Hidden Markov Models project-2

Christian M. Adriano **(Chris)**
christian.adriano@hpi.de

Motivation

<u>Context:</u> Sequence data can have correlations (non-i.i.d.) which can be exploited to make predictions:

- Weather forecast
- Speech recognition
- Robot localization
- Activity recognition

<u>Problems:</u>

- How to estimate states given observation (prediction problem)?
- How to produce observations given states (data generation problem)?

<u>Solution is to assume that:</u>

- states are not always visible
- only observations are visible

Models for sequential data

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

	Independent classification	Correlated classification
Generative models	Mixture of Gaussians	**Hidden Markov Model**
Discriminative models	Logistic Regression	**Conditional Random Field**
	Feed Forward Neural Network	**Recurrent Neural Network**

source: CS480/680 Spring 2019 Pascal Poupart

Markov Property

The Future is independent of the Past if we know the Present

$$(S_{t+1} \perp S_{t-1}) | S_t$$

Or equivalently

$$P(S_{t+1} | S_t) = P(S_{t+1} | S_t, S_{t-1}, \dots, S_{t-n})$$

<u>What are the implications?</u>

- The present state needs to hold all the information necessary to predict the future
- Future inherently stochastic
- State space explosion

State

History of observations $h_t = \{O_1, O_2, \dots, O_t\}$

The <u>state</u> at given time is function of this history $S_t = F(h_t)$

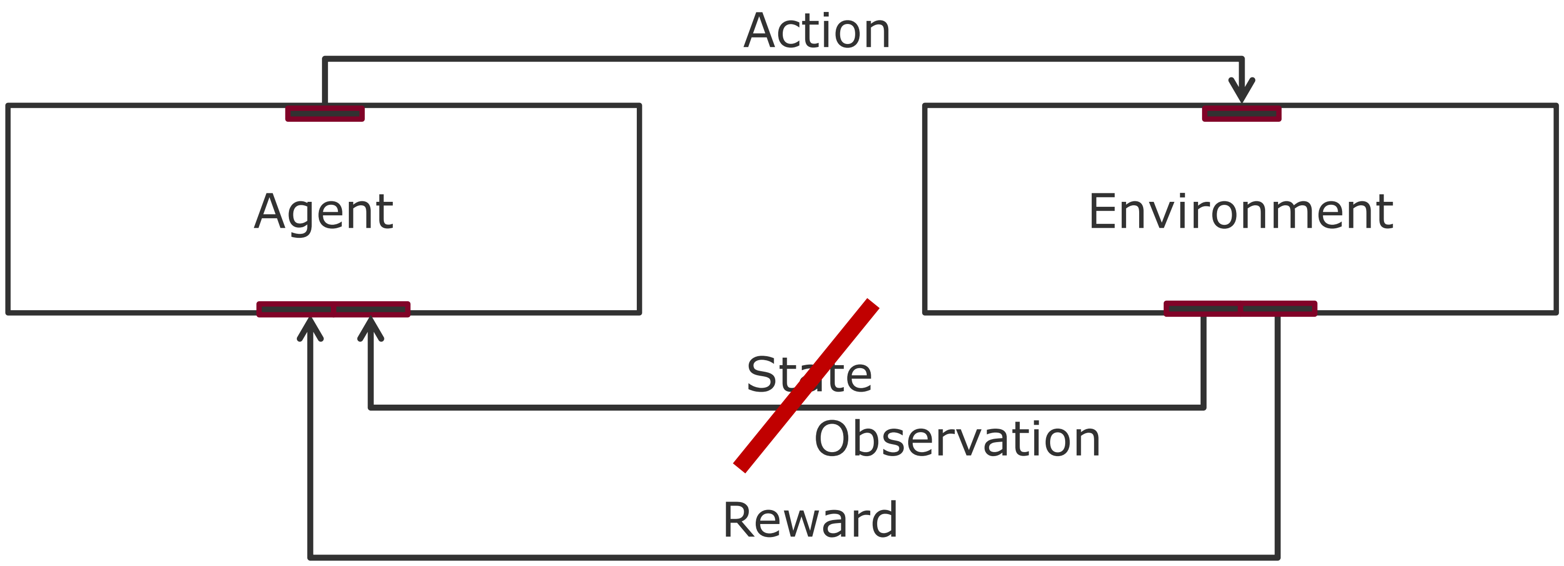
hence, the state summarizes all information needed to make decisions

<u>Intuition</u> (why is this a good idea?)

- Conditioned on the histories the actions are independent.
- This allows us to factor one decision into small decisions, or actions.

