Data-driven Intelligent Systems

Lecture 6
Dimensionality Reduction Techniques

KNOWLEDGE
TECHNOLOGY

http://www.informatik.uni-hamburg.de/WTM/

Dimensionality Reduction Techniques

- Features Reduction
- Correlation Analysis
- Data Transformation
  - Normalization
  - PCA
- Sampling

Data Reduction Strategies

Why data reduction? — A data warehouse may store terabytes ...

- Data analysis may take a very long time to run on the complete data set
- A reduced representation may produces (almost) same analytical results

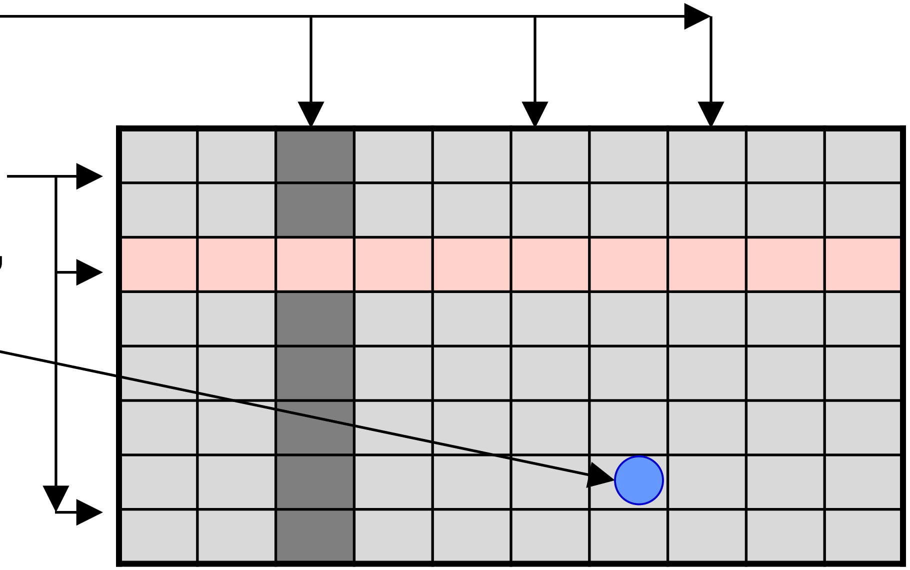
Data reduction strategies

- ***Dimensionality reduction***, e.g., remove unimportant attributes
  - Feature subset selection, feature creation
  - Wavelet-, Fourier transforms
  - Principal Components Analysis (PCA)
- ***Numerosity*** reduction (some simply call it: Data Reduction)
  - Regression, e.g. linear or log-linear models
  - Histograms, clustering, sampling
  - Data cube aggregation
- ***Data compression***

Dimensions Reduction of Large Data Sets

Main dimensions:

- ***columns*** (features),
  dimensionality reduction
- ***rows*** (cases or samples),
  numerosity reduction
  → sampling
- ***values*** of the features
  for the given sample
  data compression



Dimensionality Reduction

Databases store a lot of attributes (variables, features), which determine the **dimensionality** of the variable- or feature space. Problems:

- Amount of stored data in ranges of terabytes
- ***Curse of dimensionality***
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points becomes less meaningful, which is critical to, e.g., clustering, outlier analysis
  - The possible combinations of subspaces will grow exponentially

Dimensionality reduction

- Avoid the curse of dimensionality
- Help eliminate irrelevant features and reduce noise
- Reduce time & space required by data mining code
- Allow easier visualization of feature space and possible correlations

Dimensionality Reduction

Two standard approaches:

- ***Feature selection***: A process that chooses an optimal subset of features according to an objective function:
  - feature ranking algorithms,
  - minimum subset algorithms.
- ***Feature extraction***: refers to the mapping of the original high-dimensional data onto a lower-dimensional space.
  - Descriptive setting: minimizes information loss
  - Predictive setting: maximizes the class discrimination

Feature Selection –
Example for Optimal Features' Subset

F_1	F_2	F_3	F_4	F_5	C
0	0	1	0	1	0
0	1	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
0	0	1	1	0	0
0	1	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1

- Data set (whole set)
  - Five Boolean features
  - $C = F_1 \vee F_2$
  - $F_3 = \neg F_2$, $F_5 = \neg F_4$
  - Optimal subset:

    $\{F_1, F_2\}$ or $\{F_1, F_3\}$
- Combinatorial nature of searching for an optimal subset

Features Selection: A Univariate Method

Prerequisite: known class labels, i.e. two classes (A and B).

Intuition: keep a feature only if it separates the classes well.

Simplification: Use only means and variances.

- Test: $$\frac{|\mathrm{mean}(A) - \mathrm{mean}(B)|}{\mathrm{SE}(A-B)} > threshold\_value$$

  where standard error: $$\mathrm{SE}(A-B) = \sqrt{\frac{\mathrm{var}(A)}{n_A} + \frac{\mathrm{var}(B)}{n_B}}$$

  numbers of samples for classes A and B

For one class: $SE = \frac{stddev}{\sqrt{n}}$

indicates how the sample mean differs from the population mean

Features Selection: A Univariate Method

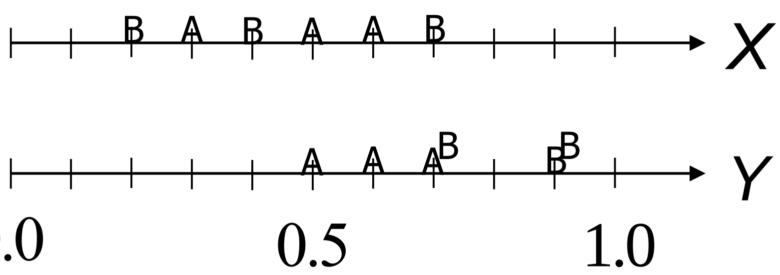
- Comparison of *means* and *variances* – **Example**:

X	Y	C
0.3	0.7	A
0.2	0.9	B
0.6	0.6	A
0.5	0.5	A
0.7	0.7	B
0.4	0.9	B

threshold_value is 0.5

$X_A = \{0.3, 0.6, 0.5\}, \quad X_B = \{0.2, 0.7, 0.4\},$

$Y_A = \{0.7, 0.6, 0.5\}, \quad Y_B = \{0.9, 0.7, 0.9\}$



Are X or Y candidates for reduction?

Features Selection: A Univariate Method

- Comparison of *means* and *variances* – **Example**:

X: $$SE(X_A - X_B) = \sqrt{\frac{var(X_A)}{n_A} + \frac{var(X_B)}{n_B}} = \sqrt{\frac{0.0233}{3} + \frac{0.06333}{3}} = 0.170$$

Y: $$SE(Y_A - Y_B) = \sqrt{\frac{var(Y_A)}{n_A} + \frac{var(Y_B)}{n_B}} = \sqrt{\frac{0.01}{3} + \frac{0.0133}{3}} = 0.0875$$

Tests:

X: $$\frac{|mean(A) - mean(B)|}{SE(A - B)} = \frac{|0.4667 - 0.4333|}{0.170} < 0.5$$

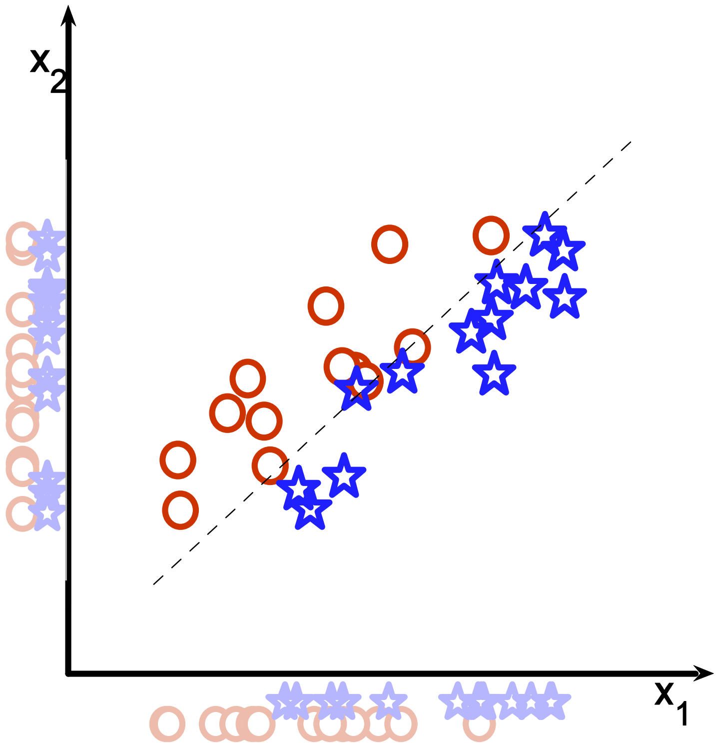
Y: $$\frac{|mean(A) - mean(B)|}{SE(A - B)} = \frac{|0.6 - 0.8333|}{0.0875} > 0.5$$

