Robotics 2
Data Association

Giorgio Grisetti, Cyrill Stachniss,
Kai Arras, Wolfram Burgard

Slides by Kai Arras, Gian Diego Tipaldi, v.1.1, Dec 2010

Data Association

"Data association is the process of associating uncertain measurements to known tracks."

- **Problem types**
  - Track creation, maintenance, and deletion
  - Single or multiple sensors
  - Target detection
  - False alarm model and rates
  - Single or multiple targets

- **Approaches**
  - **Bayesian:** compute a full (or approx.) distribution in DA space from priors, posterior beliefs, and observations
  - **Non-Bayesian:** compute a maximum likelihood estimate from the possible set of DA solutions

Data Association

Overall procedure:

- **Make observations** (= measurements).

  Measurements can be raw data (e.g. processed radar signals) or the output of some target detector (e.g. people detector)

- **Predict the measurements** from the predicted tracks.

  This yields an area in sensor space where to expect an observation. The area is called **validation gate** and is used to narrow the search

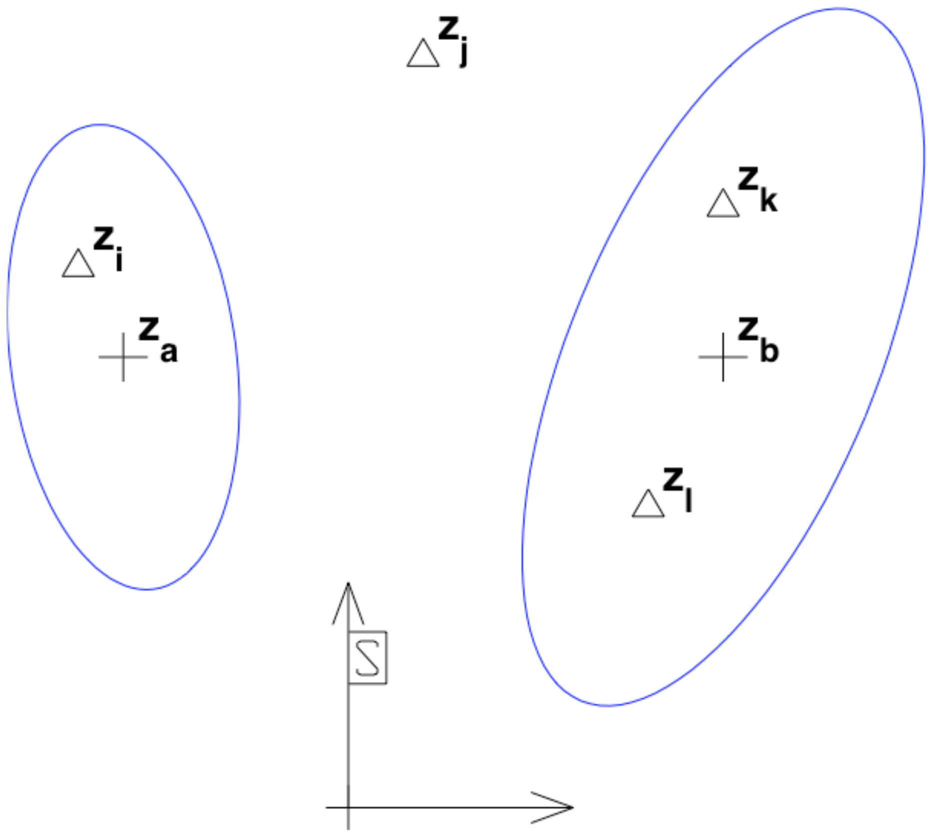
- **Check if a measurement lies in the gate**.

  If yes, then it is a valid candidate for a pairing/match

Data Association

What makes this a difficult problem

- **Multiple targets**
- **False alarms**
- **Detection uncertainty** (occlusions, sensor failures, ...)
- **Ambiguities** (several measure-ments in the gate)



Measurement Prediction

- Measurement and measurement cov. prediction
  - This is typically a frame transformation into sensor space

$$\hat{z}(k) = H(k)\hat{x}(k|k-1)$$

$$\hat{R}(k) = H(k)\hat{P}(k|k-1)H^T(k)$$

- If only the **position** of the target is observed (typical case), the measurement matrix is

$$\mathbf{z} = \left[\begin{array}{cc} x & y \end{array}\right]^T \qquad H = \left[\begin{array}{cccc} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \end{array}\right]$$

- Note: One can also observe
  - Velocity (Doppler radar)
  - Acceleration (accelerometers)

Validation Gate

- Assume that measurements are distributed according to a Gaussian, centered at the measurement prediction $\widehat{z}(k)$ with covariance $\widehat{S}(k)$

$$p(z(k)) = \mathcal{N}(z(k); \widehat{z}(k), \widehat{S}(k))$$

  This is the **measurement likelihood model**

- Let further

$$d = \sqrt{(\mathbf{x} - \mu)^{\mathbf{T}} \mathbf{C}^{-1} (\mathbf{x} - \mu)}$$

  be the **Mahalanobis distance** between $\mathbf{x}$ and μ

Validation Gate

- Then, the measurements will be in the area

$$\begin{aligned}\mathcal{V}(k,\gamma) &= \{z : (z-\hat{z})^T \hat{S}^{-1}(z-\hat{z}) \leq \gamma\} \\ &= \{z : d^2 \leq \gamma\}\end{aligned}$$

  with a probability defined by the gate threshold γ (omitting indices k)

- This area is called **validation gate**
- The threshold is obtained from the inverse χ^2 cumulative distribution at a **significance level** α
- χ^2 = "chi square"

Validation Gate

- The **shape** of the validation gate is a hyper-ellipsoid
- This follows from setting

$$c = \frac{1}{(2\pi)^{k/2}|S|^{1/2}} \exp\left(-\frac{1}{2}(z-\hat{z})^T S^{-1}(z-\hat{z}))\right)$$

  leading to

$$c' = (z-\hat{z})^T S^{-1}(z-\hat{z})$$

  which describes a **conic section** in matrix form

$$\mathbf{x}^T\mathbf{Q}\mathbf{x} = 0 \qquad \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \mathbf{x}^T = [x, y, 1] \quad \mathbf{Q} = \begin{bmatrix} A & B & D \\ B & C & E \\ D & E & F \end{bmatrix}$$

- The gate is a **iso-probability contour** obtained when intersecting a Gaussian with a hyper-plane.

Validation Gate

Why a χ^2 distribution?

- Let X_i be a set of k i.i.d. standard normally distributed random variables, $X_i \sim \mathcal{N}(x; 0, 1)$.

  Then, the variable Q

  $$Q = \sum_{i=1}^{k} X_i^2$$

  follows a χ^2 distribution with k "degrees of freedom"

- We will now show that the Mahalanobis distance is a sum of squared standard normally distributed RVs.

Validation Gate in 1D

- Assume 1D measurements and $\mu = \hat{z}(k)$, $\sigma^2 = \widehat{S}(k)$
- The Mahalanobis distance is then

$$d^2 = (z-\mu)^T(\sigma^2)^{-1}(z-\mu) = \frac{(z-\mu)^2}{\sigma^2}$$

- By changing variables, $y = (z-\mu)/\sigma$, we have

$$y \sim \mathcal{N}(0,1)$$

- Thus, $d^2 = y^2$ and is χ^2 distributed with 1 degree of freedom

Validation Gate in ND

- Assume ND measurements and $\mu = \hat{z}(k),\, \Sigma = \hat{S}(k)$
- The Mahalanobis distance is then

$$d^2 = (z-\mu)^T \Sigma^{-1} (z-\mu)$$

- By changing variables, $y = C^{-1}(z-\mu),\, \Sigma = CC^T$ we have $y \sim \mathcal{N}(0, I)$ and therefore

$$d^2 = y^T I^{-1} y \quad \Rightarrow \quad d^2 = \sum_{i=1}^{k} y_i^2$$

  which is χ^2 distributed with k degrees of freedom.
- C is obtained from a Cholesky decomposition

Validation Gate

Where does the threshold γ come from?

- γ, often denoted $\chi^2_{k,\alpha}$, is taken from the inverse χ^2 cumulative distribution at a level α and k d.o.f.s
- The values are typically given in tables, e.g. in most statistics books (or by the Matlab function `chi2inv`)
- Given the level α, we can now understand the interpretation of the validation gate:

> The validation gate is a **region of acceptance** such that $100(1-\alpha)\%$ of **true measurements** are **rejected**

- Typical values for α are 0.95 or 0.99

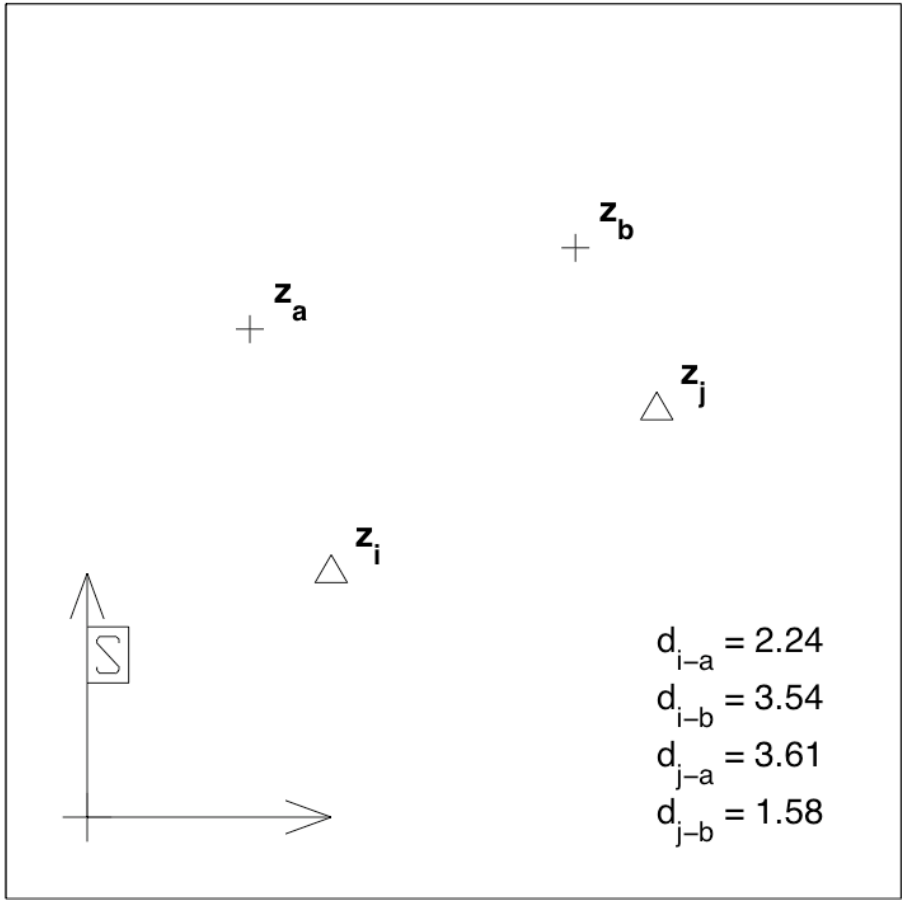
Validation Gate

Euclidian distance

Takes into account:

