Target Tracking: Lecture 6
Multiple Target Tracking: Part II

Umut Orguner
umut@metu.edu.tr
room: EZ-12 tel: 4425

Department of Electrical & Electronics Engineering
Middle East Technical University
Ankara, Turkey

Lecture Outline

- Multiple Hypothesis Tracking (MHT) (50%)
  - Conceptual MHT
    (hypothesis based and most of the time infeasible or inefficient)
- Feasible Implementations of MHT (45%)
  - Hypothesis Based Implementation
    - N-best solutions to the assignment problem
  - Track Based Implementation
  - Output Presentation for MHT
- General: Which multi TT method to choose? (5%)

Multiple Hypothesis Tracking (MHT)

- MHT is the name given to a type of TT algorithms that keep at each time step multiple hypotheses about the past and current association uncertainties.
- The first structured MHT (which we will call "conceptual MHT") was described by D. B. Reid in 1979.
- These algorithms do not use a separate track initialization procedure and hence track initiation is integrated into the algorithm.
- Between consecutive time instants, if the implemented MHT algorithm keeps
  - hypotheses
  - tracks

  the corresponding MHT implementation is called
  - hypothesis based
  - track based

  respectively.

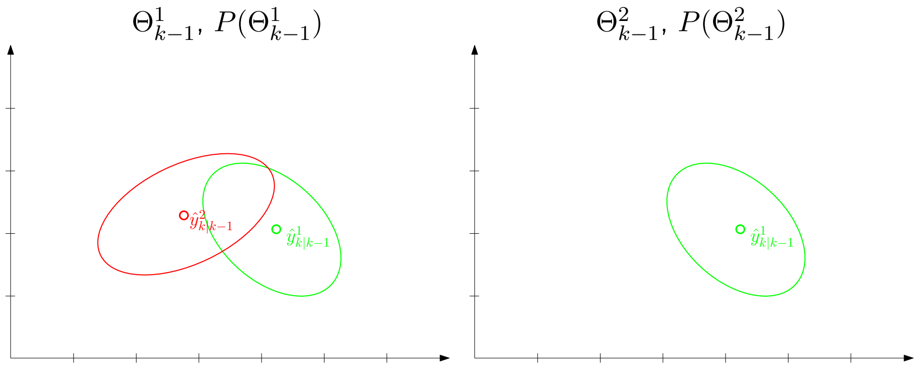
Conceptual MHT

- It was first described by D. B. Reid in 1979.
- It is an hypothesis based brute force implementation i.e., between consecutive time instants, different hypotheses $\{\Theta_{k-1}^i\}_{i=1}^{N_h}$ about the past are kept in the memory.
- The idea is to generate all possible hypotheses and then to depend on pruning of these hypotheses, otherwise, it has a combinatoric explosion in the number of hypotheses.
- Uses techniques such as
  - Clustering;
  - Pruning of low probability hypotheses;
  - N-scan pruning;
  - Combining similar hypotheses

  to reduce the number of hypotheses.

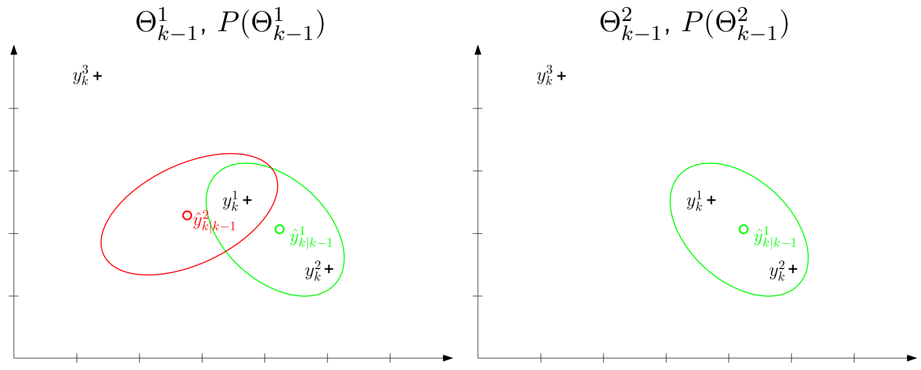
Conceptual MHT

Each of the hypotheses $\{\Theta_{k-1}^i\}_{i=1}^{N_h}$ kept about the past are characterized by their assumed number of targets (tracks) and their corresponding sufficient statistics.



