ECE 286: Bayesian Machine Perception

Class 11: Graph-Based Multiobject Tracking II

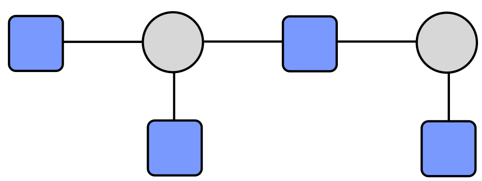
Florian Meyer

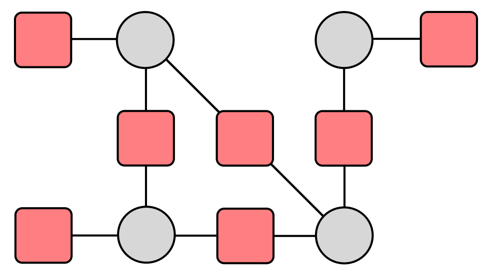
Electrical and Computer Engineering Department
University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Tree vs Cyclic Graph

- A factor graph represents the statistical model of an inference problem
- Message passing (the sum-product algorithm) on a factor graph can strongly reduce the computational complexity of calculating marginal distributions
- Marginalization is exact if the **factor graph is a tree** and approximate if the **factor graph has cycles**

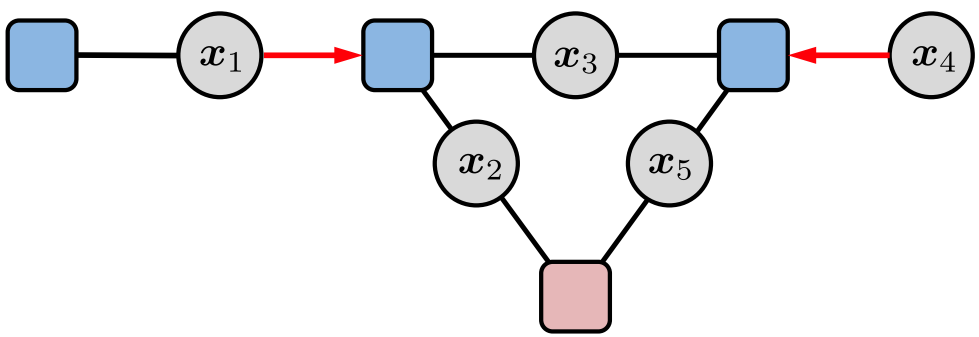




- The factor graph is often not unique. A more ``detailed'' graph
  - has lower-dimensional operations ⟶ lower computational complexity
  - may introduce additional cycles ⟶ lower inference accuracy

Factor Graphs with Cycles

- Problem: When the factor graphs has cycles, message passing gets stuck
- Solution: Determine message passing order by introducing artificial constant messages
- New Problem 1: Message passing keeps running forever
- Solution: Stop message passing after some time and compute marginals
- New Problem 2: Message passing tend to very large or very small values
  $\longrightarrow$ numerical issues
- Solution: After calculating a message, normalize it



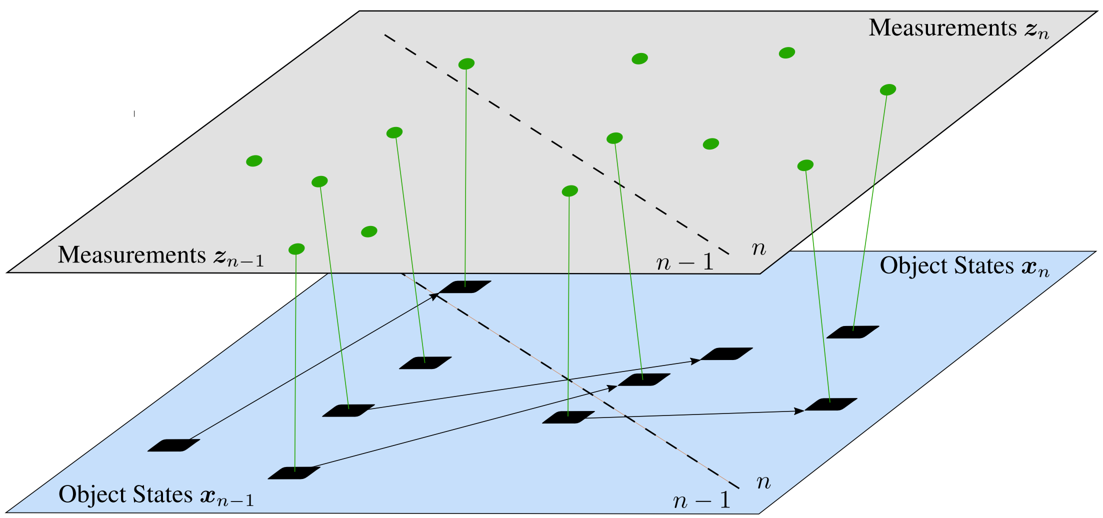
Factor Graphs with Cycles

- It turns out that when we normalize messages
  - the sum-product algorithm (SPA) can still give good results
  - marginal posterior distributions are not exact, but approximations (except means for Gaussian models)
- Approximate marginal distributions are called ``beliefs''
- The SPA on loopy graphs can be derived by approximating the posterior by the ``Bethe free energy''
- Theoretical performance analysis is notoriously difficult in general
- Many practical applications of the SPA involve factor graphs with cycles: turbo codes, LDPB codes, MIMO detection, cooperative localization, data association, …

