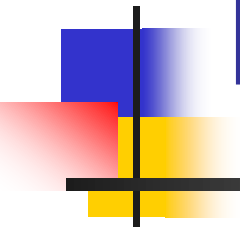
A Personal Journey: From Signals and Systems to Graphical Models and Machine Learning



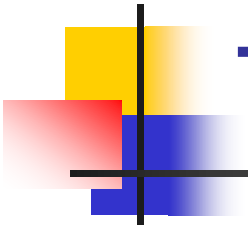
Alan S. Willsky

willsky@mit.edu

http://lids.mit.edu

http://ssg.mit.edu

December 2010

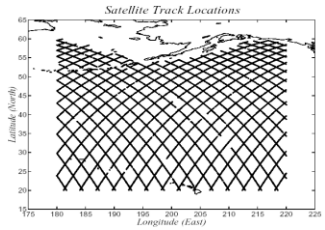


The Agenda: One group's path

- The launch (1988): Collaboration with Albert Benveniste and Michelle Basseville
  - Initial question: what are wavelets **really** good for (in terms that a card-carrying statistical signal processor would like)
    - Wavelets often introduced/used for ***analysis***
    - Statistical processing, however, usually begins with ***models*** – e.g., from ***synthesis***
  - This led us to stochastic models defined on multiresolution ***trees***

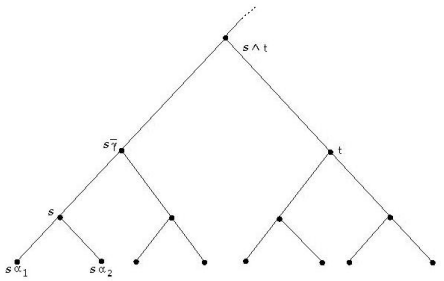
Why use multiresolution models?

- What can be multiresolution?
  - The ***phenomenon*** being modeled
  - The ***data*** that are collected
  - The ***objectives*** of statistical inference
  - The ***algorithms*** that result
- Some applications that motivated us (and others)
  - Oceanography
  - Groundwater hydrology
  - "Fractal priors" in regularization formulations in computer vision, mathematical physics, …
  - Texture discrimination
  - Helioseismology (?) …



Specifying MR models on trees

- MR ***synthesis***, leads, as with Markov chains, to thinking about ***directed trees***:



- E.g.: $x(s) = A(s)x(s\overline{\gamma}) + w(s)$
- Hidden Markov trees
- Models in which coarser variables are weighted averages of finer ones
- Midpoint deflection
- Note that the ***dimension*** of the variables comes into play

A control theorist's view - I

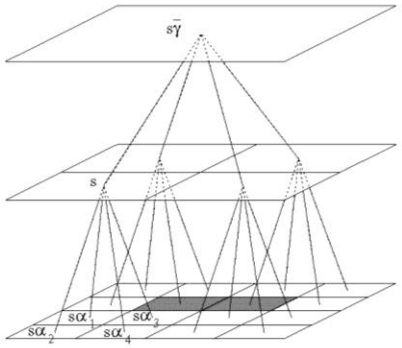
- It's a tree – so we all know that there are many ways to solve inference problems efficiently
- However, for a control theorist…
  - Upward sweep: (generalization of) Kalman Filter to compute estimates based on descendent subtrees
  - Generalizations of known Riccati equation/stability results**
  - Downward sweep: (generalization) of Rauch-Tung-Striebel smoothing equations
- Building models: (Approximate) Stochastic Realization
  - Base case: Process to be modeled resides at finest scale
  - Tree structure specified**
  - ***Dimensions*** of variables at hidden nodes are ***NOT**

A control theorist's idea: **Internal models**

- Variables at coarser nodes
  - All are ***linear functionals*** of the finest-scale variables**
  - Some of these may be given (corresponding to coarser-scale measurements to be taken or variables we wish to estimate)
  - The rest are to be ***chosen or designed***
    - To approximate the condition for tree-Markovianity
    - To yield a model that is "close" to the true fine-scale statistics
- Scale-recursive algebraic design**
  - Alternate using wavelets (in a minute)
- Confounding the control theorist
  - Internal models need not have minimal state dimension**

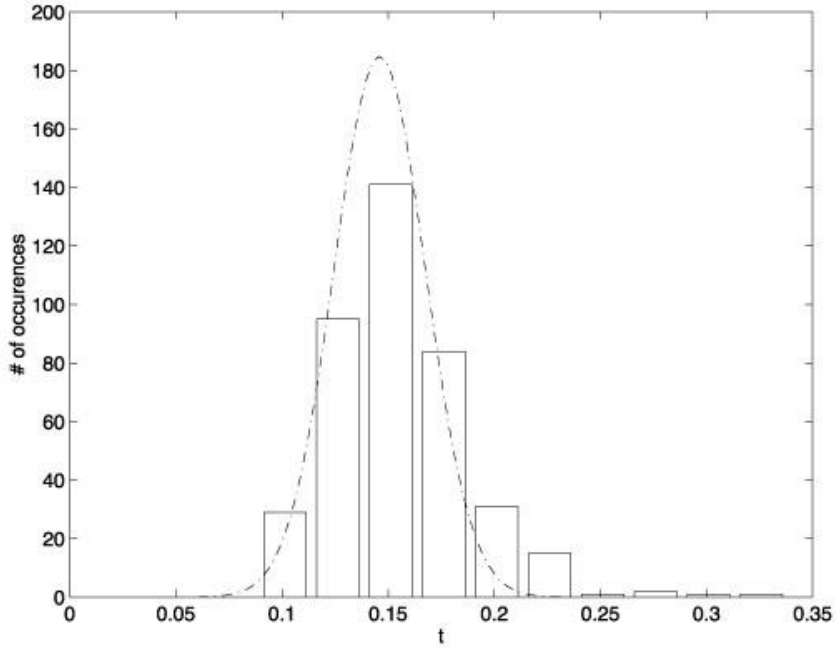
$$x(s) = x(s\overline{\gamma}) + w(s)$$

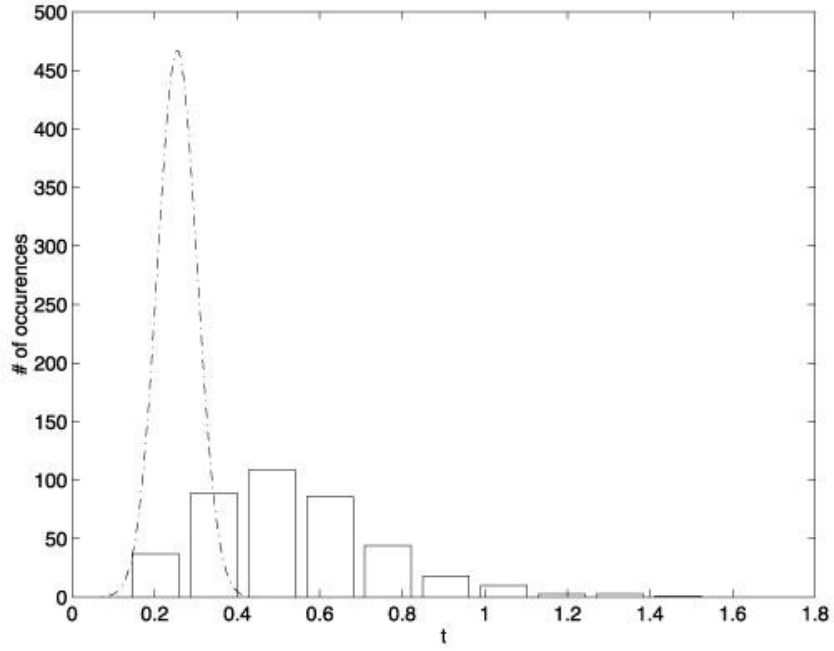
The dark side of trees = The bright side: No loops



- So, what do we do?
- Try #1: Pretend the problem isn't there
  - If the real objectives are at coarse scales, then fine-scale artifacts may not matter
- Try #2: Beat the dealer
  - Cheating: Averaging multiple trees
  - Theoretically precise cheating: Overlapping trees
  - Populate the tree with wavelets
- Try #3: Surrender
  - Put the &#%!*@# loops in!!