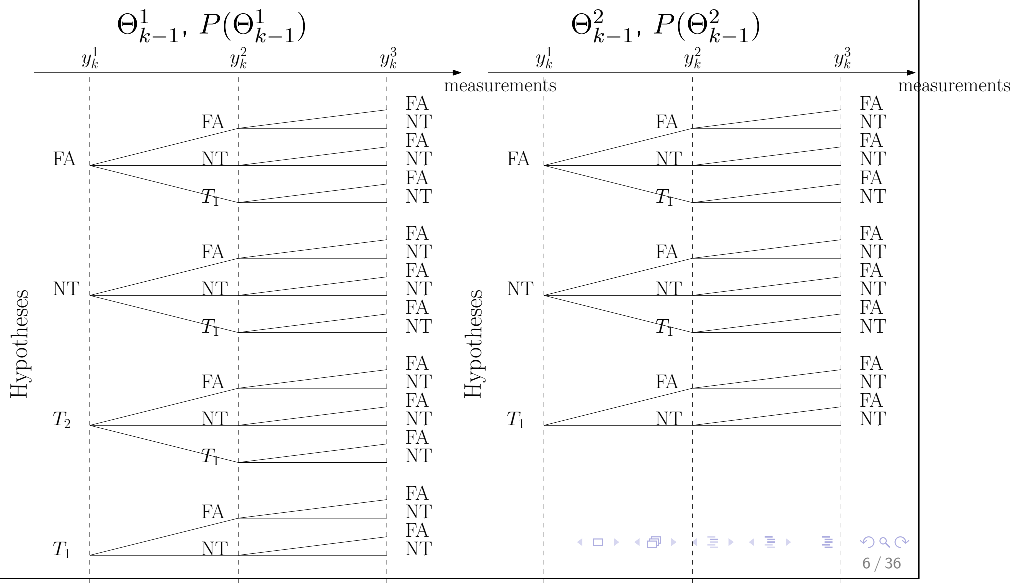
Conceptual MHT

Each of the hypotheses $\{\Theta_{k-1}^i\}_{i=1}^{N_h}$ kept about the past are characterized by their assumed number of targets (tracks) and their corresponding sufficient statistics.



Conceptual MHT

Hypothesis Generation: Form $\Theta_k^\ell \triangleq \{\theta_k, \Theta_{k-1}^i\}$



Conceptual MHT

Hypothesis Probability: Let $\Theta_k^\ell \triangleq \{\theta_k, \Theta_{k-1}^i\}$

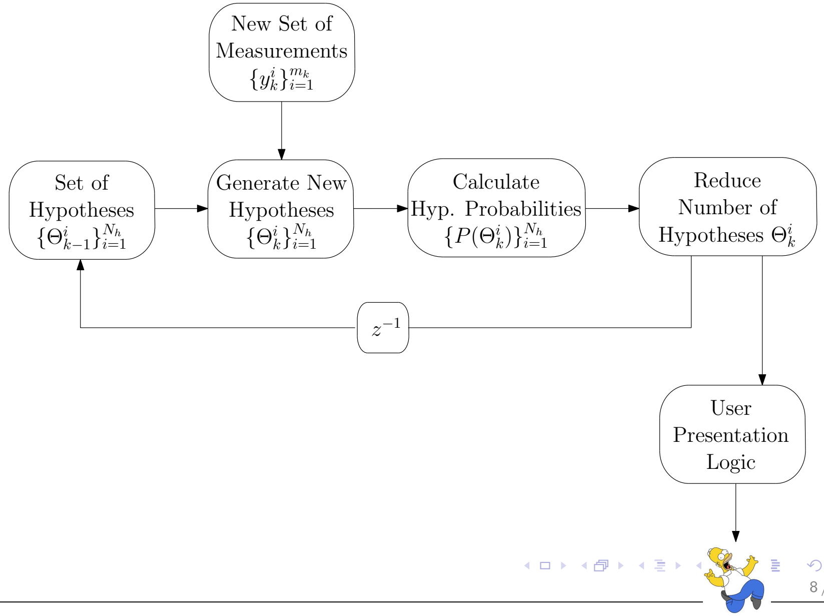
$$P(\Theta_k^\ell|y_{0:k}) \propto p(y_k|\Theta_k^\ell, y_{0:k-1})P(\theta_k|\Theta_{k-1}^i, y_{0:k-1})P(\Theta_{k-1}^i|y_{0:k-1})$$

$$\propto \beta_{FA}^{m_k^{FA}}\beta_{NT}^{m_k^{NT}}\left[\prod_{j\in\mathcal{J}_D^i} P_D^j p_{k|k-1}^j(y_k^{\theta_k^{-1}(j)})\right]\left[\prod_{j\in\mathcal{J}_{ND}^i}(1-P_D^jP_G^j)\right]P(\Theta_{k-1}^i|y_{0:k-1})$$

where we have used the "Fundamental Theorem of TT" introduced at the last lecture.

Important remark: Note that the sets $\mathcal{J}_D^i$ and $\mathcal{J}_{ND}^i$ are dependent on the previous hypothesis Θ_{k-1}^i because the number of targets and estimates of the targets can be different for each different previous hypothesis.

