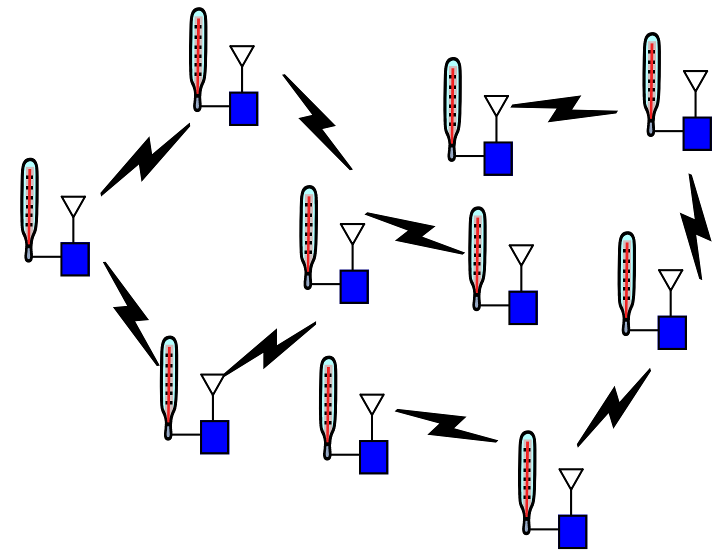
Coding Perspectives for Collaborative Estimation over Networks

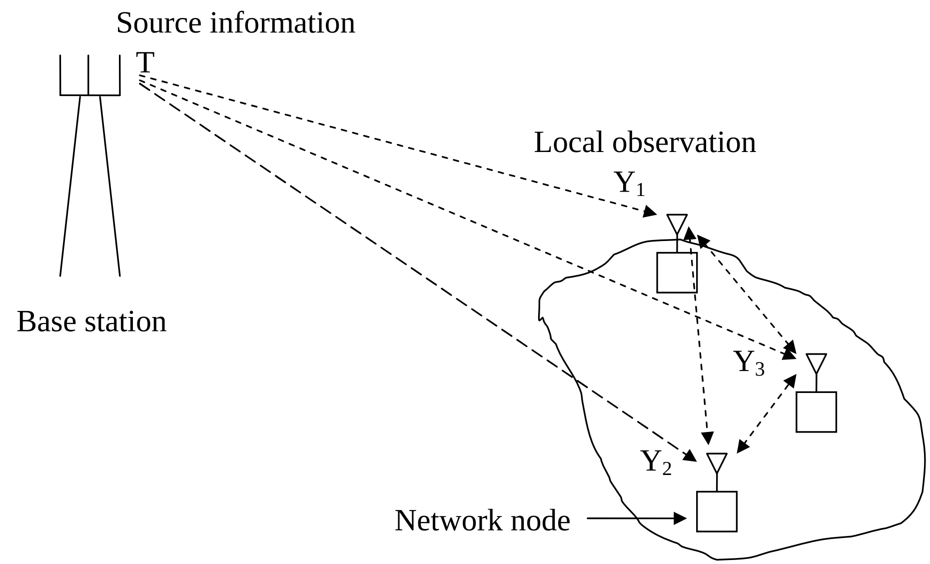
John M. Walsh & Sivagnanasundaram Ramanan
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, PA
jwalsh@ece.drexel.edu

Outline

1. What is collaborative estimation?
2. What are the major research issues/perspectives?
3. Collaborative Estimation from a Signal Processing/Machine Learning perspective
   (a) architecture for using belief/expectation propagation for collaborative estimation in sensor networks
   (b) example w/ benefits: channel gain estimation
4. Collaborative Estimation from an Information Theory/Coding Perspective
   (a) proper architecture for the lossy network source code
   (b) related known lossy source coding problems
   (c) inner and outer bounds on the rate distortion region
5. How might these perspectives be reconciled?
6. Future Work

What is collaborative estimation?

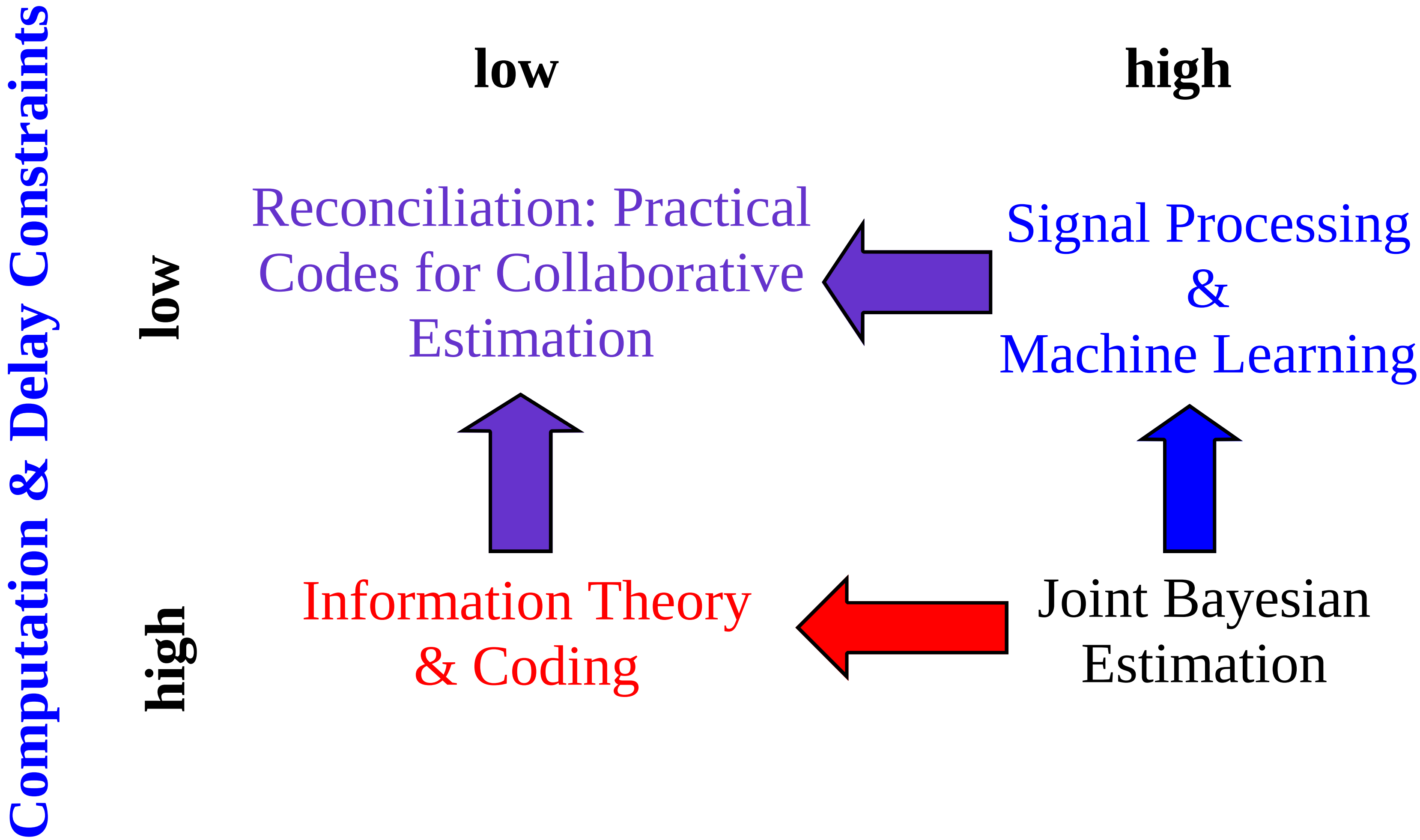




- M nodes. Node m w/ local observations $\mathbf{R}_m$.
- Collection of random parameters $\boldsymbol{T}$ jointly distrib. w/ $\{\mathbf{R}_m\}$
- Node m wants to estimate $\boldsymbol{T}$ with $\hat{T}_m$ to minimize a local Bayesian cost function, i.e. $d_m(\hat{T}_m, \boldsymbol{T})$ given avail. info.
- nodes share information over a network to help form their estimates

What are the major research issues/perspectives?

Communication Network & Energy Constraints



Communication Network & Energy Constraints

Computation & Delay Constraints

	low	high
low		
high		Joint Bayesian Estimation

Joint Bayesian Estimation

- each node broadcasts its observations $\mathbf{r}_m$ to all of the other nodes
- given $\mathbf{r} := [\mathbf{r}_m | m \in [M]]$ each node forms the posterior distribution $p_{\mathbf{T}|\mathbf{R}}(\mathbf{T}|\mathbf{r})$.
- each node chooses its estimate $\hat{\mathbf{T}}_m$ as the estimate minimizing its own Bayesian risk function

$$\hat{\mathbf{T}}_m \in \arg\min_{\hat{\mathbf{T}}_m} \int d_m(\hat{\mathbf{T}}_m, \mathbf{T}) p_{\mathbf{T}|\mathbf{R}}(\mathbf{T}|\mathbf{r}) \mathrm{d}\mathbf{T}$$

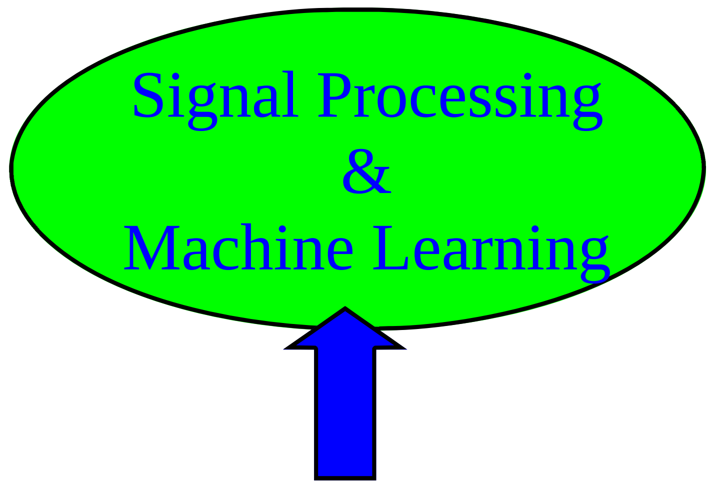
Communication Network & Energy Constraints

Computation & Delay Constraints	low	high
low		
high		Joint Bayesian Estimation

Communication Network & Energy Constraints

Computation & Delay Constraints

	low	high
low		Signal Processing & Machine Learning
high		



Joint Bayesian
Estimation

The Signal Processing/Machine Learning Perspective

$$\text{PROBLEM 1:} \quad \arg\min_{\hat{\mathbf{T}}_m} \int d_m(\hat{\mathbf{T}}_m, \mathbf{T}) p_{\mathbf{T},\mathbf{R}}(\mathbf{T}, \mathbf{r}) \mathrm{d}\mathbf{T} \quad \text{IS HARD!}$$

- one important major difficulty: the integration over the posterior distribution is usually difficult computationally and analytically, as can be the minimization of the local risk.
- enter approximate Bayesian inference: attempt to (iteratively) approximate the posterior distribution within a particular exponential family of distributions
  - $\Rightarrow$ Expectation/Belief Propagation
- How can we organize network communications to keep energy consumption low, but yield message passing that gets the cases where belief propagation gives good estimates?

Expectation Propagation (EP), I [1, 2, 3]

- parameters $\mathbf{T}$ whose a.p.d.'s we want
- observations $\mathbf{r}$
- joint stat. model that factors $\boldsymbol{\theta}_a \subseteq \boldsymbol{\theta}$

$$\mathsf{p}_{\mathbf{r},\mathbf{T}}(\mathbf{r}, \boldsymbol{\theta}) \propto \prod_{a=1}^{M} \mathsf{f}_{a,\mathbf{r}}(\boldsymbol{\theta}_a)$$

- Goal: calculate $\boldsymbol{\lambda}_a(\mathbf{r})$ to approximate

$$\mathsf{p}_{\mathbf{T}|\mathbf{r}}(\boldsymbol{\theta}|\mathbf{r}) \approx \prod_{a=1}^{M} \mathsf{g}_{a,\boldsymbol{\lambda}_a(\mathbf{r})}(\boldsymbol{\theta}_a)$$

- $\mathsf{g}_{a,\boldsymbol{\lambda}_a(\mathbf{r})}(\boldsymbol{\theta}_a) \propto \exp\left(\mathbf{h}_a(\boldsymbol{\theta}_a) \cdot \boldsymbol{\lambda}_a(\mathbf{r})\right)$
- Designer selects $\mathbf{h}(\cdot) := [\mathbf{h}_a(\cdot)]$
- Given design + $\mathbf{r}$, EP $\rightarrow \boldsymbol{\lambda}_a(\mathbf{r})\ \forall a$.