$$\pi(a_{0:T}) = \prod_{t=0}^{T} \pi(a_t | h_t)$$

Agent and Environment Interaction



Agent's Goal: Maximize its reward

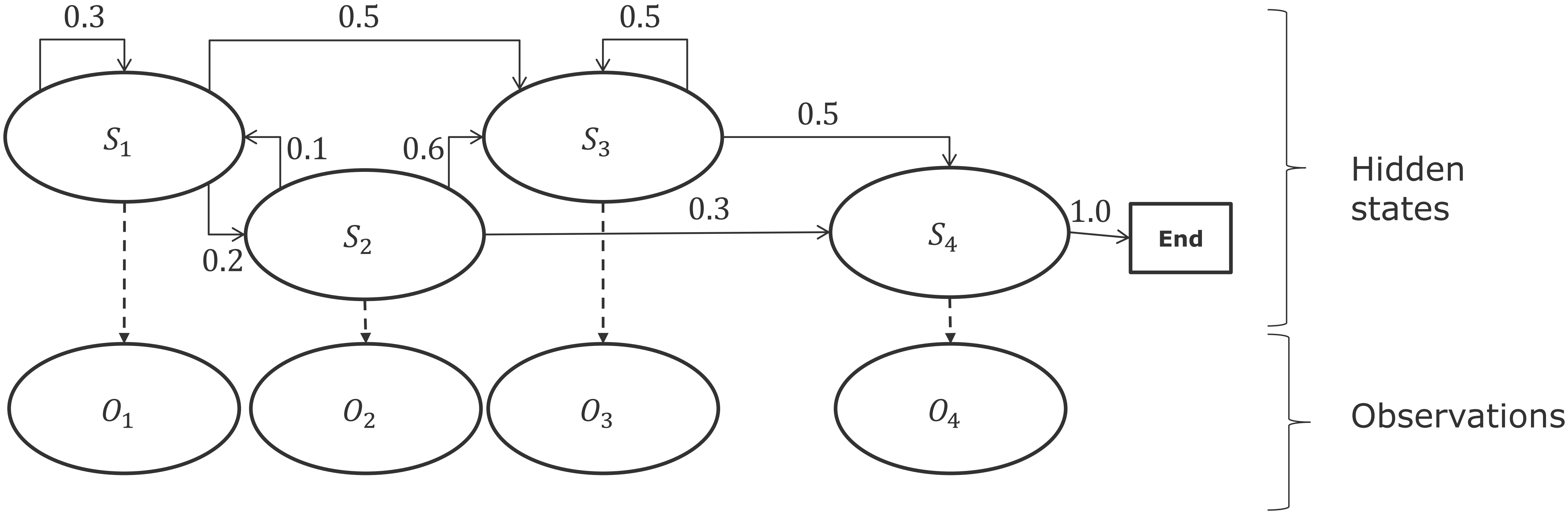
Our Goal: Discover a strategy that allows the agent to achieve its goal under various circumstances

Search, Estimate, Learn

Policy = $\pi(S, A)$

knowlege about the environment

Hidden Markov Model



Assumptions

First order Markovian: $P(S_{t+1} | S_t) = P(S_{t+1} | S_t, S_{t-1}, \ldots, S_{t-n})$

Stationary:

$P(S_{t+1} | S_t) = P(S_t | S_{t-1})\ \forall t$ hidden states

$P(O_{t+1} | S_t) = P(O_t | S_{t-1})\ \forall t$ observations

Definition for HMM

Has a graphical representation

Has a parameterization:

- initial distribution $P(S_i)$ //multinomial
- Transition distribution $P(S_{i+1}|S_i)$ //multinomial
- Emission distribution $\mathrm{P}(O_i|S_i)$ //Gaussian for continuous case, or multinomial for discrete case

Joint distribution

$$P(S_{1\dots t}, O_{1\dots t}) = P(S_1) * \prod_{i=1}^{t-1} P(S_{i+1}|S_i) \prod_{i=1}^{t} P(O_i|S_i)$$

Applications of Hidden Markov Model (HMM)

Speech recognition

Robot localization

Patient monitoring

Monitoring Algorithm

Weather prediction

Stock market prediction

Prediction Algorithm

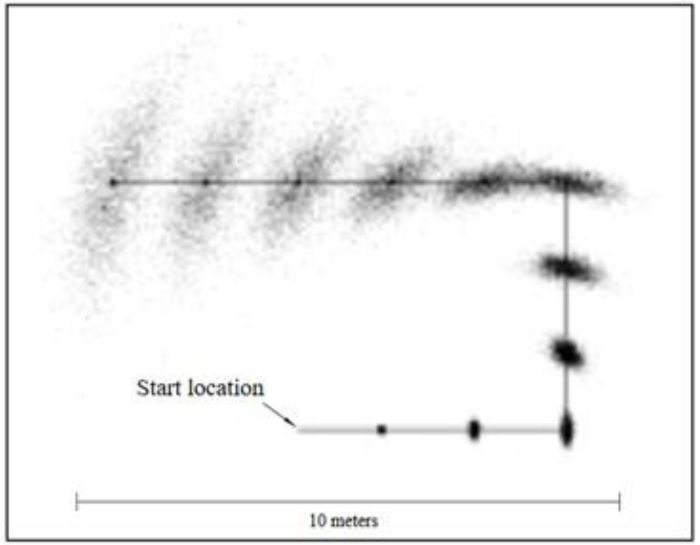
Other algorithms:

Robot Localization

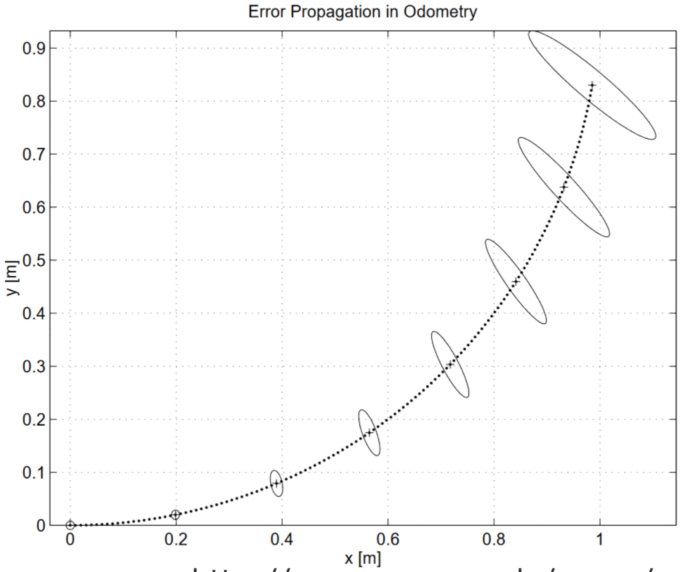
Given the following…

- S = coordinates of the robot in space
- O = distances to objects measured by a lidar (laser beam)
- $P(S_t|S_{t-1})$: transitions of the robot (inherently uncertain)
- $P(O_t|S_t)$: uncertainty in the measurements provided by the laser beam