Let's pretend: Internal model for groundwater travel time estimation





- Can we do better in the fully nonlinear case?
  - Sure (he says casually) – probably with "particles" – e.g., NBP

Rapprochement with wavelets

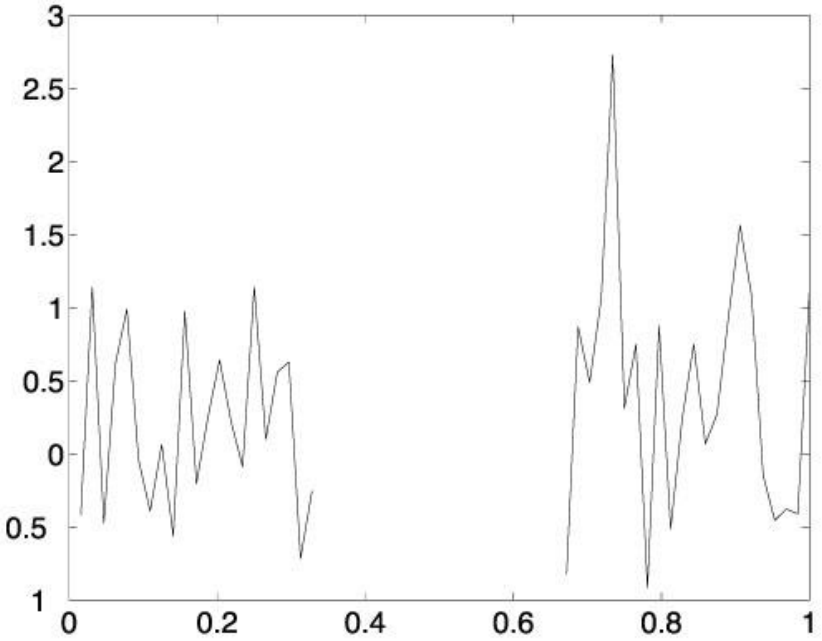
- Haar decompositions

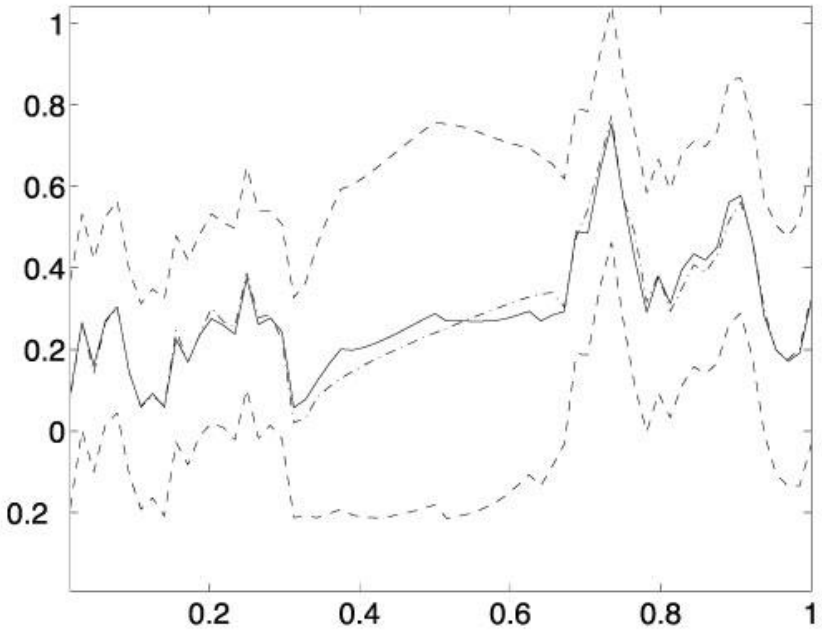
$$x(s\alpha_1) = x(s) + w(s\alpha_1)$$

$$x(s\alpha_2) = x(s) + w(s\alpha_2)$$

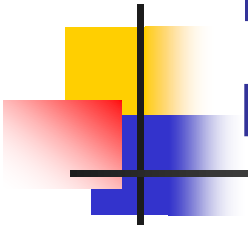
  - ***But the two w's are negatives of each other***
- To make this Markovian, need to have ***state*** at each node consist of both scaling and wavelet coefficient
  - Deterministic dynamics for coarse-to-fine scaling coeff.
  - Stochastic dynamics for finer wavelet coefficient given the scaling and wavelet coefficients at its parent
- Higher-order wavelets with overlapping support
  - Need more scaling/wavelet coefficients at each node for same reason
  - Key, however, is ***internality –*** Making fine-to-coarse dynamics deterministic ➔ We need ***additional*** coefficients at each node
  - For compactly supported wavelets: ***That's it – NO vicious cycle!***

This works pretty well!





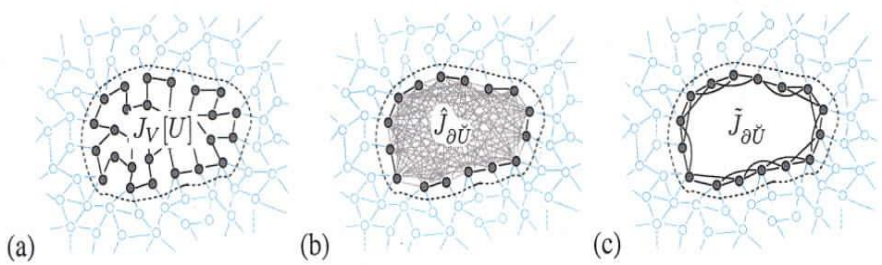
Borrowing/adapting a first idea from numerical solution of PDEs



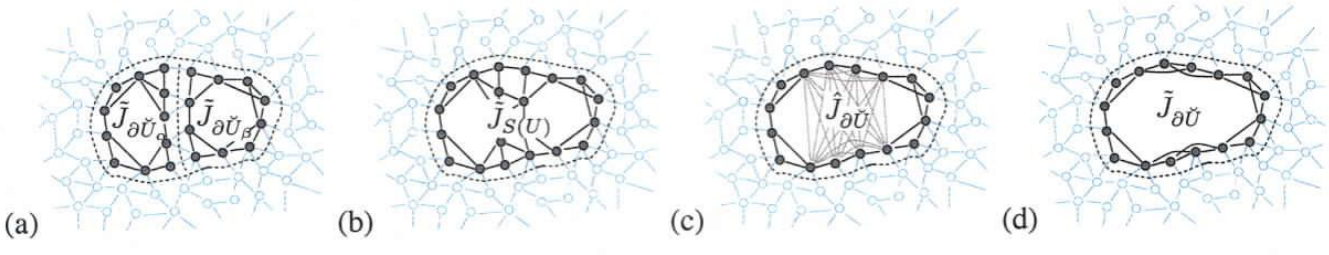
- Nested dissection
  - Cut the PDE graph via a nested set of separators
  - Solve the problem from small to large
  - Top-down has a problem
    - High-dimensional variables for large separators
- Recursive cavity models
  - Start from the ***bottom*** rather than top of the tree
  - Construct approximate ***models*** (i.e., inverse covariances)
  - Do this by variable elimination and ***model thinning***
    - Variable elimination introduces new edges due to "fill"
    - Remove edges so that the thinned model is optimally close to the unthinned model (convex optimization – sshhh….)