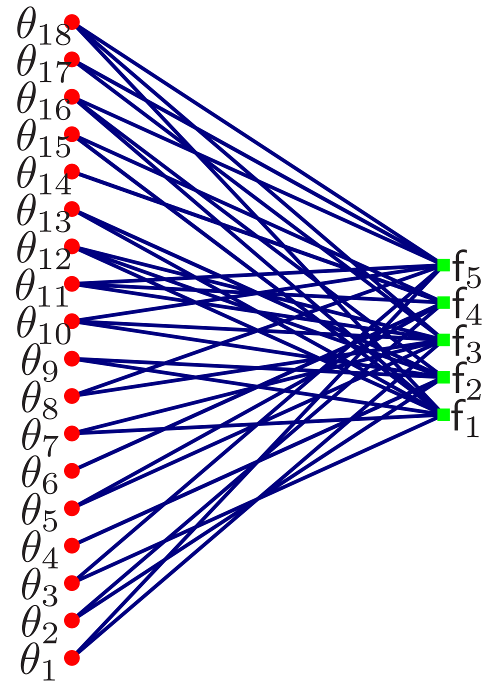


Figure 1: The parameter factor graph.

EP, II. Design Choices: The log-basis Functions $\mathbf{h}(\boldsymbol{\theta})$

- In choosing $\mathbf{h}(\cdot)$ one trades between
  1. Accuracy: level of stat. dep. amongst $T_{\mathtt{i}}$ allowed in approx.
  2. Complexity: control amnt of computation + comm. req.'d
- Requirement: Sufficiency $\forall \boldsymbol{\theta}_{\mathtt{a}}$

$$\mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) = \hat{\mathsf{f}}_{\mathtt{a}}(\mathbf{h}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}))$$

  so all information $\mathsf{f}_{\mathtt{a}}$ depends on is in $\mathbf{h}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}})$.
- Requirement: Reciprocity $\mathbf{h}_{\mathtt{a}}$s are concatenations of $\mathbf{v}_{\mathtt{i}}(\boldsymbol{\vartheta}_{\mathtt{i}})$ with each $T_{\mathtt{j}}$ in only one $\boldsymbol{\vartheta}_{\mathtt{i}}$.
- $\implies$ everywhere we are approximating $T_{\mathtt{i}}$ we are using the same type of density.

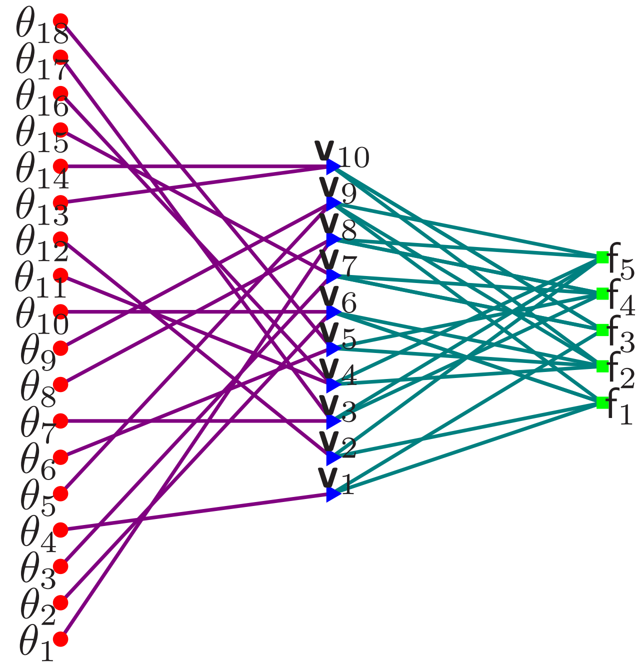


Figure 2: The parameter statistics factor graph.

EP, III. Model Selection via h(θ) : Examples

Discrete Θ (recall $\mathbf{T} \in \Theta$): Consider $\Theta = \{0,1\}^N$

- **independent bits: h$(\boldsymbol{\theta}) = \boldsymbol{\theta}$**
- **pairwise dependent bits:**

$$\mathbf{h}(\boldsymbol{\theta}) = [T_1, \cdots, T_{\mathtt{N}}, T_1T_2, T_1T_3, \cdots, T_1T_{\mathtt{N}}, T_2T_3, \cdots, T_2T_{\mathtt{N}}, \cdots, T_{\mathtt{N}-1}T_{\mathtt{N}}]^T$$

Continuous Θ: Consider $\Theta = \mathbb{R}^N$

- $\{T_{\mathtt{i}}\}$ **independent Gaussian:**

$$\mathbf{h}(\boldsymbol{\theta}) := \left[T_1, T_1^2, T_2, T_2^2, \cdots, T_{\mathtt{N}}, T_{\mathtt{N}}^2\right]^T$$

- $\{T_{\mathtt{i}}\}$ **jointly normal:**

$$\mathbf{h}(\boldsymbol{\theta}) := \left[T_1, T_1^2, T_2, T_2^2, \cdots, T_{\mathtt{N}}, T_{\mathtt{N}}^2, T_1T_2, T_1T_3, \cdots, T_1T_{\mathtt{N}}, T_2T_3, \cdots, T_2T_{\mathtt{N}}, \cdots, T_{\mathtt{N}-1}T_{\mathtt{N}}\right]^T$$

Other possible distribution types: exponential, beta, gamma, Poisson, any finite distribution

EP, IV: The Message Passing Algorithm

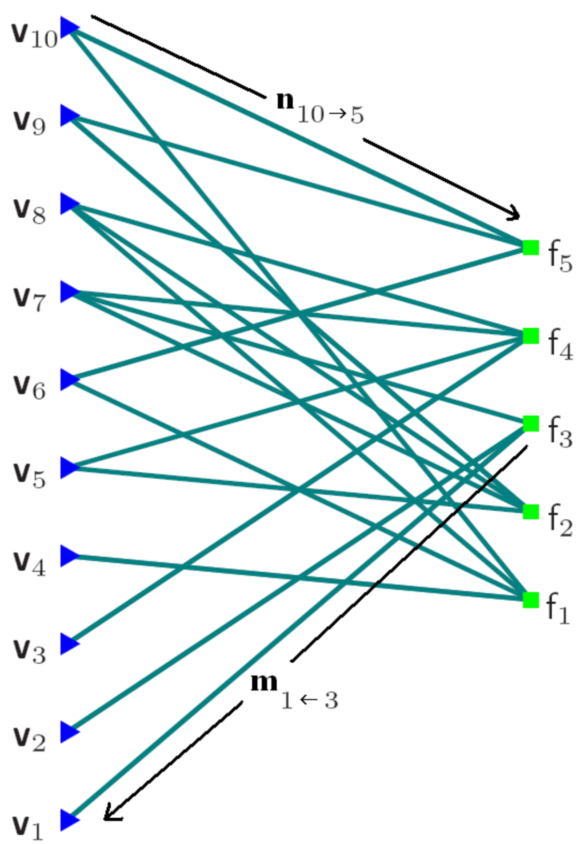


Figure 3: Message Passing.

- Try to make the approx. by passing messages
- right going messages

$$[\boldsymbol{\lambda}_{\mathrm{in}}]_{\mathtt{i}} := \sum_{\mathtt{c}\in\mathcal{N}(\mathtt{i})\setminus\{\mathtt{a}\}} [\boldsymbol{\lambda}_{\mathtt{c}}]_{\mathtt{i}} =: [\mathbf{n}_{\mathtt{j}\to\mathtt{a}}]_{\mathtt{i}}$$

- left going messages

$$\mathbf{m}_{\mathtt{j}\leftarrow\mathtt{a}} := \left[[\boldsymbol{\lambda}_{\mathtt{a}}]_{\mathtt{i}} \,|\mathtt{h}_{\mathtt{i}} \in \mathbf{v}_{\mathtt{j}}\right]$$

$$\frac{\int_{\Theta_a} \mathbf{h}_a(\boldsymbol{\theta}_a)\hat{f}_a(\mathbf{h}_a(\boldsymbol{\theta}_a)) \exp\left(\mathbf{h}_a(\boldsymbol{\theta}_a)\cdot\boldsymbol{\lambda}_{\mathrm{in}}\right) d\boldsymbol{\theta}_a}{\int_{\Theta_a} \hat{f}_a(\mathbf{h}_a(\boldsymbol{\theta}_a)) \exp\left(\mathbf{h}_a(\boldsymbol{\theta}_a)\cdot\boldsymbol{\lambda}_{\mathrm{in}}\right) d\boldsymbol{\theta}_a}$$
$$= \frac{\int_{\Theta_a} \mathbf{h}_a(\boldsymbol{\theta}_a) \exp\left(\mathbf{h}_a(\boldsymbol{\theta}_a)\cdot(\boldsymbol{\lambda}_{\mathrm{in}}+\boldsymbol{\lambda}_a)\right) d\boldsymbol{\theta}_a}{\int_{\Theta_a} \exp\left(\mathbf{h}_a(\boldsymbol{\theta}_a)\cdot(\boldsymbol{\lambda}_{\mathrm{in}}+\boldsymbol{\lambda}_a)\right) d\boldsymbol{\theta}_a}$$

When can you prove the approximation is good?

- When the factor node updates are equivalent to passing parameters for the associated marginal density, EP $=$ belief propagation.[4, 5]
- In this case if the factor graph is a cycle free, BP gives exact marginal distributions for the (clustered) Θ_i
- Since it is a local message passing algorithm, if a particular computation neighborhood of size 2ℓ is a tree, then an exact posterior is calculated over data available in that neighborhood. [6, 7]

How can this be used to simplify the Risk Minimization/Calculation?

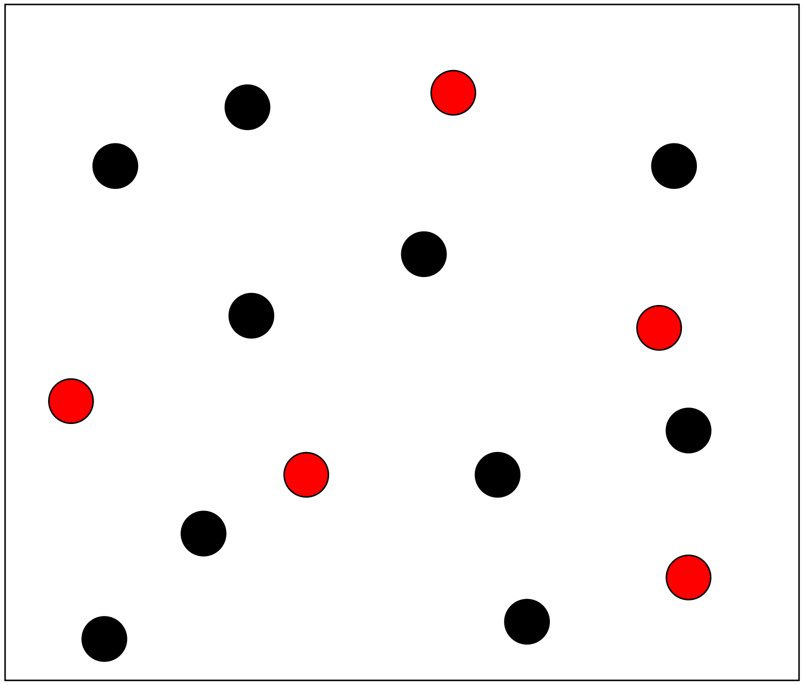
$$\arg\min_{\hat{\mathbf{T}}_m} \int d_m(\hat{\mathbf{T}}_m, \mathbf{T}) p_{\mathbf{T},\mathbf{R}}(\mathbf{T}, \mathbf{r}) \mathrm{d}\mathbf{T}$$

- if d_m depends only on a certain collection of elements of $\mathbf{T}$ Θ_m, then can select a factoring & approximating family to get marginals for Θ_m from BP.
- More broadly, if there are parts of the posterior which yield risk computation difficult, they can be approximated with exponential families in which it is simple (e.g. Gaussians). [8, 9]
- Factoring can be set up to give tree like neighborhoods via random duty cycling, as shown in the next example. [10, 11]
- The message passing nature of the algorithm describes a way to handle decentralization of the data (group it with factor nodes). [12, 13, 14, 15, 16]

Example: Wireless Sensor Network Initialization

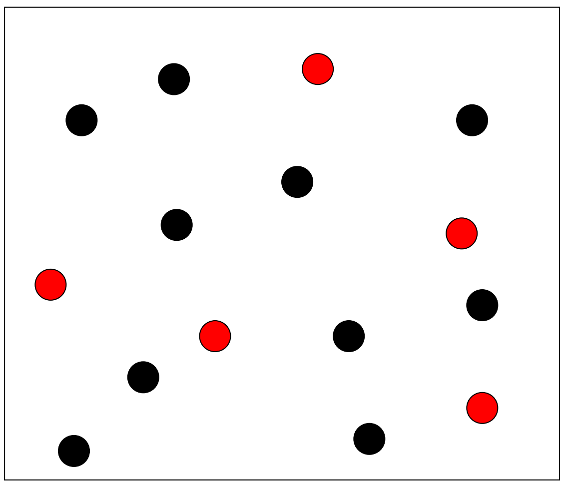
- Plane flies over forest/ field of interest, drops many sensor nodes.
- Placement is random, nodes do not know: positions, their neighbors, nor the gains in the wireless channels between them.
- Nodes must conserve power $\Longrightarrow$ duty cycling a must, but can not yet be done in an organized fashion.
- Organization performed in network, egalitarian (non-hierarchical) structure.
- Each node would like to organize communications in an energy efficient manner (power control, MAC, and routing), but this requires knowledge of channel gains.
- Initialization Phase: Using a random sleep (duty cycling) strategy for communications, estimate the wireless channel gains in the network.