The Multiobject Tracking Problem

- At each time n: localize and track multiple objects $\boldsymbol{x}_n = [\boldsymbol{x}_{1,n}^{\mathrm{T}} \dots \boldsymbol{x}_{I,n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{1,n}^{\mathrm{T}} \dots \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin

- Data association is challenging because of false clutter measurements and missing measurements



Prior Distributions

- Assumptions:
  1. Object detections are independent Bernoulli trials with success probability $0 < p_{\mathrm{d}} \leqslant 1$
  2. The number of clutter measurements is Poisson distributed with mean μ_{c}
  3. At most one measurement is generated by each object
  4. A measurement can be generated from at most one object
- Assumptions 1-3 are parallel to the single object tracking case
- Every association event expressed by a vector $\boldsymbol{a}_n = [a_{1,n} \dots a_{I,n}]^{\mathrm{T}}$ automatically fulfills Assumption 3 (scalar association variable $a_{i,n}$ for each object)
- Assumption 4 can be enforced by the following check function

$$\varphi(\boldsymbol{a}_n) \triangleq \begin{cases} 0, & \exists i, j \in \{1, 2, \dots, I\} \text{ such that } i \neq j \text{ and } a_{i,n} = a_{j,n} \neq 0 \\ 1, & \text{otherwise} \end{cases}$$

The Factor Graph

- Recall factorization of the joint posterior distribution:

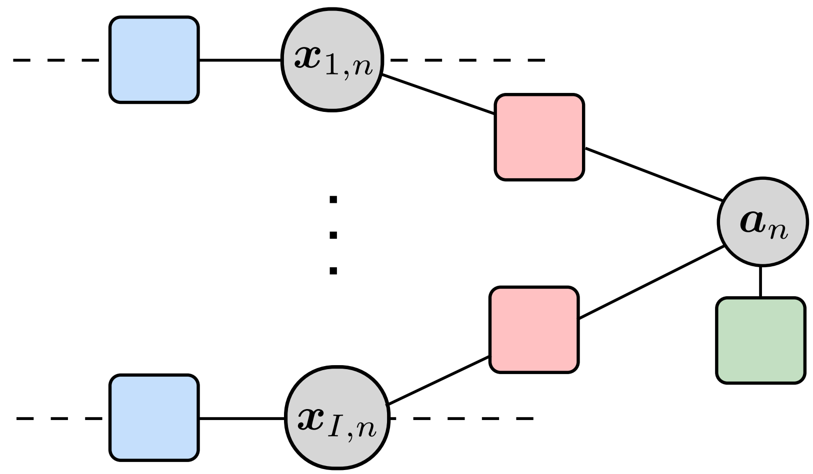
$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) \propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \, g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'}) \right) \varphi(\boldsymbol{a}_{n'})$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Message Passing Order

- Recall Prediction step:

$$\phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) = \int f(\boldsymbol{x}_{i,n}|\boldsymbol{x}_{i,n-1})\, \nu_{\rightarrow i,n}(\boldsymbol{x}_{i,n-1})\, \mathrm{d}\boldsymbol{x}_{i,n-1}$$

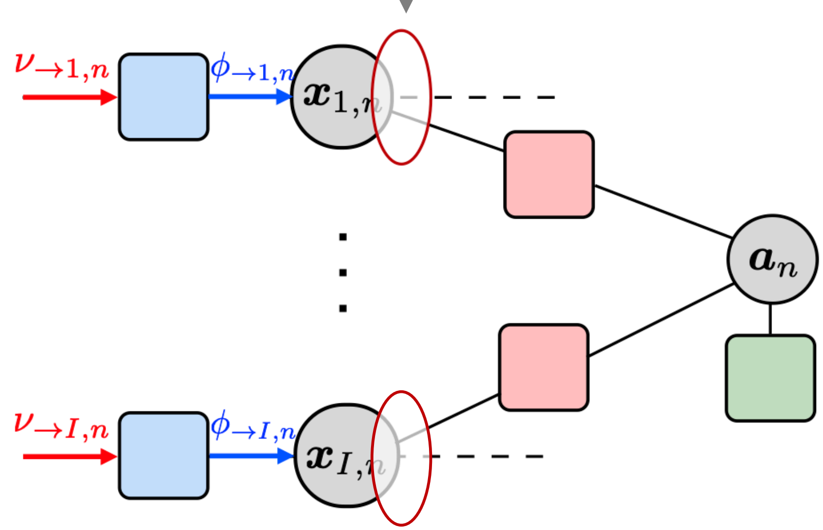
- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$