RCM in pictures

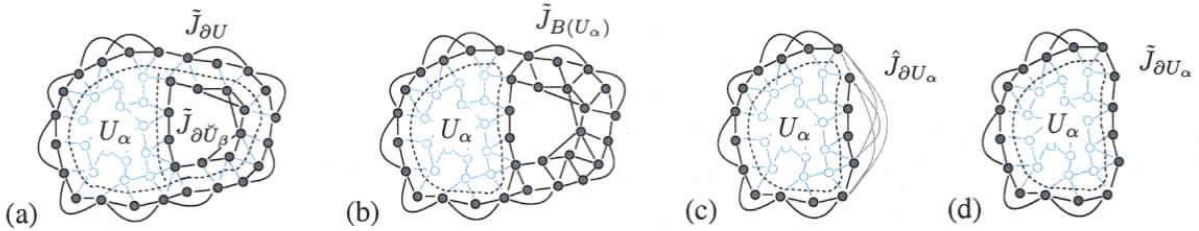
- Cavity thinning



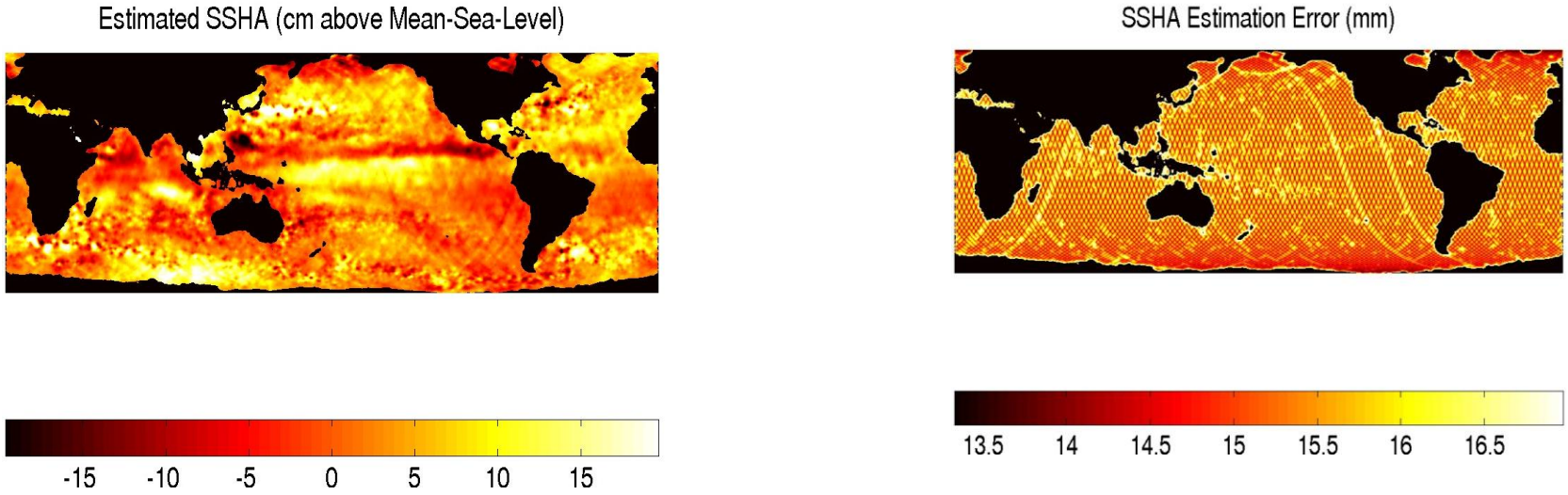
- Collision



- Reversing the process (bring your own blanket)



RCM in action: We are the world



- This is the information-form of RTS, with a thinning approximation at each stage
- How do the thinning errors propagate?
  A control-theoretic stability question
- We can iteratively refine estimates using
  Richardson iterations

Walk-sums, BP, and efficient message-scheduling for Gaussian models

$$X_V \sim \mathcal{N}(\mu, P)$$

- Information-form: $J = P^{-1} \quad h = P^{-1}\mu$
- Inference: Go from (*h, J*) to means and variances

$$J\widehat{x} = h$$

- Assume *J* normalized to have unit diagonal

$$J^{-1} = (I - R)^{-1} = I + R + R^2 + \ldots$$

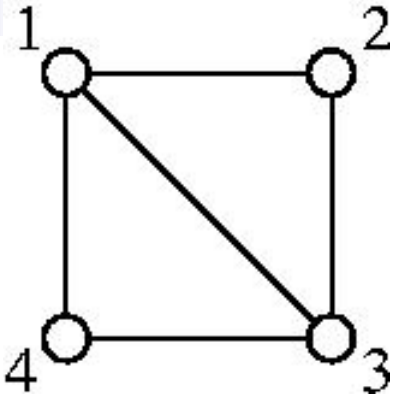
  - *R* is the matrix of ***partial correlation coefficients***, with sparsity pattern identical to *J*
  - $(R^\ell)_{s,t}$ corresponds to ***sum over weighted length-l walks*** from *s* to *t* in graph

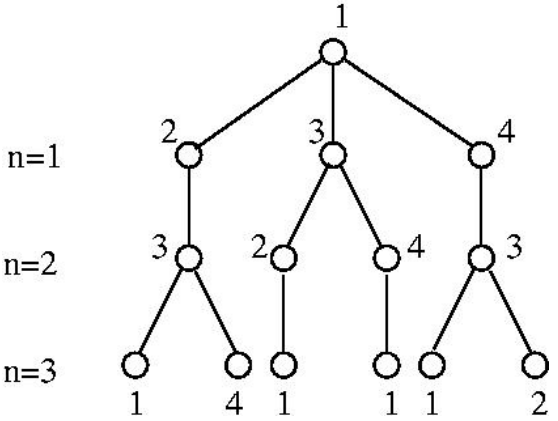
Walk-summable models

$$P_{s,t} = \phi(s \to t), \quad \mu_t = \sum_{s \in V} h_s \phi(s \to t)$$

- Inference algorithms may "compute" walks in arbitrary order (different message schedules)
  - Would like convergence ***regardless of order***
  - Condition of ***walk-summability*** $\varrho(\bar{R}) < 1$
- For BP
  - Walk-summability guarantees convergence
  - If BP converges it collects all walks for μ_i but only ***some*** of the ***self-return walks*** required for P_{ii}
- The ***computation tree*** provides insight about this and about what can happen for Non-WS models