Conceptual MHT



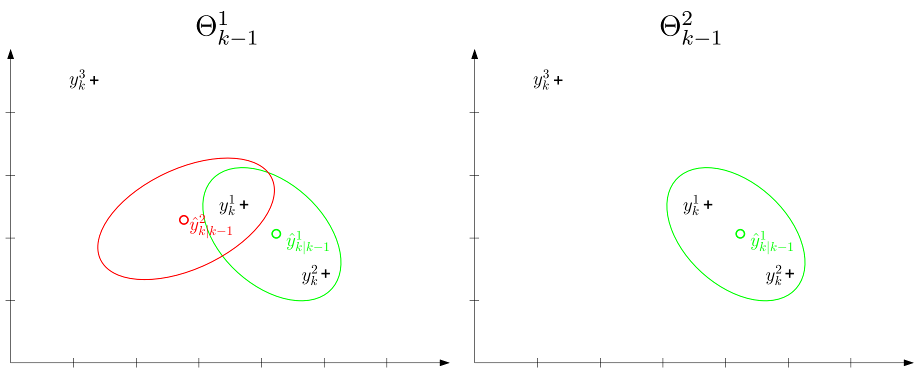
Conceptual MHT

Computation and hypothesis number reduction techniques

- Clustering.
- Pruning of low probability hypotheses.
- N-scan pruning
- Combining similar hypotheses

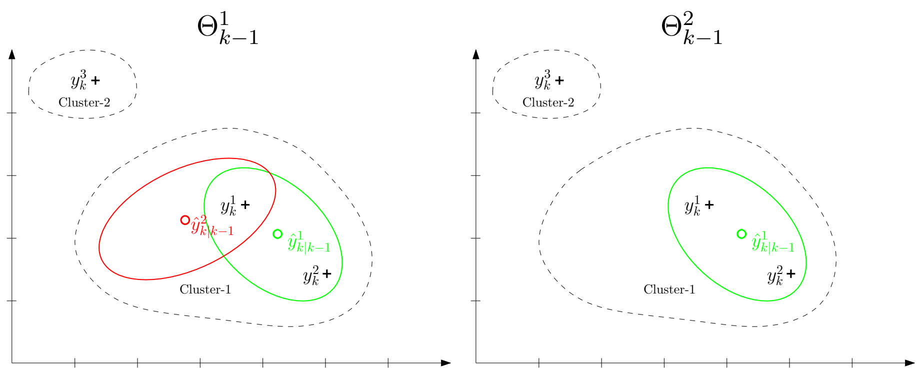
Conceptual MHT

Clustering is the processing of hypotheses about the groups of targets (tracks) that do not share measurements (in the gates) separately.



Conceptual MHT

Clustering is the processing of hypotheses about the groups of targets (tracks) that do not share measurements (in the gates) separately.



Conceptual MHT

Cluster Management:

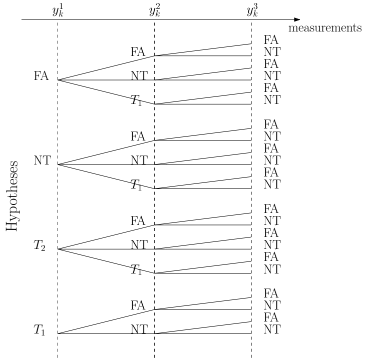
- When targets get closer
  - If at any time instant a measurement falls inside the gates of two tracks that have different clusters, the corresponding two clusters must be merged into a super-cluster.
  - The hypotheses for each cluster are combined into super-hypotheses.
- When targets separate
  - If a group of tracks in a cluster did not share measurements (inside their gates) with the rest of the of the tracks in the cluster for some specified time period, the cluster can be divided into two smaller clusters.
  - Hypotheses for the cluster are also divided into smaller hypotheses corresponding to two smaller clusters.

Conceptual MHT

Process Each Cluster Separately: Form $\Theta_k^\ell \triangleq \{\theta_k, \Theta_{k-1}^i\}$ for each cluster as if the other clusters do not exist.

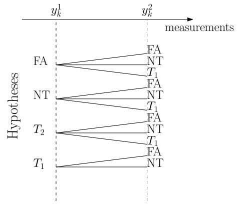
Before clustering

$\Theta_{k-1}^1,\ P(\Theta_{k-1}^1)$



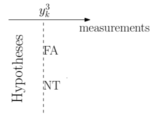
Cluster-1

$\Theta_{k-1}^1,\ P(\Theta_{k-1}^1)$



Cluster-2

$\Theta_{k-1}^1,\ P(\Theta_{k-1}^1)$

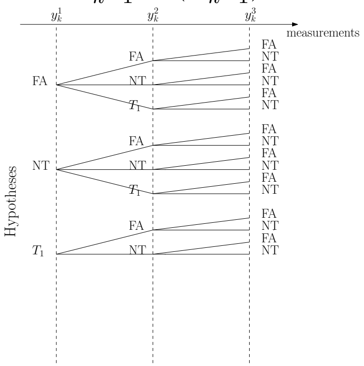


Conceptual MHT

Process Each Cluster Separately: Form $\Theta_k^\ell \triangleq \{\theta_k, \Theta_{k-1}^i\}$ for each cluster as if the other clusters do not exist.