- ✓ Position
- ✗ Uncertainty
- ✗ Correlations

→ It seems that i-a and j-b belong together



△ Observations   + Predictions

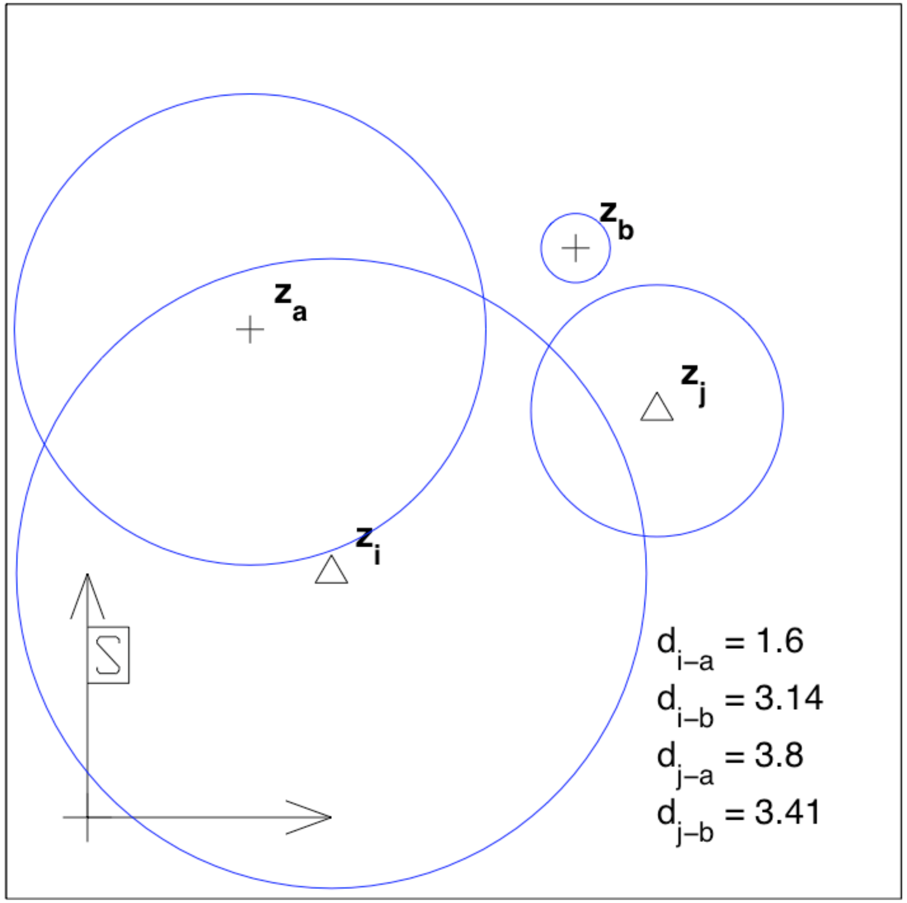
Validation Gate

Mahalanobis distance with **diagonal** covariance matrices

Takes into account:

- ✓ Position
- ✓ Uncertainty
- ✗ Correlations

→ Now, i-b is "closer" than j-b



△ Observations + Predictions

Validation Gate

Mahalanobis distance

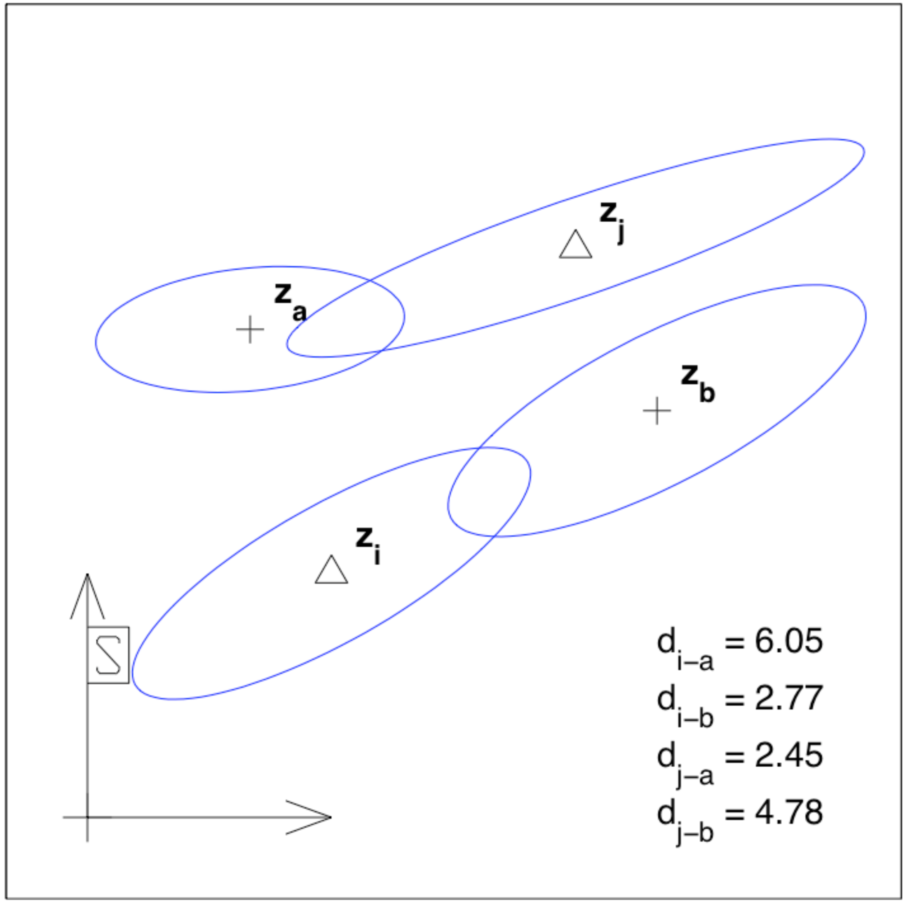
Takes into account:

✓ Position

✓ Uncertainty

✓ Correlations

→ It's actually i-b and j-a that belong together!



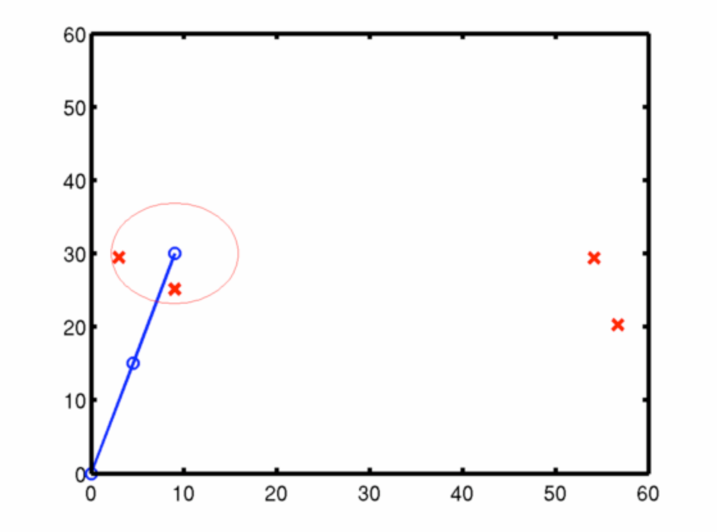
△ Observations + Predictions

False Alarms

- False alarms are **false positives**
- They can come from sensor imperfections or detector failures
- They raise the two questions:

  What is actually inside my **validation gate**?
  - The real measurement or
  - a false alarm?