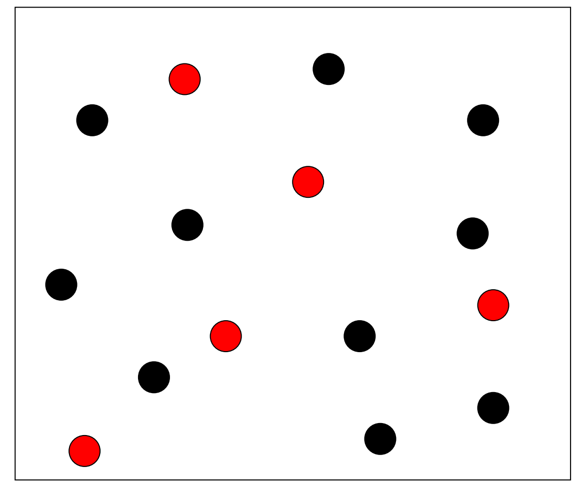
Duty cycling to the network



- Keep only a small subset of sensors "awake" at each time instant

Regular cyclic random sleep strategy to the network





- Regular cyclic random sleep strategy: a random subset of nodes are awake at a time and the sleep pattern repeats after certain amount of time
- At each time instant same number of nodes are awake and each maintains the same average power consumption

Regular cyclic random sleep strategy to the network

- d nodes are awake at a time instant
- Each node is awake c times in a sleep cycle
- K time instants in a sleep cycle

$$K = \frac{c}{d} N$$

- Define the set of nodes awake at time instant k to be $\{\mathcal{S}(k) | k \in \{1, \ldots, K\}\}$

Model for the channels

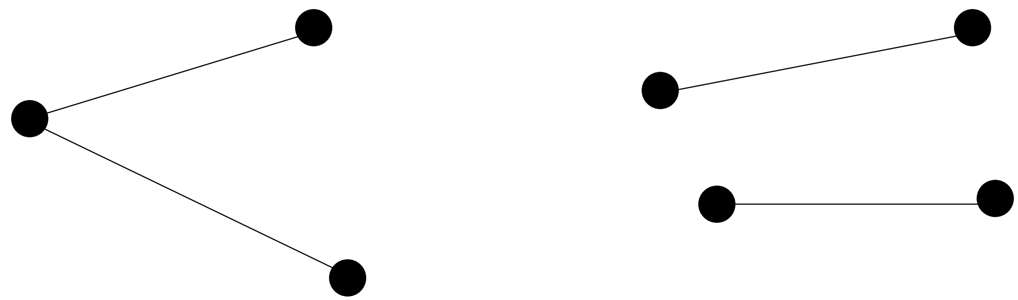
- Channel gain of a link between any two nodes heavily depends on the distance between them
- Pathloss model

$$h \propto R^{-n}$$

  where pathloss exponent n $(2 \leq n \leq 6)$
- Gain of the link between node i and node j

$$h_{i,j} \propto \|\mathbf{x}_i - \mathbf{x}_j\|_2^{-4}$$

The prior joint distribution of the channels



- Any two channel gains incident on the same node are dependent

$$\text{dependent:} \qquad h_{i,j} \propto \|\mathbf{x}_i - \mathbf{x}_j\|_2^{-4} \qquad h_{i,m} \propto \|\mathbf{x}_i - \mathbf{x}_m\|_2^{-4}$$

$$\text{independent:} \qquad h_{i,j} \propto \|\mathbf{x}_i - \mathbf{x}_j\|_2^{-4} \qquad h_{m,n} \propto \|\mathbf{x}_m - \mathbf{x}_n\|_2^{-4}$$

- The prior joint distribution of the channel gains is analytically complex, because of the inverse nonlinear dependence on the node positions
- Under EP, we approximate it with a Gaussian with the same mean and covariance
- Ability to exploit this prior information is key from a *network* perspective.

Channel Training

- Train the channel using a training sequence $u_1, \ldots, u_M$
- Model the observation as

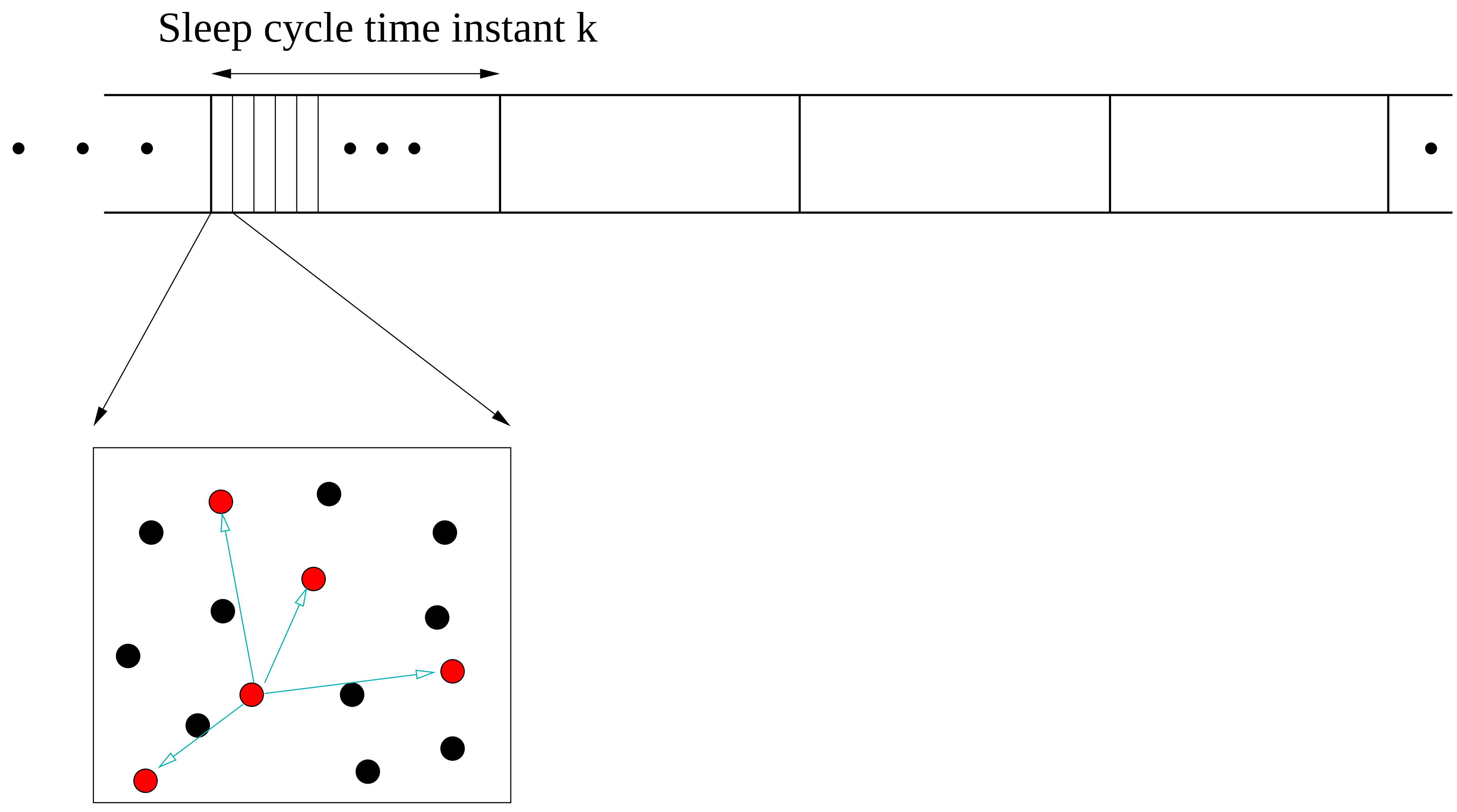
$$r_m = hu_m + v_m$$

where v_m is Gaussian distributed noise, which is i.i.d. over time and space

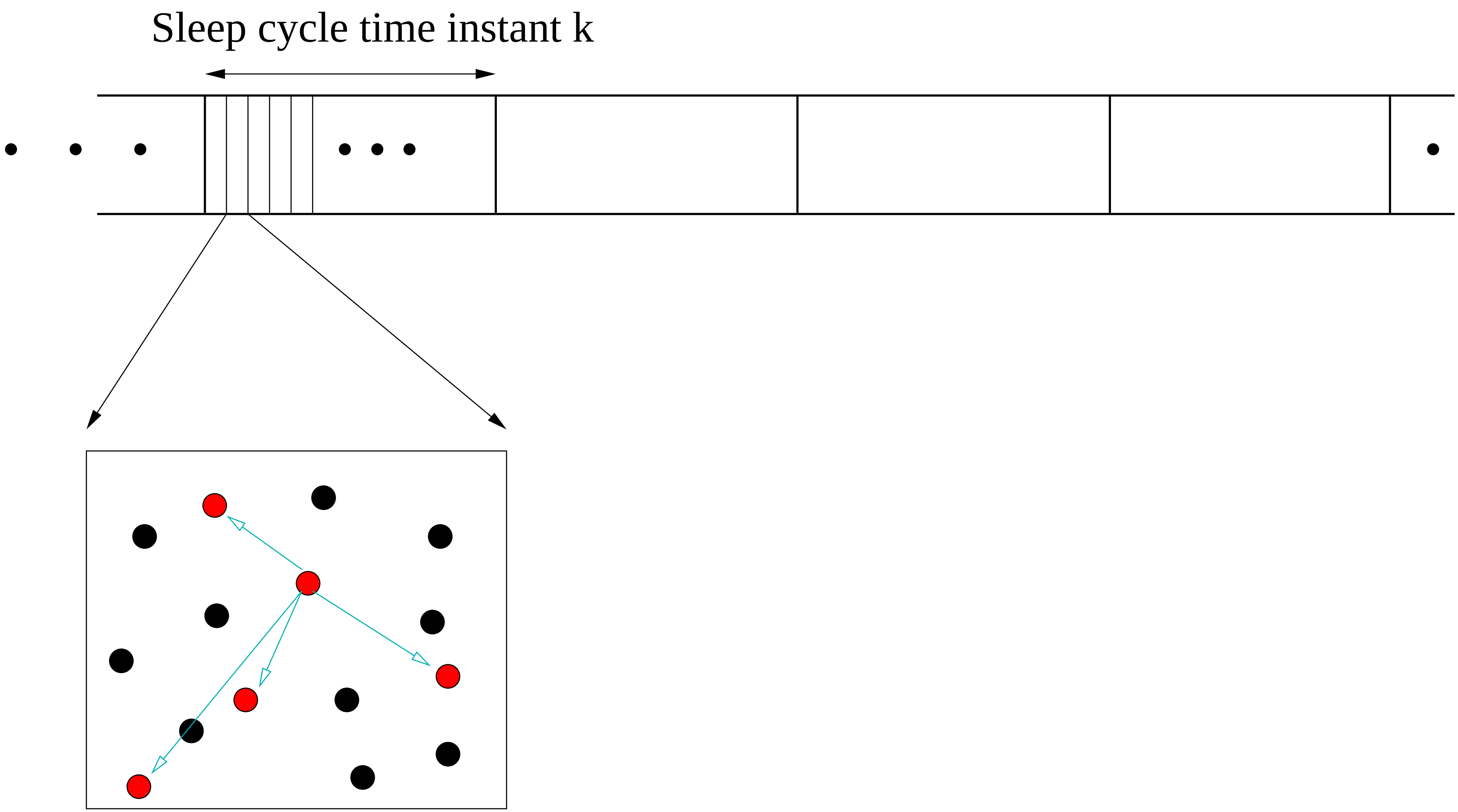
- Each time instant k is further divided into $2c$ time slots
- In the first c time slots, nodes which are awake during k take turns transmitting their training sequences

$$r_{k,i,j,m} = h_{i,j} u_{i,m} + v_{k,i,j,m}$$

Channel Training



Channel Training



- Collect the observations during sleep cycle time instant k

$$\mathbf{r}_k := [r_{k,i,j,m} \mid i,j \in \mathcal{S}(k), m \in \{1,\ldots,M\}, i \neq j]$$

