Machine Learning

4 – Basics of Data Mining

SS 2018

Gunther Heidemann

Overview

1. Data preprocessing
   - Data format
   - Outlier detection
   - Missing values
2. Similarity measures

Data preprocessing

From now on, we mainly deal with continuous valued attributes.

Why do we need data preprocessing?

- Machine needs unique data format
- Outliers should be detected
- Missing values should be filled in

Unique data format

Aim:

Convert data of different formats and sources to a common format.

We will only outline the problem and not provide solutions.

Problem: Differing formats of attributes, e.g.,

Universität Osnabrück, Wachsbleiche 27

Univ. Osnabrück, D-49069, Postfach

A. Schulze

Schulze, Andrea

Dr. med. Schulze

Andrea Paulina Ingeborg Ottilie Schulze-Hinterwimmer

26 % - 0,26 - 0.26 - >1/4

Unique data format

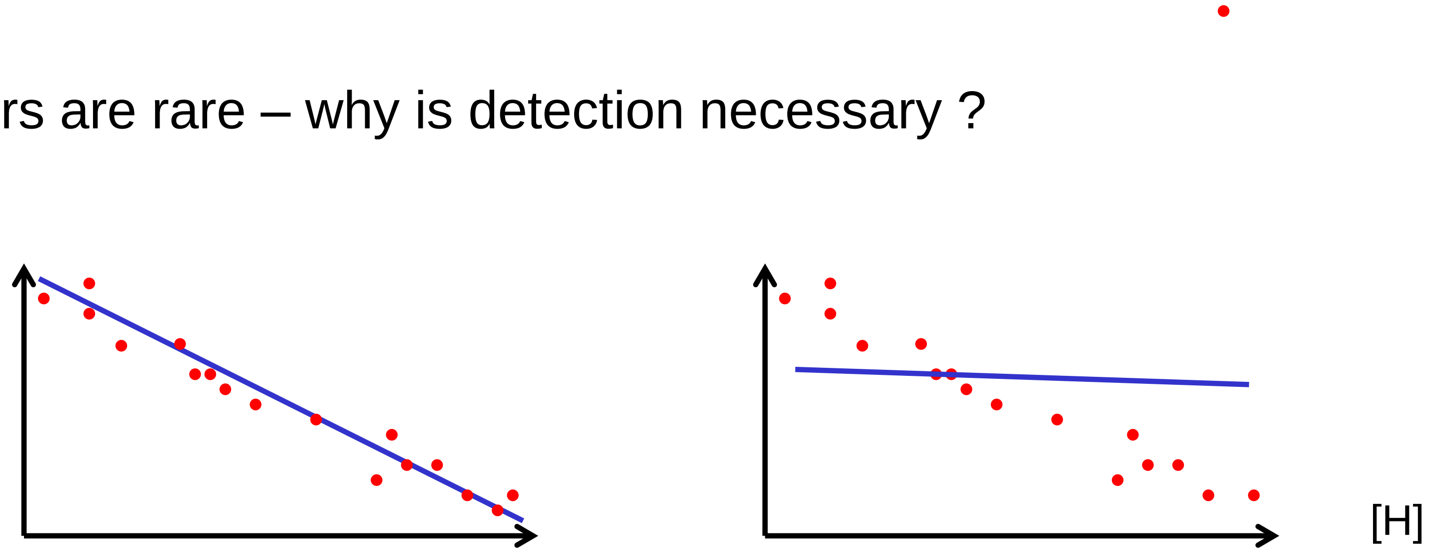
- Unique identifiers such as the US social security number are often missing.
- Free formats
- Missing attributes
- Irrelevant additions
- Differing scales
- Systematic changes, e.g, change of name after marriage.
- Data formatted for different purpose. Example: Data records of persons, but you need records of households.

Major problems in data processing!

To date: Only problem specific solutions.

Outliers

Outliers are rare – why is detection necessary ?



Trend of the regression line is spoiled by a single outlier

Outliers may have extreme values (often due to technical reasons) which can spoil statistics, in particular for small data sets.

Example:

Mean value μ : 1,4 1,6 1,3 1,2 1,4 1,2 1,3 → $\mu \approx 1{,}34$.

Additional value: 7,3 → $\mu \approx 2{,}09$.

Robustness against outliers

Some measures provide robustness against outliers even without explicit outlier detection.

Example: Replace mean by median *m* :

1,4 1,6 1,3 1,2 1,4 1,2 1,3 → $\mu \approx 1{,}34$.

1,2 1,2 1,3 1,3 1,4 1,4 1,6 → $m = 1{,}3$.

Outlier:

1,4 1,6 1,3 1,2 1,4 1,2 1,3 7,3 → $\mu \approx 2{,}09$.

1,2 1,2 1,3 1,3 1,4 1,4 1,6 7,3 → $m = 1{,}35$.

Types of outliers

Causes of outliers:

1. Errors by measurement / technical errors
2. Unexpected “true” effect
3. Data with high variation – outliers are a natural part of the distribution

Effect related to 1:

Cut-off, e.g., limited range of measurement leads to high concentration of values at the boundaries.

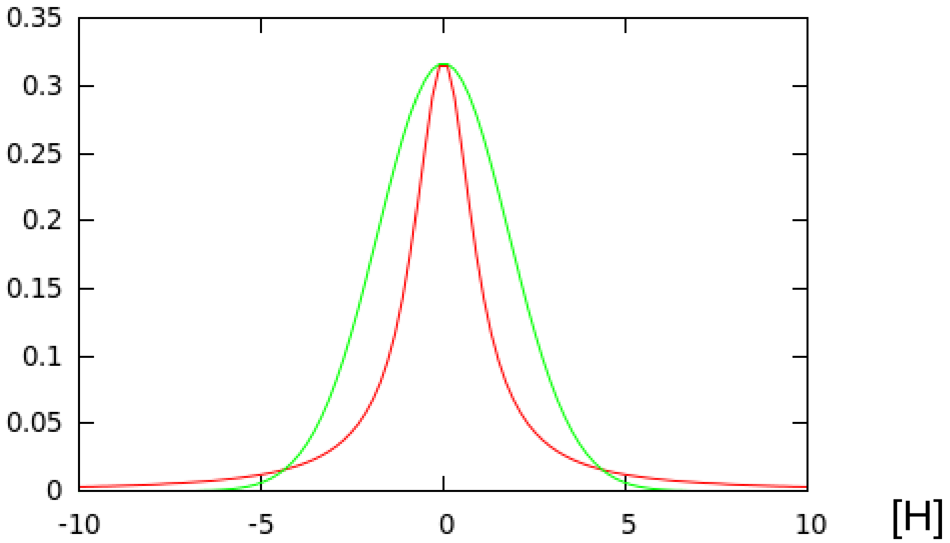
Problem 2 can be modeled by two or more overlaid distributions, e.g.,

$$P(x) = (1 - p) \cdot P_a(x) + p \cdot P_b(x), \qquad p \ll 1.$$

Problem 3 can be modeled by a distribution with broad flanks, e.g.,

$$P(x) = 1 / (\pi(1+x^2)).$$

University of Osnabrück, Institute of Cognitive Science



$$1\ /\ (\pi(1+x^2))$$

$$1\ /\ ((2\pi)^{½}\,\sigma)\cdot\exp(-½\,(x/\sigma)^2),\quad \sigma=1.26$$

Outlier detection

- Usually outliers are detected and removed.
- To consider a data point an outlier, we need to define what is *regular* !
- Most often a normal data distribution is assumed.
- For multivariate data, clustering algorithms can be applied and a normal distribution is assumed for each cluster.

z-test

Outliers of univariate distributions can be detected from z-values:

$$z_i = |x_i - \mu| / \sigma.$$

z_i is a measure for the distance of x_i from the mean μ in terms of the standard deviation σ.

Commonly, data with $z_i > 3$ are considered outliers.

Improvement:

Outliers influence μ, so use median instead. In this case, a threshold of 3,5 for outliers is used.

Rosner test

Detection of several outliers:

Idea: Iteratively remove outliers until z-tests finds no more.

1. Calculate mean μ or median m and standard deviation σ.
2. Find data point x_{i*} with largest z-value:

$$i* = \text{argmax}_i \; z_i.$$

3. If x_{i*} is an outlier, remove it from the data and goto 1.
4. Stop.

More efficient version:

Remove the $k > 1$ outliers with largest z-values in each step.

What to do with outliers

Options:

- Removal: Simple, but loss of information.
- Don’t remove outliers completely, but weight according to z-values.
- Remove and fill up gaps using the methods of the following section.

Missing values

Why are missing values a major problem?

Example:

Data set of vectors from $\Re^{100}$.

Probability that a value is missing is p = 2%.

Thus probability that a vector is complete is $(1 - p)^{100}$ = 13%.

87% of all vectors are unusable unless we compute substitutes !