  How to **model false alarms**?
  - Uniform over sensor space
  - Independent across time

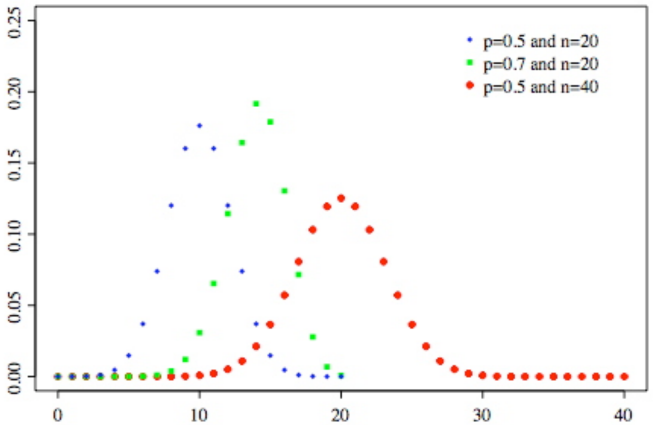


False Alarm Model

- Assume (temporarily) that the sensor field of view V is discretized into N discrete cells, $c_i, \ i = 1, \ldots, N$
- In each cell, false alarms occur with probability P_F
- Assume independence across cells
- The occurrence of false alarms is a Bernoulli process (flipping an unfair coin) with probability $p = P_F$
- Then, the number of false alarms m_F follows a **Binomial distribution**

$$P(K = m_F) = \binom{N}{m_F} p^{m_F} (1-p)^{N-m_F}$$

with expected value Np



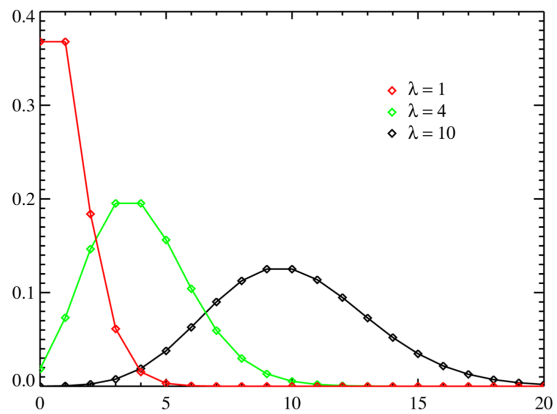
False Alarm Model

- Let the spatial density λ be the number of false alarms over space

$$\lambda = \frac{Np}{V}$$ [occurrences per m²]

- Let now $N \rightarrow \infty$, that is, we reduce the cell size until the continuous case. Then the Binomial becomes a Poisson distribution with

$$\mu_F(m_F) = e^{-\lambda V} \frac{(\lambda V)^{m_F}}{m_F!}$$



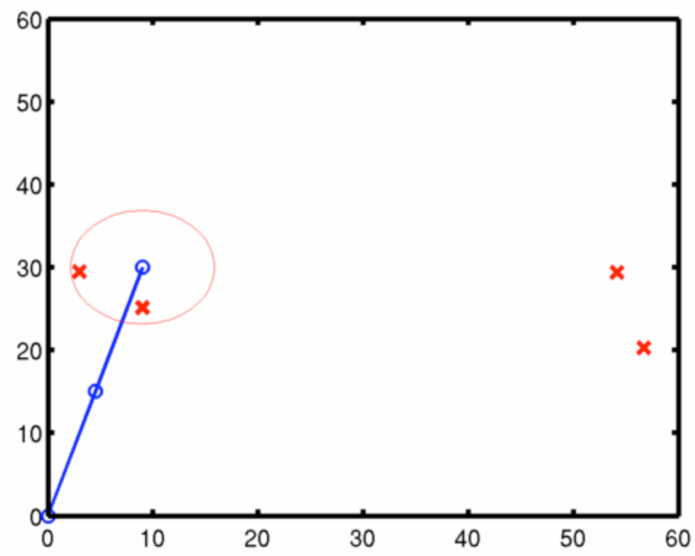
- The **measurement likelihood** of false alarms is assumed to be uniform,

$$p(z|z \text{ is a false alarm}) = \frac{1}{V}$$

Single Target Data Association

Assumptions

- A **single** target to track
- Track already initialized
- Detection probability < 1
- False alarm probability > 0



Data Association approaches

Non-Bayesian: no prior association probabilites

- Nearest neighbor Standard filter (NNSF)
- Track splitting filter

Bayesian: prior association probabilites

- Probabilistic Data Association Filter (PDAF)

Single Target DA: NNSF

Nearest Neighbor Standard Filter (NNSF)

1. Compute Mahalanobis distance to all measurements
2. Accept the **closest measurement**
3. Update the track as if it were the correct one

Problem: with some probability the selected measurement is not the correct one. This can lead to overconfident covariances, filter divergence and track loss. Covariances will collapse in any case.

Conservative NNSF variant:
Do not associate in case of ambiguities

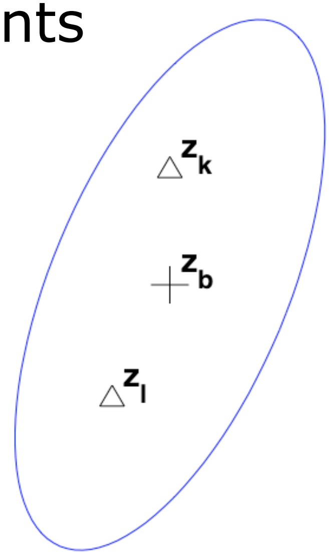
Other variant: **Strongest Neighbor Standard filter:**
Used, e.g., with sonar sensors

Single Target DA: PDAF

Probabilistic Data Association filter (PDAF)

- Integrates **all** measurements in the validation gate
  - Conditioning the update on the association events

$$\theta_i(k) = \begin{cases} z_i(k) \text{ is the correct measurement} & i = 1, \dots, m(k) \\ \text{no correct measurement is present} & i = 0 \end{cases}$$



- $\beta_i \triangleq P(\theta_i | Z^k)$ is the **association probability** which for a Poisson false alarm model is

$$\begin{aligned} \beta_i(k) &= \begin{cases} \frac{e_i}{b+\sum_{j=1}^{\mu_F}} & i = 1, \dots, m(k) \\ \frac{b}{b+\sum_{j=1}^{\mu_F}} & i = 0 \end{cases} \\ e_i &= \mu_F(m(k)-1) \cdot P_D P_G \cdot P_G^{-1} \mathcal{N}(\nu_i(k); 0, \hat{S}(k)) \\ b &= \mu_F(m(k))(1 - P_D P_G) \end{aligned}$$

(Derivation skipped, Notation: m(k) = N)

Single Target DA: PDAF

State update

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\nu(k)$$

- With the combined innovation

$$\nu(k) = \sum_{i=1}^{N} \beta_i(k)\nu_i(k)$$

summed over all *N* association events $\theta_i(k)$.

The events $\theta_i(k)$ are assumed to be **exhaustive** (their probabilities sums up to one) and **mutually exclusive** (they cannot occur at the same time)

Single Target DA: PDAF

Covariance update

$$P(k|k) = \beta_0(k)P(k|k-1) + (1-\beta_0(k))P(k|k) + \tilde{P}(k)$$

- With prob. $\beta_0(k)$ none of the measurements is correct, the predicted cov. appears with this weighting ("no update")
- With prob. $(1-\beta_0(k))$ the correct measurement is available and the posterior cov. appears with this weighting
- Since it is unknown which if the measurements is correct, the term

$$\tilde{P}(k) = K(k)\left[\sum_{i=1}^{N} \beta_i(k)\nu_i(k)\nu_i(k)^T - \nu(k)\nu(k)^T\right] K(k)^T$$

(*spread of innovations*) increases the covariance to account for the data association uncertainty

(Derivation skipped)

Single Target DA: Summary

- The NNSF takes a **hard** association decision
  - This hard decision is sometimes correct and sometimes wrong
- The PDAF relies on a **soft** decision since it averages over all the association possibilities
  - This soft decision is never totally correct but never totally wrong
- This is why **the PDAF is a suboptimal** strategy
  - To be precise: the PDAF is suboptimal since it approximates the conditional pdf of the target's state at every stage as a Gaussian with moments matched to the mixture $$p(x \mid Z) = \sum_{A_i \in \mathcal{A}} p(x \mid A_i, Z)\, \beta_i$$

Single Target DA: Summary

- **Nearest Neighbor Standard filter** (NNSF)
  - Simple to implement
  - Can integrate wrong measurements (false alarms), and thus, produce overconfident estimates
  - Good if prediction and measurement models are accurate and/or track cycles are fast with respect to the target dynamics
  - In other words, good if DA ambiguity is low.

- **Probabilistic Data Association filter** (PDAF)
  - A bit more involved to implement
  - The PDAF can significantly improve tracking in regions of high clutter densities where the NNSF becomes unreliable because of its high probability of track loss
  - Suboptimal method

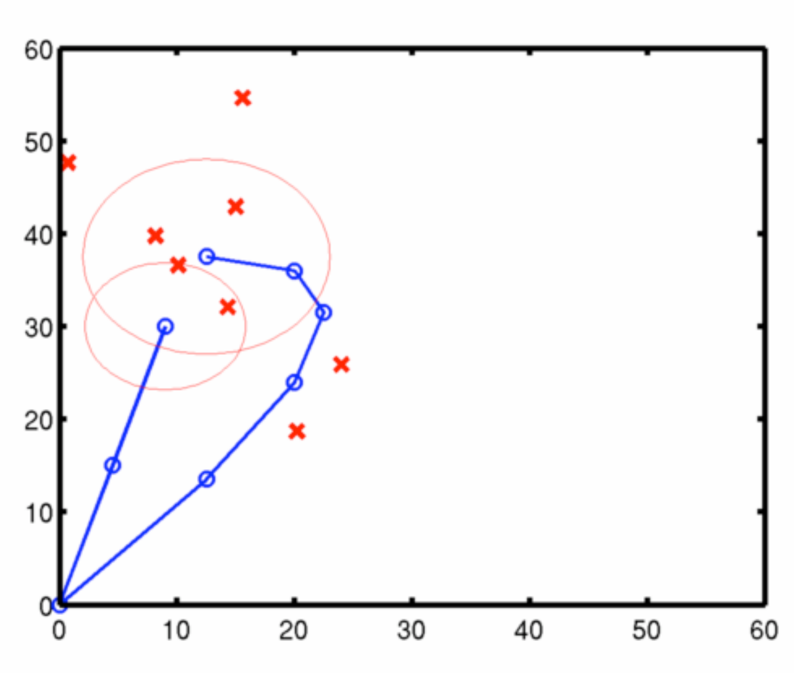
Multi-Target Data Association

Assumptions

- **Multiple** targets to track
- Tracks already initialized
- Detection probability $P_D < 1$
- False alarm probability $P_F > 0$