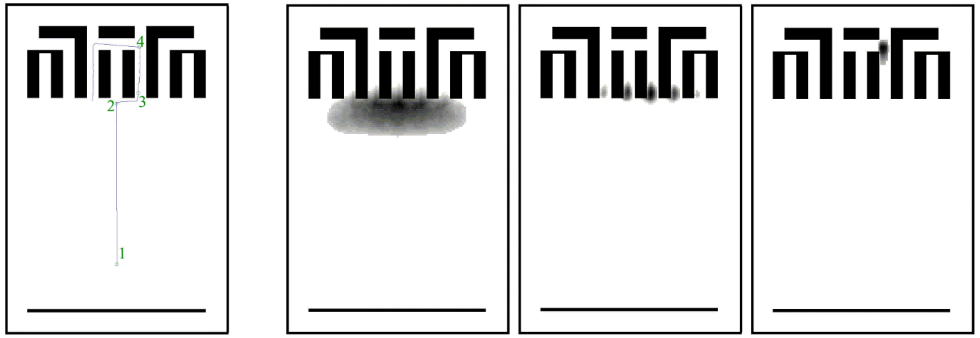
What is the robot's localization $P(S_t|O_{1\ldots t})$?



source: Fox et al., 1999, Monte Carlo Localization: Efficient Position Estimation for Mobile Robots



source: http://www.cs.cmu.edu/~rasc/Download/AMRobots5.pdf

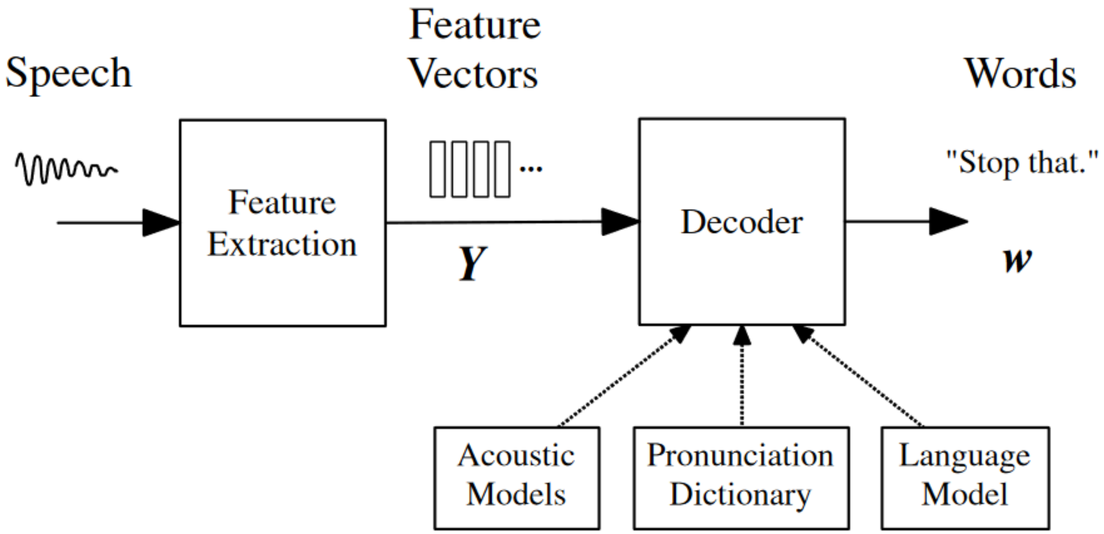


Path of the robot *Belief states at positions 2, 3 and 4*

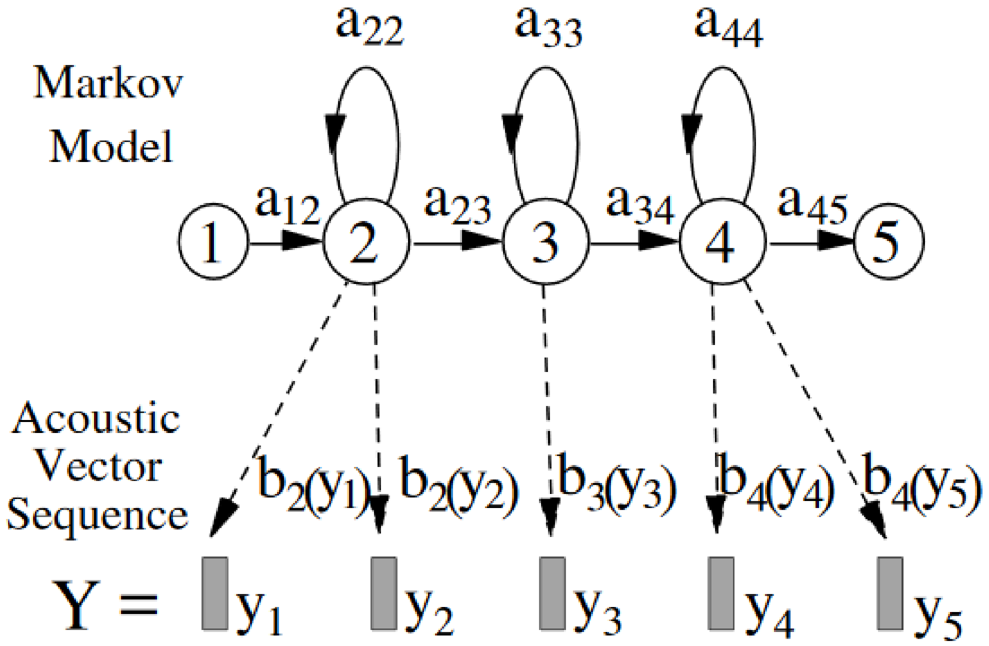
source: Choset et al., (2005). *Principles of robot motion: theory, algorithms, and implementation*. MIT press.

