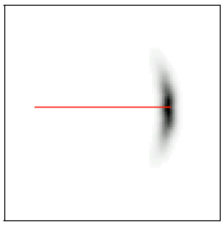
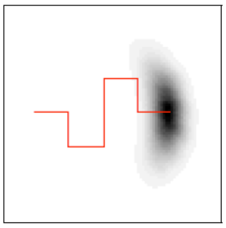
# Mobile Robot Localization

**Given a map of the environment, how can a robot determine its pose (planar coordinates + orientation)?**

**Two sources of uncertainty:**

**- observations depend probabilistically on robot pose**
**- pose changes depend probabilistically on robot actions**





**Example:**
**Uncertainty of robot position after travelling along red path ( shaded area indicates probability distribution)**

Slides on Robot Localization are partly adapted from

Sebastian Thrun, http://www-2.cs.cmu.edu/~thrun/papers/thrun.probrob.html
Michael Beetz, http://wwwradig.in.tum.de/vorlesungen/as.SS03/folien_links.html

1

# Formalization of Localization Problem

| | |
|---|---|
| m | model of environment (e.g. map) |
| $s_t$ | pose at time t |
| $o_t$ | observation at time t |
| $a_t$ | action at time t |
| $d_{0...t}$ | $= o_0, a_0, o_1, a_1, \ldots, o_t, a_t$<br>observation and action data up to t |

**Task:** Estimate $p(s_t \mid d_{0...t}, m) = b_t(s_t)$ "robot´s belief state at time t"

**Markov properties:**

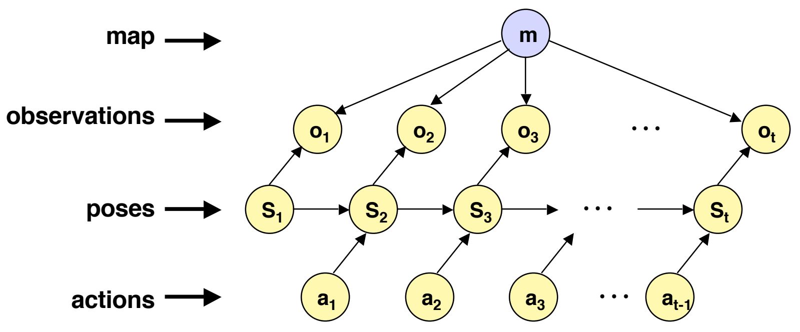
- Current observation depends only on current pose
- Next pose depends only on current pose and current action

"Future is independent of past given current state"

Markov assumption implies static environment!
(Violation, for example, by robot actions changing the environment)

2

## Structure of Probabilistic Localization




3

## Recursive Markov Localization

$$
\begin{aligned}
b_t(s_t) &= p(s_t \mid o_0, a_0, o_1, a_1, \ldots, a_{t-1}, o_t, m) \\
&\overset{\text{Bayes}}{=} \alpha_t\, p(o_t \mid o_0, a_0, \ldots, a_{t-1}, s_t, m)\ \ p(s_t \mid o_0, a_0, \ldots, a_{t-1}, m) \\
&\overset{\text{Markov}}{=} \alpha_t\, p(o_t \mid s_t, m)\ \ p(s_t \mid o_0, a_0, \ldots, a_{t-1}, m) \\
&\overset{\text{Total Prob.}}{=} \alpha_t\, p(o_t \mid s_t, m) \int p(s_t \mid o_0, a_0, \ldots, a_{t-1}, s_{t-1}, m)\ \ p(s_{t-1} \mid o_0, a_0, \ldots, a_{t-1}, m)\, ds_{t-1} \\
&\overset{\text{Markov}}{=} \alpha_t\, p(o_t \mid s_t, m) \int p(s_t \mid a_{t-1}, s_{t-1}, m)\ \ p(s_{t-1} \mid o_0, a_0, \ldots, a_{t-1}, m)\, ds_{t-1} \\
&= \alpha_t\, p(o_t \mid s_t, m) \int p(s_t \mid a_{t-1}, s_{t-1}, m)\ \ b_{t-1}(s_{t-1})\, ds_{t-1}
\end{aligned}
$$