The joint distribution of channel gains and observations

- The joint probability distribution of $\mathbf{r}$ and $\mathbf{h}$

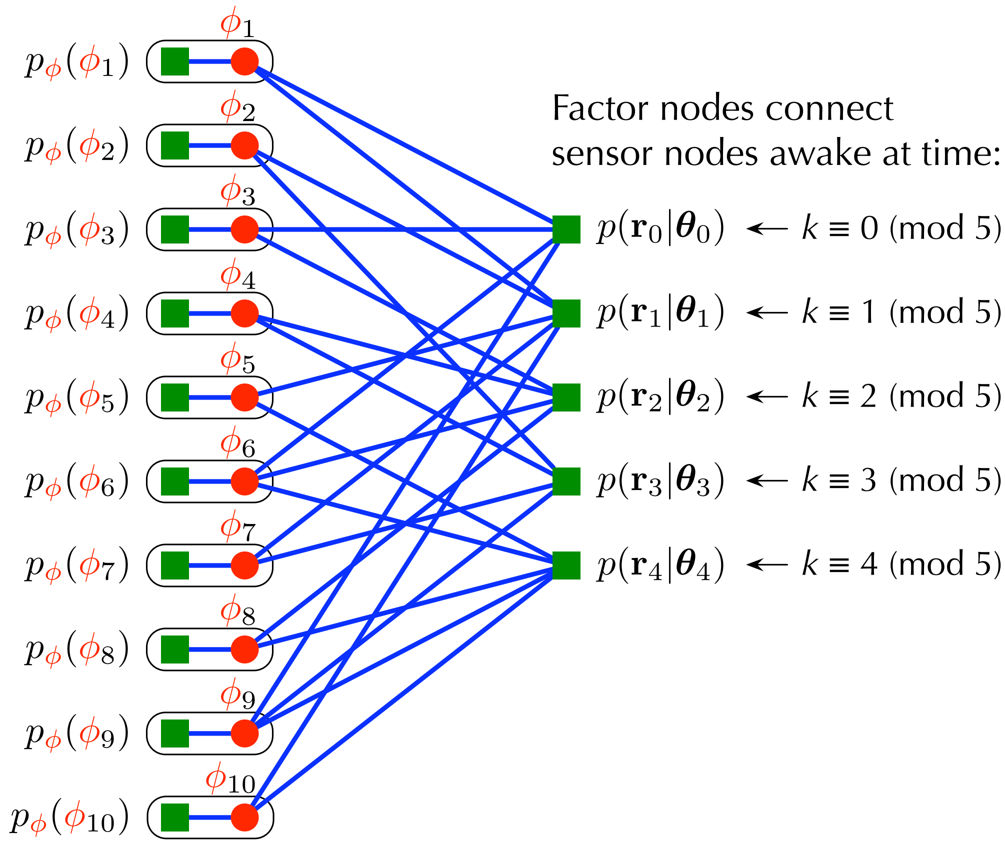
$$p_{\mathbf{r},\mathbf{h}} = p_{\mathbf{h}} \prod_{k=1}^{K} p_{\mathbf{r}_k|\mathbf{h}}$$

  where $\mathbf{r} := [\mathbf{r}_k \mid k \in \{1, \ldots, K\}]$
- Each node has a copy $\mathbf{h}_i$ of $\mathbf{h}$
- Write the joint distribution as

$$p_{\mathbf{r},\mathbf{h},\mathbf{h}_1,\ldots,\mathbf{h}_N} = \prod_{k=1}^{K} p_{\mathbf{r}_k|\mathbf{h}} \prod_{i=1}^{N} \delta(\mathbf{h} - \mathbf{h}_i)(p_{\mathbf{h}}(\mathbf{h}_i))^{\frac{1}{N}} \tag{1}$$

  where δ is the point mass distribution at zero.
- We can associate this model with a factor graph

Factor graph

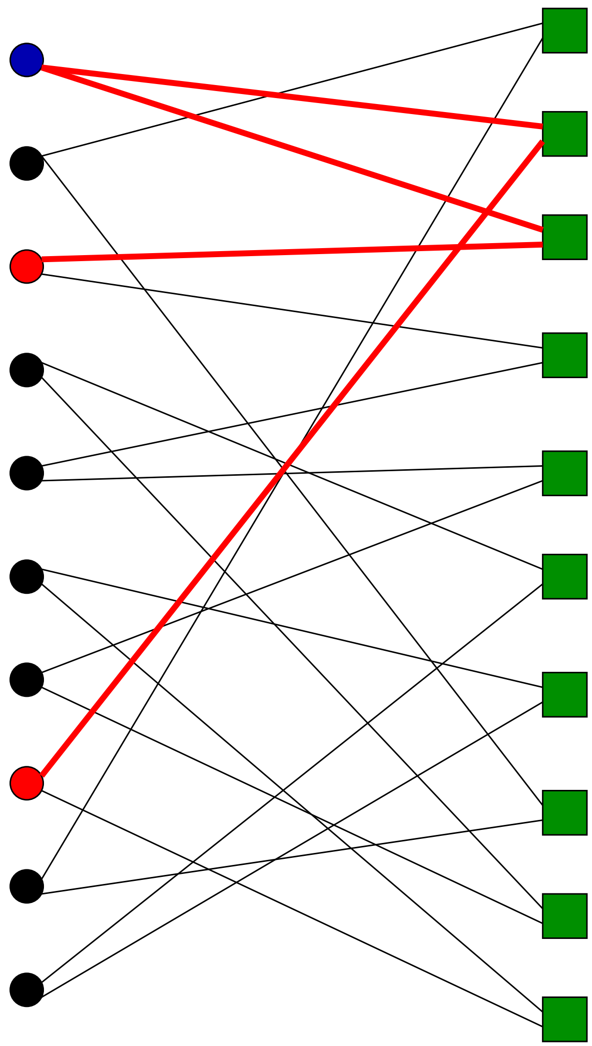


- A bipartite graph
- Left nodes: variable nodes, Right nodes: factor nodes
- Represent sensor nodes with the variable nodes and sleep cycle time instant with the factor nodes

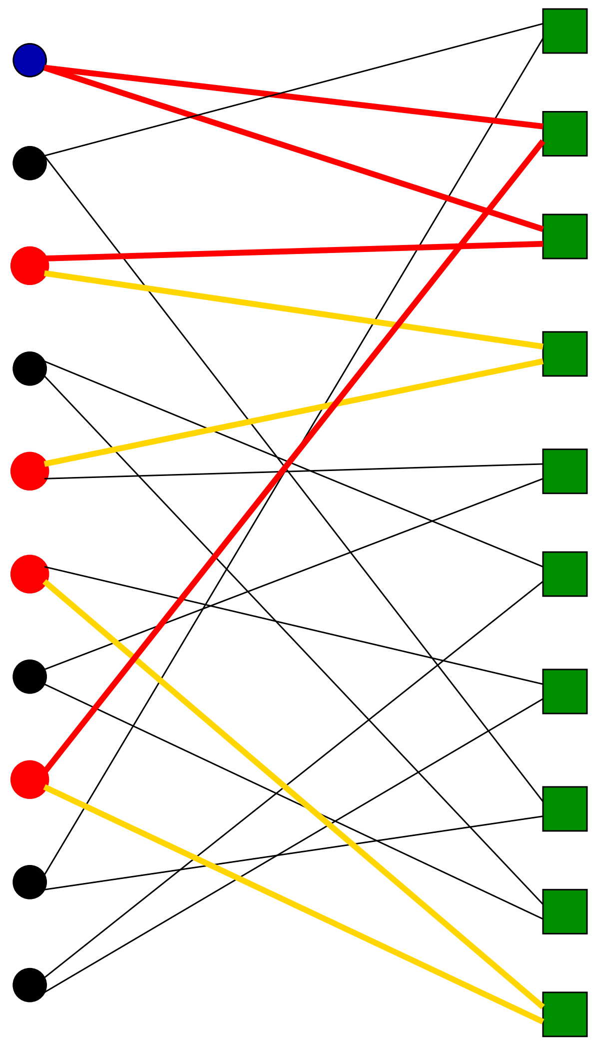
The channels observed by a node after ℓ sleep cycles

- Only a subset of the channels can be observed after ℓ sleep cycles
- Assumption: a node cannot disseminate data received during time instant k at another time instant k' in the same sleep cycle
- Decoding of data from other nodes takes time on the order of one complete sleep cycle
- After ℓ sleep cycles, nodes can directly or indirectly receive information about links observed by nodes only upto 2ℓ edges away from them in the factor graph.

The channels observed by a node after 1 sleep cycle



The channels observed by a node after 2 sleep cycles



Expectation Propagation

- The approximated prior joint distribution on $\mathbf{h}$ can be written as

$$p_{\mathbf{h}}(\mathbf{h}) \propto \exp\{-\frac{1}{2}[(\mathbf{h}-\mathbf{m_h})^T \mathbf{\Sigma}_{\mathbf{h}}^{-1}(\mathbf{h}-\mathbf{m_h})]\} \tag{2}$$

- Initial estimate at each node $\hat{\mathbf{h}} = \mathbf{m_h}$
- They may want to update their estimates by updating the statistics (mean and covariance)
- Once we have associated the joint distribution on $\mathbf{h}$ with a factor graph, we can apply Expectation Propagation to calculate the posterior distribution

Selection of message family

- The approximated prior joint distribution on $\mathbf{h}$

$$p_{\mathbf{h}}(\mathbf{h}) \propto \exp\{-\frac{1}{2}[(\mathbf{h}-\mathbf{m_h})^T \mathbf{\Sigma}_{\mathbf{h}}^{-1}(\mathbf{h}-\mathbf{m_h})]\} \tag{3}$$

- The conditional joint distribution on the observations $\mathbf{r}_k$ collected during sleep cycle instant k

$$p_{\mathbf{r}_k|\mathbf{h}_k}(\mathbf{r}_k|\mathbf{h_k}) \propto \exp\{-\frac{1}{2}[(\mathbf{r}_k-\mathbf{m}_{\mathbf{r}k})^T \mathbf{\Sigma}_{\mathbf{r}_k}^{-1}(\mathbf{r}_k-\mathbf{m}_{\mathbf{r}k})]\} \tag{4}$$

- Select the the message exponential family to be used in EP to be multivariate Gaussian distributed as

$$\mathbf{v}(\mathbf{h}) = \begin{pmatrix} \mathbf{h}_y & \mathbf{h}_z & \mathbf{h} \end{pmatrix}^T$$

$$\begin{aligned} \mathbf{h}_y &:= [h_{i,j}^2|i,j \in \{1,\ldots,N\}, i<j] \\ \mathbf{h}_z &:= [h_{i,j}h_{m,n}|i,j,m,n \in \{1,\ldots,N\}, i<j, m<n, m>i] \\ \mathbf{h} &:= [h_{i,j}|i,j \in \{1,\ldots,N\}, i<j] \end{aligned}$$