Speech recognition

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam



Each spoken word **w** is decomposed into a sequence of q^w basic sounds called base phones. This sequence is called its pronunciation $q(w)1{:}q^w = q^1, \ldots, q^w$. To allow for the possibility of multiple pronunciations, the likelihood $p(Y|w)$ c… …ns.



$\overline{Q}$

where the summation is over all valid pronunciation sequences for w, Q is a particular sequence of pronunciations

$$P(\boldsymbol{Q}|\boldsymbol{w}) = \prod_{l=1}^{L} P(\boldsymbol{q}^{(w_l)}|w_l),$$

and where each $q(w_l)$ is a valid pronunciation for word w_l

source: Gales, M., & Young, S. (2008). The application of hidden Markov models in speech recognition. Now Publishers Inc.

Monitoring algorithm

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

$Pr(Y_t|X_{1...t})$: Distribution over current state (Y_t) given observations $X_{1...t}$

Recursive Computation:

$$\Pr(y_t|x_{1..t}) \propto \Pr(x_t|y_t, x_{1..t-1})\Pr(y_t|x_{1..t-1}) \text{ by Bayes' thm}$$

$$= \Pr(x_t|y_t) \Pr(y_t|x_{1..t-1}) \text{ by conditional independence}$$

$$= \Pr(x_t|y_t) \textstyle\sum_{y_{t-1}} \Pr(y_t, y_{t-1}|x_{1..t-1}) \text{ by marginalization}$$

$$= \Pr(x_t|y_t) \textstyle\sum_{y_{t-1}} \Pr(y_t|y_{t-1}, x_{1..t-1}) \Pr(y_{t-1}|x_{1..t-1})$$

by chain rule

$$= \underbrace{\Pr(x_t|y_t)}_{\text{Prediction of my observation}} \textstyle\sum_{y_{t-1}} \underbrace{\Pr(y_t|y_{t-1})}_{\text{State transitions}} \underbrace{\Pr(y_{t-1}|x_{1..t-1})}_{\text{State probability given history of observations}} \text{ by cond ind}$$

Forward computation

Compute $\Pr(Y_t|X_{1...t})$ by forward computation:

$\Pr(y_1|x_1) \propto \Pr(x_1|y_1)\Pr(y_1)$

For $i = 2$ to t do

$\Pr(y_i|x_{1..i}) \propto \Pr(x_i|y_i)\sum_{y_{i-1}}\Pr(y_i|y_{i-1})\Pr(y_{i-1}|x_{1..i-1})$

End

From Bayes Rule

From the Recursive Computation

source: CS480/680 Spring 2019 Pascal Poupart

Prediction Algorithm

$Pr(Y_{t+k}|X_{1...t})$: Distribution over **future states** (Y_{t+k}) given observations $X_{1...t}$

Recursive Computation:

$$\Pr(y_{t+k}|x_{1..t}) = \sum_{y_{t+k-1}} \Pr(y_{t+k}, y_{t+k-1}|x_{1..t}) \text{ by marginalization}$$

$$= \sum_{y_{t+k-1}} \Pr(y_{t+k}|y_{t+k-1}, x_{1..t}) \Pr(y_{t+k-1}|x_{1..t}) \text{ by chain rule}$$

$$= \sum_{y_{t+k-1}} \Pr(\underline{y_{t+k}|y_{t+k-1}}) \Pr(\underline{y_{t+k-1}}|x_{1..t}) \text{ by cond ind}$$

Future State transitions

Future State probability given history of observations

Forward computation

1. Compute $\Pr(Y_t|X_{1\ldots t})$ by forward computation (same step as before).

$\Pr(y_1|x_1) \propto \Pr(x_1|y_1)\Pr(y_1)$
For $i = 1$ to t do

$$\Pr(y_i|x_{1..i}) \propto \Pr(x_i|y_i)\sum_{y_{i-1}}\Pr(y_i|y_{i-1})\Pr(y_{i-1}|x_{1..i-1})$$

End

2. Compute $\Pr(Y_{t+k}|X_{1\ldots t})$ by forward computation

For $j = 1$ to k do

$$\Pr(y_{t+j}|x_{1..t}) = \sum_{y_{t+j-1}}\Pr(y_{t+j}|y_{t+j-1})\Pr(y_{t+j-1}|x_{1..t})$$

End

Topics

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

- Review of Baum-Welch Algorithm
- Example
- HMMLearn API
- Tasks

HMM definitions used

$$HMM = (S, O, A, B, \pi)$$

- S= the hidden states
- O= the set of observations
- A= the transition matrix among states
- B= the probability of each state generating an observation (also called emission probability)
- π = the probability of being at any given state

Reference:
Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257-286.

How the HMM works

1. At each step t (out of a total T steps) we have one observation O_t emitted by a state S_t and a transition to state S_{t+1}

2. Transitions between states are given by transition matrix A, e.g., $a_{t,t+1}$ is the transition probability between state S_t and state S_{t+1}

3. Emission probabilities are given by emission matrix B, e.g., $b_t(i)$ is the emission probability of state S_t $generating\ observation\ O_i$

4. Note, that any state could generate any observation, so we would sometimes write emission probabilities as $O_t(i)$, where t is the state index and i is the observation index.

