Issues and Approaches for Data Fusion

Peter Willett
University of Connecticut

Outline

- A Communications Perspective on Data Fusion
  - multi-user information theory
- Data Fusion for Detection & Estimation
  - scan statistics for sensor networks
  - consensus in sensor networks
  - data fusion with intermittent detections
  - quantized estimation: a note
  - decentralized learning
  - decentralized estimation with MOU
- Data Fusion for Tracking
  - architectures
  - bias
  - track fusion
- Mapping a "Soft" Problem to "Hard" Terms
  - an example

Multi-Sensor Information Theory

- What are the bounds?
- Basic IT motivated by "typicality"
  - Source coding, channel capacity, rate-distortion theory
- Capacity for networks
  - Multi-Access Channel (MAC)
  - Broadcast Channel
  - General Networks
- Distributed Coding
  - Noisy and Noiseless
- Distributed Inference (CEO Problem)

- Consider a discrete iid source $\{X_i\}$ with probabilities $p_j=Pr(X_i=x_j)$
- Suppose we have $X^n=\{X_1,X_2, \ldots , X_n\}$
  - On average there will be np_1 x_1's, np_2 x_2's, etc.: Pr("ABBA")=$p_A^2 p_B^2$
  - then

$$\Pr(X^n) \approx \prod_{j=1}^{m} p_j^{np_j} = 2^{n\sum_{j=1}^{n} p_j \log(p_j)} = 2^{-nH(X)} \quad where \quad H(X) = \sum_{j=1}^{n} p_j \log(1/p_j)$$

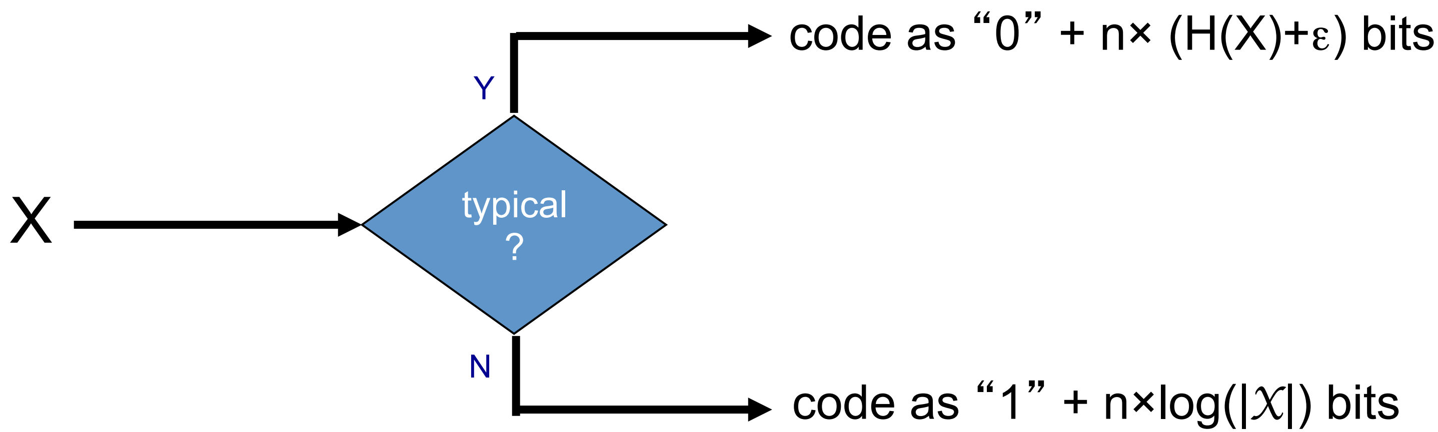
- Typical X^n is one for which

$$H(X) - \varepsilon \le \frac{-1}{n}\log\left(p(X^n)\right) = \frac{-1}{n}\sum_{i=1}^{n}\log\left(p(X_i)\right) \le H(X) + \varepsilon$$

  - LLN says probability that X^n is typical is $1-\varepsilon$, small ε as you like
  - "Only typical X's ever happen."
  - Typical set has $2^{nH(X)}$ elements, each with probability $2^{-nH(X)}$

Entropy

- Source coding scheme:

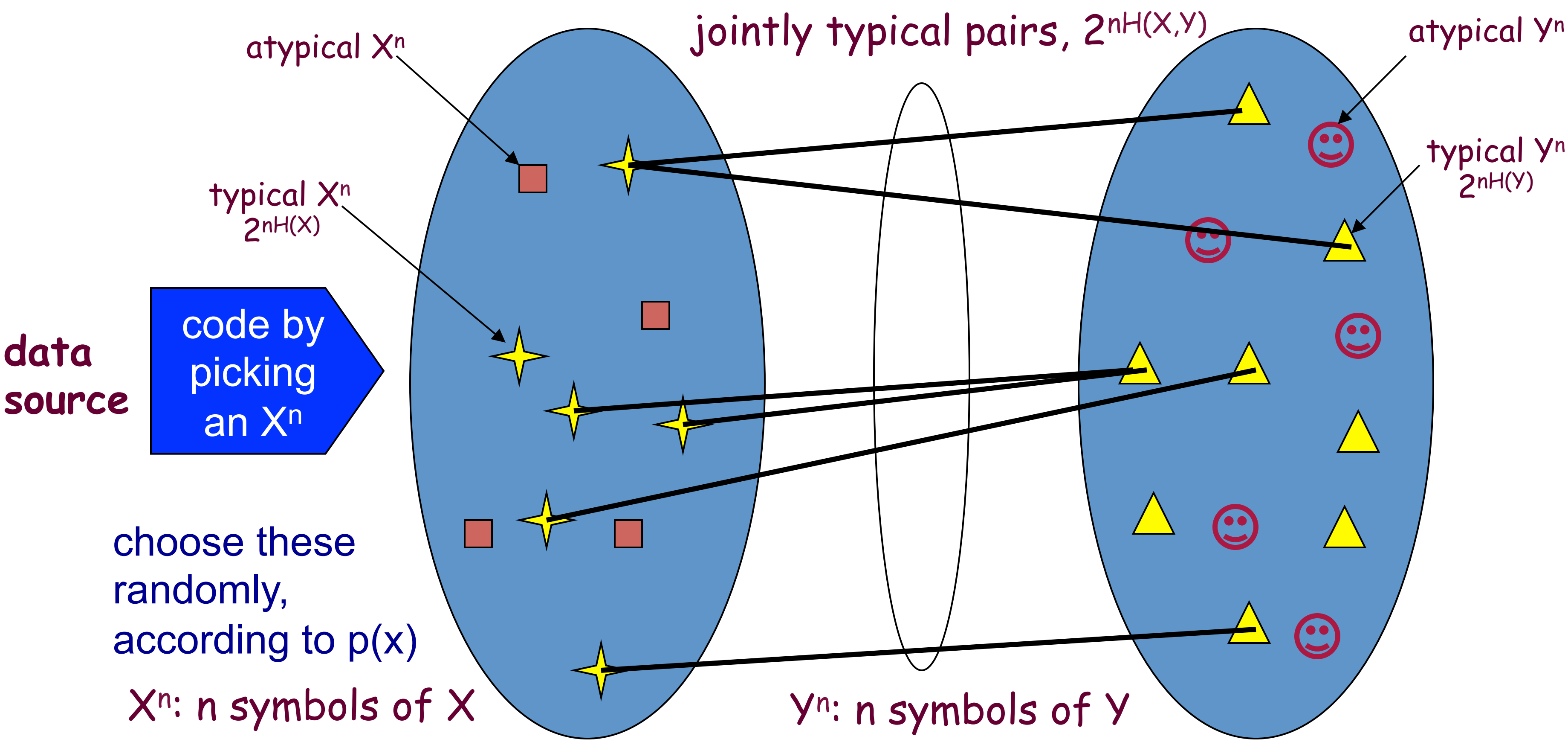


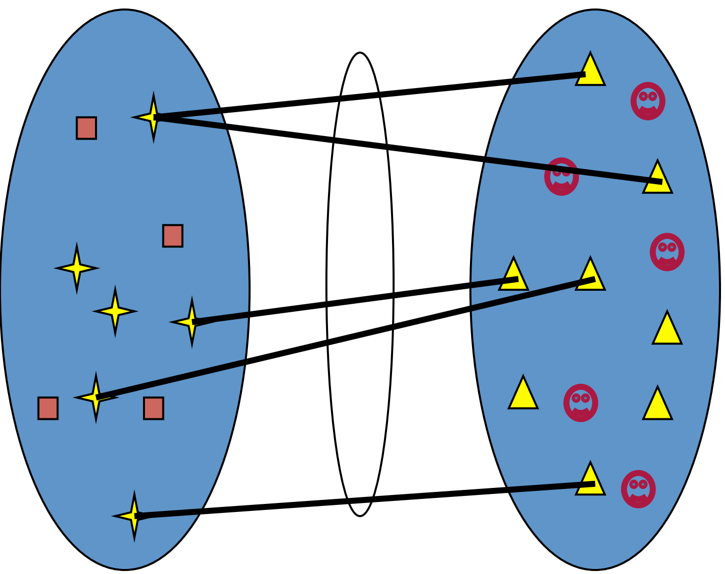
- Code length per source symbol is

$$\bar{L} = \frac{1}{n}\left[(1-\varepsilon)\left(1+nH(X)+n\varepsilon\right)\right]+\varepsilon\left(1+nm\right) \approx H(X)$$

Information

- Information is $I(X;Y) = H(X)-H(X|Y)$
- Communication channel:





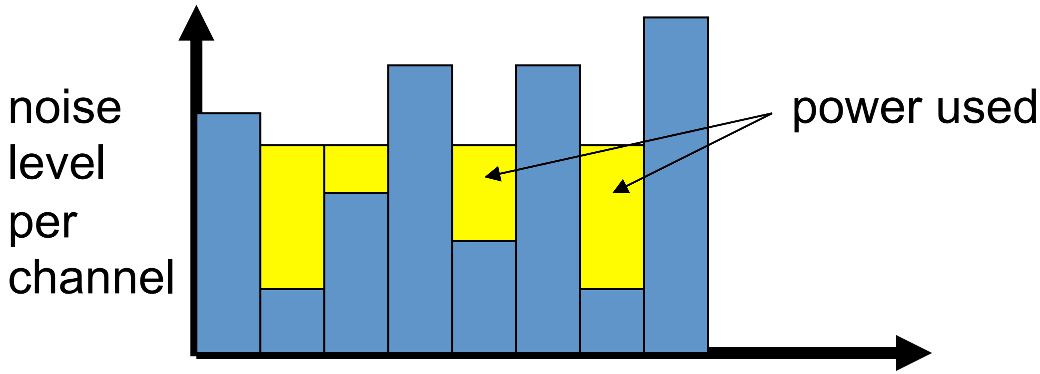
- number of typical X^n' s: $2^{nH(X)}$
- number of typical Y^n' s: $2^{nH(Y)}$
- number of jointly-typical (X^n,Y^n) pairs: $2^{nH(X,Y)}$
- code procedure:
  - look at our Y^n, and if jointly-typical with exactly one X^n, then we decode to that, otherwise error
- but X^n each is joined to $2^{n(H(X,Y)-H(Y))}$ Y^n' s
- so use only a fraction $2^{n(H(Y)-H(X,Y))}$ of the $2^{nH(X)}$ available X^n codewords
- then we have left $2^{nH(X)}2^{n(H(Y)-H(X,Y))} = 2^{nI(X;Y)}$ typical X^n codewords left
- $I(X;Y)=H(X)+H(Y)-H(X,Y)$ defines the rate we can send data
- this information is the "capacity"

Capacity

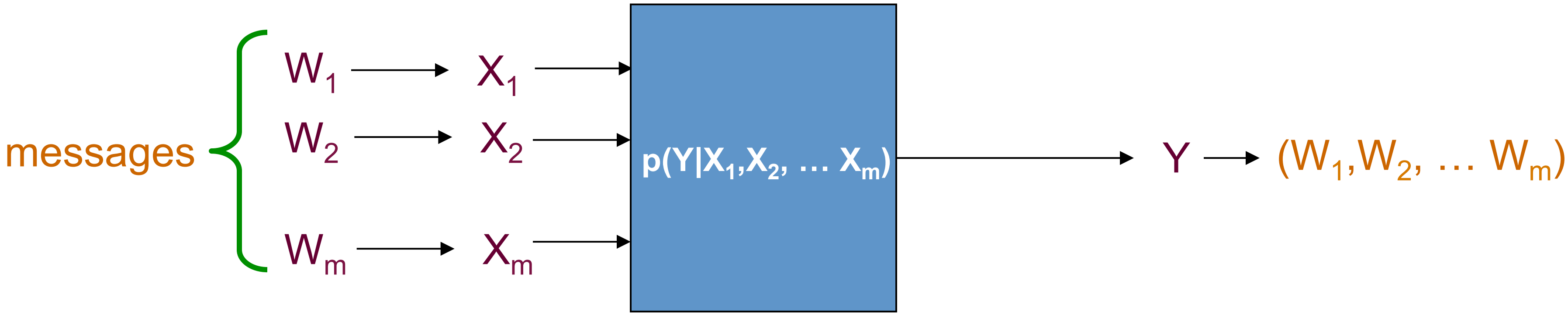
- capacity: $C=\max_{p(x)}\{I(X;Y)\}$
  - means that you choose a code to match the channel
- in the Gaussian case

  $$C=B*\log(1+P/N_0B)$$

  where B is the bandwidth and P is the transmitted power
- parallel **cooperative** Gaussian channels: water-filling



Multi-Access Channel (MAC)

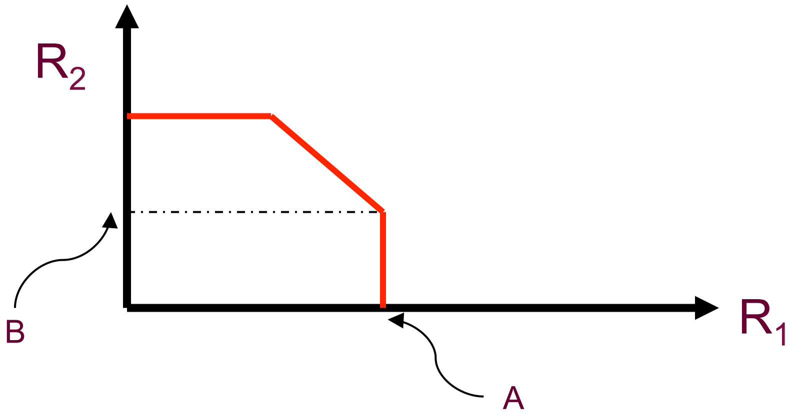


$$R(S) \leq I\left(X(S);Y \mid X(S^c)\right) \quad \forall S$$

where

- S is a subset of the users {1,2, … ,m} and S^c is its complement
- R(S) is the sum of the rates of the users in S
- the information uses a product (independent) distribution of X(S)
- this is exact region, not a bound on the region

MAC Region for Two Users

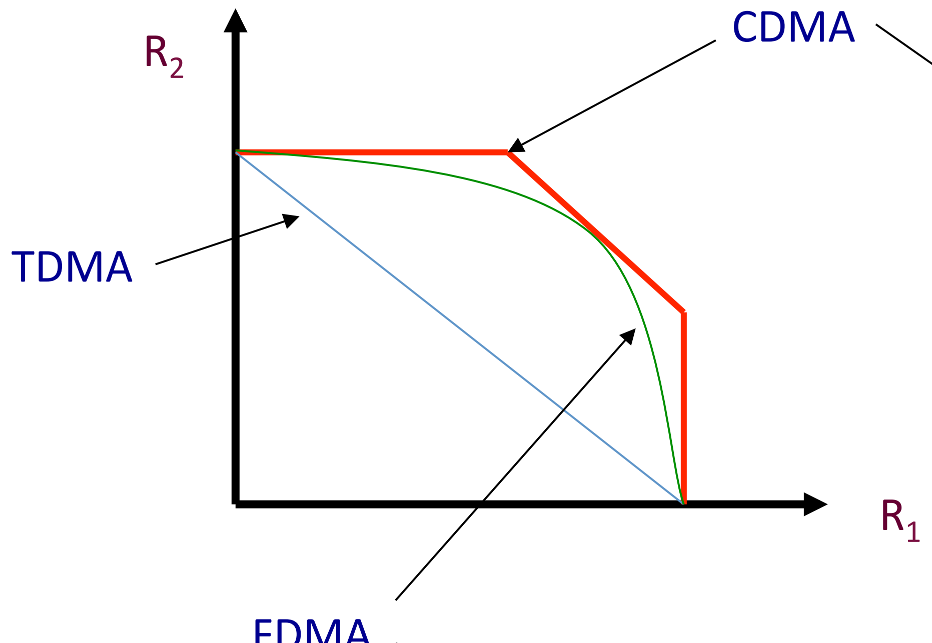


$$R_1 \leq I(X_1;Y \mid X_2)$$
$$R_2 \leq I(X_2;Y \mid X_1)$$
$$R_1 + R_2 \leq I(X_1,X_2;Y)$$

- source 2 starts at R_2~0
  - source 2 is easy to decode, so X_2 is known
  - then source 1 can transmit at $A=I(X_1;Y|X_2)$
- now source 2 increases its rate up to B
  - source 2 can still be decoded (first) while $R_2<I(X_2;Y)$
  - up to that point X_1 is just "noise"

$$\begin{aligned} B &= I(X_1,X_2;Y) - A \\ &= I(X_1,X_2;Y) - I(X_1;Y|X_2) \\ &= H(Y) - H(Y|X_1,X_2) - \left(H(Y|X_2) - H(Y|X_1,X_2)\right) \\ &= H(Y) - H(Y|X_2) \\ &= I(X_2;Y) \end{aligned}$$

