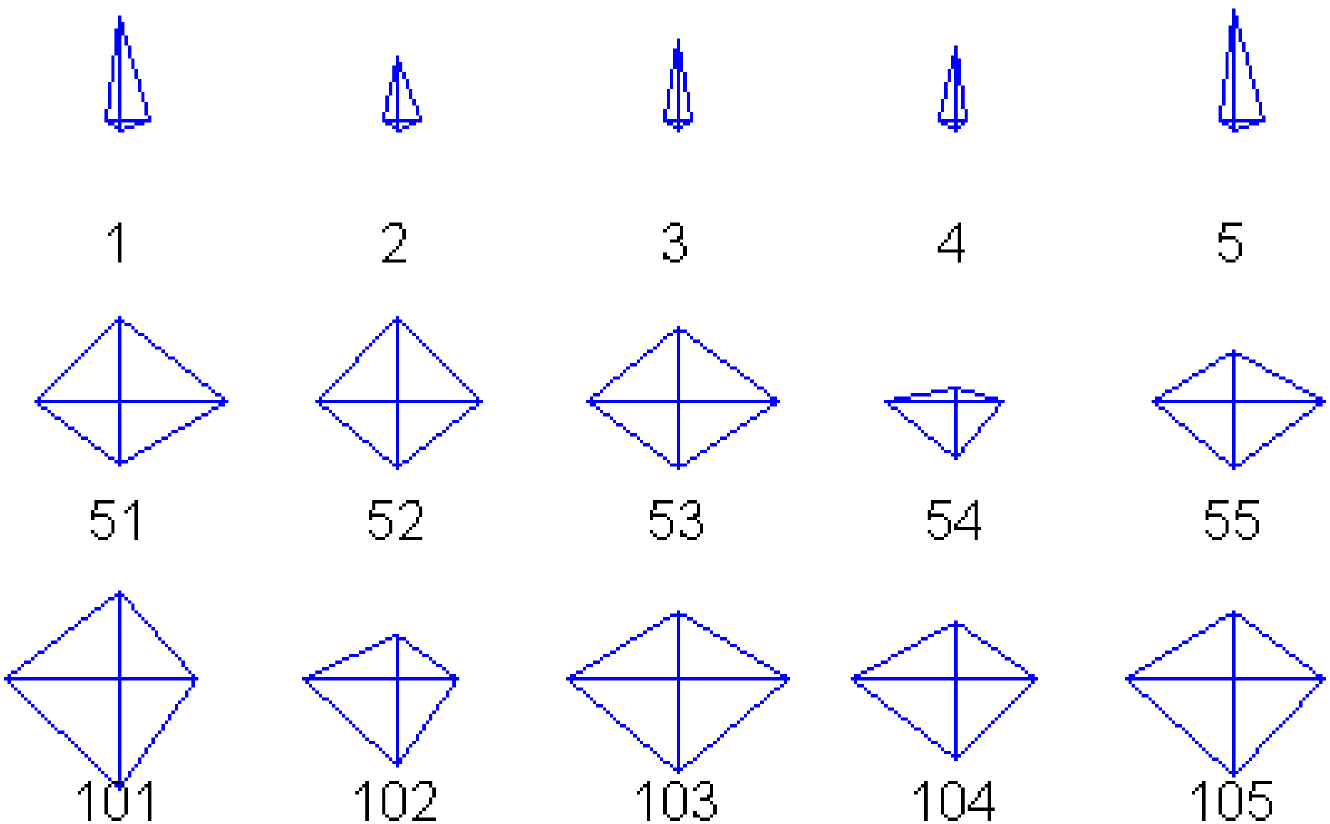
Other Visualization Techniques

- Star Plots
  - Similar approach to parallel coordinates, but axes radiate from a central point.
  - The line connecting the values of an object is a polygon.