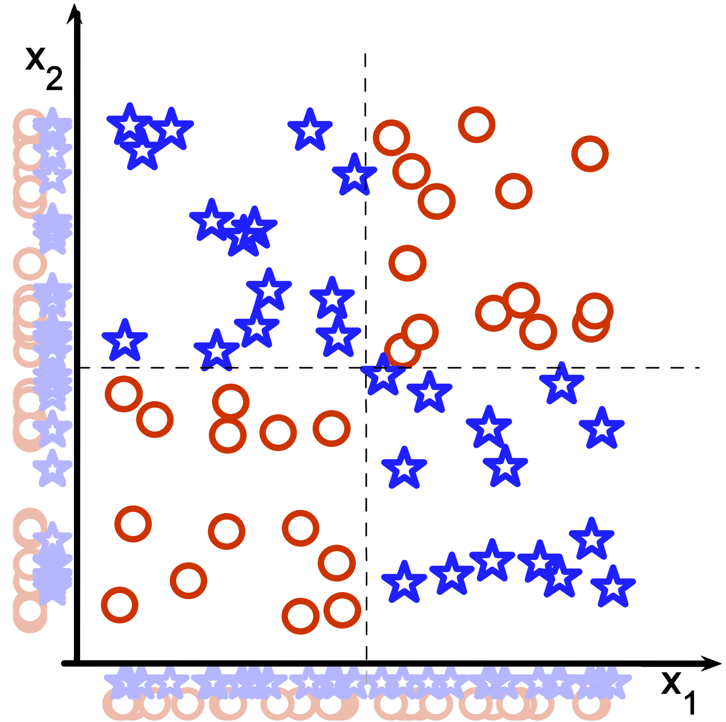
X is a candidate feature for reduction
because its mean values are close;
the final test is below threshold value.

Methods of Feature Selection

- Univariate methods
  - Consider one variable (feature) at a time.
- Filter methods
  - Separate feature selection from classifier learning
  - Rely on general characteristics of data (information, distance, dependence, consistency)
    - Drop features based on general characteristics, e.g. no correlation with the class
  - No bias toward any learning algorithm, fast
- Wrapper methods
  - Rely on a predetermined classification algorithm
  - Using predictive accuracy as goodness measure
    - Drop features that do not help the model to predict the class
  - High accuracy, computationally expensive
- Embedded methods
  - Combine Filter and Wrapper approaches

Feature Selection may Fail!





[Guyon-Elisseeff, JMLR 2004; Springer 2006]

Dimensionality Reduction Techniques

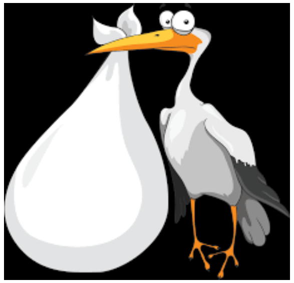
- Features Reduction
- Correlation Analysis
- Data Transformation
  - Normalization
  - PCA
- Sampling

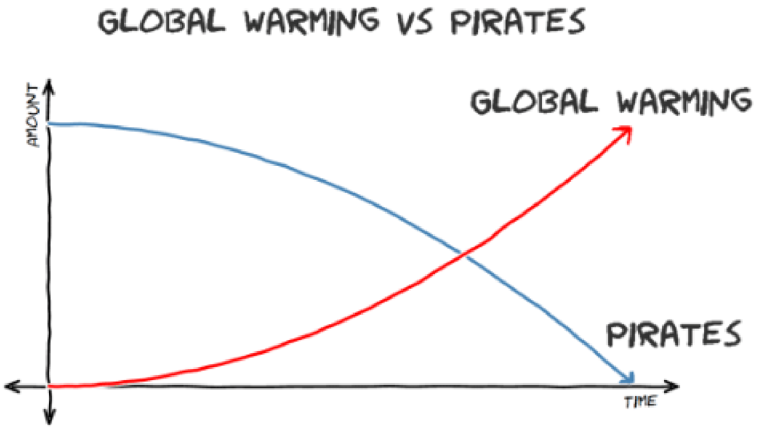
Handling Redundancy in Data Integration

- ***Redundant data*** occur often when integrating multiple databases
  - ***Object identification***: The same attribute or object may have different names in different databases
  - ***Derivable data:*** One attribute may be a "derived" attribute in another table, e.g., annual revenue
- Redundant attributes may be possible to detect by ***correlation analysis***

Correlation Analysis

- *"A mutual relationship or connection between two or more things."* (Oxford Dictionary)
- "things" can be:
  - Variables (e.g. regression ✓)
  - Features (e.g. PCA, later in this lecture)
  - Items (apriori algorithm, → later Lecture)
- Correlation ≠ Causation!
  - Positive correlation between birth rate and stork population
  - Negative correlation between number of pirates and global warming





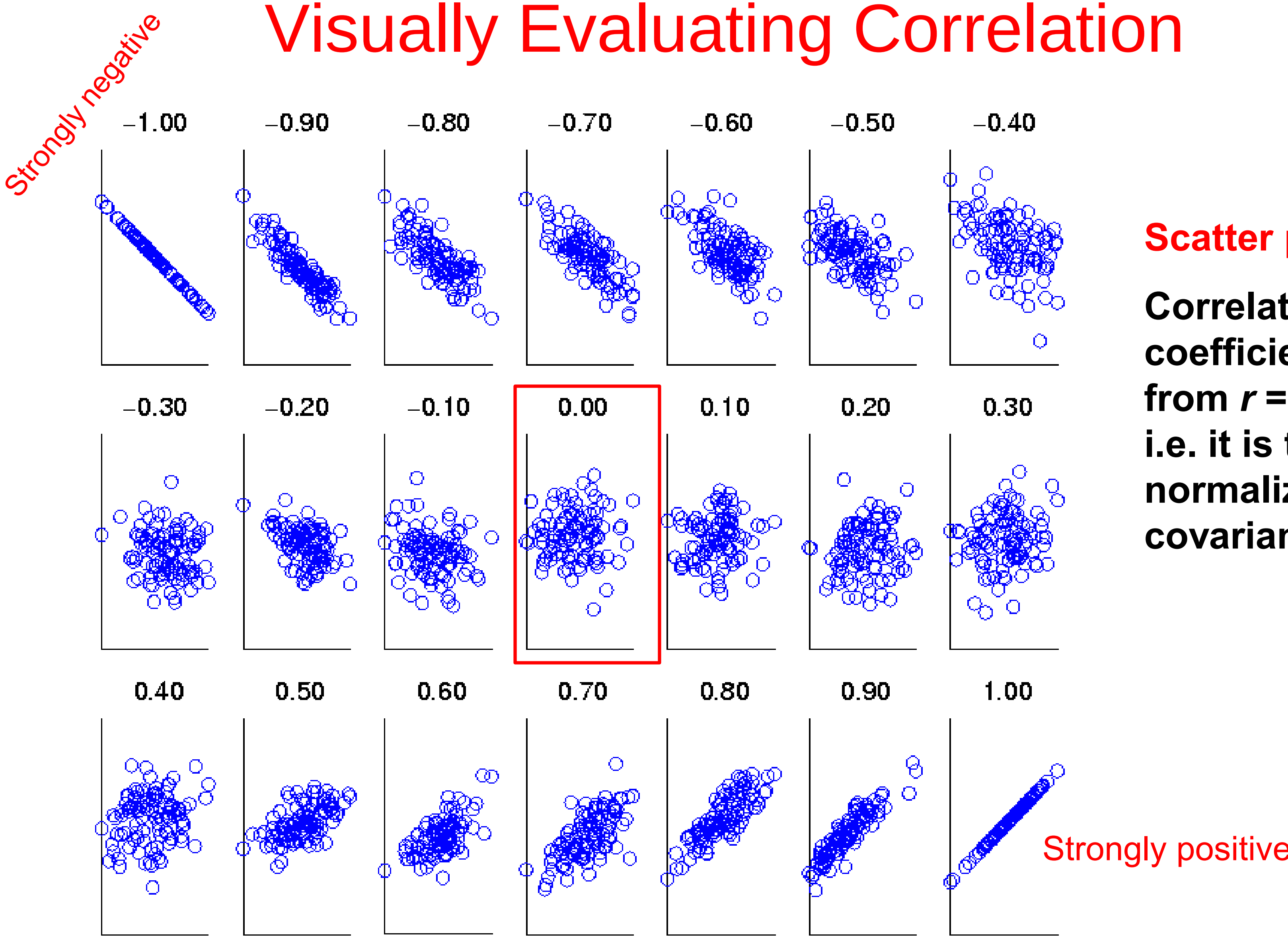
Correlation Analysis (Numeric Data)

- Correlation coefficient (also called ***Pearson's product moment coefficient***)

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \overline{A})(b_i - \overline{B})}{n\sigma_A\sigma_B}$$

  - n = number of tuples
  - $\overline{A}$ and $\overline{B}$ = means or expected values of attributes A and B
  - σ_A and σ_B = standard deviations of A and B
- $r_{A,B} > 0$: A and B are positively correlated
  - A's values increase as B's. Larger $r_{A,B} \rightarrow$ stronger correlation.
- $r_{A,B} = 0$: uncorrelated, not necessarily independent
- $r_{AB} < 0$: negatively correlated

Visually Evaluating Correlation



Scatter plots

Correlation coefficients range from *r* = –1 to 1, i.e. it is the normalized covariance

Covariance (Numeric Data)

- Covariance:

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{n=1}^{N}(a_n-\bar{A})(b_n-\bar{B})}{N}$$

Related to correlation coefficient: $r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$

  - N = number of tuples
  - $\overline{A}$ and $\overline{B}$ = mean or expected values of A and B
  - σ_A and σ_B = standard deviation of A and B.
- ***Positive covariance*** ($Cov_{A,B} > 0$): A and B tend to be together larger or together smaller than their expected values
- ***Negative covariance*** ($Cov_{A,B} < 0$): if A is larger than its expected value, B is likely to be smaller than its expected value.
- ***No relationship***: $Cov_{A,B} = 0$ (does not necessarily imply statistical independence between variables!)