A computation tree

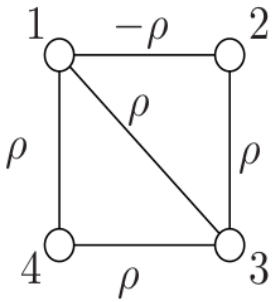




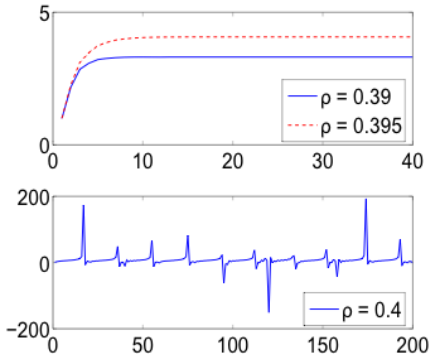
- BP includes the walk (1,2,3,2,1)
- BP does ***not*** include the walk (1,2,3,1)
- Note that back-tracking paths traverse each edge an even number of times
  - This implies that the *sign* of the ρ's has no effect on BP variance computations
- **BUT:** For Non-WS models, the tree **may be nonsensical**

A simple example

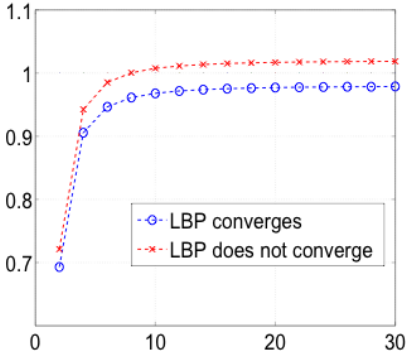
ρ	$\varrho(\bar{R})$	WS	ρ_∞	LBP vars	LBP means
.39	0.9990	Yes	< 1	Yes	Yes
.395	1.0118	**No**	< 1	Yes	Yes
.39867	1.0212	**No**	< 1	Yes	**No**
.4	1.0246	**No**	$\mathbf{> 1}$	**No**	**No**



G



LBP variances vs. iteration



$\varrho(R_n)$ vs. iteration.

Embedded "Trees"

- Preconditioned Richardson iterations
- Stationary $J = J_{\mathcal{S}} - K_{\mathcal{S}} \Rightarrow J_{\mathcal{S}}\widehat{x}_n = K_{\mathcal{S}}\widehat{x}_{n-1} + h$
  - Efficient if $\mathcal{S}$ corresponds to some tractable subgraph
- Non-stationary: $J_{\mathcal{S}_n}\widehat{x}_n = K_{\mathcal{S}_n}\widehat{x}_{n-1} + h$
  - ***Can significantly outperform stationary*** iterations

- Variations: Gauss-Seidel iterations, where we fix some variables and update others
  - Mixtures: Gauss-Seidel and then Richardson iteration for the ones to be updated

- For WS models: We get exact answers if all walks are collected

Choosing spanning trees

- Define $e_n = \widehat{x} - \widehat{x}_n;\ h_n = h - J\widehat{x}_n$

$$\|e_n\|_1 \leq \bar{\phi}(h_{n-1}; \mathcal{G}) - \bar{\phi}(h_{n-1}; \mathcal{S}_n)$$

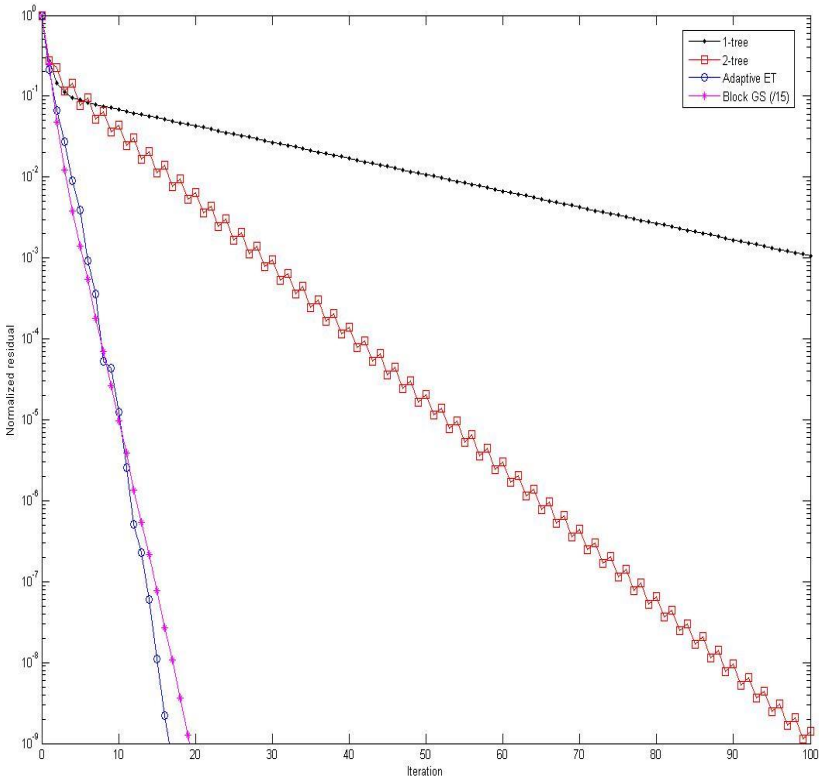
- Have a ***max-walksum-tree*** problem

$$\arg \max_{\mathcal{S}_n \ a\ tree} \bar{\phi}(h_{n-1}; \mathcal{S}_n)$$

- Seems intractable; minimize looser upper bound based on walks on single edge
  - ***Max-weight spanning tree*** with edge weights:

$$\frac{|R_{s,t}|}{1-|R_{s,t}|}\left(|(h_{n-1})_s| + |(h_{n-1})_t|\right)$$

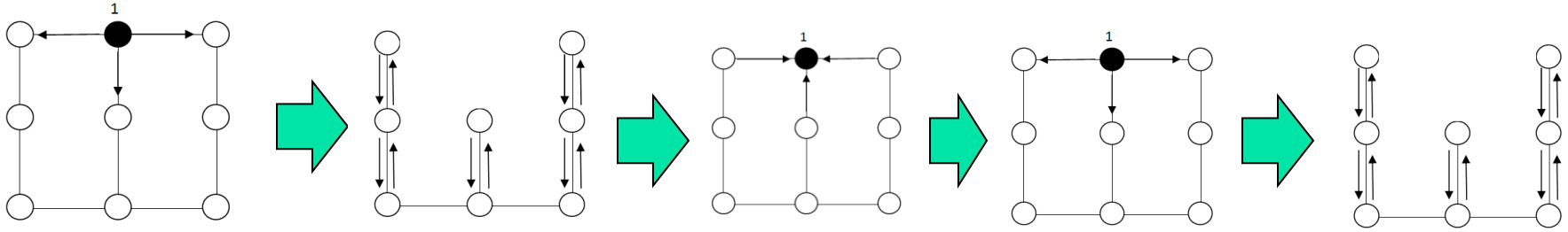
Performance on a 15×15 grid



Average number of iterations over 100 randomly generated 15x15 models to reduce normalized residual below 10^{-10}

Method	Avg. iterations
One-tree	143.07
Two-tree	102.70
Adaptive Max. Spanning Tree	44.04
Adaptive Block Gauss-Seidel (/15)	35.78