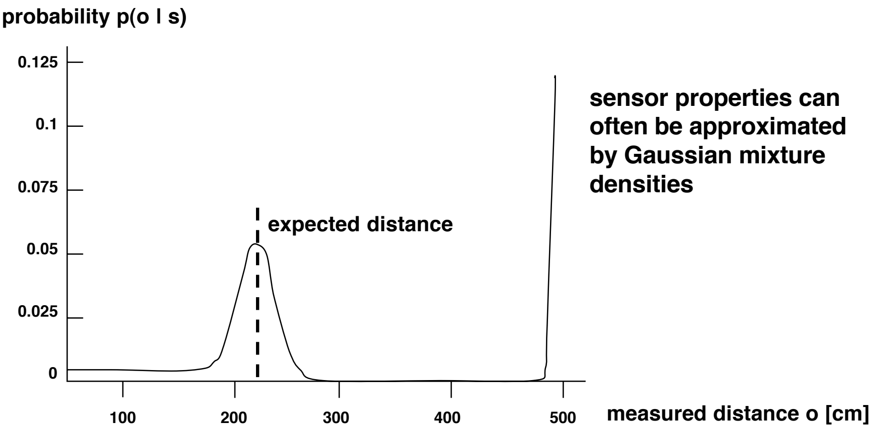
$\alpha_t$ is normalizing factor

$$
\boxed{b_t(s_t) = \alpha_t\, p(o_t \mid s_t, m) \int p(s_t \mid a_{t-1}, s_{t-1}, m)\ \ b_{t-1}(s_{t-1})\, ds_{t-1}}
$$

| | |
|---|---|
| $p(o_t \mid s_t, m)$ | probabilistic perceptual model - often time-invariant: $p(o \mid s, m)$ |
| $p(s_t \mid a_{t-1}, s_{t-1}, m)$ | probabilistic motion model - often time-invariant: $p(s' \mid a, s, m)$ |

4

# Probabilistic Sensor Model for Laser Range Finder

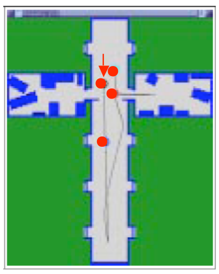



sensor properties can often be approximated by Gaussian mixture densities

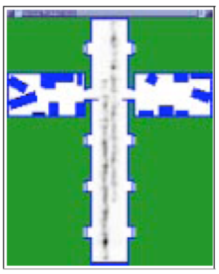
Adapted from: Sebastian Thrun, Probabilistic Algorithms in Robotics
http://www-2.cs.cmu.edu/~thrun/papers/thrun.probrob.html



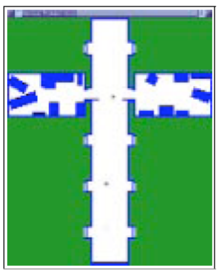
# Grid-based Markov Localization (Example 1)



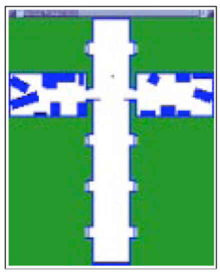
robot path with 4 reference poses, initially belief is equally distributed



