ECE 286: Bayesian Machine Perception

Class 13: Graph-Based Multiobject Tracking III

Florian Meyer

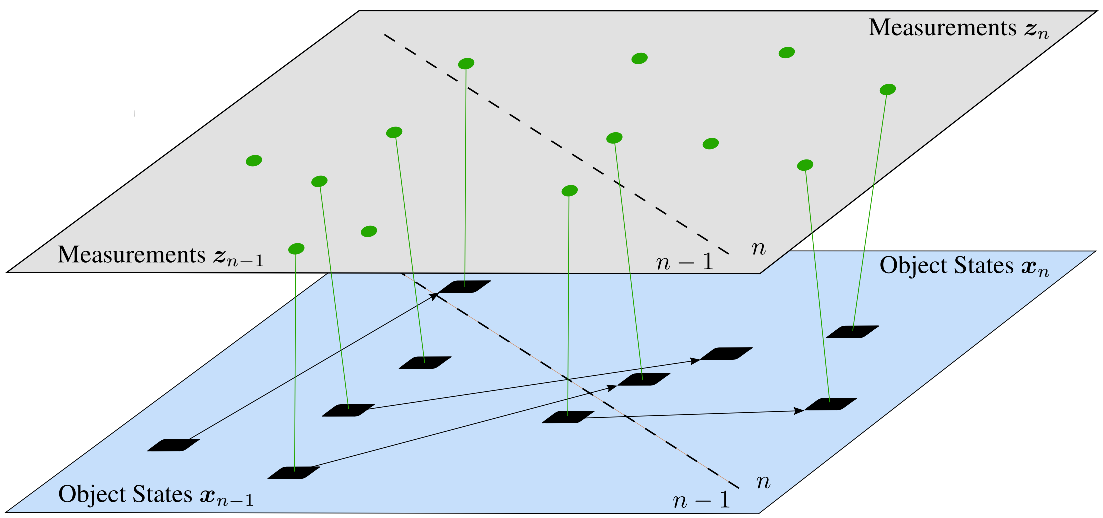
Electrical and Computer Engineering Department
University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

The Multiobject Tracking Problem

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

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



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

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

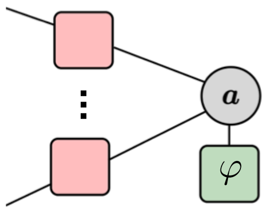
→ need scalable methods for approximate calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

``Stretching'' the Graph

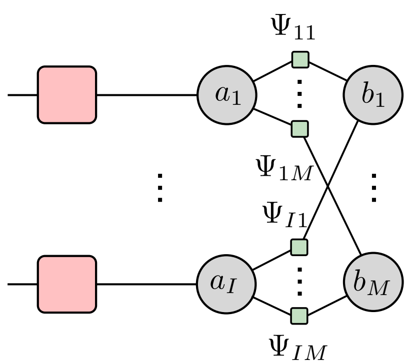
- We use a hybrid description of data association uncertainty to replace $\varphi(\boldsymbol{a})$ by

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

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



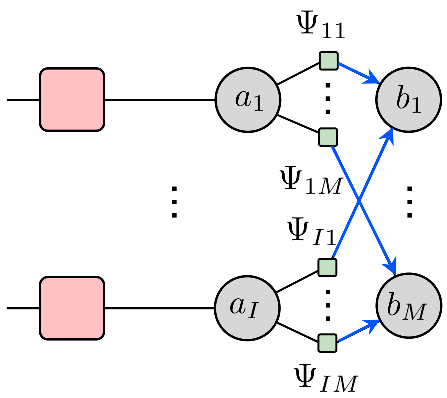
- Properties of $\psi(\boldsymbol{a}, \boldsymbol{b})$:
  – is non-zero only if $\boldsymbol{a}$ and $\boldsymbol{b}$ describe the same event
  – checks consistency by low-dimensional factors $\Psi_{km}(a_k, b_m)$
  – does not alter marginal distributions since there is a deterministic one-to-one mapping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