Here we would get stuck since messages from two edges are not available



© Florian Meyer, 2020

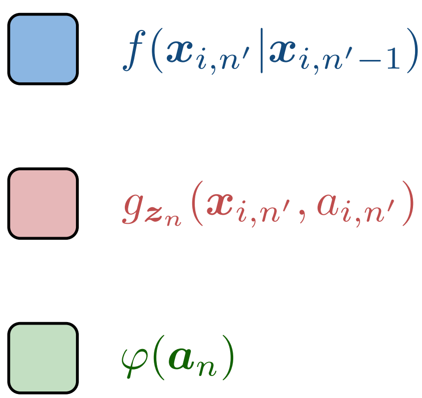
Message Passing Order

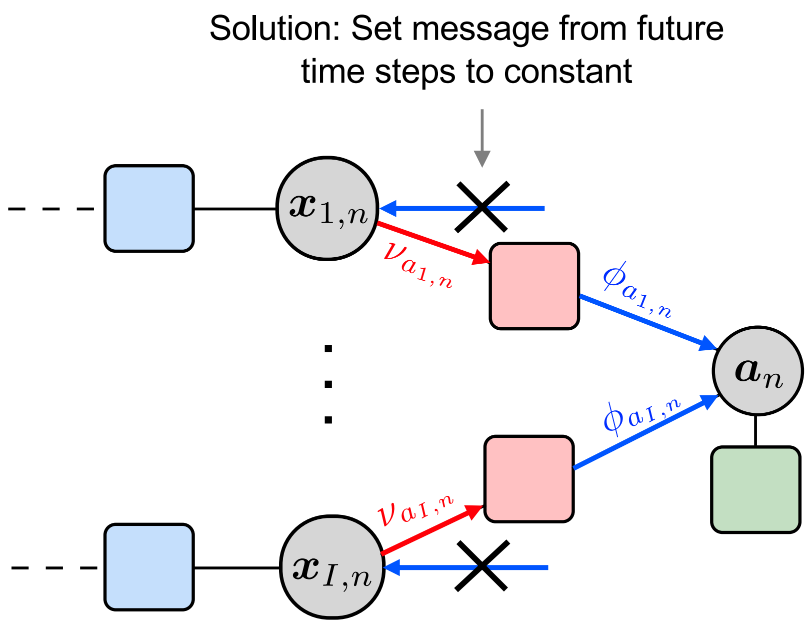
- Measurement evaluation:

$$\nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) = \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$$

$$\phi_{a_{i,n}}(a_{i,n}) = \int g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) \mathrm{d}\boldsymbol{x}_{i,n}$$

- Factor graph for time step n





© Florian Meyer, 2020

Update Step

- Update step:

$$\tilde{f}(\boldsymbol{x}_{i,n}) \propto \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})\,\phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n})$$

$$\boxed{f(\boldsymbol{x}_{i,n}|\boldsymbol{z}_{1:n}) \approx \tilde{f}(\boldsymbol{x}_{i,n})}$$

$$\nu_{\rightarrow i,n+1}(\boldsymbol{x}_{i,n}) = \phi_{i,n}(\boldsymbol{x}_{i,n})\,\phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$$

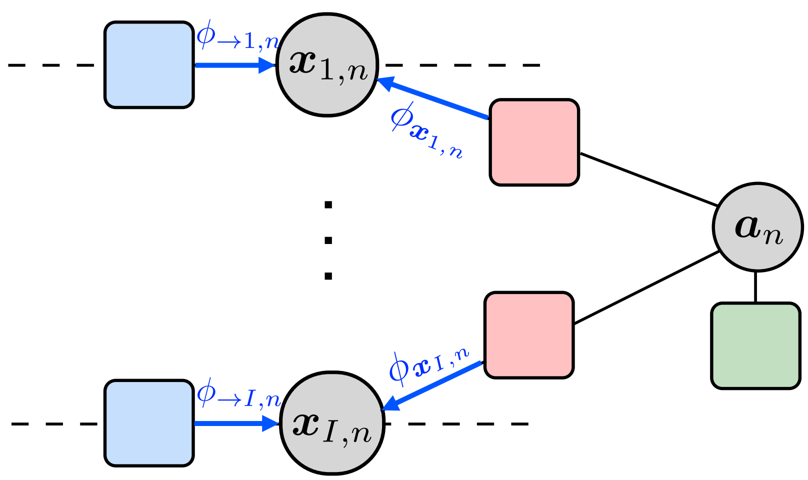
approx. since factor graph is not cycle-free

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Update Step

- Update step revised

$$\tilde{f}(\boldsymbol{x}_{i,n}) \propto \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})\boxed{\phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n})}$$

Data association message derived in class 10

$$= \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{\boldsymbol{a}_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)$$

$$= \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{a_{i,n}=0}^{M_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n}) \underbrace{\sum_{a_{1,n}=0}^{M_n} \cdots \sum_{a_{i-1,n}=0}^{M_n} \sum_{a_{i+1,n}=0}^{M_n} \cdots \sum_{a_{I,n}=0}^{M_n} \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)}_{\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})}$$

$$= \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{a_{i,n}=0}^{M_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$$

$$= \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{a_{i,n}=0}^{M_n} \tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})$$

influence of other objects in the environment

where $\tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n}) = g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\boxed{\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})}$

- The single object updated step of the multiobject tracking solution has the same form as the single object tracking in clutter update step

Multiobject Tracking Filters

- Let's assume at time n, approximate posteriors $\tilde{f}(\boldsymbol{x}_{i,n-1}) \approx f(\boldsymbol{x}_{i,n-1}|\boldsymbol{z}_{1:n-1})$ for all objects $i \in \{1,\ldots,I\}$ are available