distribution of belief at second pose



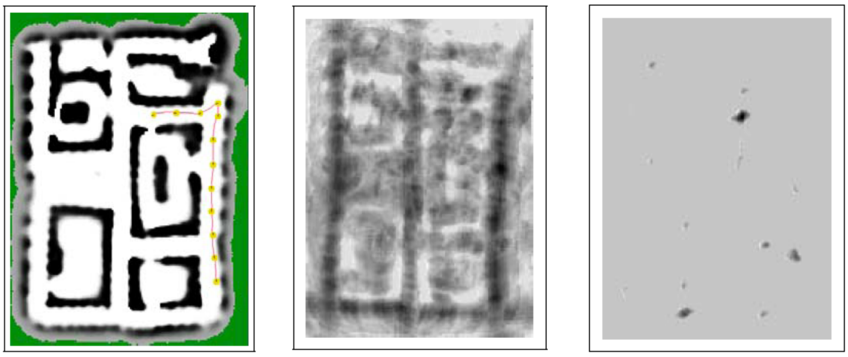
distribution of belief at third pose



distribution of belief at fourth pose

**Ambiguous localizations due to a repetitive and symmetric environment are sharpened and disambiguated after several observations.**

# Grid-based Markov Localization (Example 2)



map and robot path

maximum position probabilities after 6 steps

maximum position probabilities after 12 steps

[Burgard et al. 96]

7

# Approximating Probabilistic Update by Monte Carlo Localization (MCL)

"Importance Sampling"
"Particle Filters"
"Condensation Algorithm"

$\}$ different names for a method to approximate a probability density by discrete samples (see slide "Sampling Methods")

Approximate implementation of belief update equation

$$b_t(s_t) \;=\; \alpha_t\, p(o_t \mid s_t, m) \int p(s_t \mid a_{t-1}, s_{t-1}, m)\; b_{t-1}(s_{t-1})\, ds_{t-1}$$

1. Draw a sample $s_{t-1}$ from the current belief $b_{t-1}(s_{t-1})$ with a likelihood given by the importance factors of the belief $b_{t-1}(s_{t-1})$.
2. For this $s_{t-1}$ guess a successor pose $s_t$ according to the distribution $p(s_t \mid a_{t-1}, s_{t-1}, m)$.
3. Assign a preliminary importance factor $p(o_t \mid s_t, m)$ to this sample and add it to the new sample ser representing $b_t(s_t)$.
4. Repeat Step 1 through 3 m times. Finally, normalize the importance factors in the new sample set $b_t(s_t)$ so that they add up to 1.

MCL is very effective and can give good results with as few as 100 samples.

8

# Simultaneous Localization and Mapping (SLAM)

**Typical problem for a mobile robot in an unknown environment:**

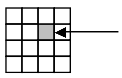
- **learn the environment ("mapping")**
- **keep track of position and orientation ("localization")**

**"Chicken-and-egg" problem:**

- **robot needs knowledge of environment in order to interpret sensor readings for localization**
- **robot needs pose knowledge in order to interpret sensor readings for mapping**

**Make the environment a multidimensional probabilistic variable!**

**Example: Model of environment is a probabilistic occupancy grid**



$$P_{ij} = \begin{cases} e_{ij} & (x_i, y_j) \text{ is empty} \\ 1-e_{ij} & (x_i, y_j) \text{ is occupied} \end{cases}$$

9

# Bayes Filter for SLAM

**Extend the localization approach to simultaneous mapping:**

$$b_t(s_t) = \alpha_t \, p(o_t \mid s_t, m) \int p(s_t \mid a_{t-1}, s_{t-1}, m) \; b_{t-1}(s_{t-1}) \, ds_{t-1}$$

$$b_t(s_t, m_t) = \alpha_t \, p(o_t \mid s_t, m_t) \iint p(s_t, m_t \mid a_{t-1}, s_{t-1}, m_{t-1}) \; b_{t-1}(s_{t-1}, m_{t-1}) \; ds_{t-1} \, dm_{t-1}$$

**Assuming a time-invariant map:**

$$\boxed{b_t(s_t, m) = \alpha_t \, p(o_t \mid s_t, m) \int p(s_t, m_t \mid a_{t-1}, s_{t-1}, m) \; b_{t-1}(s_{t-1}, m) \, ds_{t-1}}$$

**$b_t(s_t, m)$ is (N+3)-dimensional with N variables for m (N >> 1000) and 3 for $s_t$**
**=> complexity problem**

**Important approaches to cope with this complexity:**

- **Kalman filtering (Gaussian probabilities and linear updating)**
- **estimating only the mode of the posterior, $\text{argmax}_m$ b(m)**
- **treating the robot path as "missing variables" in Expectation Maximization**

10

# Kalman Filter for SLAM Problems (1)

**Basic Kalman Filter assumptions:**

1. Next-state function is linear with added Gaussian noise
2. Perceptual model is linear with added Gaussian noise
3. Initial uncertainty is Gaussian

Ad 1) Next state in SLAM is pose $s_t$ and model m.