Gaussian MAC



$$R_1 \le I(X_1;Y \mid X_2) = \tfrac{1}{2}\log\left(1+\frac{P_1}{N}\right)$$

$$R_2 \le I(X_2;Y \mid X_1) = \tfrac{1}{2}\log\left(1+\frac{P_2}{N}\right)$$

$$R_1 + R_2 \le I(X_1,X_2;Y) = \tfrac{1}{2}\log\left(1+\frac{P_1+P_2}{N}\right)$$

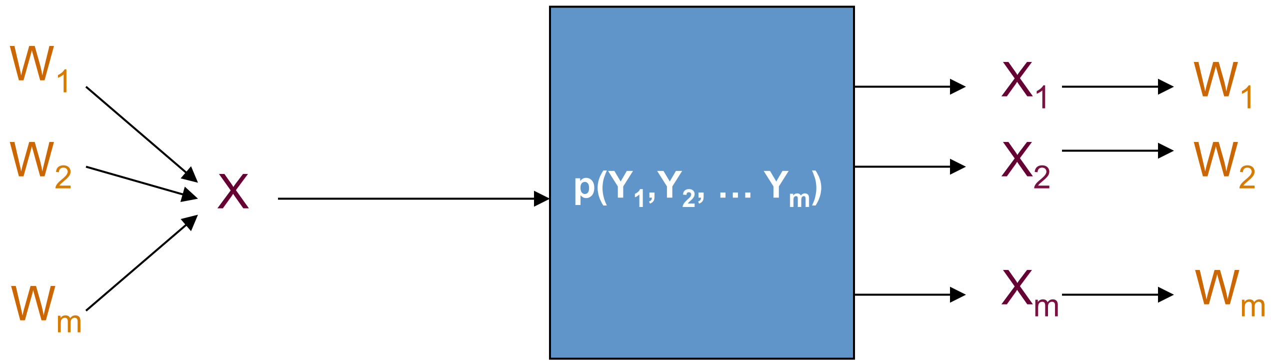
$$R_1 \le \frac{\alpha}{2}\log\left(1+\frac{P_1}{\alpha N}\right)$$

$$R_2 \le \frac{(1-\alpha)}{2}\log\left(1+\frac{P_2}{(1-\alpha)N}\right)$$

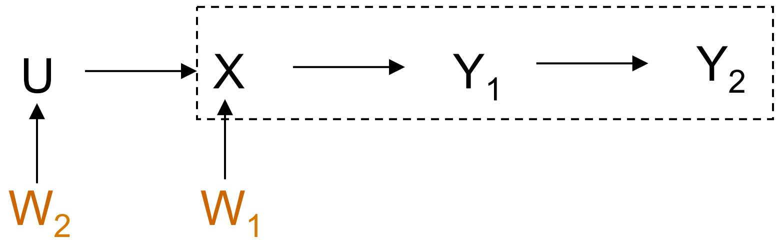
$$0 \le \alpha \le 1$$

frequency division multi-access is actually not so bad

Broadcast Channel

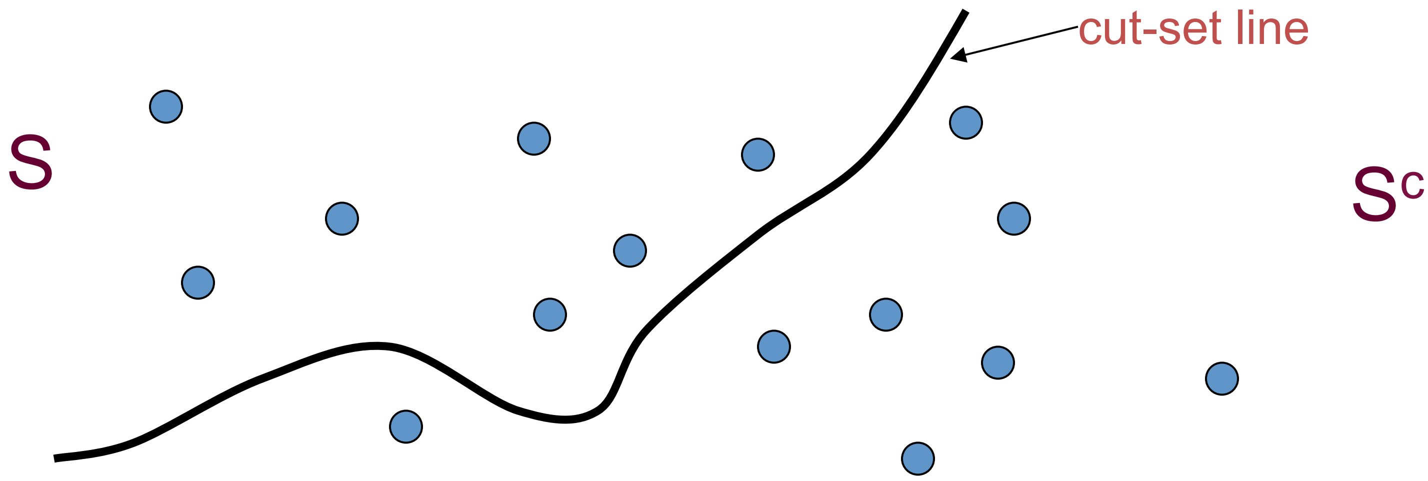


- exact bound not known in general, but "degraded broadcast channel" is:



- if Y_2 can decode W_2, so can Y_1: $R_2<I(U;Y_2)<I(U;Y_1)$
- then $R_2<I(U;Y_2)<I(U;Y_1)$, so U is known at Y_1
- if U is demodulated, then $R_1<I(X;Y_1|U)$

General Networks

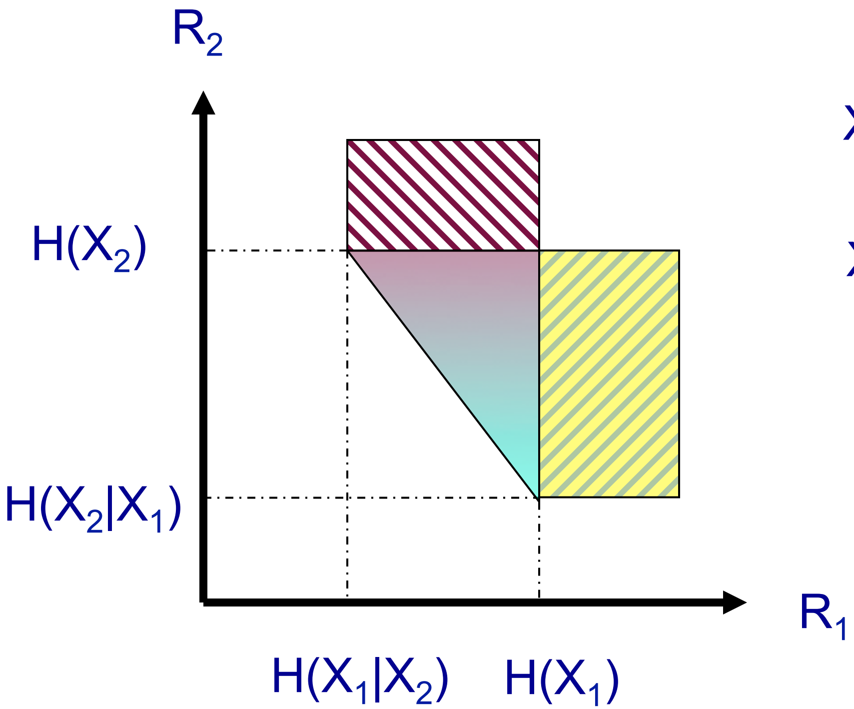


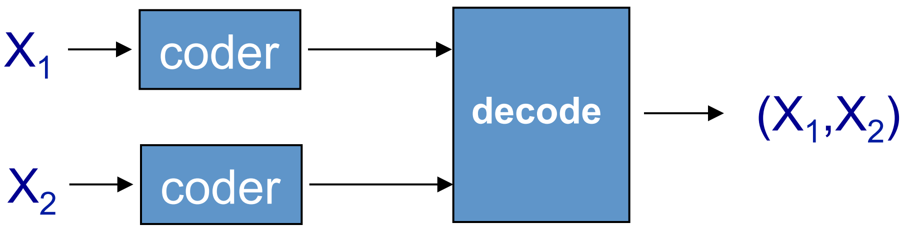
$$\sum_{i \in S,\, j \in S^c} R_{i \to j} \le I\left(X(S); Y(S^c) \mid X(S^c)\right)$$

This is an "outer bound," not in general tight for achievable region.

The Slepian-Wolf Problem

- distributed noiseless source coding [1973]
  - for dependent sources
  - one source can help the other reduce its rate

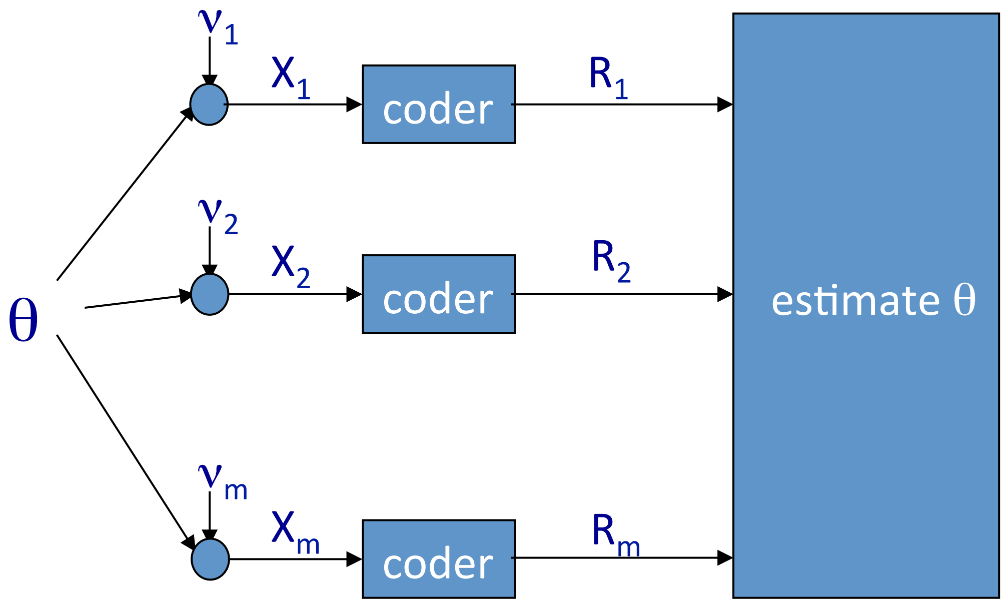




- clearly $R_1>H(X_1)$ & $R_2>H(X_2|X_1)$
- clearly $R_2>H(X_2)$ & $R_1>H(X_1|X_2)$
- triangular region is filled in by time-sharing
- $R_1+R_2>H(X_1,X_2)$

Distributed Inference: The CEO Problem

- Berger, Zhang & Viswanathan [1996]
- Viswanathan & Berger [1997]
- Oohama [1998]
- Zamir & Berger [1999]
- Chen, Zhang, Berger & Wicker [2004]
- Prabhakaran, Tse & Ramchandran [2004]



- everything is Gaussian
- everything is independent
- we have an MSE criterion on q

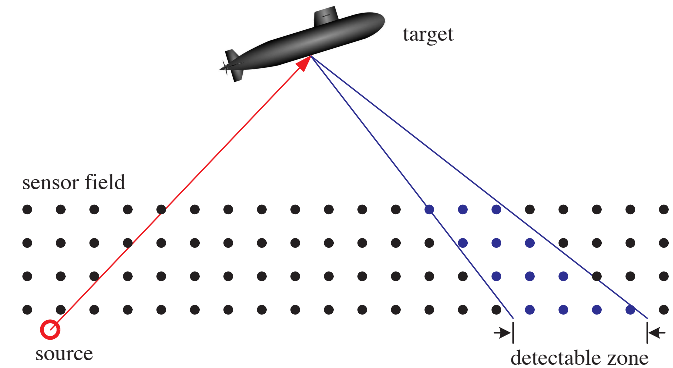
$$R(D) = \frac{1}{2}\log^{+}\left(\frac{\sigma_\theta^2}{D}\left[\frac{D\sigma_\theta^2 N}{D\sigma_\theta^2 N - \sigma_\theta^2\sigma^2 + D\sigma^2}\right]\right)$$

- there are vector versions of this
- there are "successive refinement" versions of this
  - I have not seen a Kalman filter involved
- I am not aware of a data-association version

Data Fusion for Decision-Making and Estimation: Some Topics

- Scan statistics for sensor networks
- Consensus in sensor networks
- Data fusion with intermittent detections
- Quantized estimation: a note
- Decentralized learning
- Decentralized estimation with MOU

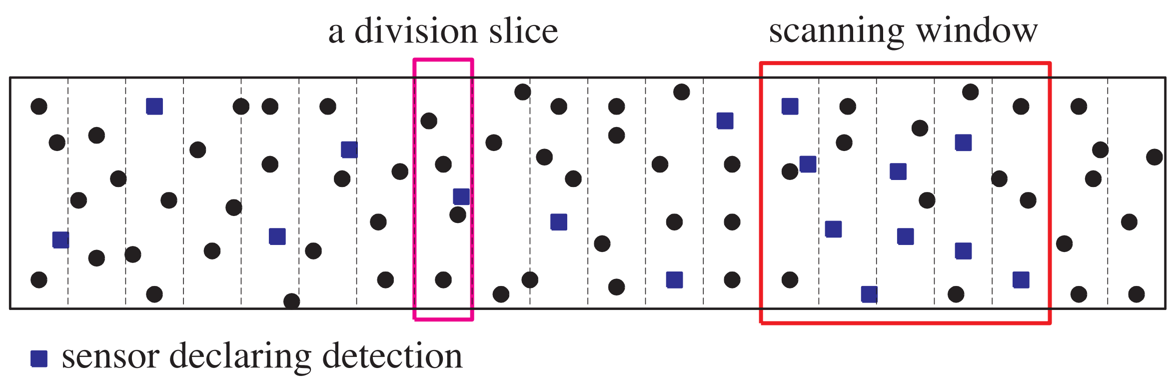
Scan Statistics for Sensor Networks

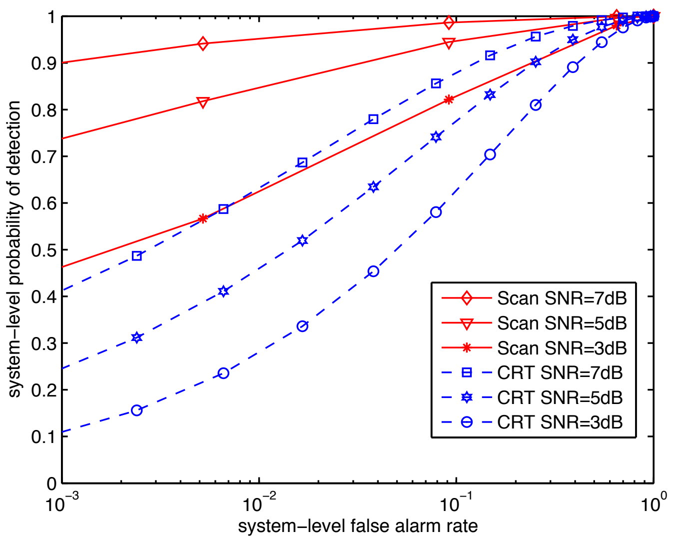


- Barrier sensor network: a narrow but long sensor band along coastline.
  - How to effectively fuse the binary local decisions in the fusion center?
- Angle dependent reflection
  - Only a small area of sensors can reliably detect

- Song, Willett, Glaz & Zhou, “Active Detection With A Barrier Sensor Network Using A Scan Statistic,” JOE 2012.
- Glaz, Guerriero & Sen, “Approximations for a three dimensional scan statistic,” J. Comp. in App. Prob., 2009.

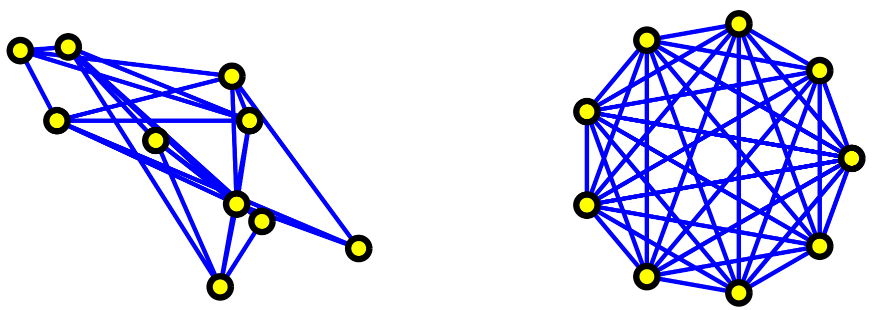
Poisson Field





- Scan statistics have broad application:
  - epidemiology
  - ecology
  - quality control and reliability
  - intrusion detection
- The key ingredient to scan statistics is that the threshold can be set analytically and explicitly.
  - admittedly, the formula is complicated

Consensus in Sensor Networks



- Consider peer-to-peer communication
  - as opposed to "parallel" or "serial" topology
- Each sensor has its own observation and sends its information to its "neighbors" defined by the graph

$$\mathbf{s}_0 = \mathbf{z} \quad \rightarrow \quad \mathbf{s}_n = \mathbf{W}\mathbf{s}_{n-1}$$

- Require obvious condition on eigenvalues of W and that it be doubly-stochastic

- Braca, Marano, Matta & Willett, "Consensus-Based Page's Test in Sensor Networks," Sig. Proc. 2009.