- We can develop a multiobject tracking algorithm by performing for each $i \in \{1,\ldots,I\}$
  - the conventional prediction step, i.e., $\phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) = \int f(\boldsymbol{x}_{i,n}|\boldsymbol{x}_{i,n-1})\,\tilde{f}(\boldsymbol{x}_{i,n-1})\mathrm{d}\boldsymbol{x}_{i,n-1}$
  - calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$
  - the update step of the single object tracking (in clutter) solution where $g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})$ is replaced by $\tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n}) = g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\,\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

- Multiobject tracking is based on the calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

$\longrightarrow$ Joint probabilistic data association

Closed-Form Update Step (cf. Class 9)

- Step 1: Calculate means and covariances of mixture components:

$$\boldsymbol{\mu}_m = \boldsymbol{\mu}^-_{\boldsymbol{x}_{i,n}} + \boldsymbol{K}_{i,n}\big(\boldsymbol{z}_{m,n} - \boldsymbol{H}_{i,n}\boldsymbol{\mu}^-_{\boldsymbol{x}_{i,n}}\big) \quad m = 1,\ldots,M_n$$

$$\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_{i,n}} - \boldsymbol{K}_{i,n}\boldsymbol{H}_{i,n}\boldsymbol{\Sigma}^-_{\boldsymbol{x}_{i,n}}$$

$$\boldsymbol{K}_{i,n} = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_{i,n}}\boldsymbol{H}^{\mathrm{T}}_{i,n}\big(\boldsymbol{H}_{i,n}\boldsymbol{\Sigma}^-_{\boldsymbol{x}_{i,n}}\boldsymbol{H}^{\mathrm{T}}_{i,n} + \boldsymbol{\Sigma}_{\boldsymbol{v}_{i,n}}\big)^{-1}$$

$$\boldsymbol{\mu}_{M_n+1} = \boldsymbol{\mu}^-_{\boldsymbol{x}_{i,n}}$$

$$\boldsymbol{\Sigma}_{M_n+1} = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_{i,n}}$$

- Step 2: Calculate unnormalized weights:

$$\tilde{w}_m = \frac{p_{\mathrm{d}} f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{H}_{i,n}\boldsymbol{\mu}^-_{\boldsymbol{x}_{i,n}}, \boldsymbol{H}_{i,n}\boldsymbol{\Sigma}^-_{\boldsymbol{x}_{i,n}}\boldsymbol{H}^{\mathrm{T}}_{i,n} + \boldsymbol{\Sigma}_{\boldsymbol{v}_{i,n}})\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n} = m)}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})} \quad m = 1,\ldots,M_n$$

$$\tilde{w}^{(i)}_{M_n+1} = (1 - p_{\mathrm{d}})\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n} = 0)$$

- Step 3: Normalize weights: $w_m = \tilde{w}_m / \big(\sum_{m'=1}^{M_n+1} \tilde{w}_{m'}\big)$

- Step 4: Approximate Gaussian mixture by a single Gaussian with same mean and covariance (moment matching):

$$\boldsymbol{\mu}_{\boldsymbol{x}_{i,n}} = \sum_{m=1}^{M_n+1} w_m \boldsymbol{\mu}_m$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_{i,n}} = \sum_{m=1}^{M_n+1} w_m \boldsymbol{\Sigma}_m + \sum_{m=1}^{M_n+1} w_m \boldsymbol{\mu}_m \boldsymbol{\mu}_m^{\mathrm{T}} - \boldsymbol{\mu}_{\boldsymbol{x}_{i,n}} \boldsymbol{\mu}^{\mathrm{T}}_{\boldsymbol{x}_{i,n}}$$

- **Result:** Mean $\boldsymbol{\mu}_{\boldsymbol{x}_{i,n}}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{i,n}}$ representing the posterior distribution $\tilde{f}(\boldsymbol{x}_{i,n})$

Y. Bar-Shalom, F. Daum, and J. Huang, *The Probabilistic Data Association Filter*, IEEE Contr. Syst. Mag., 2009

Particle-Based Update Step (cf. Class 4)

- **Given:** Particles $\{(\boldsymbol{x}_{i,n}^{(j)})\}_{j=1}^{J} \simeq \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$ representing the predicted posterior PDF

- **Wanted:** Particles $\{(\overline{\boldsymbol{x}}_{i,n}^{(j)})\}_{j=1}^{J} \simeq \tilde{f}(\boldsymbol{x}_{i,n})$ representing the posterior PDF

- Perform importance sampling with proposal distribution $f_{\mathrm{p}}(\boldsymbol{x}_{i,n}) = \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$ and target distribution $f_{\mathrm{t}}(\boldsymbol{x}_{i,n}) \propto \phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n})\phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) = \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{m=0}^{M_n} \tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n} = m)$
  - calculate unnormalized weights $\tilde{w}_{i,n}^{(j)} = \boxed{\sum_{m=0}^{M_n} \tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}^{(j)}, a_{i,n} = m)} \propto f_{\mathrm{t}}(\boldsymbol{x}_{i,n}^{(j)})/f_{\mathrm{p}}(\boldsymbol{x}_{i,n}^{(j)})$
  - normalize weights $w_{i,n}^{(j)} = \tilde{w}_{i,n}^{(j)} / \sum_{j'=1}^{J} \tilde{w}_{i,n}^{(j')}, \quad j = 1, \dots, J$

