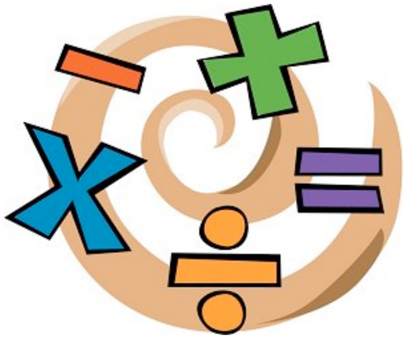
Scalar consensus [Tsitsiklis'84]



Model

- N sensors, form a network
- Observation at sensor i: $z_i \in R$

Goal

- *Average consensus:* compute the average of the observations, without centralized processing
- *Max-consensus:* compute the maximum of the observations, without centralized processing

Comments

- Only share information with neighbors
- Use broadcasts

A first attempt

- Everyone think of a number
- Two communication networks
  - You can only communicate with your immediate neighbors
  - You can communicate with everyone

- Goal 1: compute average
- Goal 2: compute maximum

- How would you do it?
- How long would it take?
- What side-information would you need?

Average consensus

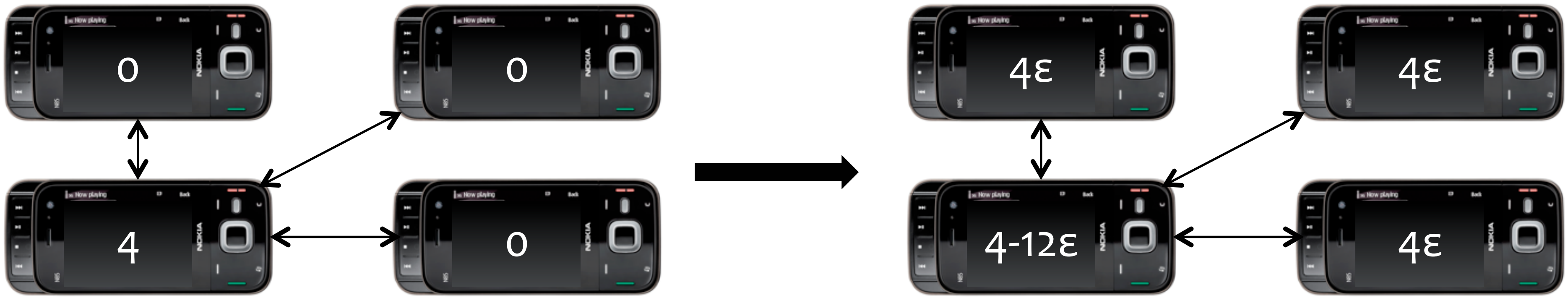
- Introduce state $z_i(k)$, $i=1,\dots,N$; $k=0,1,2,\dots$
- Update rule, with $z_i(0)=z_i$

$$z_i(k) = z_i(k-1) + \varepsilon \sum_{j \in \mathcal{N}_i} \left(z_j(k-1) - z_i(k-1)\right)$$

- Then, when ε is sufficiently small

$$z_i(k) \to \frac{1}{N}\sum_{i=1}^{N} z_i(0)$$

- Basic idea
  - Every iteration average is maintained
  - At every iteration, states get closer (contraction)



Vector consensus

Model

- N sensors, form a network
- Observation at sensor i: $\mathbf{z}_i \in R^L$

Goal

- *Average consensus:* compute the average observations, without centralized processing, per dimension

Solution

- Use average scalar consensus per dimension

Application to distributed inference

- Goal: compute $p(x|\mathbf{y}) \propto \prod_{i=1}^{N} p(y_i|x)p(x)$
- Likelihood is key

$$\prod_{i=1}^{N} p(y_i|x) = \exp\left(\sum_{i=1}^{N} \log p(y_i|x)\right)$$
$$= \exp\left(\frac{N}{1} \boxed{\frac{1}{N}\sum_{i=1}^{N} \log p(y_i|x)}\right)$$

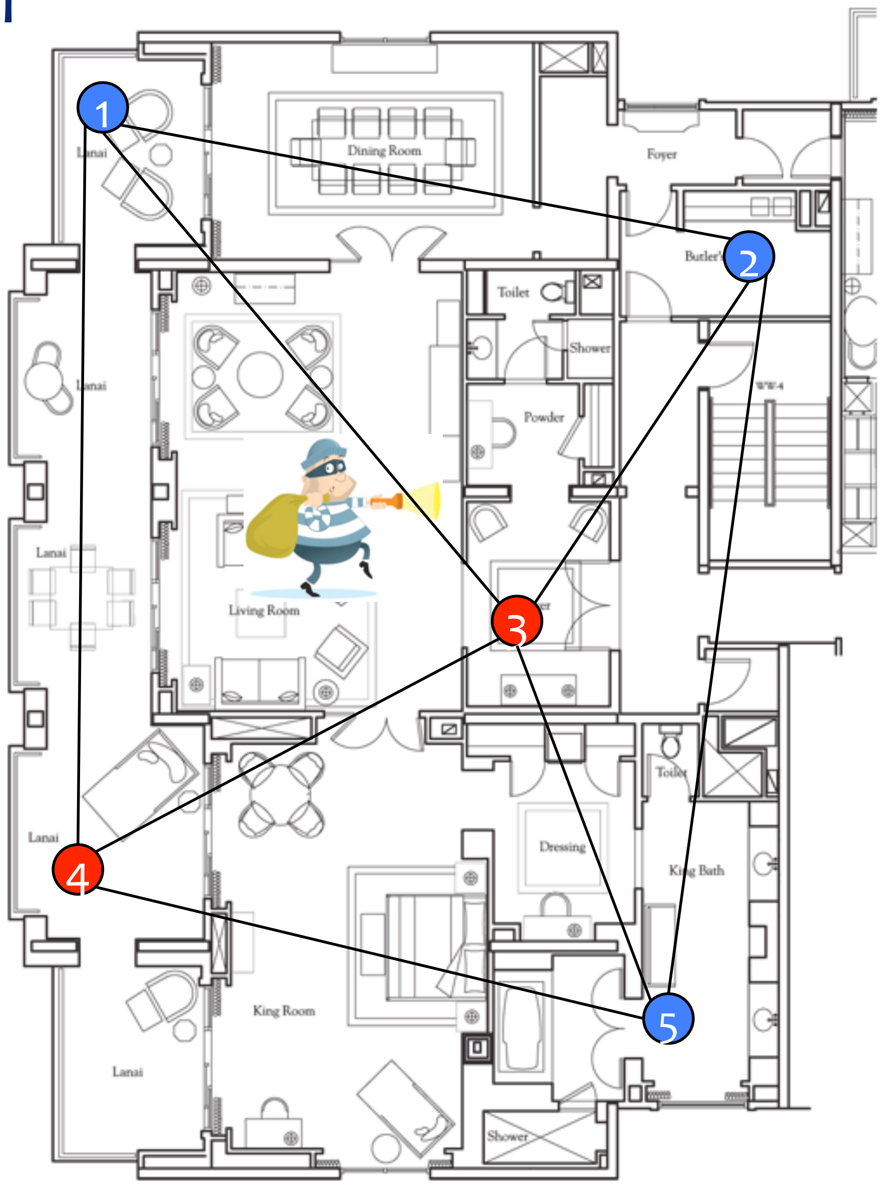
- Run consensus for x=0 and for x=1

$$\left.\begin{aligned} L_0 &= \frac{1}{N}\sum_{i=1}^{N} \log p(y_i|x=0) \\ L_1 &= \frac{1}{N}\sum_{i=1}^{N} \log p(y_i|x=1) \end{aligned}\right\} \quad \begin{aligned} p(x=0|y) &\propto p(x=0)e^{NL_0} \\ p(x=1|y) &\propto p(x=1)e^{NL_1} \end{aligned}$$

*In addition to max. degree, N needs to be known

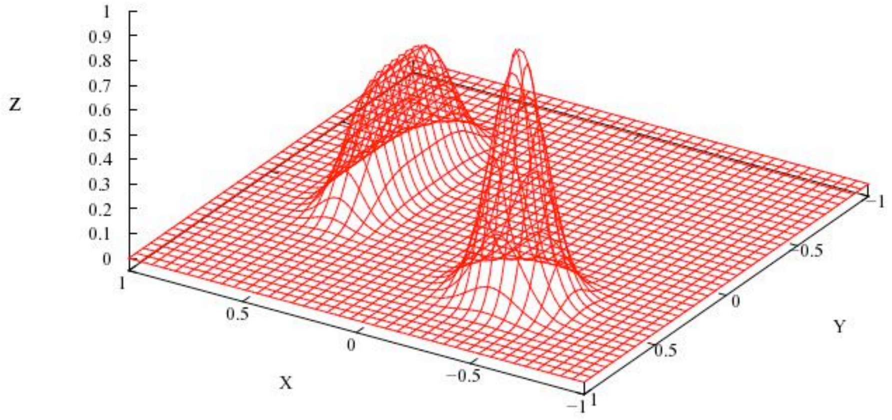
Example: intruder detection

- Goal: detect presence of thief $x \in \{0,1\}$, uniform prior
- Every node makes a measurement $y_i \in \{0=\text{blue},1=\text{red}\}$, with missed detection prob. q_{MD}/ false alarm prob. q_{FA} (with $q_{MD} >> q_{FA}$)
- Determine
  - a local likelihood
  - the initial states
  - good value for ε
  - consensus rules
  - final state
- Can you do consensus without the logs?



Extension to continuous parameters

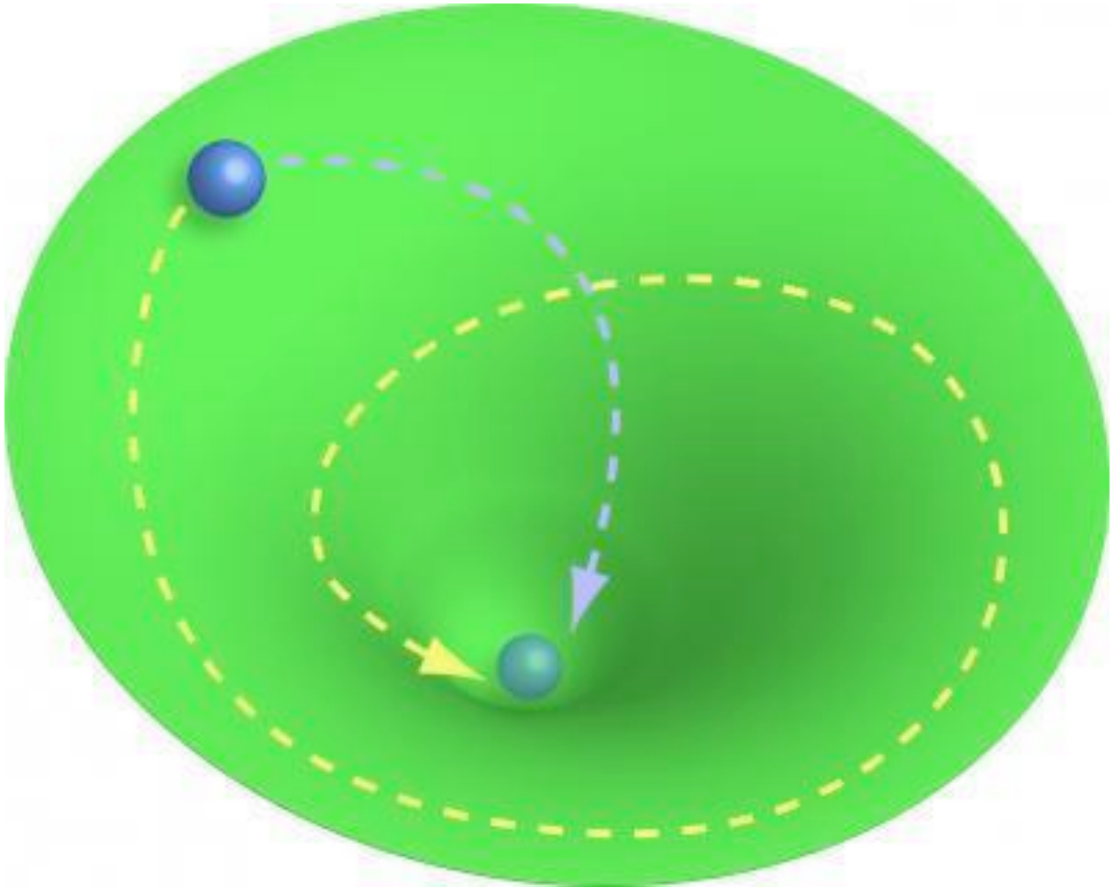
- Need to exchange and add functions
- Parametric and sample-based approaches exist



Outline

- Applications
- Background and terminology
- Bayesian detection
- Tool 1: Bayesian graphical models
  - Basics
  - Recipe for Bayesian inference
  - Practicalities
  - A worked example: a digital receiver
- Tool 2: Belief consensus
  - Basics
  - Convergence
- Applications revisited
- Variational interpretation
- Summary and conclusions

Convergence behavior



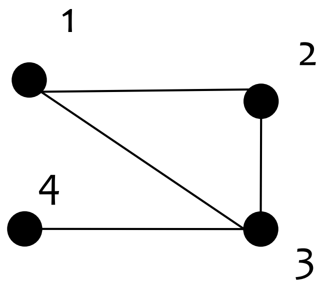
Convergence of average consensus

- Write in vector notation

$$z_i(k) = z_i(k-1) + \varepsilon \sum_{j \in \mathcal{N}_i} (z_j(k-1) - z_i(k-1))$$

$$\begin{aligned}\mathbf{z}(k) &= \mathbf{z}(k-1) + \varepsilon \mathbf{L}\mathbf{z}(k-1) \\ &= \mathbf{P}\mathbf{z}(k-1)\end{aligned}$$

- Example



$$\mathbf{L} = \begin{bmatrix} -2 & 1 & 1 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & 1 & -3 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1-2\varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & 1-2\varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & 1-3\varepsilon & \varepsilon \\ 0 & 0 & \varepsilon & 1-\varepsilon \end{bmatrix}$$

