Optimal, Robust Information Fusion in Uncertain Environments

MURI Review Meeting

Integrated Fusion, Performance Prediction, and Sensor Management for Automatic Target Exploitation

Alan S. Willsky

September 14, 2007

What is needed: An expressive, flexible, and powerful framework

- **Capable of capturing uncertain and complex sensor-target relationships**
  - Among a multitude of different observables and objects being sensed
- **Capable of incorporating complex relationships about the objects being sensed**
  - Context, behavior patterns
- **Admitting scalable, distributed fusion algorithms**
- **Admitting effective approaches to learning or discovering key relationships**
- **Providing the "glue" from front-end processing to sensor management**

Our choice: Graphical Models

- Extremely flexible and expressive framework
  - Allows the possibility of capturing (or learning) relationships among features, object parts, objects, object behavior, and context
    - E.g., constraints or relationships among parts, spatial and spatio-temporal relationships among objects, etc.
  - Natural framework to consider distributed fusion
- While we can't beat the dealer (NP-Hard is NP-Hard),
  - The flexibility and structure of graphical models provides the potential for developing scalable, approximate algorithms

What did we say at the kickoff? What have we done? - I

- Scalable, broadly applicable inference algorithms
  - Build on the foundation we have
  - Provide performance bounds/guarantees
- Some of the accomplishments this year
  - ***Tractable, "low-rank" uncertainty estimation in graphical inference***
  - Walk-sum analysis and guaranteed convergence
  - Lagrangian relaxation methods for tractable inference (in progress – tune in next year)

Low-rank uncertainty estimation - I

- Gauss-Markov Random Fields in Information Form
  - $p(x) \propto exp\{-x^T Jx/2 + h^T x\}$
  - $P = J^{-1}$
  - $Jm = h$
- J is sparse (captures graph structure)
  - Lots of efficient ways to solve for mean in O(N) comps.
  - How do we get variances, i.e., diag(P)??
- An intractable approach
  - $JP = I = [e_1, \ldots, e_N]$
  - Solve column by column: $JP_i = e_i$, $i = 1, \ldots, N$
  - This is $O(N^2)$ – infeasible for large problems

Low-rank uncertainty estimation - II

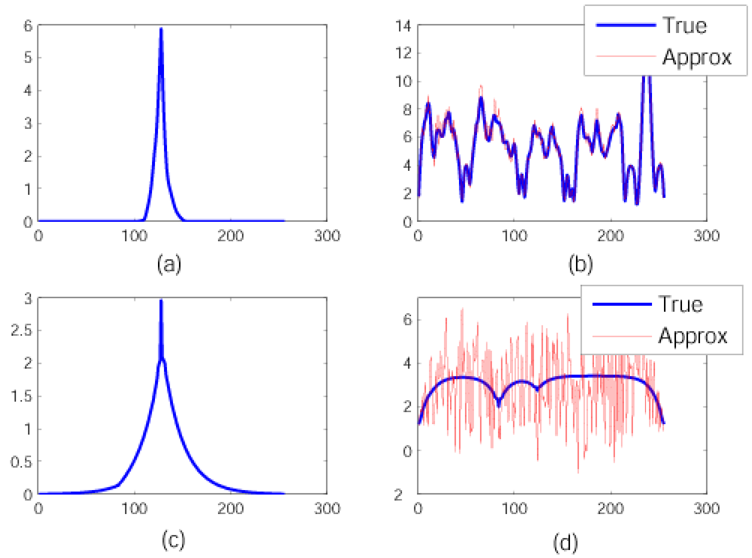
- Let's create a low-rank approximation to *I (!?!?)*
  - B - $N \times M$ $(M << N)$
    - **Rows** b_i of B all have unit norm
    - But they are overcomplete (N of them in M dimensions)
  - Solve $JP^a = BB^T \approx I$ *(!?!?)*
    - Actually, solve $JR = B$ $O(MN)$ complexity (solve column-wise)
    - Then $P^a = RB^T$
  - Here's the key – there are ***aliasing/splicing errors***
    - $(P^a)_{ii} = P_{ii} + \sum_{i \neq j} P_{ij} b_i^T b_j$
    - So: If P_{ij} is significant, we want b_i and b_j orthogonal
    - But: If $P_{ij} \approx 0$, ***we don't care***
      - So, we repeat some rows, with random sign flips so that the dot product is zero mean, variance = 1

Low-rank uncertainty estimation - III

- So, $(P^a)_{ii}$ is an unbiased estimate of P_{ii}
  - Variance equals the sum of squares of the P_{ij} for which $b_i = \underline{+} b_j$
    - Can reduce variance by averaging several solutions
  - More importantly, if we know the exponential fall-off in correlation structure, we have a ***graph-coloring problem***
    - Different colors mean orthogonal b_i
    - Same color means the b_i are equal except for random sign changes
  - In this case, we have bounds on error variance on $(P^a)_{ii}$ that decay exponentially with number of colors