Star Plots for Iris Data

Visualize all the15 data records



Setosa

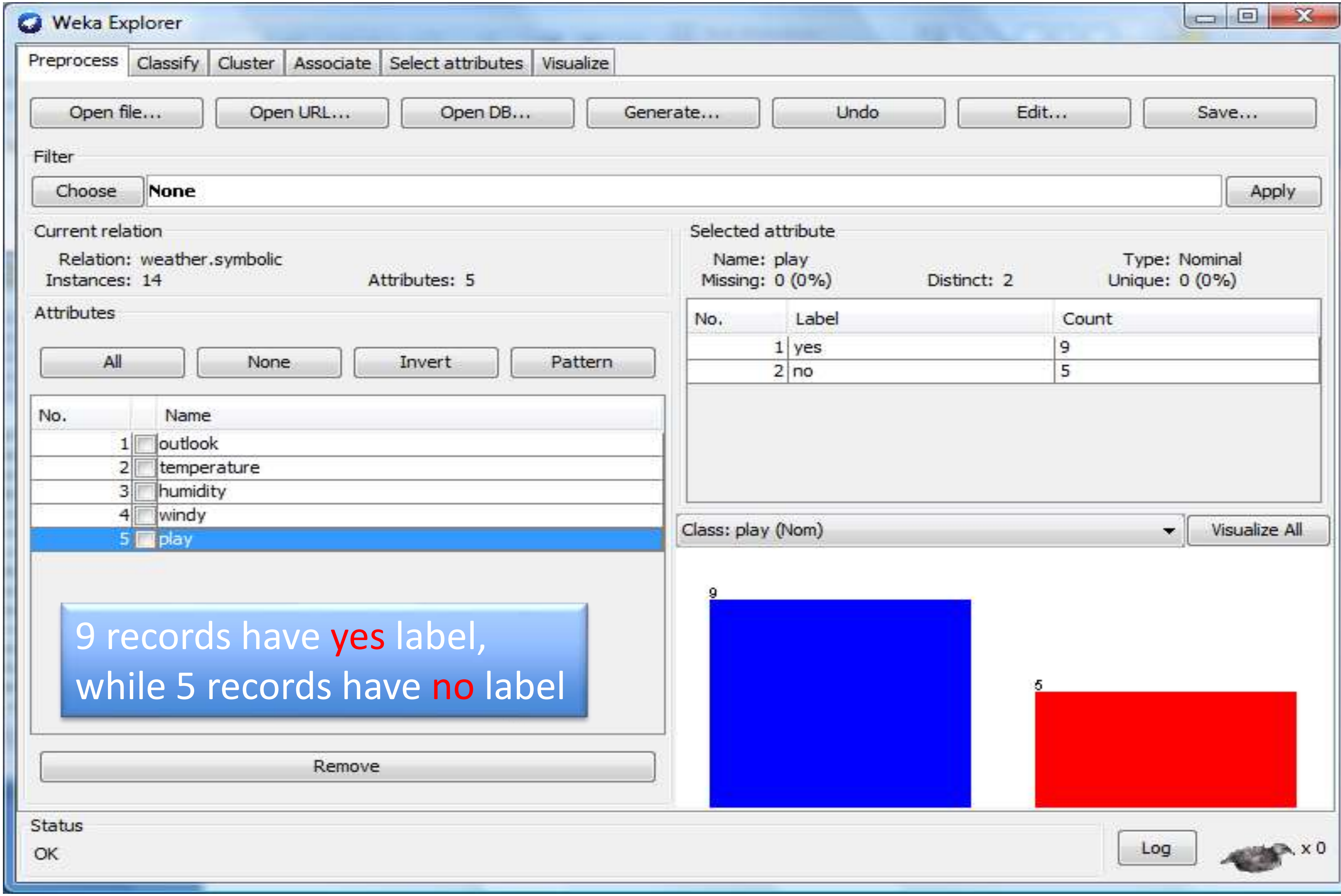
Versicolour

Virginica

Weather Data