Co-Variance: an Example

- It can be simplified in computation as

$$Cov(A,B) = E\left(A \cdot B\right) - \bar{A} \cdot \bar{B}\big)$$

- Suppose two stocks A and B have the following values in one week: (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- **Question**: If the stocks are affected by the same industry trends, will their prices rise or fall together?
  - $E(A)$ = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4
  - $E(B)$ = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6
  - $Cov(A,B)$ = (2 ·5 + 3 ·8 + 5 ·10 + 4 ·11 + 6 ·14)/5 − 4 ·9.6 = 4
- Thus, A and B rise together since $Cov(A, B) > 0$.

Co-Variance Matrix

Let *X* be a set of *N* data vectors $\{x_n\}$

$$Cov(X) = E((X-\bar{X})(X-\bar{X})^T) = \frac{\sum_{n=1}^{N}(x_n-\bar{x})(x_n-\bar{x})^T}{N}$$

- Element *i,j* of the covariance matrix

$$Cov_{ij}(X) = E((X_i-\bar{X}_i)\left(X_j-\bar{X}_j\right)^T) = \frac{\sum_{n=1}^{N}(x_{ni}-\bar{x}_i)(x_{nj}-\bar{x}_j)}{N}$$

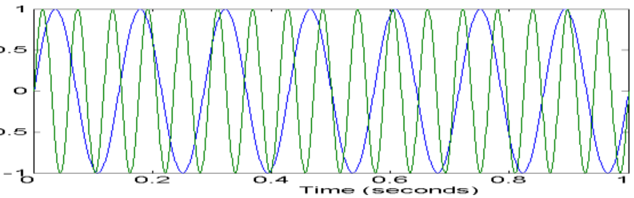
  computes the covariance between feature *i* and feature *j* .
- The covariance matrix is symmetric: $Cov_{ij} = Cov_{ji}$
- Diagonal element Cov_{ii} is the variance along dimension *i* .

Dimensionality Reduction Techniques

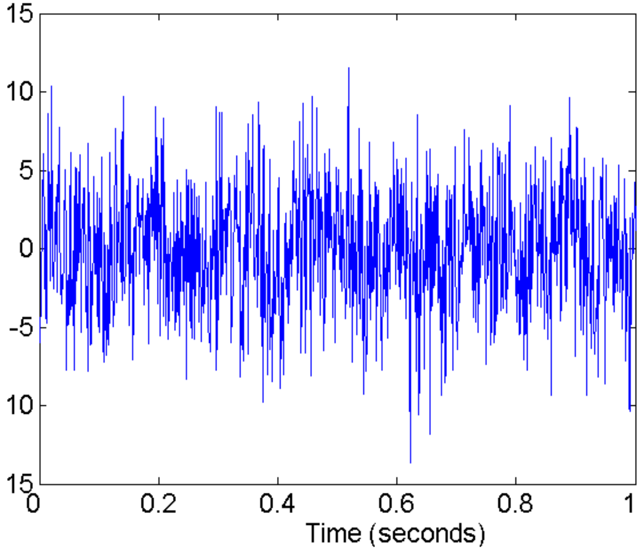
- Features Reduction
- Correlation Analysis
- Data Transformation
  - Normalization
  - PCA
- Sampling

Motivation: Mapping Data to a New Space

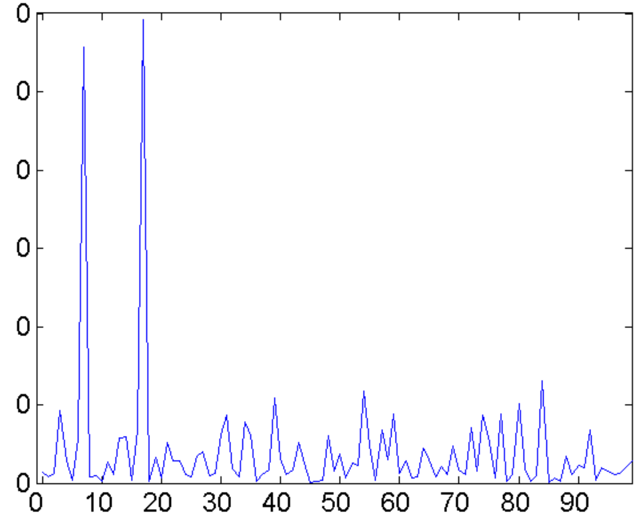
- Example: Fourier analysis. Mapping from time to frequency domain allows detection of frequencies not observable by time-series only



Two Sine Waves



Two Sine Waves + Noise



Frequencies

- Noise detection & easy removal (suppression of certain frequencies)
- Related: wavelet transform

Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values so that each old value can be identified with one of the new values
- Methods
  - Normalization: Scaled to fall within a smaller, specified range
    - min-max normalization
    - z-score normalization
    - whitening
    - normalization by decimal scaling
  - Attribute/feature construction
    - New attributes constructed from the given ones
    - E.g. PCA
  - Other methods such as smoothing or aggregation could also be seen as transformations

Dimensionality Reduction Techniques

- Features Reduction
- Correlation Analysis
- Data Transformation
  - Normalization
  - PCA
- Sampling

Normalization

- Normalization means rescaling data into a specific range or keeping data statistics into account
- Why data normalization is necessary?
  - Data attributes have different ranges and units
    - Example: a car can be described by weight, power, motor life span, fuel consumption, run time etc. → all features have different values and units
    - Even units can be different: Kilos vs. Tons? Km/h vs. mph? (→ check data consistency)
    - Databases have many attributes of many different objects
  - Consequence: Data Mining techniques produce wrong results or may even fail
    - E.g. Clustering, classification

Normalization (I/IV)

- ***Min-max normalization:*** *to [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:

Let income range from \$12,000 to \$98,000 and we want to normalize all \$-values to a new range [0.0, 1.0].

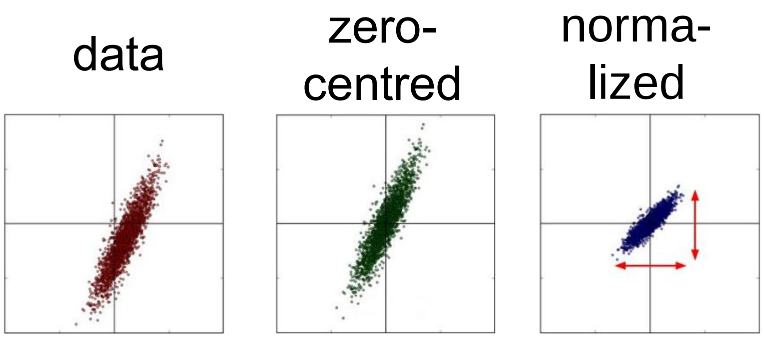
Then a test value \$73,600 will be mapped to:

$$\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$$

Normalization (II/IV)

- ***Z-score normalization***
  - Takes data statistics into account
  - An attribute value *v* is transformed into a new value *v’*

$$v' = \frac{v - \mu_A}{\sigma_A}$$



  - *μ*: mean
  - *σ*: standard deviation of a set *A*

- Example: Let the mean of an attribute set be μ = $ 54,000 with standard deviation σ = $16,000.
  Then, the value $73,600 will be normalized to:

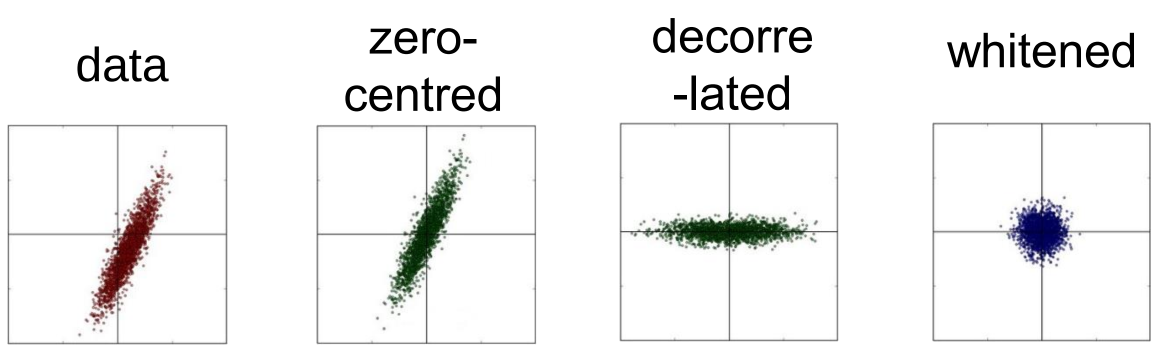
$$\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$$

Normalization (III/IV)