Diffusion LMS [17]

- Least-Mean Squares (LMS) is a stochastic gradient-descent algorithm
- During sleep cycle time instant k, when node i transmits, the nodes $i' \in \mathcal{S}(k) \setminus i$ make observations
- Node i' has access to $\{u_{i,m}, r_{k,i,i',m}\}$ $u_{i,m}$: regression vector, $r_{k,i,i',m}$: desired signal
- Estimate $\hat{h}_{i,i'}^{k,m}$ of $h_{i,i'}$ at node i'

$$\hat{h}_{i,i'}^{k,m} = \hat{h}_{i,i'}^{k,m-1} + \mu u_{i,m}(r_{k,i,i',m} - \hat{h}_{i,i'}^{k,m} u_{i,m})$$

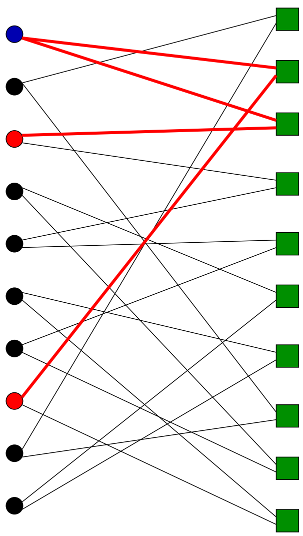
- At the end of the sleep cycle instant k, diffuse the estimate by

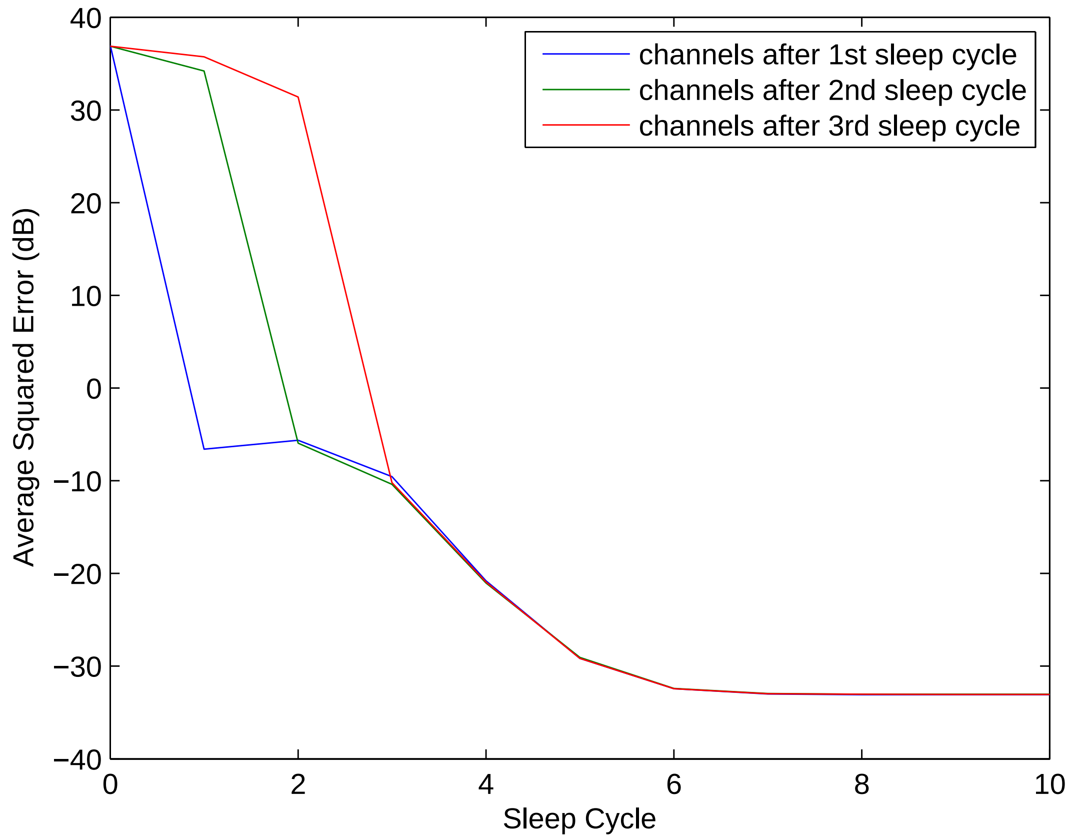
$$\tilde{\mathbf{h}}^k = \sum_{i \in \mathcal{S}(k)} a(k,i) \hat{\mathbf{h}}_i^k$$

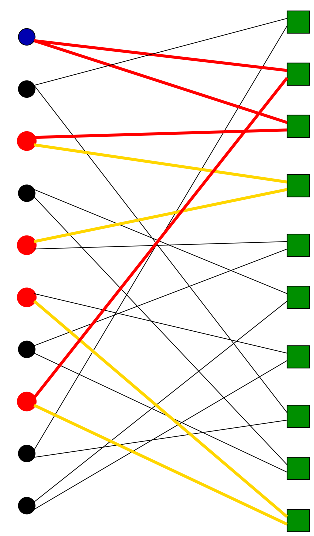
Simulation

- A network with 20 sensors
- Random sleep strategy with $K = 10, d = 4, c = 2$
- Training sequence of length 1000
- Monte Carlo Simulations 400
- Plot average estimation error of only those channel gains observed directly or indirectly after ℓ iterations

Simulation results: EP [8, 9]

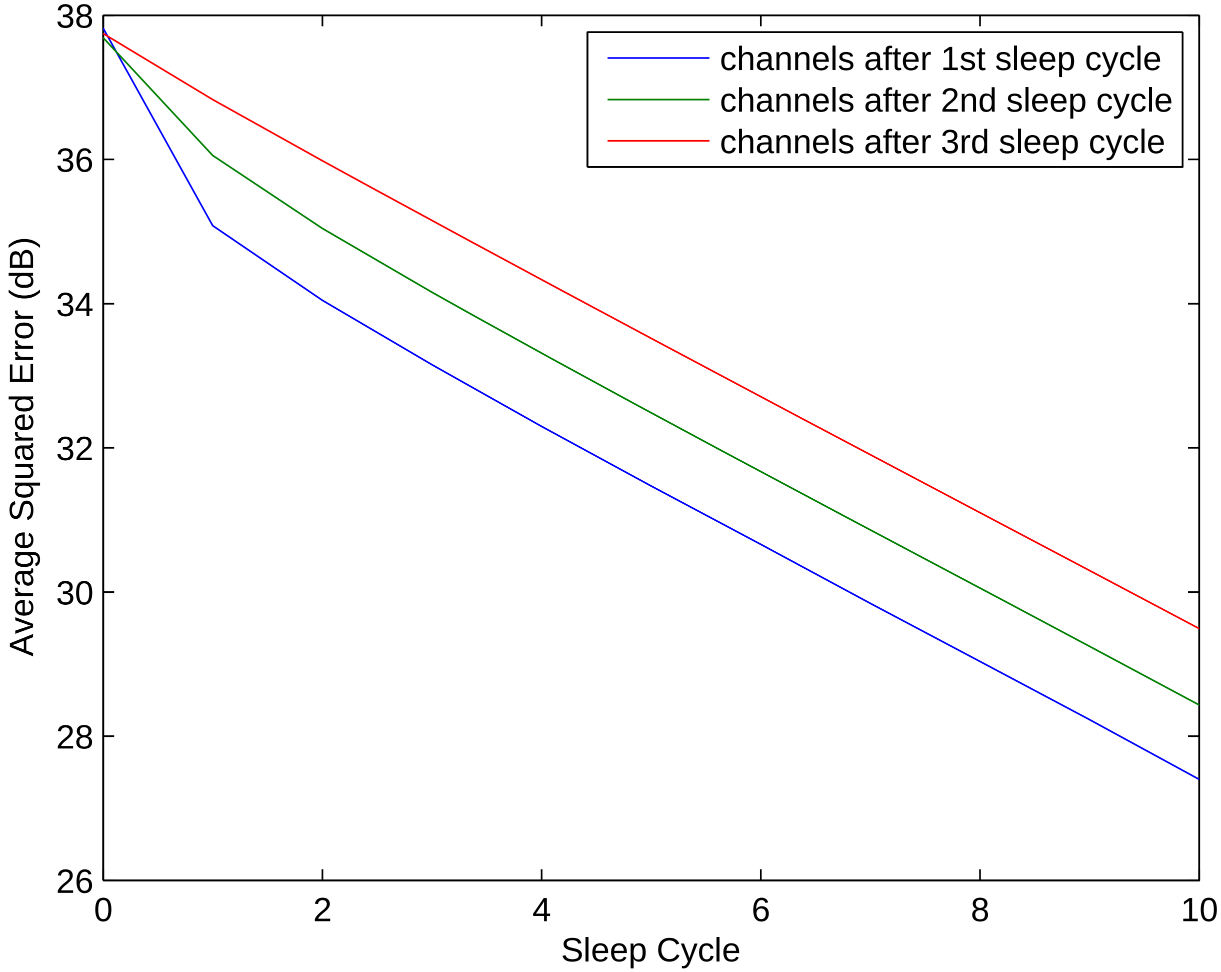






- For directly observed links, drastic change after first sleep cycle
- Drastic change shifts for the indirectly observed links
- Estimation error decreases even after that

Simulation results: LMS

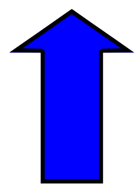


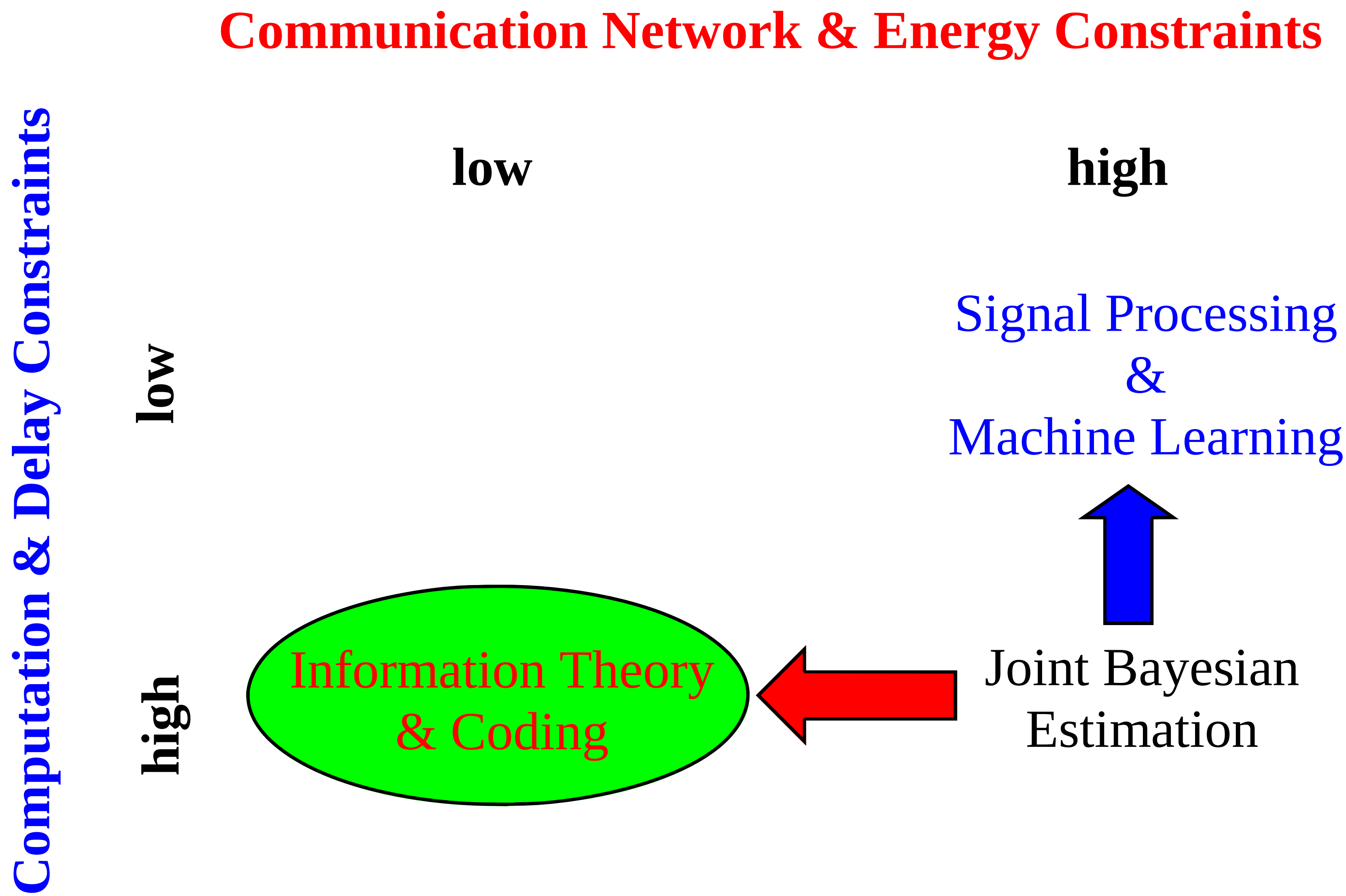
- Maximum step size before it starts to diverge: 1.995