- m is assumed constant
- $s_t$ is non-linear in general, approximately linear in a first-degree Taylor series expansion ("Extended Kalman Filtering")

Let $x_t$ be the state variable ($s_t$, m) and $\varepsilon_{control}$ Gaussian noise with covariance $\Sigma_{control}$, then

$$p(x_t \mid a_{t-1}, x_{t-1}) = A\, x_{t-1} + B\, a_{t-1} + \varepsilon_{control}$$

Ad 2) Sensor measurements are usually nonlinear in robotics, with non-white Gaussian noise. Approximation by first-degree Taylor series and $\varepsilon_{measure}$ Gaussian noise with covariance $\Sigma_{measure}$.

$$p(o_t \mid x_t) = C\, x_t + \varepsilon_{measure}$$

11

# Kalman Filter for SLAM Problems (2)

**Bayes Filter equation**

$$b_t(s_t, m) = \alpha_t\, p(o_t \mid s_t, m) \int p(s_t, m_t \mid a_{t-1}, s_{t-1}, m)\; b_{t-1}(s_{t-1}, m)\, ds_{t-1}$$

can be rewritten using the standard Kalman Filter equations:

$$\begin{aligned}
\mu'_{t-1} &= \mu_{t-1} + B\, a_t \\
\Sigma'_{t-1} &= \Sigma_{t-1} + \Sigma_{control} \\
K_t &= \Sigma'_{t-1}\, C^T\, (C\Sigma'_{t-1}C^T + \Sigma_{measure})^{-1} \\
\mu_t &= \mu'_{t-1} + K_t\, (o_{t-1} - C\mu'_{t-1}) \\
\Sigma_t &= (I - K_t C)\, \Sigma'_{t-1}
\end{aligned}$$

Compare with slides on Kalman Filtering in "Bildverarbeitung".

- Kalman Filtering estimates the full posterior distribution for all poses (not only the maximum)
- Guaranteed convergence to true map and robot pose
- Gaussian sensor noise is a bad model for correspondence problems

12

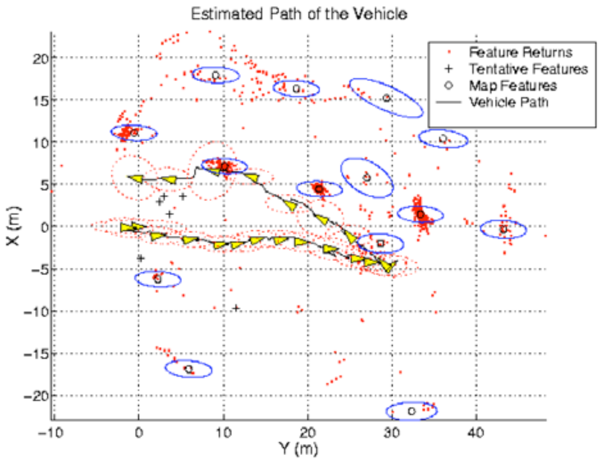
# Example: Underwater Sonic Mapping

From: S.Williams, G. Dissanayake, and H.F. Durrant-Whyte. Towards terrain-aided navigation for underwater robotics. *Advanced Robotics*, 15(5), 2001.