- Perform resampling to get $\{(\overline{\boldsymbol{x}}_{i,n}^{(j)})\}_{j=1}^{J} \simeq \tilde{f}(\boldsymbol{x}_{i,n})$ from $\{(\boldsymbol{x}_{i,n}^{(j)}, w_{i,n}^{(j)})\}_{j=1}^{J} \simeq \tilde{f}(\boldsymbol{x}_{i,n})$

Joint Probabilistic Data Association

- Recall measurement evaluation step:

$$\nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) = \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$$

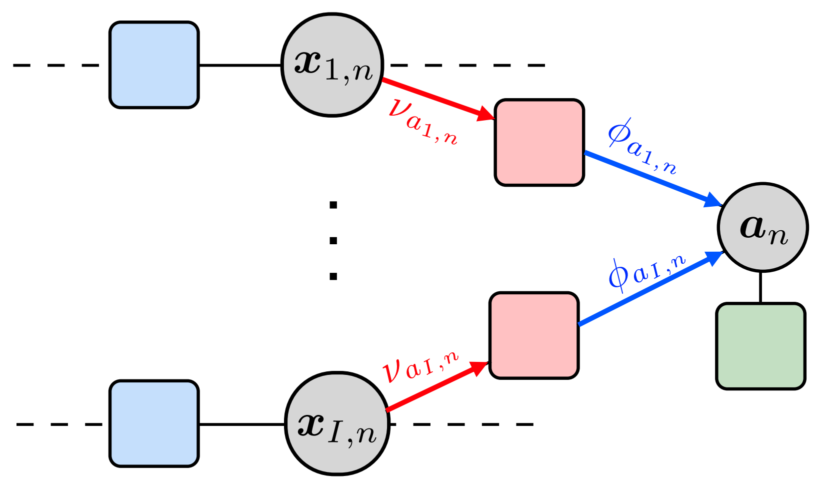
$$\phi_{a_{i,n}}(a_{i,n}) = \int g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{a_{i,n}}(\boldsymbol{x}_{i,n})\, \mathrm{d}\boldsymbol{x}_{i,n}$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Joint Probabilistic Data Association

- Closed-form measurement evaluation step (for linear-Gaussian meas. models)

$$\phi_{a_{i,n}}(a_{i,n} = 0) = (1 - p_{\mathrm{d}}) \qquad \nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) = f_{\mathrm{g}}(\boldsymbol{x}_{i,n}; \boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}}, \boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}})$$

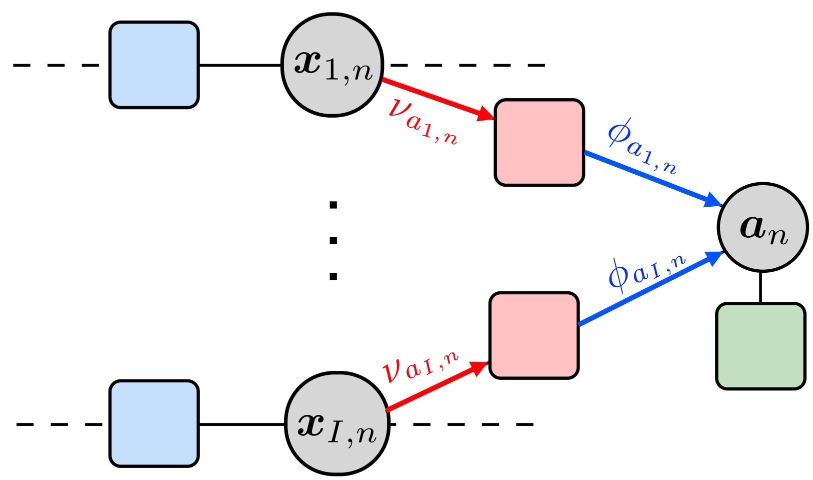
$$\phi_{a_{i,n}}(a_{i,n} = m) = \frac{p_{\mathrm{d}} f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{H}_{i,n}\boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}}, \boldsymbol{H}_{i,n}\boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}}\boldsymbol{H}^{\mathrm{T}}_{i,n} + \boldsymbol{\Sigma}_{\boldsymbol{v}_{i,n}})}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})}, \; m \in \{1, \ldots, M_n\}$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Joint Probabilistic Data Association