Consensus for Quickest Detection

- Consider at all sensors *j* a switch in distribution:

$$f_0(x) : x_{1,j}, x_{2,j}, \ldots, x_{n_0-1,j} \searrow$$

$$f_1(x) : \qquad\qquad x_{n_0,j}, x_{n_0+1,j}, \ldots$$

- Require

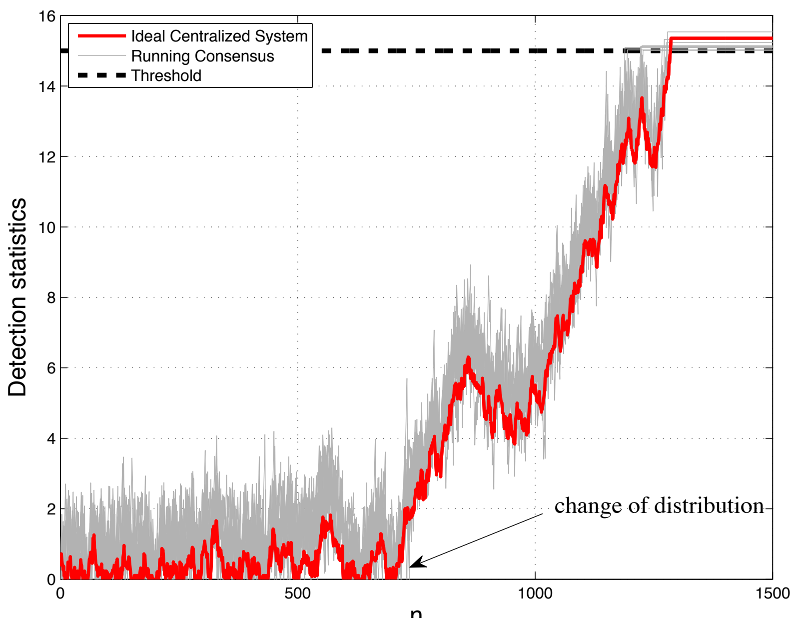
$$\begin{pmatrix} S_{n,1} \\ S_{n,2} \\ \vdots \\ S_{n,M} \end{pmatrix} = \mathbf{W}_n \begin{pmatrix} S_{n-1,1} \\ S_{n-1,2} \\ \vdots \\ S_{n-1,M} \end{pmatrix} + M\,\mathbf{W}_n \begin{pmatrix} \log \frac{f_1(x_{n,1})}{f_0(x_{n,1})} \\ \log \frac{f_1(x_{n,2})}{f_0(x_{n,2})} \\ \vdots \\ \log \frac{f_1(x_{n,M})}{f_0(x_{n,M})} \end{pmatrix}$$

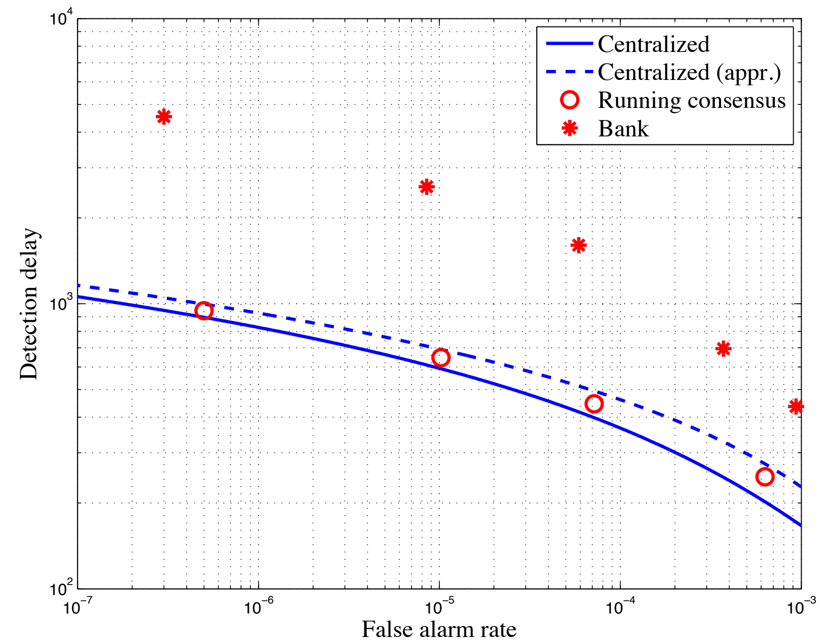
- For example, pair-wise averaging

$$\mathbf{W}_n = \mathbf{I} - \frac{(\mathbf{u}_k - \mathbf{u}_h)(\mathbf{u}_k - \mathbf{u}_h)^T}{2}$$

Example

- Change from N(0,1) to N(0,1.032), 10 sensors

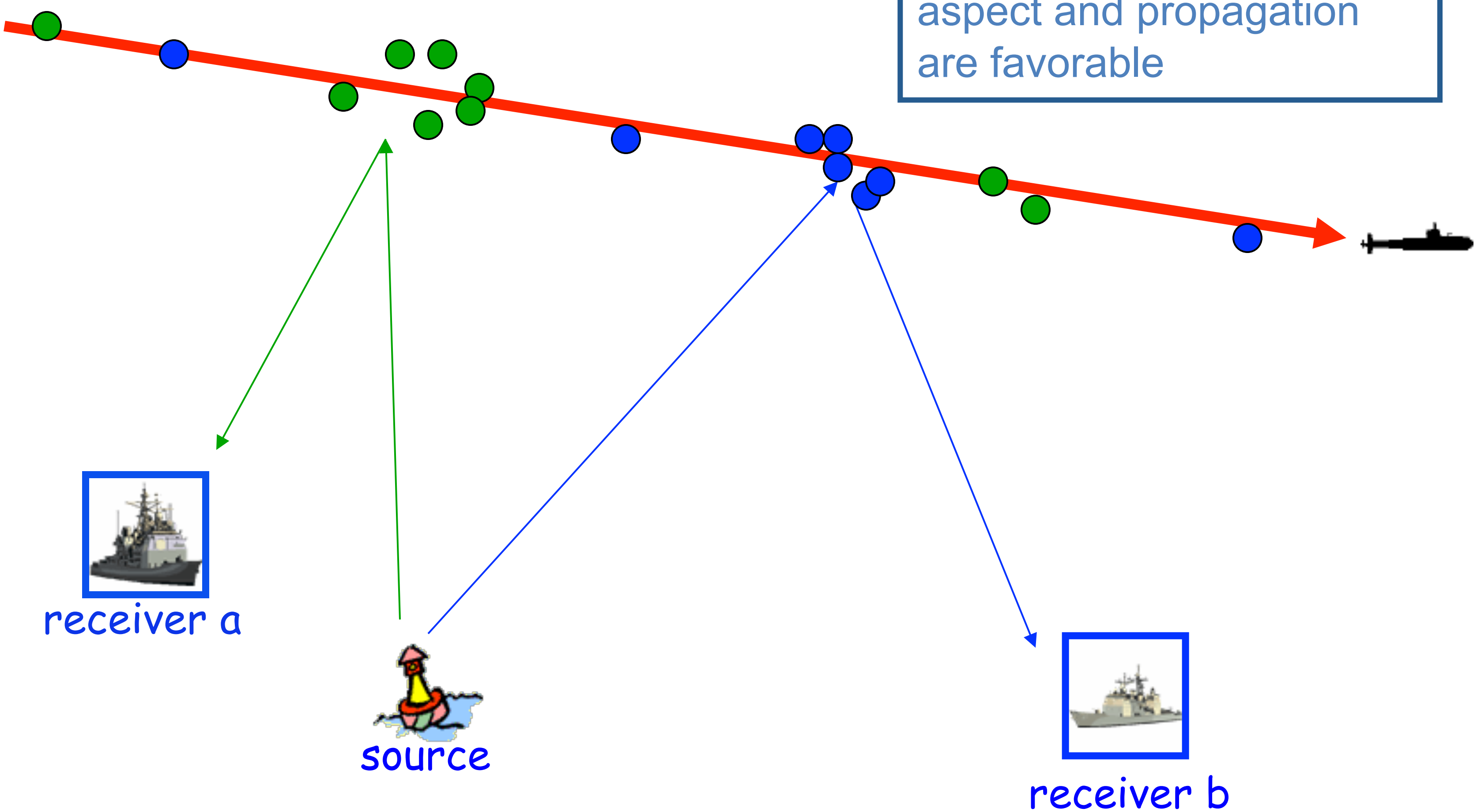




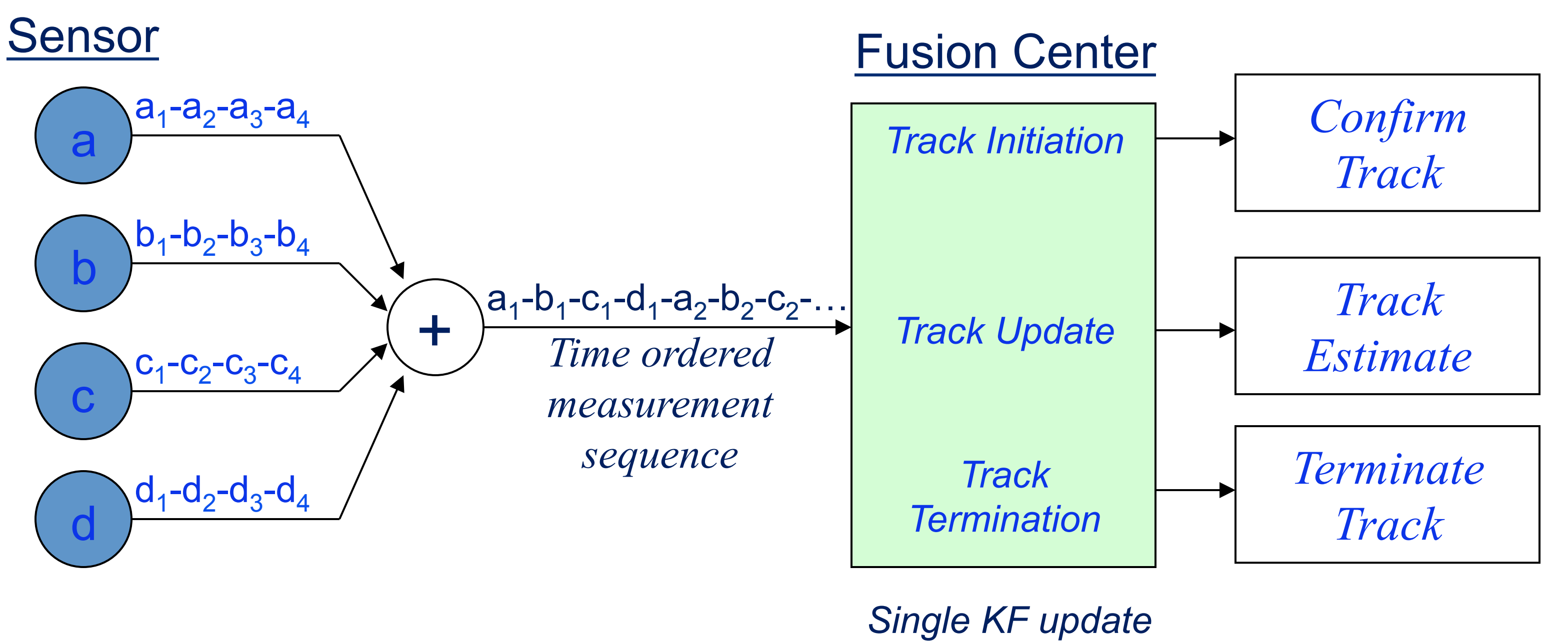
- Consensus / Page is asymptotically optimal (compared to centralized) and much better than an OR rule (bank of Page tests)

Intermittent Detections

hits clustered when target aspect and propagation are favorable



Track Management Architecture



Track Termination

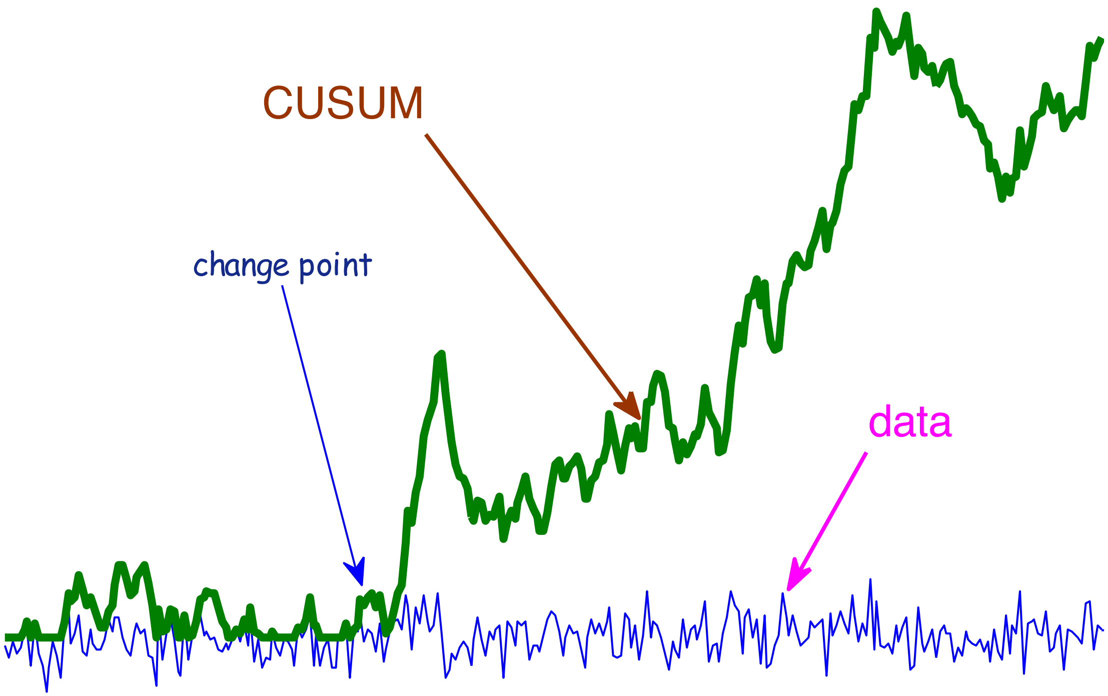
- Goal: quickest detection of change in measurement distribution from a true track (H_1) to a false track (H_0)

- Page test
  - Proven global optimality for i.i.d. case and some Markov models
  - Some recent asymptotic optimality results for HMMs (Fuh 2003), *not applicable for this case*
  - A sequential test that minimizes delay in detection of a distribution change at a given false alarm rate

$$s_k = \ln \frac{\Pr\{\delta_k \mid \delta_1^{k-1}, H_0\}}{\Pr\{\delta_k \mid \delta_1^{k-1}, H_1\}} \qquad c_k = \max(c_{k-1} + s_k, 0)$$

- Blanding, Willett, Coraluppi & Bar-Shalom, "Multisensor Track Management for Targets with Fluctuating SNR," TAES 2009.

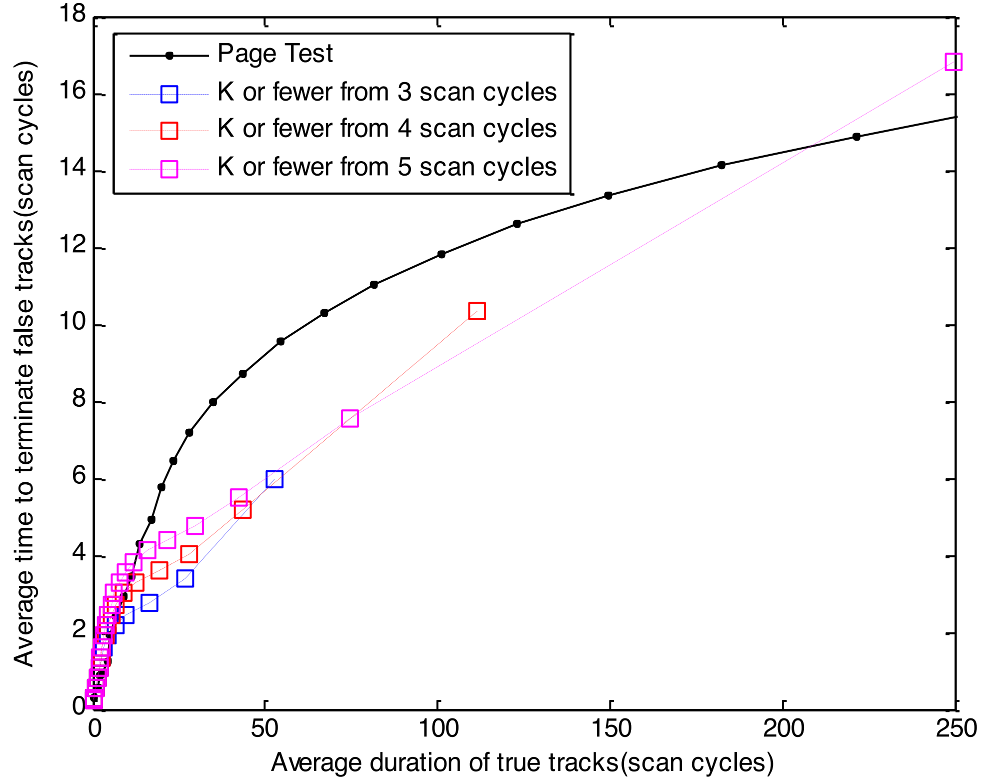
- Page test example:
  - unit Gaussian with mean +/- 0.2
  - can you see it?



Page Test

- Simulation methodology:
  - Track termination tests begin on first measurement after track confirmation
  - 10^4 simulations under H_1
  - 10^4 simulations under H_0

- Surprising result:
  - Page test is *not* globally optimal
  - LLR innovations are not i.i.d.



