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.

1

Idea of Expectation Maximization

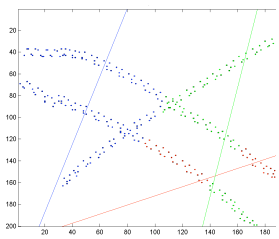
Consider the problem of fitting 3 straight lines to data, not knowing which data belong to which line.

(Example by Anna Ergorova, FU Berlin)

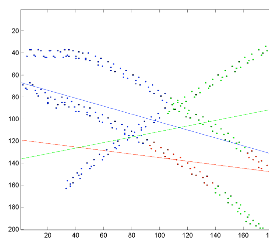
Algorithm:

- **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

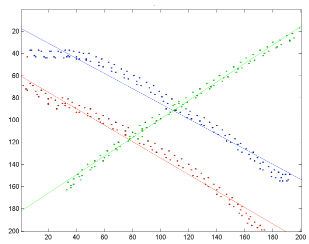
1. Iteration (after B)



1. Iteration (after C)



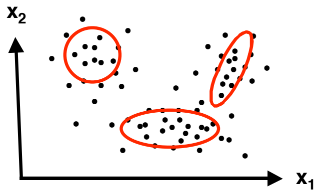
6. Iteration (after B)



2

Learning Mixtures of Gaussians

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



Multivariate Gaussian mixture distribution:

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

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

- **A** Select w_i, μ_i and Σ_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 $N(\mu_i, \Sigma_i)$:
  $$p_{ij} = w_i \, N(\mu_i, \Sigma_i)$$
- **C** Compute new weights w_i', mean μ_i', and covariance Σ_i' by maximum likelihood estimation:
  $$w_i' = \sum_j p_{ij} \qquad \mu_i' = \sum_j p_{ij} \, \underline{x}_j / w_i' \qquad \Sigma_i' = \sum_j p_{ij} \, \underline{x}_j \, \underline{x}_j^T / w_i'$$
- **D** Repeat B and C until no further changes occur

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

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

M-step: Maximization of expected likelihood w.r.t. parameters θ

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

Expectation Maximization for Estimating Bayes Net with Hidden Variable

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

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

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

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

$\underline{a}^* = [* , X_2{=}a_{m2}, X_3{=}a_{m3}, \dots]$ $\quad w_{\underline{a}^*}$ $\quad \vdots$

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

missing value   absolute frequency   completed database

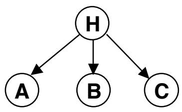
Recommended reading: Borgelt & Kruse, Graphical Models, Wiley 2002

5

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 !

6

Example for Expectation Maximization (2)

Initial (random) probability assignments:

H	P(H)	A	H	P(A|H)	B	H	P(B|H)	C	H	P(C|H)
T	0.3	T	T	0.4	T	T	0.7	T	T	0.8
F	0.7	T	F	0.6	T	F	0.8	T	F	0.5
		F	T	0.6	F	T	0.3	F	T	0.2
		F	F	0.4	F	F	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

7

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)	A	H	P(A|H)	B	H	P(B|H)	C	H	P(C|H)
T	0.3	T	T	0.51	T	T	0.25	T	T	0.27
F	0.7	T	F	0.71	T	F	0.39	T	F	0.60
		F	T	0.49	F	T	0.75	F	T	0.73
		F	F	0.29	F	F	0.61	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)	A	H	P(A|H)	B	H	P(B|H)	C	H	P(C|H)
T	0.5	T	T	0.5	T	T	0.2	T	T	0.4
F	0.5	T	F	0.8	T	F	0.5	T	F	0.6
		F	T	0.5	F	T	0.8	F	T	0.6
		F	F	0.2	F	F	0.2	F	F	0.4

8

Learning Models for High-level Image Interpretation

What parts of a scene constitute "meaningful occurrences" and should be recognized?

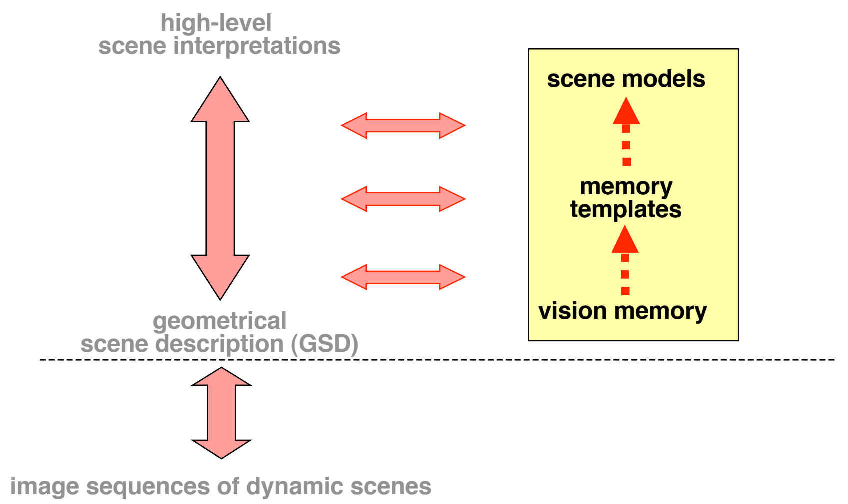
Basic engineering applications:
Fixed recognition tasks, determined by the application context.
=> handcrafted models