**Kalman Filter map and pose estimation**

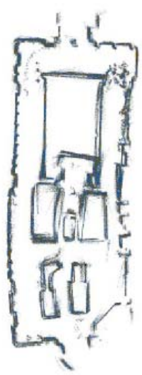
**Figure shows:**

- **estimated path of underwater vehicle with ellipses indicating position uncertainty**
- **14 landmarks obtained by sonar measurements with ellipses indicating uncertainty, 5 artificial landmarks, the rest other reflective objects**
- **additional dots for weak landmark hypotheses**




13

# Solving the Correspondence Problem



**Map obtained from raw sensory data of a cyclic environment (large hall of a museum) based on robot´s odometry**

**correspondence problem!**

**Map obtained by EM algorithm: Iterative maximization of both robot path and model**

**non-incremental procedure!**

14

# Mapping with Expectation Maximization

**Principle of EM mapping algorithm:**

> Repeat until no more changes
> E-step: Estimate robot poses for given map
> M-step: Calculate most likely map given poses

The algorithm computes the maximum of the expectation of the joint log likelihood of the data $d^t = \{a_0, o_0, \ldots, a_t, o_t\}$ and the robot´s path $s^t = \{s_0, \ldots, s_t\}$.

$$m^{[i+1]} = \underset{m}{\operatorname{argmax}} E_{s^t}\left[\log p\left(d^t, s^t \mid m\right) \mid m^{[i]}, d^t\right]$$

$$\Downarrow$$

$$m^{[i+1]} = \underset{m}{\operatorname{argmax}} \sum_{\tau} \int p\left(s_\tau \mid m^{[i]}, d^t\right) \log p(o_\tau \mid s_\tau, m)\, ds_\tau$$

E-step: Compute the posterior of pose $s_\tau$ based on $m^{[i]}$ and all data including $t > \tau$ : => different from incremental localization

M-step: Maximize $\log p(o_\tau \mid s_\tau, m)$ for all $\tau$ and all poses $s^t$ under the expectation calculated in the E-step

15

# Mapping with Incremental Maximum-Likelihood Estimation

Stepwise maximum-likelihood estimation of map and pose is inferior to Kalman Filtering and EM estimation, but less complex.

Obtain series of maximum-likelihood maps and poses

$m_1^*, m_2^*, \ldots$

$s_1^*, s_2^*, \ldots$

by maximizing the marginal likelihood:

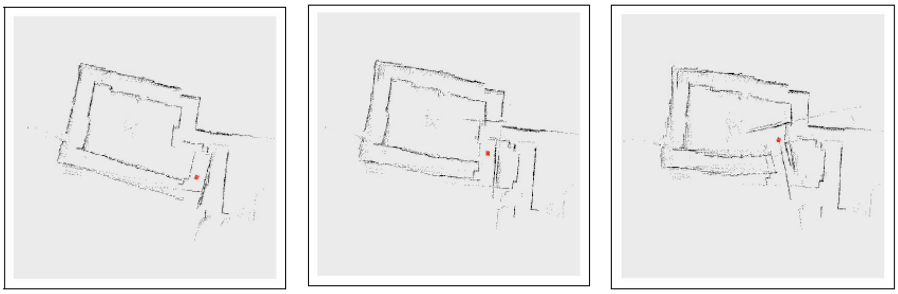
$$\langle m_t^*, s_t^* \rangle = \underset{m_t, s_t}{\operatorname{argmax}}\; p(o_t \mid m_t, s_t)\, p(m_t, s_t \mid a_t, s_{t-1}^*, m_{t-1}^*)$$

This equation follows from the Bayes Filter equation by assuming that map and pose at t-1 are known for certain.