- ***Whitening***
  - Use eigen-decomposition of the covariance matrix: $Cov = U \Lambda U^{-1}$
  - Transform an attribute value v into a new value v' *as:*

$$v' = \Lambda^{-1} U^{T}(v - \mu)$$

  - μ: mean
  - Λ: variances (= eigenvalues of Cov on diagonal matrix Λ)
  - U*:* eigenvectors of *Cov* (= principal components)



Normalization (IV/IV)

- ***Normalization by decimal scaling***

$$v' = \frac{v}{10^j}$$ where *j* is the smallest integer such that Max(|v'|) < 1

- Simplest form of scaling of any decimal number into [-1.0;1.0]
- Move the decimal point according to the maximum value in your data
  - Example: Assume we have values between -5000 and 200. Our condition is Max(|v'|) < 1, so we compute:

    $\frac{-5000}{10000}$ = -0.5, where j=4 the decimal point was moved

Other Transformations of Raw Data

- ***Data smoothing***

  $F = \{0.93, 1.01, 1.001, 3.02, 2.99, 5.03, 5.01, 4.98\},$

  $F_{smoothed} = \{1.0, 1.0, 1.0, 3.0, 3.0, 5.0, 5.0, 5.0\}.$

- ***Differences and ratios***, e.g. for time series:

  $$s(t+1) - s(t)$$

  $$s(t+1) / s(t)$$

- ***Composing new features***
  - Example, Body Mass Index:

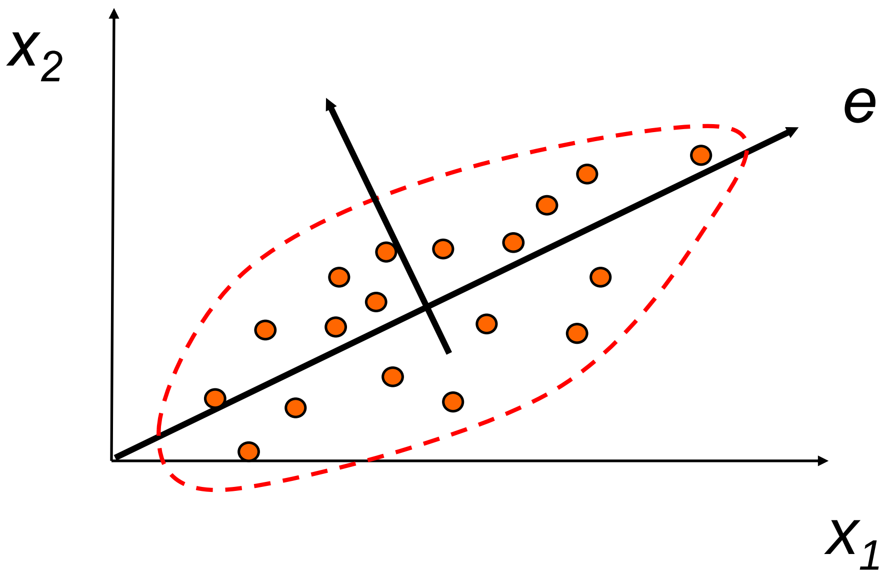
    $BMI = k \cdot F(Weight, Height)$

Dimensionality Reduction Techniques

- Features Reduction
- Correlation Analysis
- Data Transformation
  - Normalization
  - PCA
- Sampling

Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- Works for numeric data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



PCA (Steps)

- Given N data vectors from D-dimensions, find $m \leq D$ orthogonal vectors (***principal components***) to represent the data
  - Subtract mean from the input data, each attribute has mean zero
  - Compute m orthonormal (unit) vectors, i.e., *principal components*
  - Each input data (vector) is a linear combination of the m principal component vectors
  - The principal components are sorted in order of decreasing "***significance***" (1st component: data has maximum variance)
  - Since the components are sorted, the size of the data can be reduced by eliminating the *insignificant components*, i.e., those with low variance *(often: $m \ll D$)*
  - Thus, using only the most significant principal components, it is possible to reconstruct a good approximation of the original data.

PCA Algorithm

1. Compute the $D \times D$ ***covariance matrix*** Cov

$$Cov_{ij} = \frac{1}{N-1} \cdot \sum_{n=1}^{N} (x_i^n - \bar{x}_i)^T \cdot (x_j^n - \bar{x}_j) \qquad \text{where} \qquad \bar{x} = \frac{1}{N-1} \cdot \sum_{n=1}^{N} x^n$$

2. Calculate the ***eigenvalues*** of Cov for the given data and sort them:

   $\{\lambda_1, \lambda_2, \ldots, \lambda_D\}$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D \geq 0.$

   The eigenvalues are the ***variances*** of the data in the directions of the respective eigenvectors.

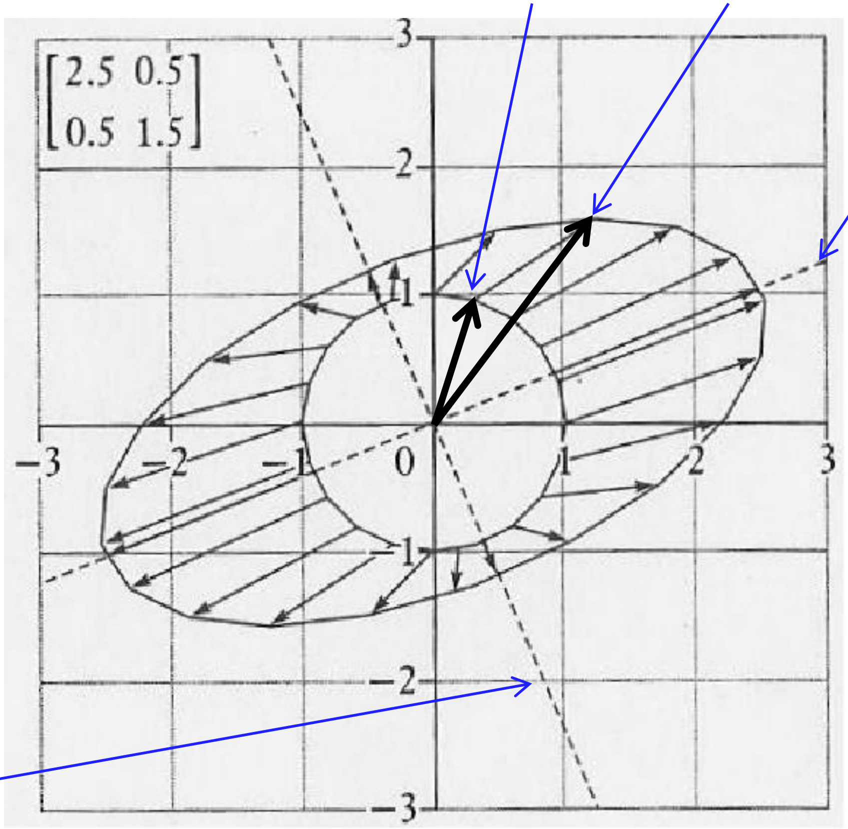
3. The ***eigenvectors*** $e_1, e_2, \ldots, e_D$ correspond to respective eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_D$,

   The eigenvectors are called the ***principal axes***.

Multiply a Vector with a D×D Matrix

- Symmetric matrix S

e.g. $S = Cov$



Eigenvector(s),

$S \cdot e_1 = \lambda_1 \cdot e_1$

large eigenvalue λ_1

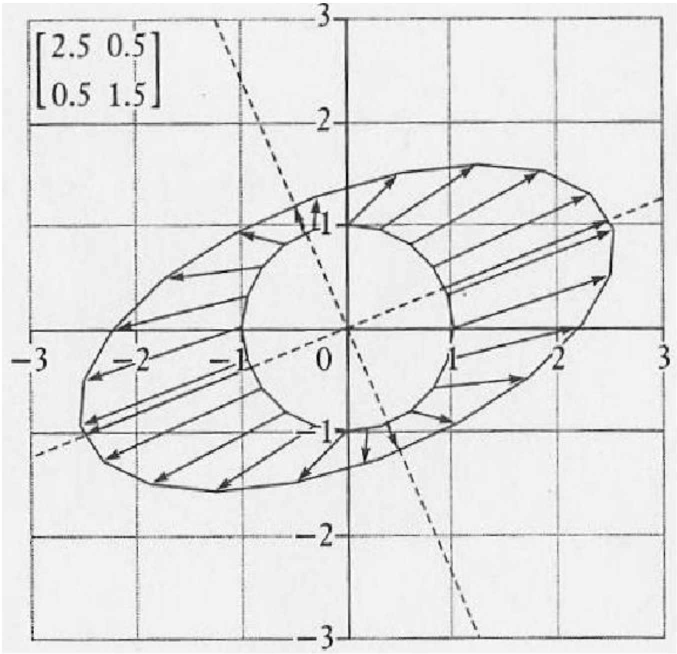
Eigenvector(s)