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

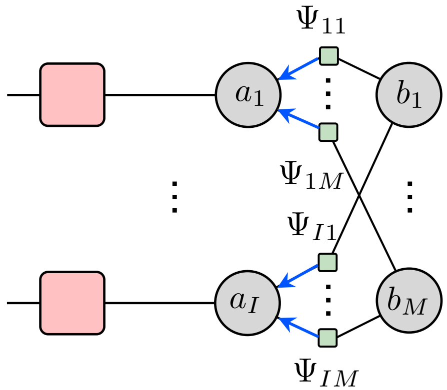
$$\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_{i,n}}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1 \\ m'\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'}\to a_i}(a_i)$$

- 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 $\ell \in \{1, \dots, L\}$ we calculate the following SPA messages in parallel



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

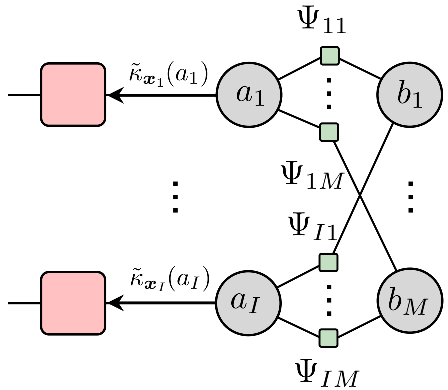
$$\phi^{[\ell]}_{\Psi_{im}\rightarrow b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1 \\ m'\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'}\rightarrow a_i}(a_i)$$

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

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 $(i \in \{1,\ldots,I\}, m \in \{1,\ldots,M\})$



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

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

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

The Sum-Product Algorithm for Data Association (SPADA)

- The complexity of the probabilistic assignment algorithm can be reduced further by performing the following steps

  1. Since the constraint $\Psi_{im}(a_i, b_m)$ is binary, messages can be represented by only two different values

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

$$\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m) = \begin{cases} \phi_{a_i}(m) \prod_{\substack{m'=1 \\ m'\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'}\to a_i}(m), & \text{for } b_m = i \\ \sum_{\substack{a_i=0 \\ a_i\neq m}}^{M} \phi_{a_i}(a_i) \prod_{\substack{m''=1 \\ m''\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im''}\to a_i}(a_i), & \text{for } b_m \neq i \end{cases}$$

The Sum-Product Algorithm for Data Association (SPADA)

- The complexity of the probabilistic assignment algorithm can be reduced further by performing the following steps

2. Since messages can be multiplied by an arbitrary constant, we divide each message by one of its values

$$\phi^{[\ell]}_{\Psi_{im}\to a_i}(a_i) \propto \begin{cases} \dfrac{\prod_{\substack{i'=1\\ i'\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm}\to b_m}(i)}{\sum_{\substack{b_m=0\\ b_m\neq i}}^{I} \prod_{\substack{i''=1\\ i''\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i''m}\to b_m}(b_m)}, & \text{for } a_i = m \\ 1, & \text{for } a_i \neq m \end{cases}$$

$$\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m) \propto \begin{cases} \dfrac{\phi_{a_i}(m) \prod_{\substack{m'=1\\ m'\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'}\to a_i}(m)}{\sum_{\substack{a_i=0\\ a_i\neq m}}^{M} \phi_{a_i}(a_i) \prod_{\substack{m''=1\\ m''\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im''}\to a_i}(a_i)}, & \text{for } b_m = i \\ 1, & \text{for } b_m \neq i \end{cases}$$

The Sum-Product Algorithm for Data Association (SPADA)

- The complexity of the probabilistic assignment algorithm can be reduced further by performing the following steps

3. Messages can now be replaced by there normalized counterpart

$$\phi^{[\ell]}_{\Psi_{im}\to a_i}(a_i) \propto \begin{cases} \frac{1}{1+\sum_{\substack{i'=1\\ i'\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm}\to b_m}(i')}, & \text{for } a_i = m \\ 1, & \text{for } a_i \neq m \end{cases}$$

$$\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m) \propto \begin{cases} \frac{\phi_{a_i}(m)}{\phi_{a_i}(0)+\sum_{\substack{m'=1\\ m'\neq m}}^{M} \phi_{a_i}(m')\,\phi^{[\ell]}_{\Psi_{im'}\to a_i}(m')}, & \text{for } b_m = i \\ 1, & \text{for } b_m \neq i \end{cases}$$