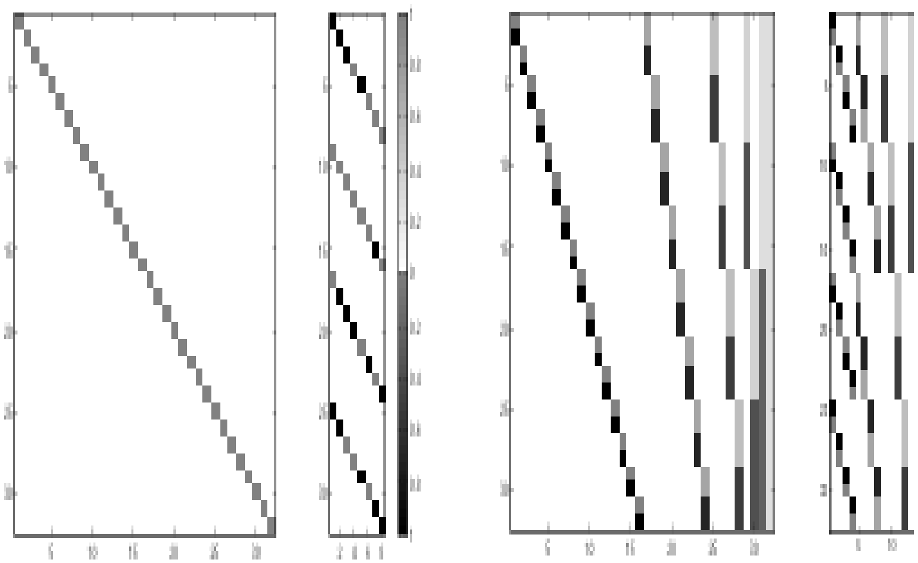
Low-rank uncertainty estimation - IV

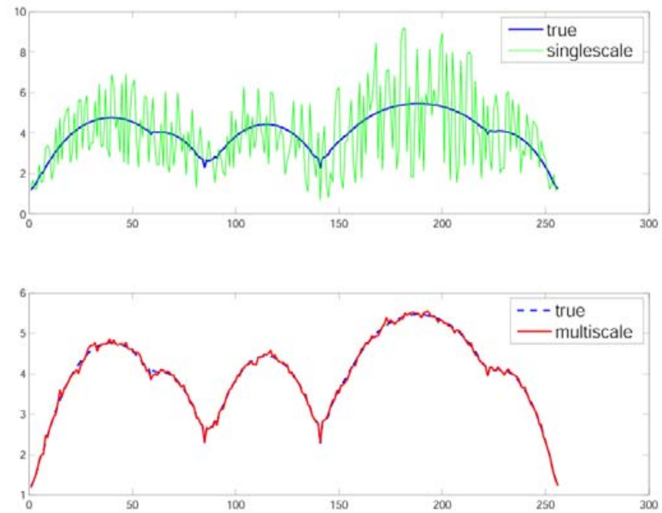
- 1-D examples (N = 256, M = 1)
  - Conditional variances for stationary processes with sparse measurements
    - Top: Process with short correlation
    - Bottom: Process with long correlation



Low-rank uncertainty estimation - V

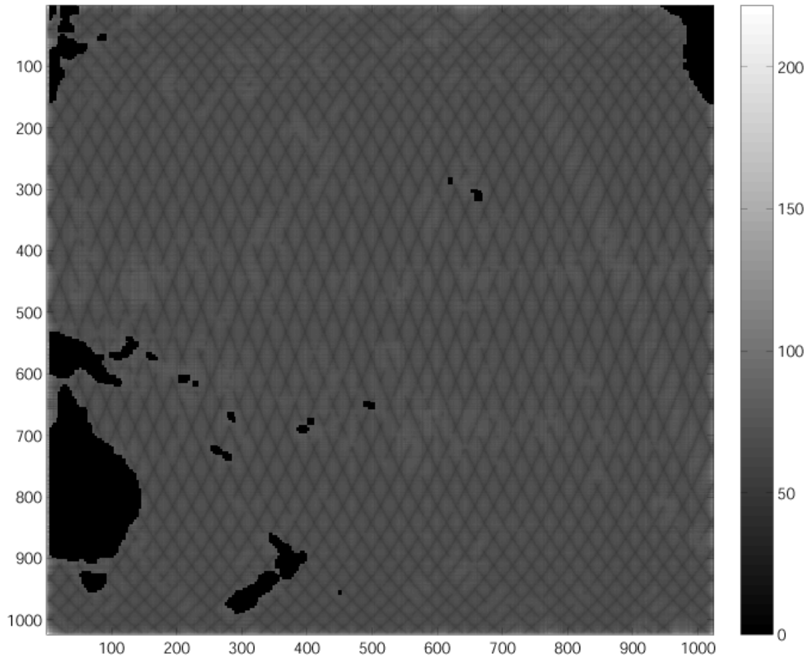
- Wavelets to the rescue: Let's splice ***them***
  - Within scale only





Low-rank uncertainty estimation - VI

- This really ***is*** low rank: An example with
  - $N = (1024)^2$ and $M = 448$



Low-rank uncertainty estimation - VII

- Extensions
  - Adapting bases (e.g., wavelet packets, curvelets, etc.)
  - For more general graphs: Diffusion wavelets
    - With thanks to Greg Arnold

What did we say at the kickoff? What have we done? - II

- Graphical-model-based methods for sensor fusion for tracking, and identification
  - Graphical models to capture motion patterns
  - Graphical models to capture relationships among features-parts-objects
- Some of the accomplishments this year
  - ***Hierarchical Dirichlet Processes to learn motion patterns and behavior***
  - HDPs for feature-part-object modeling and recognition

HDPs for Learning/tracking motion patterns

- Objective – learn motion patterns of targets of interest
  - Having such models can assist tracking algorithms
  - Detecting such coherent behavior may be useful for higher-level activity analysis
- Our first effort
  - Learning of maneuver models
  - Tracking algorithms (e.g., IMM) use such models but these are usually assumed to be prespecified
  - Can we learn them?