Communication Network & Energy Constraints

Computation & Delay Constraints

	low	**high**
low		Signal Processing & Machine Learning
high		Joint Bayesian Estimation





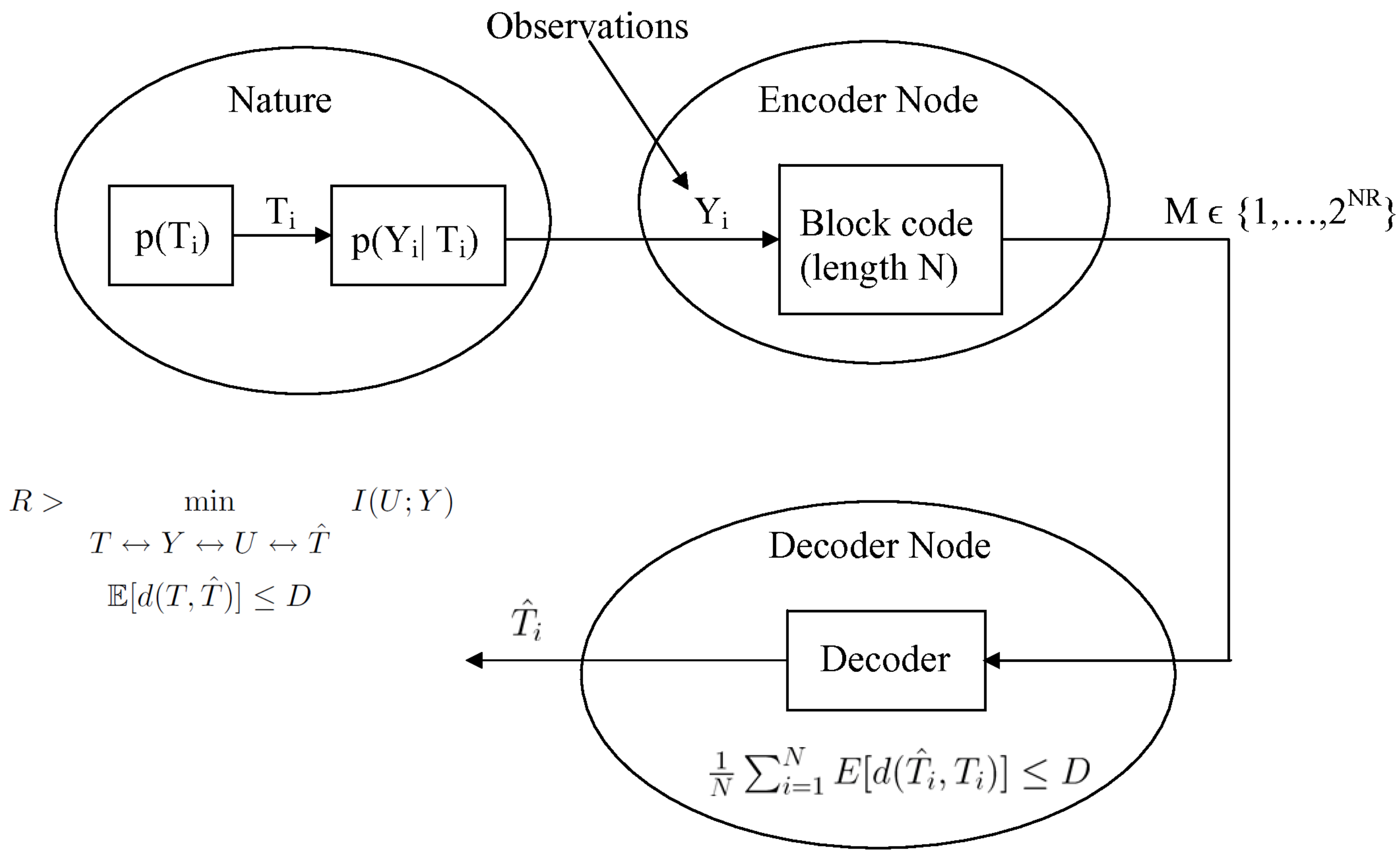
Communication Network & Energy Constraints
Computation & Delay Constraints
low
high
low
high
Signal Processing
&
Machine Learning
Information Theory
& Coding
Joint Bayesian
Estimation

The Information Theory/Coding Perspective

PROBLEM 2: $$\text{only } \mathbf{r}_m \text{ is originally available to node } m,$$ $$\text{any communication is over finite rate links}$$

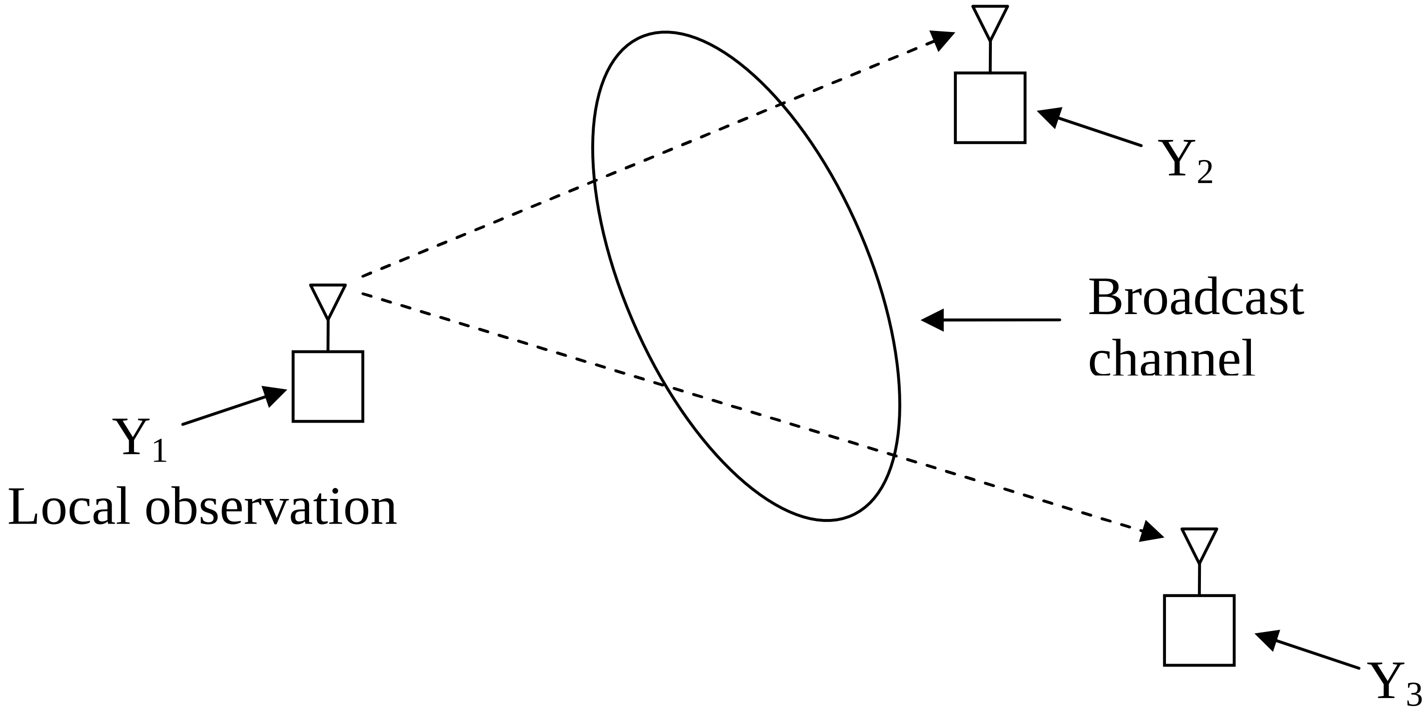
- another major important difficulty: the nodes must send their messages over a rate limited wired or wireless communication network. The information exchanged can not exceed the capabilities of this network.
- In a wireless network, the capabilities of the network are strongly related to the energy expenditures of the network nodes, due to a large amount of power spent on transmission.
- How should the communications be organized to allow for the best estimate performance when adapted to different communications networks? (I.e. what is the code structure?)
- What is the best estimate performance we can have subject to these constraints?

Relationship Between Remote Bayesian Estimation and Lossy Source Coding



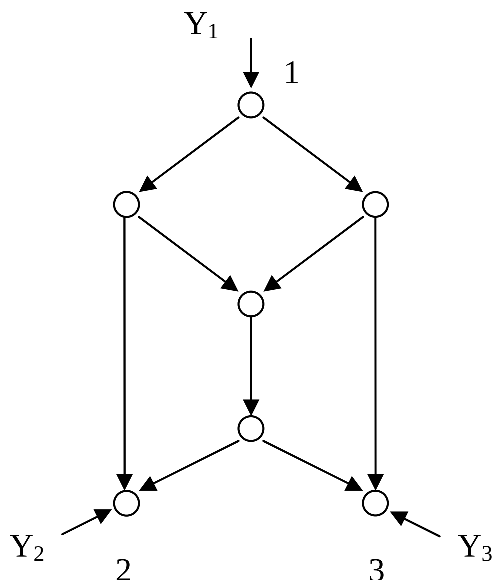
- $\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[d(\hat{T}_i, T_i)] < D$ plays the role of an average Bayesian cost. Dobrushin & Tsybakov '62 [18] showed minimum rate necessary to attain $< D$ is
- Just like rate distortion function but with T_i instead of Y_i, and Markov requirement.

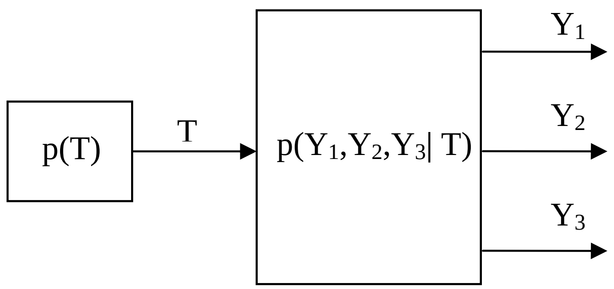
What should the source code architecture be?



- Scalar Gaussian broadcast channel is degraded:
  - everything that receiver w/ ↓ SNR gets, the receiver with ↑ SNR gets
  - receiver w/ ↑ SNR can get extra info
- Source code construction should reflect this:
  - If source code sends only individual messages $S_{1\to 2}, S_{1\to 3}$ the ability of receiver w/ ↑ SNR to hear everything sent to the receiver w/ ↓ SNR is *wasted*
  - ⇒ should use *multicast* messages! $S_{1\to\{2,3\}}, S_{1\to 2}, S_{1\to 2}$.

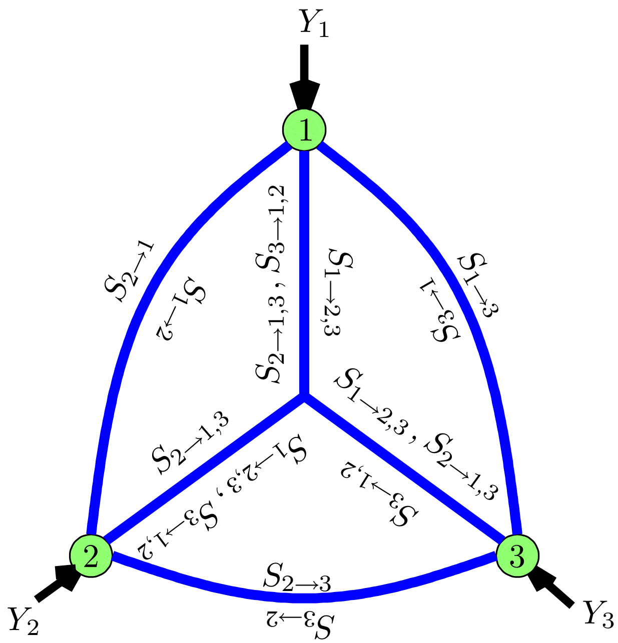
What should the source code architecture be?