Track termination performance (4 sensors)

Shiryaev Test

- Optimum quickest detection when the problem is formulated using a Bayesian approach (in the i.i.d. case)
  - *a priori* probability of change time k_c:

$$\Pr\{k_c = k\} = \begin{cases} \pi_0 & k = 0 \\ (1-\pi_0)\rho(1-\rho)^{k-1} & k > 0 \end{cases}$$

  - Using Bayes rule, *a posteriori* change probability:

$$\pi_k = \frac{[\pi_{k-1} + (1-\pi_{k-1})\rho]\Pr\{\delta_k \mid \delta_1^{k-1}, H_0\}}{[\pi_{k-1} + (1-\pi_{k-1})\rho]\Pr\{\delta_k \mid \delta_1^{k-1}, H_0\} + (1-\pi_{k-1})(1-\rho)\Pr\{\delta_k \mid \delta_1^{k-1}, H_1\}}$$

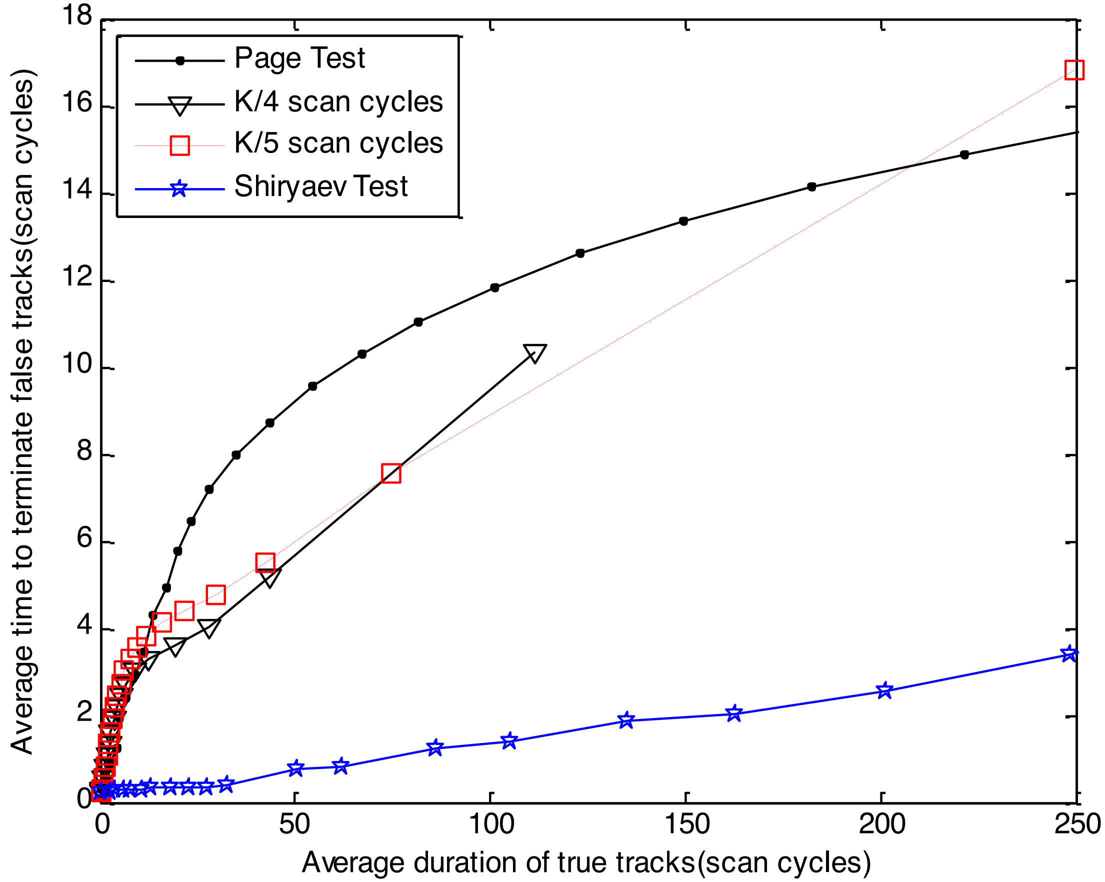
  - The Shiryaev stopping rule becomes:

$$g_k = \ln\frac{\pi_k}{1-\pi_k}$$

$$g_k = \ln(\rho + e^{g_{k-1}}) - \ln(1-\rho) + \ln\frac{\Pr\{\delta_k \mid \delta_1^{k-1}, H_0\}}{\Pr\{\delta_k \mid \delta_1^{k-1}, H_1\}}$$

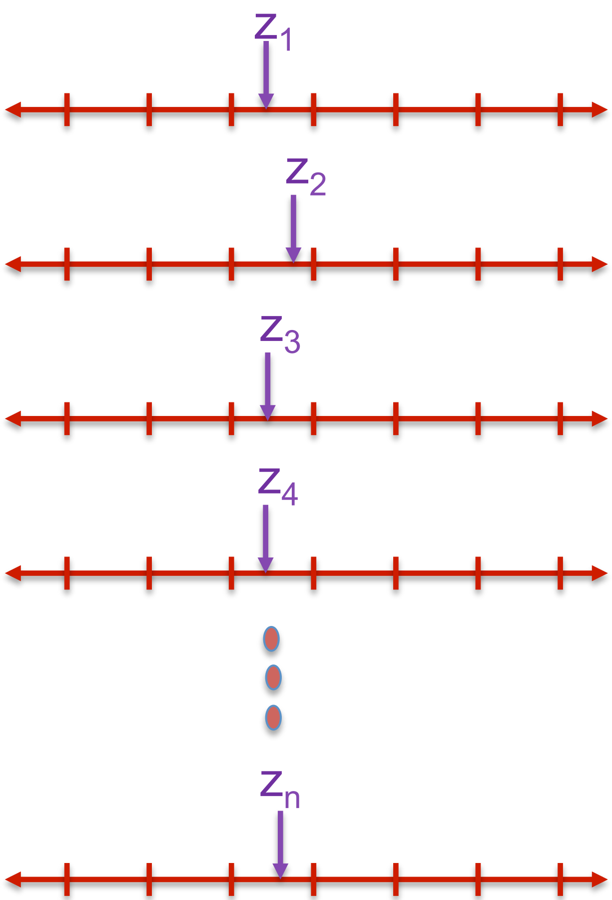
Track Termination Tests

- Sequential tests
  - Page test
  - Shiryaev test
- Rule-Based
  - K/N rule
- Conclusion:
  - Shiryaev test performs best

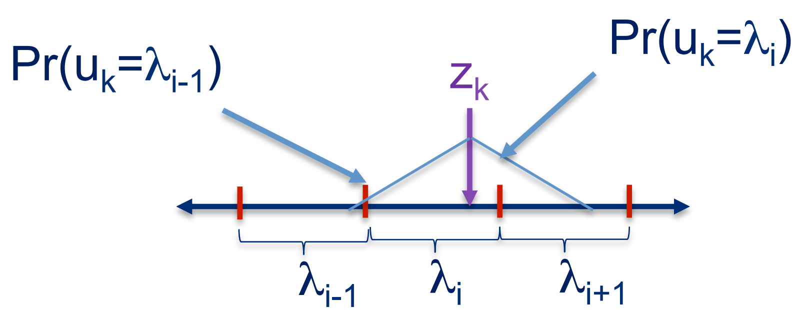


Comparison of average track duration for different track termination rules

A Note on Quantized Estimation



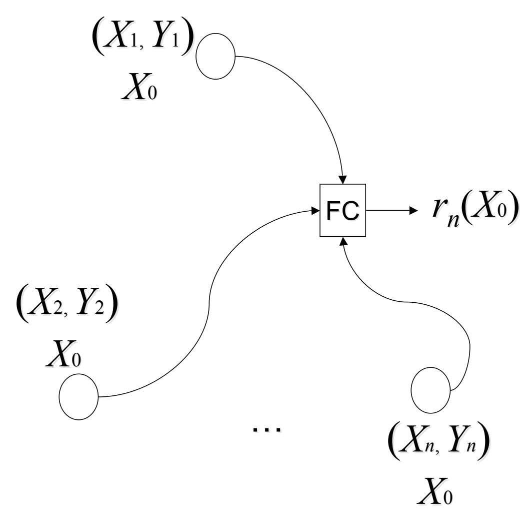
- It is fairly clear that the estimation performance here is limited by the quantization fineness and does not improve beyond a certain point with n.
- Paradoxically, the lower the sensor noise the worse this behavior is.
- Luo's solution is to use a randomized quantizer.



- Luo, "Universal Decrentalized Estimation in a Bandwidth constrained Sensor Network," T-IT 2005.

Decentralized Learning

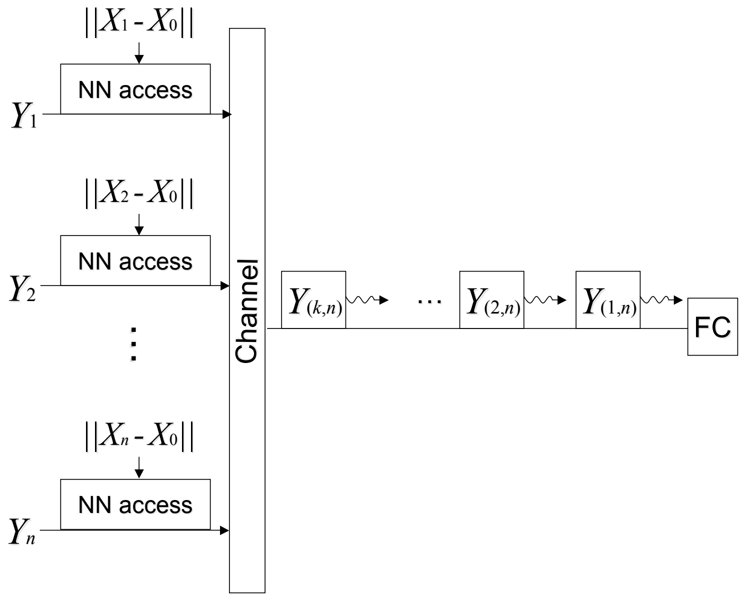
- Suppose one has a repository of training data and a collection of local agents.
- As opposed to our usual decision-making based on distributed observations, let us here assume that all decision-makers observe the same datum but that the "database" of training data is distributed.
- Explore extreme case: each sensor has only only training datum.
- Application here is regression.

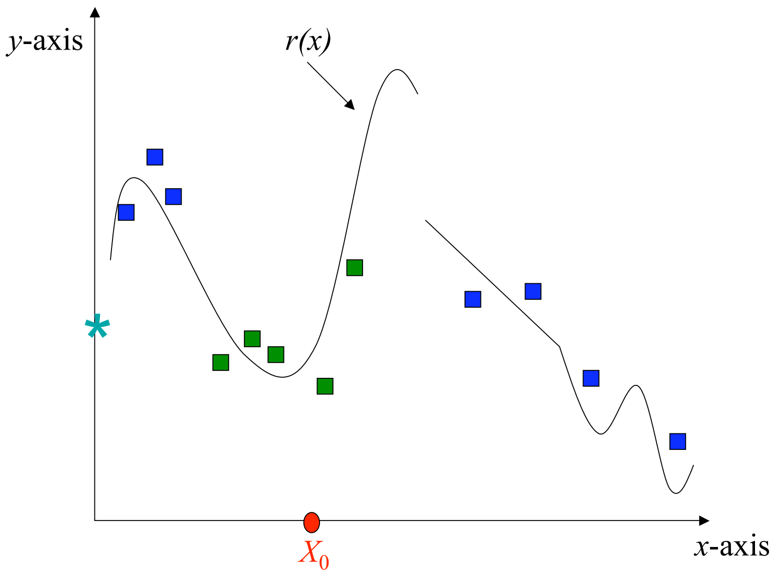


Decentralized NN Learning

- Nearest-neighbor learning gets ignored sometimes.
  - it can be shown that the NN *decision* is (asymptotically) no worse than twice the optimum, $2P^*(e)$
  - with k-NN we have P(e) goes to $(1+1/k)P^*(e)$
- There is a similar suite of results with NN regression.
  - MMSE asymptotically no worse than twice $MMSE^*$
- How do we achieve this?
  - transmission rule like with censoring
  - sensor i transmits after delay proportional to $\alpha_n d(X,X_i)$
  - as long as α_n is proportional to $k_n n$ this works

Blum & Sadler's Access Rule

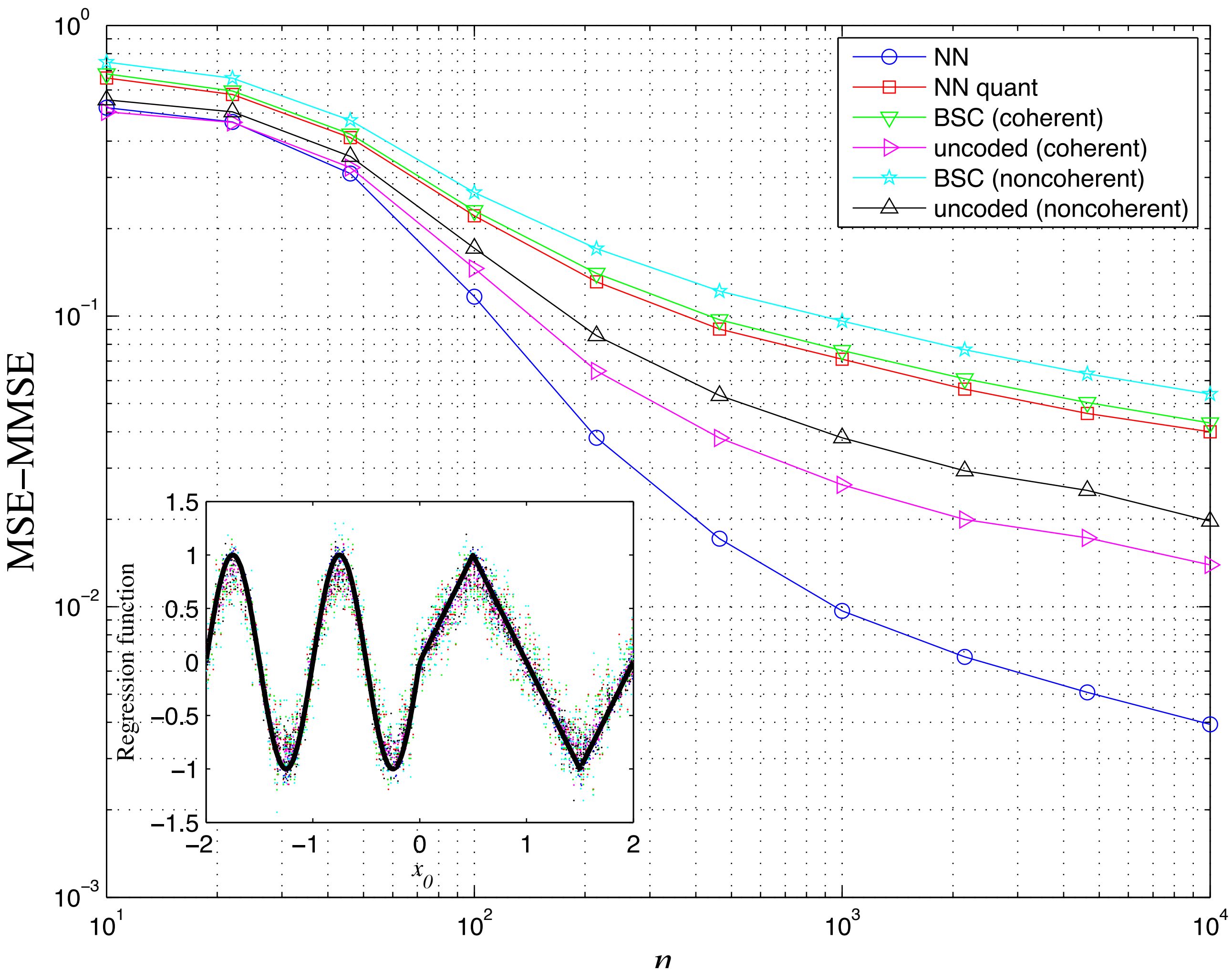




First 5 neighbors

When k_n agents have been heard from, FC sends broadcast to stop.

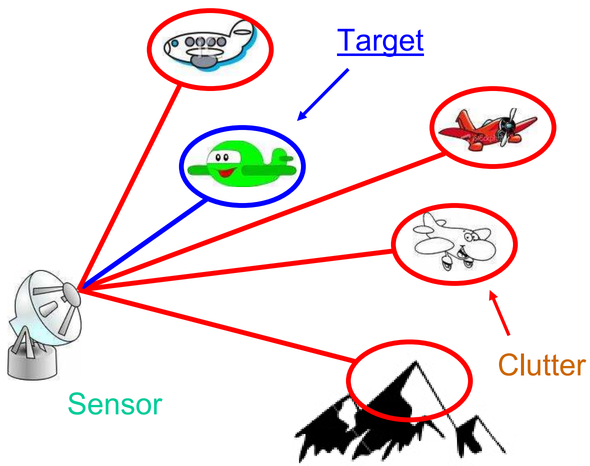
- Blum & Sadler, "Energy efficient signal detection in sensor networks using ordered transmissions," TSP 2008.



- The various lines here refer to different schemes to communicate the agents' regression data to the FC.
- The number of sensors is n.

- Marano, Matta & Willett, "Nearest-neighbor distributed learning by ordered transmissions," TSP 2013.