$S \cdot e_2 = \lambda_2 \cdot e_2$

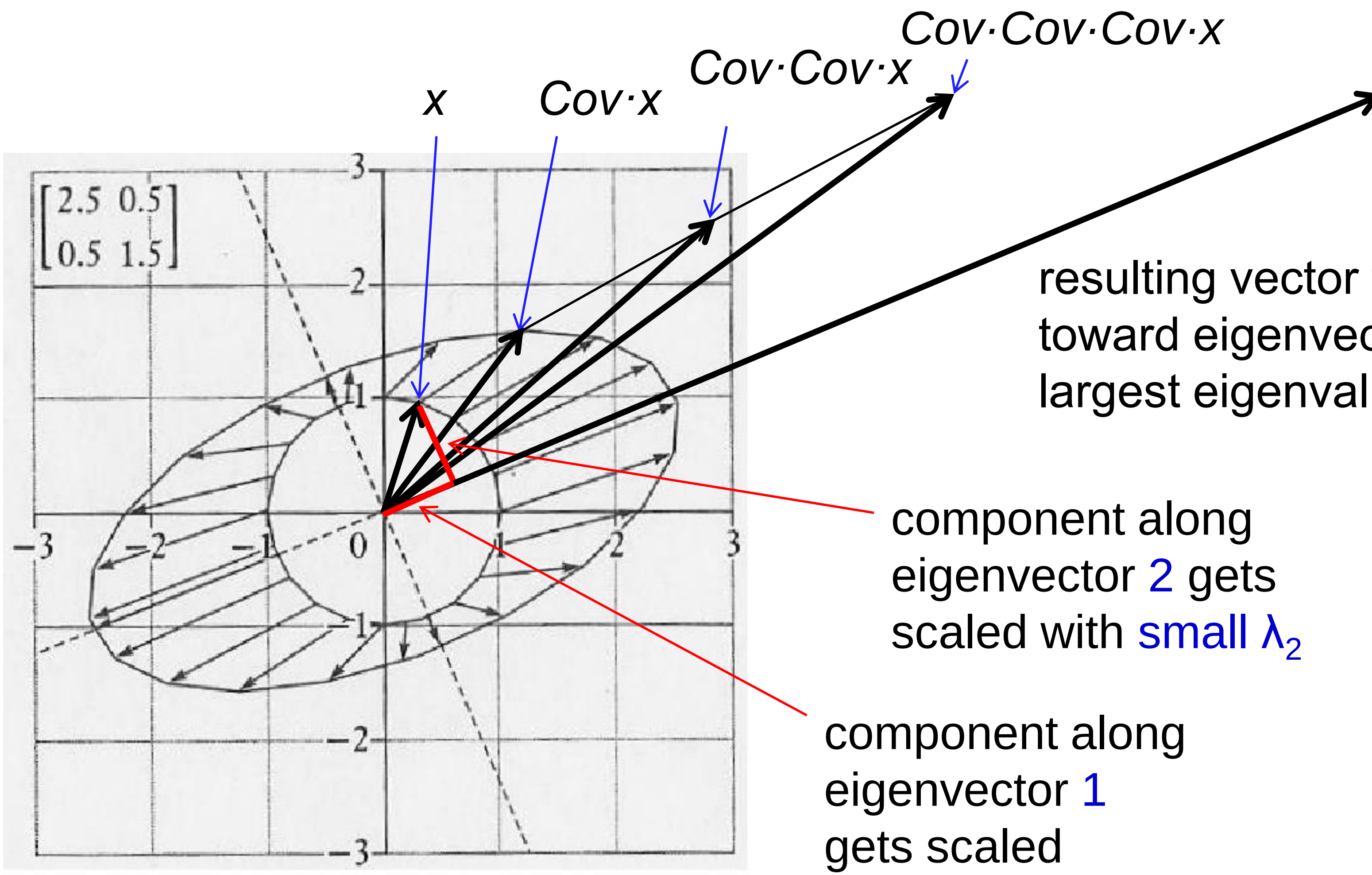
small eigenvalue λ_2

Properties of the Covariance Matrix

- The covariance matrix *Cov* is a symmetric matrix
  - It always has eigenvectors
  - Eigenvectors are orthogonal to each other, if their eigenvalues differ
- Its eigenvalues are ≥ 0
  - *Cov* is positive (semi-) definite
- Eigenvalues are variances of the data in the directions of the corresponding eigenvectors
- Multiplying by *Cov* does not rotate any vector by more than 90°



Repeatedly Multiply with the Covariance Matrix

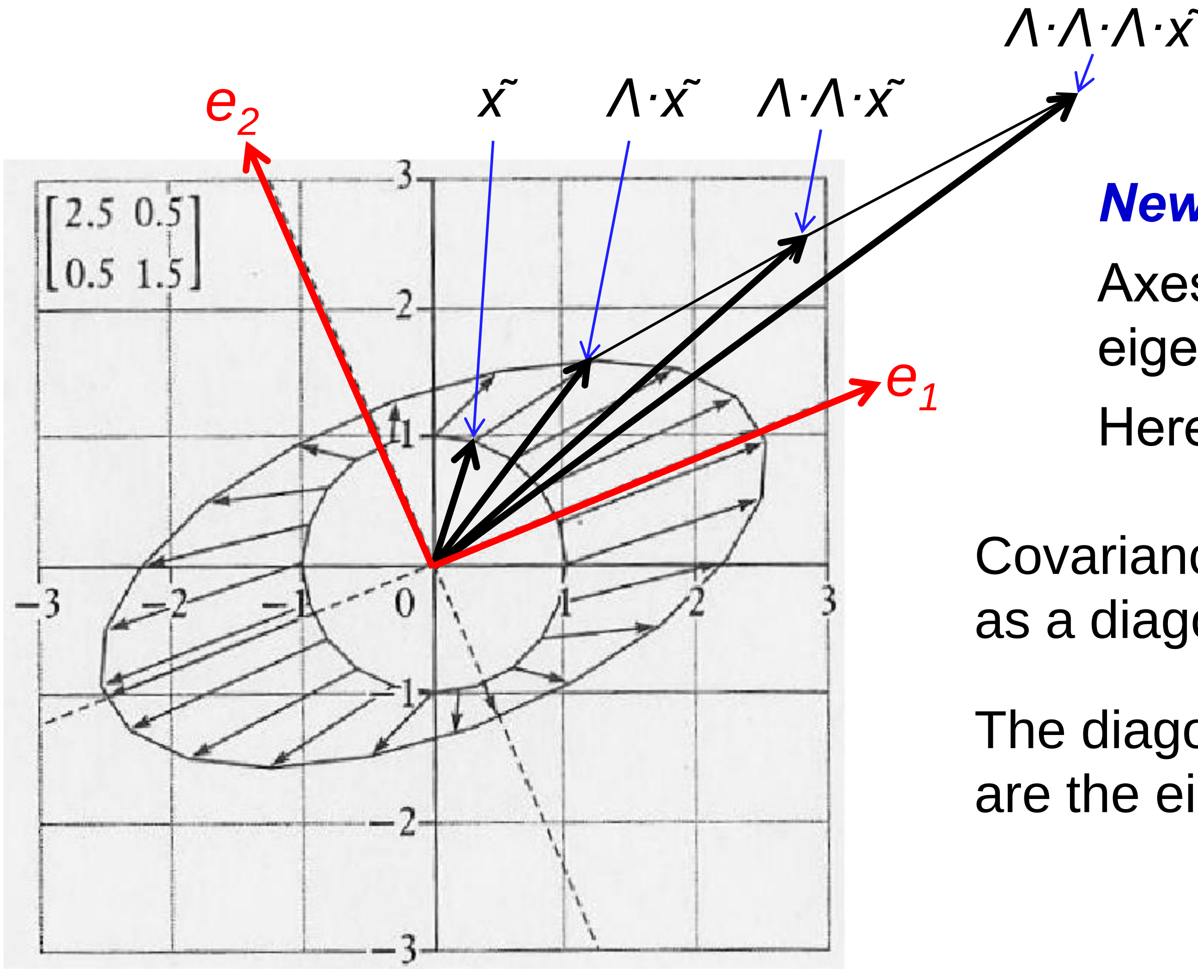


resulting vector turns toward eigenvector with largest eigenvalue, λ_1

component along eigenvector 2 gets scaled with small λ_2

component along eigenvector 1 gets scaled with **large λ_1**

Repeatedly Multiply with the Covariance Matrix



New coordinate system:

Axes defined by the eigenvectors e_i

Here: $x \rightarrow \tilde{x}$

Covariance matrix *Cov* expressed as a diagonal matrix Λ

The diagonal elements of Λ are the eigenvalues λ_i

Easy to see: individual dimensions/components *i* of $\tilde{x}$ are multiplied by λ_i

Obtain Largest Eigenvalue and Eigenvector

Iterative method

- Choose an initial random vector x
- Repeat
  - $x \leftarrow Cov \cdot x$
  - Normalize x to length 1
- Until converged. Then ***normalized eigenvector*** $e_1 = x$

Compute corresponding ***eigenvalue*** as the norm:

$$\lambda_1 = | Cov \cdot e_1 |$$

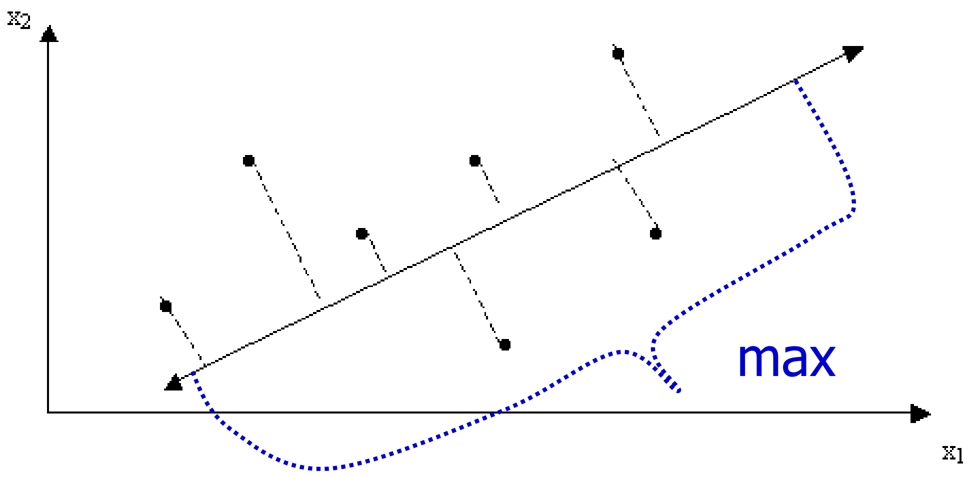
- Other linear algebra methods, e.g. SVD, compute ***all*** eigenvectors and eigenvalues.