Convergence of average consensus

- Now $\mathbf{z}(k) = \mathbf{P}^k \mathbf{z}(0)$
- Properties of P
  - N orthonormal eigenvectors $\mathbf{v}_1=\mathbf{1}/\sqrt{N}$, $\mathbf{v}_2$, … , $\mathbf{v}_N$
  - N eigenvalues $\lambda_1=1$, λ_2 , … , λ_N
  - When $\varepsilon<1/\max$ (node degree), $|\lambda_n| < 1$, $n>1$
- Convergence
  - Eigen-expansion of $\mathbf{z}(0)$

$$\mathbf{z}(0) = \sum_{n=1}^{N} a_n \mathbf{v}_n$$

$$= a_1 \mathbf{1}/\sqrt{N} + \sum_{n>1}^{N} a_n \mathbf{v}_n$$

$$= \sum_i z_i(0)\mathbf{1}/N + \sum_{n>1}^{N} a_n \mathbf{v}_n$$

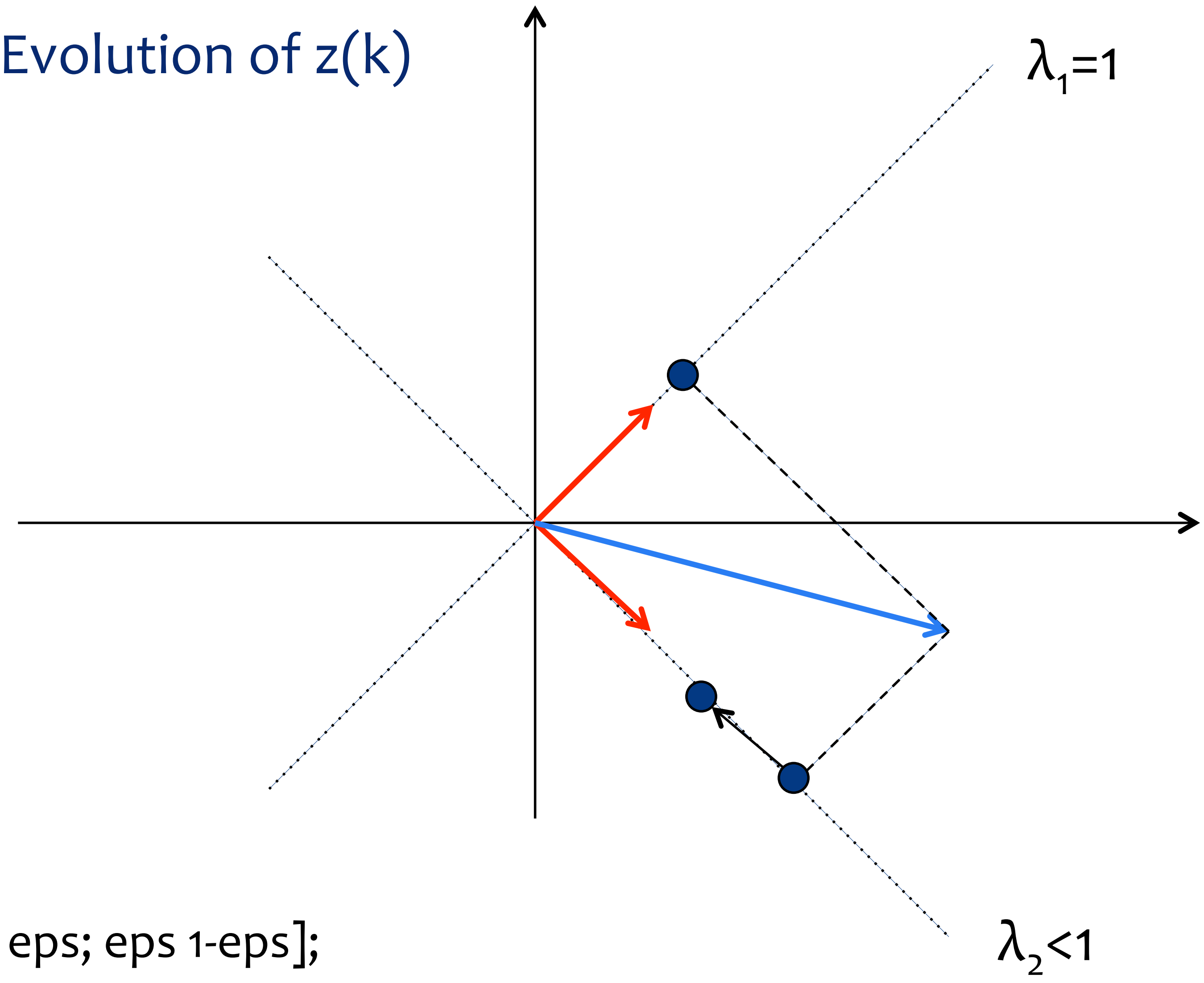
Apply $\mathbf{P}^k$

$$\mathbf{z}(k) = \sum_{n=1}^{N} a_n \lambda_n^k \mathbf{v}_n$$

$$= \frac{1}{N} \sum_i z_i(0)\mathbf{1} + \sum_{n>1}^{N} a_n \lambda_n^k \mathbf{v}_n$$

Graphically

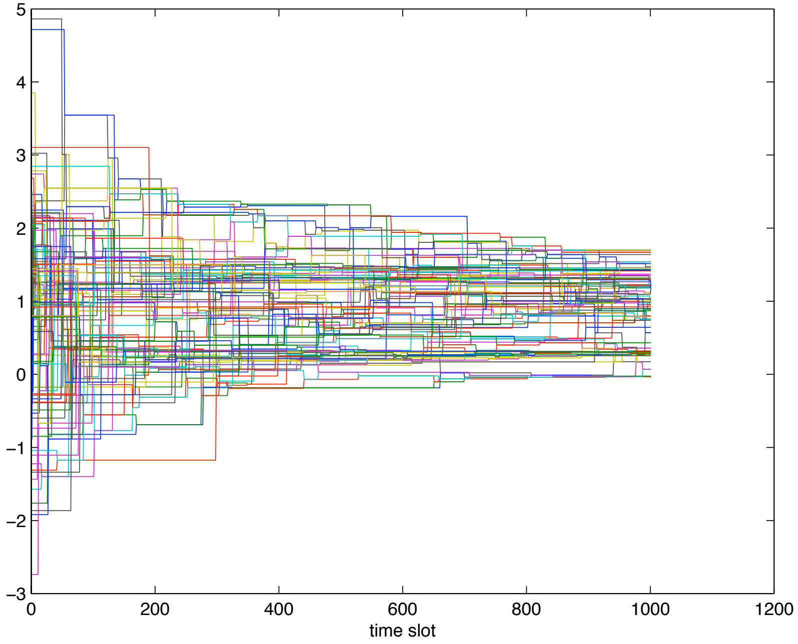
- Evolution of z(k)

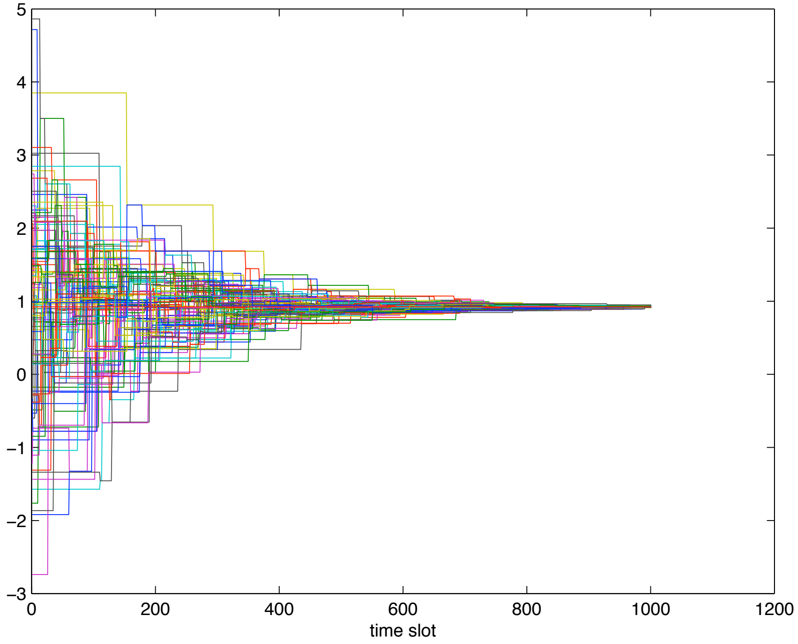


P=[1-eps eps; eps 1-eps];

Examples

- Chain network: second eigenvalue close to 1
- Clique network: second eigenvalue close to 0



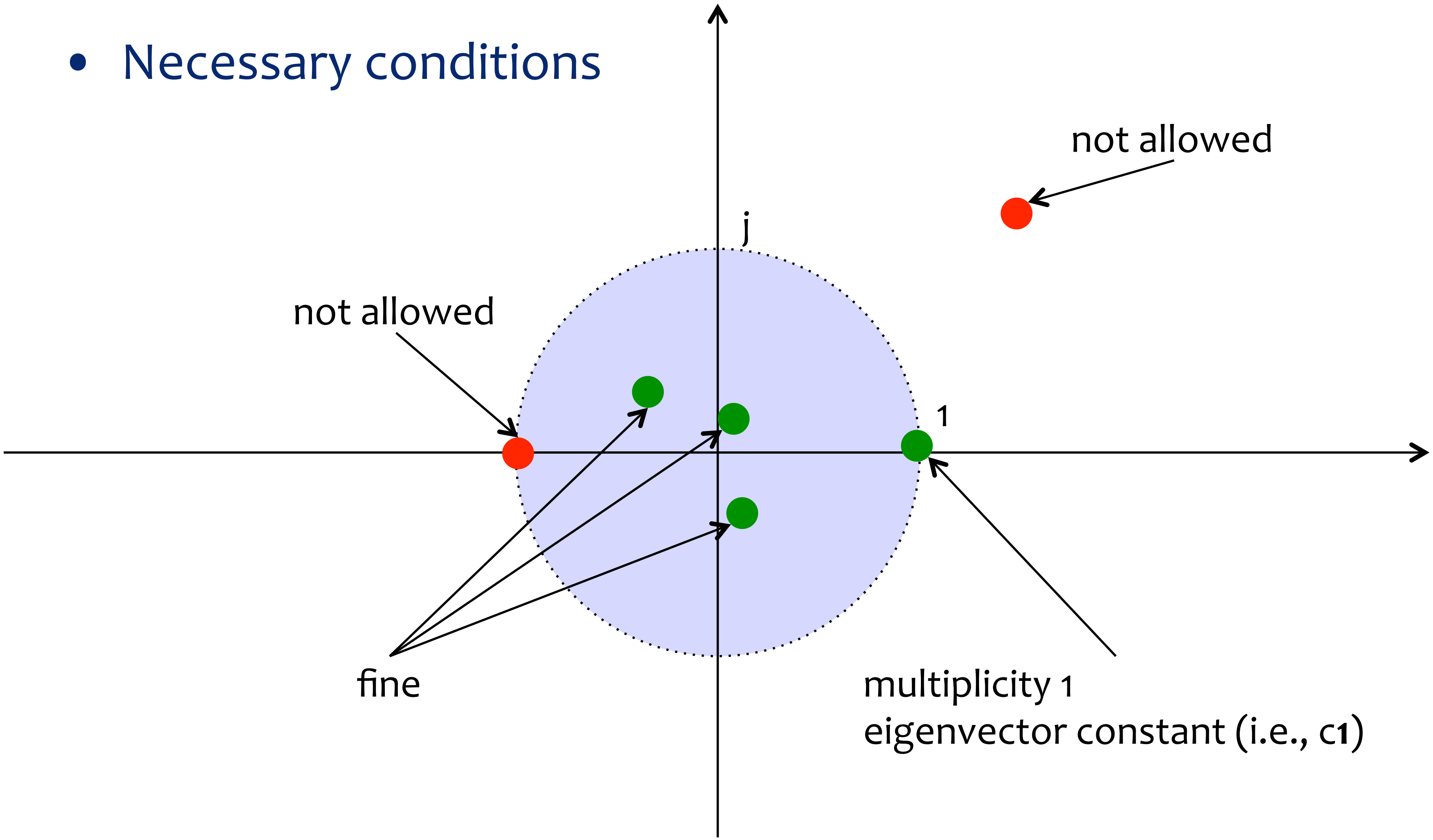


Generalizations

- We had $\mathbf{z}(k) = \mathbf{P}^k \mathbf{z}(0)$
- Properties of P
  - Basis of N orthonormal eigenvectors $\mathbf{v}_1=\mathbf{1}/\sqrt{N}$, $\mathbf{v}_2, \ldots, \mathbf{v}_N$
  - N eigenvalues $\lambda_1=1 > \lambda_2 > \lambda_3, \ldots, > \lambda_N \geq 0$
  - $P_{ij}=0$ when (i,j) is not an edge
- Any P that satisfies these conditions can be used for average consensus
  - Many such matrices exist
  - Varying convergence rates
  - Varying amount of local/global network information