- real-time computation possible
- unable to handle cycles

16

## Example of Incremental Maximum-Likelihood Mapping



**At every time step, the map is grown by finding the most likely continuation. Map estimates do not converge as robot completes cycle because of accumulated pose estimation error.**

**Examples from: Sebastian Thrun, Probabilistic Algorithms in Robotics**
**http://www-2.cs.cmu.edu/~thrun/papers/thrun.probrob.html**

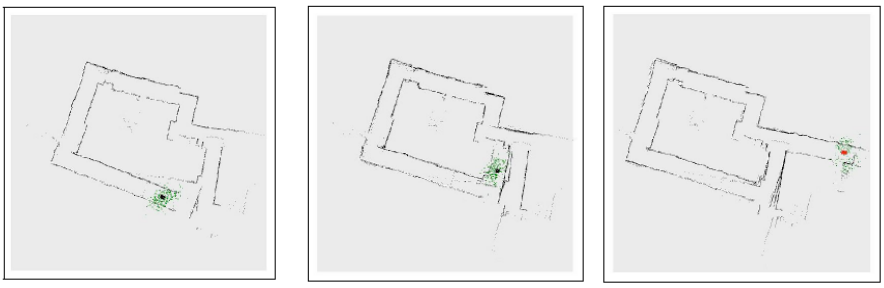
17

## Maintaining a Pose Posterior Distribution

**Problems with cyclic environment can be overcome by maintaining not only the maximum-likelihood pose estimate at t-1 but also the uncertainty distribution using Bayes Filter:**

$$p(s_t \mid o^t, a^t) = \alpha\, p(o_t \mid s_t) \int p(s_t \mid a_{t-1}, s_{t-1})\; p(s_{t-1} \mid o^{t-1}, a^{t-1})\, ds_{t-1}$$

**Last example repeated, representing the pose posterior by particles. Uncertainty is transferred onto map, so major corrections remain possible.**



18

## Estimating Probabilities from a Database

Given a sufficiently large database with tupels $\underline{a}^{(1)} \dots \underline{a}^{(N)}$ of an unknown distribution $P(\underline{X})$, we can compute maximum likelihood estimates of all partial joint probabilities and hence of all conditional probabilities.

$X_{m_1}, \dots, X_{m_K}$ = subset of $X_1, \dots X_L$ with $K \leq L$

$w_{\underline{a}}$ = number of tuples in database with $X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}$

$N$ = total number of tuples

**Maximum likelihood estimate of $P(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K})$ is**

$$P'(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}) = w_{\underline{a}} / N$$

If a priori information is available, it may be introduced via a bias $m_{\underline{a}}$ :

$$P'(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}) = (w_{\underline{a}} + m_{\underline{a}}) / N$$

Often $m_{\underline{a}} = 1$ is chosen for all tupels $\underline{a}$ to express equal likelihoods in the case of an empty database.

19

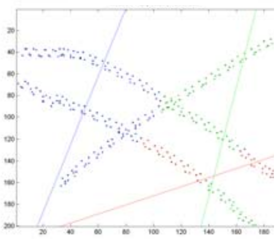
## Idea of Expectation Maximization

Consider the problem of fitting 3 straight lines to data, not knowing which data belong to which line.
(Example by Anna Ergorova, FU Berlin)

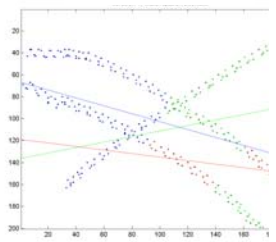
**Algorithm:**

A Select 3 random lines initially
B Assign data points to each line by minimum distance criterion
C Determine best-fitting straight line for assigned data points
D Repeat B and C until no further changes occur

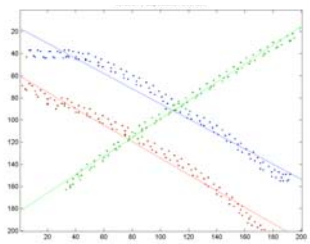
1. Iteration (after B)



1. Iteration (after C)



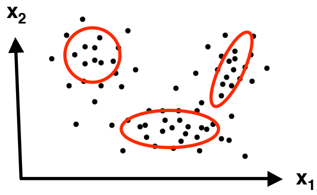
6. Iteration (after B)



20

# Learning Mixtures of Gaussians

Determine Gaussian mixture distribution with K multivariate Gaussians which best describes given data => unsupervised clustering