Jump-mean processes

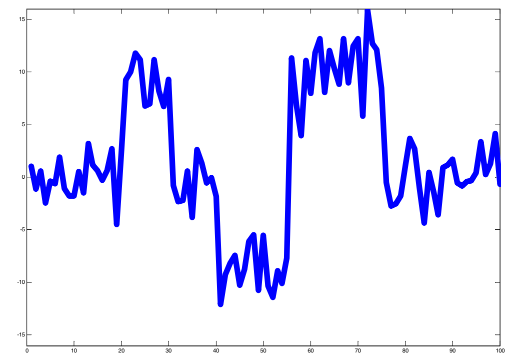
- Markov jump-mean process
  - System "jumps" between finite set of acceleration means
  - Hybrid continuous-discrete state:

$$\bar{x}_t = \begin{bmatrix} x_t \\ z_t \end{bmatrix}$$

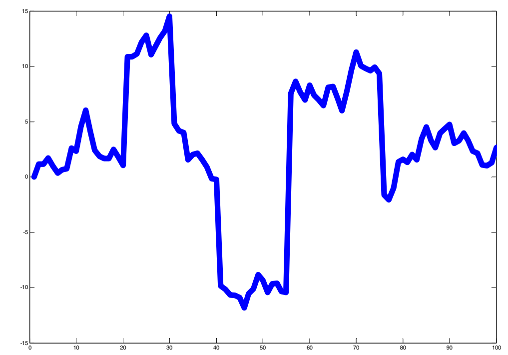
  - Dynamics described by:

$$\begin{aligned} x_t &= Ax_{t-1} + Bu_t(z_t) + v_t \\ &= Ax_{t-1} + \tilde{u}_t(z_t) \\ u_t|z_t &\sim \mathcal{N}(\mu_{z_t}, \Sigma_{z_t}) \end{aligned}$$

  - System is non-linear due to mode uncertainty

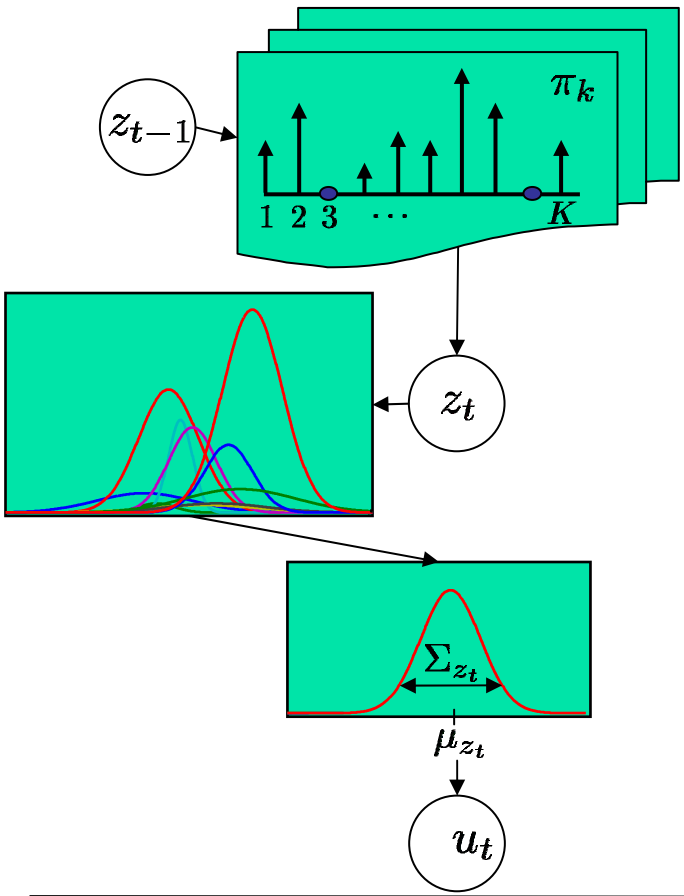


Constant Velocity (CV)

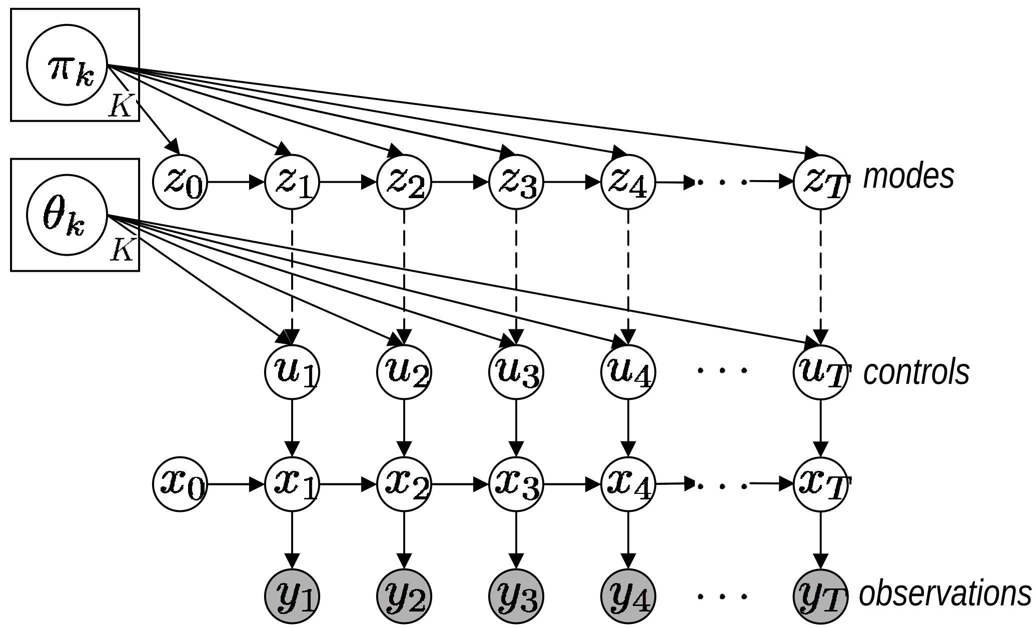


Constant Acceleration (CA)

Markov Jump-Mean System (MJMS)