Dimensionality Reduction by PCA

- The data can be expressed in the new coordinate system

$$x = \bar{x} + \sum_{j=1}^{m} w_j \cdot e_j$$

- w_j are the data coordinates along the eigenvector axes
- $m < D$: data are only approximately reconstructed



The first principal component is the axis in the direction of maximum variance.

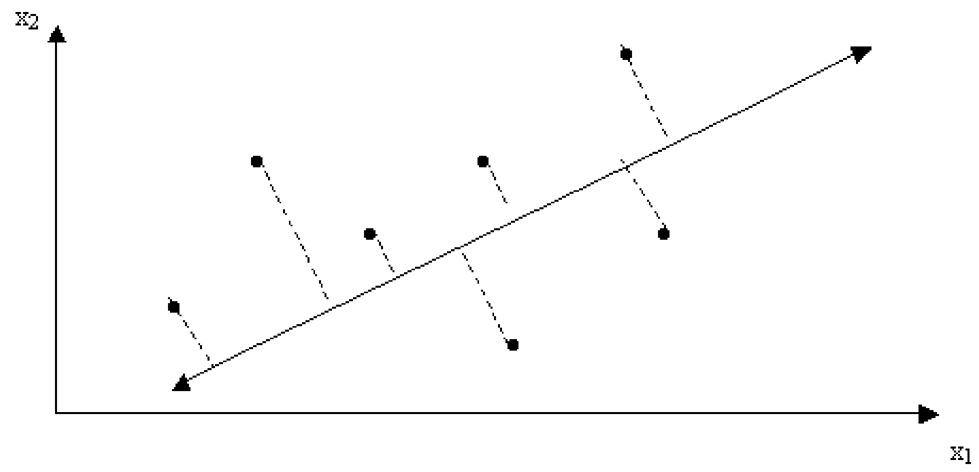
Dimensionality Reduction by PCA

- The criterion for features selection is based on the ratio *R* of the sum of the *m* largest eigenvalues (*m*≤*D*) of *Cov* to the trace of *Cov* (for example R>90%):

$$R = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{D} \lambda_i}$$

sum over all *explained* variances

Trace of *Cov*
= sum over *all* variances



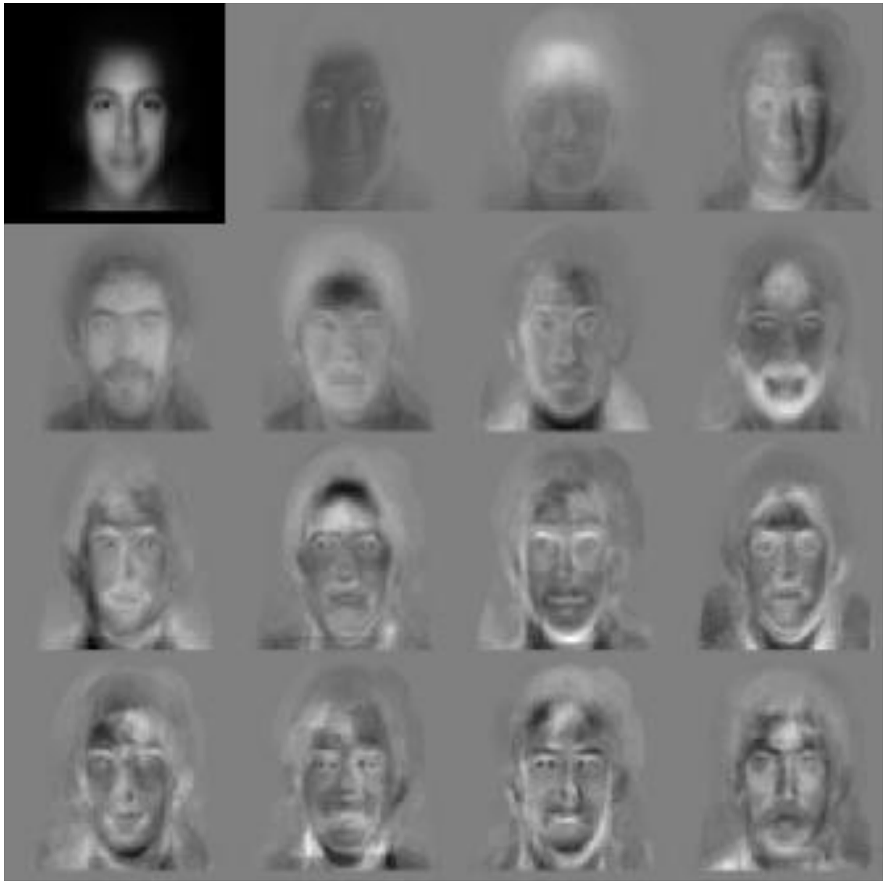
- Benefit: non-explained variance $\sum_{i=m+1}^{D} \lambda_i$ may be small, even if *m* << *D.*

Principle Components Example: Eigenfaces

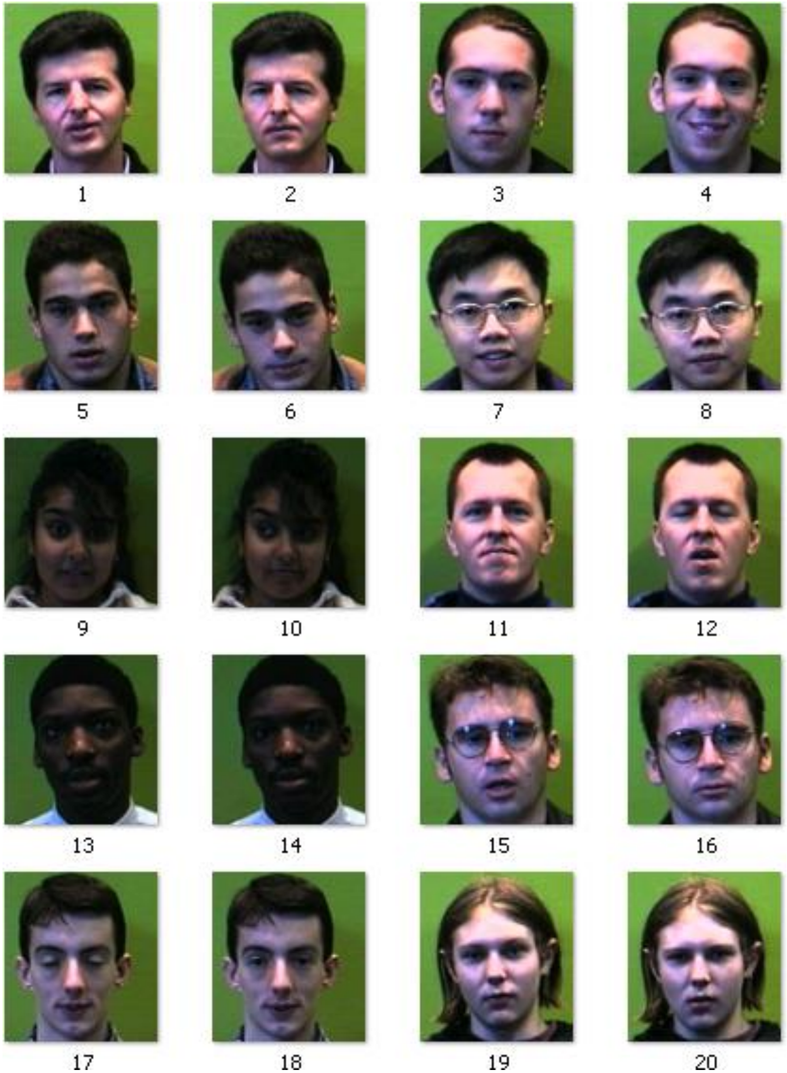
Eigenfaces are the (first few) principle components of a face image database.

reconstructed face

eigenfaces



A Practical Example with Eigenfaces (1/3)



- Data set
  - 20 images (too few …)
  - color (will be converted to grey scale)
  - controlled position and light
  - uniform background
  - size: 180 x 200 pixels: each data point is a 36000-dimensional vector