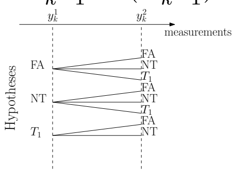
Before clustering

$\Theta_{k-1}^2,\ P(\Theta_{k-1}^2)$



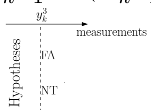
Cluster-1

$\Theta_{k-1}^2,\ P(\Theta_{k-1}^2)$



Cluster-2

$\Theta_{k-1}^2,\ P(\Theta_{k-1}^2)$



Conceptual MHT

Pruning low probability hypotheses: For each cluster

- One can delete hypotheses that has probability less than a threshold (e.g. 0.001).

$$\text{Deletion Condition: } P(\Theta_k^i) < \gamma_p$$

- Another idea is to sum the probabilities of the hypotheses in descending order and discard the ones over a predetermined probability mass (e.g. 0.99).

$$\text{Deletion Condition: } \sum_{\ell=1}^{i} P(\Theta_k^\ell) > \gamma_c$$

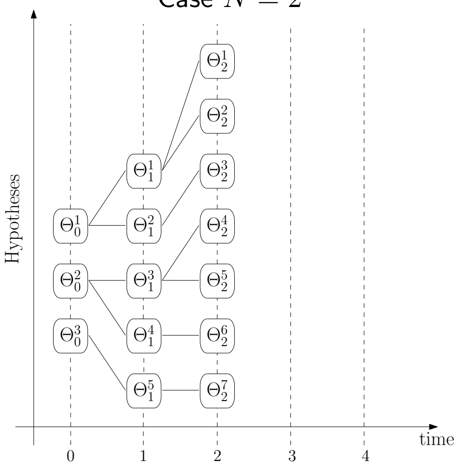
where the ordering $\ell = 1, \ldots, N_h$ is descending in probabilities.

Conceptual MHT

N-scan Pruning:

- This scheme assumes that any uncertainty at time $k - N$ is perfectly resolved by the time k for all k.
- It is a general commonsense to choose $N \geq 5$
- For this purpose, N last ancestors of each created hypothesis is kept in memory.

Case $N = 2$

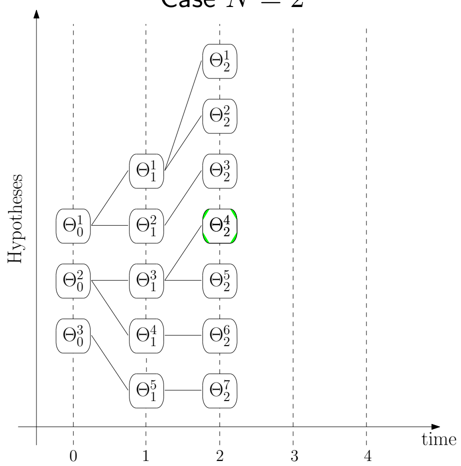


Conceptual MHT

N-scan Pruning:

- This scheme assumes that any uncertainty at time $k - N$ is perfectly resolved by the time k for all k.
- It is a general commonsense to choose $N \geq 5$
- For this purpose, N last ancestors of each created hypothesis is kept in memory.

Case $N = 2$

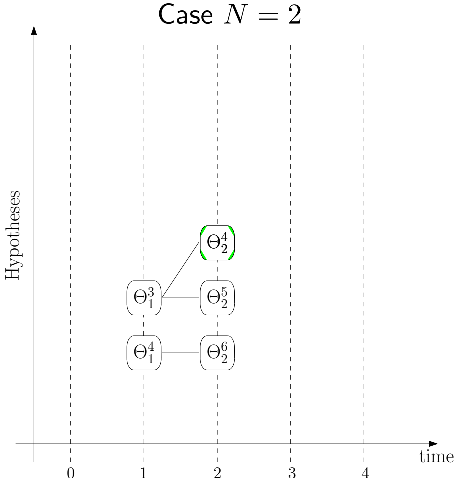


Conceptual MHT

N-scan Pruning:

- This scheme assumes that any uncertainty at time $k - N$ is perfectly resolved by the time k for all k.
- It is a general commonsense to choose $N \geq 5$
- For this purpose, N last ancestors of each created hypothesis is kept in memory.

Case $N = 2$

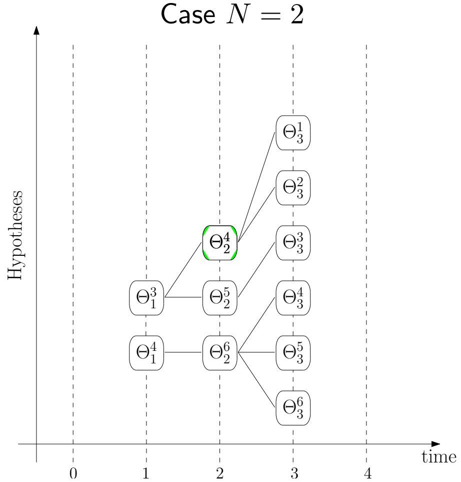


Conceptual MHT

N-scan Pruning:

- This scheme assumes that any uncertainty at time $k - N$ is perfectly resolved by the time k for all k.
- It is a general commonsense to choose $N \geq 5$
- For this purpose, N last ancestors of each created hypothesis is kept in memory.