Advanced engineering applications:
Flexible recognition tasks, determined by user.
=> models result from supervised learning

Biological vision:
Recognition should support expectation generation and hence survival.
=> models result from unsupervised learning

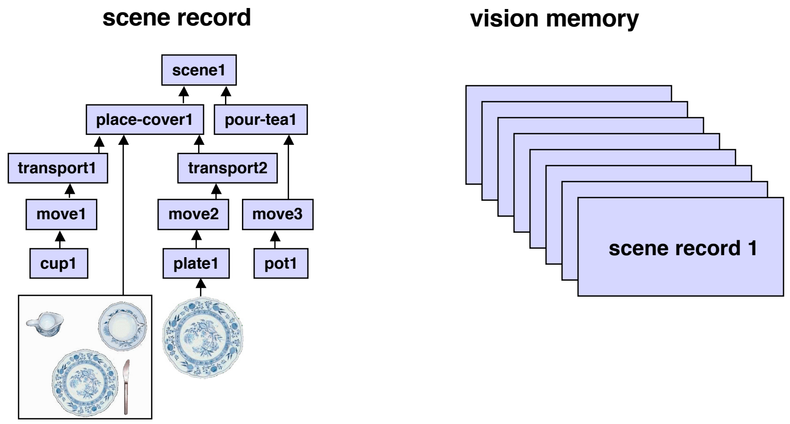
9

Learning in Support of High-level Scene Interpretation



10

Basic Structure of Vision Memory



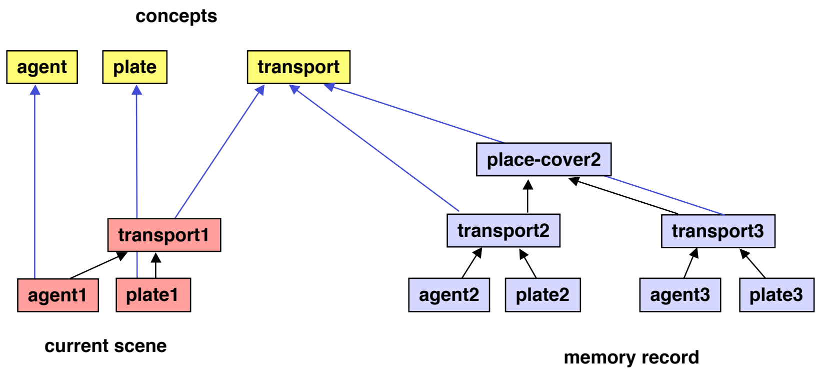
It is an open research question, how much imagery should (can) be preserved in a vision memory.

11

Case-based Expectation Generation from Memory Records

Memory records are "cases" which may provide missing information for an ongoing scene:

- identify memory records which partially match current scene
- adapt memory information to current scene
- provide expectations about current scene



12

Basic Learning Tasks

Michalski 86: Learning is the construction or modification of representations of experiences.

Unsupervised learning

determine reoccuring patterns in scene records

=> conceptual clustering

Supervised learning

determine description covering several examples

=> inductive generalization

Example of Supervised Learning

1. "This is how you lay a table"

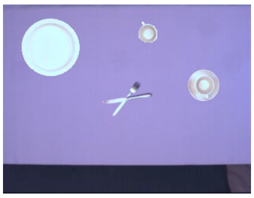
2. "This is how you lay a table"

•
•
•

42. "This is how you lay a table"

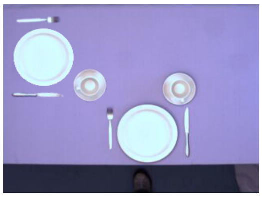
determine covering description

Example of Unsupervised Learning



Scene 1

Scene 2

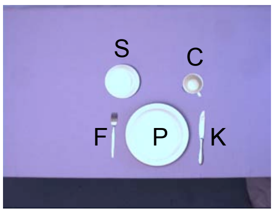


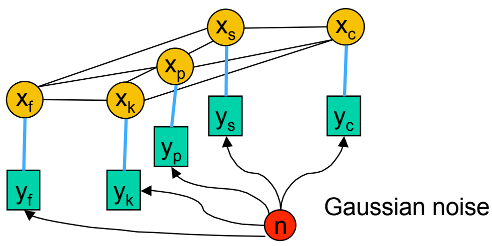
Scene 3

Find reoccurring aggregates => clustering

Event Space Modelling with Markov Networks

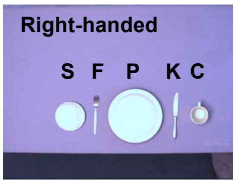
(Approach developed by Somboon Hongeng 2004)