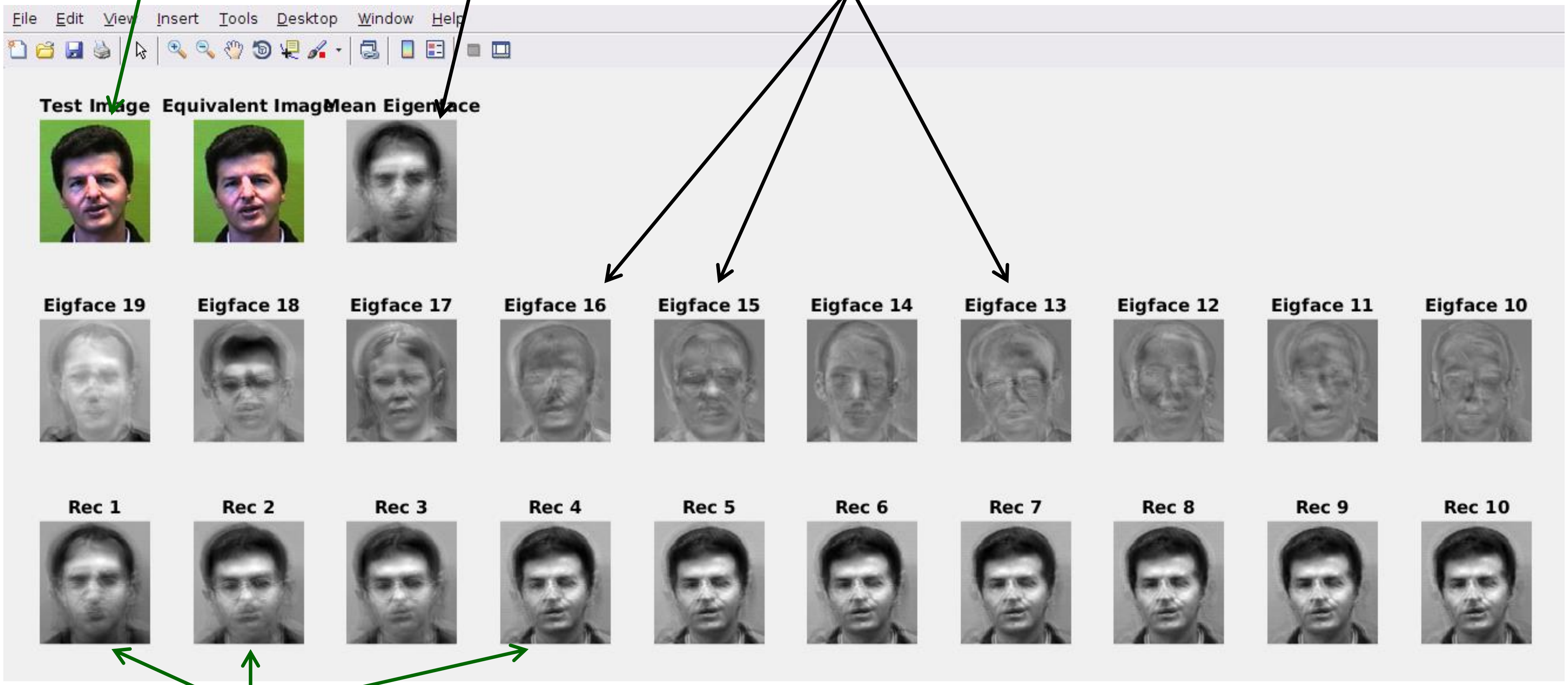
A Practical Example with Eigenfaces (2/3)

- Steps (in Matlab, but similar in Python)
  - imread
  - rgb2gray
  - reshape (img -> 1D row vector)
  - mean
  - p = data - mean (for every data point)
  - Cov = p^T p (covariance matrix, symmetric)
  - eig(Cov) (-> eigenvalues & eigenvectors)
  - sort
  - reshape (1D -> img) for display

mean vector of all data points

A Practical Example with Eigenfaces (3/3)

test image mean eigenvectors



reconstruction of test image from mean plus with 1, 2, 3, … eigenvectors

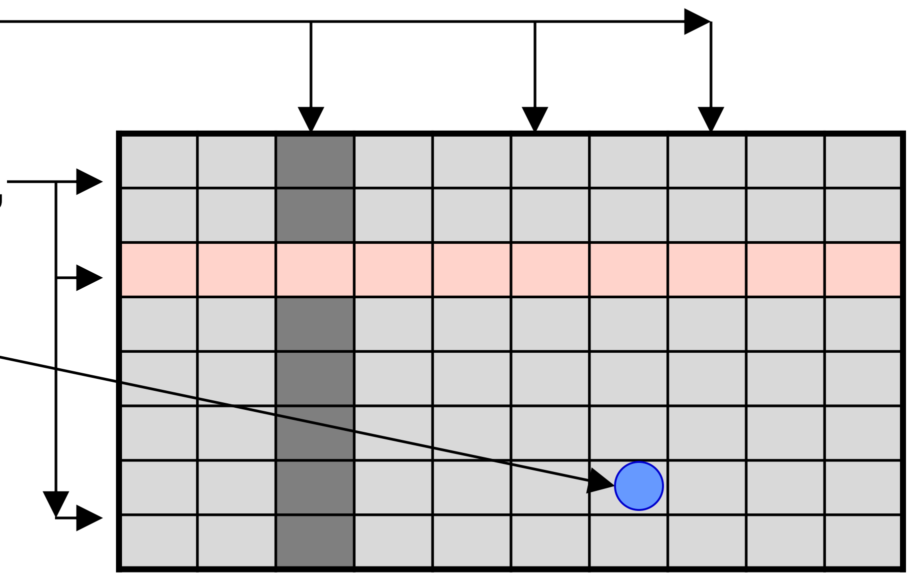
Dimensionality Reduction Techniques

- Features Reduction
- Correlation Analysis
- Data Transformation
  - Normalization
  - PCA
- Sampling

Dimensions Reduction of Large Data Sets

Main dimensions:

- ***columns*** (features),
- ***rows*** (cases or samples),
- ***values*** of the features for the given sample



Cases Reduction: Sampling

- Sampling: obtaining a small sample *S* to represent the whole data set *N*

- Key principle: Choose a ***representative*** subset of the data

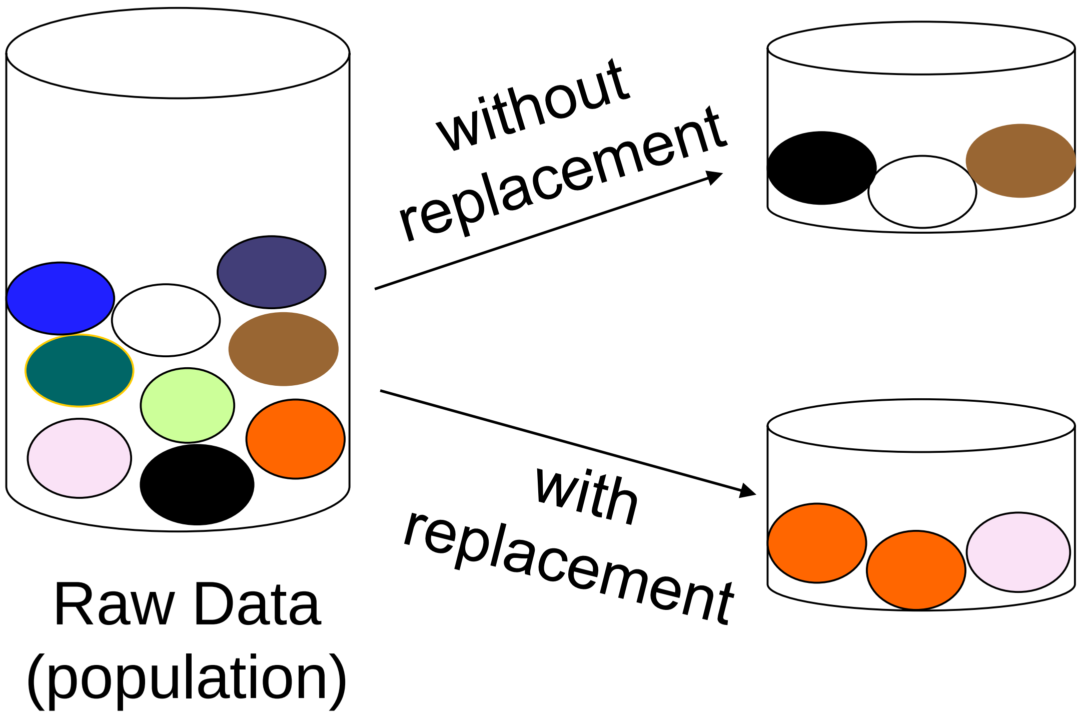
  - Using a *representative* sample will work almost as well as using the entire data set

  - A sample is representative if it has approximately the ***same property (of interest) as the original set*** of data

Types of Sampling

- ***Systematic sampling*:**
  - For example 50% of a data set (every second sample)
  - Simplest
  - Problem: regularities in data set!
- ***Random sampling***
  - There is an equal probability of selecting any particular item
  - ***Sampling without replacement***
    - Once an object is selected, it is removed from the population
  - ***Sampling with replacement***
    - A selected object is not removed from the population
- ***Stratified sampling***