Eigenvalues

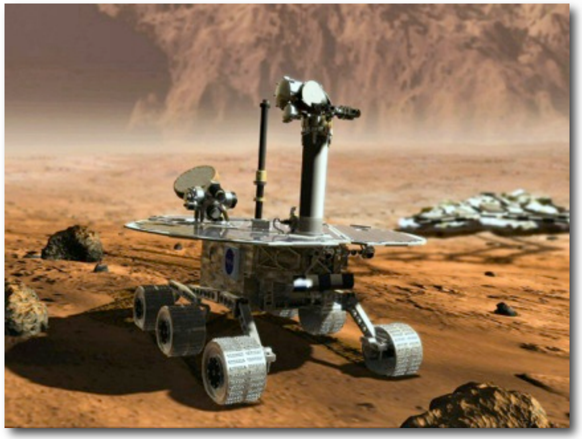
- Necessary conditions



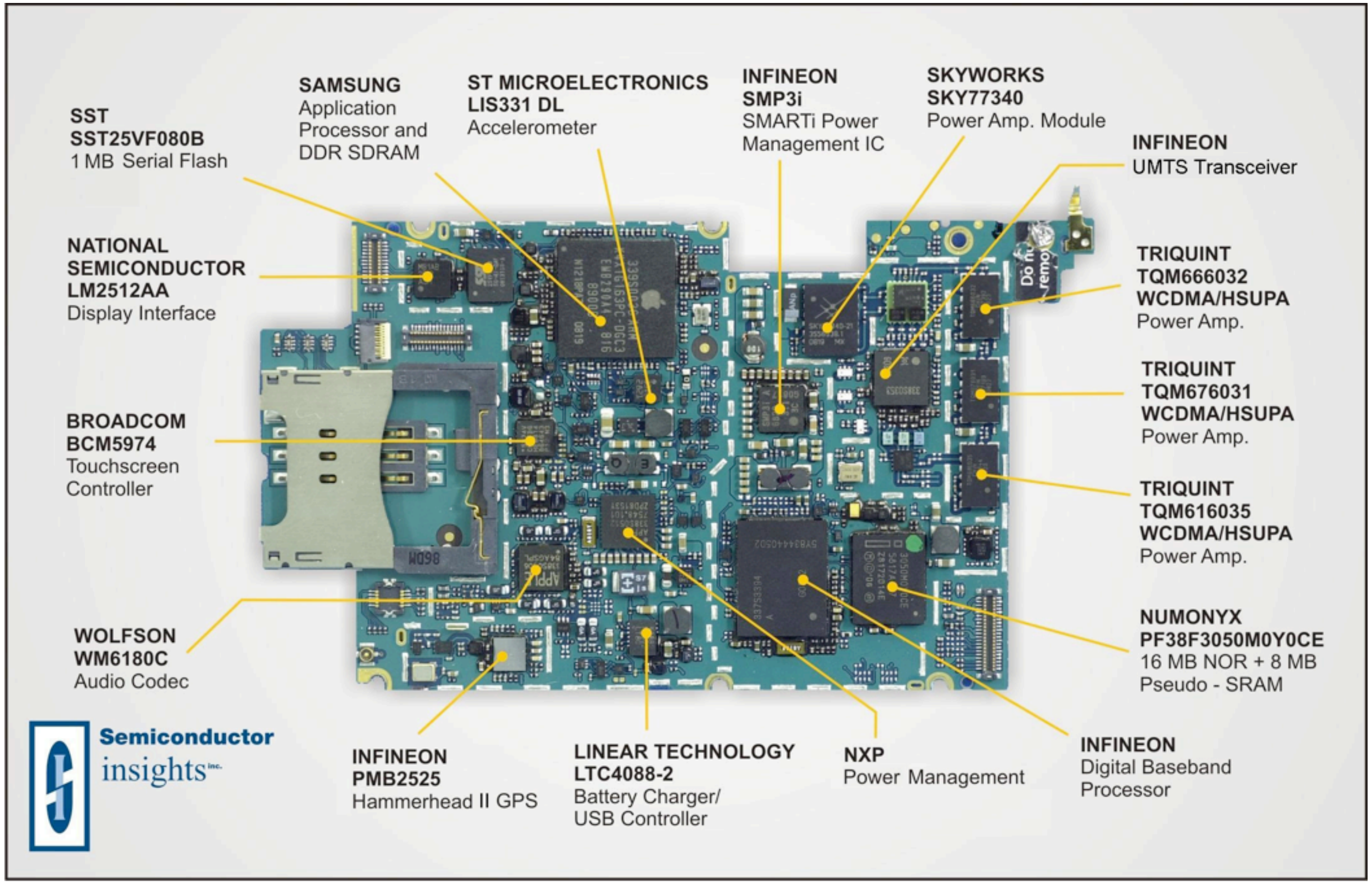
Outline

- Applications
- Background and terminology
- Bayesian detection
- Tool 1: Bayesian graphical models
  - Basics
  - Recipe for Bayesian inference
  - Practicalities
  - A worked example: a digital receiver
- Tool 2: Belief consensus
  - Basics
  - Convergence
- Applications revisited
- Variational interpretation
- Summary and conclusions

Applications



Centralized Processing

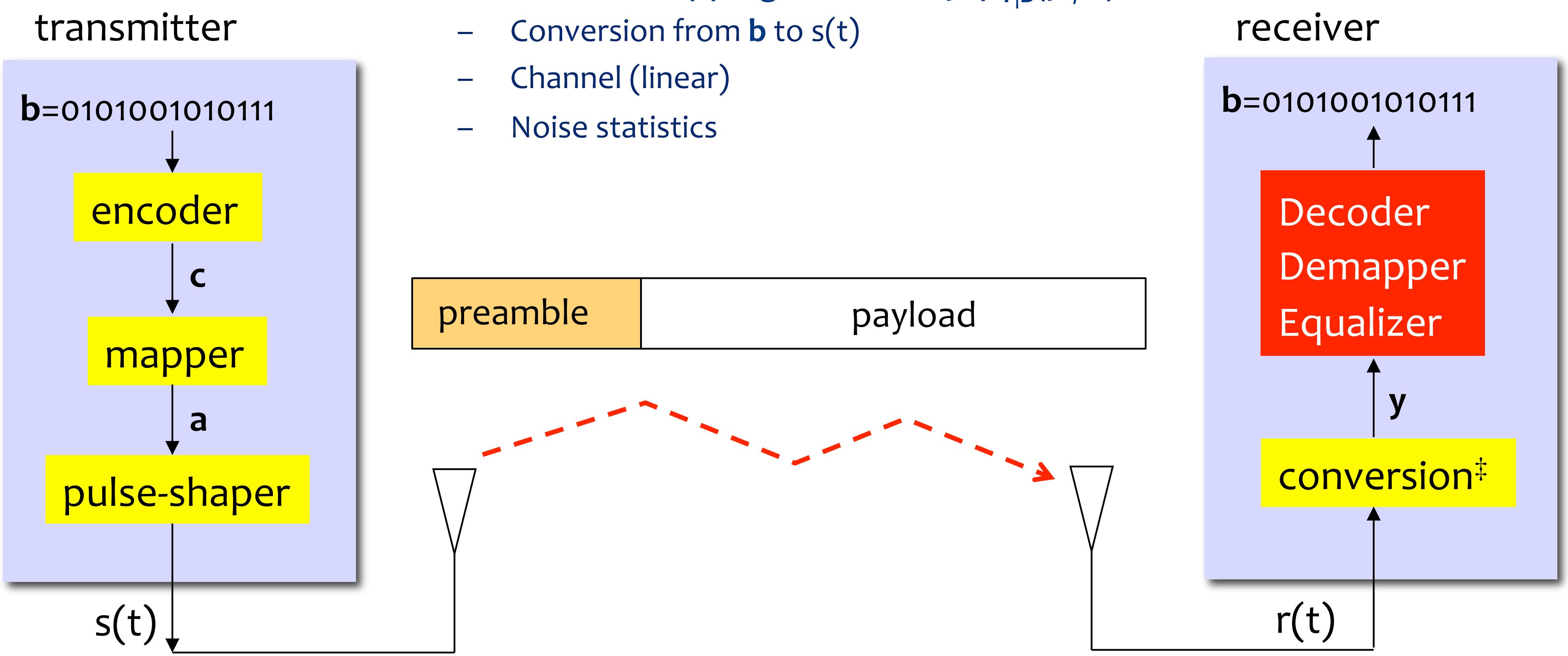


Problem 1: receiver design

- Data detection problem: recover **b** from **y** (optimally)
- Many variations: codes, mapping, channels, antennas, users

The receiver knows

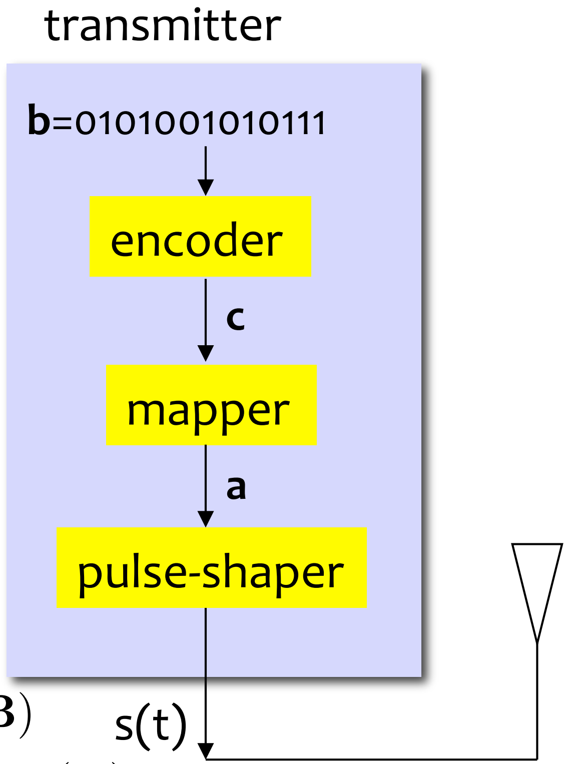
- A priori distribution $p_B(\boldsymbol{b})$
- Probabilistic mapping from **b** to **y**: $p_{Y|B}(\boldsymbol{y}/\boldsymbol{b})$
  - Conversion from **b** to s(t)
  - Channel (linear)
  - Noise statistics



‡involves synchronization, channel estimation, filtering, sampling, etc

Harness structure

- Transmitter
  - Encode **b** to **c**=f_C(**b**)
  - Map **c** to **a**=f_A(**c**)
  - Pulse-shape **a** to s(t)



- Factorize p(**B,C,A,Y=y**)

$$\begin{aligned} p(\mathbf{B},\mathbf{C},\mathbf{A},\mathbf{Y}=\mathbf{y}) &= p(\mathbf{Y}=\mathbf{y}|\mathbf{B},\mathbf{C},\mathbf{A})p(\mathbf{B},\mathbf{C},\mathbf{A}) \\ &= p(\mathbf{Y}=\mathbf{y}|\mathbf{A})p(\mathbf{B},\mathbf{C},\mathbf{A}) \\ &= p(\mathbf{Y}=\mathbf{y}|\mathbf{A})p(\mathbf{A}|\mathbf{B},\mathbf{C})p(\mathbf{C}|\mathbf{B})p(\mathbf{B}) \\ &= p(\mathbf{Y}=\mathbf{y}|\mathbf{A})\underbrace{p(\mathbf{A}|\mathbf{C})}_{\mathbb{I}\{\mathbf{A}=f_A(\mathbf{C})\}}\underbrace{p(\mathbf{C}|\mathbf{B})}_{\mathbb{I}\{\mathbf{C}=f_C(\mathbf{B})\}}p(\mathbf{B}) \end{aligned}$$

- Usually iid bits $p(\mathbf{B}) = \prod_k p(B_k)$

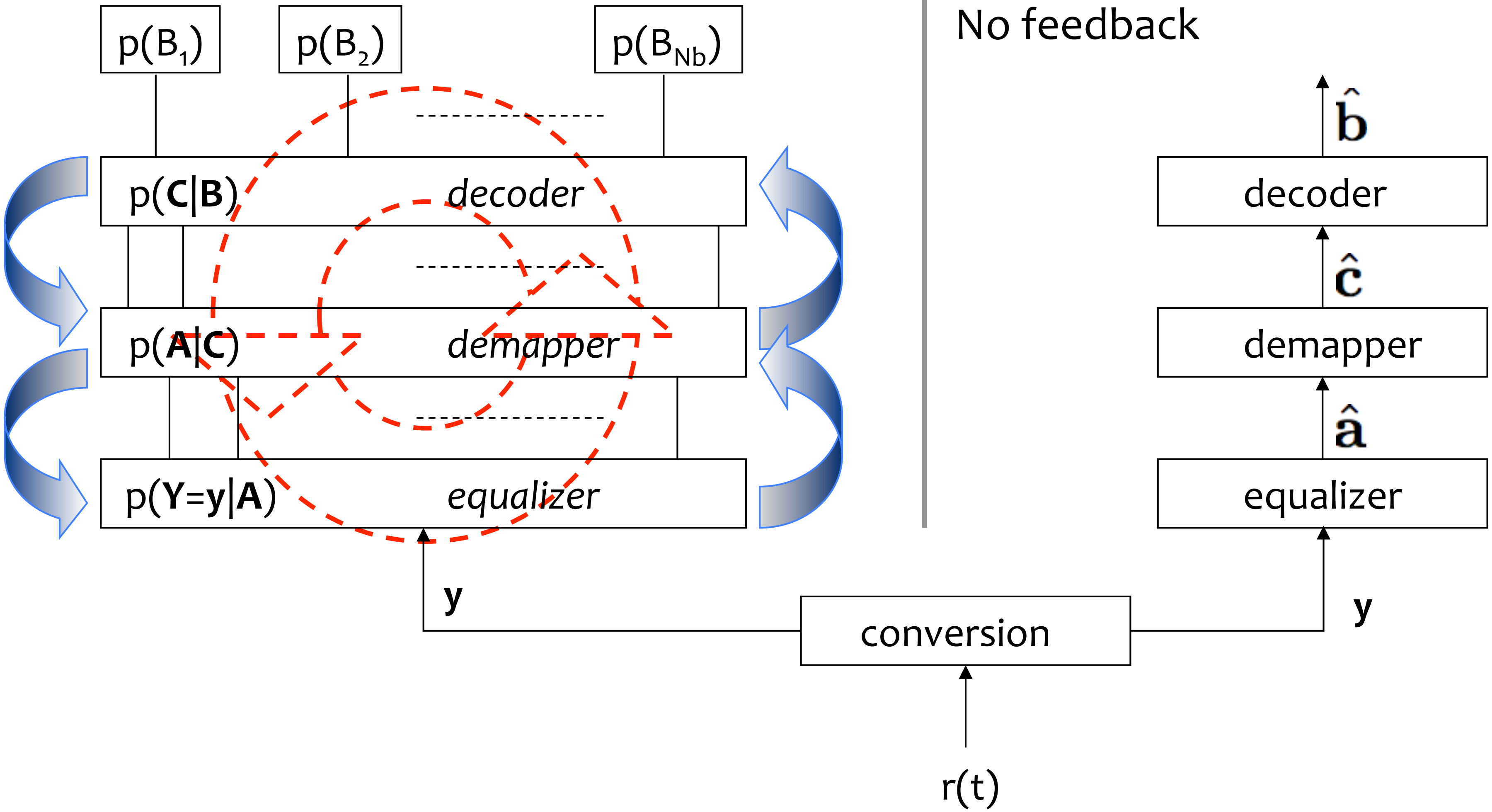
Overall factor graph for digital receivers

Factor graph receiver

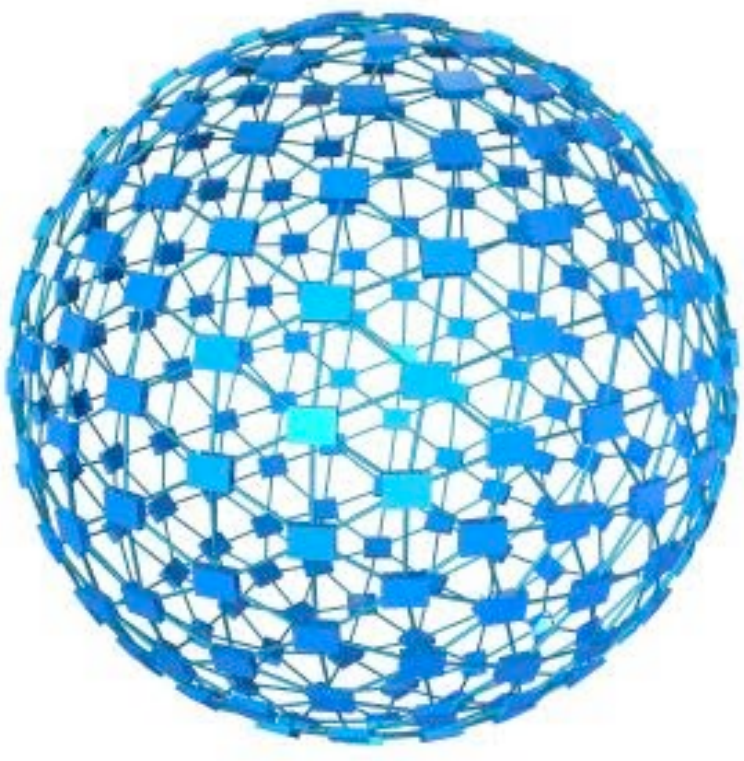
$$p(\mathbf{B},\mathbf{C},\mathbf{A},\mathbf{Y}=\mathbf{y}) = p(\mathbf{Y}=\mathbf{y}|\mathbf{A})p(\mathbf{A}|\mathbf{C})p(\mathbf{C}|\mathbf{B})p(\mathbf{B})$$