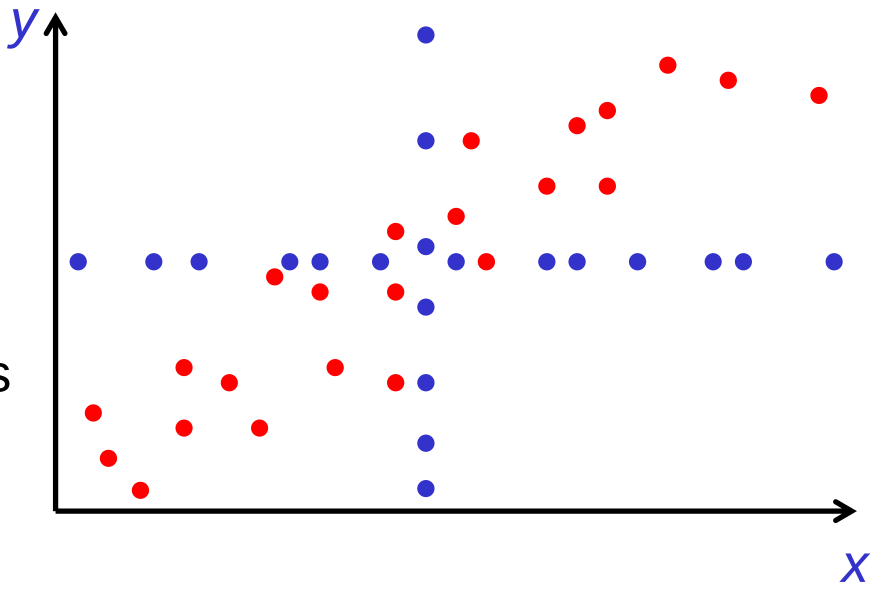
Problem: What can we do with missing values in a vector?

Missing values

Example:

Data $\{(x_i, y_i)\}$ show a clear trend.

How can missing values (i.e., x or y are missing) be filled in ?



[H]

Idea 1:

For data with only x, replace y by the mean or median μ_y from the rest of the data set (and vice versa) !

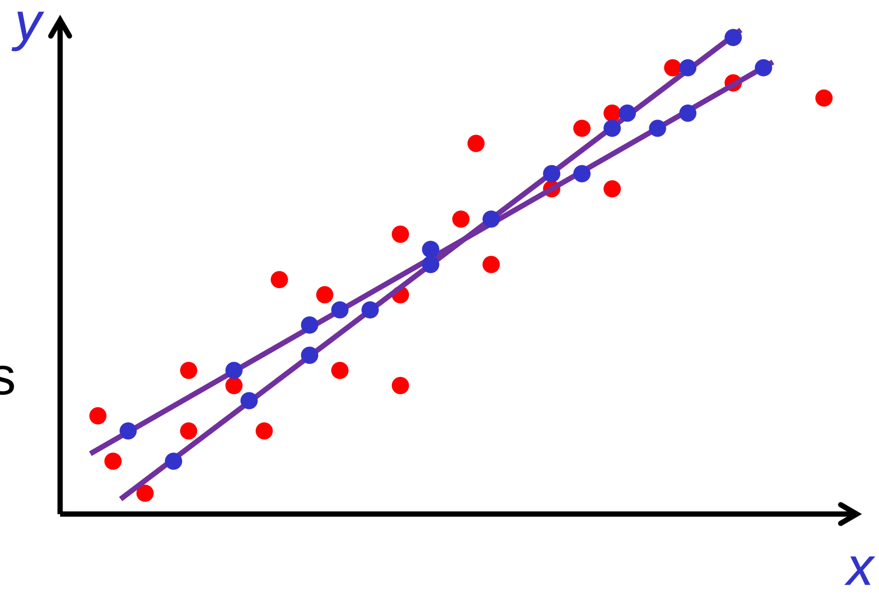
Effect: Artifacts.

Missing values

Example:

Data $\{(x_i, y_i)\}$ show a clear trend.

How can missing values (i.e., x or y are missing) be filled in ?



[H]

Idea 2: Estimate a model to predict missing values.

Example: Linear regression

$$y' = y_s \cdot x + y_0, \qquad y_s = C_{xy} / C_{xx}, \qquad y_0 = \mu_y - y_s \, \mu_x ,$$

$$x' = x_s \cdot y + x_0, \qquad x_s = C_{xy} / C_{yy}, \qquad x_0 = \mu_x - x_s \, \mu_y ,$$

where $C_{xy} = \sum_{i=1..n} (x_i - \mu_x) \cdot (y_i - \mu_y)$ is the covariance of x and y.

Effect: Artificial concentration of values along regression lines.

Question

Why two different regression lines?

$$y' = y_s \cdot x + y_0, \qquad y_s = C_{xy} / C_{xx}, \qquad y_0 = \mu_y - y_s\,\mu_x\,.$$

$$x' = x_s \cdot y + x_0, \qquad x_s = C_{xy} / C_{yy}, \qquad x_0 = \mu_x - x_s\,\mu_y\,.$$

Answer: Regression minimizes the mean square error of y depending on x (or alternatively x on y).

So y' minimizes

$$\sum_{i=1..n} (y_i - (y_s \cdot x_i + y_0))^2 \rightarrow \text{min},$$

but x' minimizes

$$\sum_{i=1..n} (x_i - (x_s \cdot y_i + x_0))^2 \rightarrow \text{min},$$

with n = # data.

Remark

Do not confuse

$$C_{xy} = \sum_{i=1..n} (x_i - \mu_x)\cdot(y_i - \mu_y)$$

and Pearsons correlation coefficient

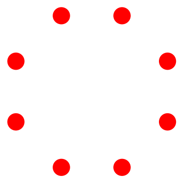
$$\rho_{xy} = C_{xy} \,/\, (\sigma_x\, \sigma_y),$$

with the standard deviation σ.

The correlation coefficient is a measure for linear dependence, taking values between –1 (anti-correlation) and 1 (maximum correlation).

0 indicates independence.

The absence of correlation $\rho_{xy} = 0$ does not mean there is no structure in the data:

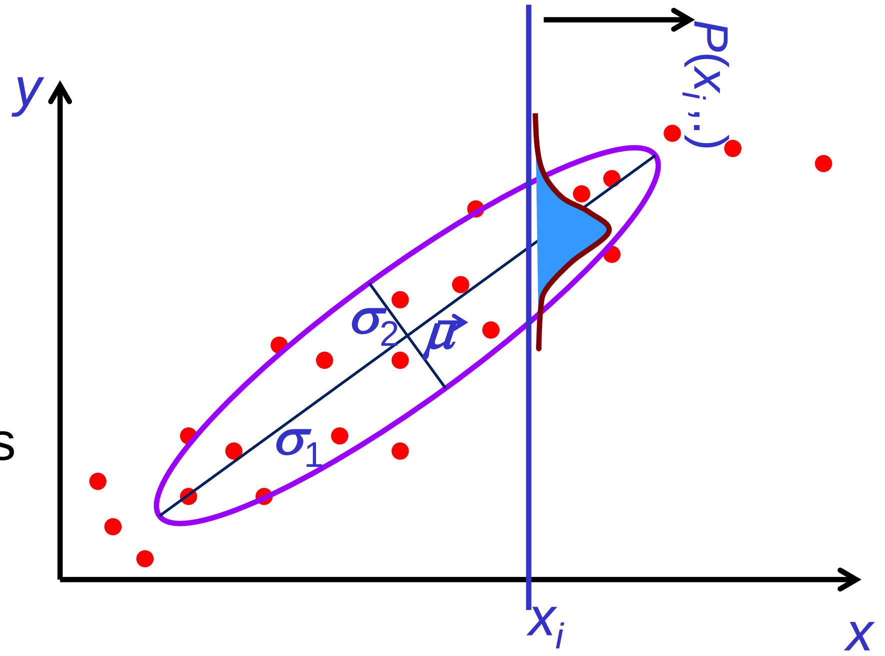


Missing values

Example:

Data $\{(x_i, y_i)\}$ show a clear trend.

How can missing values (i.e., x or y are missing) be filled in ?



[H]

Idea 3:

Estimate the data distribution $P(x,y)$, e.g., assuming a normal distribution $P(x,y) = N(\vec{\mu},C)$, C = covariance matrix.

Once we have estimated $\vec{\mu}$ and C, generate missing values x using a random number generator with probability distribution $P(. ,y)$. and missing y from $P(x, .)$.

Estimating the distribution

Problem when estimating $P(x,y)$:

We can use only the complete data vectors. The information of all data vectors with a missing value remains unused. This leads to

Idea 4: Estimate P using **Expectation Maximization** (**EM**) (Dempster, Laird, Rubin, 1977)

EM: Uses both complete and partial data vectors in an iterative procedure.

In the following, let

x denote all specified data (complete and partial vectors),

h all “hidden” (missing) values,

θ all parameters of the chosen distribution (such as mean and variances for a Gaussian).

EM algorithm

The probability of the known values depends on the distribution (specified by θ) as $P(x \mid \theta)$.

The hidden values h are subject to the probability distribution (and thus to θ) as much as the known values x.
In addition, the hidden values h depend on x: $P(h \mid x, \theta)$.

The total distribution is thus

$$P(x,h \mid \theta) = P(h \mid x,\theta) \cdot P(x \mid \theta).$$

The *likelihood* of parameters θ as a function of x and h is

$$L(\theta\,; x,h) = P(x,h \mid \theta).$$

For convenience, we consider the *log-likelihood* instead:

$$l(\theta) = \log L(\theta\,; x,h) = \log P(h \mid x,\theta) + \log P(x \mid \theta).$$

We want the parameters θ^* that maximize the log-likelihood $l(\theta)$.