Data Association approaches

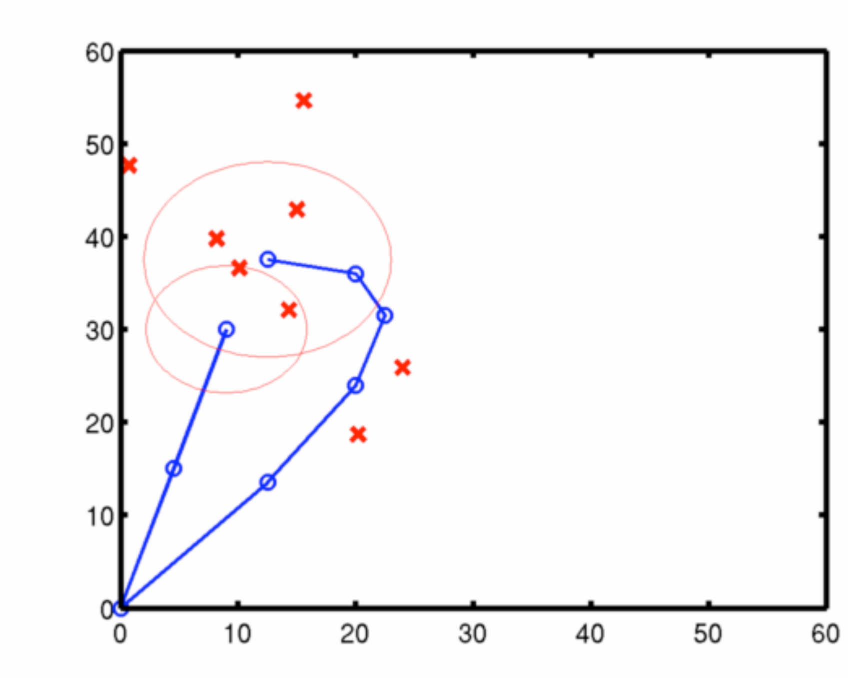
- *Non Bayesian:* no priors
  - NNSF
  - Global NNSF
- *Bayesian:* prior association probabilites
  - JPDAF
  - MHT
  - MCMC



Multi-Target DA: NNSF

Nearest Neighbor Standard Filter (NNSF)

1. Build the assignment matrix $A = \left[d_{ij}^2 \right]$ with
   $d_{ij}^2 = \nu_{ij}(k)^T S_j^{-1}(k) \nu_{ij}(k)$



- Rectangular

$$A = \begin{bmatrix} d_{11}^2 & d_{12}^2 & d_{13}^2 & d_{14}^2 & d_{15}^2 & d_{16}^2 & d_{17}^2 & d_{18}^2 \\ d_{21}^2 & d_{22}^2 & d_{23}^2 & d_{24}^2 & d_{25}^2 & d_{26}^2 & d_{27}^2 & d_{28}^2 \end{bmatrix}$$

- Square

$$A = \begin{bmatrix} d_{11}^2 & d_{12}^2 & d_{13}^2 & d_{14}^2 & d_{15}^2 & d_{16}^2 & d_{17}^2 & d_{18}^2 \\ d_{21}^2 & d_{22}^2 & d_{23}^2 & d_{24}^2 & d_{25}^2 & d_{26}^2 & d_{27}^2 & d_{28}^2 \\ p_F & p_F & p_F & p_F & p_F & p_F & p_F & p_F \\ p_F & p_F & p_F & p_F & p_F & p_F & p_F & p_F \\ p_F & p_F & p_F & p_F & p_F & p_F & p_F & p_F \\ p_F & p_F & p_F & p_F & p_F & p_F & p_F & p_F \\ p_F & p_F & p_F & p_F & p_F & p_F & p_F & p_F \\ p_F & p_F & p_F & p_F & p_F & p_F & p_F & p_F \end{bmatrix}$$

Multi-Target DA: NNSF

Nearest Neighbor Standard Filter (NNSF)

1. Build the assignment matrix $A = \left[d_{ij}^2 \right]$ with
   $d_{ij}^2 = \nu_{ij}(k)^T S_j^{-1}(k) \nu_{ij}(k)$
2. Iterate
   - Find the minimum cost assignment in A
   - Remove the row and column of that assignment
3. Check if assignment is in the validation regions
   - Unassociated tracks can be used for **track deletion**
   - Unassociated meas. can be used for **track creation**

- Problem: Does generally not find the *global* minimum
- Conservative variant: no association in case of ambiguities

Multi-Target DA: Global NNSF

1. Build the assignment matrix $A = [d_{ij}^2]$ with

$$d_{ij}^2 = \nu_{ij}(k)^T S_j^{-1}(k) \nu_{ij}(k)$$

2. Solve the **linear assignment problem**

$$min \sum d_{ij}^2 \cdot x_{ij} \quad x_{ij} \in \{0, 1\}$$

$$\sum_i x_{ij} = 1 \quad \sum_j x_{ij} = 1$$

   - **Hungarian** method for square matrices
   - **Munkres** algorithm for rectangular matrices

3. Check if assignments are in the validation gate

Performs DA jointly, finds **global optimum.**

Multi-Target DA: Global NNSF

Linear assignment problem

- Is one of the most famous problems in linear programming and in combinatorial optimization
- Used to find the best assignment of n differently qualified workers to n jobs
- Also called "the personnel assignment problem", first solutions in the 1940s.

- By today, many efficient methods exist. The **Hungarian method**, while not the most efficient one, is still a popular algorithm
- Can also be solved for non-square problems, e.g. by **Munkres' algorithm**

Multi-Target DA: Global NNSF

Linear assignment problem

Problem statement:

We are given an $n \times n$ cost matrix $C = (cij)$, and we want to select n elements of C, so that there is exactly one element in each row and one in each column,

$$\sum_i x_{ij} = 1 \quad \sum_j x_{ij} = 1$$

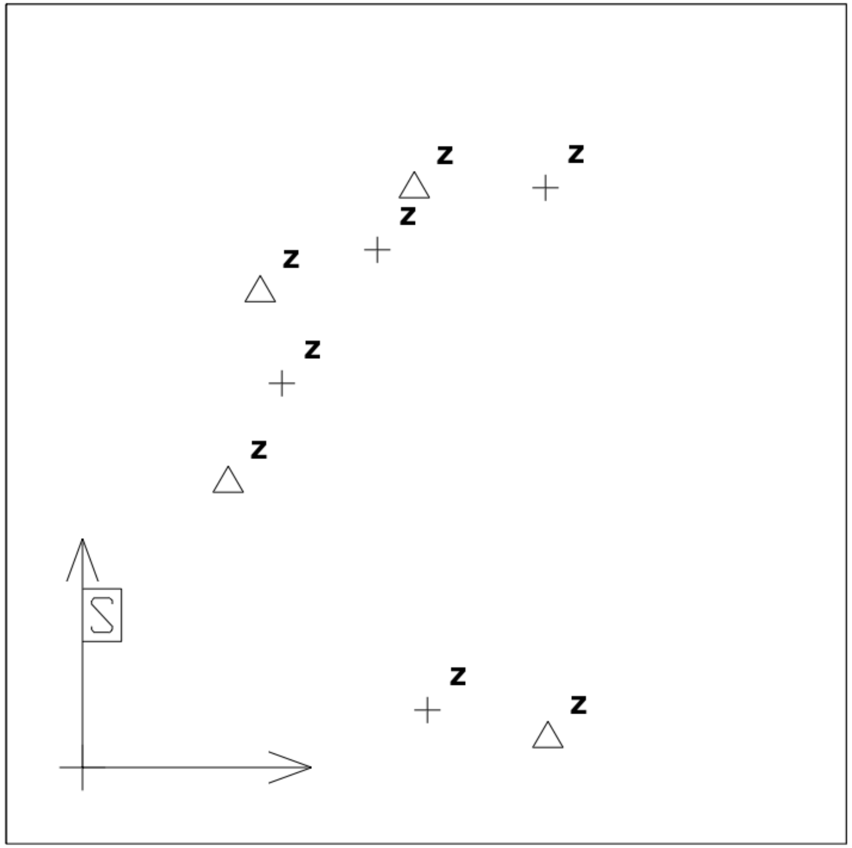
and the sum of the corresponding costs

$$min \sum d_{ij}^2 \cdot x_{ij} \quad x_{ij} \in \{0, 1\}$$

is a minimum.

Multi-Target DA: Global NNSF

Example: NNSF versus Global NNSF

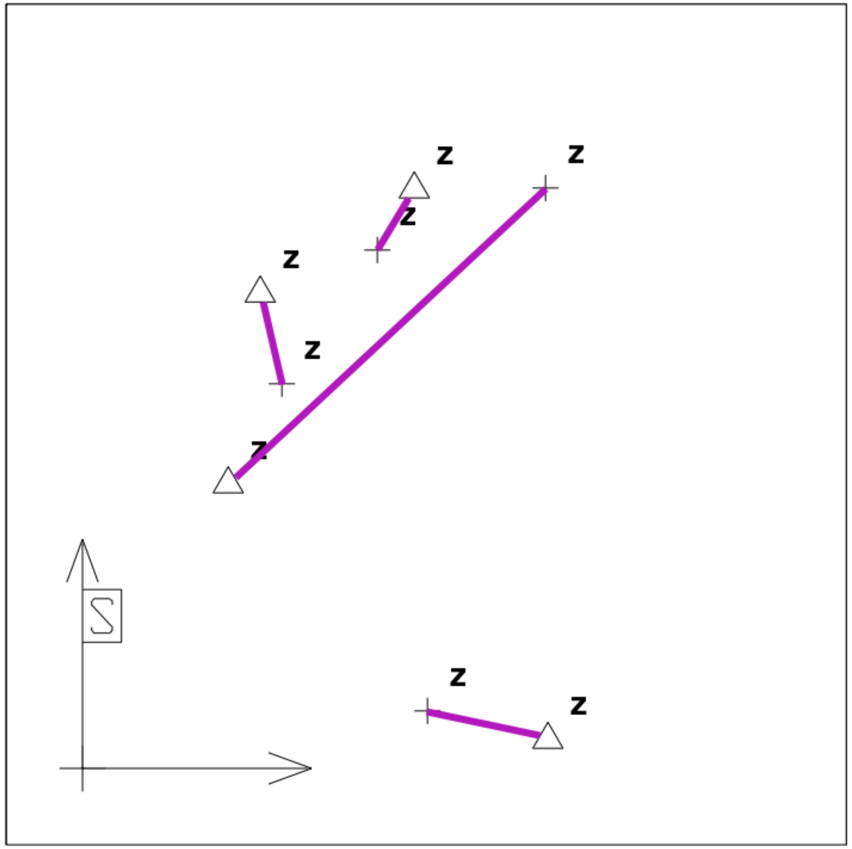


△ Observations + Predictions

Which is the best assignment?