Conventional receiver

Hard decisions
No feedback

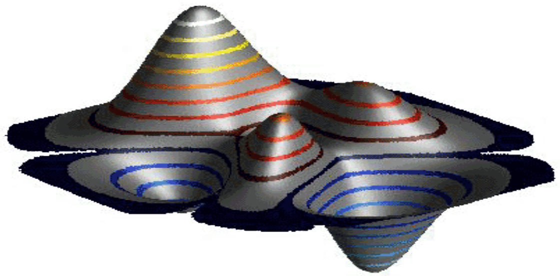


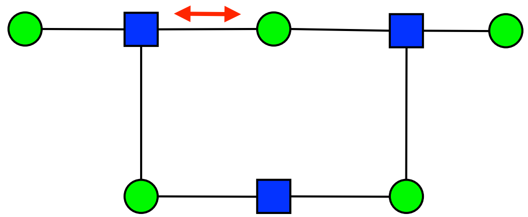
Network inference with factor graphs



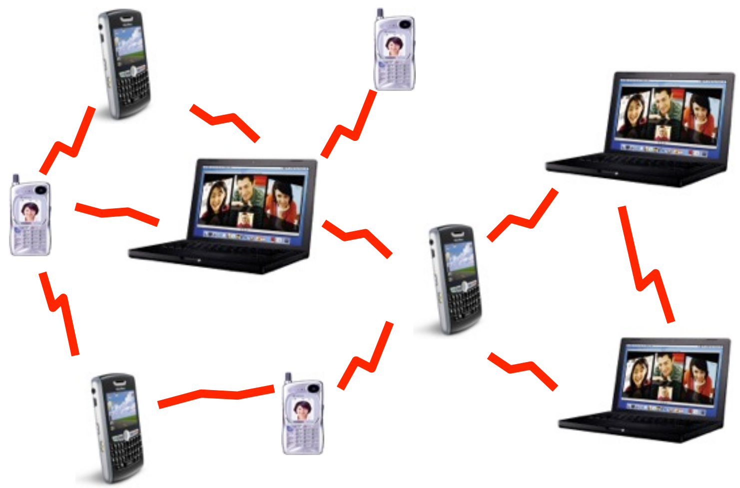
Graphical models and wireless networks

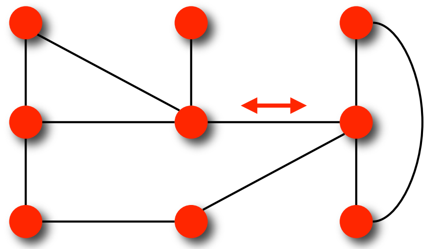
1. Graphical Models





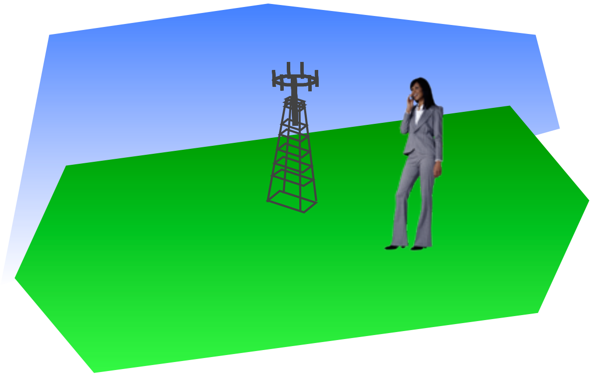
2. Wireless Networks



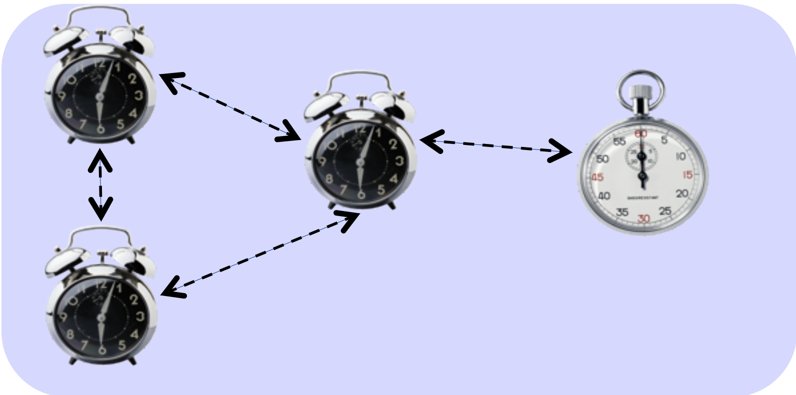


Four applications

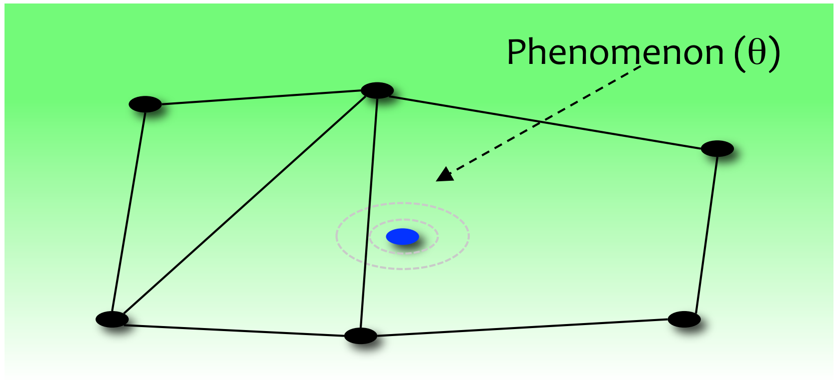
cooperative localization



distributed beamforming



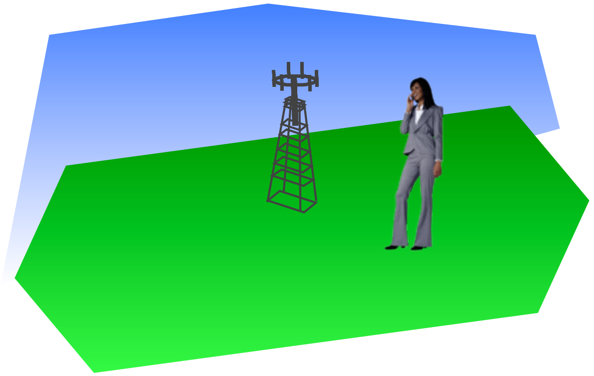
cooperative synchronization



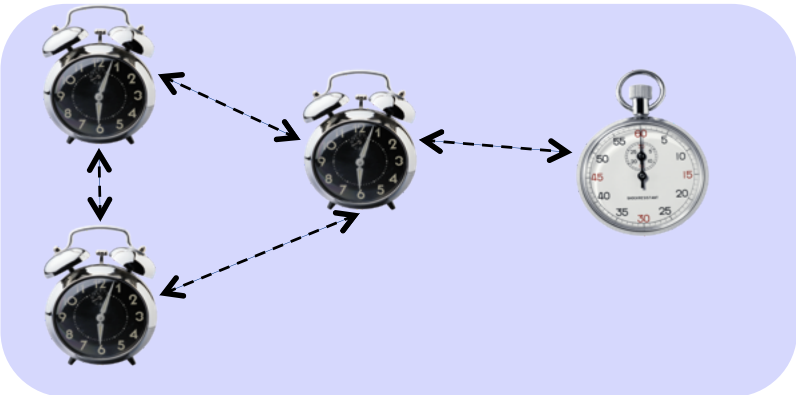
distributed estimation

Four applications

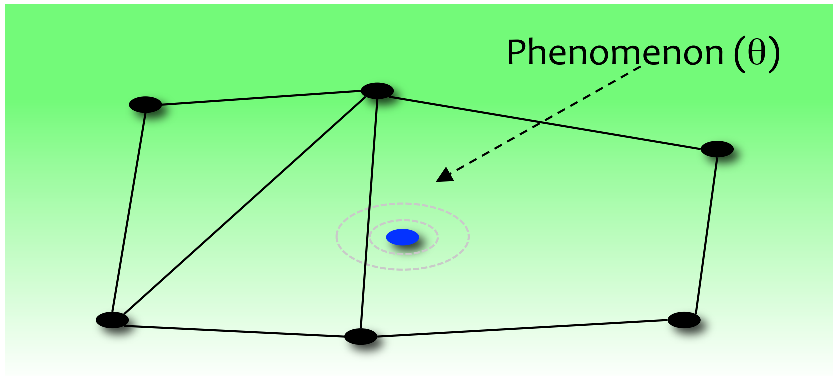
cooperative localization



distributed beamforming

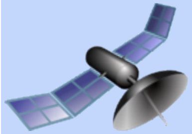
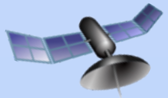
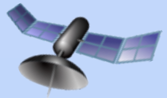
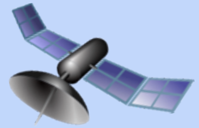


cooperative synchronization



distributed estimation

Cooperative localization

Scenario

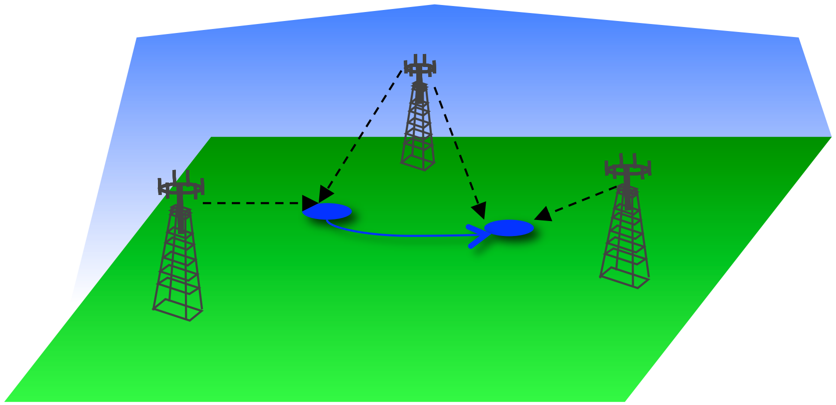
- N mobile nodes (agents) with **unknown** positions $x_1(t),\ldots, x_N(t)$
- M reference nodes (anchors) with **known** positions $x_A(t)$, $x_B(t)$, $x_C(t)$, …
- **Measurements** $z_{ij}(t)$ between $x_i(t)$ and $x_j(t)$, where j is mobile or reference

Goals

- For every mobile node to determine its own position, given all measurements up to now

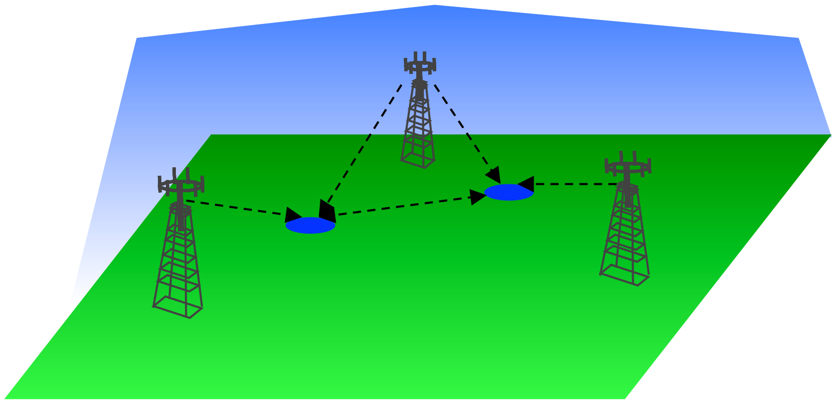
Two non-trivial cases

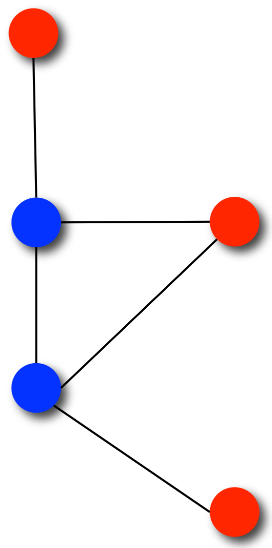
- One mobile agent

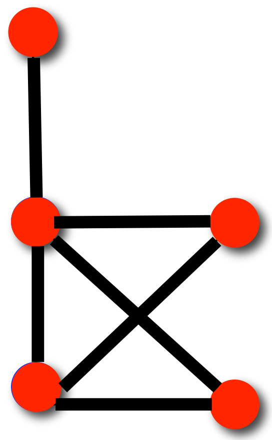


"After abstracting a problem, look at the simplest nontrivial version."
Robert Gallager, 1998, Shannon Day, Bell Labs