But $l(\theta)$ depends on the hidden values h.

EM algorithm

We get $l(\theta) = \log P(x,h,\theta) = \log P(h \mid x,\theta) + \log P(x \mid \theta)$ by

- removing the hidden values h by “averaging out” to obtain an averaged $<l(\theta)>_h$.
- To do so, we need the probability $P(h \mid x,\theta)$,
- but this probability depends on the unknown θ.
- So we need an estimate θ_t for the real θ.

The dilemma is solved by iteratively improving the estimate θ_t for the real θ (*M-step*) and averaging over h using the obtained θ_t (*E-step*). θ_t will converge to a local maximum θ^* of l (hopefully close to θ).

Thus we maximize the averaged likelihood

$$Q(\theta,\theta_t) = <l(\theta)>_h = \int P(h \mid x,\theta_t) \cdot \log P(h \mid x,\theta)\, dh + \log P(x \mid \theta)$$

So we have traded the h-dependence of l for a θ_t-dependence of Q.

EM algorithm

Procedure:

1. Choose a function to approximate $P(x,y \mid \theta)$ with parameters θ.
2. Choose start values θ_t.
3. Initialize step counter $t = 0$.
4. E-step: Calculate the integral of

$$Q(\theta,\theta_t) = \int P(h \mid x,\theta_t) \cdot \log P(h \mid x,\theta)\, dh + \log P(x \mid \theta)$$

   to obtain the function Q depending on θ and θ_t.

5. M-step: Maximize Q with respect to θ :

$$\theta_{t+1} = \arg\max_\theta Q(\theta,\theta_t).$$

6. t++.
7. If $Q(\theta,\theta_t)$ does not meet the convergence condition goto 4.

Steps 4 and 5 may include heavy numerics.

EM-algorithm: Proof of convergence

Show that the data log-likelihood $\log P(x|\theta)$ increases with each EM-step.

Let's use $\mathscr{P}_t$ for short for $P(h|x,\theta_t)$ and $\mathscr{P}$ for $P(h|x,\theta)$, so we get

$$Q(\theta,\theta_t) = \int P(h|x,\theta_t)\cdot\log P(h|x,\theta)\,dh + \log P(x|\theta) = \int\mathscr{P}_t\cdot\log\mathscr{P}\,dh + \log P(x|\theta).$$

Change Δ_t of $\log P(x|\theta) = Q(\theta,\theta_t) - \int\mathscr{P}_t\cdot\log\mathscr{P}\,dh$ under one EM-step:

$$\Delta_t := \log P(x|\theta_{t+1}) - \log P(x|\theta_t)$$

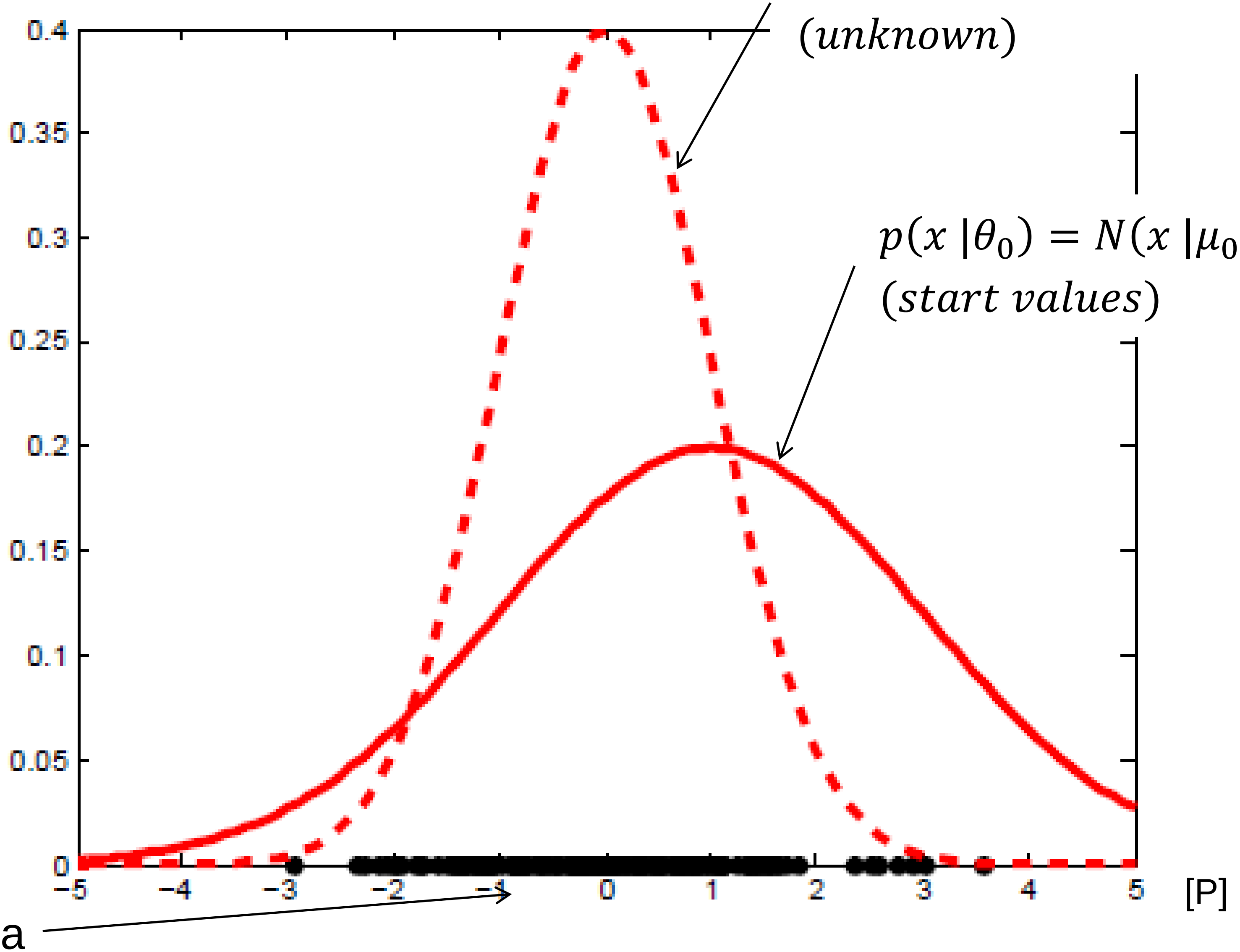
$$= Q(\theta_{t+1},\theta_t) - \int\mathscr{P}_t\cdot\log\mathscr{P}_{t+1}\,dh - Q(\theta_t,\theta_t) + \int\mathscr{P}_t\cdot\log\mathscr{P}_t\,dh$$

$Q(\theta_{t+1},\theta_t) - Q(\theta_t,\theta_t) =: \mathrm{Q}_t \geq 0$ holds for all t by definition of the M-step.

$$\Delta_t = \mathrm{Q}_t + \int\mathscr{P}_t\cdot\log\mathscr{P}_t\,dh - \int\mathscr{P}_t\cdot\log\mathscr{P}_{t+1}\,dh = \mathrm{Q}_t + \int\mathscr{P}_t\cdot\log(\mathscr{P}_t/\mathscr{P}_{t+1})\,dh \geq 0$$

holds for all t because the integral is the Kullback-Leibler divergence of $\mathscr{P}_t$ and $\mathscr{P}_{t+1}$, which is always non-negative (Gibbs inequality).