Decentralized Estimation with MOU



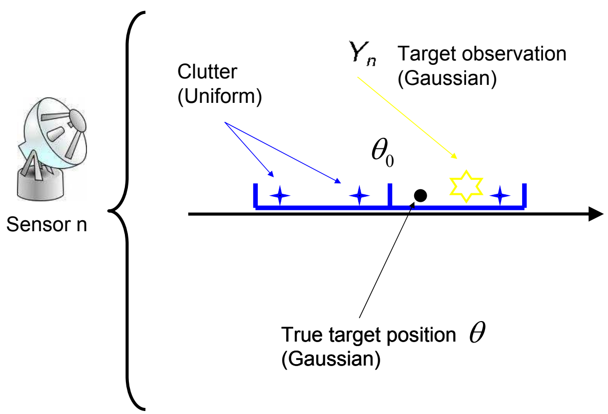
- bandwidth constraint:
  - sensors each transmit one measurement
  - which is the most informative?
- here we discuss k-mos
  - "modulus order statistic"
  - transmit the k^{th}-nearest measurement to where the target is expected to be

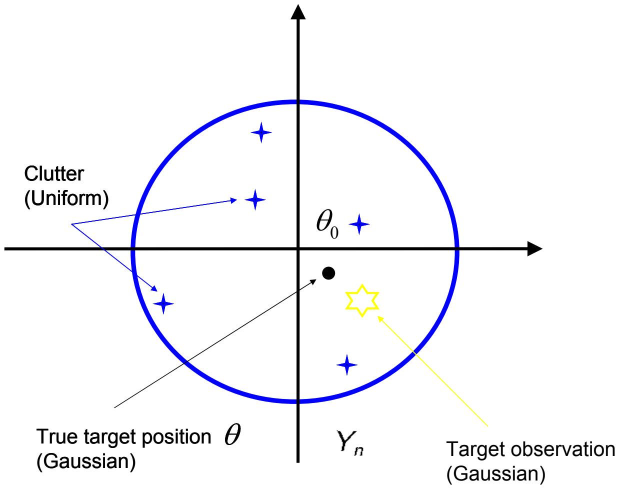
- Braca, Guerriero, Marano, Matta & Willett, "Selective Measurement Transmission in Distributed Estimation with DA," TSP 2010.

Data association likelihood for frame $Z=\{z_1,z_2, \ldots ,z_n\}$, probability of detection P_d, clutter intensity λ, observation volume V and likelihood model $p(z|\theta)$ (this is commonly a Gaussian pdf centered at θ).

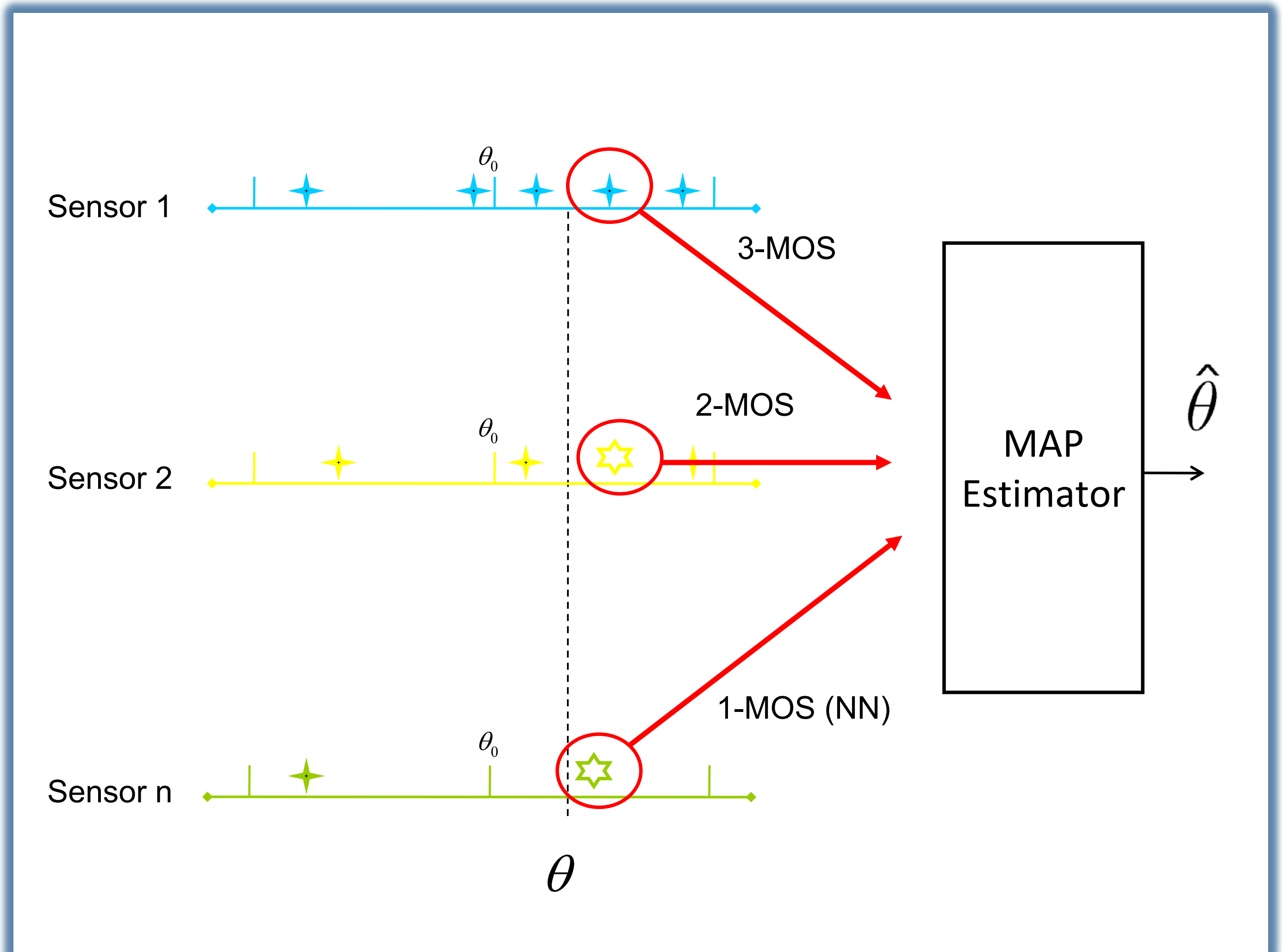
$$p(Z \mid \theta,n) = (1-P_d)\left(1/V\right)^n \mu(n) + \frac{1}{n} P_d \left(1/V\right)^{n-1} \mu(n-1) \sum_{i=1}^{n} p(z_i \mid \theta)$$

$$\mu(n) = (\lambda V)^n e^{-\lambda V} / n!$$

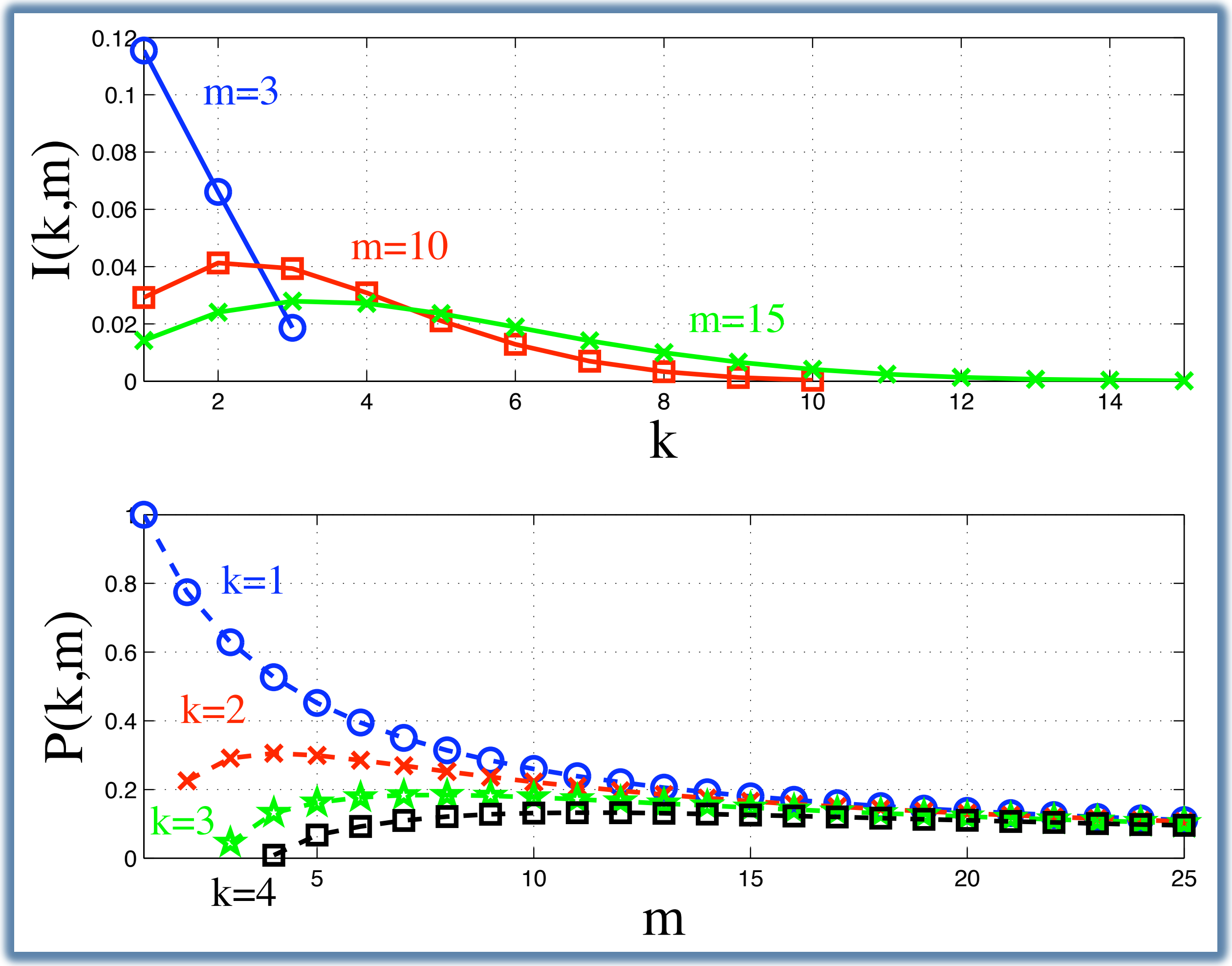




1-Dimensional Gate (1D) 2-Dimensional Gate (2D)

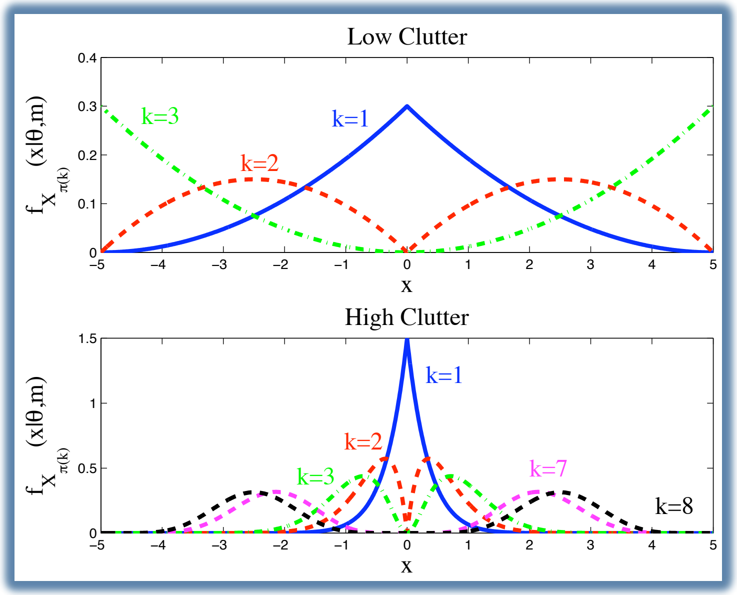


Note that the sensors corroborate one another, for case that θ is far away from expected location θ_0



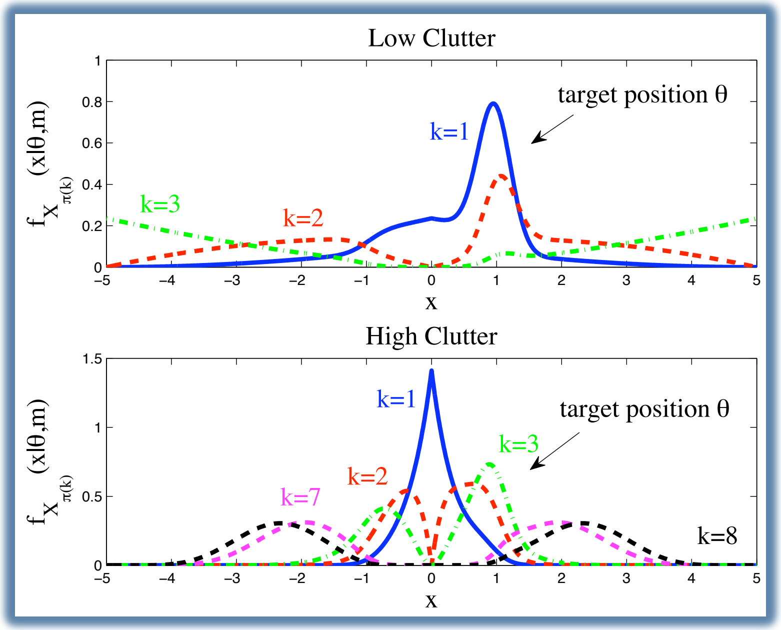
A Fisher Information analysis suggests that when the clutter is high it is better to transmit a higher k-mos.

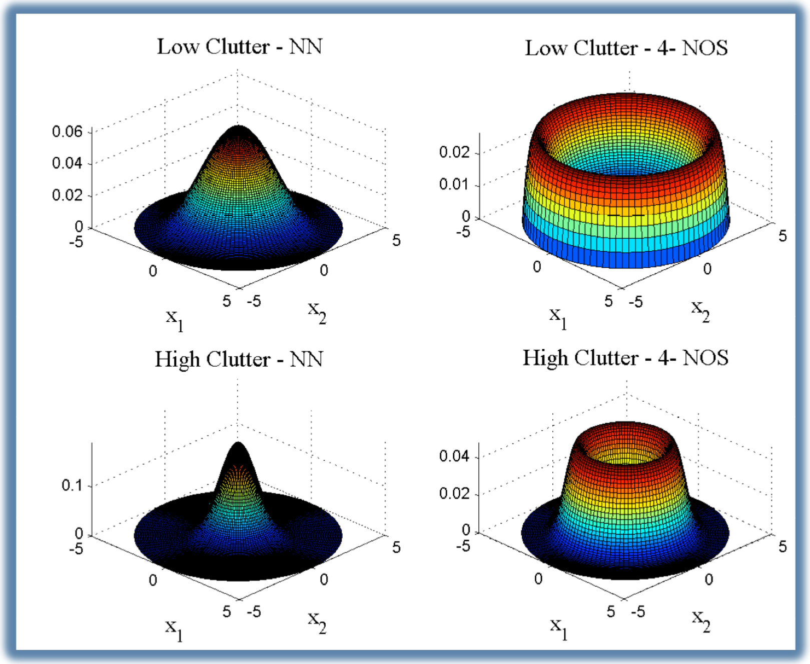
Apparently, however, the probability that a given k-mos is target originated is always highest for the nearest neighbor.



Probability densities of various k-mos for the clutter-only situation. Low clutter is three gated contacts, and high clutter is ten gated contacts.

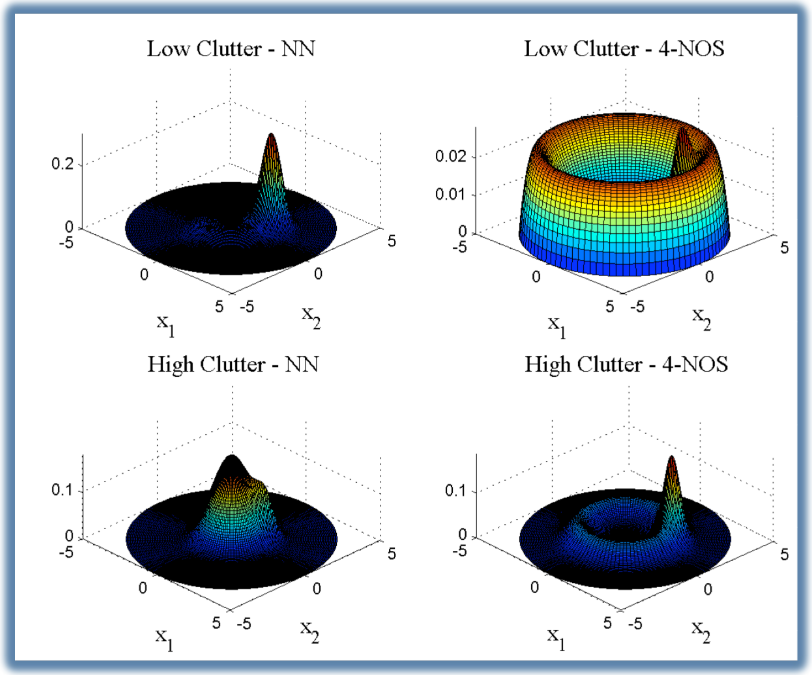
Probability densities of various k-mos for the target-present situation (true θ is unity). Note the appearance of a "bump" around the true θ for the high-clutter case – this is why a higher k-mos may be a better choice.





Probability densities of various k-mos for the clutter-only situation and two-dimensional observations.

Probability densities of various k-mos for the target-present situation and two dimensions. Note that the "bump" at the true θ persists.



Data Fusion for Tracking

- Some slides from "Industrial Strength Real World Multi-Sensor Fusion" by Fred Daum (May 2nd 2016).
- Track-to-Track (T2T) association
  - in the two-sensor case it is relatively easy
  - auction algorithm
- Bias estimation
  - example of passive-sensor tracking with angle biases

Taxonomy of Fusion for Tracking

- Type I configuration:
  - Single sensor situation, which serves as a baseline.
- Type II configuration:
  - Single sensor tracking followed by *track to track association and fusion*. Subtypes include with/without memory, and with/without feedback.
- Type III configuration:
  - Measurement to measurement association across sensors with all the measurements from the same time (the sensors are assumed perfectly synchronized), i.e., *static association*, followed by *central dynamic association and tracking*.
- Type IV configuration:
  - Completely *centralized association and tracking*.