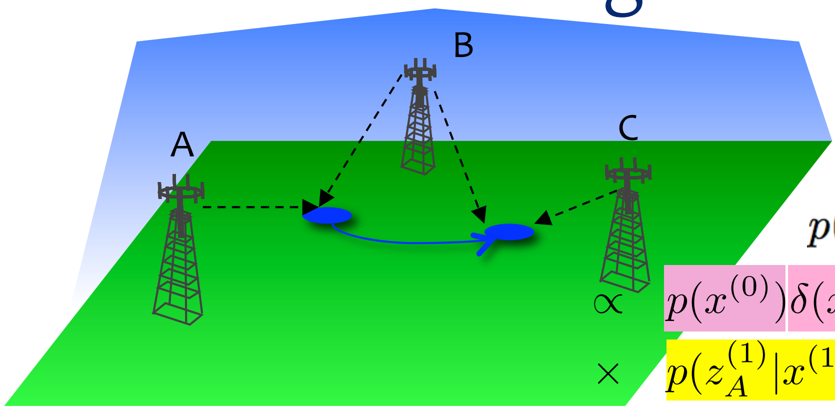
- Multiple static agents







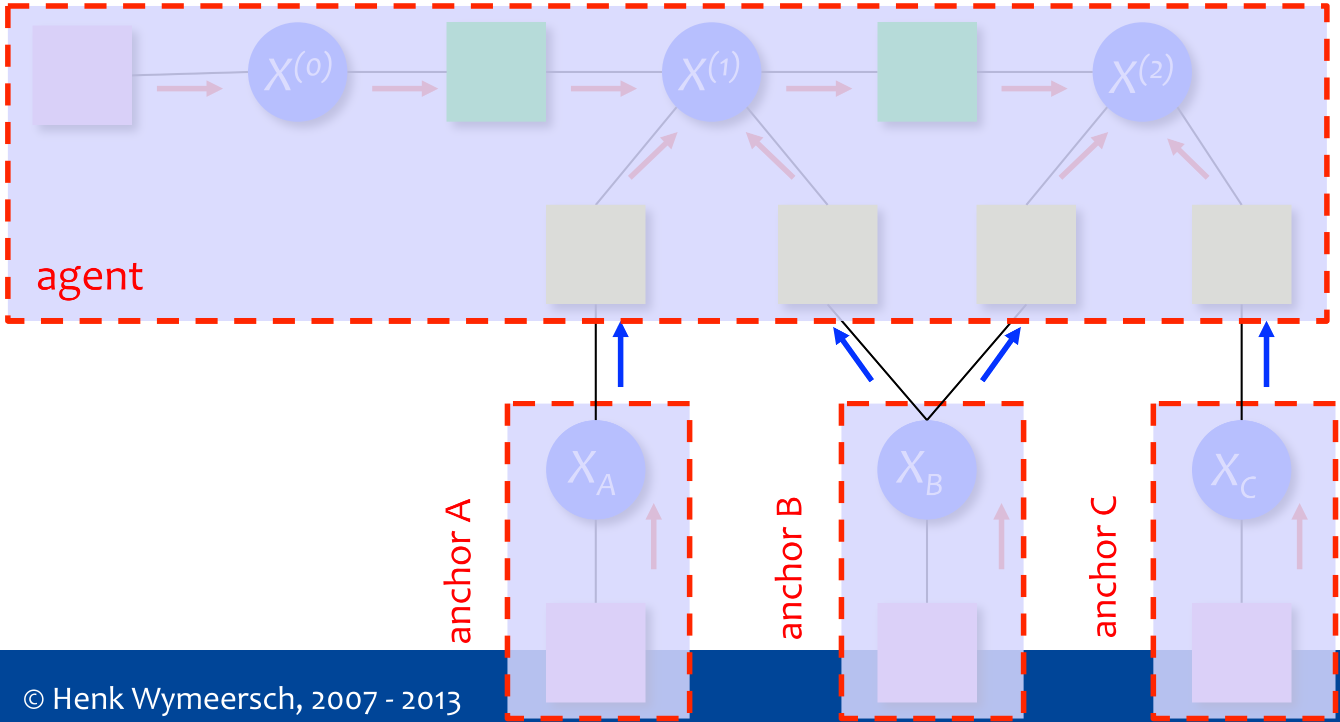
One mobile agent



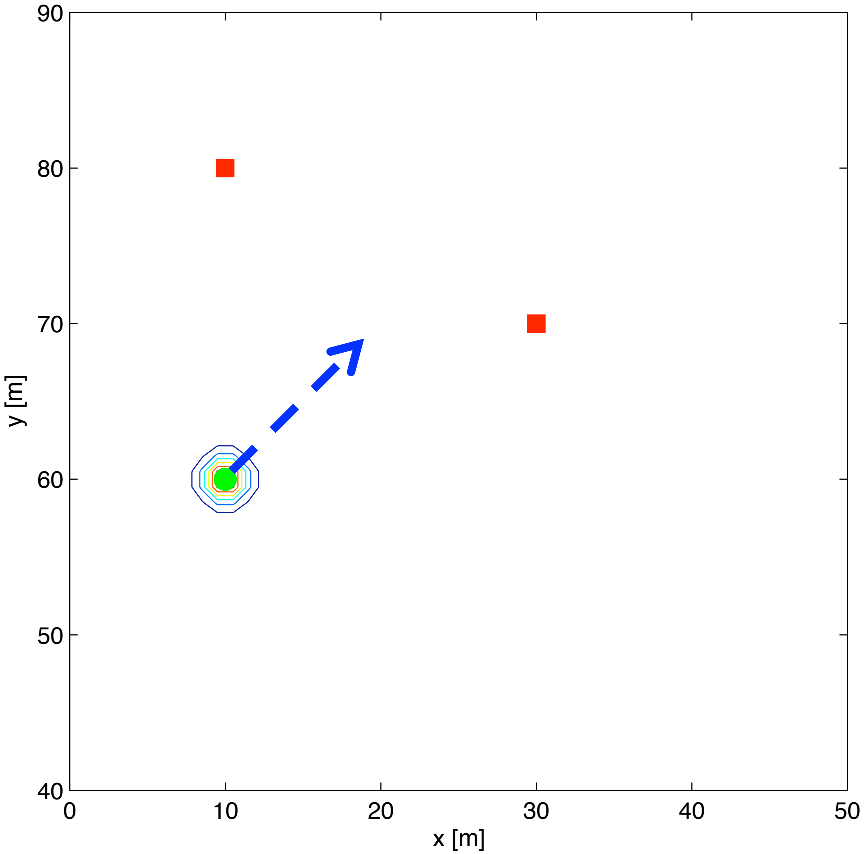
$$p(x^{(0)}, x^{(1)}, x^{(2)}, x_A, x_B, x_C | s^{(1:2)}, z_A^{(1)}, z_B^{(1:2)}, z_C^{(2)})$$

$$\propto p(x^{(0)})\delta(x_A - x_A^*)\delta(x_B - x_B^*)\delta(x_C - x_C^*)p(x^{(1)}|x^{(0)}, s^{(1)})p(x^{(2)}|x^{(1)}, s^{(2)})$$

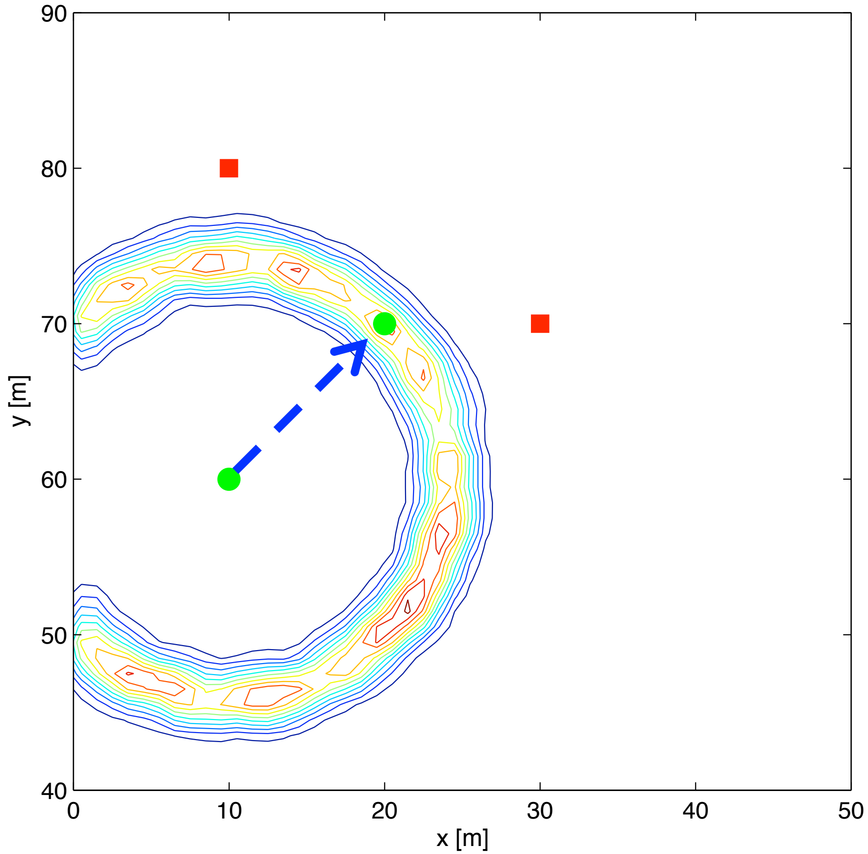
$$\times\, p(z_A^{(1)}|x^{(1)}, x_A)p(z_B^{(1)}|x^{(1)}, x_B)p(z_C^{(2)}|x^{(2)}, x_C)p(z_B^{(2)}|x^{(2)}, x_B)$$