EM algorithm – 1D-Example

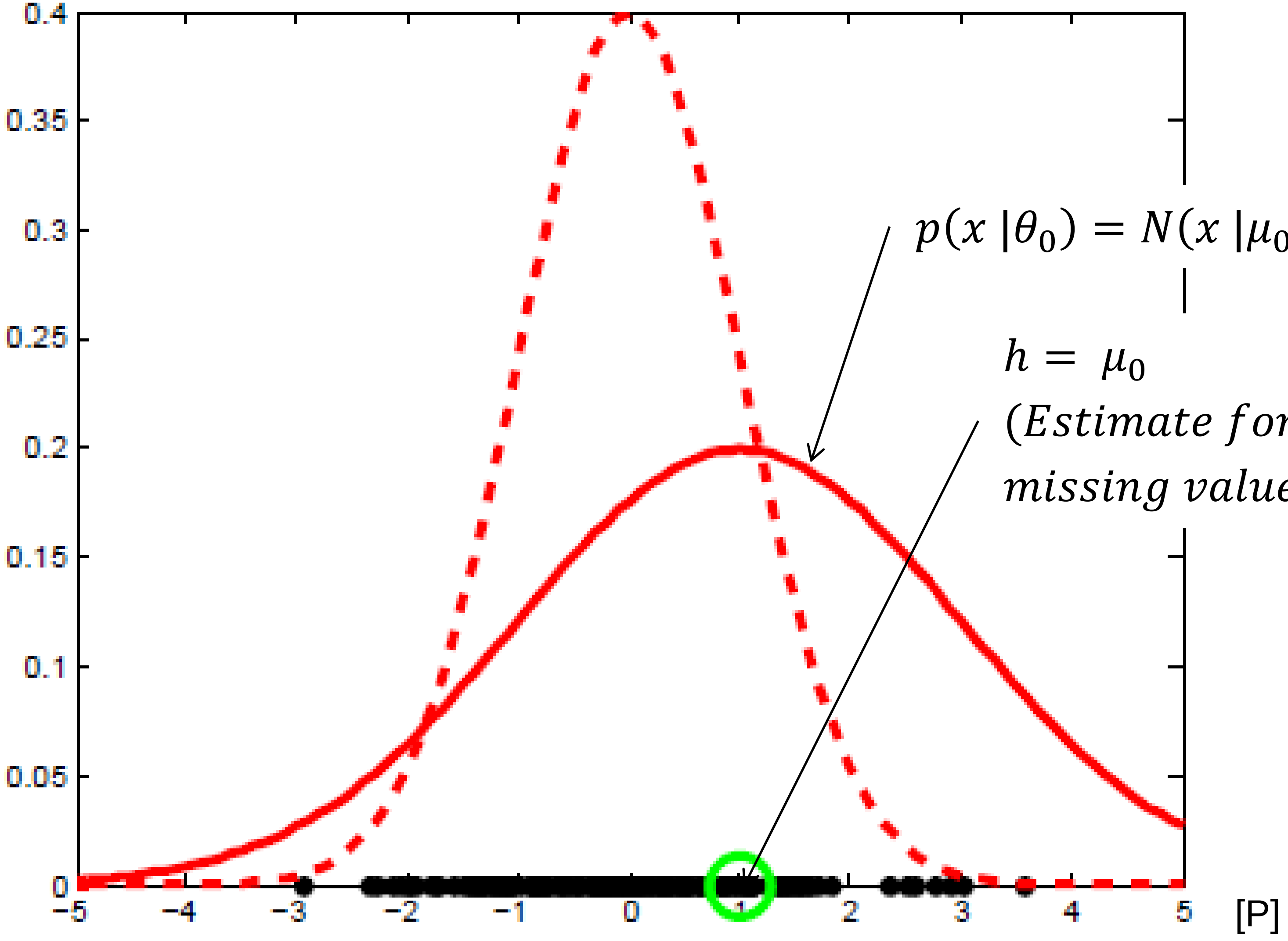


$p(x|\theta) = N(x\,|\mu, \sigma)$
$(unknown)$

$p(x\,|\theta_0) = N(x\,|\mu_0, \sigma_0)$
$(start\ values)$

Data

EM algorithm – Expectation step

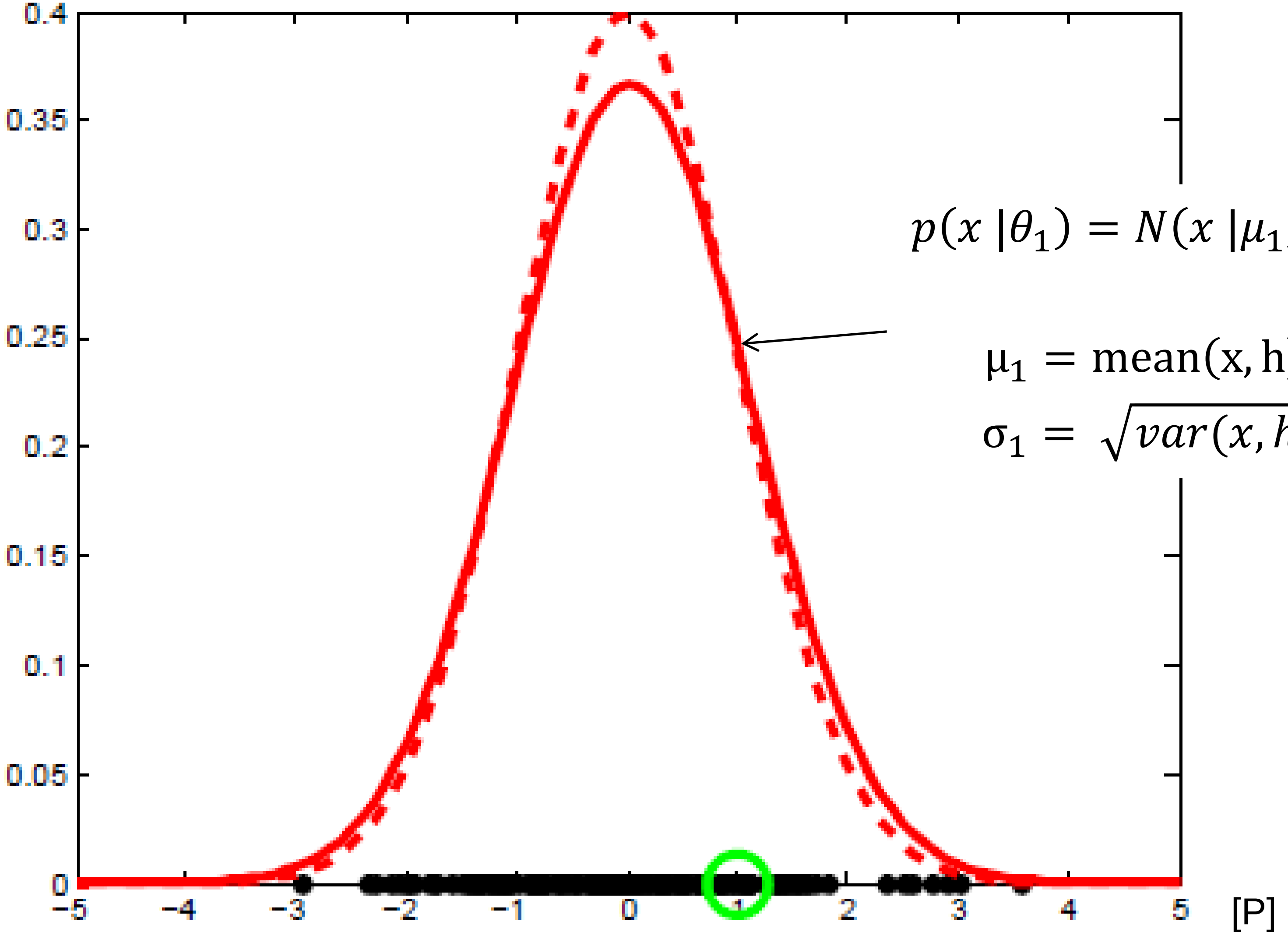


$p(x \mid \theta_0) = N(x \mid \mu_0, \sigma_0)$

$h = \mu_0$

$(Estimate\ for$
$missing\ value)$

EM algorithm – Maximisation step

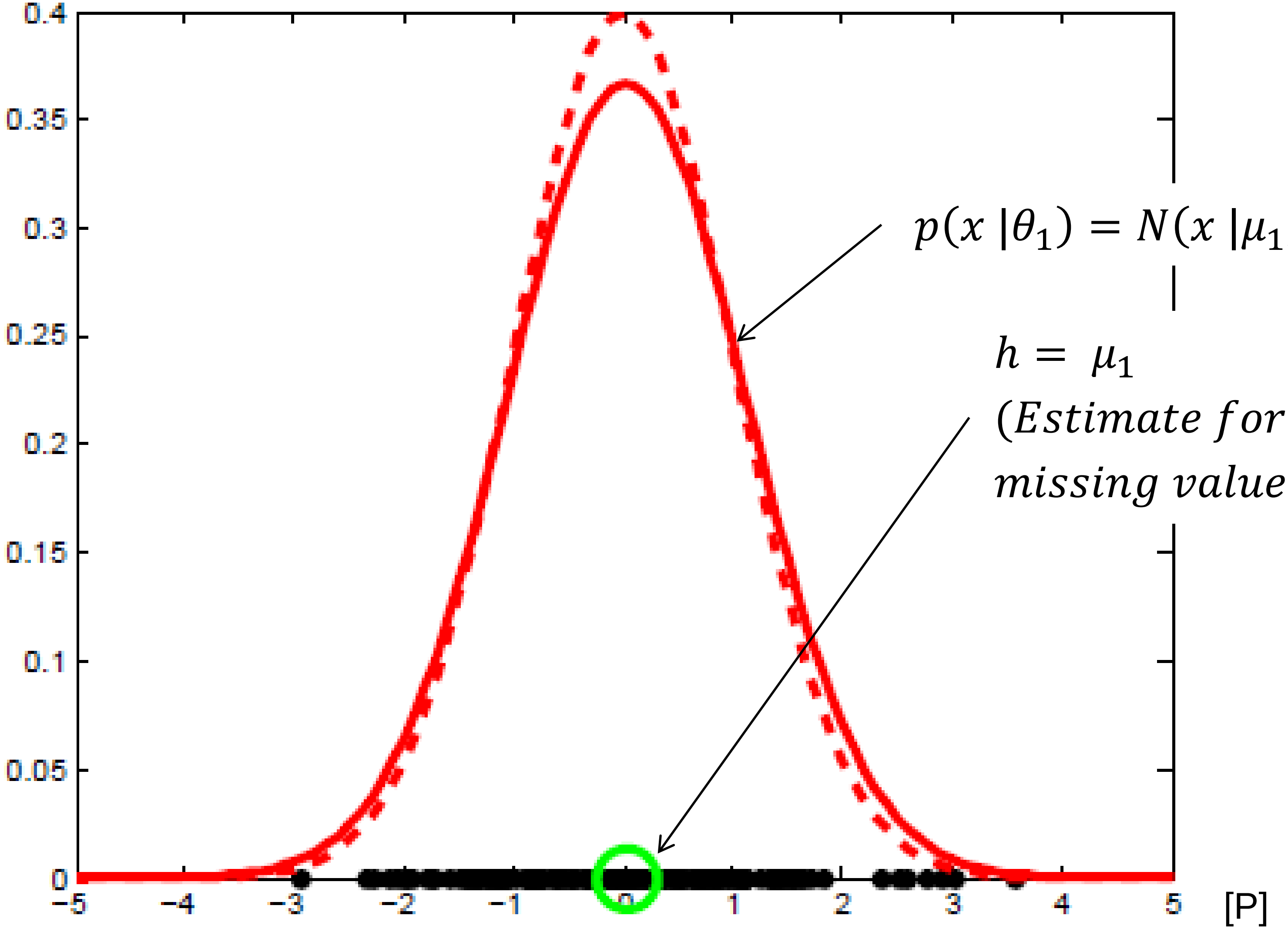