Case $N = 2$

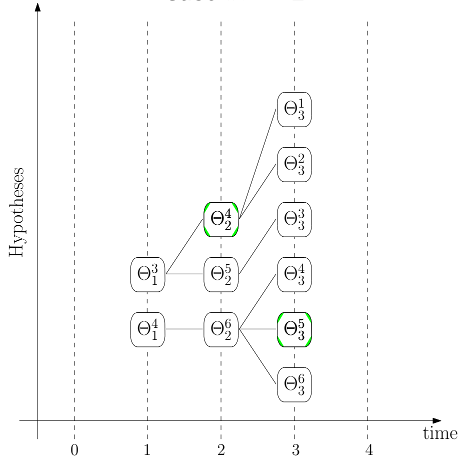


Conceptual MHT

N-scan Pruning:

- This scheme assumes that any uncertainty at time $k-N$ is perfectly resolved by the time k for all k.
- It is a general commonsense to choose $N \geq 5$
- For this purpose, N last ancestors of each created hypothesis is kept in memory.

Case $N = 2$

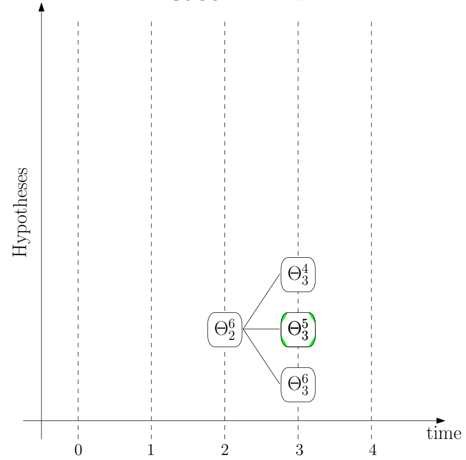


Conceptual MHT

N-scan Pruning:

- This scheme assumes that any uncertainty at time $k-N$ is perfectly resolved by the time k for all k.
- It is a general commonsense to choose $N \geq 5$
- For this purpose, N last ancestors of each created hypothesis is kept in memory.

Case $N = 2$



Conceptual MHT

Hypothesis Merging: It is also suggested in Reid's original paper to check for every hypothesis pair that

- two hypotheses have same number of targets (tracks)
- the estimates of the tracks are close to the corresponding ones in the other hypothesis

If these conditions are satisfied

- the two hypotheses are merged into a single hypothesis
- the resulting single hypothesis is assigned the probability that is obtained by summing the probabilities of the individual hypotheses.

Conceptual MHT

- Conceptual MHT is attractive in the sense that each hypothesis saved between time instants is an alternative representation of reality and can be interpreted easily.
- However, except for toy examples, generating all possible hypotheses and then discarding most of them was deemed inefficient because we are basically spending computation for hypotheses that we, at the end, throw away.
- Moreover, some hypotheses keep different combinations of exactly the same tracks. Hence the number of actual tracks we are considering is much less than the number of hypotheses.
- For these reasons, an alternative track based implementation was adopted until an efficient way to implement a hypothesis based MHT was found in [Cox (1996)] in 1996.

Hypothesis Based Implementation

- First proposed by Cox and Hingorani in 1996 in [Cox (1996)].
- Instead of generating all hypotheses in the Conceptual MHT, they proposed generating only the best hypotheses without generating hypotheses that will possibly be deleted.
- N-best solutions to the assignment problem, which was introduced at the last lecture with GNN, is used.
- The so called **Murty's algorithm** found in 1968 is used to find the N-best solutions to the assignment problem.
- In the hypothesis based implementation, N_h-best hypothesis are found with minimum number of unnecessary hypothesis generations.
- The same number of hypothesis reduction techniques can still be used.

Hypothesis Based Implementation

Reminder of Assignment Problem Let $\Theta_k^\ell \triangleq \{\theta_k, \Theta_{k-1}^i\}$.

$$P(\Theta_k^\ell|y_{0:k}) \propto p(y_k|\Theta_k^\ell, y_{0:k-1})P(\theta_k|\Theta_{k-1}^i, y_{0:k-1})P(\Theta_{k-1}^i|y_{0:k-1})$$

$$\propto \beta_{FA}^{m_k^{FA}} \beta_{NT}^{m_k^{NT}} \left[\prod_{j\in\mathcal{J}_D^i} P_D^j p_{k|k-1}^j(y_k^{\theta_k^{-1}(j)})\right]\left[\prod_{j\in\mathcal{J}_{ND}^i}(1-P_D^jP_G^j)\right]P(\Theta_{k-1}^i|y_{0:k-1})$$

Divide and multiply the right hand side by

$$C_i \triangleq \prod_{j=1}^{n_T^i}(1-P_D^jP_G^j) = \prod_{j\in\mathcal{J}_D^i}(1-P_D^jP_G^j)\prod_{j\in\mathcal{J}_{ND}^i}(1-P_D^jP_G^j)$$

Hypothesis Based Implementation

Reminder of Assignment Problem

$$P(\Theta_k^\ell|y_{0:k}) \propto \beta_{FA}^{m_k^{FA}} \beta_{NT}^{m_k^{NT}} \left[\prod_{j\in\mathcal{J}_D^i} \frac{P_D^j p_{k|k-1}^j(y_k^{\theta_k^{-1}(j)})}{1-P_D^jP_G^j}\right] C_i P(\Theta_{k-1}^i|y_{0:k-1})$$

Taking the logs and forming the assignment matrices

- $\times$ represents $-\infty$.
- $\ell_{ij} \triangleq \log \frac{P_D^j p_{k|k-1}^j(y_k^i)}{(1-P_D^jP_G^j)}$.