Example: one mobile agent

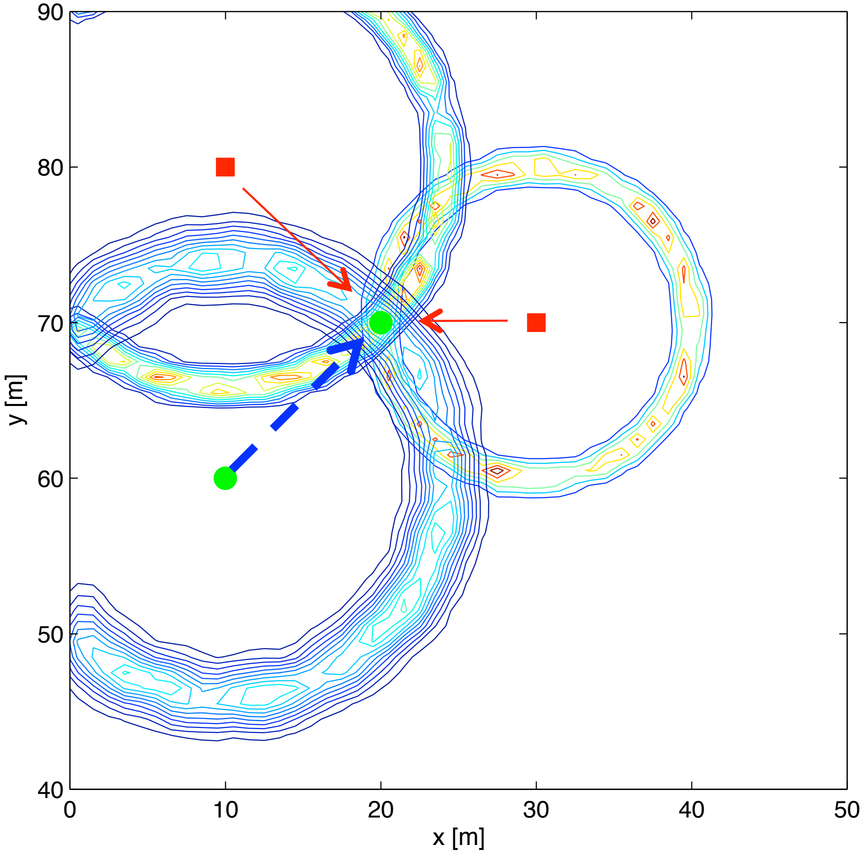


Before movement

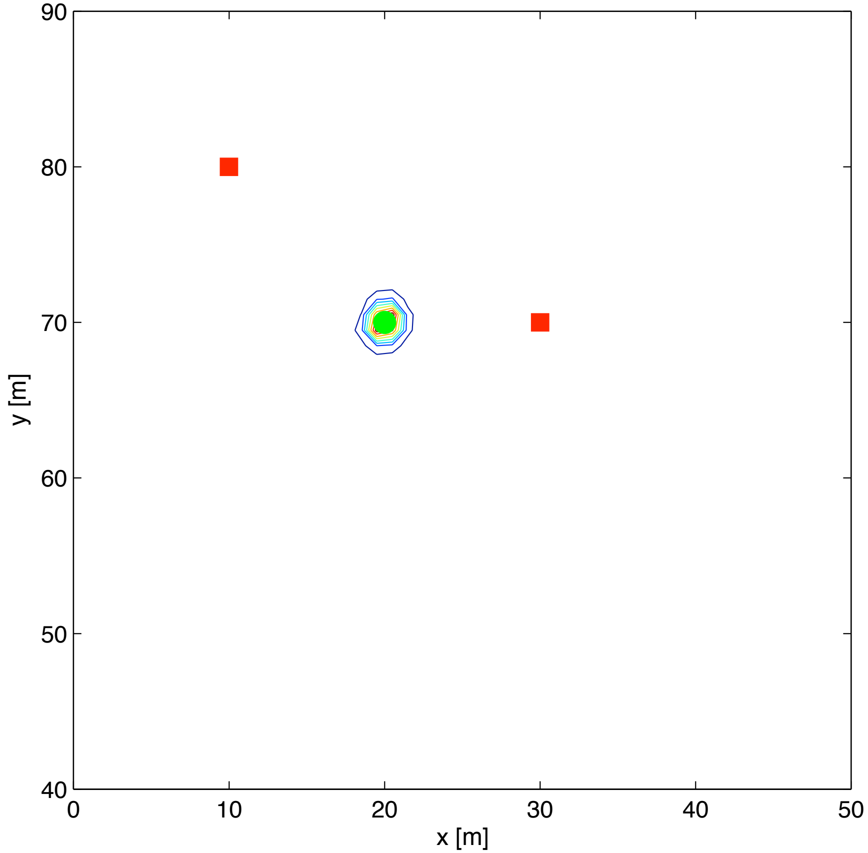


After movement

Example: one mobile agent

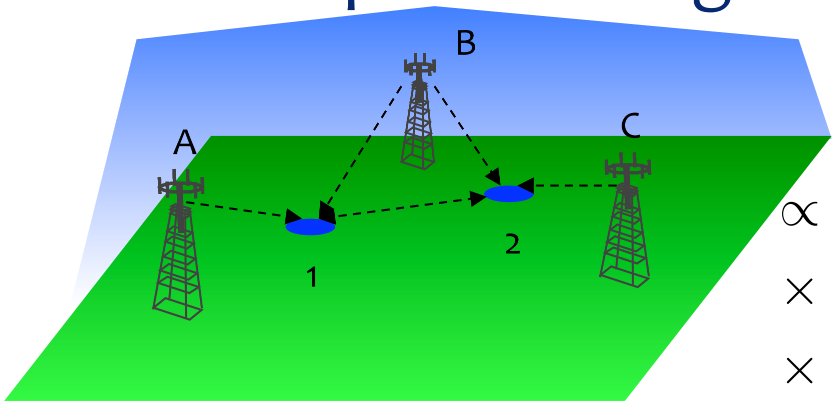


Information from anchors

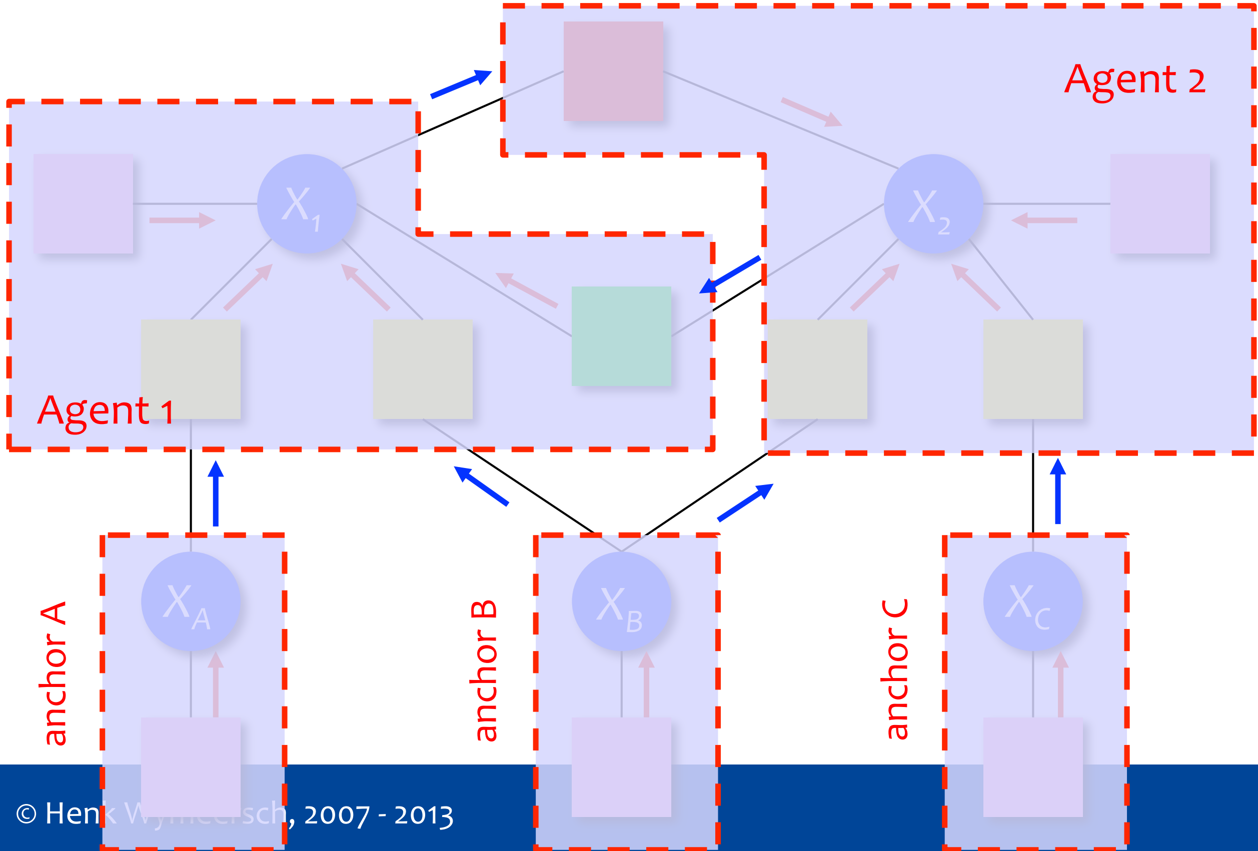


Fuse information

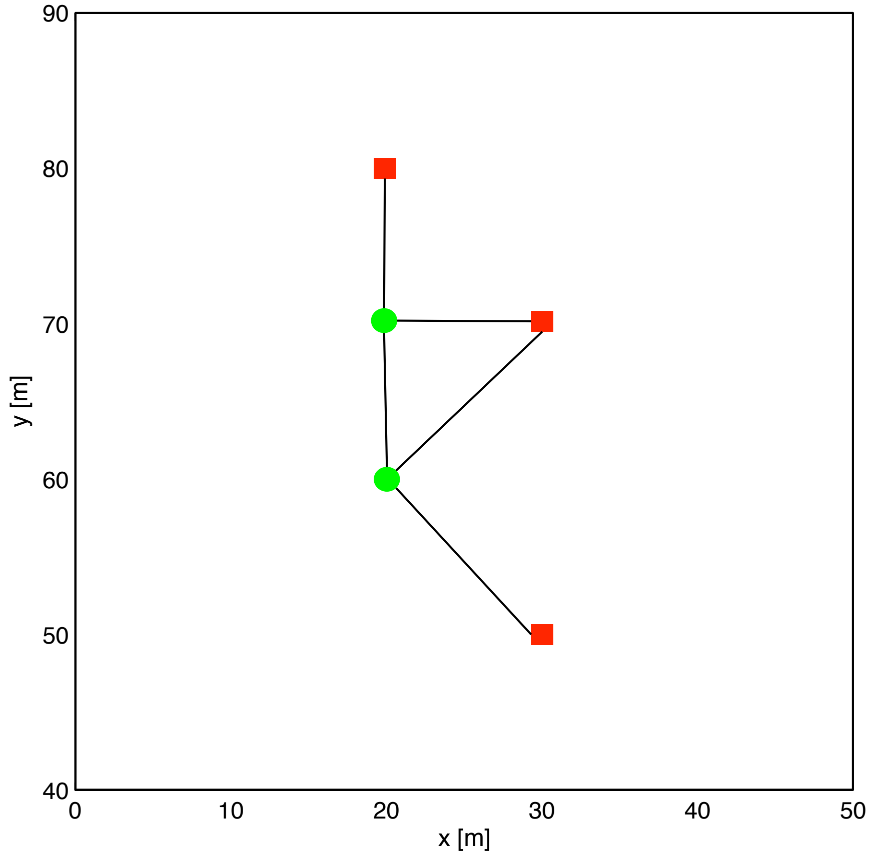
Multiple static agents

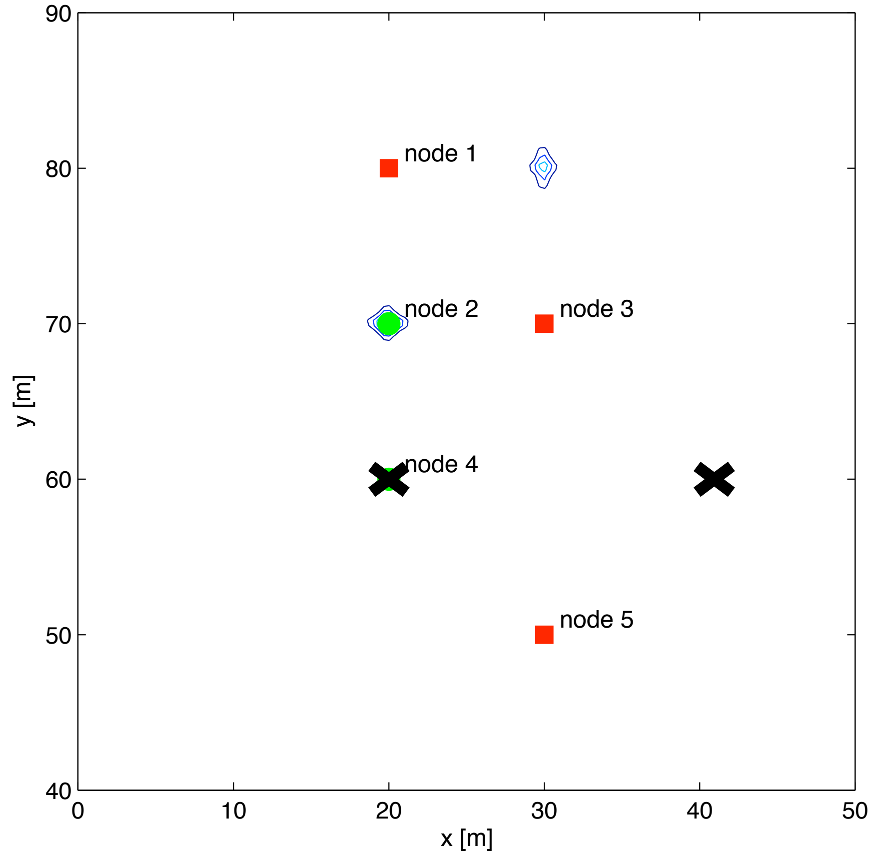


$$p(x_1, x_2, x_A, x_B, x_C | z_{A,1}, z_{B,1}, z_{B,2}, z_{C,2}, z_{1,2}, z_{2,1})$$
$$\propto p(x_1)p(x_2)\delta(x_A - x_A^*)\delta(x_B - x_B^*)\delta(x_C - x_C^*)$$
$$\times p(z_{A,1}|x_1, x_A)p(z_{B,1}|x_1, x_B)p(z_{B,2}|x_2, x_B)$$
$$\times p(z_{1,2}|x_1, x_2)p(z_{2,1}|x_1, x_2)p(z_{C,2}|x_2, x_C)$$