Multi-Target DA: Global NNSF

Example: NNSF versus Global NNSF



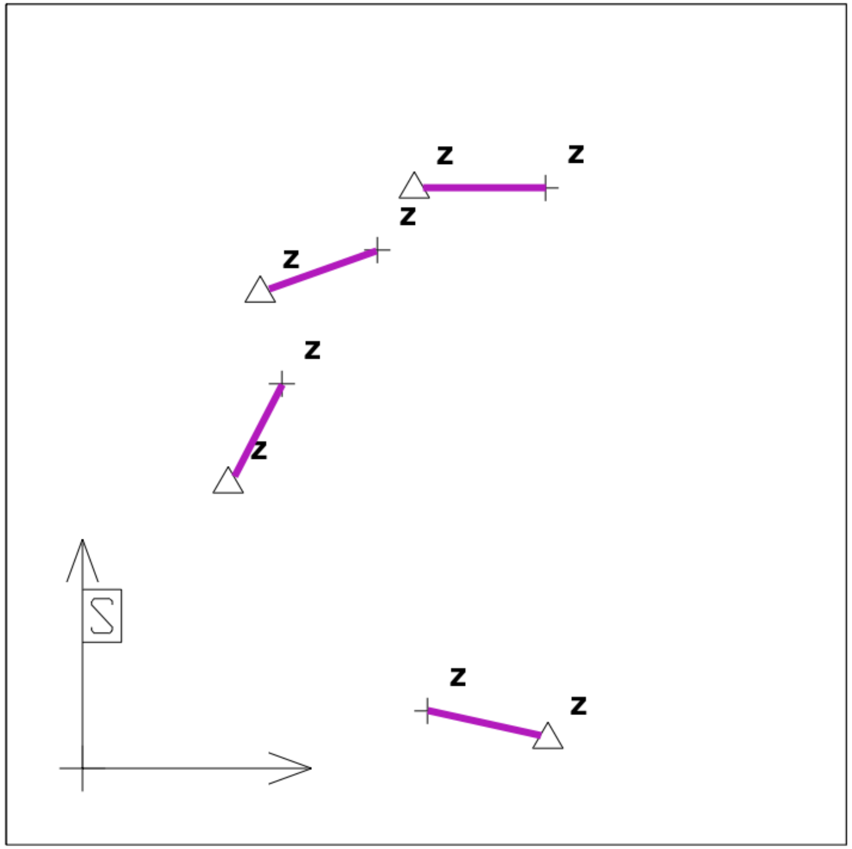
△ Observations + Predictions

NNSF:

Greedy

Multi-Target DA: Global NNSF

Example: NNSF versus Global NNSF



△ Observations + Predictions

Global NNSF:

Globally optimal

Multi-Target DA: MHT

Multiple Hypothesis Tracking

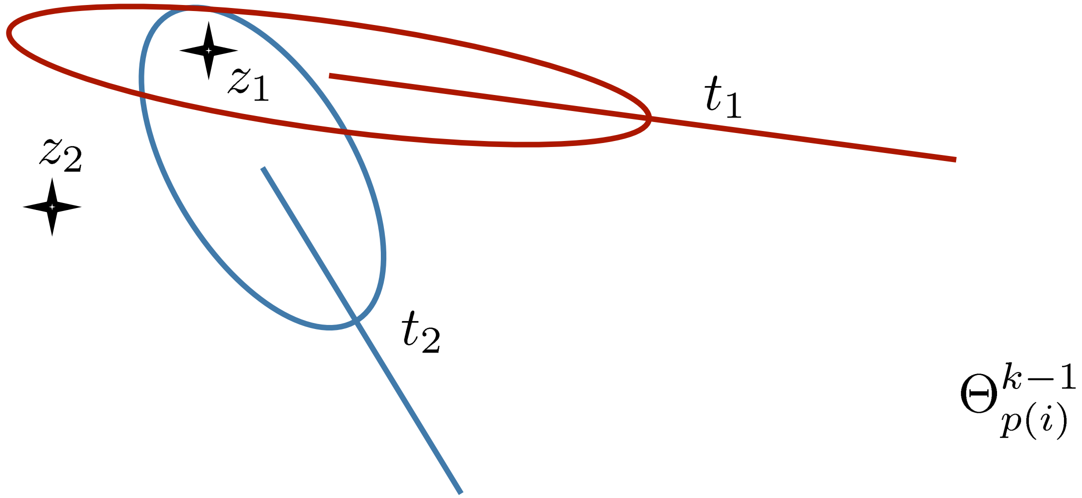
- Reasons about the
  - associations of measurements to tracks, the
  - interpretations of measurements as **false alarms** and **new tracks** and the
  - interpretations of tracks as **occluded** and **deleted**

- Optimal Bayesian solution

- Can deal with the **entire life cycle of tracks** (from initialization, confirmation, to deletion) in a probabilistically consistent way

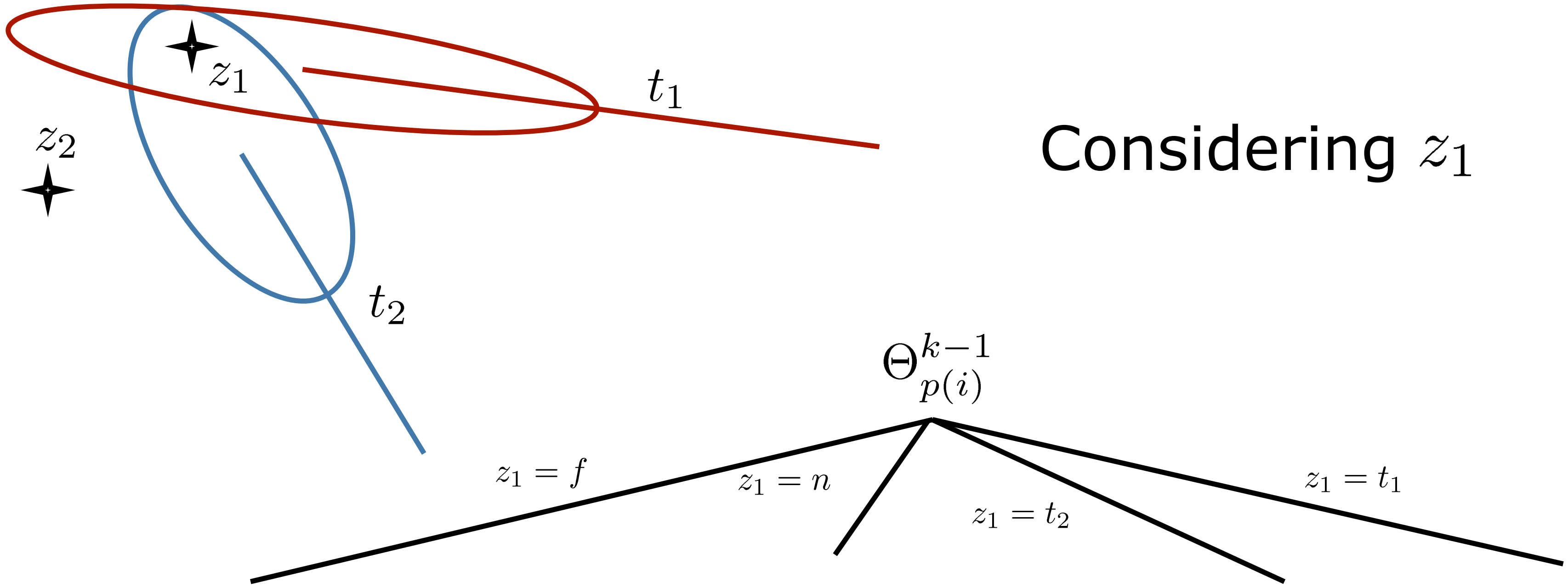
- Full solution is **exponentially complex**
  - Pruning strategies are necessary