- Particle-based measurement evaluation step

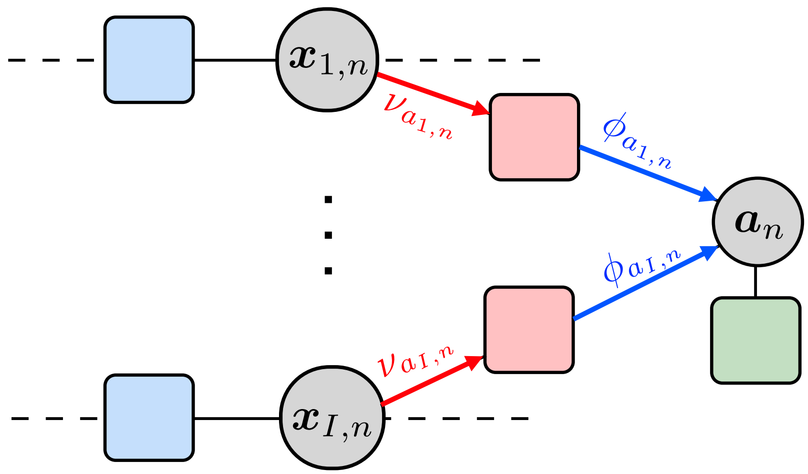
$$\phi_{a_{i,n}}(a_{i,n}) \approx \frac{1}{J}\sum_{j=1}^{J} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}^{(j)}, a_{i,n}) \qquad \nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) \simeq \left\{\left(\boldsymbol{x}_{i,n}^{(j)}\right)\right\}_{j=1}^{J}$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Joint Probabilistic Data Association

- Recall data association step:

$$\nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n) = \varphi(\boldsymbol{a}_n) \prod_{\substack{i=1\\ i=i'}} \phi_{a_{i',n}}(a_{i',n})$$

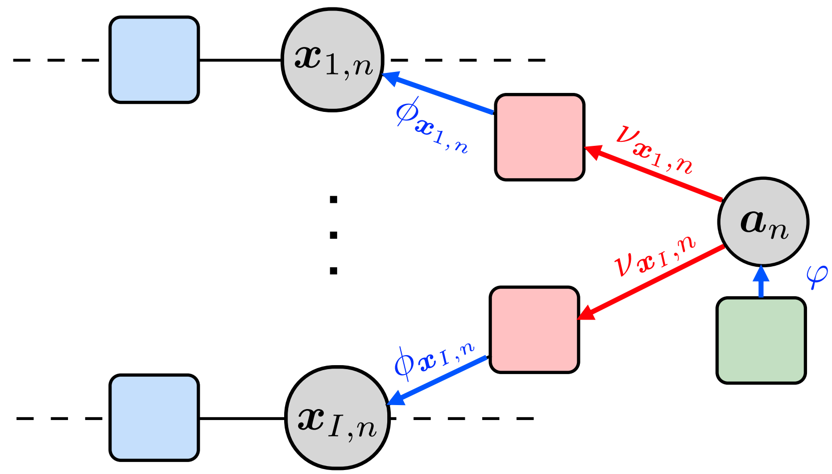
$$\phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n}) = \sum_{\boldsymbol{a}_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



© Florian Meyer, 2020

Joint Probabilistic Data Association

- Data association:

$$\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n}) = \sum_{a_{1,n}=0}^{M_n} \cdots \sum_{a_{i-1,n}=0}^{M_n} \sum_{a_{i+1,n}=0}^{M_n} \cdots \sum_{a_{I,n}=0}^{M_n} \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)$$

$$= \sum_{a_{1,n}=0}^{M_n} \cdots \sum_{a_{i-1,n}=0}^{M_n} \sum_{a_{i+1,n}=0}^{M_n} \cdots \sum_{a_{I,n}=0}^{M_n} \varphi(\boldsymbol{a}_n) \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi_{a_{i',n}}(a_{i',n})$$

$$\varphi(\boldsymbol{a}_n) \triangleq \begin{cases} 0, & \exists i, j \in \{1, 2, \ldots, I\} \text{ such that } i \neq j \text{ and } a_{i,n} = a_{j,n} \neq 0 \\ 1, & \text{otherwise} \end{cases}$$

- Computational complexity of calculating $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$ scales as $\mathcal{O}\big((M_n + 1)^I\big)$ and is thus only feasible for small I

  → need scalable methods for approximate calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

Data Association Representations

- For simplicity we consider a single time step and drop n in the notation
- Recall description of object-measurement associations by object-oriented association vectors $\boldsymbol{a} = [a_1, a_2, \ldots, a_I]^{\mathrm{T}}$

$$a_i \triangleq \begin{cases} m \in \{1, 2, \ldots, M\}, & \text{if object } i \text{ generated measurement } m \\ 0 & \text{if object } i \text{ did not generate a measurement} \end{cases}$$

- Alternative description of object-measurement associations by measurement-oriented association vectors $\boldsymbol{b} = [b_1, b_2, \ldots, b_M]^{\mathrm{T}}$ with entries

$$b_m \triangleq \begin{cases} i \in \{1, 2, \ldots, I\}, & \text{if measurement } m \text{ is generated by object } i \\ 0 & \text{if measurement } m \text{ was not generated by an object} \end{cases}$$

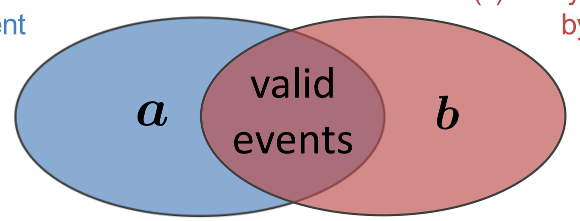
- Recall data association assumptions: An (i) object can generate at most one measurement and a (ii) measurement can be generated by at most one object