The Sum-Product Algorithm for Data Association (SPADA)

- The complexity of the probabilistic assignment algorithm can be reduced further by performing the following steps

4. Each message can be represented by a single value $\big(i \in \{1,\dots,I\}, m \in \{1,\dots,M\}\big)$

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

Initialization: $$\phi^{[0]}_{\Psi_{im}\to b_m} = \frac{\phi_{a_i}(m)}{\phi_{a_i}(0)+\sum_{\substack{m'=1\\ m'\neq m}}^{M} \phi_{a_i}(m')}$$

Result after L iterations:

$$\tilde{\kappa}_{\boldsymbol{x}_i}(a_i) = \begin{cases} \phi^{[L]}_{\Psi_{im}\to a_i}(m), & \text{for } a_i = m \in \{1,\dots,M\} \\ 1, & \text{for } a_i = 0 \end{cases}$$

Properties

- The complexity of the probabilistic assignment algorithm is essentially determined by that of calculating the sums $\sum_{i=1}^{I} \phi^{[\ell]}_{\Psi_{im}\to b_m}(i)$ and $\phi^{[\ell]}_{\Psi_{im}\to b_m} = \sum_{m=1}^{M} \phi_{a_i}(m)\phi^{[\ell]}_{\Psi_{im}\to a_i}(m)$, which scales as $\mathcal{O}(IM)$

- It can be shown that the loopy SPA algorithms for joint probabilistic data association
  - solves a convex optimization problem
  - is guaranteed to converge
  - provides the correct MAP solution

M. Bayati, D. Shah, and M. Sharma, "Max-product for maximum weight matching: Convergence, correctness, and LP duality," *IEEE Trans. Inf. Theory*, no. 3, pp. 1241–1251, Mar. 2008.

J. L. Williams and R. A. Lau, "Multiple scan data association by convex variational inference," *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 2112–2127, Apr. 2018.

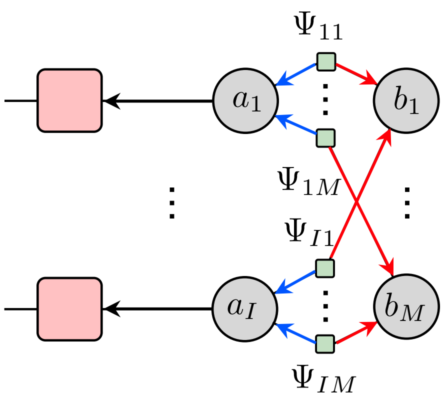
F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, Feb. 2018.

Joint Probabilistic Data Association

- The sum-product message passing rules are applied to the stretched factor graph we obtain $\phi^{[\ell]}_{\Psi_{i,m}\to a_i}(a_i)$ and $\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m)$ for the ℓth iteration