Multivariate Gaussian mixture distribution:

$$p(\underline{x}) = \sum_{i=1 \,..\, K} w_i \, N(\underline{\mu}_i, \Sigma_i)$$

$$\text{with} \sum_{i=1 \,..\, K} w_i = 1$$

A Select $w_i$, $\underline{\mu}_i$ and $\Sigma_i$, i = 1 .. K, at random (K is given)

B For each datum $\underline{x}_j$ compute probability $p_{ij}$ that $\underline{x}_j$ was generated by $N(\underline{\mu}_i, \Sigma_i)$:

$$p_{ij} = w_i \, N(\underline{\mu}_i, \Sigma_i)$$

C Compute new weights $w_i'$, mean $\underline{\mu}_i'$, and covariance $\Sigma_i'$ by maximum likelihood estimation:

$$w_i' = \sum_j p_{ij} \qquad \underline{\mu}_i' = \sum_j p_{ij} \, \underline{x}_j / w_i' \qquad \underline{\Sigma}_i' = \sum_j p_{ij} \, \underline{x}_j \, \underline{x}_j^T / w_i'$$

D Repeat B and C until no further changes occur

21

# General Form of EM Algorithm

Compute unknown distribution for data with hidden variables

$\underline{x}$ observed values of all samples
$\underline{Y}$ variables with hidden values for all samples
$\underline{\Theta}$ parameters for probabilistic model

EM algorithm: $\underline{\theta}' = \underset{\underline{\theta}}{\text{argmax}} \sum_{\underline{y}} p(\underline{Y} = \underline{y} \mid \underline{x}, \underline{\theta}) \, L(\underline{x}, \underline{Y} = \underline{y} \mid \underline{\theta})$

E-step: Computation of summation
=> Likelihood of "completed" data w.r.t. distribution $p(\underline{Y} = \underline{y} \mid \underline{x}, \underline{\theta})$

M-step: Maximization of expected likelihood w.r.t. parameters $\underline{\theta}$

- The computed parameters increase the likelihood of data and hidden values with each iteration
- The algorithm terminates in a local maximum

22

# Expectation Maximization for Estimating Bayes Net with Hidden Variable

<u>Expectation step of EM</u>:

Use current (initial) probability estimates to compute probability $P(\underline{a})$ for all attribute combinations $\underline{a}$ (including values for hidden variables).

$\underline{a}_1 = [X_1{=}a_1 \,,\, X_2{=}a_{m2},\, X_3{=}a_{m3},\, \ldots\,] \quad w_{\underline{a}^*} \cdot P(\underline{a}_1)$

$\underline{a}_2 = [X_1{=}a_2 \,,\, X_2{=}a_{m2},\, X_3{=}a_{m3},\, \ldots\,] \quad w_{\underline{a}^*} \cdot P(\underline{a}_2)$

$\underline{a}^* = [\, * \,,\, X_2{=}a_{m2},\, X_3{=}a_{m3},\, \ldots\,] \qquad w_{\underline{a}^*}$

$\vdots$

$\underline{a}_M = [X_1{=}a_M \,,\, X_2{=}a_{m2},\, X_3{=}a_{m3},\, \ldots\,] \quad w_{\underline{a}^*} \cdot P(\underline{a}_M)$

missing value    absolute frequency    completed database

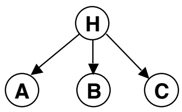
**Recommended reading: Borgelt & Kruse, Graphical Models, Wiley 2002**



# Example for Expectation Maximization (1)

(adapted from Borgelt & Kruse, Graphical Models, Wiley 2002)

Given 4 binary probabilistic variables A, B, C, H with known dependency structure:




Given also a database with tuples [ * A B C] where H is a missing attribute.

| H | A | B | C | w |
|---|---|---|---|---|
| * | T | T | T | 14 |
| * | T | T | F | 11 |
| * | T | F | T | 20 |
| * | T | F | F | 20 |
| * | F | T | T | 5 |
| * | F | T | F | 5 |
| * | F | F | T | 11 |
| * | F | F | F | 14 |

absolute frequencies of occurrence

Estimate of the conditional probabilities of the Bayes Net nodes !