Baum-Welch algorithm GOAL

What is the HMM model $\bar{\theta}$ that most probably generated a given sequence of observations O?

- $\bar{\theta}$: learned HMM model
- $O = Oo_1 \to Oo_2 \to Oo_1 \to Od_1 \to Ou_1 \to Ou_2 \to Oo_1 \to Od_1 \to Od_2 \to Ou_2 \to Oo_1$

Baum-Welch Algorithm: Components

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

$\theta = (S, O, A, B, \pi)$: prior HMM model

O: sequence of observations

$\alpha_t(i)$:computed by the forward step

$P(O, S_t = s_i | \theta)$ probability of arriving at state s_i after a set of observations

$\beta_t(i)$:computed by the backward step

$P(O | S_t = s_i, \theta)$ probability of seeing a set of observations if I start at state s_i

$\gamma_t(i)$: $P(S_t = s_i | O, \theta)$ having seen observations ahead and behind, what is the probability of being at state s_i

Now how to aggregate these estimates in an iterative ways?

$$\xi_t(i,j) = P\Big((S_t = S_i, S_{t+1} = S_j)\Big|(O,\theta)\Big)$$

$$\xi_t(i,j) = \frac{\alpha_t(i) * a_{ij} * b_j(O_{t+1}) * \beta_{t+1}(j)}{P(O|\theta)}$$

$P(O|\theta) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) * a_{ij} * b_j(O_{t+1}) * \beta_{t+1}(j)$, normalizing factor

$$\xi_t(i,j) = \frac{\alpha_t(i) * a_{ij} * b_j(O_{t+1}) * \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) * a_{ij} * b_j(O_{t+1}) * \beta_{t+1}(j)}$$

Baum-Welch Algorithm: Iterative Procedure

Algorithm 2: The Baum-Welch algorithm

Initialization:

$\Theta_0, \{O_{1:T}\}$

Looping:

for $l = 1, ..., l_{max}$ **do**

1. Forward-Backward calculations:

$$\alpha_1(i) = \pi_i b_i(O_1),\ \beta_T(i) = 1,$$
$$\alpha_t(i) = \left[\sum_{j=1}^{K} \alpha_{t-1}(j) a_{ji}\right] b_j(O_t),\ \beta_t(i) = \sum_{j=1}^{K} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

for $1 \leq i \leq K,\ 1 \leq t \leq T-1$

2. E-step:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{K} \alpha_t(j)\beta_t(j)},\ \xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}\beta_t(j))}$$

for $1 \leq i \leq K,\ 1 \leq j \leq K,\ 1 \leq t \leq T-1$

3. M-step:

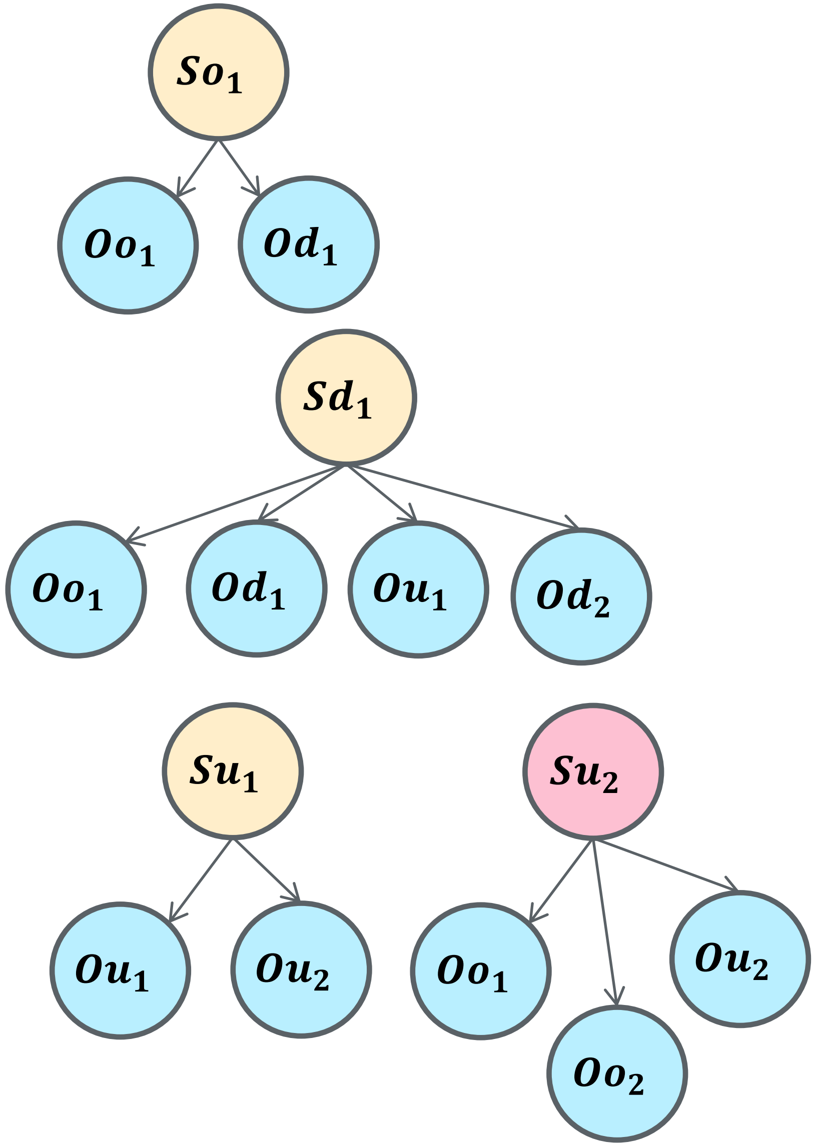
$$\pi_i = \frac{\gamma_1(i)}{\sum_{j=1}^{K} \gamma_1(j)},\ a_{ij} = \frac{\sum_{t=1}^{T} \varepsilon_t(i,j)}{\sum_{k=1}^{K}\sum_{t=1}^{T} \varepsilon_t(i,k)},\ w_{kd} = \frac{\sum_{t=1}^{T} \gamma_t(k,d)}{\sum_{t=1}^{T}\sum_{r=1}^{D} \gamma_t(k,r)}$$