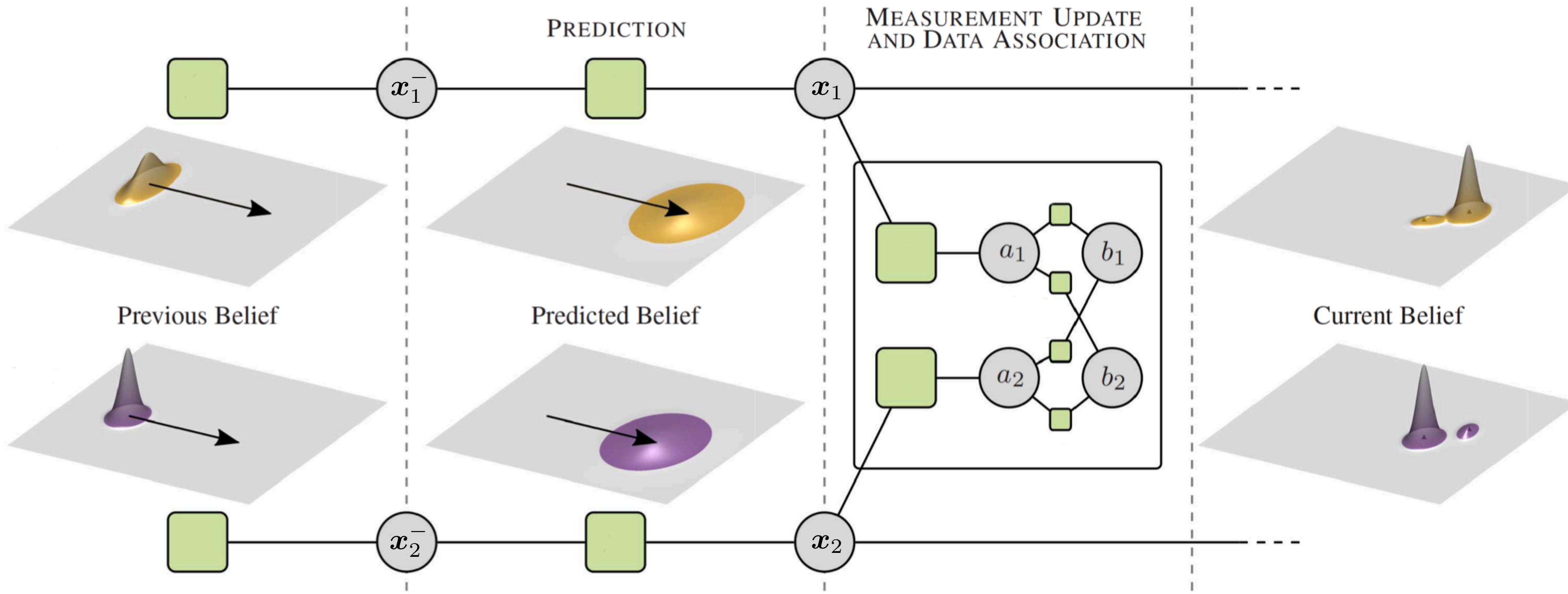
- Due to the binary consistency constrains, $\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m)$ takes only two values (one for $b_m = i$ and one for $b_m \neq i$); similarly $\phi^{[\ell]}_{\Psi_{i,m}\to a_i}(a_i)$ takes one value for $a_i = m$ and one value for $a_i \neq m$

- Can be implemented by performing pointwise operations on $I \times M$ matrices



- All $\tilde{\kappa}_{\boldsymbol{x}_i}(a_i)$ needed for multiobject tracking can be obtained with a complexity that only scales as $\mathcal{O}(IM)$

Multiobject Tracking Example



Hard Measurement Validation

- To further reduce computational complexity of multiobject tacking, measurements that with a high probability have not been generate by an object, can be removed in a suboptimum preprocessing step
- For each object a multidimensional gate is set up and only measurements that fall within the gate are considered association candidates
- Joint probabilistic data association has only to be performed for objects that share association candidates; thus its complexity is $\mathcal{O}(I'M')$ with $I' \leqslant I$ and $M' \leqslant M$