Graph of Markov jump-mean system



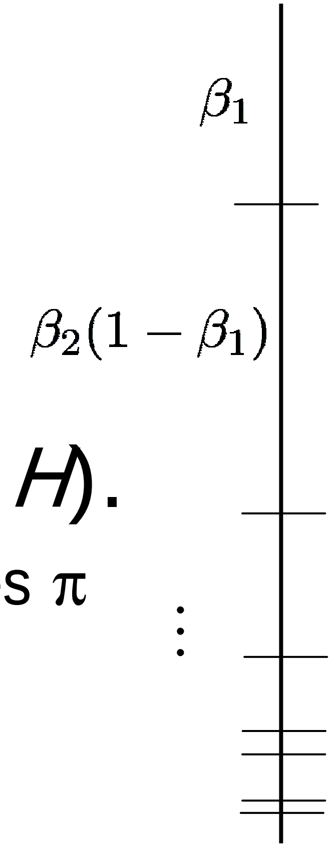
$$z_t | z_{t-1} \sim \pi_{z_{t-1}}$$
$$u_t | z_t \sim \mathcal{N}(\mu_{z_t}, \Sigma_{z_t})$$

Some questions

- How many possible maneuver modes are there?
- What are their individual statistics?
- What is the probabilistic structure of transitions among these modes?
- Can we learn these
  - Without placing an *a priori* constraint on the number of modes
  - Without having *everything* declared to be a different “mode”
- The key to doing this: Dirichlet processes

Dirichlet Process via Stick Breaking

$$\begin{aligned} \beta_i &\sim \mathrm{Beta}(1, \alpha) \quad i = 1, 2, \ldots \\ \pi_1 &= \beta_1 \\ \pi_i &= \beta_i \prod_{j=1}^{i-1} (1 - \beta_j) \quad i = 2, 3, \ldots \end{aligned}$$



- Corresponds to a draw from DP(α, *H*).
  - Mixture components drawn with probabilities π and with parameters drawn from *H*

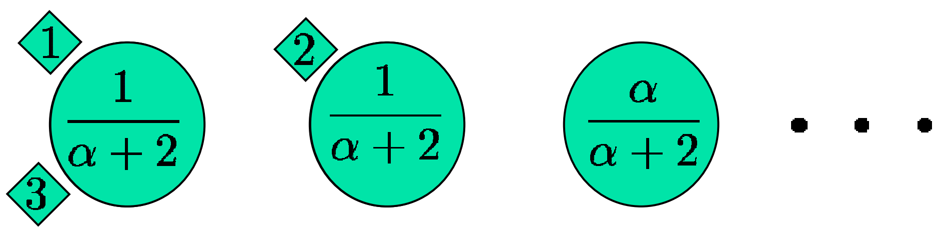
Chinese Restaurant Process

- Predictive distribution:

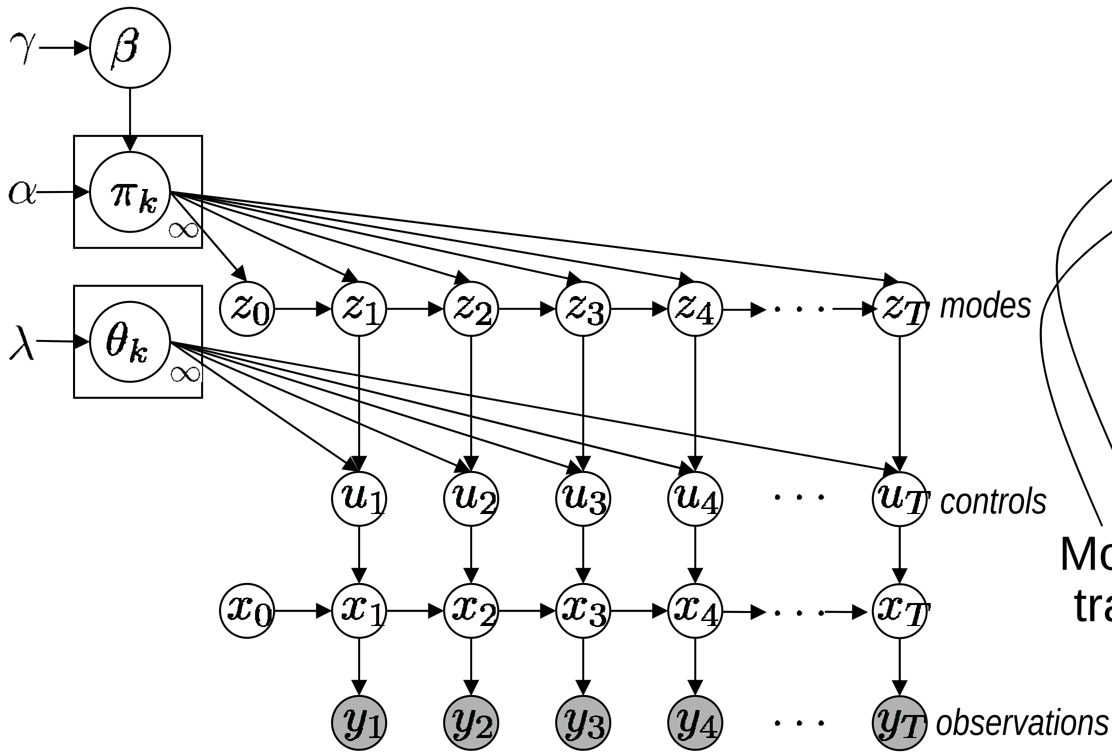
$$p(z_t = z \mid z_{\setminus t}, \alpha, H) = \frac{\alpha}{\alpha + T}\delta(z, K+1) + \frac{1}{\alpha + T}\sum_{k=1}^{K} T_k \delta(z, k)$$

Number of current assignments to mode k

- Chinese restaurant process:



Graphical Model of HDP-HMM-KF



$$\beta \sim \text{Stick}(\alpha)$$
$$\pi_k \sim \text{DP}(\alpha, \beta)$$
$$\theta_k \sim \text{NIW}(\lambda)$$