for $1 \leq i \leq K,\ 1 \leq j \leq K,\ 1 \leq k \leq K, 1 \leq d \leq D$

end

Result: $\{\Theta_l\}_{l=0}^{l_{max}}$

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam



Emission Matrix B

Probability that a given state S will produce an observation O

Intermittent cycle

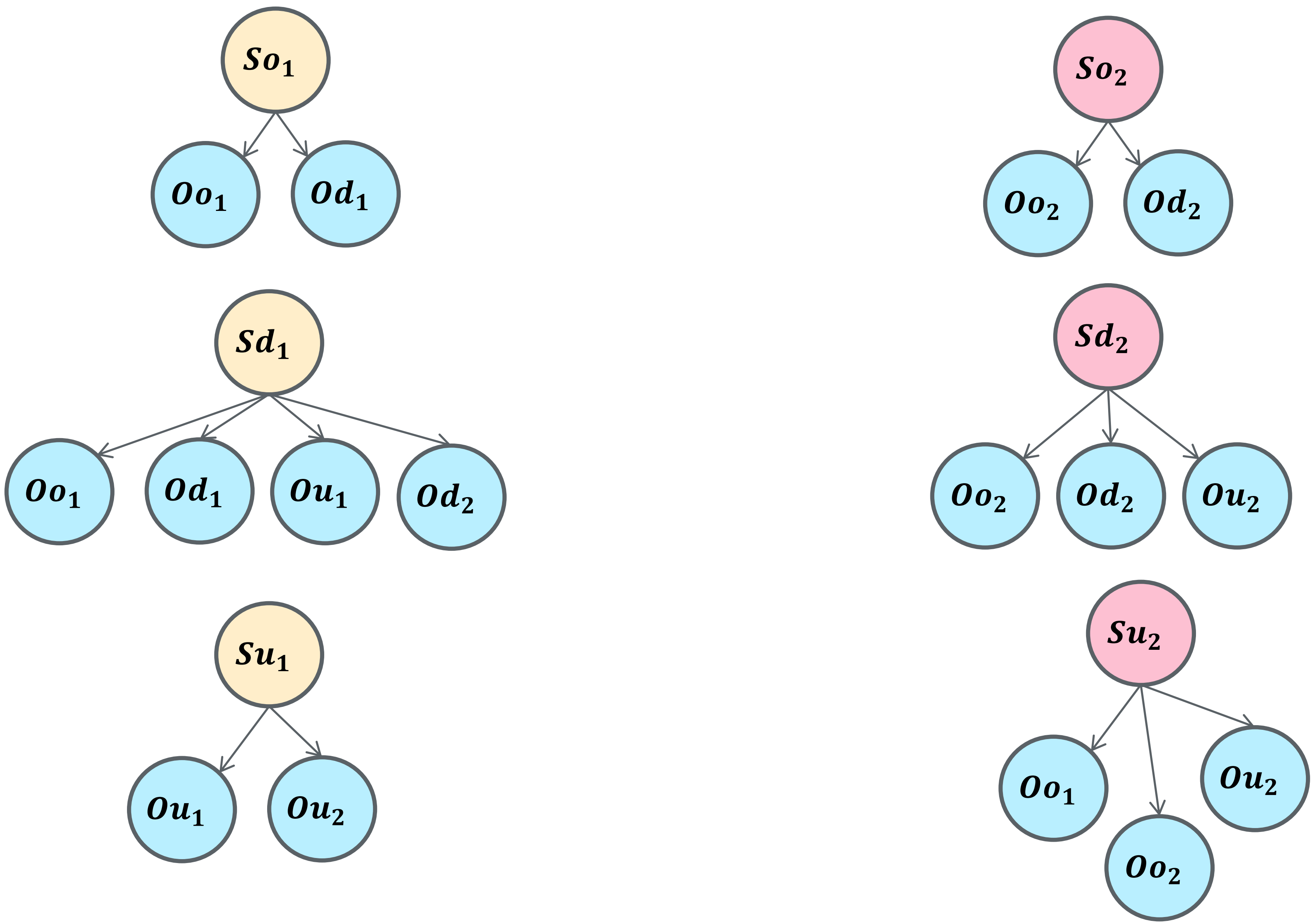
Effect of Failure Masking

States

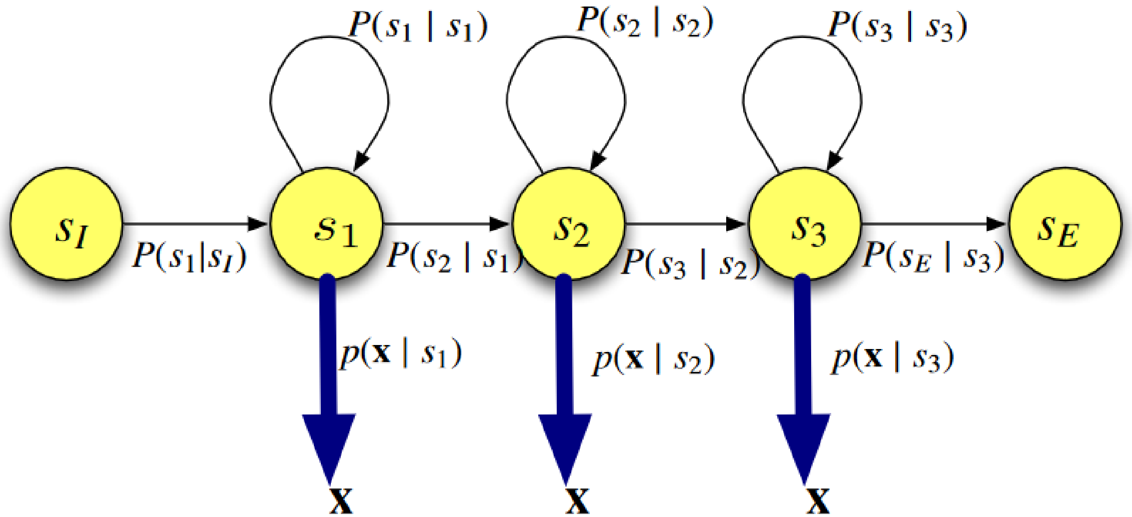
Observations

	So_1	Sd_1	Su_1	So_2	Sd_2	Su_2
Oo_1	0.80	0.10	0.0	0.0	0.0	0.20
Od_1	0.20	0.60	0.0	0.0	0.0	0.0
Ou_1	0.0	0.10	0.70	0.0	0.0	0.0
Oo_2	0.0	0.0	0.0	0.90	0.10	0.10
Od_2	0.0	0.20	0.0	0.10	0.60	0.0
Ou_2	0.0	0.0	0.30	0.0	0.30	0.70
$\sum$	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Emission Scenarios



Output Distribution



- Single multivariate Gaussian with mean $\boldsymbol{\mu}_j$, covariance matrix $\boldsymbol{\Sigma}_j$:

$$b_j(\mathbf{x}) = p(\mathbf{x} \,|\, S = j) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

- M-component Gaussian mixture model:

$$b_j(\mathbf{x}) = p(\mathbf{x} \,|\, S = j) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm})$$

Tool: Hmm Learn

- Initializing and Sampling a Model

```
>>> import numpy as np
>>> from hmmlearn import hmm
>>> np.random.seed(42)
>>>
>>> model = hmm.GaussianHMM(n_components=3, covariance_type="full")
>>> model.startprob_ = np.array([0.6, 0.3, 0.1])
>>> model.transmat_ = np.array([[0.7, 0.2, 0.1],
...                             [0.3, 0.5, 0.2],
...                             [0.3, 0.3, 0.4]])
>>> model.means_ = np.array([[0.0, 0.0], [3.0, -3.0], [5.0, 10.0]])
>>> model.covars_ = np.tile(np.identity(2), (3, 1, 1))
>>> X, Z = model.sample(100)
```

- Learning

```
>>> remodel = hmm.GaussianHMM(n_components=3, covariance_type="full", n_iter=100)
>>> remodel.fit(X)
GaussianHMM(...
>>> Z2 = remodel.predict(X)
```

Task-1

1. Apply the Baum-Welch algorithm **to learn a model** that allows to predict the traces of observations produced by your DTMC