$$p(x\,|\theta_1) = N(x\,|\mu_1, \sigma_1)$$

$$\mu_1 = \mathrm{mean}(x, h)$$

$$\sigma_1 = \sqrt{var(x, h)}$$

EM algorithm – Expectation step 2

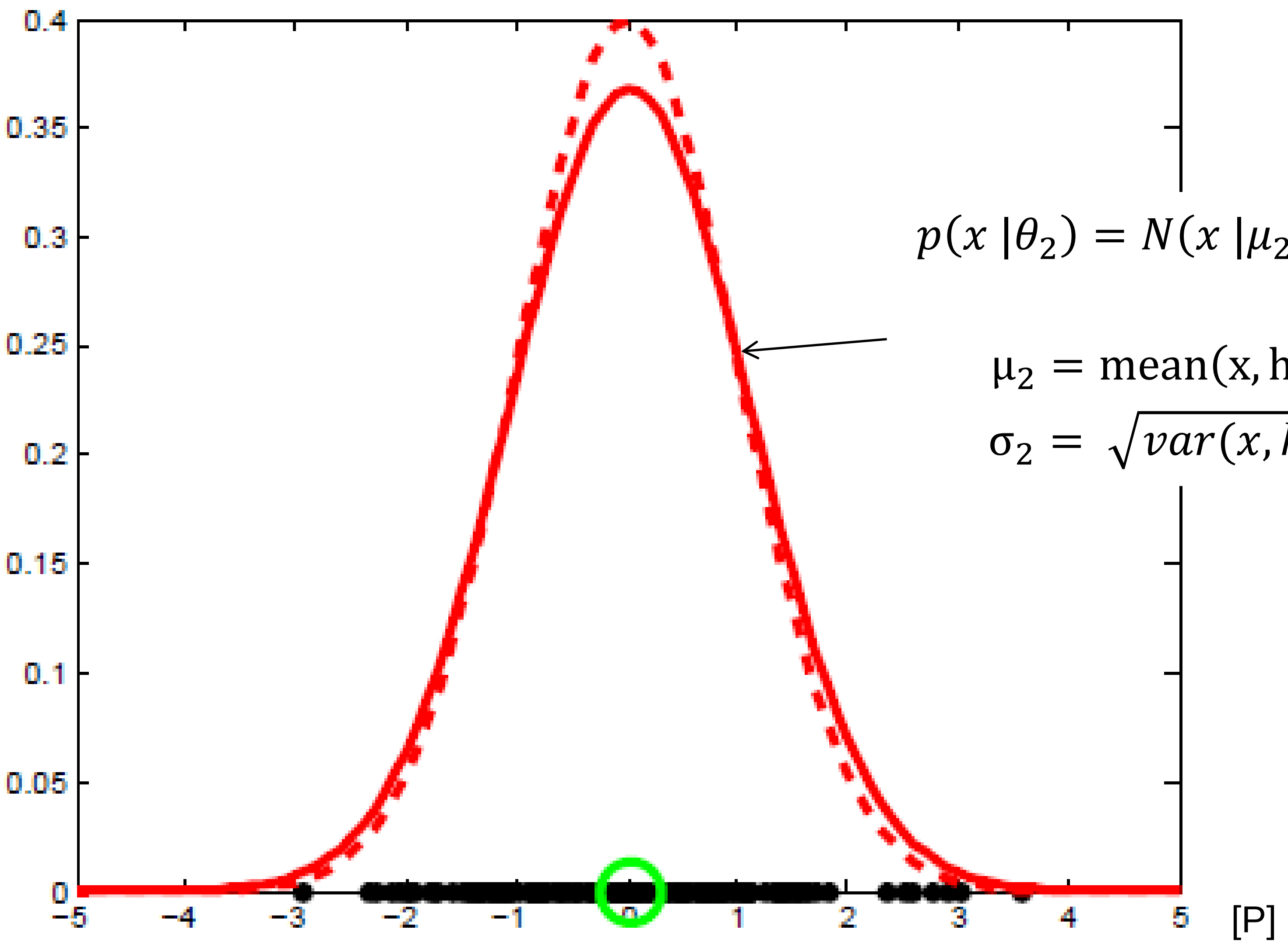


$p(x\,|\theta_1) = N(x\,|\mu_1, \sigma_1)$

$h = \mu_1$

$(Estimate\ for\ missing\ value)$

EM algorithm – Maximisation step 2



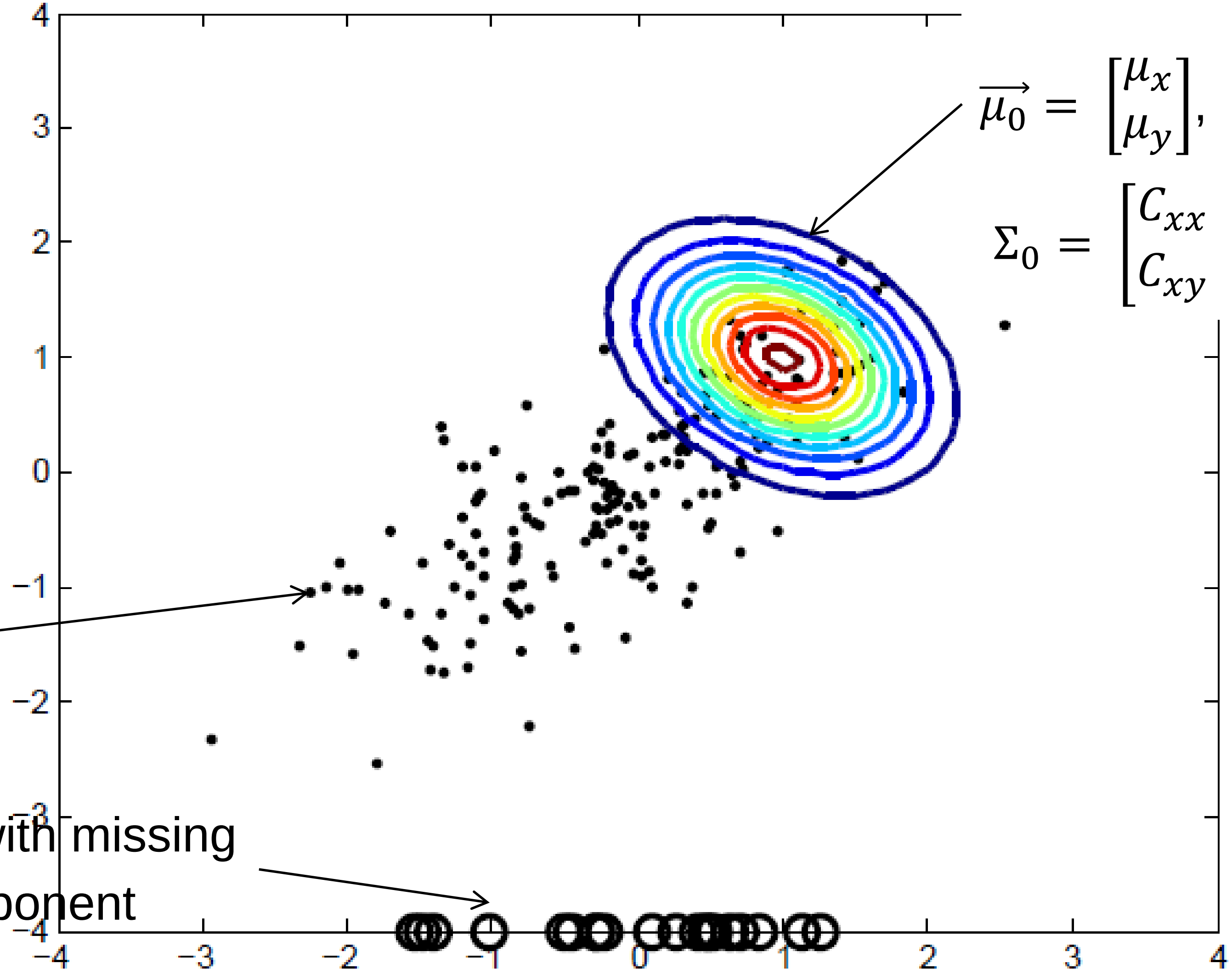
$$p(x \,|\theta_2) = N(x \,|\mu_2, \sigma_2)$$