Example: Multiple Static Agents

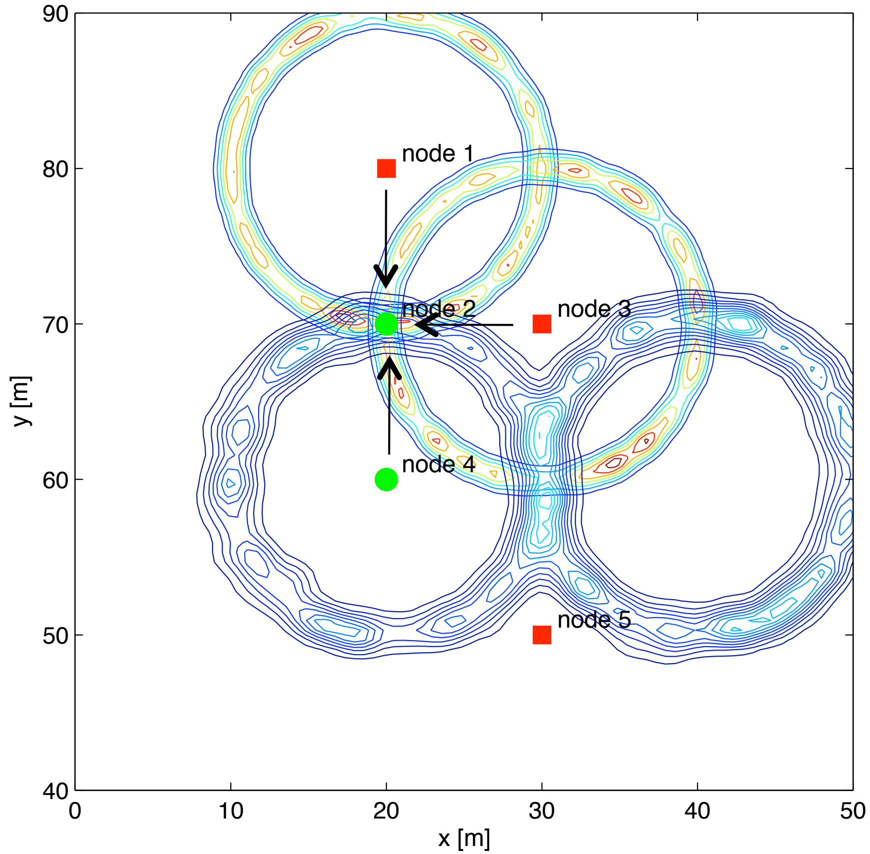


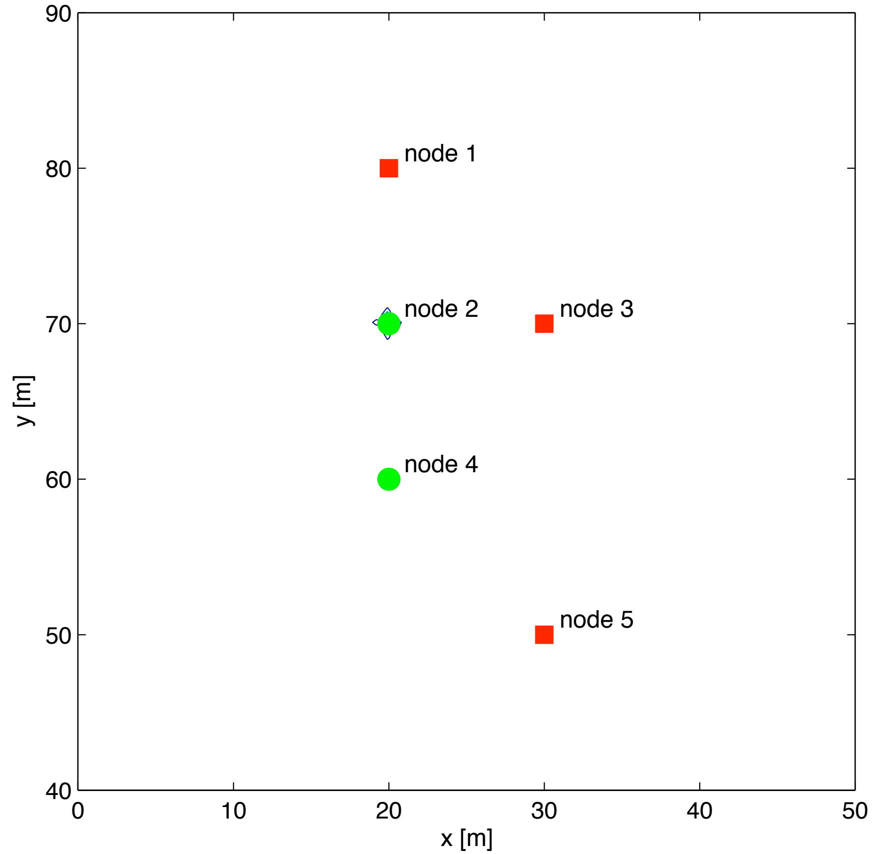


Information from anchors

Fuse information

Example: Multiple Static Agents



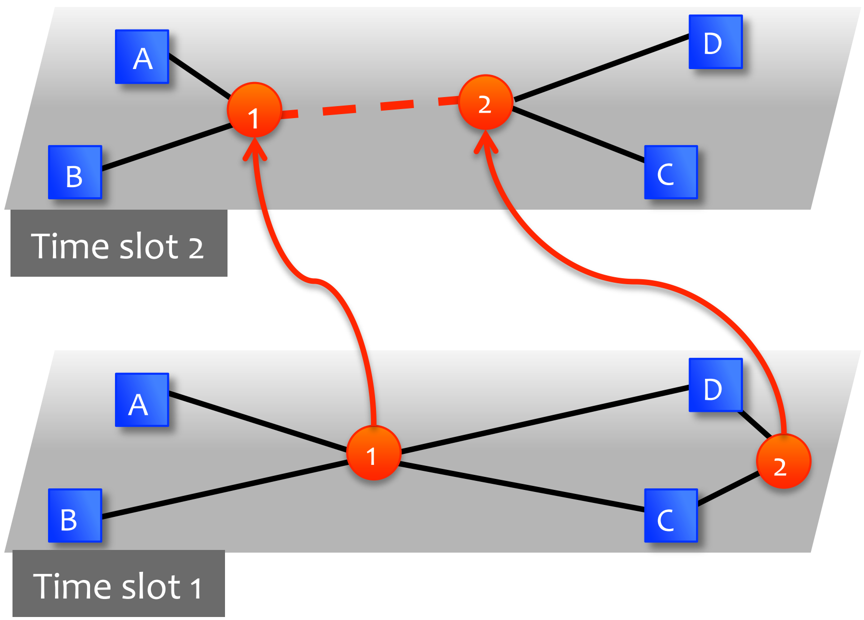


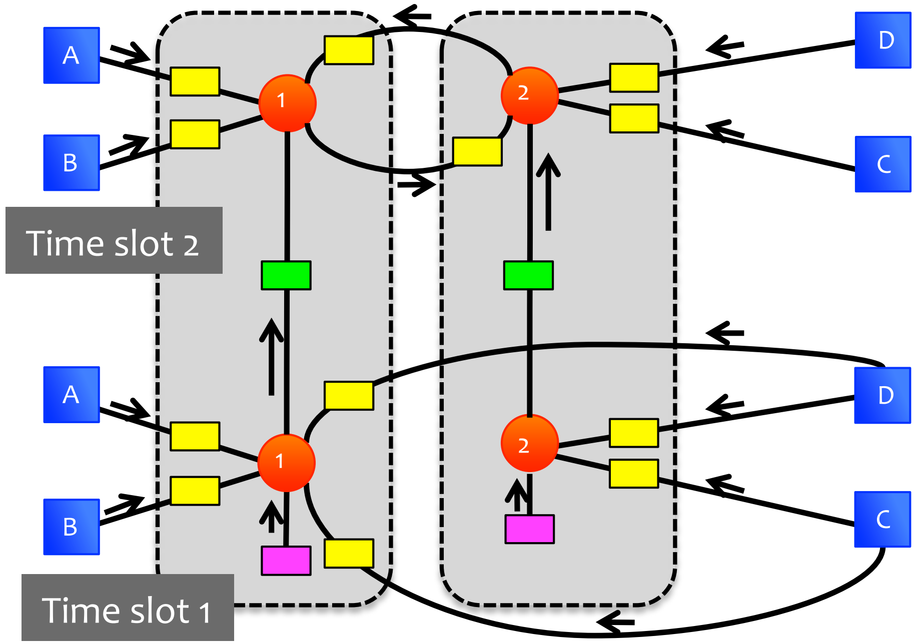
Information from anchors and agent 4

Fuse information

A space-time factor graph?

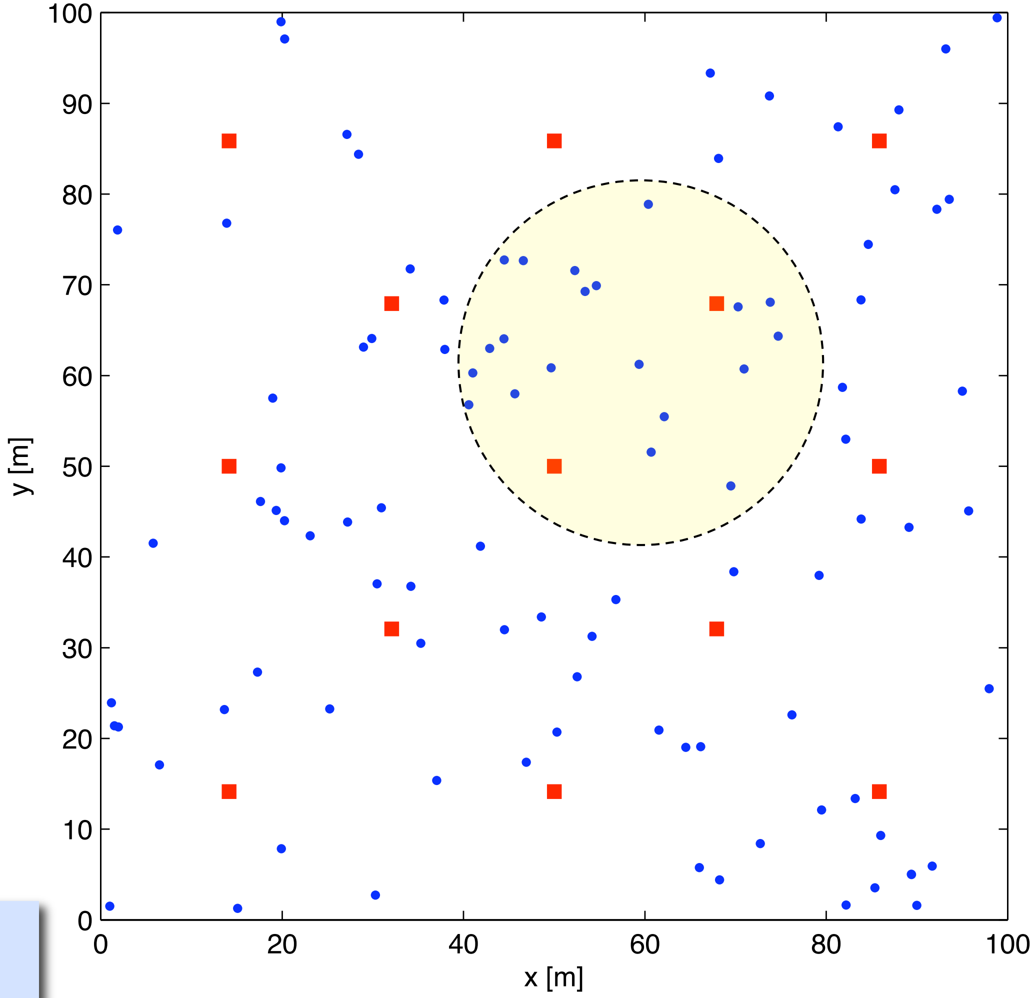
- Factor graph over space and time
  - FG is organic: nodes require only local knowledge of this FG
  - Schedule can be opportunistic, adaptive
  - Network can operate in continuous time
  - Possibility to include fault detection, control, planning, …





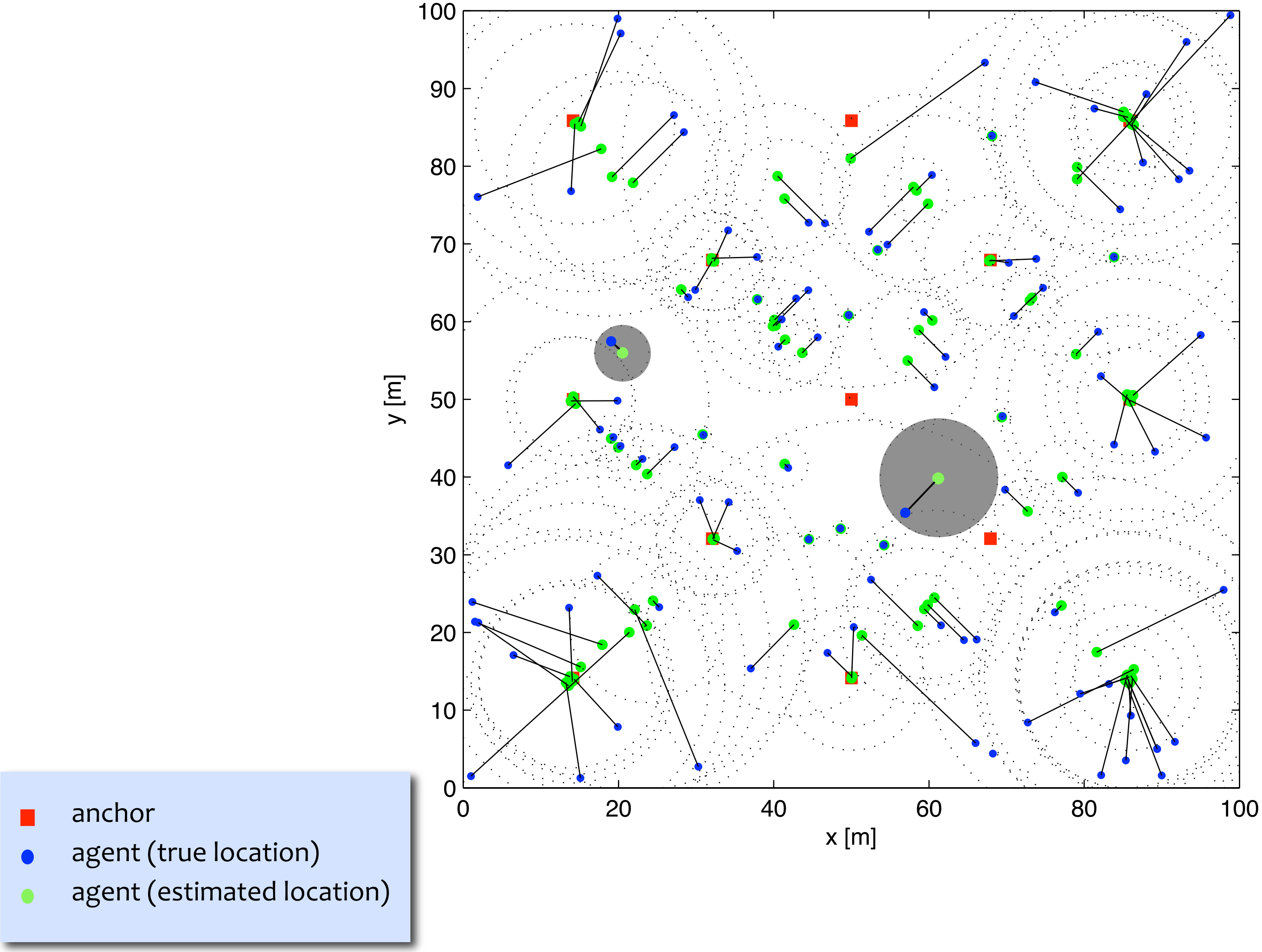
[Ihler et al, 2005, Wymeersch et al, 2009]

Large-scale example

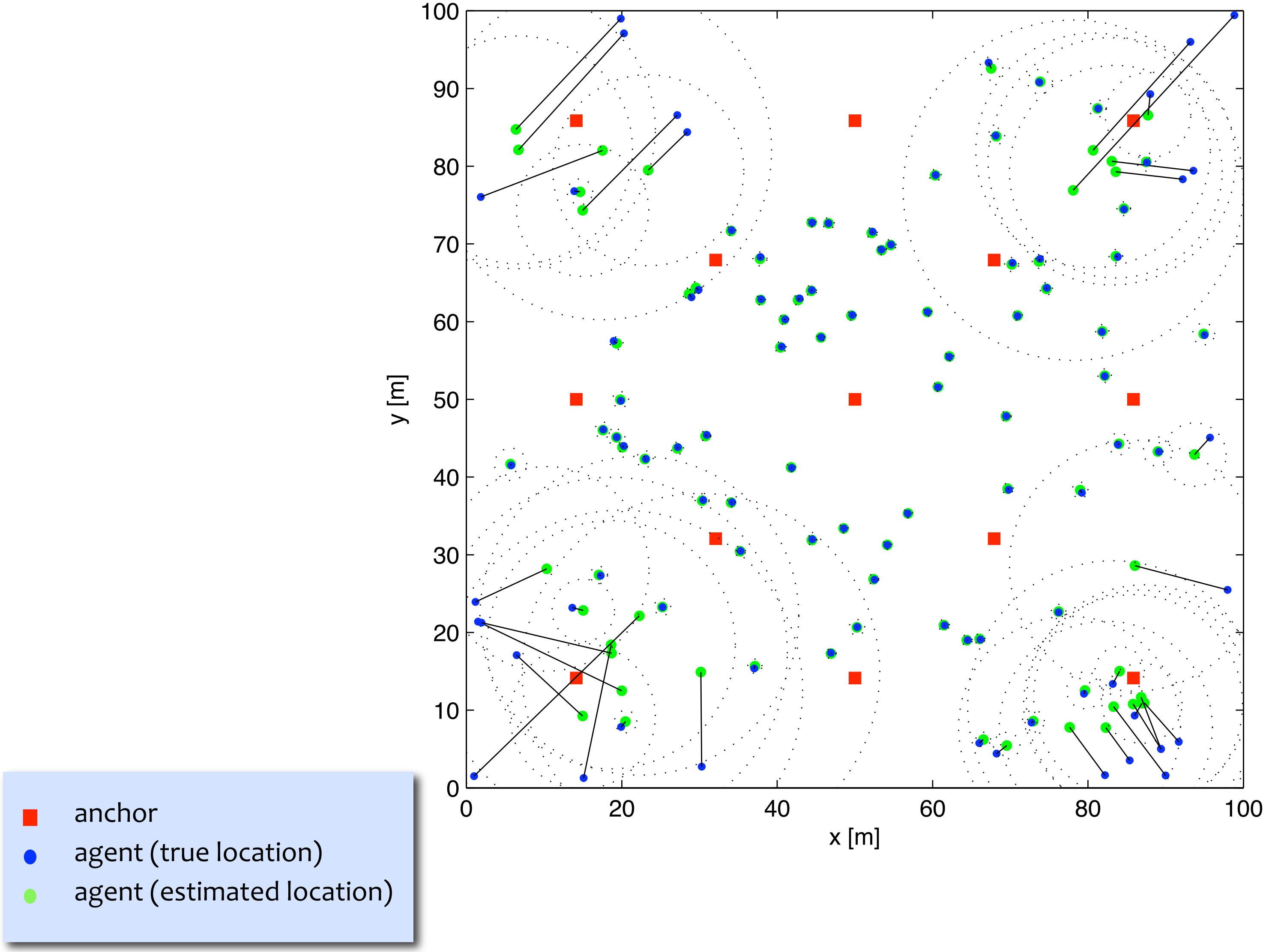


- anchor
- agent (true location)
- agent (estimated location)