An alternate approach: Using (Pseudo-) ***Feedback Vertex Sets***



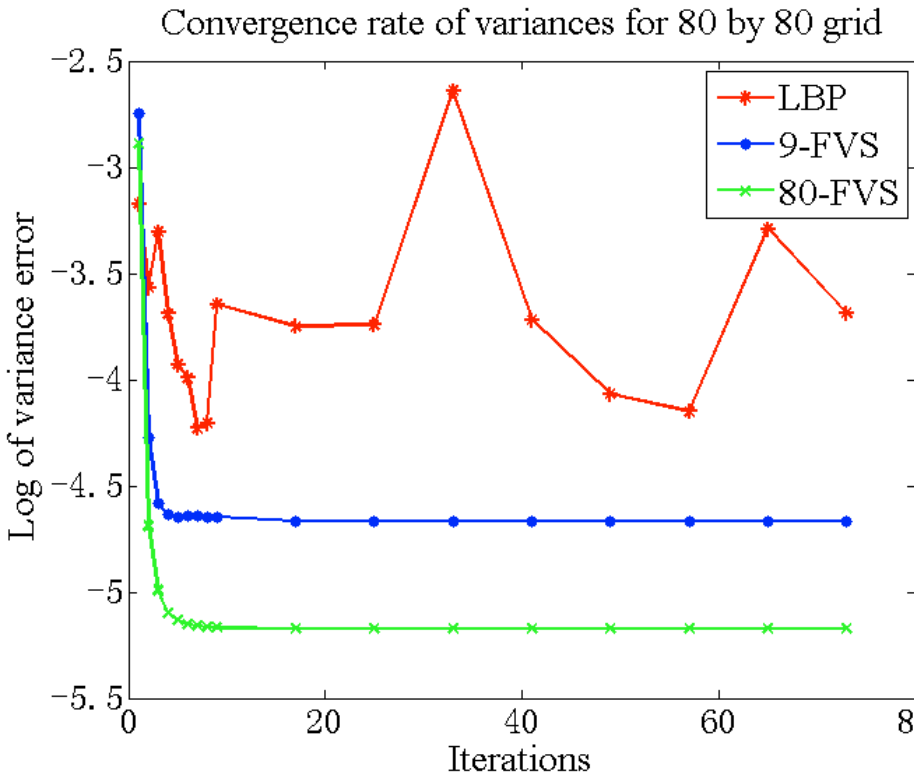
1. Provide additional potentials to allow computation of quantities needed in mean/variance/covariance computation in the FVS
2. Run BP with both original potentials and the additional set(s)
3. Feed back information to FVS to allow computation of exact variance and mean within the FVS
4. Send modified information potentials to neighbors of FVS
5. Run BP with modified information potentials
    1. Yields exact means immediately
    2. Combining with results from Steps 2, 3 yields exact variances

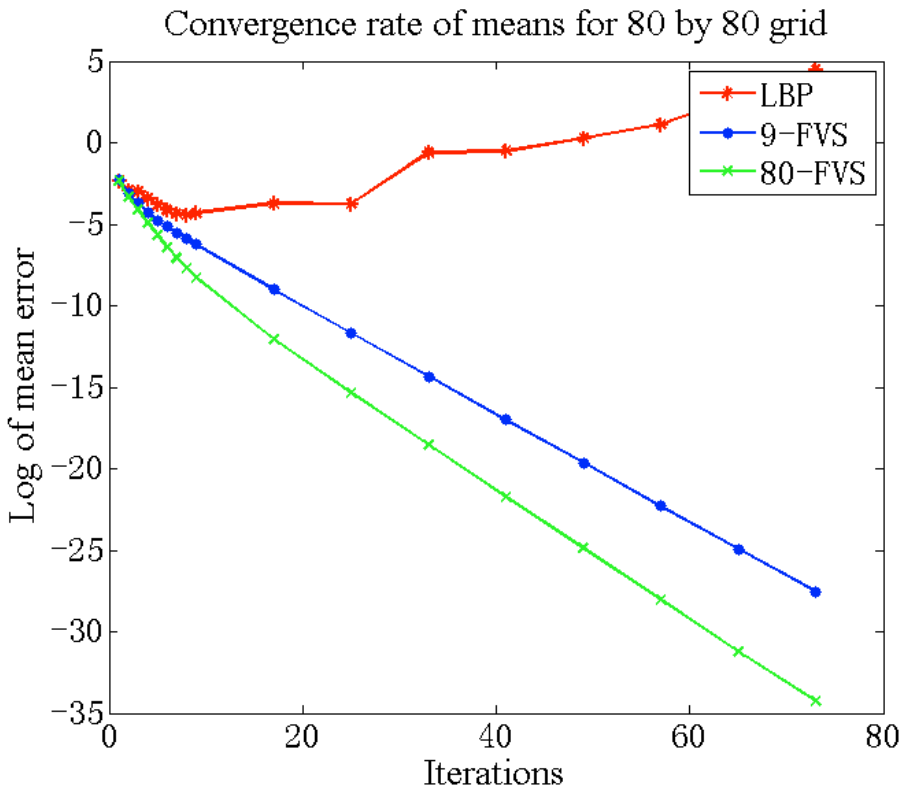
Approximate FVS

- Complexity is $\mathcal{O}(k^2 n)$, where $k = |\mathcal{F}|$
- If k is too large
  - Use a pseudo- (i.e., partial) FVS, breaking only some loops
  - On the remaining graph, run BP or some other algorithm – e.g., ET
- Assuming convergence
  - Always get the correct means and variances on $|\mathcal{F}|$, exact means on $\mathcal{T}$, and (for BP) approximate variances on $\mathcal{T}$
  - The approximate variances collect *more walks* than BP on the original graph
    - For attractive models, this yields better lower bounds
- Local (fast) method for choosing nodes for the pseudo-FVS to:
  - Enhance convergence
  - Collect the most important wants

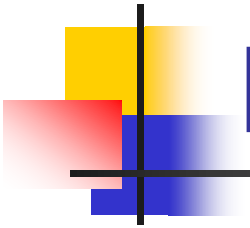
Illustration for 80x80 non-walk-summable 2-D grid