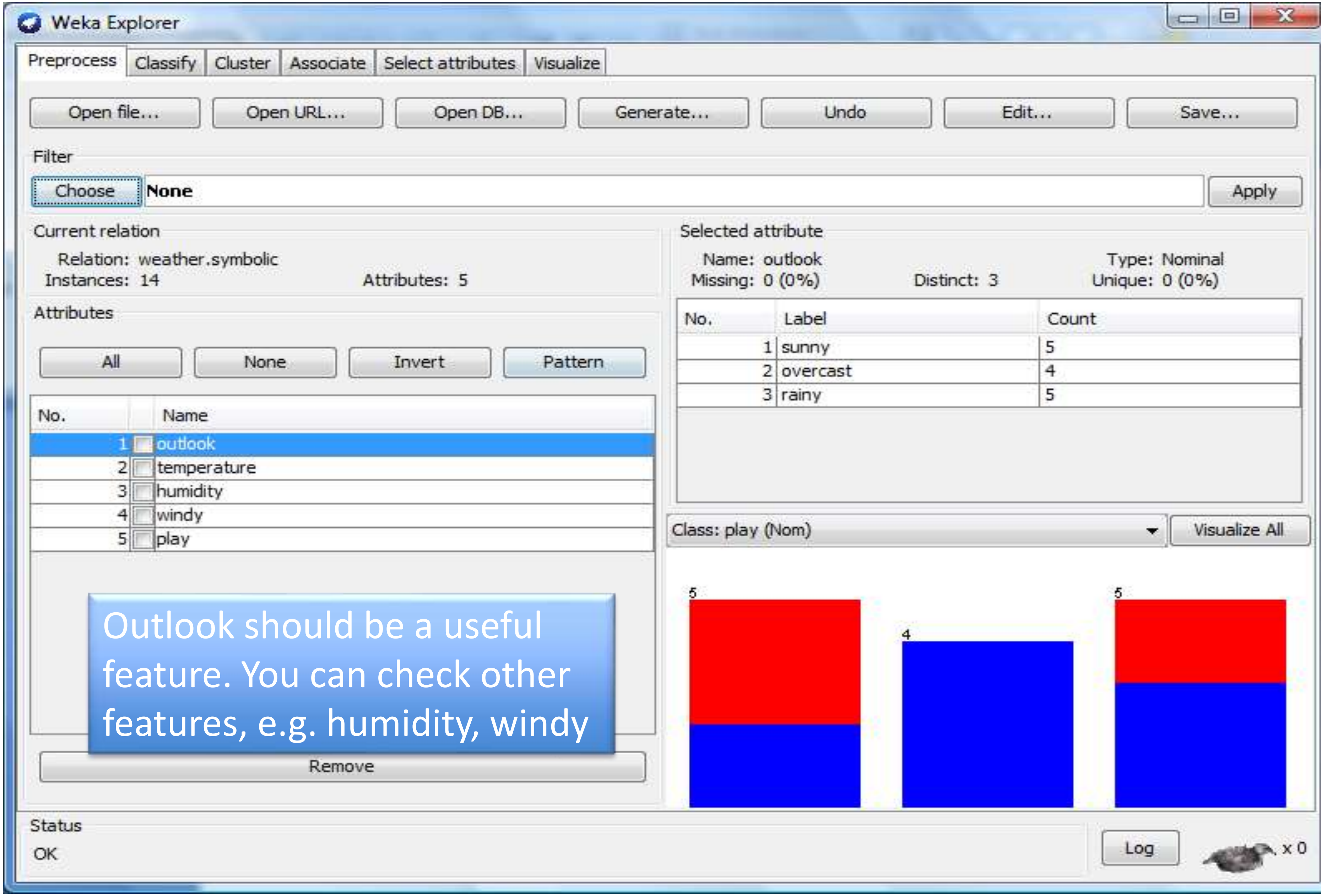
outlook	temperature	humidity	windy	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Play or Not to Play? Label distribution