Sampling With or Without Replacement

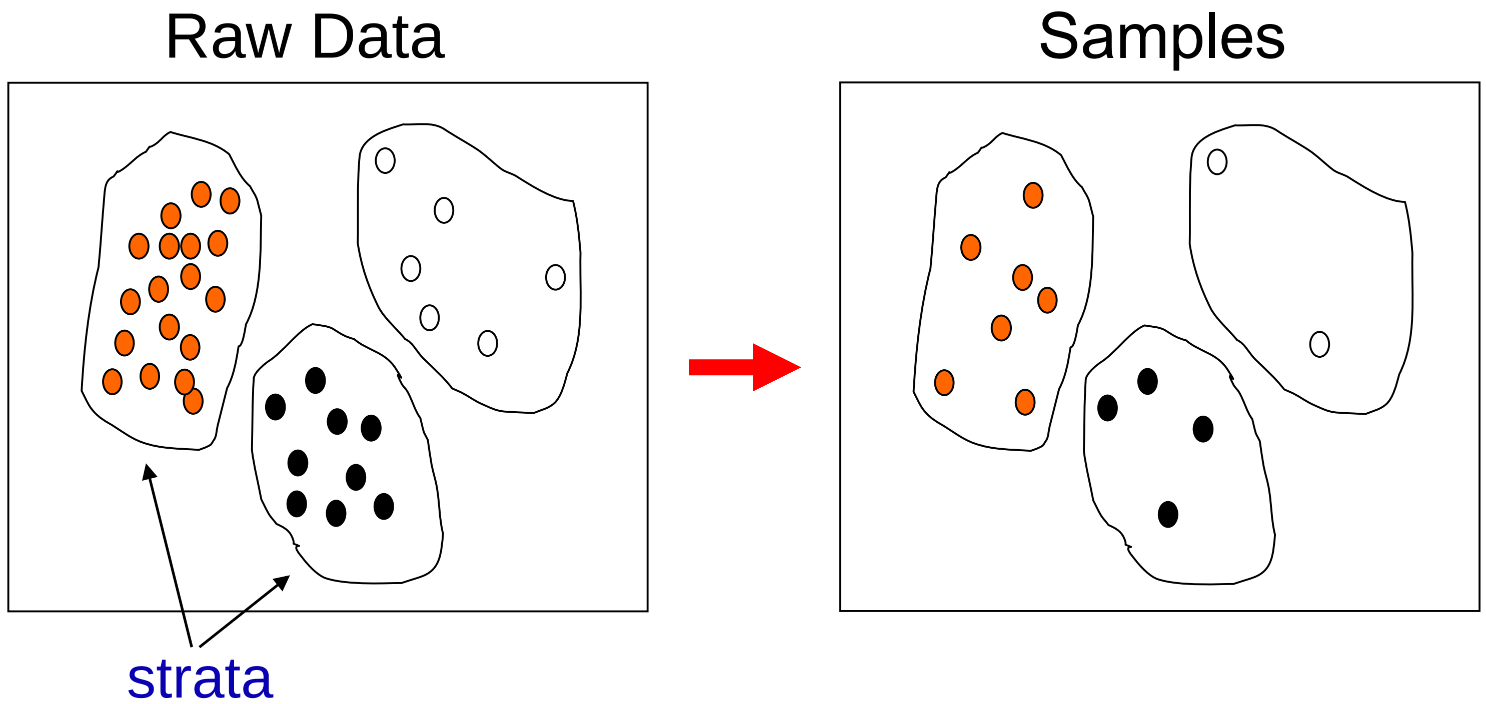


Once an object is selected, it is removed from the population

A selected object is not removed from the population

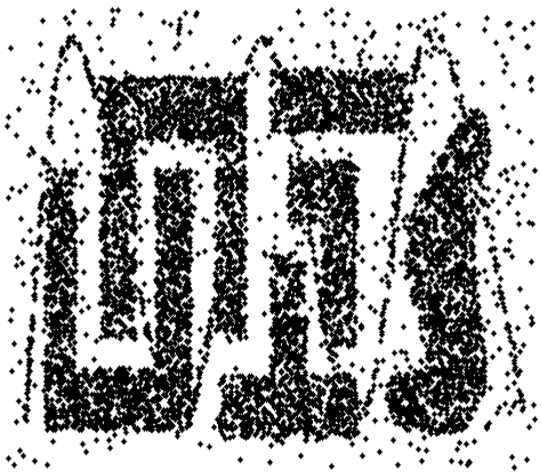
Stratified Sampling

- Partition the data set into strata (non-overlapping)
- Draw samples from each partition proportionally to its percentage in the data – important for skewed data

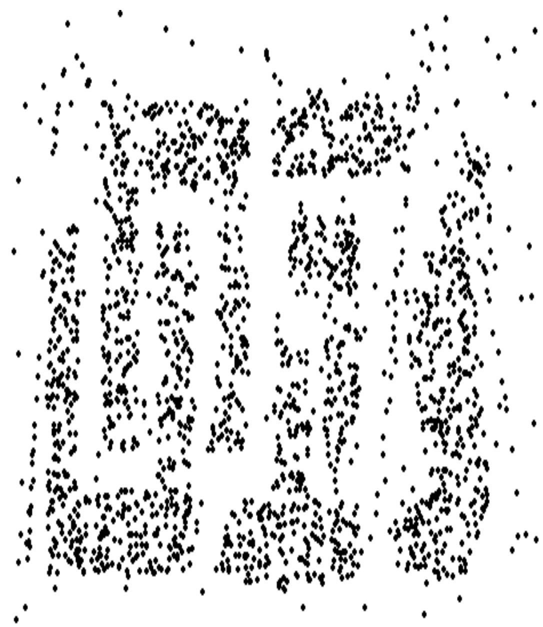


Cases Reduction: Sample Size

8000 points



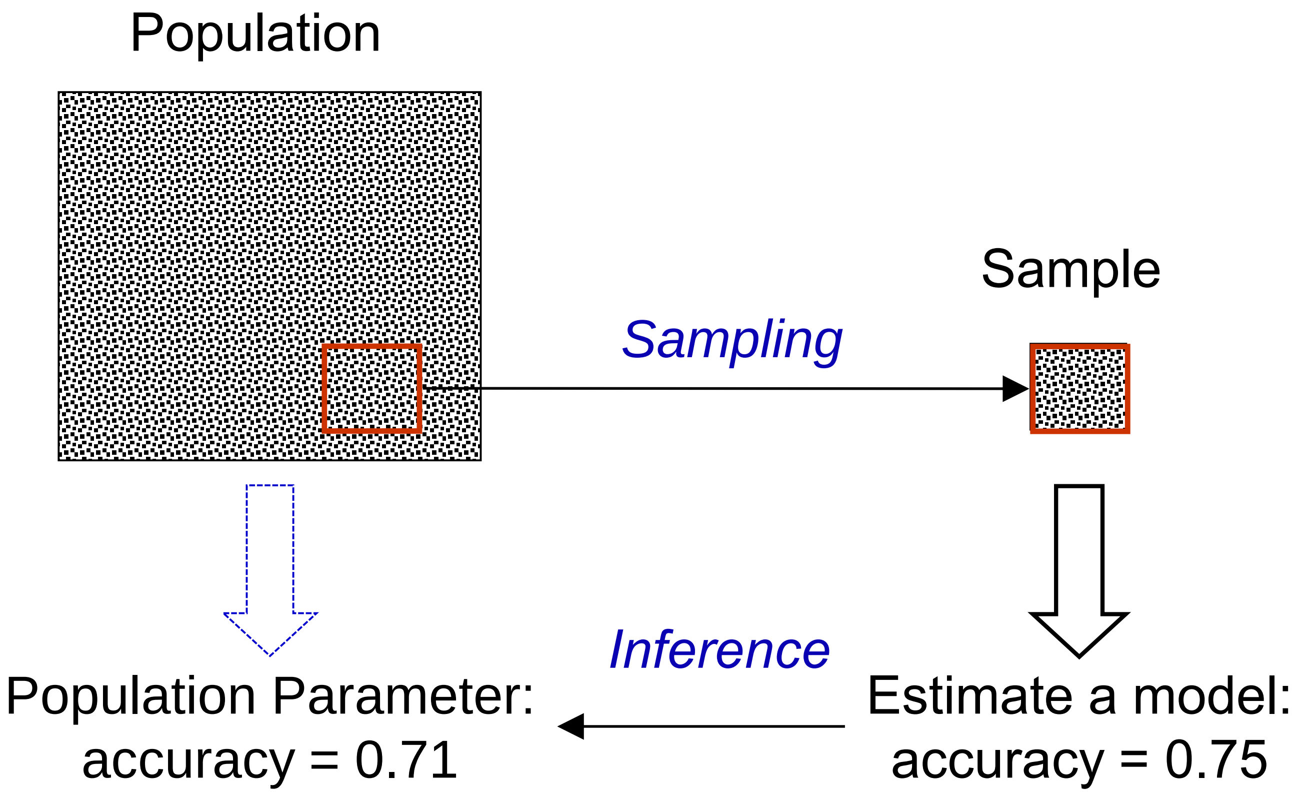
2000 Points



500 Points

Cases Reduction: Accuracy Parameter Estimation

- ***Challenging task***: Infer the value of a population parameter based on a sample model.



Summary of Data Reduction

- ***Values reduction***
  - Chi-merge
  - Binning
- ***Feature reduction***
  - Feature selection
  - Feature extraction/transformation: PCA
    - (Transformation: Normalization)
- ***Numerosity reduction***
  - Sampling