- Empirically, $k = \mathcal{O}(\log n)$ works well





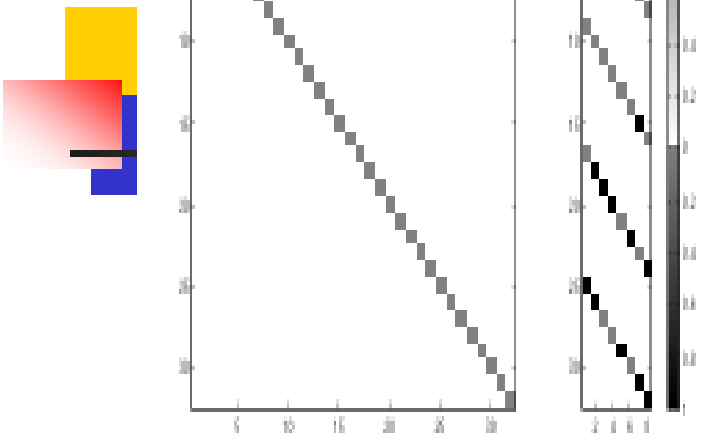
Low-rank variance estimation

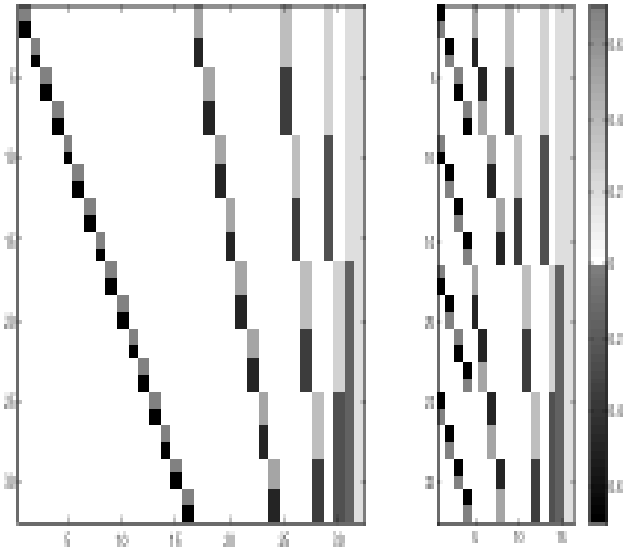


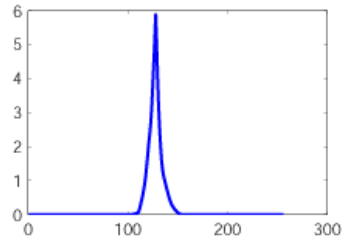
- Given inverse covariance, J, compute diagonal elements of $P = J^{-1}$
- An intractable approach
  - $JP = I = [e_1, \ldots, e_N]$
  - Solve column by column: $JP_i = e_i$, $i = 1, \ldots, N$
  - Each of these is $O(N)$ (for sparse graphs)
  - This is $O(N^2)$ – infeasible for large problems
- Let's 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 $P^a = RB^T$

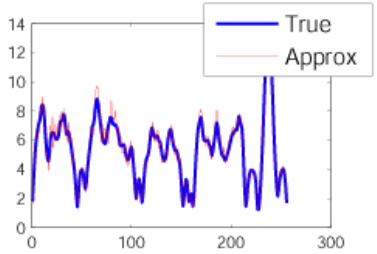
The key

- 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
    - A graph coloring problem if there is graphical correlation decay, with guarantees on accuracy
  - If there is slow correlation decay, we need another basis – Wavelets to the rescue



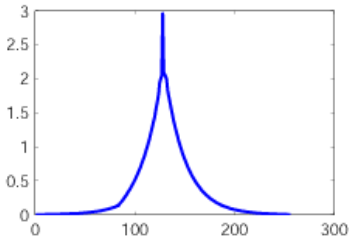


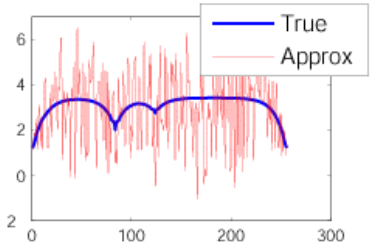




(a)

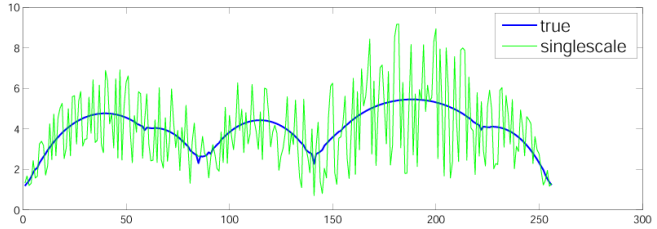
(b)

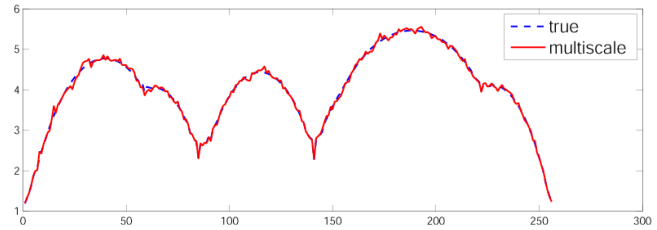




(c)

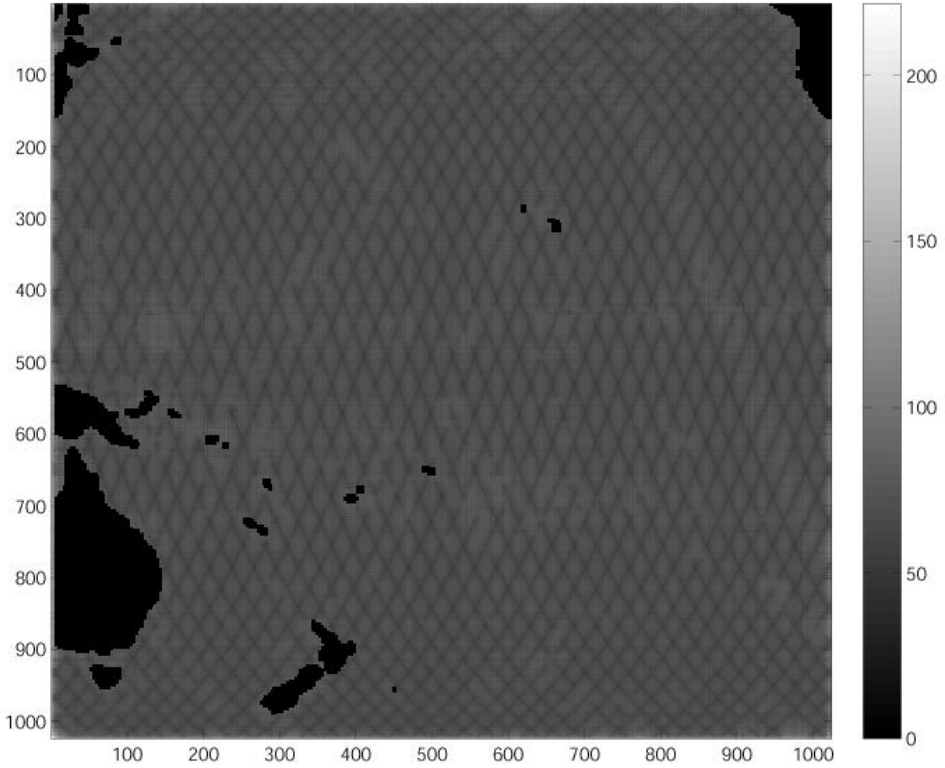
(d)



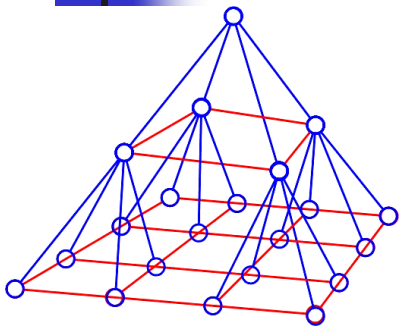


This can be ***very*** low rank

- $N = (1024)^2$ and $M = 448$



Motivation from PDEs, Part II: **Multigrid Models**



Models of this form can be created (via convex optimization (sshh…) with:

- Long-range correlations/dependencies
- Short range ***conditional correlations*** within each scale when conditioned on neighboring scales

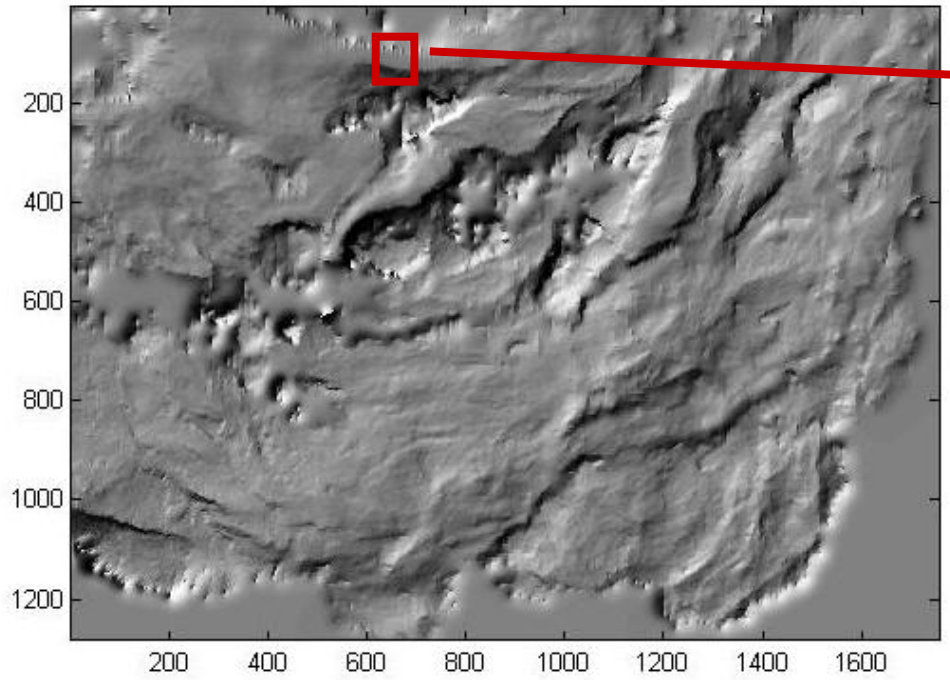
Multigrid algorithms

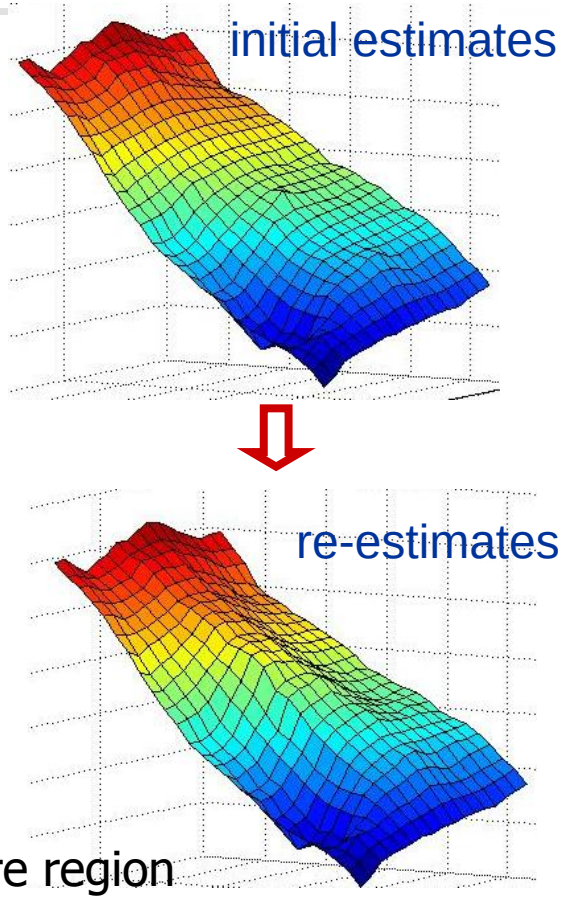
Fast fine-to-coarse initial processing – very fast
Coarse-to-fine local updates – very fast
Fine-to-coarse-to-fine iterative corrections

Adaptive algorithms

Use adaptive embedded subgraph algorithms to choose subgraphs to update to enhance rate of error reduction

Re-estimation Example





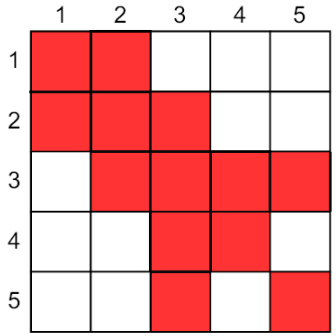
- 1757 X 1284 surface, 377384 measurements
- 3 million nodes in the pyramidal graph
- Introduce 100 new measurements in a 17 X 17 square region
- Use adaptive methods to update in 10 iterations, each of which involves fewer than 1000 nodes

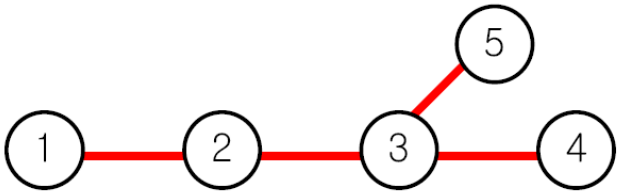
Motivation from PDEs, Part III: **MultiPOLE Models**

- Motivation from methods for efficient preconditioners for PDEs
  - Influence of variables at a distance are well-approximated by coarser approximation
  - We then only need to do ***LOCAL*** smoothing and correction
- The idea for statistical models:
  - Pyramidal structure in scale
  - However, when conditioned on neighboring scales, ***the remaining correlation structure at a given scale is sparse and local***

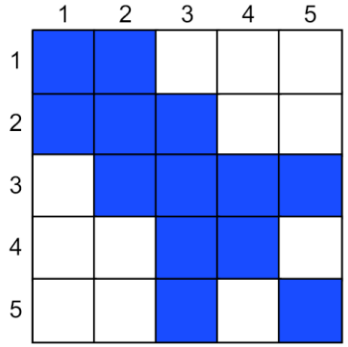
Models on graphs and on ***conjugate graphs***

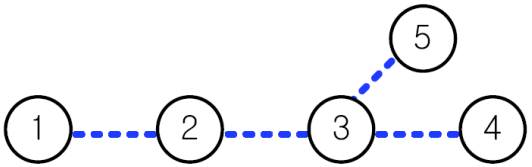
Garden variety graphical model: sparse ***inverse*** covariance