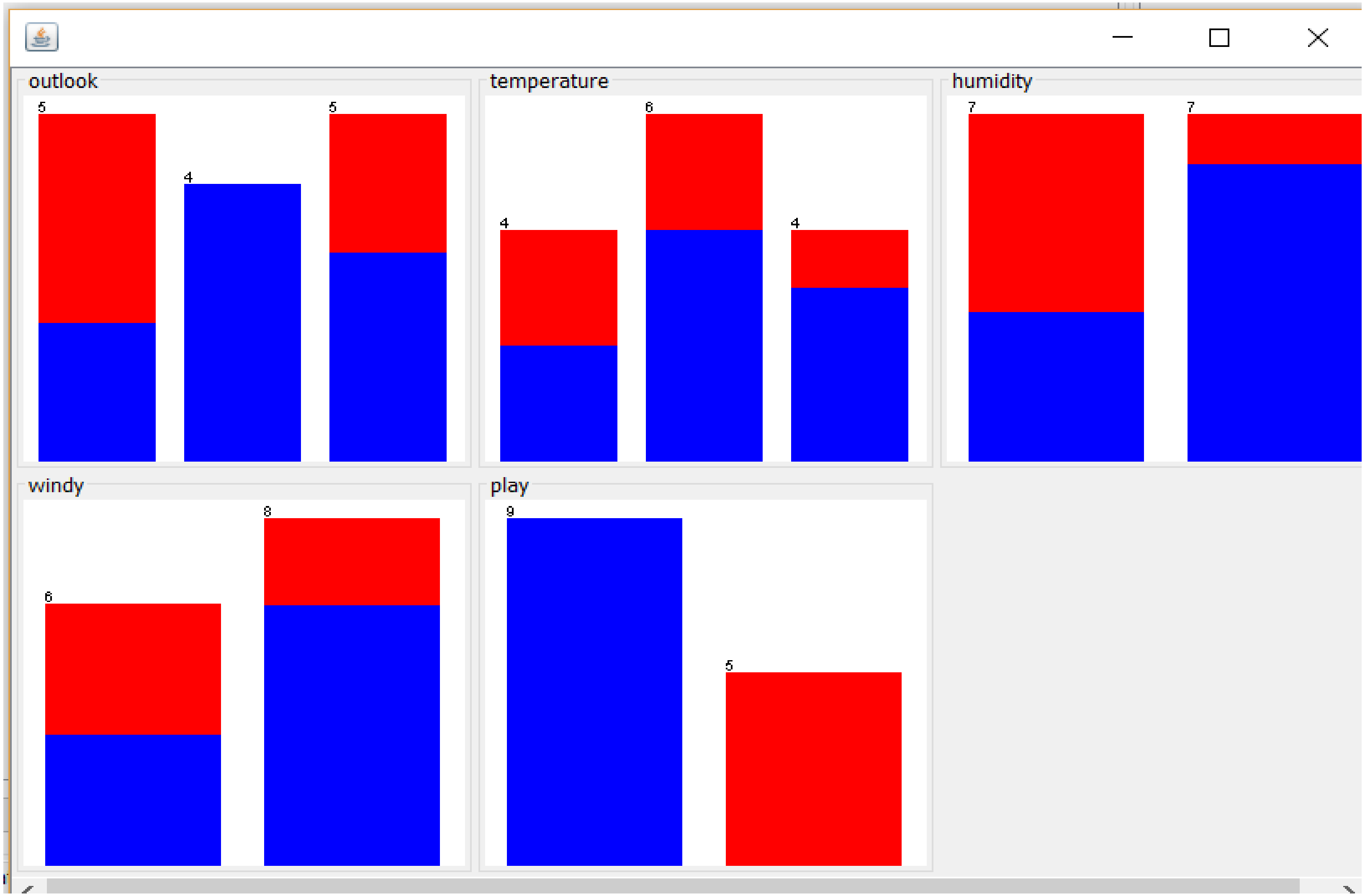


9 records have yes label, while 5 records have no label

How about the Outlook feature?



“Visualize All”



Get all the class distribution in one GO

How to compute mean and standard deviation [e.g from wiki]

For a finite set of numbers, the standard deviation is found by taking the square root of the average of the squared deviations of the values from their average value. For example, the marks of a class of eight students (that is, a **population**) are the following eight values:

$$2,\ 4,\ 4,\ 4,\ 5,\ 5,\ 7,\ 9.$$

These eight data points have the mean (average) of 5:

$$\frac{2+4+4+4+5+5+7+9}{8} = 5.$$

First, calculate the deviations of each data point from the mean, and square the result of each:

$$\begin{array}{ll} (2-5)^2 = (-3)^2 = 9 & (5-5)^2 = 0^2 = 0 \\ (4-5)^2 = (-1)^2 = 1 & (5-5)^2 = 0^2 = 0 \\ (4-5)^2 = (-1)^2 = 1 & (7-5)^2 = 2^2 = 4 \\ (4-5)^2 = (-1)^2 = 1 & (9-5)^2 = 4^2 = 16. \end{array}$$

https://www.mathsisfun.com/data/standard-deviation.html

The variance is the mean of these values:

$$\frac{9+1+1+1+0+0+4+16}{8} = 4.$$

and the *population* standard deviation is equal to the square root of the variance:

$$\sqrt{4} = 2.$$

This formula is valid only if the eight values with which we began form the complete population. If the values instead were a random sample draw from some larger parent population (for example, they were 8 marks randomly chosen from a class of 20), then we would have divided by 7 (which is $n-1$) instead of 8 (which is n) in the denominator of the last formula, and then the quantity thus obtained would be called the *sample* standard deviation.

Cross Validation

5-fold cross validation

- Divide samples into k roughly equal disjoint parts
- Each part has similar proportion of samples from different classes.
- Use each part to test other parts
- Average accuracy and F-measure etc