$$\mu_2 = \mathrm{mean}(x, h)$$

$$\sigma_2 = \sqrt{var(x,h)}$$

EM algorithm – 2D Data

$$p(\vec{x}|\theta_0) = N(\vec{x}|\vec{\mu}_0, \Sigma_0)$$



$$\vec{\mu_0} = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix},$$

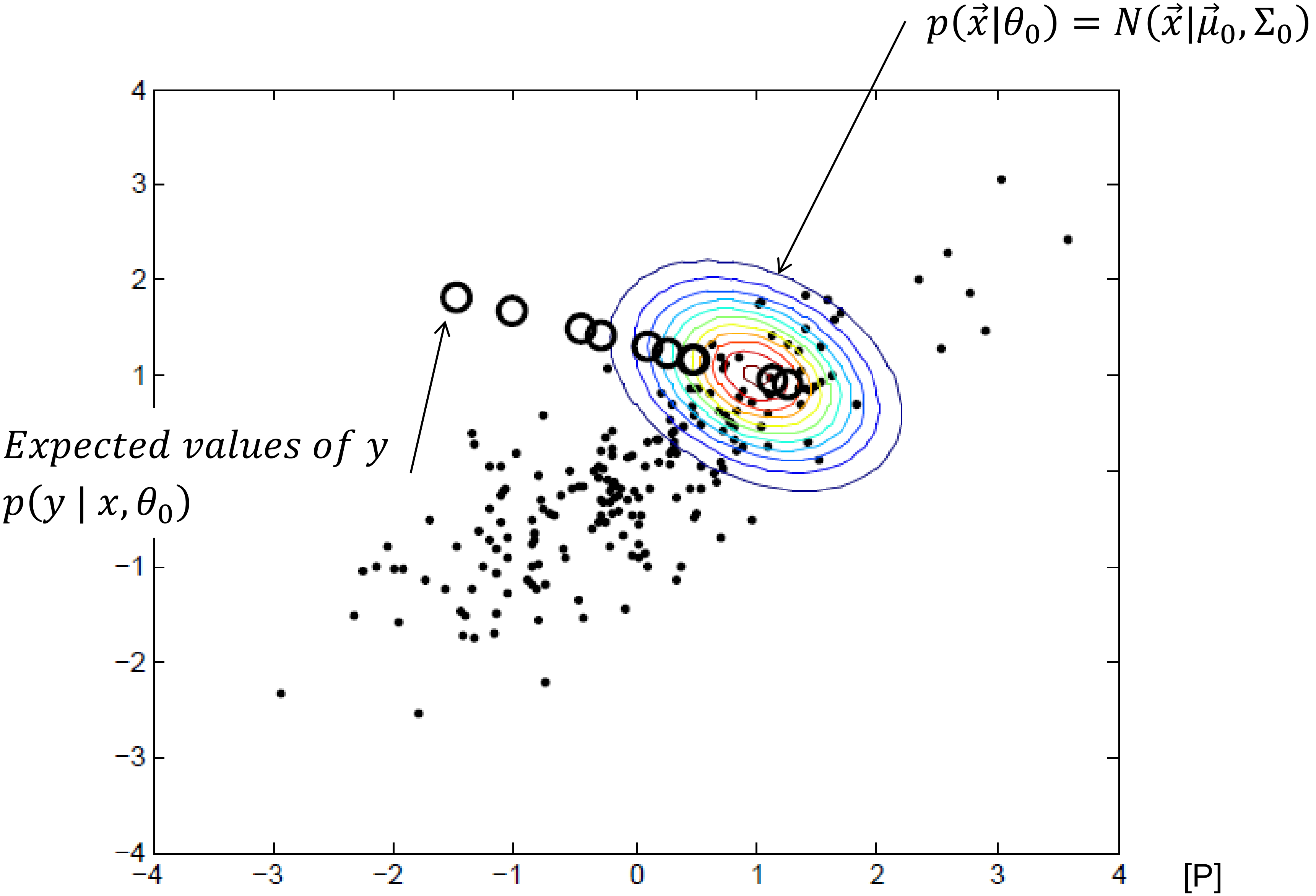
$$\Sigma_0 = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{xy} & C_{yy} \end{bmatrix}$$

Data

Data with missing y-component

[P]