## Example for Expectation Maximization (2)

Initial (random) probability assignments:

| H | P(H) |
|---|---|
| T | 0.3 |
| F | 0.7 |

| A | H | P(A\|H) |
|---|---|---|
| T | T | 0.4 |
| T | F | 0.6 |
| F | T | 0.6 |
| F | F | 0.4 |

| B | H | P(B\|H) |
|---|---|---|
| T | T | 0.7 |
| T | F | 0.8 |
| F | T | 0.3 |
| F | F | 0.2 |

| C | H | P(C\|H) |
|---|---|---|
| T | T | 0.8 |
| T | F | 0.5 |
| F | T | 0.2 |
| F | F | 0.5 |

With $$P(H \mid A,B,C) = \frac{P(A \mid H) \cdot P(B \mid H) \cdot P(C \mid H) \cdot P(H)}{\sum_H P(A \mid H) \cdot P(B \mid H) \cdot P(C \mid H) \cdot P(H)}$$

one can complete the database:

| H | A | B | C | w |
|---|---|---|---|---|
| T | T | T | T | 1.27 |
| T | T | T | F | 3.14 |
| T | T | F | T | 2.93 |
| T | T | F | F | 8.14 |
| T | F | T | T | 0.92 |
| T | F | T | F | 2.37 |
| T | F | F | T | 3.06 |
| T | F | F | F | 8.49 |

| H | A | B | C | w |
|---|---|---|---|---|
| F | T | T | T | 12.73 |
| F | T | T | F | 7.86 |
| F | T | F | T | 17.07 |
| F | T | F | F | 11.86 |
| F | F | T | T | 4.08 |
| F | F | T | F | 2.63 |
| F | F | F | T | 7.94 |
| F | F | F | F | 5.51 |



## Example for Expectation Maximization (3)

Based on the modified complete database, one computes the maximum likelihood estimates of the conditional probabilities of the Bayes Net.

Example: $$P(A = T \mid H = T) \approx \frac{1.27 \cdot 3.14 \cdot 2.93 \cdot 8.14}{1.27 \cdot 3.14 \cdot 2.93 \cdot 8.14 \cdot 0{,}92 \cdot 2.73 \cdot 3.06 \cdot 8.49} \approx 0.51$$

This way one gets new probability assignments:

| H | P(H) |
|---|---|
| T | 0.3 |
| F | 0.7 |

| A | H | P(A\|H) |
|---|---|---|
| T | T | 0.51 |
| T | F | 0.71 |
| F | T | 0.49 |
| F | F | 0.29 |

| B | H | P(B\|H) |
|---|---|---|
| T | T | 0.25 |
| T | F | 0.39 |
| F | T | 0.75 |
| F | F | 0.61 |

| C | H | P(C\|H) |
|---|---|---|
| T | T | 0.27 |
| T | F | 0.60 |
| F | T | 0.73 |
| F | F | 0.40 |

This completes the first iteration. After ca. 700 iterations the modifications of the probabilities are less than $10^{-4}$. The resulting values are

| H | P(H) |
|---|---|
| T | 0.5 |
| F | 0.5 |

| A | H | P(A\|H) |
|---|---|---|
| T | T | 0.5 |
| T | F | 0.8 |
| F | T | 0.5 |
| F | F | 0.2 |

| B | H | P(B\|H) |
|---|---|---|
| T | T | 0.2 |
| T | F | 0.5 |
| F | T | 0.8 |
| F | F | 0.2 |

| C | H | P(C\|H) |
|---|---|---|
| T | T | 0.4 |
| T | F | 0.6 |
| F | T | 0.6 |
| F | F | 0.4 |