Use the learned model to answer the following questions.

2 What is the probability that your model has generated the following sequence of observations:

1. Intermittent failure in component-1 $Oo_1 \rightarrow Od_1 \rightarrow Oo_1$
2. Intermittent failure in component-2 $Oo_2 \rightarrow Od_2 \rightarrow Oo_2$
3. Failure cascade $Od_1 \rightarrow Od_2 \rightarrow Ou_2$
4. Failure cascade $Od_1 \rightarrow Ou_1 \rightarrow Ou_2$
5. Failure masking $Ou_1 \rightarrow Ou_2 \rightarrow Oo_1$

3 What is the most probable sequence of states for each of the five above observations?

Task-2

1. Apply the Baum-Welch algorithm to learn a model for your mRubis scenario
2. Generate a sequence of observations from your DTMC that correspond to the three scenarios:
   - Intermittent failure
   - Failure cascade
   - Failure masking

Use your learned model to

3. Compute the probability of each scenario
4. Estimate the sequence of states that most probably generated each observation trace

Some derivations

1. What is the probability that a model generated a sequence of observations?

$$P(O|\theta), where\ O = O_1 \rightarrow O_2 \rightarrow \ \dots \rightarrow O_{T-1} \rightarrow O_T, \theta = (S, O, A, B, \pi), S = S_1 \rightarrow \ S_2 \rightarrow \cdots \rightarrow S_{T-1} \rightarrow S_T$$

$$P(O|(S,\theta)) = \prod_{t \in T} P(O_t|(S_t,\theta))$$

2. What is the probability of seeing the sequence of observations O and sequence of states S?

$P((O,S)|\theta) = P(O|(S,\theta)) * P(S|\theta)$ Chain Rule

Expression of seeing a particular sequence of observations O given sequence of states S and model θ:

$P(O|(S,\theta)) = b_{s1}(O_1) * b_{s2}(O_2) * \cdots * b_{sT}\ (O_T)$

Expression of seeing a particular sequence of states S given a model θ:

$$P(S|\theta) = \pi_{s1} * a_{s1} * a_{s2} * a_{s2} * a_{s3} * \cdots * a_{sT-1} * a_{sT}$$

$$P(O|\theta) = \sum_{i \in P} P(O|(S_i,\theta)) * P(S_i|\theta)\ , where\ P\ is\ permutations\ of\ states\ S_t$$