EM algorithm – Expectation Step

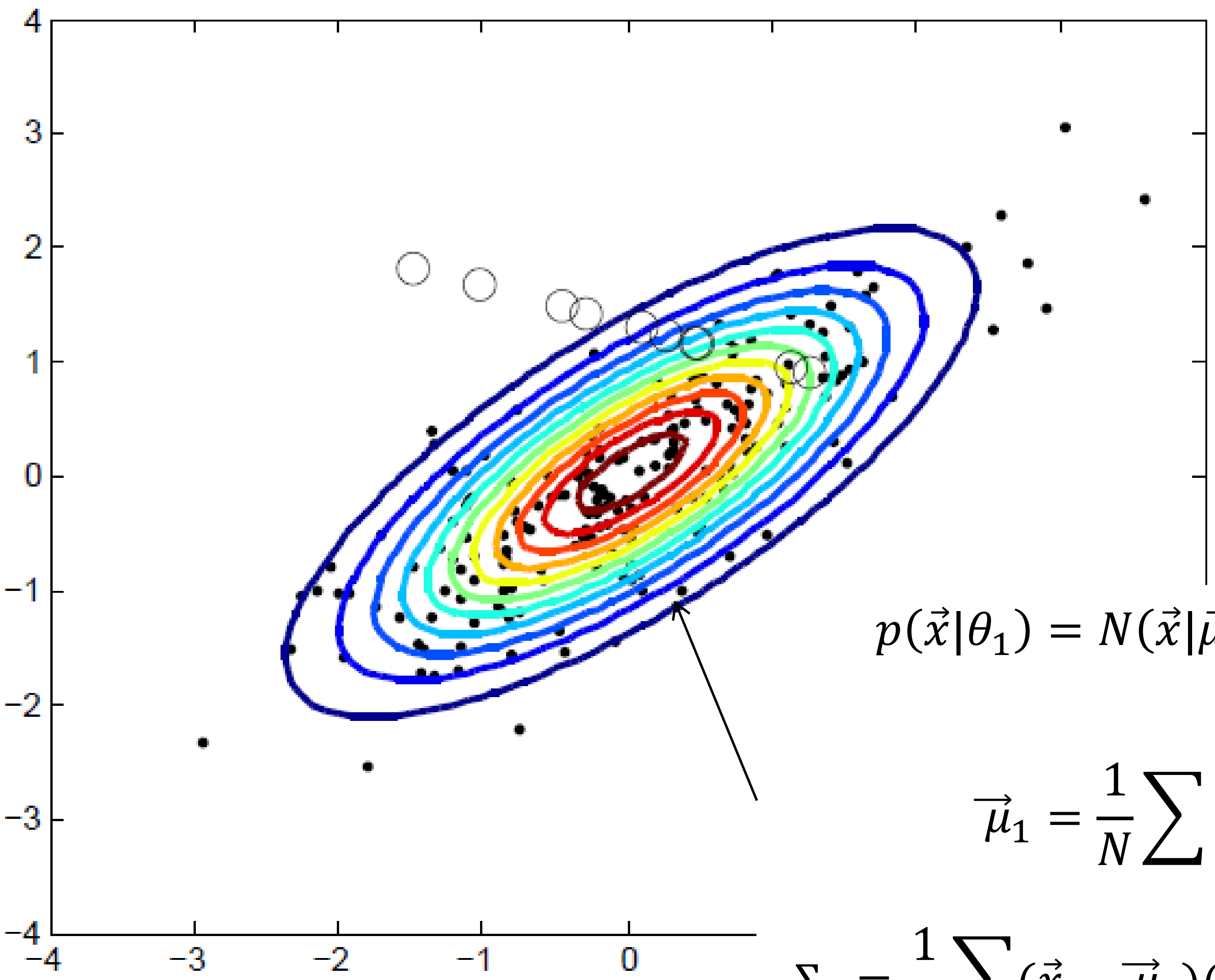


$p(\vec{x}|\theta_0) = N(\vec{x}|\vec{\mu}_0, \Sigma_0)$

Expected values of y

$p(y\,|\,x, \theta_0)$

EM algorithm – Maximisation Step



$$p(\vec{x}|\theta_1) = N(\vec{x}|\vec{\mu}_1, \Sigma_1)$$

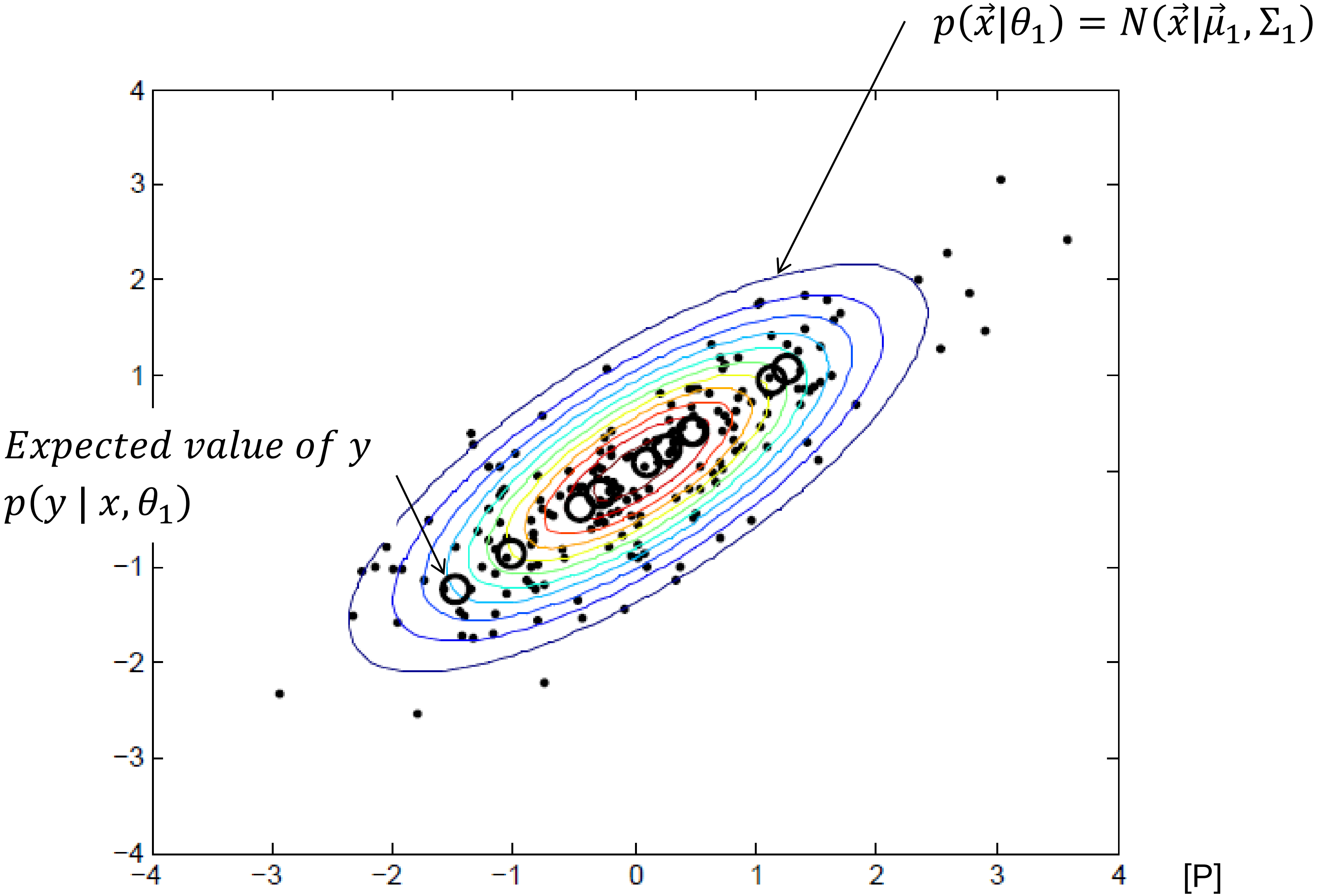
$$\vec{\mu}_1 = \frac{1}{N}\sum \vec{x}_i$$

[P]

$$\Sigma_1 = \frac{1}{N}\sum(\vec{x}_i - \vec{\mu}_1)(\vec{x_i} - \vec{\mu}_1)^T$$

EM algorithm – Expectation Step

University of Osnabrück, Institute of Cognitive Science

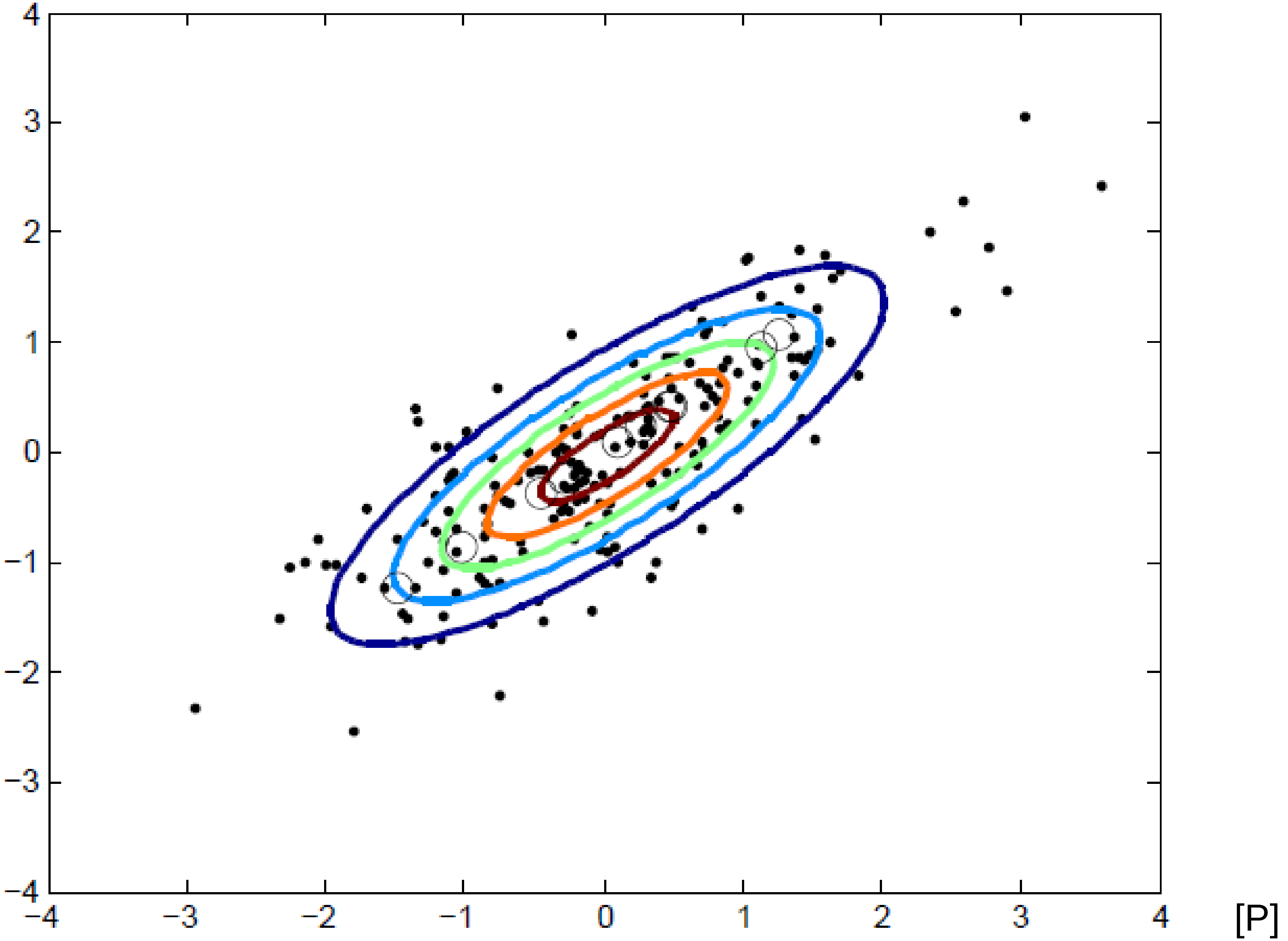


$p(\vec{x}|\theta_1) = N(\vec{x}|\vec{\mu}_1, \Sigma_1)$

$Expected\ value\ of\ y$

$p(y \mid x, \theta_1)$

EM algorithm – Maximisation Step



Remark on missing value substitution

- Missing information can not be regained.
- We can only „invent“ missing values such that they do not contradict the existing ones. For this we use models.
- At best, we do not destroy information by this procedure (but usually we do).
- Then why did we care about missing value substitution?

Answer: To make the existing values technically usable!

Summary of data preprocessing

- Data format is major issue for applications.
- Outlier detection requires some definition of what is regular.
- Missing values can be a major problem when spread among large data records, rendering most records unusable.
- Missing value substitution by mean of other data yields artifacts.
- Fitting a model to the data by regression is better, but does not make use of partial data.
- EM-algorithm solves this problem but suffers from problems of local search.

Similarity measures

Similarity measures

What we really want:

Relations between data on the semantic level, in particular, similarity / dissimilarity.

What is accessible to a machine:

Numerical measures, in particular, *distance functions* (*metrics*).

Distance function

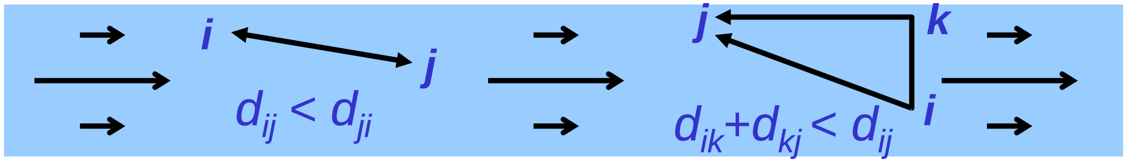
A *distance function* or *metric* d must obey the following conditions, which are reasonable for geometric distances (i to j are locations):

1. Symmetry: $d_{ij} = d_{ji}$ (from i to j is as far as from j to i).
2. Coincidence axiom: $d_{ij} = 0 \Leftrightarrow i = j$ (identity of indiscernibles).
3. Triangle equation: $d_{ik} + d_{kj} \geq d_{ij}$ (way over k is no shorter than direct path from i to j).