$\mathcal{A}_1$	T_1	T_2	FA$_1$	FA$_2$	FA$_3$	NT$_1$	NT$_2$	NT$_3$
y_k^1	ℓ_{11}	ℓ_{12}	$\log\beta_{FA}$	$\times$	$\times$	$\log\beta_{NT}$	$\times$	$\times$
y_k^2	ℓ_{21}	$\times$	$\times$	$\log\beta_{FA}$	$\times$	$\times$	$\log\beta_{NT}$	$\times$
y_k^3	$\times$	$\times$	$\times$	$\times$	$\log\beta_{FA}$	$\times$	$\times$	$\log\beta_{NT}$

$\mathcal{A}_2$	T_1	FA$_1$	FA$_2$	FA$_3$	NT$_1$	NT$_2$	NT$_3$
y_k^1	ℓ_{11}	$\log\beta_{FA}$	$\times$	$\times$	$\log\beta_{NT}$	$\times$	$\times$
y_k^2	ℓ_{21}	$\times$	$\log\beta_{FA}$	$\times$	$\times$	$\log\beta_{NT}$	$\times$
y_k^3	$\times$	$\times$	$\times$	$\log\beta_{FA}$	$\times$	$\times$	$\log\beta_{NT}$

Hypothesis Based Implementation

Find the N-best Solutions to an Assignment Problem

- Given an assignment matrix $\mathcal{A}_i$, we can find the best solution with Auction or similar algorithms in polynomial time.
- People had considered the generalization of this problem to N-best solutions.
- The key point is to express finding N-best solutions problem into a number of best solution assignment problems.
- Then for each of the best solution assignment problems Auction algorithm can be used.

Hypothesis Based Implementation

Find the N-best Solutions to an Assignment Problem

Murty's Algorithm

Given the assignment matrix $\mathcal{A}_i$,

- Find the best solution using Auction algorithm.
- 2nd best solution:
  - Express the 2nd best solution as the solution of a number of best solution assignment problems.
  - Find the solution to each of these problems by Auction.
  - The solution giving the maximum reward (minimum cost) is the second best solution.
- Repeat the procedure if further solutions are required.

Murty's Algorithm

1) Find the best solution, S_0, to P_0 (this can be done using a standard algorithm like the Hungarian method)
2) Initialize the list of problem/solution pairs with $< P_0, S_0 >$
3) Clear the list of solutions to be returned
4) For $i = 1$ to k, or until the list of problem/solution pairs is empty
   - 4.1 Search through the list of problem/solution pairs, and find the pair, $< P, S >$ that has the best solution value
   - 4.2 Remove $< P, S >$ from the list of problem/solution pairs
   - 4.3 Add S to the list of solutions to be returned
   - 4.4 For each triple, $< t, z, l >$, found in S
     - 4.4.1 Let $P' = P$
     - 4.4.2 Remove the triple $< t, z, l >$ from P'
     - 4.4.3 Look for the best solution, S', to P'
     - 4.4.4 If S' exists
       - 4.4.4.1 Add $< P', S' >$ to the set of problem/ solution pairs.
     - 4.4.5 From P, remove all triples that include t, and all triples that include z, except $< t, z, l >$ itself. (This reduces the dimension of the problem by one)

Description taken from:
I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.18, no.2, pp.138–150, Feb. 1996.

Hypothesis Based Implementation

- **Our aim:** Given the previous hypotheses $\{\Theta^i_{k-1}\}_{i=1}^{N_h}$ and current measurements $\{y^i_k\}_{i=1}^{m_k}$, we would like to find the best N_h current hypotheses $\{\Theta^\ell_k\}_{\ell=1}^{N_h}$ without generating all the hypotheses.
- **Reminder of Hypothesis Probability**

$$P(\Theta^\ell_k|y_{0:k}) \propto \underbrace{\beta_{FA}^{m_k^{FA}} \beta_{NT}^{m_k^{NT}} \left[\prod_{j\in\mathcal{J}_D^i} \frac{P_D^j p^j_{k|k-1}(y_k^{\theta_k^{-1}(j)})}{1 - P_D^j P_G^j}\right]}_{\text{Maximized by Assignment Problem}} \underbrace{C_i P(\Theta^i_{k-1}|y_{0:k-1})}_{\substack{\text{Previous Hypothesis}\\ \text{Dependent Term}}}$$

- We would like to find $\{\Theta^\ell_k\}_{\ell=1}^{N_h}$ that maximizes $P(\Theta^\ell_k|y_{0:k})$.
- This can be obtained in two steps:
  - Obtain the solution from the assignment (Murty's algorithm)
  - Multiply the obtained quantity by previous hypothesis dependent terms.