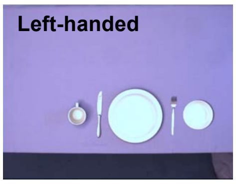




- **Node "i" is associated with hidden variable x_i and noisy observation y_i**
  - x_i is the feature vector of a primitive event "i"
- **Links of all nodes to reference object (plate)**
  - more links are added based on feature correlation
  - edges are associated with potential function $\square_{i,j}(x_i,x_j)$

Learning Event Clusters from Simulated 1D-Cover Scenes

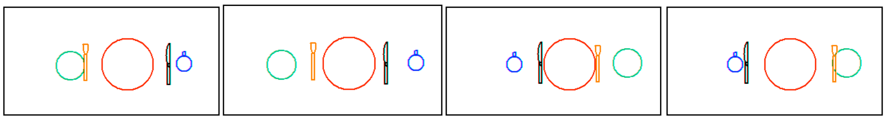




1D cover settings

200 simulated cover-laying scenes:

- **primitive events are described by starting time t and horizontal displacement u relative to plate**
- **placement orders P-K-F-C-S, P-K-F-S-C, P-F-K-S-C, P-F-K-C-S**



17

Clustering Results

- **Potential functions modeled with 200 Gaussian kernels**
- **Found 8 event classes from 200 simulated training sequences**

4 examples:

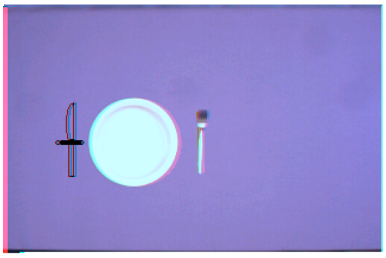
cf^1:	cf^2:	cf^3:	cf^4:
P(0,0)	P(0,0)	P(0,0)	P(0,0)
K(54,200)	F(52,200)	F(-55,200)	K(-51,200)
F(-55,400)	K(-52,400)	K(55,400)	F(51,400)
C(90,600)	C(-85,600)	S(-88,600)	C(-85,600)
S(-84,800)	S(85,800)	C(89,800)	S(85,800)

mean displacement of fork relativ to plate = 54
mean starting time of fork-laying relative to plate = 200

18

Inferencing with Learned Markov Models (1)

- Observe primitive laying events
- Generate expectations about unobserved events (past or future)

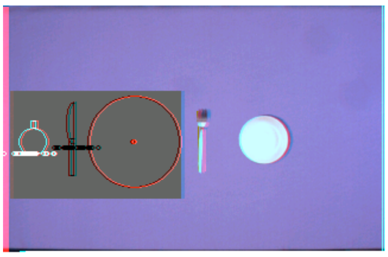


p.(u,v) = (110, 115), p.t = 0
f.(u,v) = (165, 115), f.t = 200
k.(u,v) = (-52.77,115), k.t = 398 prediction

19

Inferencing with Learned Markov Models (2)

- Observe primitive laying events
- Generate expectations about unobserved events (past or future)

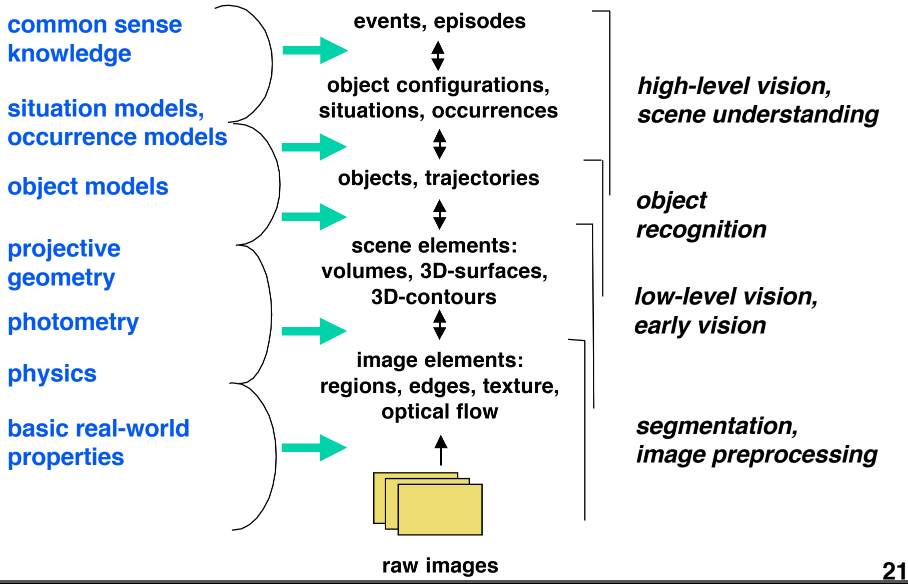


occlusion
f.(u,v) = (165, 115), f.t = 0
s.(u,v) = (220,115), s.t = 600
Expectation 1:
p.t = -400, k.t = -200, c.t = 392/220
Expectation 2:
p.t = -200, k.t = 197, c.t = 410/599

Note: Computing event probabilities in Markov Networks can in general only be done approximately by iterative procedures

20

Review of Image Understanding as a Knowledge-based Process



21