Note the axioms imply $d_{ij} \geq 0$ (non-negativity).

Question: Think of a “distance related” quantity that is not metric!

d_{ij} is fuel consumption of a vessel going from i to j in a river.



[H]

Distance function

Remark:

In mathematics, the term *distance function* is used only when the axioms are fulfilled.

In ML, *distance function* is often used like *dissimilarity function* and may be applied to quantities that do not match the axioms.

To make it crystal clear you mean a distance function fulfilling the axioms, use the term *metric*.

Distance matrix

For a data set $\{\vec{x}_1 \dots \vec{x}_n\}$, all information about distances is assembled in the distance matrix

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ \vdots & & & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{bmatrix}$$

where d_{ij} denotes the distance between $\vec{x}_i$ and $\vec{x}_j$.

Note $d_{\mathrm{ii}} = 0$ and $d_{\mathrm{ij}} = d_{\mathrm{ji}}$.

Similarity matrix

Distance calculation is motivated from geometric distances.

But in ML, distances are more broadly used to express similarity.

Similarities of data may be represented by a matrix as wells as distances.

When similarities are not computed from features of the data, but assigned explicitly from other sources (e.g., human insight), similarities may become particularly “non-geometric”.

Example: *x* likes *y* on a scale 1…10 :

No attributes specified for the persons, only distances! →

likes	Luke	Leia	Han
Luke	8	9	6
Leia	7	8	10
Han	6	9	16

Euclidean distance

Some common distances for $\vec{x}, \vec{y} \in \Re^L$:

Euclidean distance:

$$d(\vec{x}, \vec{y}) = ||\vec{x} - \vec{y}|| = \left(\sum_{i=1\ldots L} (x_i - y_i)^2\right)^{½}$$

- Simple and frequently used measure.
- No individual weighting of components.

Example: Broomstick production

Produced broomstick of dimensions x = (length, thickness)^T is compared to prototype values y = (160cm, 3cm)^T.

Problem: $||(161\text{cm}, 3\text{cm})^T - y|| = ||(160\text{cm}, 4\text{cm})^T - y||$.

Normalized euclidean distance

Idea: Weight dimensions according to variation from the mean.

Normalized euclidean distance (also: Pearson or χ^2-distance):

$$d(\vec{x}, \vec{y}) = \left(\sum_{i=1\ldots L} (x_i - y_i)^2 / \sigma_i\right)^{½}$$

with standard deviations σ_i.

Question: How does the broomstick example motivate the Pearson distance ?

- At first glance, the example suggests weighting according to mean, not deviation.
- But with smaller absolute value, the precision of production usually increases.

Pearson distance

Problem: Correlated vector components

Example:

$\vec{x}(t) = (x_1(t), x_2(t), \ldots\ x_L(t))^T$ with

$x_1(t) = x_2(t) = \ldots = x_{L-1}(t),$ $\sigma_1 = \ldots = \sigma_L = 1.$

Pearson distance:

$$d(\vec{x}(t_1), \vec{x}(t_2)) = \big((L-1) \cdot (x_1(t_1) - x_1(t_2))^2 + 1 \cdot (x_L(t_1) - x_L(t_2))^2\big)^{½}$$

→ The correlated components are over-weighted.

Mahalanobis distance

Idea: Scaling of distances using the covariance matrix C.

$$d(\vec{x}, \vec{y}) = ((\vec{x} - \vec{y})^T C^{-1} (\vec{x} - \vec{y}))^{½}$$

Properties:

- Scale and translation invariant.
- If C is unit matrix: Euclidean distance.
- Points of equal Mahalanobis distance to a center form an ellipsoid.
- Scaling might destroy structure within the data.

Interpretation: Diagonalize C

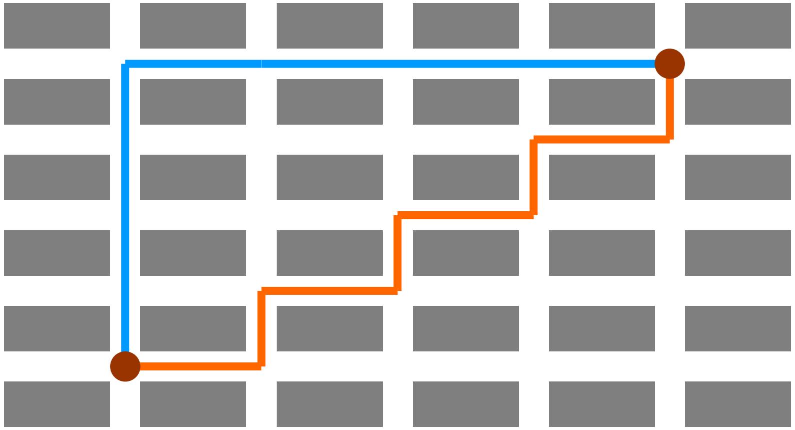
$$C = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_L \end{bmatrix}$$

with eigenvalues λ_i → scaling factors are $(\lambda_i)^{-½}$!

City block distance

Also: *Manhattan distance*

$$d(\vec{x}, \vec{y}) = \sum_{i=1 \ldots L} |x_i - y_i|$$



[H]

Remark:

The *Hamming distance* (= number of positions where two strings of equal length differ) is equal to the Manhattan distance for binary strings.

Chebyshev distance

Also: *Maximum distance, chessboard distance*

$$d(\vec{x}, \vec{y}) = \max_{i=1\ldots L} |x_i - y_i|$$

Minimum number of moves a king needs between two positions on a chessboard.

p-norm

Generalization:

$$d(\vec{x}, \vec{y}) = \left(\sum_{i=1 \ldots L} |x_i - y_i|^p\right)^{1/p}$$

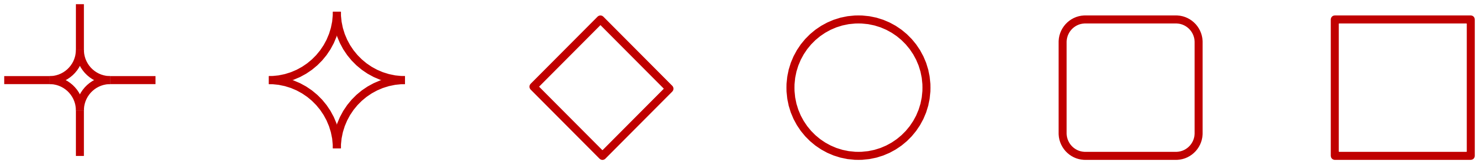
Special cases:

$p = 1$: $d(\vec{x}, \vec{y}) = \sum_{i=1 \ldots L} |x_i - y_i|$ (city block)

$p = 2$: $d(\vec{x}, \vec{y}) = (\sum_{i=1 \ldots L} (x_i - y_i)^2)^{½}$ (euclidean)

$p \to \infty$: $d(\vec{x}, \vec{y}) = \max_{i=1 \ldots L} |x_i - y_i|$ (maximum)

Unit circles (schematic):



[H]

So far, all distance measures relied on the topology of an $\Re^n$.

There are data with other topologies, e.g., angular attributes (topology of a circle):

$$||10^\circ - 30^\circ|| = 20^\circ$$

$$||0^\circ - 359^\circ|| = 359^\circ$$

Solution: *Embedding* complex topologies into an $\Re^n$.

Nominal scales

Mapping of nominal attribute values to real values:

(*low, medium, high*) → (1, 2, 3) makes sense, but

(*stone, wood, metal*) → (1, 2, 3) implies an order that is not there.

Solution: (*stone, wood, metal*) → $\left((1,0,0)^T,\ (0,1,0)^T,\ (0,0,1)^T\right)$

Problem:

For a large number *n* of attribute values, dimensionality becomes too high.

Solution: Choose normalized random vectors $v_i \in \Re^m$, $i = 1\ldots n$, $1 \ll m \ll n$ instead. Vectors drawn at random from a space of high dimension tend to be close to orthogonal (why?).

Binary scales

For binary attributes (e.g., *small / big*, *yes / no*) use the Jaccard index J as a similarity measure which is defined for sets A and B:

$$J(A, B) = (\#\text{ common elements}) / (\#\text{ all elements})$$

$$= |A \cap B| / |A \cup B|$$

Example:

$$J(\{a,c,d,e\}, \{b,c,e,f\}) = 2 / 6$$

Jaccard distance function:

$$J_d(A, B) = 1 - J(A, B)$$

Application in text mining:

Similarity of strings can be calculated by cutting strings into tokens and using J on the token sets.

Image sources

[M] Online material available at www.cs.cmu.edu/~tom/mlbook.html for the textbook: Tom M. Mitchell: *Machine Learning*, McGraw-Hill

[H] Gunther Heidemann, 2012.

[P] Michael Pardowitz, 2014.