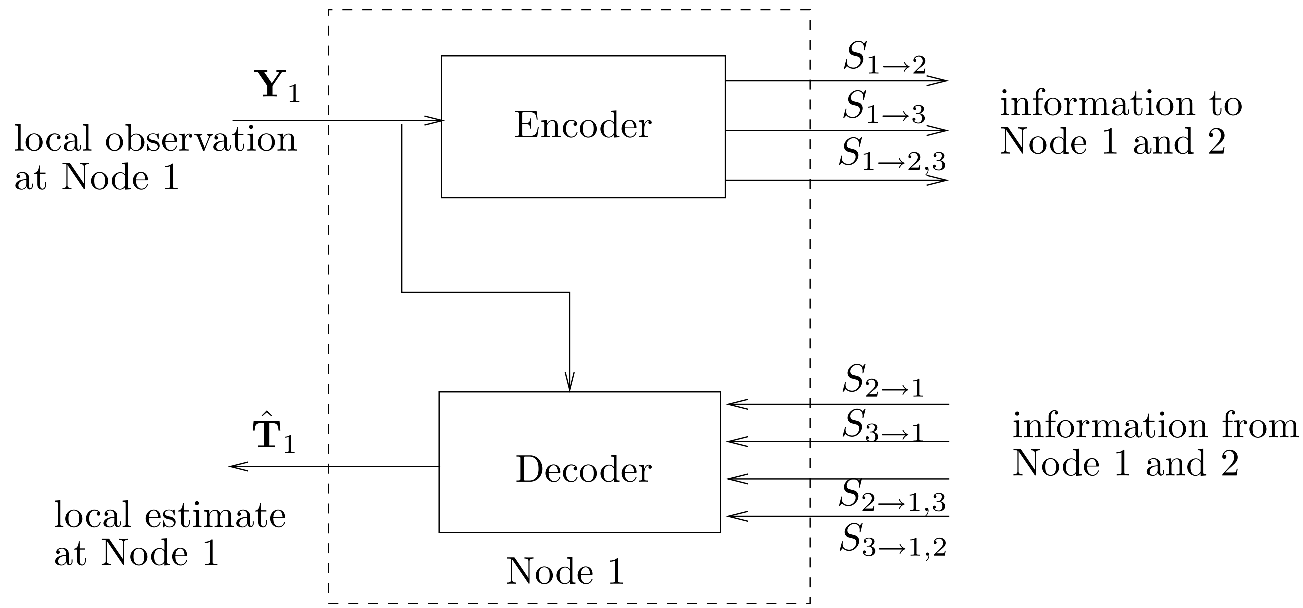




- Network coding insight: limitation for $R_{1\to\{2,3\}}$ is 2, higher than maximum equal $R_{1\to 2}, R_{1\to 3} = \frac{3}{2}$.
- Again implies that (even separated) source coding construction should allow for *multicast* rates.

What should the source code architecture be?





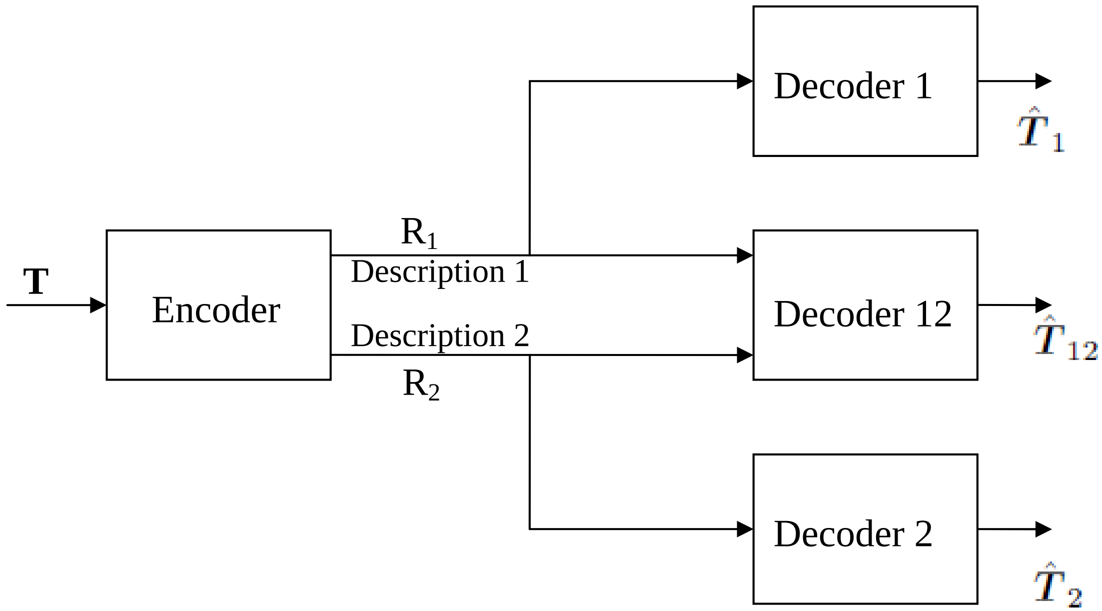
- Each node sends a (possibly different nonempty) message to each subset of other nodes. $S_{j\to\mathcal{A}},\ \mathcal{A} \subseteq [M] \setminus j$.
- Every node collects all of the received messages together with its local observations and forms an estimate which minimizes its local Bayesian cost $\mathbb{E}\left[d_m(T, \hat{T}_m)\right]$.

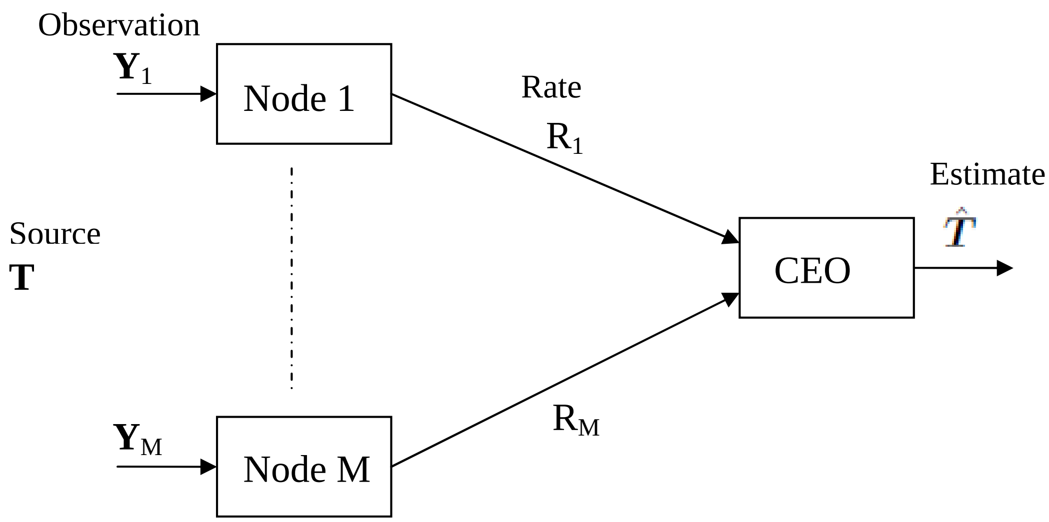
What performance do the best such codes have? (Motivation)

- Rate distortion region $\mathcal{R}$ of achievable rate vector $\mathbf{r} := [R_{j\to\mathcal{A}} | j \in [M],\ \mathcal{A} \subseteq [M] \setminus j]$ and estimation error (cost) vector $\mathbf{d} := [D_j | j \in [M]]$ pairs characterizes the best such codes.
- Capacity region $\mathcal{C}$ of a network is described in terms of all achievable $\mathbf{r}$.
- Best estimation performances attainable are those $\mathbf{d}$ associated with a $\mathbf{r}$ through $(\mathbf{r}, \mathbf{d}) \in \mathcal{R}$ with $\mathbf{r}$ on the boundary of $\mathcal{C}$.
- Hence, inner and outer bounds for the rate distortion region $\mathcal{R}$ for this problem are of interest.

Rate Distortion Region

our problem is a hybrid between two classic information theory problems...[19][20][21]





Rate Distortion Region: Inner Bound

- Multiple (M) Descriptions Achievability:
  1. select $p(\mathbf{U}|T)$ such that $\mathbb{E}[d(T, f(V, \{U_{\mathcal{A}}|\mathcal{A} \subseteq \mathcal{B}\}))] < D_{\mathcal{B}}$. Each element $U_{\mathcal{A}}$ of $\mathbf{U}$ corresponds to codeword avail. to nodes w/ all descriptions in $\mathcal{A} \subset [M]$.
  2. Generate codebook for $\mathcal{A}$ as $2^{N\tilde{R}_{\mathcal{A}}}$ length N codewords i.i.d. $p(U_{\mathcal{A}})$.

$$\sum_{\mathcal{A} \in \boldsymbol{\mathcal{P}}} \tilde{R}_{\mathcal{A}} > \sum_{\mathcal{A} \in \boldsymbol{\mathcal{P}}} H(U_{\mathcal{A}}|V) - H(\mathbf{U}_{\boldsymbol{\mathcal{P}}}|T, V) \quad \text{for all } \boldsymbol{\mathcal{P}} \subseteq 2^{[M]}$$

  makes sure $\exists$ codewords jointly typical w/ each other and T^N at encoder.

- CEO Achievability:
  1. select $p(U_i|Y_i)$, $U_{[M]\setminus i}, Y_{[M]\setminus i} \leftrightarrow Y_i V \leftrightarrow U_i$, such that $\mathbb{E}[d(T, f(U_{[M]}, V))] < D$.
  2. Generate codebook for i as $2^{N\tilde{R}_i}$ length N codewords iid $p(U_i)$ w/ $\tilde{R}_i > I(U_i; Y_i)$. Divide into 2^{NR_i} bins, send index of bin with codeword jointly typical with observations Y_i^N.

$$\sum_{i \in \mathcal{A}} R_i > I(Y_{\mathcal{A}}; U_{\mathcal{A}}|U_{[M]\setminus\mathcal{A}})$$

  makes sure $\exists$ codewords jointly typical w/ each other in bins at decoder.