MHT: Hypothesis Generation



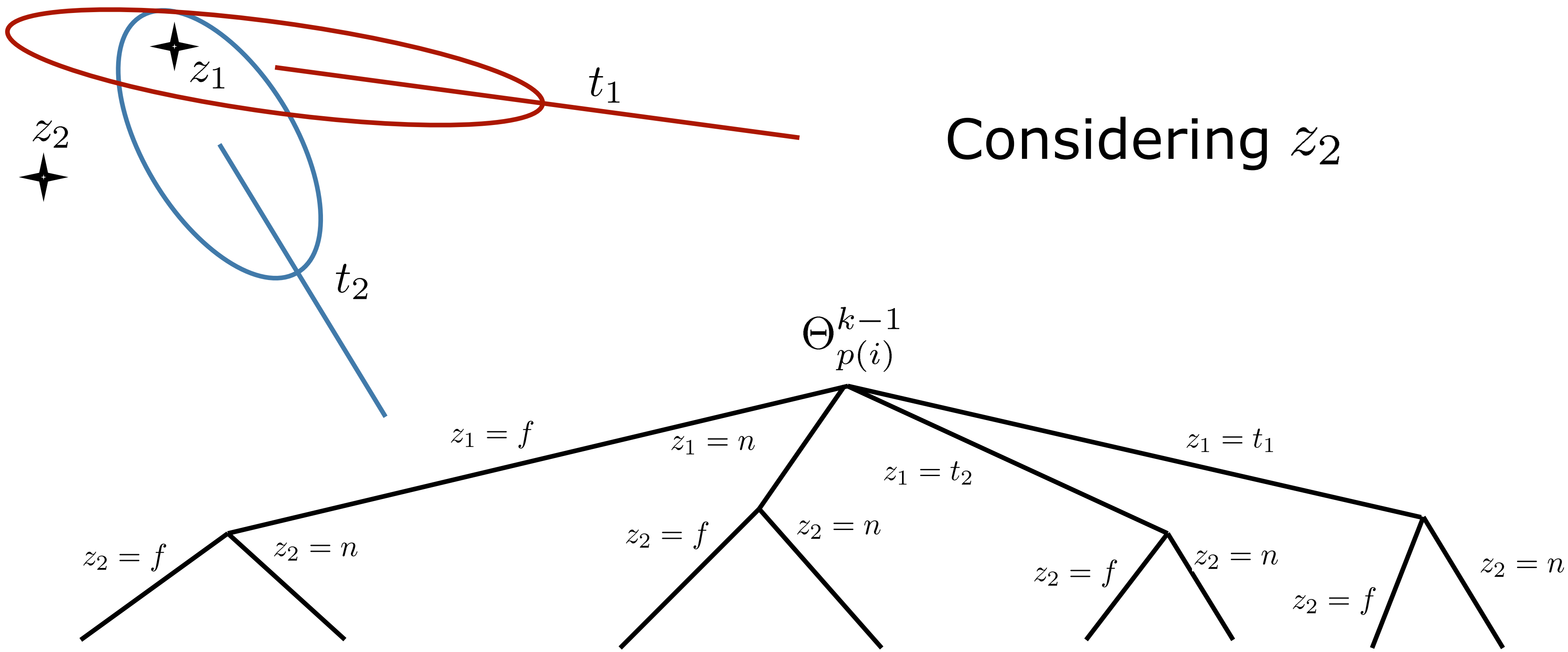
MHT: Hypothesis Generation

Considering z_1



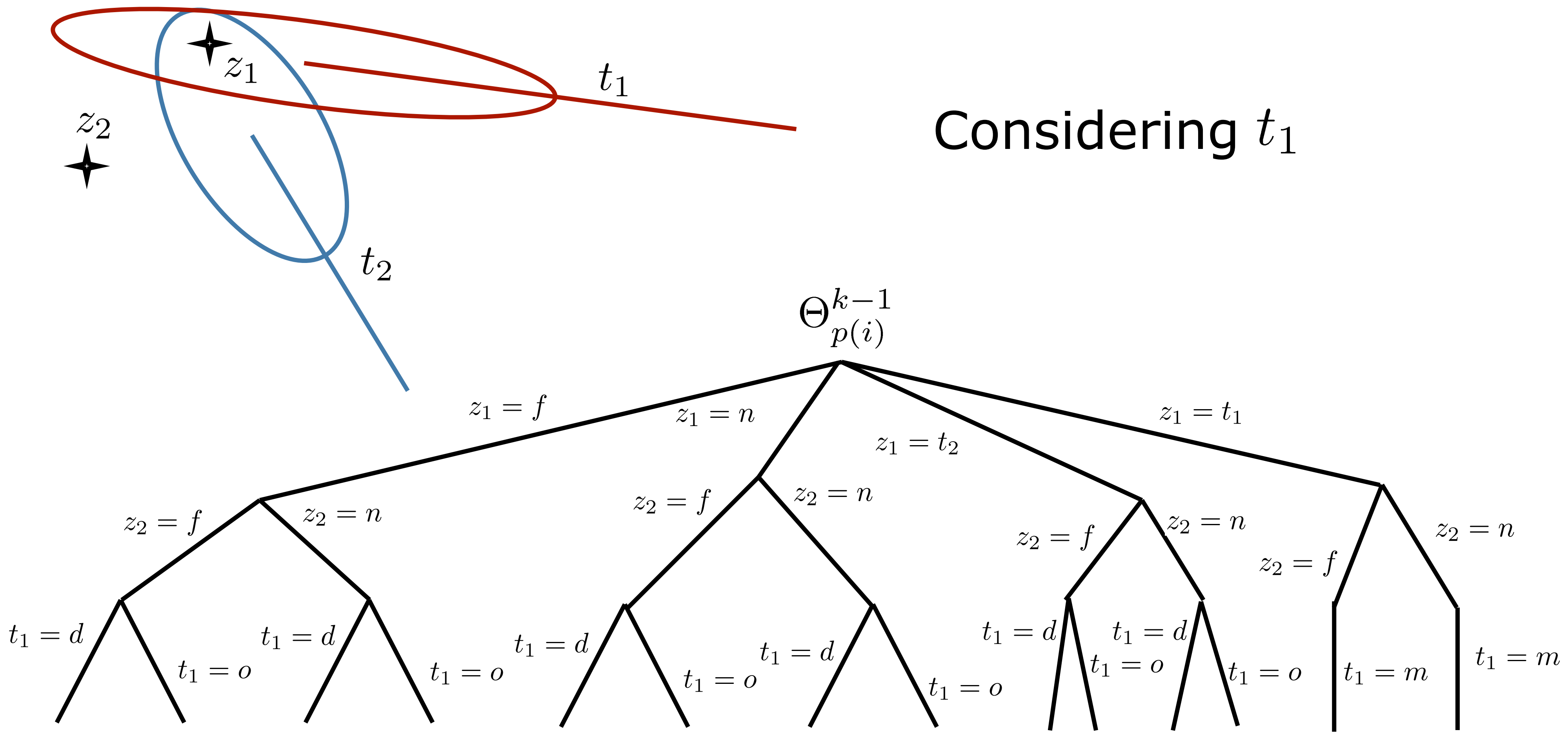
MHT: Hypothesis Generation

Considering z_2



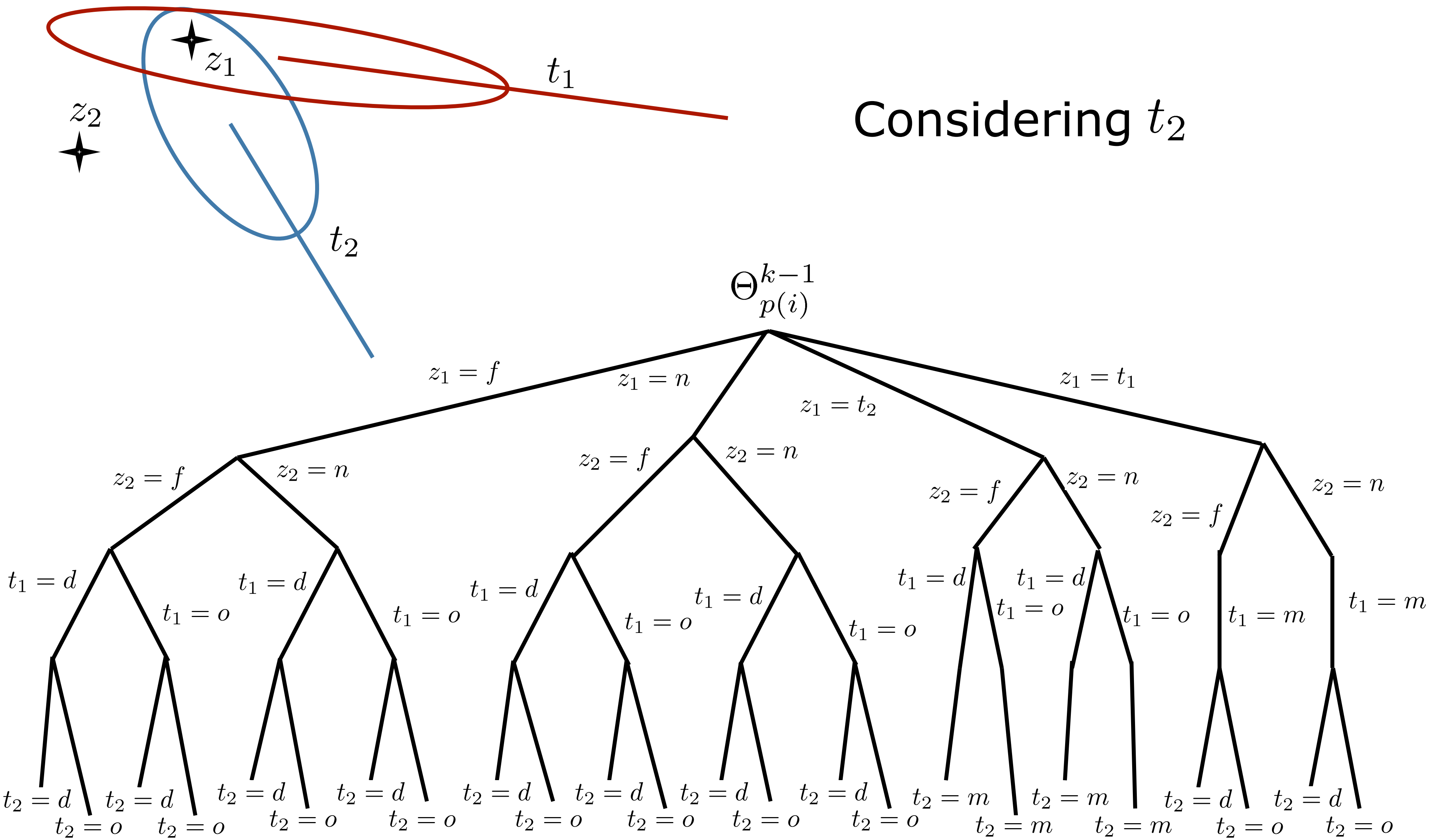
MHT: Hypothesis Generation

Considering t_1



MHT: Hypothesis Generation

Considering t_2



MHT: Hypothesis Evaluation

- The probability of an hypothesis $\Theta_i^k = \{\Theta_{p(i)}^{k-1}, \theta_{c(i)}(k)\}$ can be calculated using Bayes rules

$$P(\Theta_i^k | Z^k) = P(\Theta_{p(i)}^{k-1}, \theta_{c(i)}(k) | Z^k) =$$
$$\frac{1}{\eta} \cdot \underbrace{p(Z(k) | \Theta_{p(i)}^{k-1}, \theta_{c(i)}(k), Z^{k-1})}_{\text{Likelihood}} \cdot \underbrace{P(\theta_{c(i)}(k) | \Theta_{p(i)}^{k-1}, Z^k)}_{\text{Assignment probability}} \cdot \underbrace{P(\Theta_{p(i)}^{k-1} | Z^{k-1})}_{\text{Prior}}$$