Hypothesis Based Implementation

Generating N_h-best Hypotheses

Given previous hypotheses $\{\Theta^i_{k-1}\}_{i=1}^{N_h}$, the corresponding hypothesis probabilities $\{P(\Theta^i_{k-1}|y_{0:k-1})\}_{i=1}^{N_h}$ and current measurements $\{y^i_k\}_{i=1}^{m_k}$

- Find assignment matrices $\{\mathcal{A}_i\}_{i=1}^{N_h}$ for all previous hypotheses.
- Obtain the best hypotheses shown as $\{\Theta^{1i}_k\}_{i=1}^{N_h}$ for each assignment matrix.
- Calculate the corresponding probabilities $\{P(\Theta^{1i}_k|y_{0:k})\}_{i=1}^{N_h}$.
- Order the obtained hypotheses according to their probabilities. Call the resulting ordered list as HYP-LIST and the corresponding list of probabilities as PROB-LIST.
- ...

Hypothesis Based Implementation

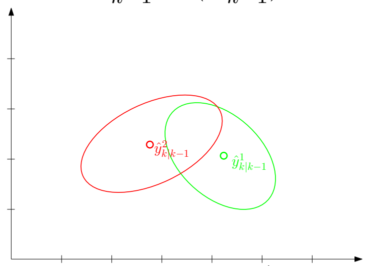
Generating N_h-best Hypotheses

- ...
- For $j = 2 : N_h$
  - Loop through the assignment matrices and find the jth best solutions $\{\Theta_k^{ji}\}_{i=1}^{N_h}$ for each of them.
  - Calculate the probabilities $\{P(\Theta_k^{ji})\}_{i=1}^{N_h}$ corresponding to $\{\Theta_k^{ji}\}_{i=1}^{N_h}$
  - If $P(\Theta_k^{ji})$ is higher than the lowest probability in PROB-LIST, add Θ_k^{ji} to HYP-LIST and the corresponding probability to PROB-LIST. Discard the lowest probability hypothesis from HYP-LIST and its corresponding probability from PROB-LIST
  - If $P(\Theta_k^{ji})$ is lower than the lowest probability in PROB-LIST discard Θ_k^{ji} and never use $\mathcal{A}_i$ again in subsequent recursions.
- The hypotheses in the HYP-LIST are the N_h best hypotheses.

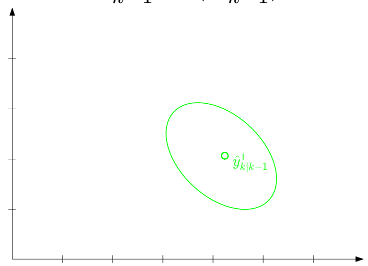
Track Based Implementation

Most of the time, hypotheses are composed of combinations of exactly same tracks. The number of tracks might be significantly lower than the number of hypotheses.

Θ_{k-1}^1, $P(\Theta_{k-1}^1)$



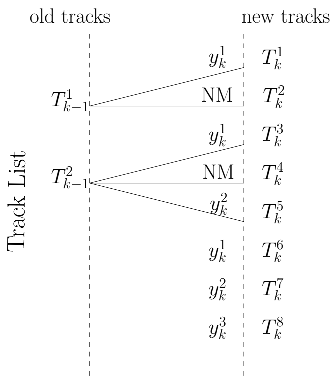
Θ_{k-1}^2, $P(\Theta_{k-1}^2)$

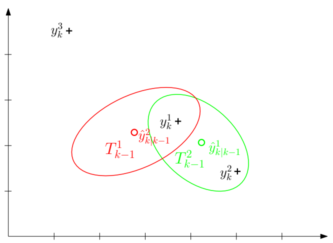


Instead of keeping Θ_{k-1}^1 and Θ_{k-1}^2 which would duplicate information of T_1, we can save T_1 and T_2 for the next time instant.

Track Based Implementation

- Tracks at time k are shown by $\{T_k^i\}_{i=1}^{N_t}$
- Each track is kept with its Score shown as $Sc(T_k^i)$.
- Instead of a hypotheses tree, form a track tree.
- Delete low score tracks.

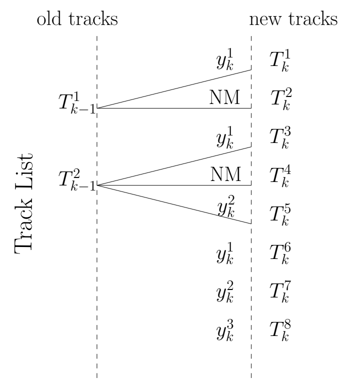




Track Based Implementation

- For reducing the number of tracks further and for user presentation, generation of hypotheses is still necessary. One advantage this time is that one can only use high score tracks for hypothesis generation.
- For generating hypotheses, keeping the track compatibility information is necessary. One can keep a binary matrix as below.