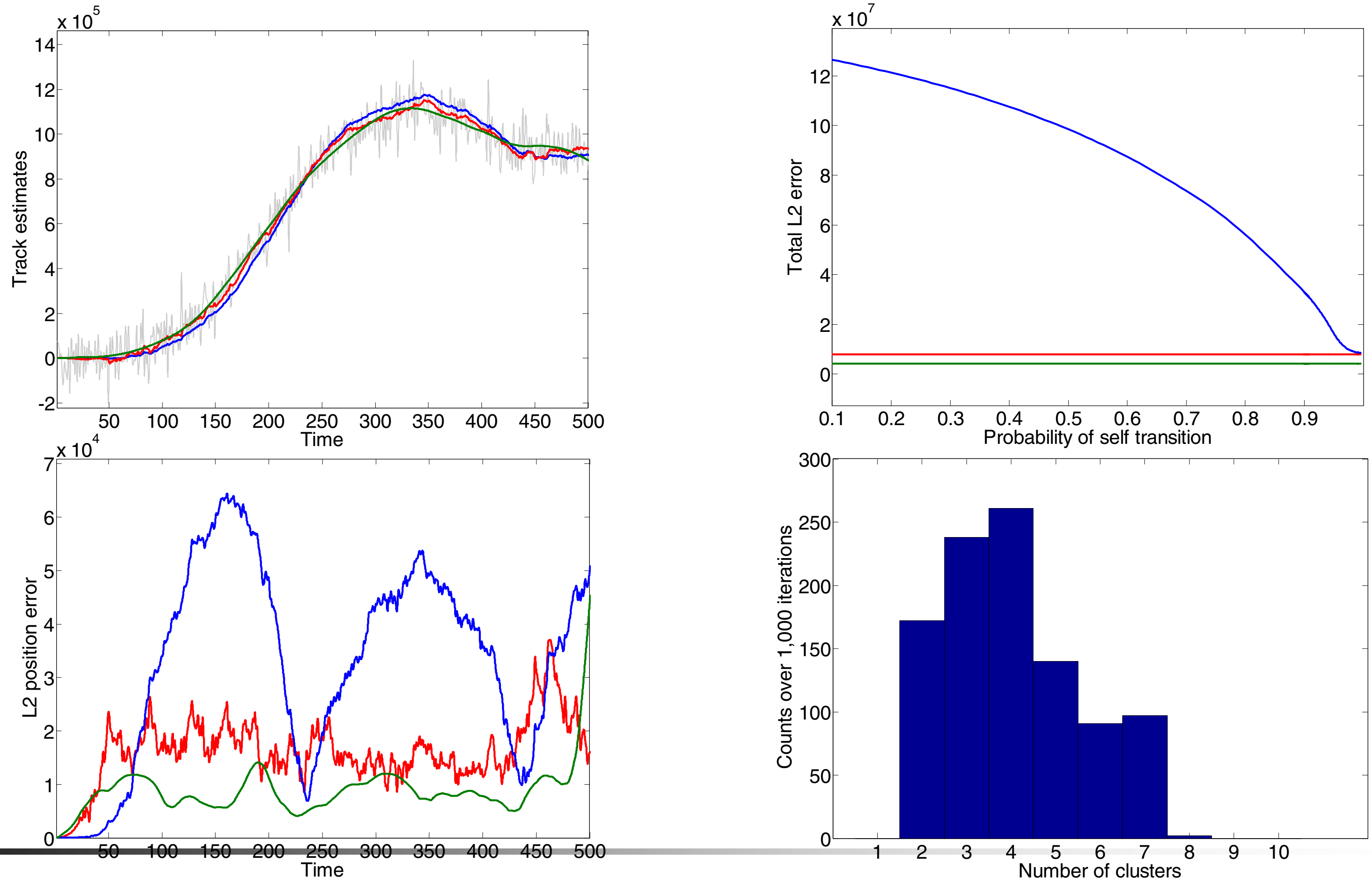
"Average" transition density which encourages states to transition back to a finite subset of the infinite state space