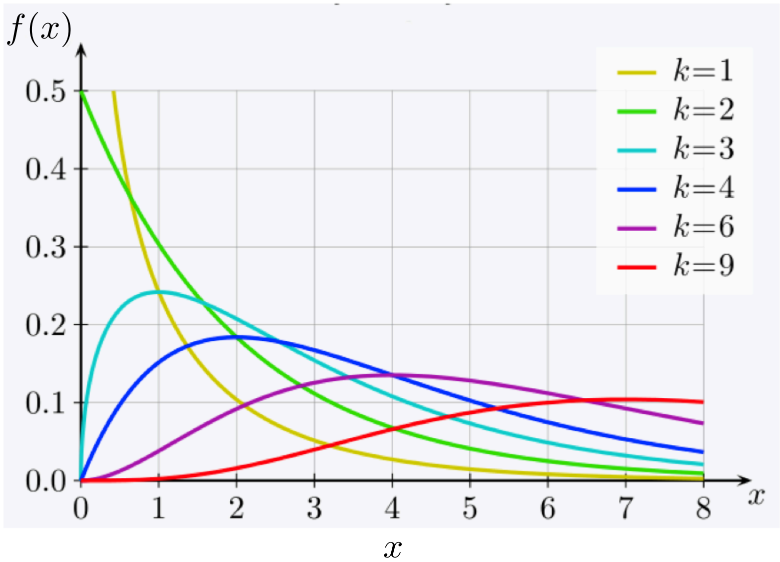
The Chi-Square Distribution

- The chi-square distribution with k degrees of freedom is the distribution of a sum of the squares of k independent normal random with unit variance

$$f(x) = \begin{cases} \frac{x^{\frac{k}{2}-1} \exp(-\frac{x}{2})}{2^{\frac{k}{2}} \Gamma(\frac{k}{2})}, & x > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\Gamma(n) = (n-1)! \qquad \text{for } n \in \mathbb{N}_0$$

Mean: k
Variance: $2k$



Hard Measurement Validation

- **Assumption:** The measurement that is originated by object i at time n is distributed according to

$$f(\boldsymbol{z}_{m,n}|\boldsymbol{z}_{1:n-1}) = f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{H}_n \boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}}, \boldsymbol{H}_n \boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}} \boldsymbol{H}_n^{\mathrm{T}} + \boldsymbol{\Sigma}_{\boldsymbol{v}_n})$$
$$= f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{\mu}^{-}_{\boldsymbol{z}_{i,n}}, \boldsymbol{\Sigma}^{-}_{\boldsymbol{z}_{i,n}})$$

- The true measurement will be in the following set

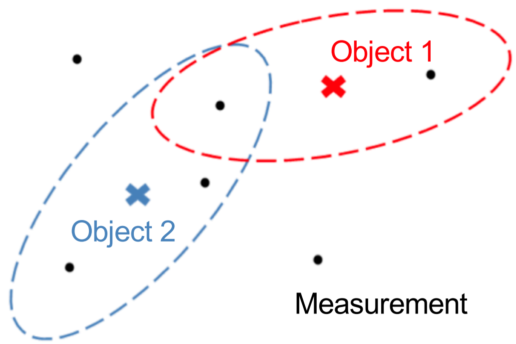
$$\mathcal{V}_{i,n}(\gamma) = \left\{\boldsymbol{z}_{m,n} \middle| (\boldsymbol{z}_{m,n} - \boldsymbol{\mu}_{\boldsymbol{z}_{i,n}})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{z}_{i,n}} (\boldsymbol{z}_{m,n} - \boldsymbol{\mu}_{\boldsymbol{z}_{i,n}}) \leqslant \gamma \right\}$$

with probability determined by the threshold γ

- The region that contains validated measurements is an ellipsoid with semiaxes given by the square roots of the eigenvalues of $\gamma \boldsymbol{\Sigma}_{\boldsymbol{z}_{i,n}}$

Hard Measurement Validation

- The quadratic form that defines the validation region is chi-square distributed with the number of degrees of freedom equal to the dimension of a measurement d
- Thus, the probability p_{g} that a measurement lies in the validation region or ``gate'' can be obtain from the cumulative distribution function of the chi-square distribution, i.e., $p_{\mathrm{g}} = \mathtt{chi2cdf}(\gamma, d)$
- Hard measurement validation trades of computational complexity and sensor performance since p_{d} is reduced to $p'_{\mathrm{d}} = p_{\mathrm{g}} p_{\mathrm{d}}$
- Example with two objects →



Summary

- Computational complexity of joint probabilistic data association can be reduced from $\mathcal{O}\big((M_n+1)^I\big)$ to $\mathcal{O}(IM_n)$ by performing a highly optimized loopy sum-product algorithm

- Hard measurement validation (``gating'') can further reduce computational complexity by extracting association candidates from the joint measurement vector and thus reducing the dimension of the data association problem