Type I Configuration

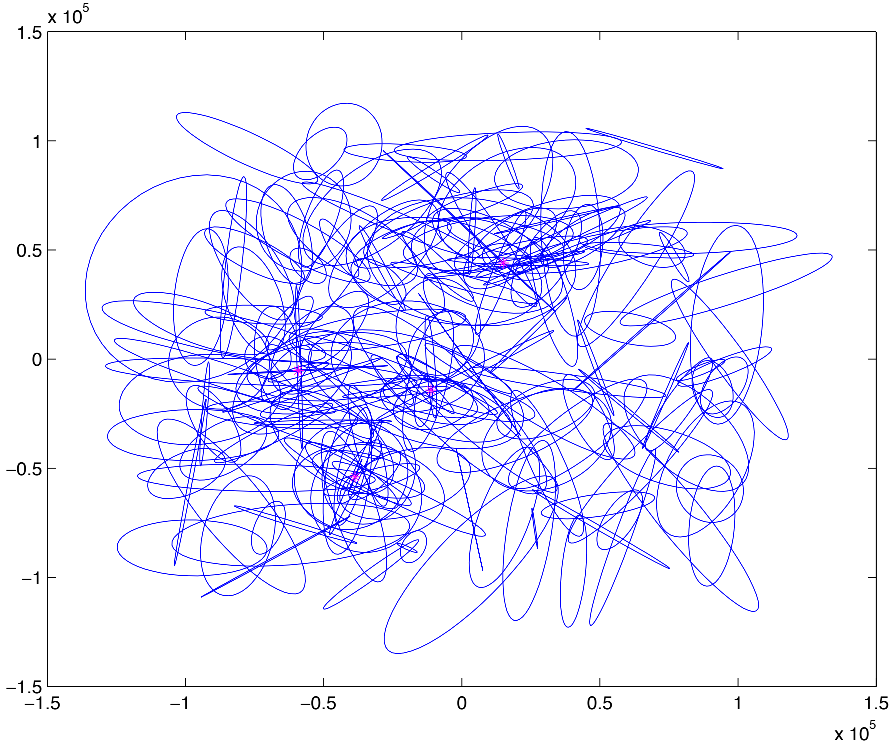
- Single sensor situation.
- In a multisensor situation this corresponds to *reporting responsibility (RR)*. Each sensor operates alone and has responsibility for a certain sector of the surveillance region — no fusion of the data (measurements or tracks) from the multiple sensors is done.
- As targets move from one sector to another, they are handed over – *handoff* – in a manner that depends on the system. Generally, the mechanism is to assign responsibility to the sensor with the highest expected accuracy, although workload and communication constraints can also play a role.

Type II Configuration

- Each sensor maintains its own (distributed) track.
  - this is often the preferred solution
  - solution is robust to failure and relatively light in its communication requirements
- Issues:
  - sensor registration & bias
  - track-to-track association (T2TF)
  - correlation between distributed tracks?
  - fused covariance?

Type III Configuration

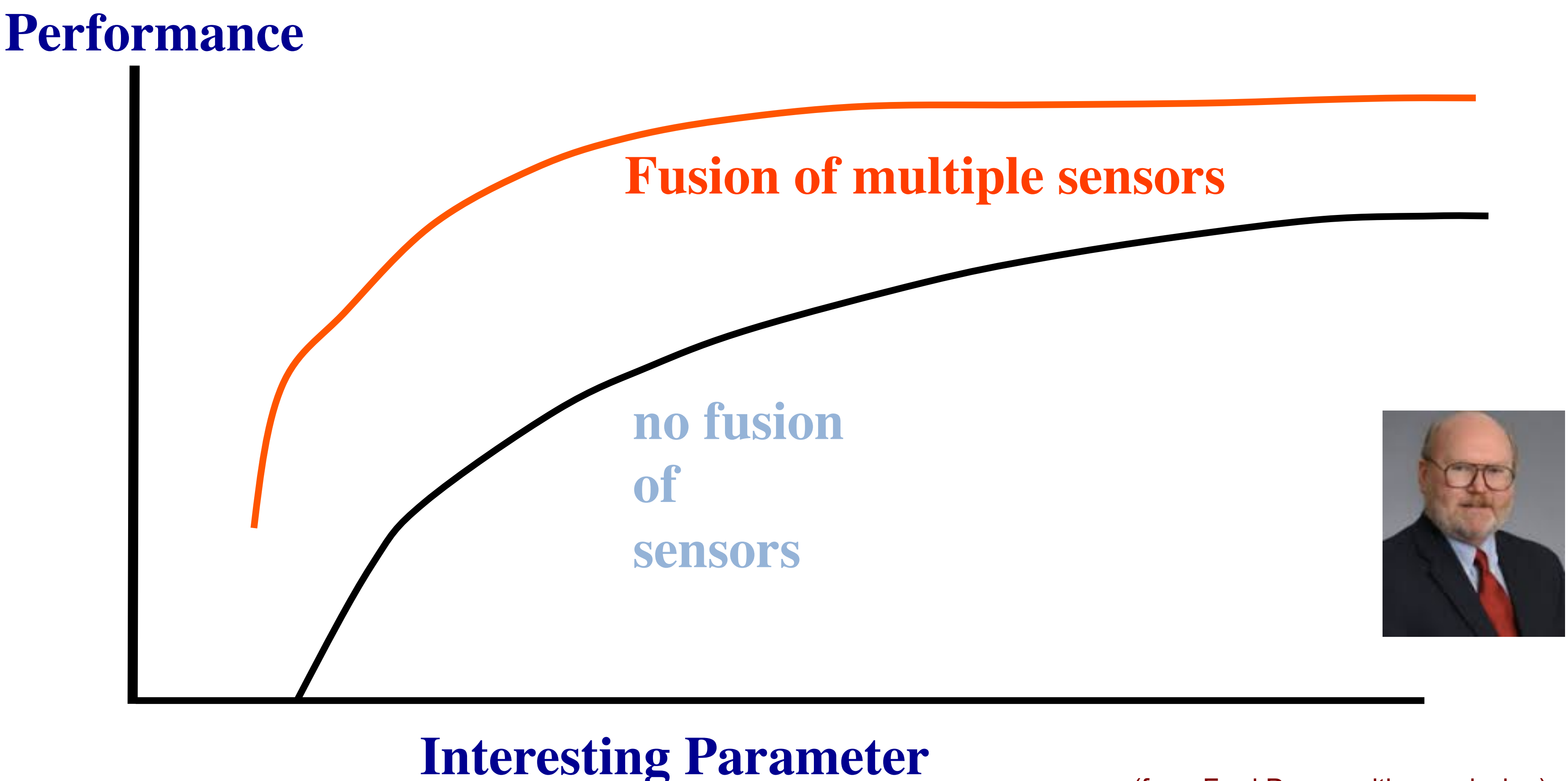
- Synthetic example of detections to be fused.
  - Covariances are random,
  - Pd = 50%, 25 sensors, λ = 5.
  - There are four "true" targets illustrated by magenta stars.
- This is not traditional pre-detection fusion!
  - The detections must be clustered before being fused.



Type IV Configuration

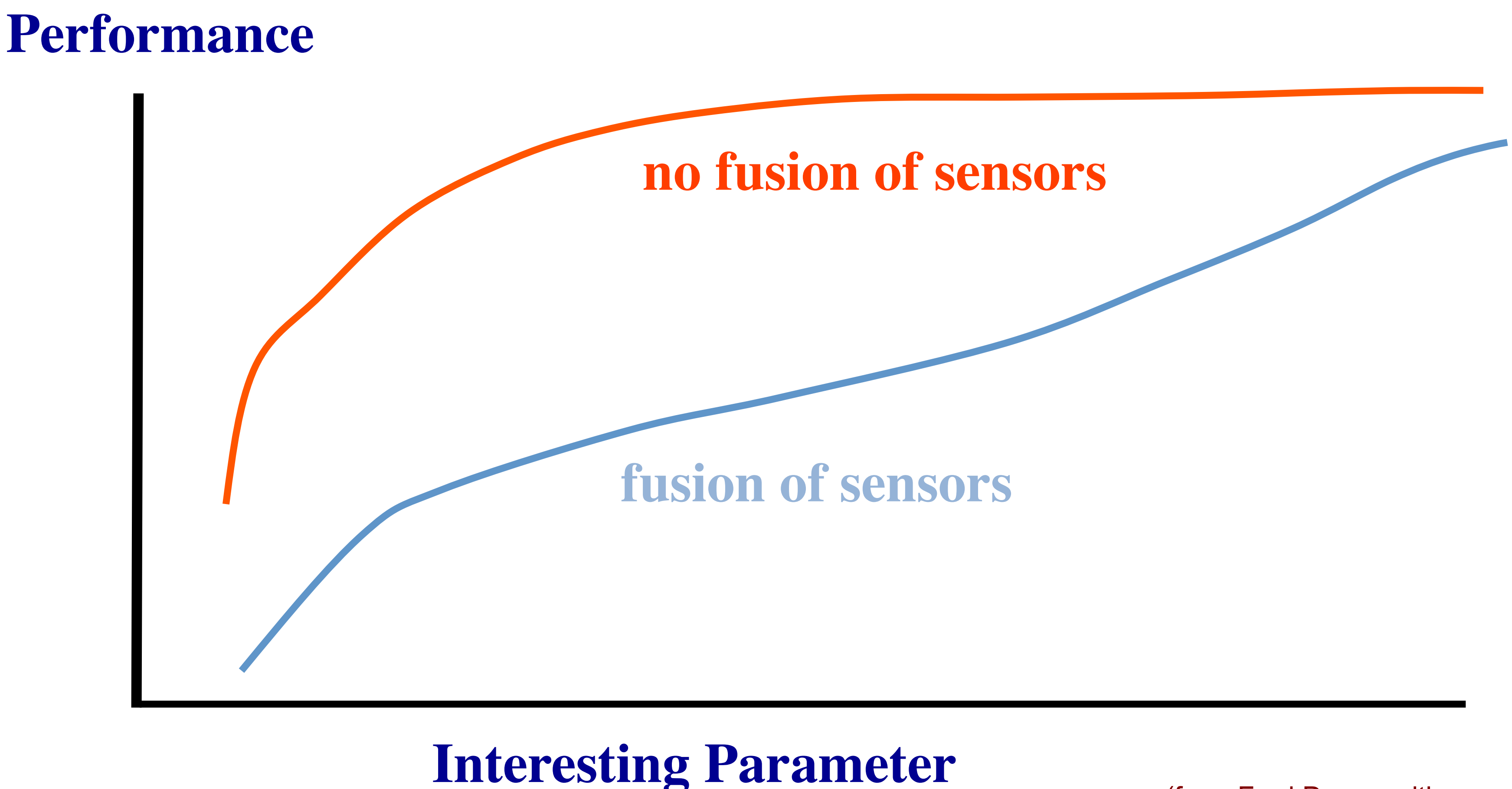
- Completely centralized association and tracking. For realistic multi-sensor processing must allow for out-of-sequence measurements (OOSMs).
  - can happen because plots arrive via network, perhaps datagram routing
  - optimally: recompute entire solution when OOSM arrives – avoid this!
  - exact single-gain "corrector" solution for single-lag case [Bar-Shalom] approximate single-gain "corrector" solution for multi-lag case [Bar-Shalom, Mallick, others]
  - exact multi-lag solution based on "accumulated state density" [Koch & Govaers]
- Sensors need not (and should not be assumed to) be synchronized.

Theoretical Multi-sensor Fusion



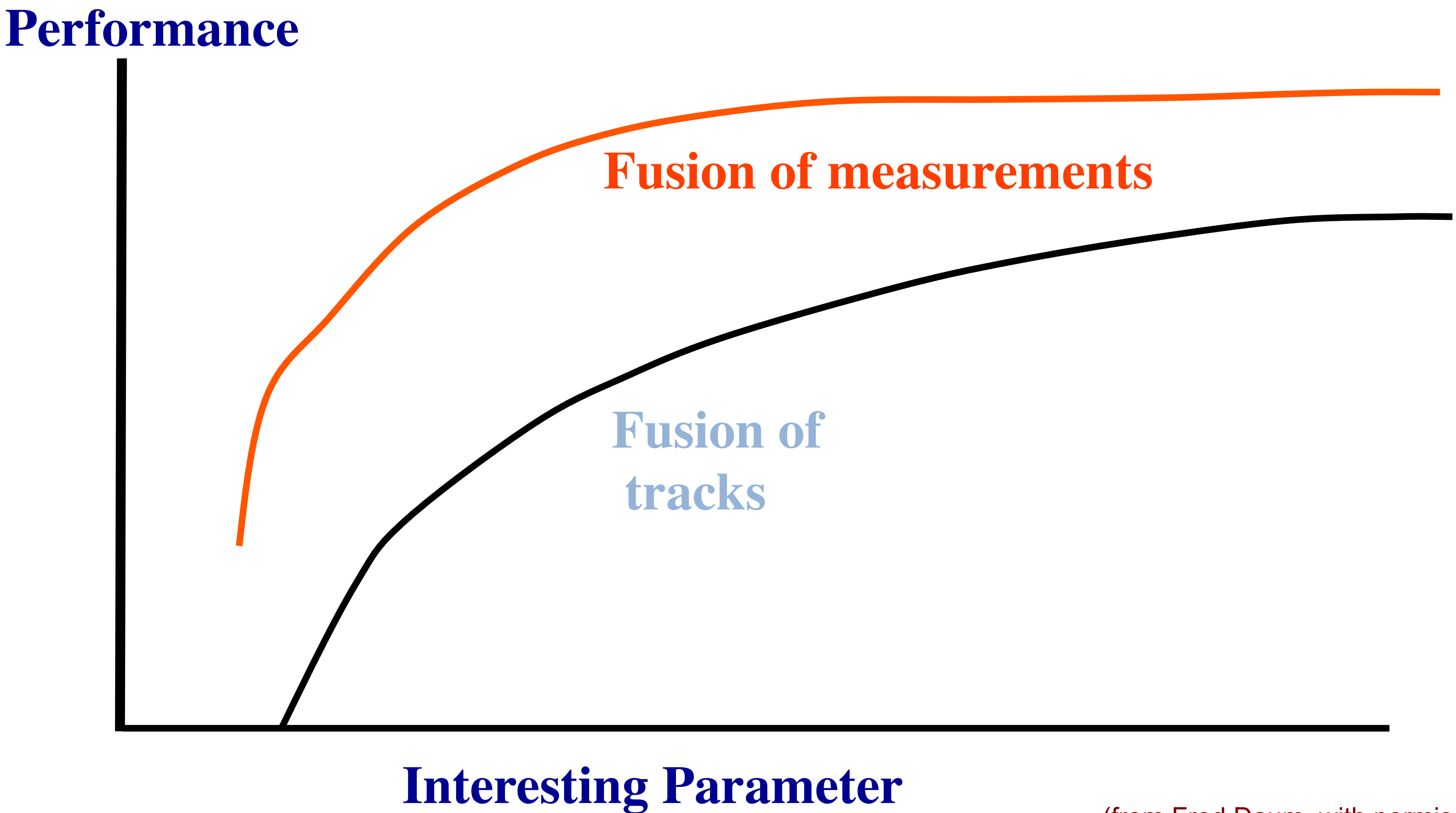
(from Fred Daum, with permission)

Real World Multi-sensor Fusion



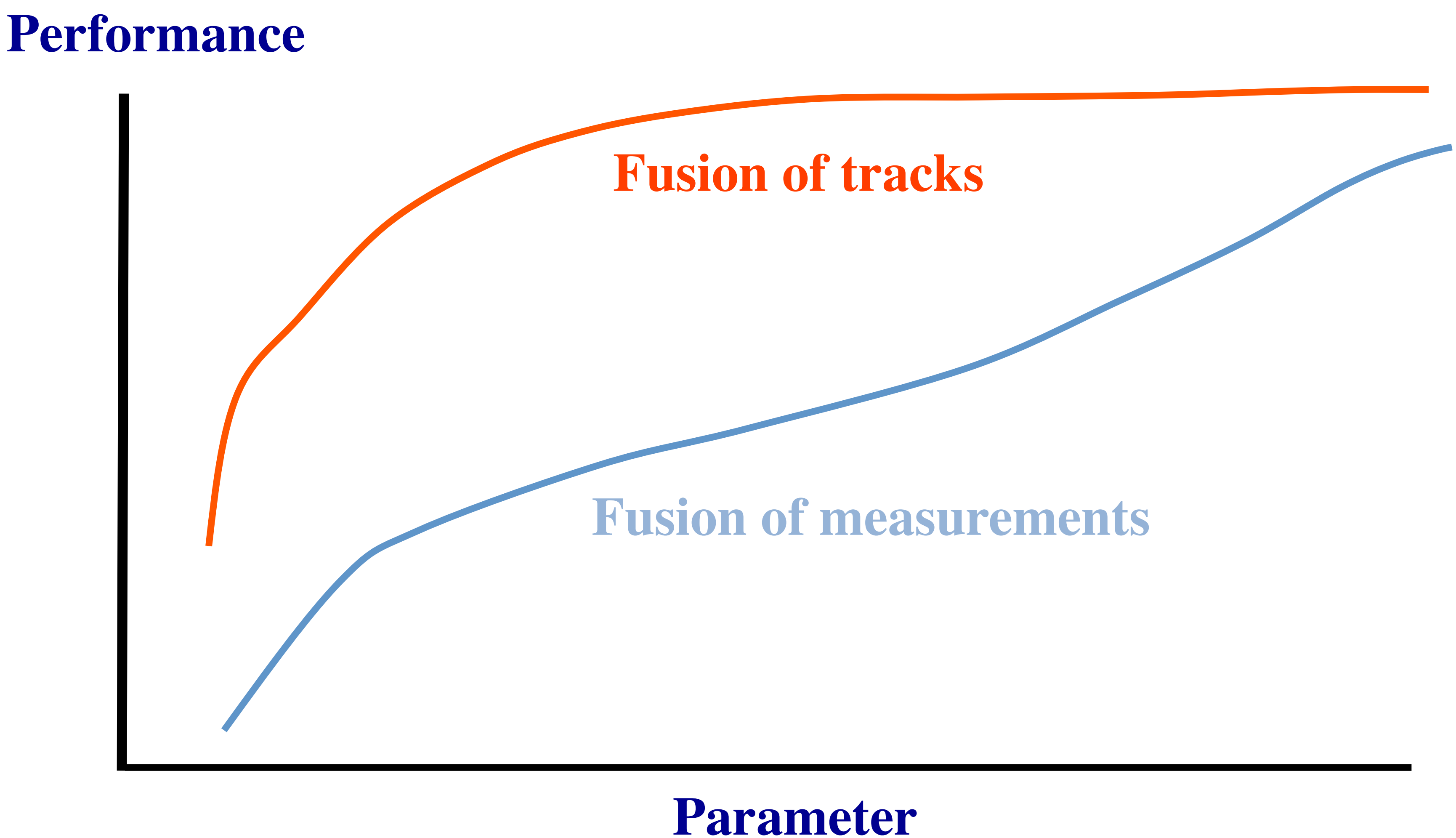
(from Fred Daum, with permission)