Mode-specific transition density

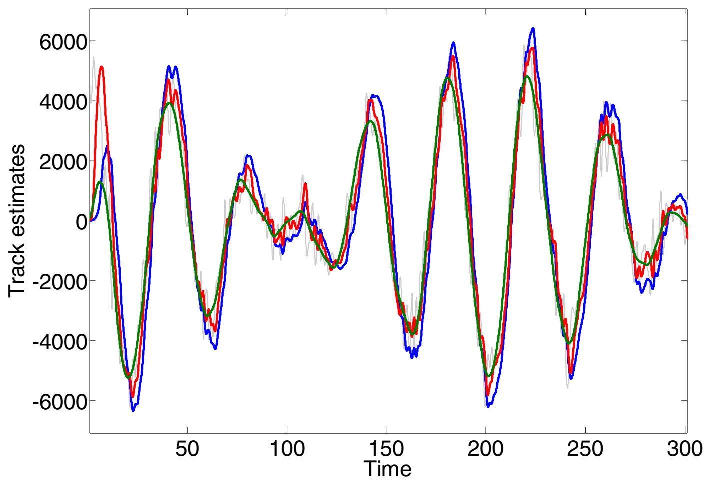
Learning and using HDP-based models

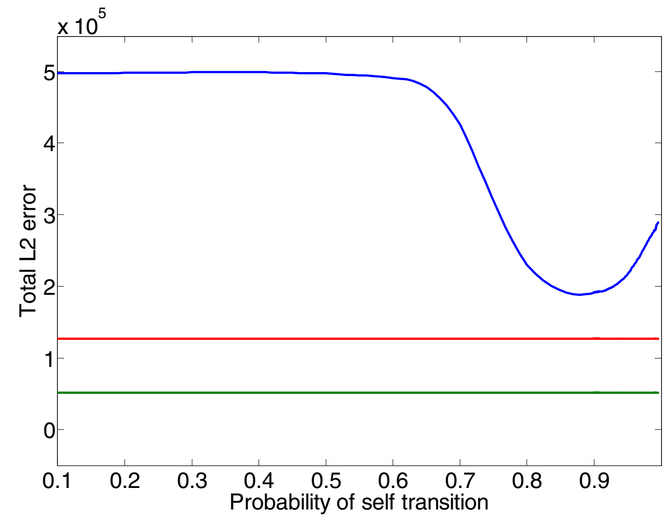
- Learning models from training data
  - Gibbs sampling-based methods
  - Exploit conjugate priors to marginalize out intermediate variables
  - Computations involve both forward filtering and reverse smoothing computations on target tracks
- Tracking
  - Use resulting learned model in an IMM-based tracker
  - Perform learning and tracking together
    - Currently that is accomplished in batch mode
    - Recursive methods are TBD

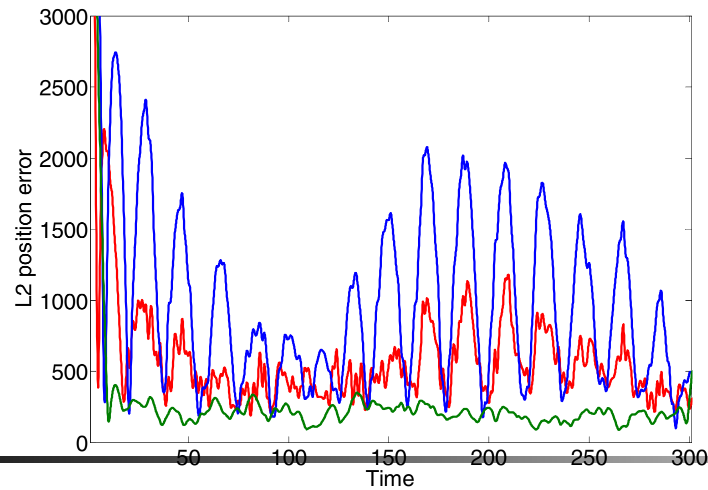
Numerical Experiment I

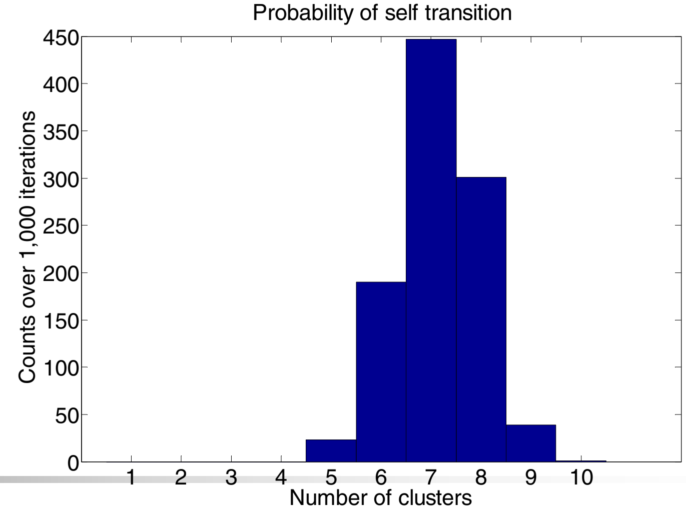


Numerical Experiment II









HDPs – where from here

- On-line, recursive algorithms
- HDPs for multi-target tracking and data association
- Learning coordinated motion models for multiple objects and for activities
- Use these for fusion of low-level object features for object recognition

What did we say at the kickoff? What have we done? - III

- Learning model structure
  - Discovering links (e.g., detecting coordination)
  - Exploiting and extending advances in learning (e.g., information-theoretic and manifold-learning methods) to build robust models for fusion
  - Direct ties to integrating signal processing products ***and*** to directing both signal processing and search
- Some of the accomplishments this year
  - ***Maximum entropy relaxation methods for learning sparse graphical models***
  - ***Learning graphical models directly for discrimination***

Learning sparse graphical models - I

- An alternative to learning via dimensionality reduction
  - Instead we seek ***complexity reduction***
- The setting
  - We have a possibly limited number of samples of a high-dimensional random phenomenon
    - E.g., multispectral images, multisensor observations, sets of multisensor features, etc.
  - From these we wish to construct a graphical model that
    - Is sparse (and tractable)
    - Is reasonably faithful to the observed data
  - We're working on two alternatives

Maximum Entropy Relaxation (MER) - I

- The basic ME problem
  - Build a model for a (high-dim.) random phenomenon $\boldsymbol{x}$ based on knowledge of some “local” statistics
    - $E[\phi_E(\boldsymbol{x}_E)] = \eta_E$
      - $E \in \mathcal{E}$, a set of subsets of components of $\boldsymbol{x}$
  - Find the probabilistic model, $p(x)$, that maximizes entropy $h(p)$ among all models that match these moments
  - Fact: If an optimal distribution exists, it is an element of the exponential family with features ϕ_E
    - $p(x) \propto \exp\{\sum_{E \in \mathcal{E}} \theta_E{}^T \phi_E(x_E)\}$
  - This distribution is Markov with respect to the graph with cliques given by $\mathcal{E}$ – Hence there is some intrinsic sparsification in ME
  - Note that the moments and features are dual parameters, so we can equally well refer to entropy as a function of the vector η