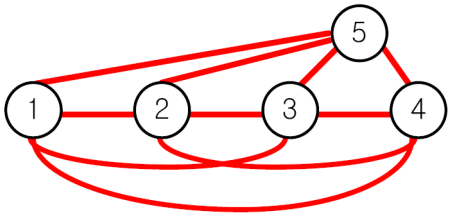




Conjugate models: sparse ***covariance***

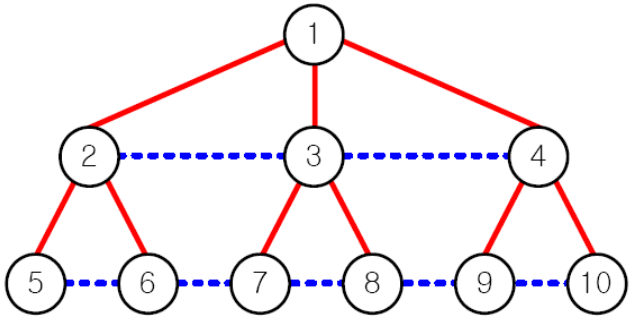






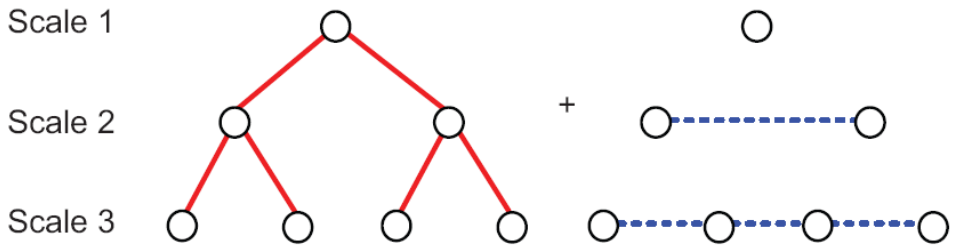
Sparse In-scale Conditional Covariance Multiresolution Model (SIM Model)

Scale 1



- Conditioned on scale 1 and scale 3, x_2 is independent of x_4.

$$J = \begin{pmatrix} J_{[1]} & J_{[1,2]} & 0 \\ J_{[2,1]} & J_{[2]} & J_{[2,3]} \\ 0 & J_{[3,2]} & J_{[3]} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & J_{[1,2]} & 0 \\ J_{[2,1]} & 0 & J_{[2,3]} \\ 0 & J_{[3,2]} & 0 \end{pmatrix}}_{J^h} + \underbrace{\begin{pmatrix} J_{[1]} & 0 & 0 \\ 0 & J_{[2]} & 0 \\ 0 & 0 & J_{[3]} \end{pmatrix}}_{J^c = (\Sigma^c)^{-1}}$$

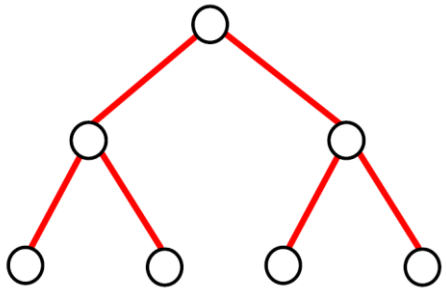


Learning such models:

"Dual" convex optimization problems (sshh)

Multipole Estimation

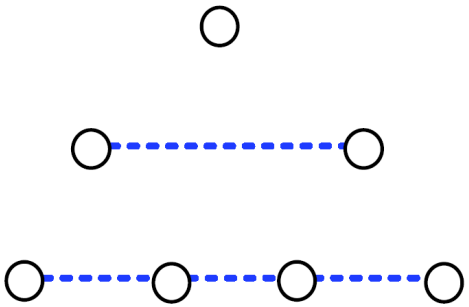
- Richardson Iteration to solve $(J^h + (\Sigma^c)^{-1})x = h$
  - Global tree-based inference
  - Sparse matrix multiplication for in-scale correction



$$J^h x_{new} = h - (\Sigma^c)^{-1} x_{old}$$

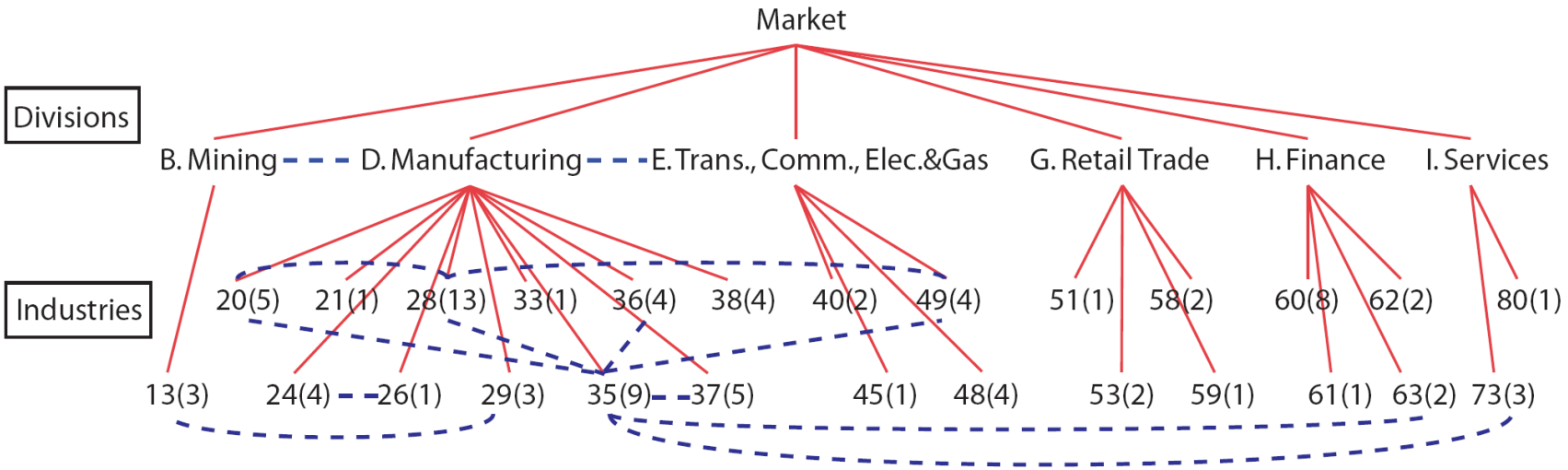
Compute last term via sparse equation

$$\Sigma^c z = x_{old}$$



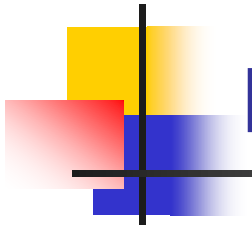
$$x_{new} = \Sigma^c (h - J^h x_{old})$$

Stock Returns Example



- Monthly returns of 84 companies in the S&P 100 index (1990-2007)
- Hierarchy based on the Standard Industrial Classification system
- Market, 6 divisions, 26 industries, and 84 individual companies
- Conjugate edges find strong residual correlations
  - Oil service companies (Schlumberger,…) and oil companies
  - Computer companies, Software companies, electrical equipment
  - …

A few other things on learning model structure



- Error exponents in learning tree models
  - For Gaussian models identification of models that are the hardest to learn (stars) and easiest (chains*)
- Algebraic realization of ***minimal*** hidden Markov trees
  - Exploits factorization properties, much as in state space realization theory
    - Although here it is the index set that is being discovered
  - Leads to relaxed algorithms akin to robust realization
- Blending graphical models and dimensionality reduction
  - Discovering ***sparse plus low-rank*** structure

What's on the horizon

- Gaussian models
  - Can we make the FVS-based algorithm completely "local" with appropriate "header" information in (multiple) message streams
  - Can we view this as a distributed dynamic system realization problem to collect walks?
  - Vector nodes
    - How do we assign dimensions and variables?
    - Walk-sums, etc., aren't as straightforward
- When messages really are messages
  - So there are ***two*** graphs (statistical and communications)
- Non-Gaussian
  - Multigrid, multipole, and realization of "hidden" models
  - Graph decomposition instead of sparse plus low-rank…