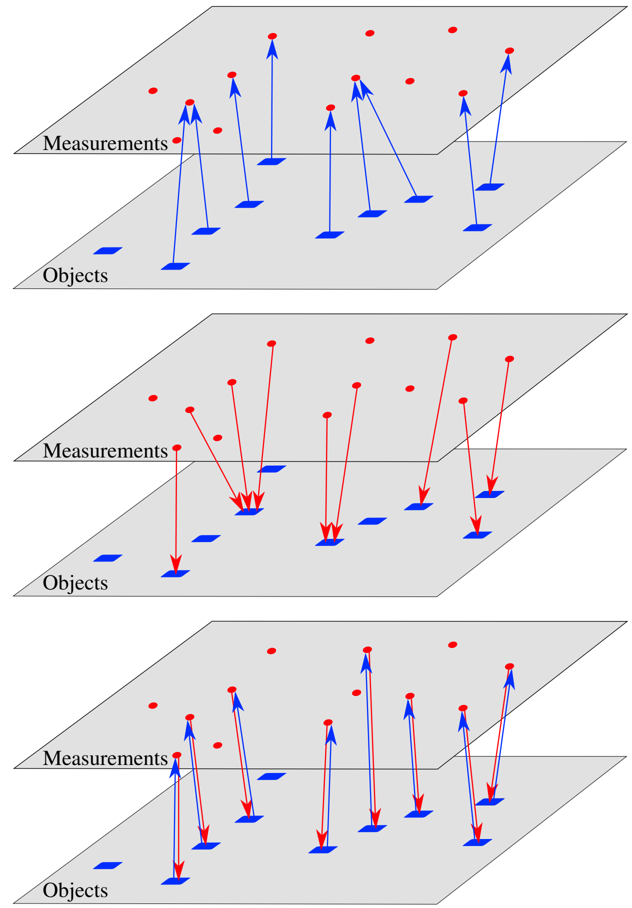
Data Association Representations

- Events represented by object-oriented vector $\boldsymbol{a} = [a_1, a_2, \ldots, a_I]^{\mathrm{T}}$ satisfy property (i)
- Events represented by measurement-oriented vector $\boldsymbol{b} = [b_1, b_2, \ldots, b_M]^{\mathrm{T}}$ satisfy property (ii)
- Events represented by object-oriented $\boldsymbol{a}$ and measurement-oriented $\boldsymbol{b}$ satisfy (i) and (ii)

(i) every object generates at most one measurement

(ii) every measurement is generated by at most one object



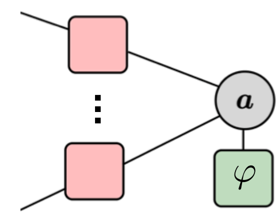


``Stretching'' the Graph

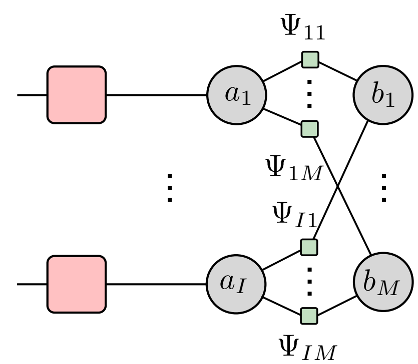
- We use a hybrid description of data association uncertainty to replace $\varphi(\boldsymbol{a})$ by

$$\psi(\boldsymbol{a},\boldsymbol{b}) \propto \prod_{i=1}^{I}\prod_{m=1}^{M} \Psi_{im}(a_i, b_m)$$

$$\Psi_{im}(a_i, b_m) \triangleq \begin{cases} 0, & a_i = m,\ b_m \neq i \\ & \text{ or } b_m = i,\ a_i \neq m \\ 1, & \text{otherwise.} \end{cases}$$



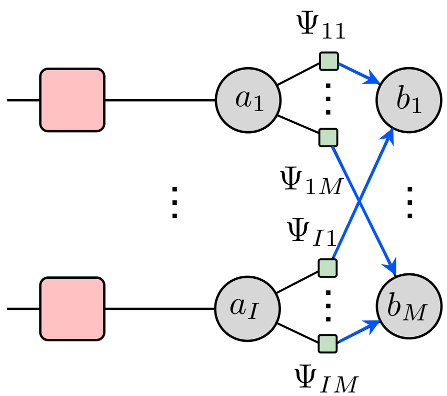
- Properties of $\psi(\boldsymbol{a},\boldsymbol{b})$:
  - is non-zero only if $\boldsymbol{a}$ and $\boldsymbol{b}$ describe the same event
  - checks consistency by low-dimensional factors $\Psi_{km}(a_k, b_m)$
  - does not alter marginal distributions since there is a deterministic one-to-one mapping from $\boldsymbol{a}$ to $\boldsymbol{b}$ and $\varphi(\boldsymbol{a}) = \sum_{\boldsymbol{b}} \psi(\boldsymbol{a},\boldsymbol{b})$



Loopy SPA for Joint Probabilistic Data Association

- Stretching the graph enables calculation of approximate $\tilde{\kappa}_{\boldsymbol{x}_{i,n}}(a_i)$ by means of the loopy SPA