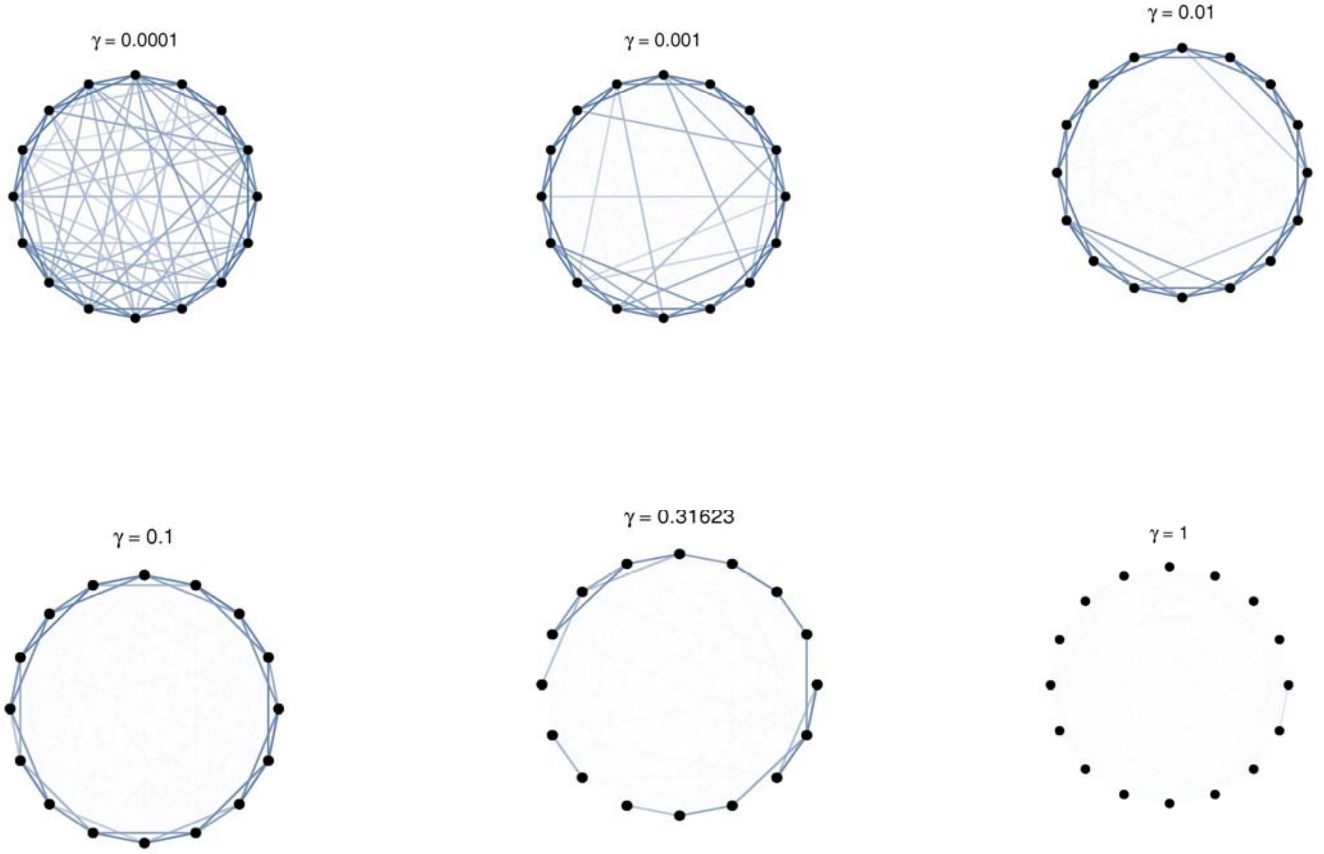
Maximum Entropy Relaxation (MER) - II

- MER: Why require exact matching of what are usually noisy estimates of statistics?
- Maximize entropy $h(\eta)$ subject to bounds on accuracy in matching specified moments, η^* – e.g,. In terms of KL-divergence, i.e., subject to inequality constraints:
  - $d_E(\eta, \eta^*) \leq \delta_E \qquad E \in \mathfrak{E}$
- MER does ***model thinning,*** yielding a model that is Markov with respect to the thinned graph corresponding to the active constraints
- How do we set δ_E ?
  - One approach, set these proportional to the cardinality of E (with proportionality constant γ)

Maximum Entropy Relaxation (MER) - III

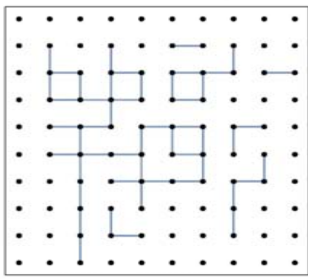
- Efficient iterative algorithms are under development
  - Primal-dual interior point method
    - Search directions involve solving linear system based on Fisher information matrix with respect to moment parameters
    - For thin chordal graphs, computing the Fisher information matrix and solving equations can be accomplished efficiently
    - Leads to an incremental approach, starting with disconnected graph and successfully computing chordal supergraphs
      - Solve reduced MER problem on each graph and check to see if constraints not yet included in this graph are satisfied or not
      - If satisfied, we're done
      - If not, need to find supergraph that includes still-violated constraints