Computing $P(O|\theta)$

- $P(O|\theta) = \sum_{i \in P} P(O|(S_i, \theta)) * P(S_i|\theta)$ *, where P is permutations of states* S_t

Intractable! , given T observations and N states, we get:

- State sequences = N^T
- Multiplications per state = $2T - 1$
- Total summations and multiplications = $(2T - 1)N^T + (N^T - 1)$

If $T = 6$ observations and $N = 4$ states, we get

$(2 * 6 - 1) * 4^6 + (4^6 - 1)$

$11 * 4096 + 4096 - 1 = 49152$ *summation and multiplications*

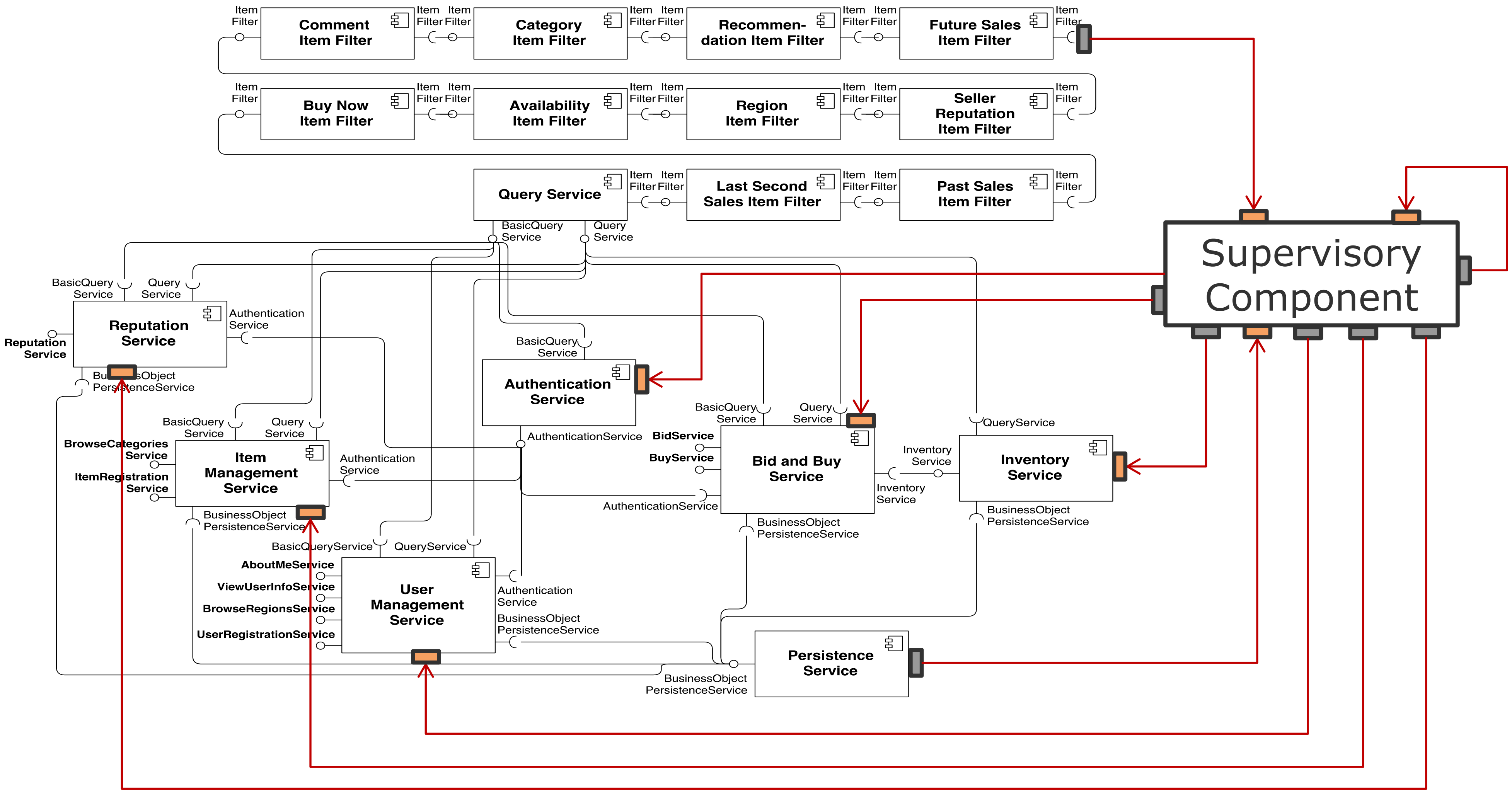
If $T = 18 * 3 = 54$ and $N = 18 * 2 = 36$

$(2 * 54 - 1) * 36^{54} + (36^{54} - 1)$

$108 * 1.09 * 10^{84} + 1.09 * 10^{84} - 1 \sim 10^{86}$ *summation and multiplications*

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

End


Comment Item Filter
Category Item Filter
Recommendation Item Filter
Future Sales Item Filter
Buy Now Item Filter
Availability Item Filter
Region Item Filter
Seller Reputation Item Filter
Query Service
Last Second Sales Item Filter
Past Sales Item Filter
Item Filter
BasicQuery Service
Query Service
Supervisory Component
Reputation Service
Authentication Service
BusinessObject PersistenceService
BrowseCategories Service
ItemRegistration Service
Item Management Service
Bid and Buy Service
BidService
BuyService
AuthenticationService
Inventory Service
QueryService
BasicQueryService
AboutMeService
ViewUserInfoService
BrowseRegionsService
UserRegistrationService
User Management Service
Persistence Service