Theoretical Multi-sensor Fusion



(from Fred Daum, with permission)

Real World Multi-sensor Fusion



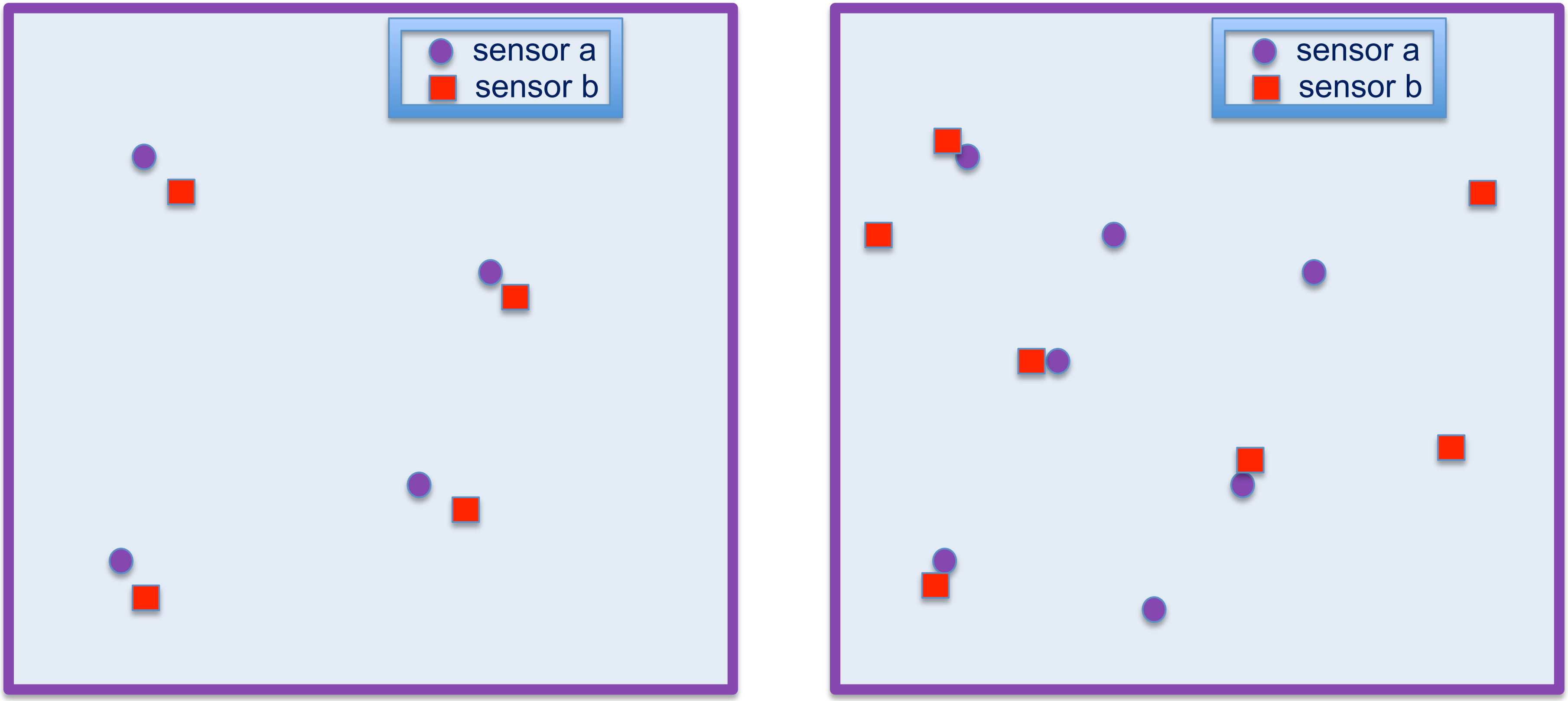
(from Fred Daum, with permission)

Key Real World Issues for Fusion

- residual bias between sensors
- targets detected by sensor A are not always the same as the targets detected by sensor B
- targets resolved by sensor A are not always the same as the targets resolved by sensor B
- targets tracked by sensor A are not always the same as the targets tracked by sensor B
- not all relevant data or tracks are reported by all data links
- inconsistent covariance matrices (of data or tracks) from sensors

(from Fred Daum, with permission)

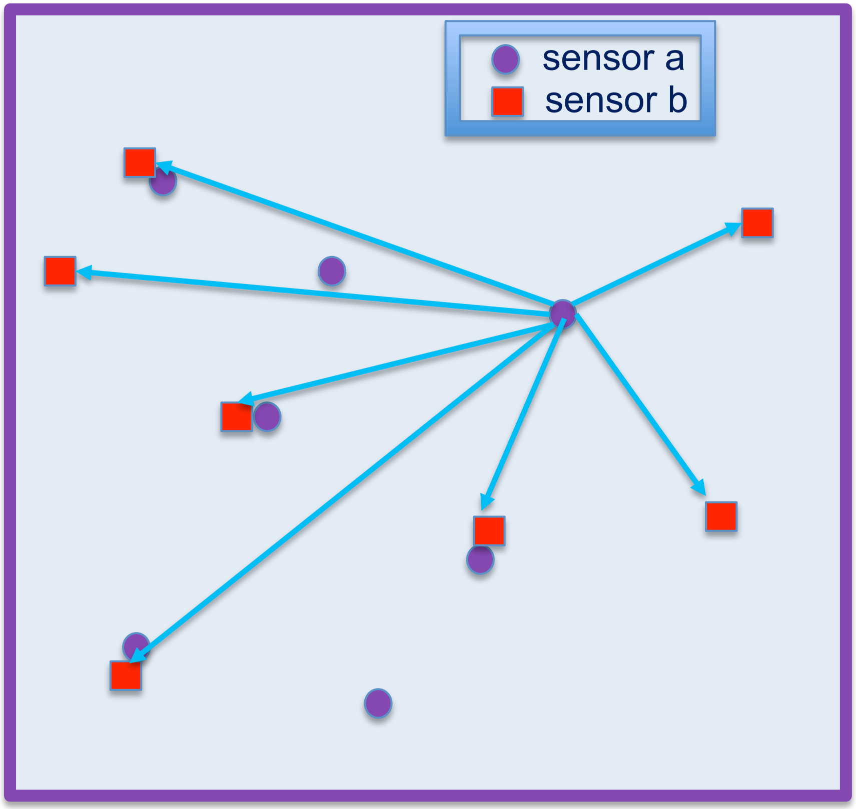
Track Association vs. Bias



Bias MOU

T2T Association (MOU)

- The goal is to minimize the "cost" such that no target gets assigned twice.
- For two sensors the problem is relatively easy and there exist polynomial-time algorithms for it.
  - we'll look at this
- For more than two sensors the problem is NP-hard
  - relaxation



Costs

Sensor 2 (columns) / Sensor 1 (rows)

	x_1	x_2	x_3	false
x_1	**c_{11}**	c_{12}	c_{13}	c_{10}
x_2	c_{21}	**c_{22}**	c_{23}	c_{20}
x_3	c_{31}	c_{32}	**c_{33}**	c_{30}
false	c_{01}	c_{02}	c_{03}	

Sensor 2 (columns) / Sensor 1 (rows)

	x_1	x_2	x_3	false
x_1	c_{11}	c_{12}	c_{13}	**c_{10}**
x_2	c_{21}	c_{22}	c_{23}	**c_{20}**
x_3	c_{31}	c_{32}	c_{33}	**c_{30}**
false	c_{01}	c_{02}	c_{03}	

Sensor 2 (columns) / Sensor 1 (rows)

	x_1	x_2	x_3	false
x_1	c_{11}	c_{12}	c_{13}	**c_{10}**
x_2	c_{21}	**c_{22}**	c_{23}	c_{20}
x_3	c_{31}	c_{32}	c_{33}	**c_{30}**
false	**c_{01}**	c_{02}	**c_{03}**	

Sensor 2 (columns) / Sensor 1 (rows)

	x_1	x_2	x_3	false
x_1	c_{11}	c_{12}	**c_{13}**	c_{10}
x_2	c_{21}	**c_{22}**	c_{23}	c_{20}
x_3	**c_{31}**	c_{32}	c_{33}	c_{30}
false	c_{01}	c_{02}	c_{03}	

T2T Assignment Costs

(etc.)

Criterion Matrix

Sensor 2 (columns) / *Sensor 1* (rows)

70	65	95	75
75	16	34	25
27	11	58	50
67	49	22	69

Prices

0
0
0
0

Assignment Matrix

Sensor 2 (columns) / *Sensor 1* (rows)

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Initial state. Begin with first association and set price to one that maximizes the difference between gain and price

Criterion Matrix

Sensor 2 (columns) / *Sensor 1* (rows)

70	65	95	75
75	16	34	25
27	11	58	50
67	49	22	69

Prices

0
5
0
0

Assignment Matrix

Sensor 2 (columns) / *Sensor 1* (rows)

0	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0

Turns out to be second target. Repeat for second sensor-2 track.

Criterion Matrix

Sensor 2 (columns) / *Sensor 1* (rows)

70	65	95	75
75	16	34	25
27	11	58	50
67	49	22	69

Prices

16
5
0
0

Assignment Matrix

Sensor 2 (columns) / *Sensor 1* (rows)

0	1	0	0
1	0	0	0
0	0	0	0
0	0	0	0

Turns out to be second sensor-1 track. Repeat for second sensor-2 track, which takes the first sensor-1 track.

Criterion Matrix

Sensor 2 (columns) / *Sensor 1* (rows)

70	65	95	75
75	16	34	25
27	11	58	50
67	49	22	69

Prices

37
5
0
0

Assignment Matrix

Sensor 2 (columns) / *Sensor 1* (rows)

0	0	1	0
1	0	0	0
0	0	0	0
0	0	0	0

Turns the third sensor-2 track likes the first target-1 track more than the second target-2 track does.

Criterion Matrix

Sensor 2 (columns); ***Sensor 1*** (rows)

70	65	95	75
75	16	34	25
27	11	58	50
67	49	22	69

Prices

37
5
0
21

Assignment Matrix

Sensor 2 (columns); ***Sensor 1*** (rows)

0	0	1	0
1	0	0	0
0	0	0	0
0	1	0	0

The second target-2 track has most gain possible, so back to that one. It turns out to like the 4th target-1 track the most. Note that price is now 21=49-(65-37).

Criterion Matrix

Sensor 2 (columns); ***Sensor 1*** (rows)

70	65	95	75
75	16	34	25
27	11	58	50
67	49	22	69

Prices

37
5
2
21

Assignment Matrix

Sensor 2 (columns); ***Sensor 1*** (rows)

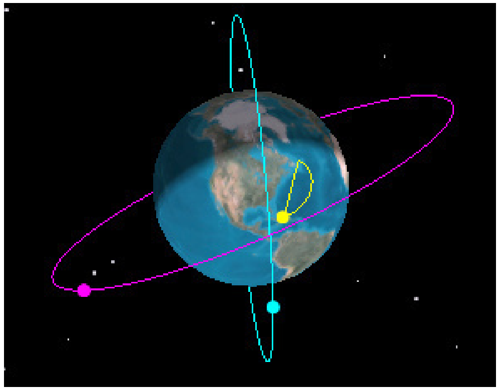
0	0	1	0
1	0	0	0
0	0	0	1
0	1	0	0

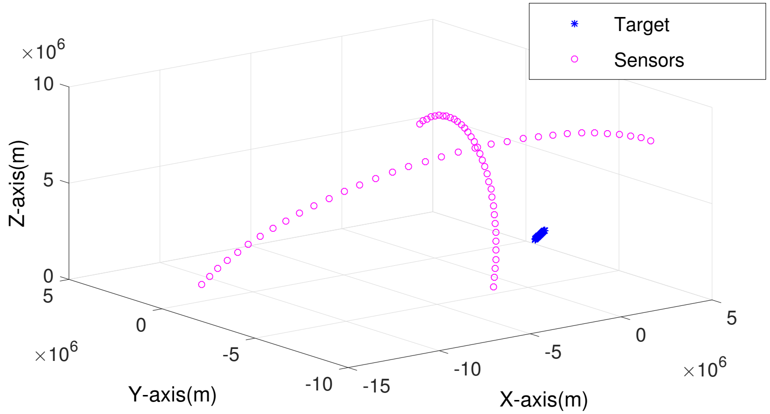
The 4th target-2 track gets assigned to the 3rd target-1 track. Price is 2=50-(69-21).

Bias: Example FPA Sensors

- There can be biases in range, time – all kinds of things – but most often they come to the fore in angle-only sensing.
- Consider (the important) application of multi-sensor tracking of threats from multiple satellites.
  - Biases here are roll (ϕ), pitch (ρ) and yaw (ψ).
- These can be estimated by using targets of opportunity or multiple frames of data.
- There are $3 \times N_{sensor}$ biases and $3 \times N_{target}$ target parameters to estimate, and $2 \times N_{sensor} \times N_{target}$ observations.
  - For 2 sensors we would need at least 6 targets.

Example of Bias Estimation





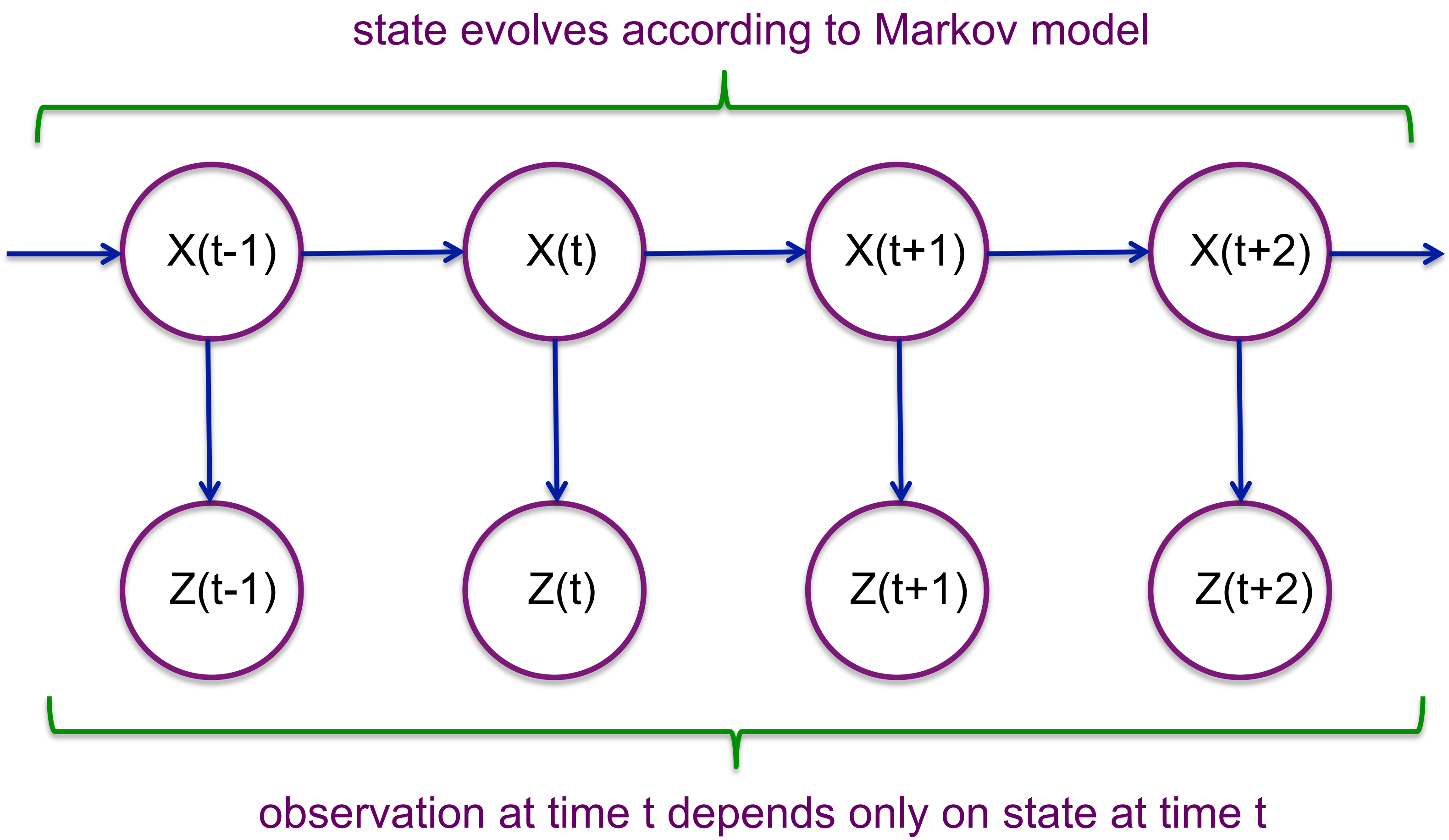
Scheme	Position RMSE	Velocity RMSE
1	107.44	5.16
2	47,161.10	25,149.32
3	494.49	19.55

Scheme 1: No bias.
Scheme 2: Ignore bias.
Scheme 3: Estimate bias.