Maximum Entropy Relaxation (MER) - IV

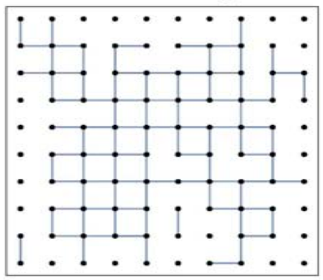


Maximum Entropy Relaxation (MER) - V

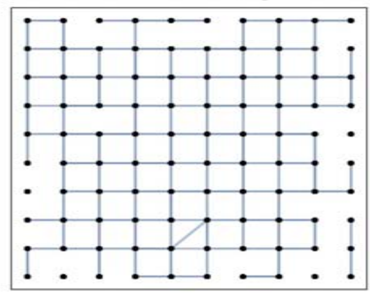
dim(G) = 150, dim(G_c) = 160



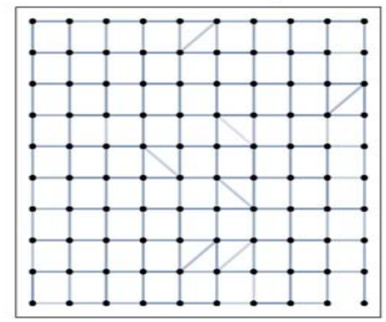
dim(G) = 200, dim(G_c) = 291



dim(G) = 250, dim(G_c) = 536



dim(G) = 287, dim(G_c) = 822



Maximum Entropy Relaxation (MER) - VI

- The way forward
  - The approach applies equally well to non-Gaussian data and models
  - Developing efficient algorithms for more general graphs
    - Tractable entropy approximations
    - Efficient algorithms a la Max-entropy for incremental construction of models as additional moments are included
  - Introducing latent variables
    - Ties to link discovery
  - Dealing with inconsistent measurement data
  - Blending of manifold learning and graphical modeling

Learning graphical models directly for discrimination - I

- If the ultimate objective of model construction is to use models for discrimination, why don't we ***design*** these models to optimize discrimination performance?
  - If there is an abundance of data, this really doesn't matter
  - However, for high-dimensional data and relatively sparse sets of data, there can be a substantial difference between learning a model for its own sake and learning one to optimize discrimination

Learning Graphical Models for Hypothesis Testing - II

$$H_0 : X \sim p \qquad H_1 : X \sim q$$

p and q not known, instead given iid labeled training sets T_0 & T_1

Traditionally: 1) learn $\widehat{p}$ from T_0 , $\widehat{q}$ from T_1

2) do likelihood ratio test (LRT) with $\widehat{p}, \widehat{q}$

We propose: learning $\widehat{p}, \widehat{q}$ jointly, each from both T_0, T_1

- $\widehat{p}, \widehat{q}$ sparse, testing via LRT

Low Complexity: learning structures of $\widehat{p}, \widehat{q}$ jointly, then projecting p, q

Higher Complexity: learning parameters as well.

Learning Graphical Models for Hypothesis Testing - III

Structure Learning

-Idea: would like $\log \frac{\widehat{p}(x)}{\widehat{q}(x)}$ to be large for $x \in T_0$, small for $x \in T_1$

Method: let $\widehat{p}, \widehat{q}$ be projections of p, q onto graphs chosen

to maximize $$\sum_{x \in T_0} \log \frac{\widehat{p}(x)}{\widehat{q}(x)} - \sum_{x \in T_1} \log \frac{\widehat{p}(x)}{\widehat{q}(x)}$$

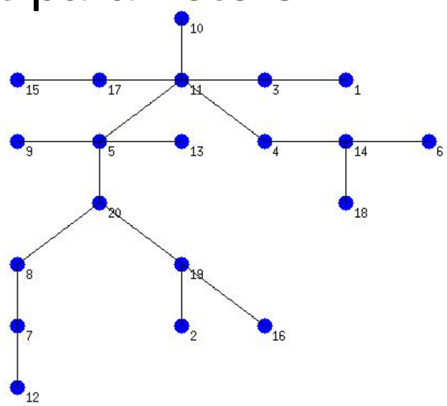
Decouples into two problems:

$$\min_{\widehat{p}} D(p_e || \widehat{p}) - D(q_e || \widehat{p}) \qquad \min_{\widehat{q}} D(q_e || \widehat{q}) - D(p_e || \widehat{q})$$

Learning Graphical Models for Hypothesis Testing - IV

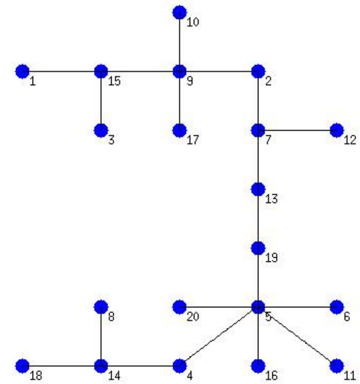
Example:

True p, q have same tree, different parameters

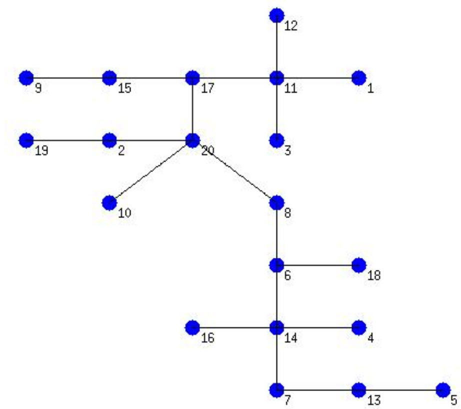


Traditional Pr(err) = **0.2000**
Our Pr(err) = **0.1585**

$\widehat{p}$



$\widehat{q}$



Trees of models NOT same as original

Learning Graphical Models for Hypothesis Testing - V

- Generally, our method has noticeably lower Pr(err) for large models with few training samples.

- Parameter learning by minimization of an upper bound on Pr(err)
  - convex programming

- On the horizon: Marriage of this approach with discriminative manifold learning work

What else is there?

- Informing resource management
  - Using informational structure of a graphical model to decide what evidence to gather
    - What nodes in discriminative graphical models should be sampled first?
    - What messages should be sent to perform discriminative inference efficiently?
- Some other accomplishments this year
  - Walk-sum analysis to optimize messaging in graphical inference
  - See other presentations

What's next

- More on scalable algorithms
  - Lagrangian relaxation, for example
- More on learning behavioral models and tracking
- More on learning tractable models for fusion and discrimination
  - Ties to low-level signal processing and feature extraction
    - E.g., to over-complete bases for wide-aperture SAR
  - Introducing hidden variables to capture hidden causes
- More on informing resource management
  - Which data should be gathered and fused
  - How to do this efficiently