- At message passing iteration $\ell \in \{1, \ldots, L\}$ we calculate the following SPA messages in parallel



$$\phi^{[\ell]}_{\Psi_{i,m}\to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i, b_m) \prod_{\substack{i'=1\\ i'\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm}\to b_m}(b_m)$$

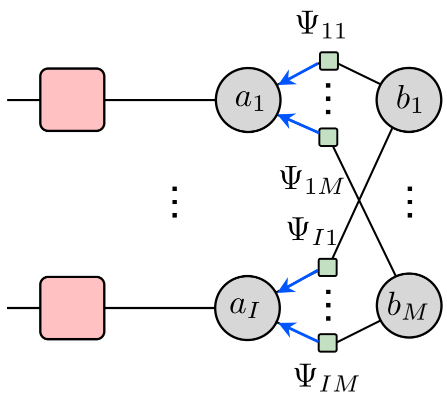
$$\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_{i,n}}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1\\ m'\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'}\to a_i}(a_i)$$

- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im}\to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i, b_m)$

Loopy SPA for Joint Probabilistic Data Association

- Stretching the graph enables calculation of approximate $\tilde{\kappa}_{\boldsymbol{x}_{i,n}}(a_i)$ by means of the loopy SPA

- At message passing iteration $\ell \in \{1, \ldots, L\}$ we calculate the following SPA messages in parallel



$$\phi^{[\ell]}_{\Psi_{i,m} \to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i, b_m) \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm} \to b_m}(b_m)$$

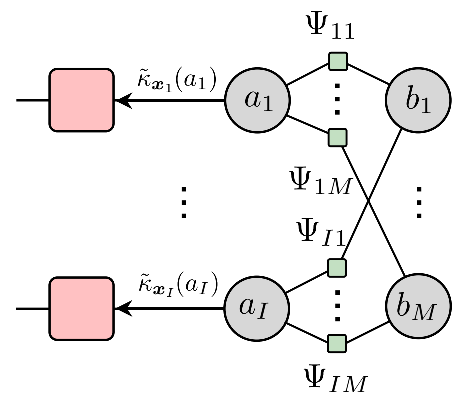
$$\phi^{[\ell]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_{i,n}}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'} \to a_i}(a_i)$$

- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i, b_m)$

Loopy SPA for Joint Probabilistic Data Association

- Stretching the graph enables calculation of approximate $\tilde{\kappa}_{\boldsymbol{x}_{i,n}}(a_i)$ by means of the loopy SPA

- At message passing iteration $\ell \in \{1, \ldots, L\}$ we calculate the following SPA messages in parallel



$$\phi^{[\ell]}_{\Psi_{i,m} \to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i, b_m) \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm} \to b_m}(b_m)$$

$$\phi^{[\ell]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_{i,n}}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'} \to a_i}(a_i)$$

- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i, b_m)$

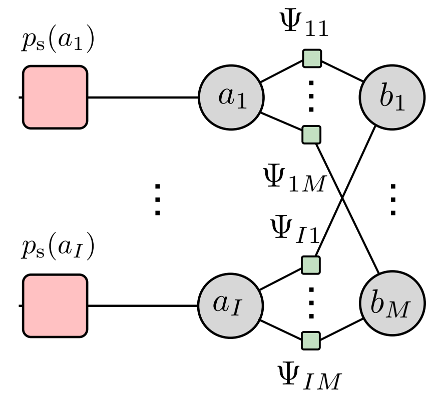
- Result after $\ell = L$ iterations: $\tilde{\kappa}_{\boldsymbol{x}_i}(a_i) = \prod_{m=1}^{M} \phi^{[L]}_{\Psi_{im} \to a_i}(a_i)$

General Probabilistic Assignment Algorithm

- Calculate joint assignment probabilities $p_{\mathrm{j}}(a_i)$ from single assignment probabilities $p_{\mathrm{s}}(a_i)$
- At message passing iteration $\ell \in \{1, \ldots, L\}$ we calculate the following SPA messages in parallel

$$\phi^{[\ell]}_{\Psi_{i,m} \to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i, b_m) \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm} \to b_m}(b_m)$$

$$\phi^{[\ell]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} p_{\mathrm{s}}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'} \to a_i}(a_i)$$



- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} p_{\mathrm{s}}(a_i)\, \Psi_{im}(a_i, b_m)$
- Result after $\ell = L$ iterations: $p_{\mathrm{j}}(a_i) = p_{\mathrm{s}}(a_i) \prod_{m=1}^{M} \phi^{[L]}_{\Psi_{im} \to a_i}(a_i)$
- Calculate MAP assignments $\hat{a}_i = \operatorname{argmax} p_{\mathrm{j}}(a_i), \qquad i \in \{1, \ldots, I\}$

Summary

- On factor graphs with cycles, the SPA
  - has to be performed iteratively
  - relies on a predefined message passing order
  - only provides approximate marginal posteriors
- The multiobject tracking problem can be represented by a factor graphs with cycles and solved by means of the SPA (messages are only send forward in time)
- The complexity of joint probabilistic data association for multiobject tracking scales exponentially with the number of objects I
- By making modifications to the graph, the scalability of joint probabilistic data association can be increased