- For multi-frame single-target data there are $3 \times N_{sensor}$ biases and 6 target parameters to estimate (velocities!), and $2 \times N_{sensor} \times N_{frame}$ observations.
  - For 2 sensors we would need at least 3 frames.

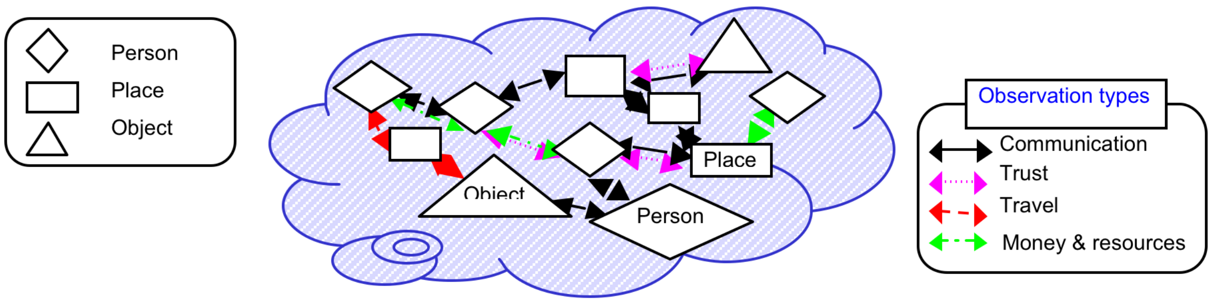
"Hard" Tools for a "Soft" Problem

- A traditional target evolves according to a Markov model
  - means that $p(\mathbf{x}(t)|\mathbf{x}(t-1),\mathbf{x}(t-2),\ldots) = p(\mathbf{x}(t)|\mathbf{x}(t-1))$.
  - usual model is $\mathbf{x}(t) = f(\mathbf{x}(t-1), \mathrm{v}(t))$ where f is some function and v is noise.
- The observation is occluded:
  - roiled by noise
  - missed detections
  - false alarms
  - multiple targets
- That is: a "hidden" Markov model (HMM).
- Can we apply our target tracking knowledge / expertise to other non-traditional models?

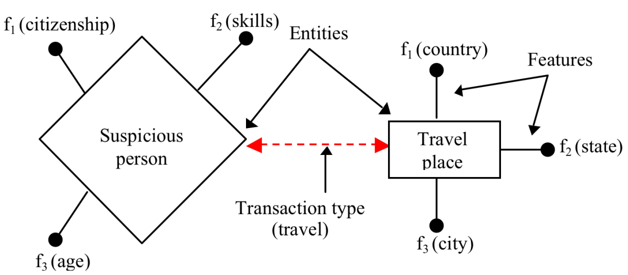

state evolves according to Markov model
X(t-1)
X(t)
X(t+1)
X(t+2)
Z(t-1)
Z(t)
Z(t+1)
Z(t+2)
observation at time t depends only on state at time t

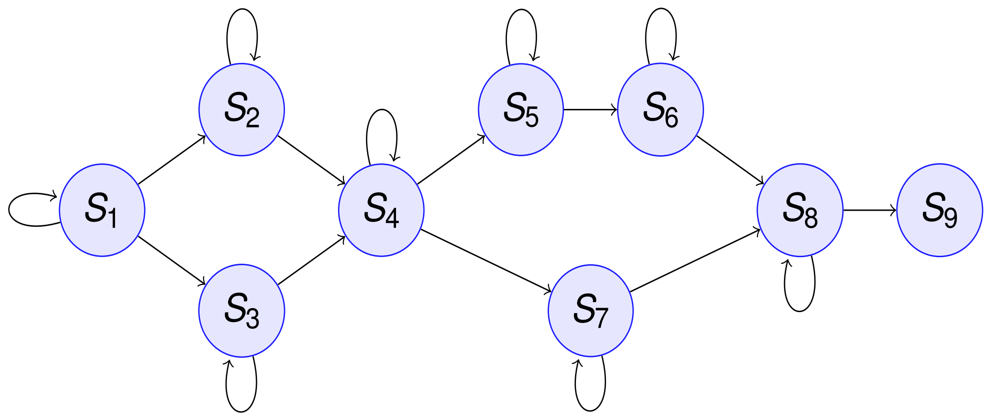
- Let's try to model some nefarious plot
- Time model
  - carrying out an attack requires planning
  - steps of the plan form a pattern
  - pattern of actions can be modeled using a Markov chain
- Observation model
  - terrorists leave detectable clues about enabling events
  - clues are not direct observations, but are related to them
  - the states in the Markov chain are hidden.
- Clutter
  - refers to false / irrelevant / spurious observations
  - example: someone has bought fertilizer
  - fertilizer bomb?
  - actual interest in farming?

Transactions are necessary in order for "plan" to evolve:



Underlying the observation stream, a puzzle is being fit together.

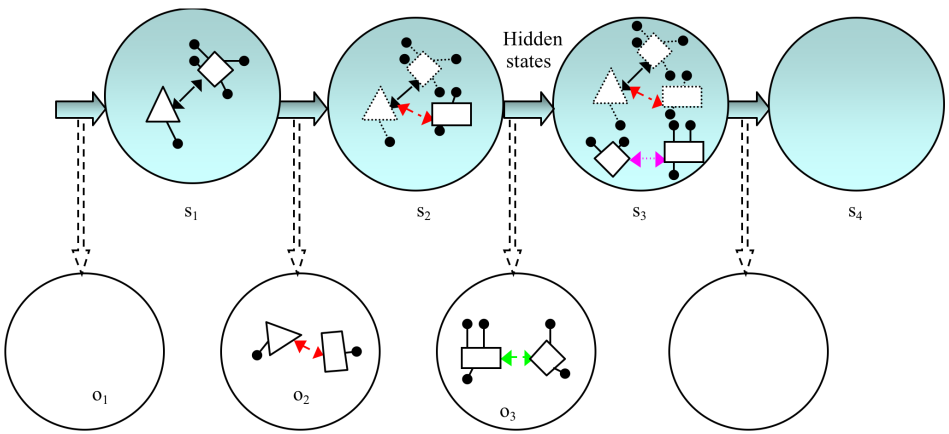




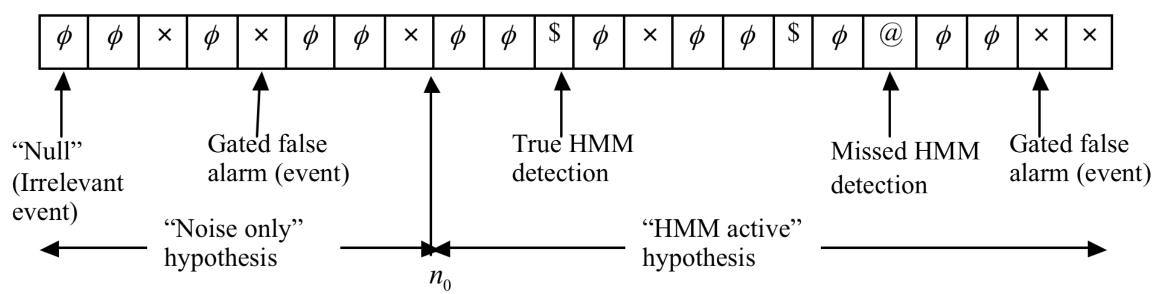
- S_1: Selection of targets and reconnaissance
- S_2: Set up cell A_1
- S_3: Set up cell A
- S_4: Acquire money for operation
- S_5: Gather resources
- S_6: Expert arrives to assemble bombs
- S_7: Target reconnaissance
- S_8: Communications and final setup
- S_9: Attack

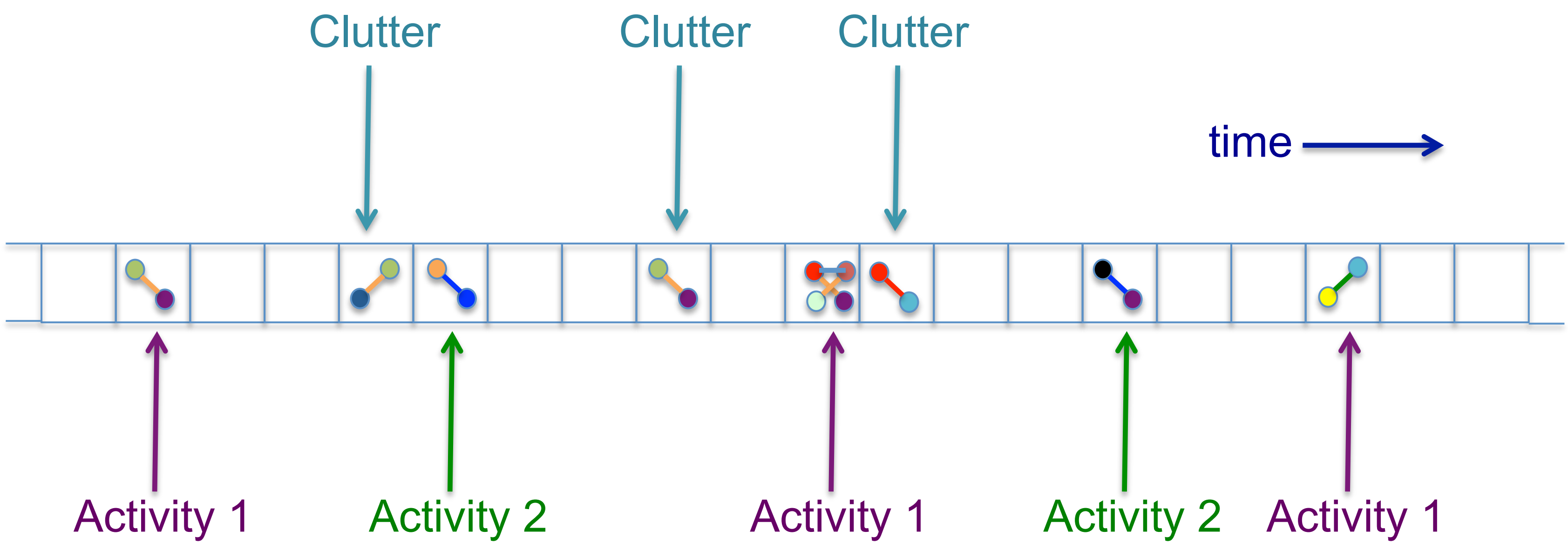
Truck bombing example: Really, too simple.

Graph evolves probabilistically from one state to the next:



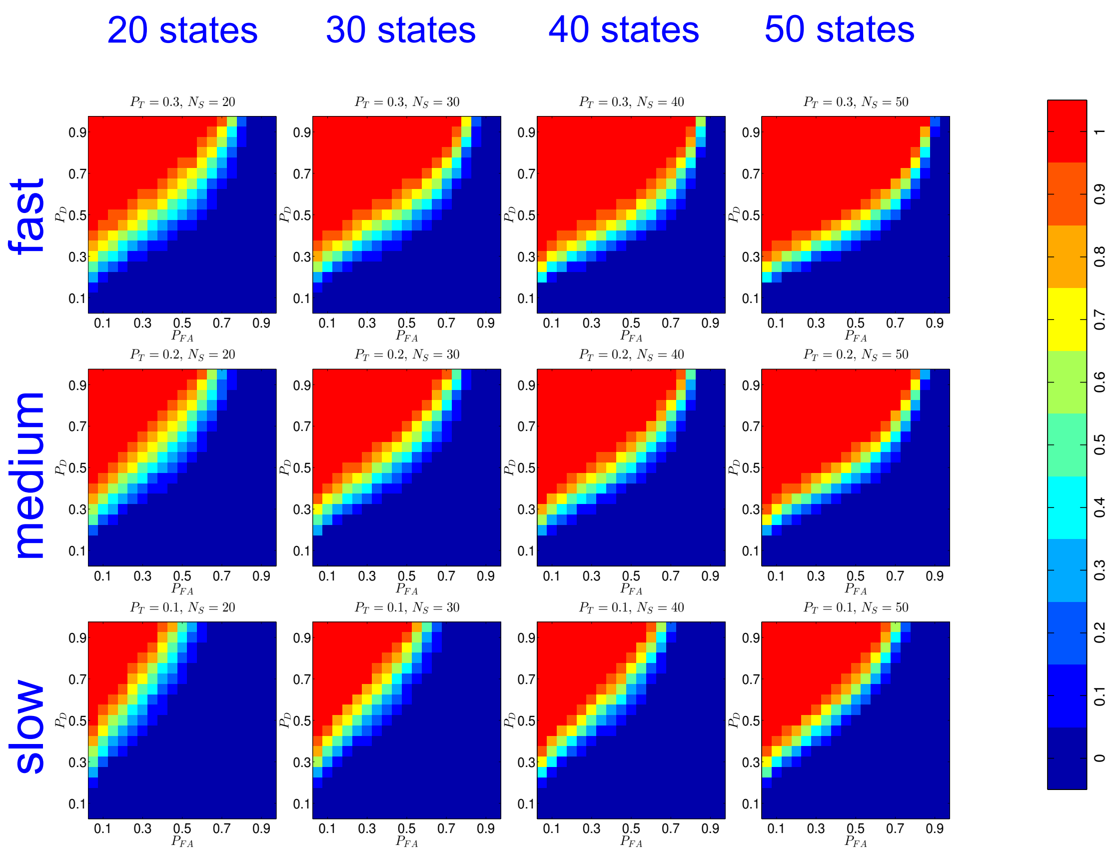
HMM's observations are new elements being added to network.





- The observation stream (transactions) is from a (logical) OR-ing of component parts from several “targets” of interest and clutter.
- We seek a multi-target tracker that is appropriate for the job.
- We have developed a **multi-Bernoulli filter (MBF)** to extract it.
- We have begun to analyze “detectability.”
- In the future we will extend it to multiple activities and features.

- Granstrom, Willett & Bar-Shalom, “Asymmetric Threat Modeling Using HMMs: Bernoulli Filtering and Detectability Analysis,” TSP 2016.

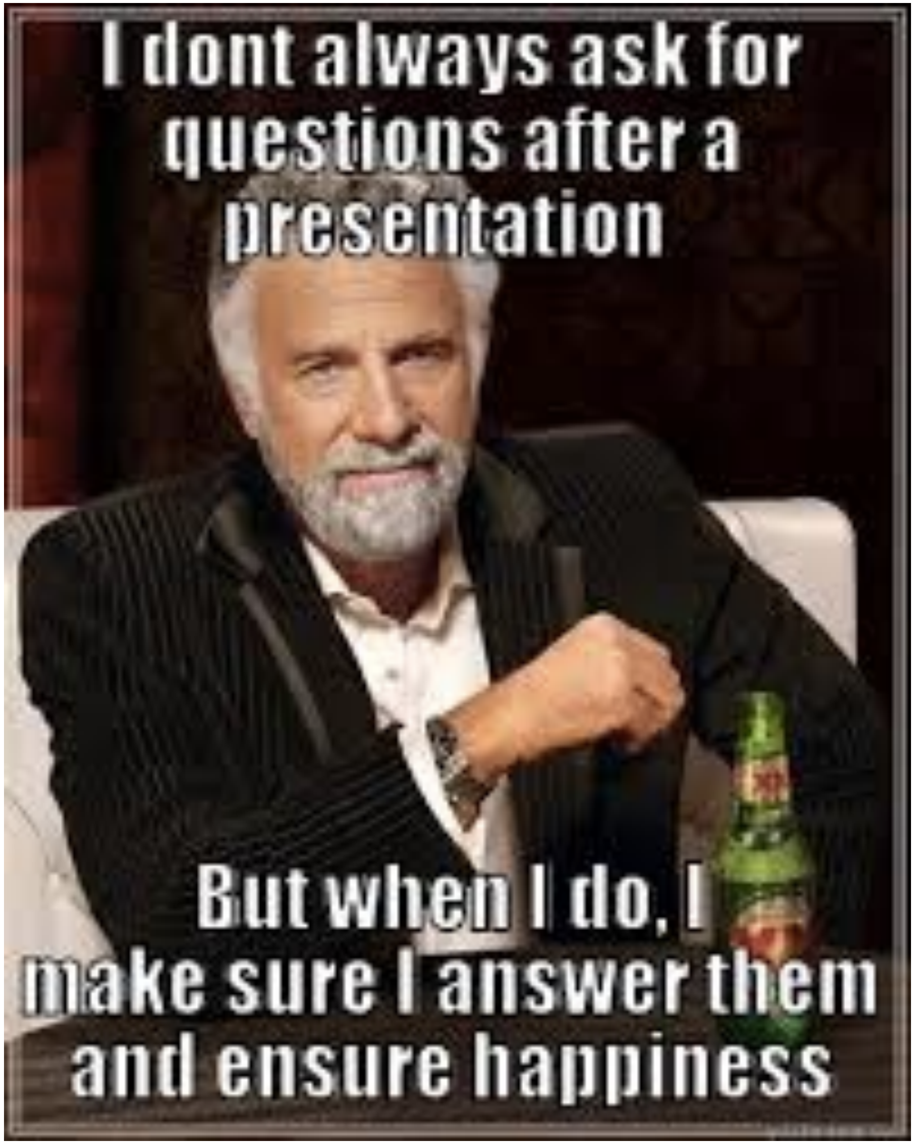


Detection rate (*D*) at 10% false alarm rate (*FA*) for daisy chain HMMs with probability of state transition *PT* , and number of states *N*.

Detectability vs. Complexity & Speed

Summary

- Multi-User Information Theory
  - typicality: entropy & capacity
  - MAC, broadcast, CEO problem
- Case studies
  - scan statistics for sensor networks
  - consensus in sensor networks
  - data fusion with intermittent detections
  - quantized estimation: a note
  - decentralized learning
  - decentralized estimation with MOU
- Data Fusion for Tracking
  - architectures
  - bias
  - track fusion
- Example of Application of Hard Methods to a Soft Problem


I dont always ask for questions after a presentation
But when I do, I make sure I answer them and ensure happiness