Rate Distortion Region: Inner Bound

To hybridize the CEO and MD constructions, let each encoder in CEO encode multiple dependent descriptions, then bin. $\Longrightarrow$ both *encoder* (codebook size) & *decoder* (bin size) inequalities nontrivial. [22]

- $\mathcal{S}_i, \mathcal{D}_i$, messages sent, recvd at node i, resp.
- **Time Sharing**: V is independent from $\boldsymbol{Y}_{[M]}, T$
- **Encoding Constraints**: $T, \hat{\boldsymbol{T}}_{[M]}, \boldsymbol{Y}_{[M]\setminus i}, \boldsymbol{U}_{\mathcal{S}\setminus\mathcal{S}_i} \leftrightarrow Y_i, V \leftrightarrow \boldsymbol{U}_{\mathcal{S}_i}$
- **Decoding Constraints**: $T, \hat{\boldsymbol{T}}_{[M]\setminus i}, \boldsymbol{Y}_{[M]\setminus i}, \boldsymbol{U}_{\mathcal{S}\setminus\mathcal{D}_i} \leftrightarrow Y_i, \boldsymbol{U}_{\mathcal{D}_i}, V \leftrightarrow \hat{T}_i$, and $D_i > \mathbb{E}\left[d_i\left(T; \hat{T}_i\right)\right]$

Rate Distortion Region: Inner Bound

- **Codebooks:** $\forall\ \mathcal{P}_j \subseteq \mathcal{S}_j,\ \forall\ j \in [M]$

$$\sum_{(j\to\mathcal{A})\in\mathcal{P}_j} \tilde{R}_{j\to\mathcal{A}} > \sum_{(j\to\mathcal{A})\in\mathcal{P}_j} H(U_{j\to\mathcal{A}}|V) - H(\boldsymbol{U}_{\mathcal{P}_j}|Y_j, V),$$

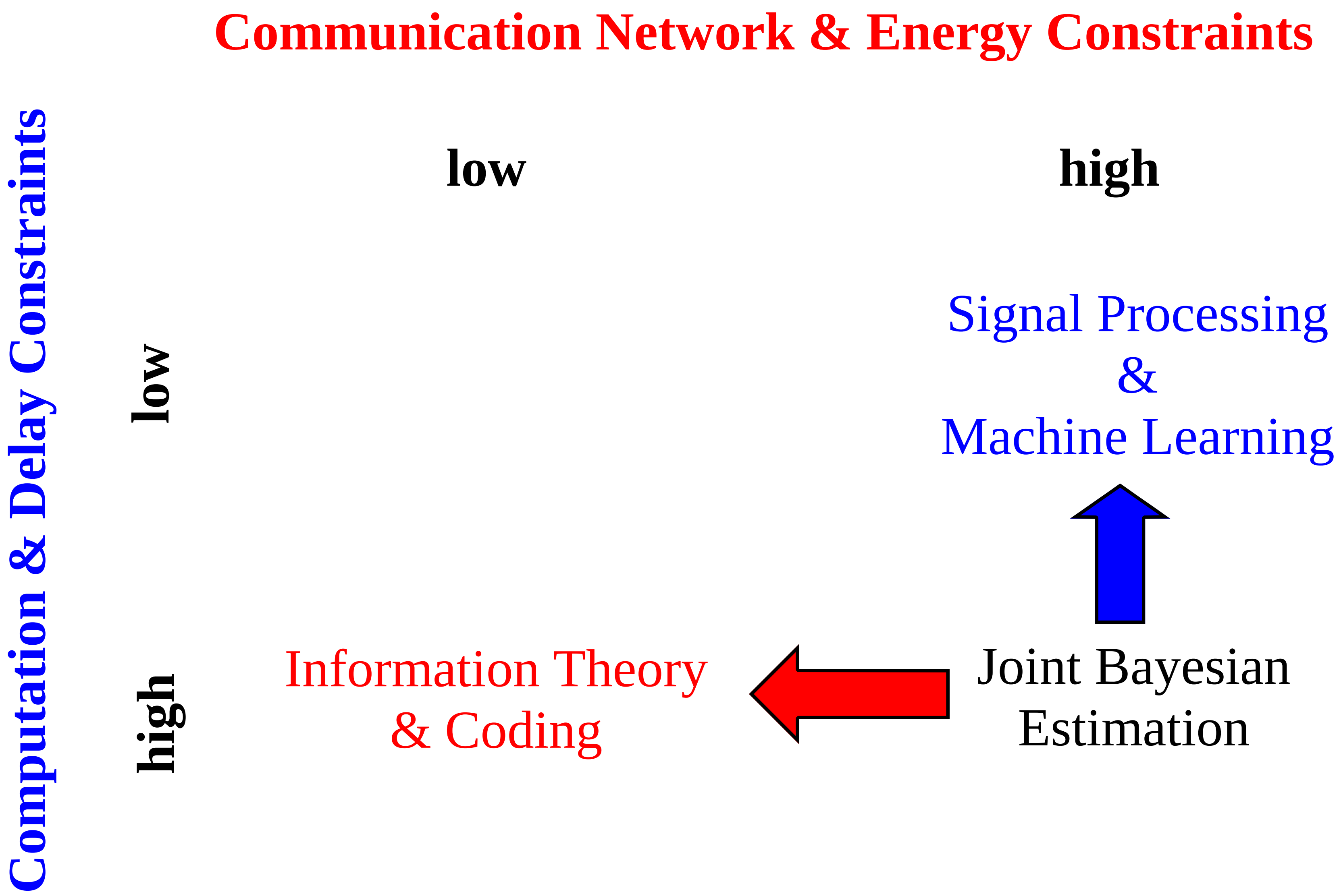
Makes sure there is a collection of codewords in the codebooks jointly typical with each other and the observations at each *encoder.*

- **Bins:** for all $\mathcal{C}_i \subseteq \mathcal{D}_i$ and $i \in [M]$

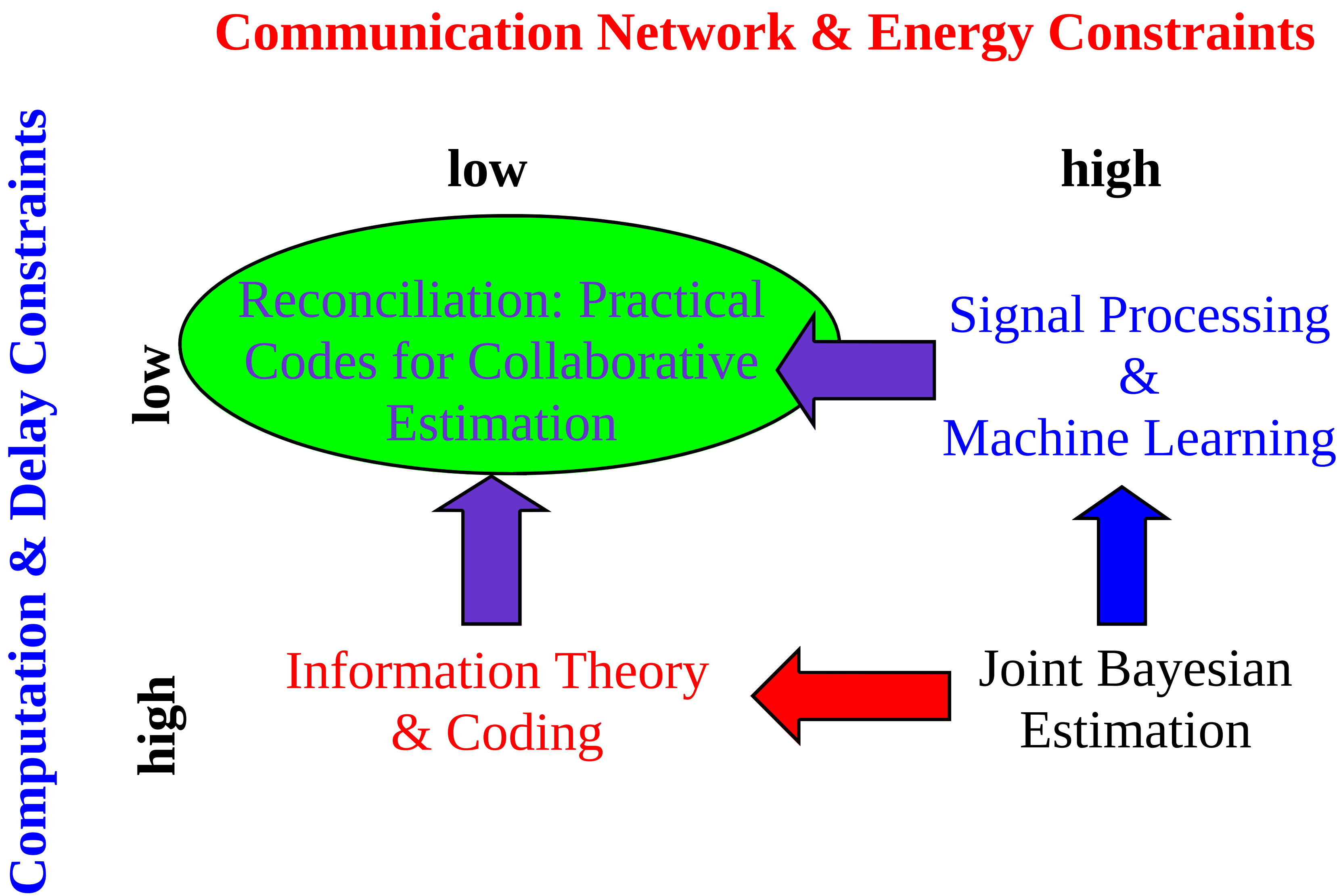
$$\sum_{(j\to\mathcal{A})\in\mathcal{C}_i} R_{j\to\mathcal{A}} > \sum_{(j\to\mathcal{A})\in\mathcal{C}_i} \Big(\tilde{R}_{j\to\mathcal{A}} - H(U_{j\to\mathcal{A}}|V)\Big) + H(\boldsymbol{U}_{\mathcal{C}_i}|V, \boldsymbol{U}_{\mathcal{D}_i\setminus\mathcal{C}_i}, Y_i)$$

Makes sure that the bins are small enough such that there is only one collection of codewords jointly typical with each other and the side information at each *decoder*.

How might these perspectives be reconciled?



How might these perspectives be reconciled?



How might these perspectives be reconciled? Future Work

- Sparse graph coding constructions and modifications of BP decoders have been adapted to some multiterminal coding problems (Wyner-Ziv, Slepian-Wolf)
- How can they be adapted and generalized to this one?
- What do the information theoretic bound evaluate to in important pragmatic estimation problems for wireless networks, such as for channel estimation?
- Belief/expectation propagation can help not only with designing the decoders, but also determining which information to compress in order to make risk minimization tractable after decoding.

References

[1] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Uncertainty in AI'01*, 2001.

[2] ——, "A family of algorithms for approximate bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[3] J. M. Walsh, "Distributed Iterative Decoding and Estimation via Expectation Propagation: Performance and Convergence," Ph.D. dissertation, Cornell University, 2006.

[4] J. Pearl, *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers, 1988.

[5] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[6] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[7] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, no. 2, pp. 619–639, Feb. 2001.

[8] S. Ramanan and J. M. Walsh, "Distributed Estimation of Channel Gains in Wireless Sensor Networks," in *Forty-Second Asilomar Conference on Signals, Systems, and Computers*, Oct. 2008. [Online]. Available: http://www.ece.drexel.edu/walsh/Ramanan_Asilomar_08.pdf

[9] S. Ramanan and J. M. Walsh, "Distributed Estimation of Channel Gains in Wireless Sensor Networks," *IEEE Trans. Signal Processing*, submitted March 16, 2009. Rejected. Revised and resubmitted August, 2009.

[10] J. M. Walsh and P. A. Regalia, "Expectation propagation for distributed estimation in sensor networks," in *8th IEEE International Workshop on Signal Processing Advances for Wireless Communications (SPAWC)*, Helsinki, Finland, June 2007. [Online]. Available: http://www.ece.drexel.edu/walsh/spawc07.pdf

[11] J. M. Walsh, P. A. Regalia, and S. Ramanan, "Optimality of Expectation Propagation Based Distributed Estimation for Wireless Sensor Network Initialization," in *9th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2008, pp. 620 – 624. [Online]. Available: http://www.ece.drexel.edu/walsh/WalshSpawc08.pdf

[12] M. Cetin, L. Chen, J. W. Fisher III, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed Fusion in Sensor Networks," *IEEE Signal Processing Mag.*, pp. 42–55, July 2006.

[13] A. T. Ihler, I. J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for sensor network self-calibration," in *Proc. The International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Quebec, May 2004.

[14] ——, "Nonparametric Belief Propagation for Self-Calibration in Sensor Networks," in *Information Processing in Sensor Networks (IPSN)*, July 2004.

[15] ——, "Nonparametric Belief Propagation for Sensor Network Self-Calibration," *IEEE J. Select. Areas Commun.*, vol. 23, april 2005.

[16] C. C. Moallemi and B. Van Roy, "Consensus propagation," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

[17] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Processing*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[18] R. L. Dobrushin and B. S. Tsybakov, "Information transmission with additional noise," *IEEE Transactions on Information Theory*, vol. IT-8, no. 5, pp. 293–304, September 1962.

[19] J. Chen, X. Zhang, T. Berger, and S. B. Wicker, "An upper bound on the sum-rate distortion function and its corresponding rate allocation schemes for the ceo problem," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 977–987, August 2004.

[20] T. Berger, Z. Zhang, and H. Viswanathan, "The CEO Problem," *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 887–902, May 1996.

[21] A. El Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Transactions on Information Theory*, vol. IT-28, no. 6, pp. 851–857, November 1982.

[22] Te Sun Han and Kingo Kobayashi, "A Unified Achievable Rate Region for a General Class of Multiterminal Source Coding Systems," *IEEE Trans. Inform. Theory*, vol. IT-26, no. 3, pp. 277–288, May 1980.