MHT: Hypothesis Evaluation

- Likelihood

$$p(Z(k)|\Theta^{k-1}, \theta(k), Z^{k-1}) = \prod_{l=1}^{m(k)} p(z_l(k)|\Theta^{k-1}, \theta(k), Z^{k-1})$$

  - Case 1: associated with track t

$$p(z_l(k)|\Theta^{k-1}, \theta(k), Z^{k-1}) = \mathcal{N}(z_l(k); \hat{z}_t(k|k-1), S_t(k))$$

  - Case 2: false alarm

$$p(z_l(k)|\Theta^{k-1}, \theta(k), Z^{k-1}) = V^{-1}$$

  - Case 3: new track

$$p(z_l(k)|\Theta^{k-1}, \theta(k), Z^{k-1}) = V^{-1}$$

MHT: Hypothesis Evaluation

- Assignment probability

$$P(\theta(k)|\Theta^{k-1}, Z^k) = P(\theta(k)|N_M, N_O, N_D, N_N, N_F) \cdot \cdot P(N_M, N_O, N_D, N_N, N_F|\theta(k), \Theta^{k-1})$$

  - $P(N_M, N_O, N_D, N_N, N_F|\theta(k), \Theta^{k-1})$ is the probability of having N_M matched tracks, N_O occluded tracks, N_D deleted tracks, N_F false alarm and N_N new tracks.

  - $P(\theta(k)|N_M, N_O, N_D, N_N, N_F)$ is the probability of a possible configuration $\theta(k)$ given the number of events defined before

MHT: Hypothesis Evaluation

- Assignment probability 1: $P(N_M, N_O, N_D, N_N, N_F | \theta(k), \Theta^{k-1})$
  - Assuming a multinomial distribution for track labels

$$P(N_M, N_O, N_D | \theta(k), \Theta^{k-1}) = \frac{N_T!}{N_M! N_O! N_D!} p_M^{N_M} p_O^{N_O} p_D^{N_D}$$

  - Assuming a Poisson distribution for new tracks

$$P(N_N | \theta(k), \Theta^{k-1}) = \frac{(V\lambda_N)^{N_N} e^{-V\lambda_N}}{N_N!}$$

  - Assuming a Poisson distribution for false alarm

$$P(N_F | \theta(k), \Theta^{k-1}) = \frac{(V\lambda_F)^{N_F} e^{-V\lambda_F}}{N_F!}$$

  - We obtain

$$\boxed{P(\cdot) = \frac{N_T!(e^{-V\lambda_N})(e^{-V\lambda_F})}{N_N! N_F! N_M! N_O! N_D!} (V\lambda_N)^{N_N} (V\lambda_F)^{N_F} p_M^{N_M} p_O^{N_O} p_D^{N_D}}$$

MHT: Hypothesis Evaluation

- Assignment probability 2: $P(\theta(k)|N_M, N_O, N_D, N_N, N_F)$
  - The possible choices of taken as matched tracks

$$\binom{m(k)}{N_M} Perm(N_M, N_T) = \frac{m(k)!}{N_M!(m(k)-N_M)!} \frac{N_T!}{(N_T-N_M)!}$$

  - The combinations of taken as new tracks or false alarms

$$\binom{m(k)-N_M}{N_N} \underbrace{\cancel{\binom{m(k)-N_M-N_N}{N_F}}}_{=1} = \frac{(m(k)-N_M)!}{N_N!(m(k)-N_M-N_N)!}$$

  - The combinations of taken as occluded or deleted

$$\binom{N_T-N_M}{N_O} \underbrace{\cancel{\binom{N_T-N_M-N_O}{N_D}}}_{=1} = \frac{(N_T-N_M)!}{N_O!(N_T-N_M-N_O)!}$$

  - The probability is 1 over all the possible choices

$$\left[\frac{m(k)!}{N_M!(m(k)-N_M)!} \frac{N_T!}{(N_T-N_M)!} \frac{(m(k)-N_M)!}{N_N!(m(k)-N_M-N_N)!} \frac{(N_T-N_M)!}{N_O!(N_T-N_M-N_O)!} \right]^{-1}$$

MHT: Hypothesis Evaluation

- Assignment probability 2: $P(\theta(k)|N_M, N_O, N_D, N_N, N_F)$
  - The possible choices of taken as matched tracks

$$\binom{m(k)}{N_M} Perm(N_M, N_T) = \frac{m(k)!}{N_M!(m(k)-N_M)!}\frac{N_T!}{(N_T-N_M)!}$$

  - The combinations of taken as new tracks or false alarms

$$\binom{m(k)-N_M}{N_N}\cancel{\binom{m(k)-N_M-N_N}{N_F}}=1 = \frac{(m(k)-N_M)!}{N_N!(m(k)-N_M-N_N)!}$$

  - The combinations of taken as occluded or deleted

$$\binom{N_T-N_M}{N_O}\cancel{\binom{N_T-N_M-N_O}{N_D}}=1 = \frac{(N_T-N_M)!}{N_O!(N_T-N_M-N_O)!}$$

  - The probability is 1 over all the possible choices

$$\left[\frac{m(k)!}{N_M!\cancel{(m(k)-N_M)!}}\frac{N_T!}{\cancel{(N_T-N_M)!}}\frac{\cancel{(m(k)-N_M)!}}{N_N!\underbrace{(m(k)-N_M-N_N)}_{N_D}!}\frac{\cancel{(N_T-N_M)!}}{N_O!\underbrace{(N_T-N_M-N_O)}_{N_F}!}\right]^{-1}$$

MHT: Hypothesis Evaluation

- Assignment probability 2: $P(\theta(k)|N_M, N_O, N_D, N_N, N_F)$
  - The possible choices of taken as matched tracks

$$\binom{m(k)}{N_M} Perm(N_M, N_T) = \frac{m(k)!}{N_M!(m(k)-N_M)!} \frac{N_T!}{(N_T-N_M)!}$$

  - The combinations of taken as new tracks or false alarms

$$\binom{m(k)-N_M}{N_N} \underbrace{\cancel{\binom{m(k)-N_M-N_N}{N_F}}}_{=1} = \frac{(m(k)-N_M)!}{N_N!(m(k)-N_M-N_N)!}$$

  - The combinations of taken as occluded or deleted

$$\binom{N_T-N_M}{N_O} \underbrace{\cancel{\binom{N_T-N_M-N_O}{N_D}}}_{=1} = \frac{(N_T-N_M)!}{N_O!(N_T-N_M-N_O)!}$$

  - The probability is 1 over all the possible choices

$$\boxed{P(\theta(k)|N_M, N_O, N_D, N_N, N_F) = \frac{N_M!N_N!N_F!N_O!N_D!}{m(k)!N_T!}}$$

MHT: Hypothesis Evaluation

- Assignment probability

$$
\begin{aligned}
P(\theta(k)|\Theta^{k-1}, Z^k) \quad &= \quad P(\theta(k)|N_M, N_O, N_D, N_N, N_F) \cdot \\
&\quad \cdot P(N_M, N_O, N_D, N_N, N_F|\theta(k), \Theta^{k-1})
\end{aligned}
$$

- Combining everything together we have

$$
P(\theta(k)|\Theta^{k-1}, Z^k) =
$$

$$
= \quad \frac{N_T!(e^{-V\lambda_N})(e^{-V\lambda_F})}{N_N!N_F!N_M!N_O!N_D!}(V\lambda_N)^{N_N}(V\lambda_F)^{N_F} p_M^{N_M} p_O^{N_O} p_D^{N_D} \frac{N_M!N_N!N_F!N_O!N_D!}{m(k)!N_T!}
$$

MHT: Hypothesis Evaluation

- Assignment probability

$$
\begin{aligned}
P(\theta(k)|\Theta^{k-1}, Z^k) &= P(\theta(k)|N_M, N_O, N_D, N_N, N_F) \cdot \\
&\cdot P(N_M, N_O, N_D, N_N, N_F|\theta(k), \Theta^{k-1})
\end{aligned}
$$

- Combining everything together we have

$$
P(\theta(k)|\Theta^{k-1}, Z^k) =
$$

$$
= \frac{\cancel{N_F!}(e^{-V\lambda_N})(e^{-V\lambda_F})}{\cancel{N_N! N_F! N_M! N_O! N_D!}} (V\lambda_N)^{N_N} (V\lambda_F)^{N_F} p_M^{N_M} p_O^{N_O} p_D^{N_D} \frac{\cancel{N_M! N_N! N_F! N_O! N_D!}}{m(k)! \cancel{N_T!}}
$$