	T_k^1	T_k^2	T_k^3	T_k^4	T_k^5	T_k^6	T_k^7	T_k^8
T_k^1	0	0	0	1	1	0	1	1
T_k^2		0	1	1	1	1	1	1
T_k^3			0	0	0	0	1	1
T_k^4				0	0	1	1	1
T_k^5					0	1	0	1
T_k^6						0	1	1
T_k^7							0	1
T_k^8								0



Track Based Implementation

Hypothesis Generation: An hypothesis is basically a collection of compatible tracks.
Examples: $\Theta_k^1 = \{T_k^1, T_k^5, T_k^8\}$, $\Theta_k^2 = \{T_k^2, T_k^3, T_k^7, T_k^8\}$

Score of an Hypothesis

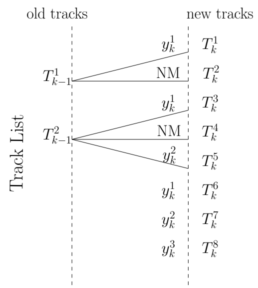
$$Sc(\Theta_k^i) = \sum_{T_k^j \in \Theta_k^i} Sc(T_k^j)$$

Probability of a Track

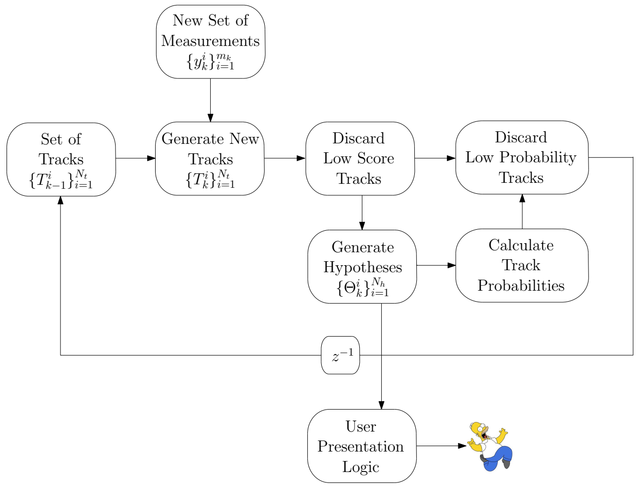
$$P(T_k^i) = \sum_{\Theta_k^j \ni T_k^i} P(\Theta_k^j)$$

Probability of an Hypothesis

$$P(\Theta_k^i) = \frac{\exp(Sc(\Theta_k^i))}{1 + \sum_{j=1}^{N_h} \exp(Sc(\Theta_k^j))}$$



Track Based Implementation



Track Based Implementation

For keeping the number of tracks and hypothesis generation computations under control

- Clustering incompatible tracks into clusters can facilitate hypothesis generation
- N-scan pruning can be applied to track trees (instead of hypothesis trees in the previous case) by keeping histories of the tracks in memory.
- Merging the tracks that have the same recent measurement history is another idea to reduce the number of tracks.

User Presentation Logic

- The simplest method is to show the user the maximum probability hypothesis.
- However, this can be a little jumpy because the maximum probability hypothesis can change quite erratically.
- Another method is to show track clusters with their overall (weighted) mean, covariance and expected number of targets in them.
- Another idea is to keep a separate track list which, at each step, is updated with a selection of tracks from different hypotheses.
- Chapter 16 of the textbook gives extensive details about track based implementation of MHT and user presentation logic. Consult it for further details.

General: Which Multi TT Method to Use?

Computation \ SNR	Low	Normal	High
Low	Group TT or PHD	GNN	GNN
Normal	MHT	GNN or JPDA	GNN
High	TBD or MHT	MHT	Any

- GNN and JPDA are very bad in low SNR.
- When using GNN, one generally has to enlarge the overconfident covariances to account for neglected data association uncertainty.
- JPDA has track coalescence and should not be used with closely spaced targets, see the "coalescence avoiding" versions.
- MHT requires significantly higher computational load but it is said to be able to work reasonably under 10-100 times worse SNR.

References

- I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.18, no.2, pp.138–150, Feb. 1996.
- R. Danchick and G. E. Newnam, "Reformulating Reid's MHT method with generalised Murty K-best ranked linear assignment algorithm," *IEE Proceedings – Radar, Sonar and Navigation*, vol.153, no.1, pp. 13–22, 16 Feb. 2006.
- S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol.19, no.1, pp.5–18, Jan. 2004.
- S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

References

- I. J. Cox and M. L. Miller, "On finding ranked assignments with application to multitarget tracking and motion correspondence," *IEEE Transactions on Aerospace and Electronic Systems*, vol.31, no.1, pp.486–489, Jan. 1995.
- I. J. Cox, M. L. Miller, R. Danchick and G. E. Newnam, "A comparison of two algorithms for determining ranked assignments with application to multitarget tracking and motion correspondence," *IEEE Transactions on Aerospace and Electronic Systems*, vol.33, no.1, pp.295–301, Jan. 1997.
- M. L. Miller, H. S. Stone and I. J. Cox, "Optimizing Murty's ranked assignment method," *IEEE Transactions on Aerospace and Electronic Systems*, vol.33, no.3, pp.851–862, Jul. 1997.
- D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol.24, no.6, pp. 843–854, Dec. 1979.