- Simplifying the expression we obtain

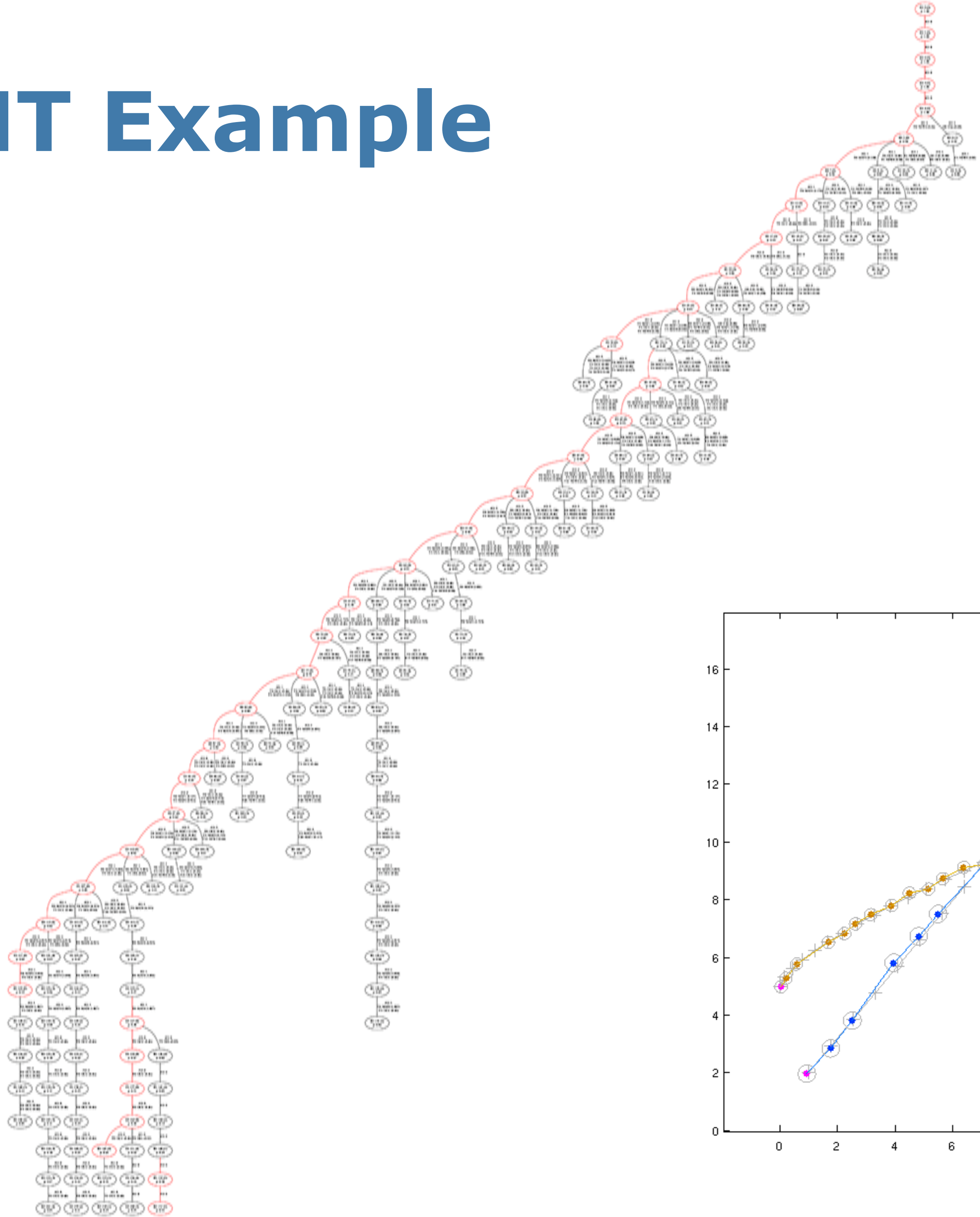
$$
\boxed{P(\theta(k)|\Theta^{k-1}, Z^k) = \frac{(e^{-V\lambda_N})(e^{-V\lambda_F})}{m(k)!} (V\lambda_N)^{N_N} (V\lambda_F)^{N_F} p_M^{N_M} p_O^{N_O} p_D^{N_D}}
$$

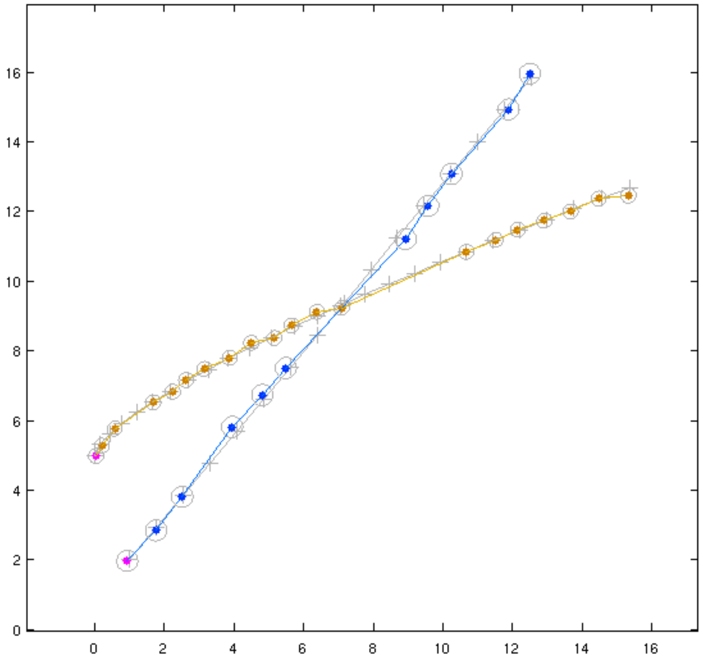
MHT Approximations

- **Spatially disjoint** hypothesis trees
  - Tracks are partitioned into clusters based on gating
  - A separate tree is grown for each cluster
- **K-best** hypothesis tree
  - Directly generate the k-best hypothesis
  - Generation and evaluation in a single step

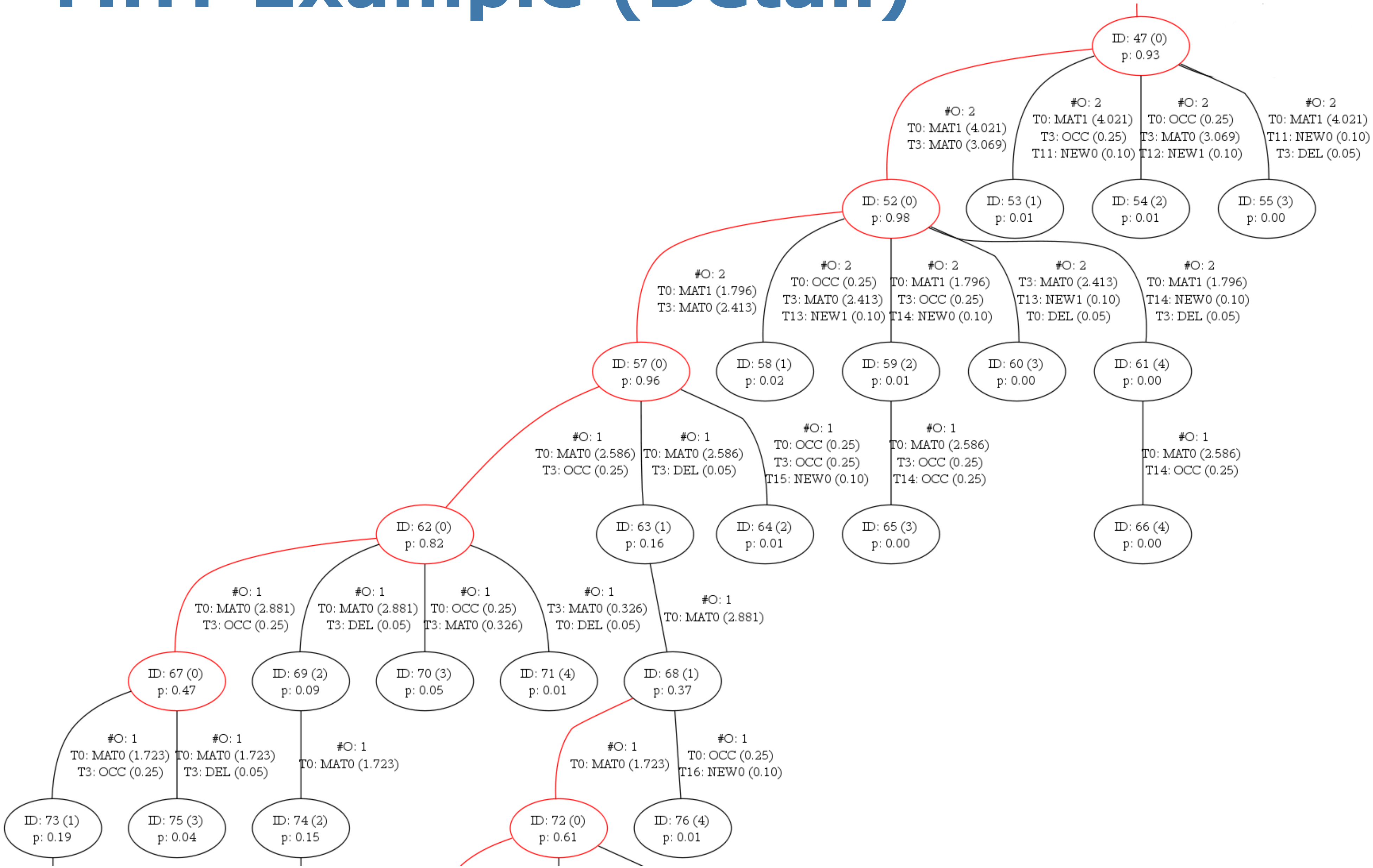
    by Murty's algorithm and a linear assignment solver
  - Implements a *generate-while-prune* versus a *generate-then-prune* strategy
- **N-Scan back** pruning
  - Ambiguities are supposed to be resolved after N steps
  - Children at step k+N give the prob. of parents at step k
  - Keep only the most probable branch

MHT Example





MHT Example (Detail)



Multi-Target DA: Summary

- **Nearest Neighbor Standard filters** (both types)
  - Simple to implement
  - NNSF: Only good if tracks are well separated, low DA ambiguity
  - Global NNSF: Finds globally optimal assignment solution
  - NNSF+Global NNSF: No integration over time

- **MHT**
  - Fully Bayesian
  - Most general DA framework
  - Complex and expensive
  - Only approximations are practical to implement

- Not treated: JPDAF and variants, MCMC