Without cooperation

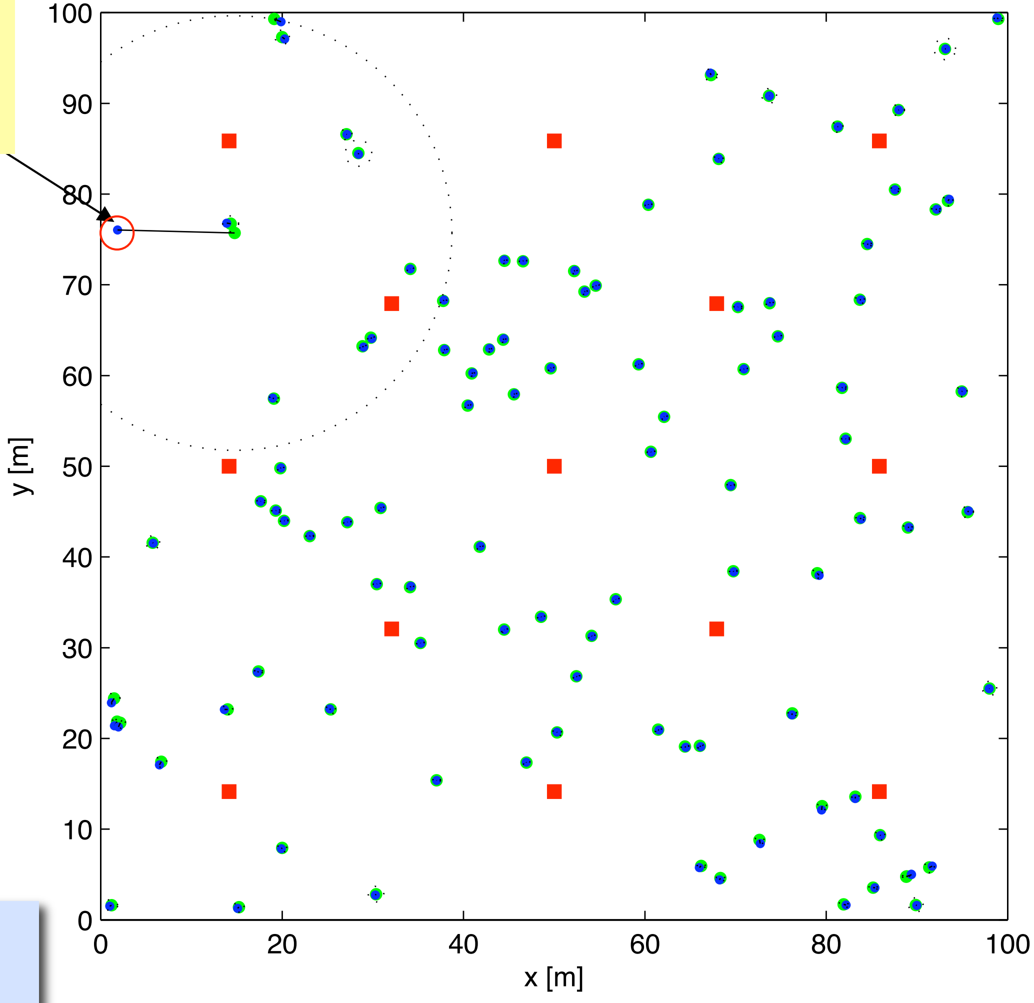


With cooperation



With cooperation

Degree 2: can never be localized without ambiguity



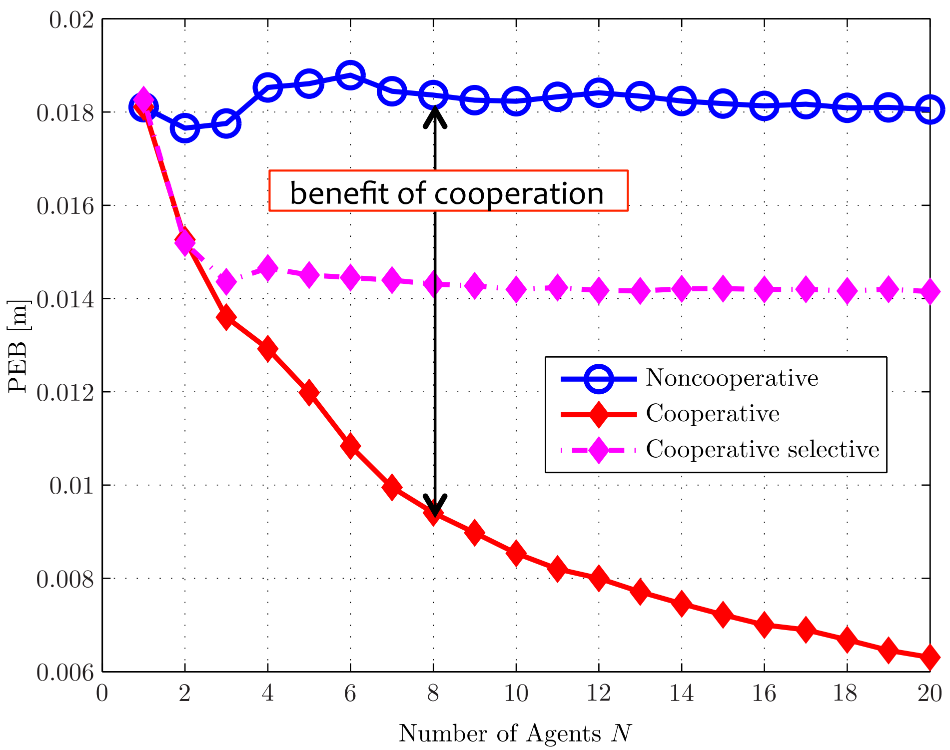
anchor
agent (true location)
agent (estimated location)

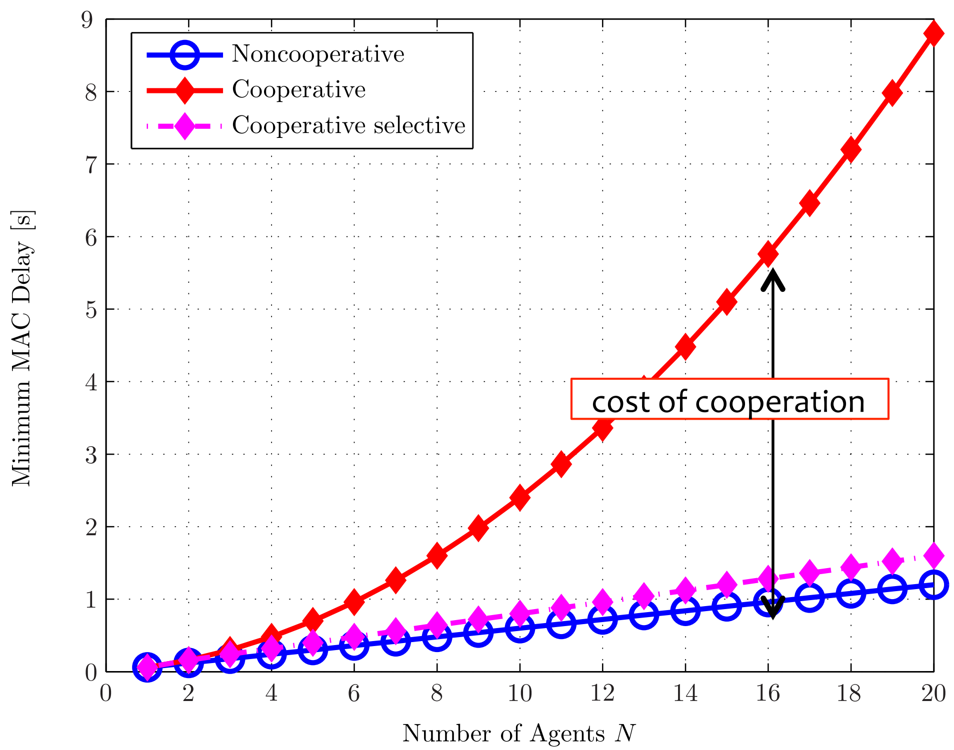
Cooperation is not for free

Scenario

[Garcia et al, 2013]

- 3 anchors, increasing number of agents, dense network, realistic ranging performance, packet delay
- Performance criteria: Cramér-Rao bound (left) vs MAC delay (right)

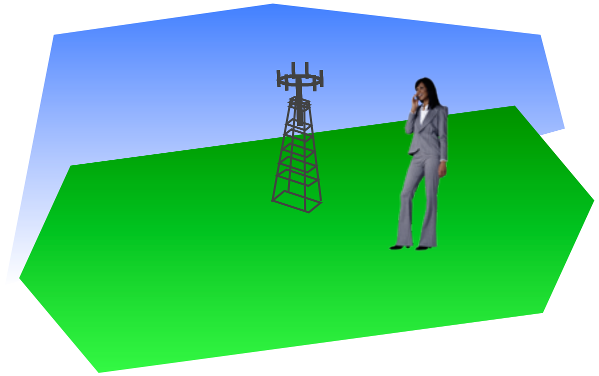




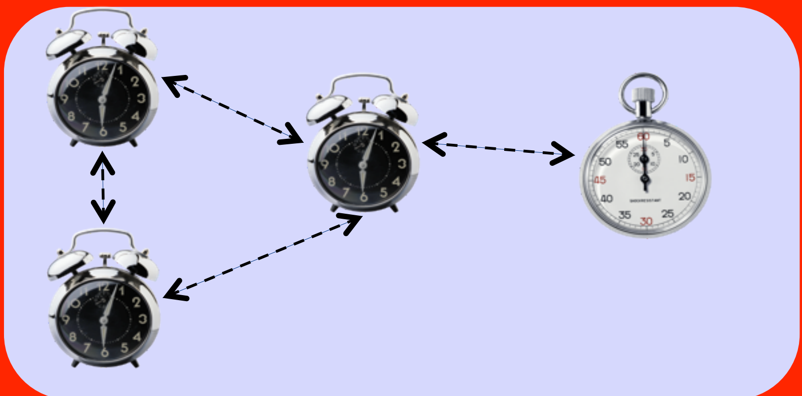
Experiments

Four applications

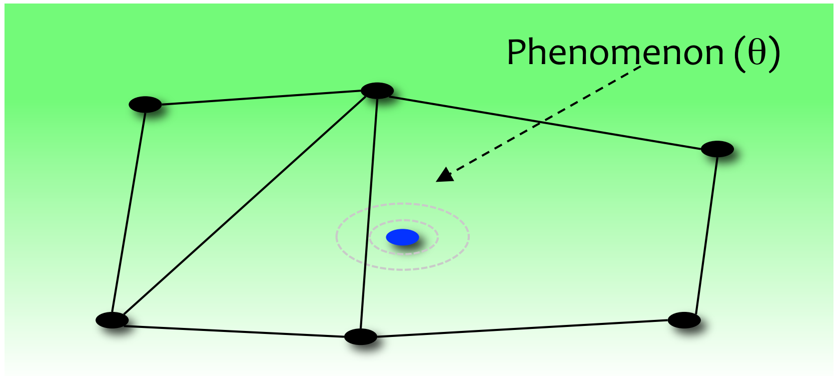
cooperative localization



distributed beamforming

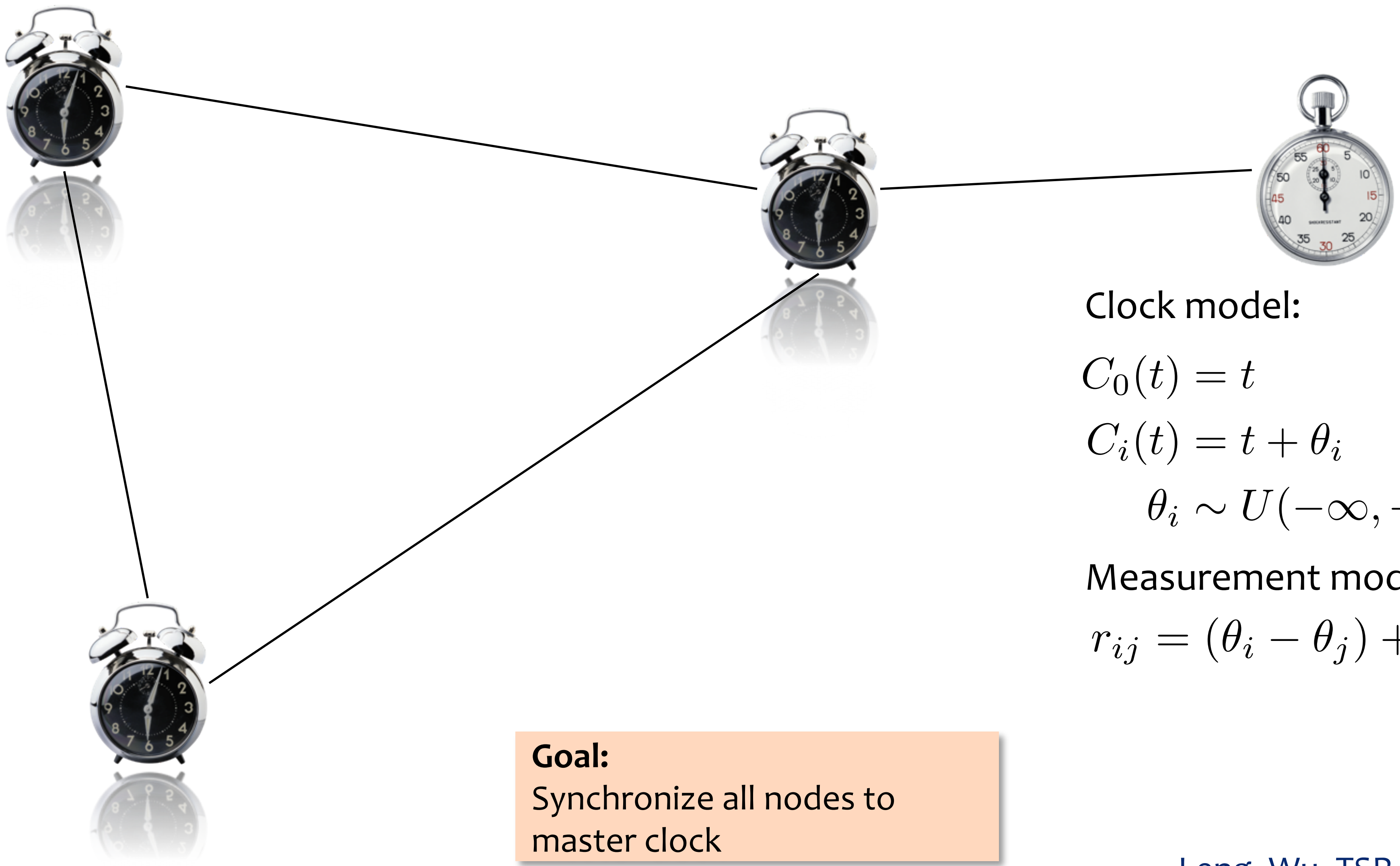


cooperative synchronization



distributed estimation

Network synchronization



Clock model:

$$C_0(t) = t$$

$$C_i(t) = t + \theta_i$$

$$\theta_i \sim U(-\infty, +\infty)$$

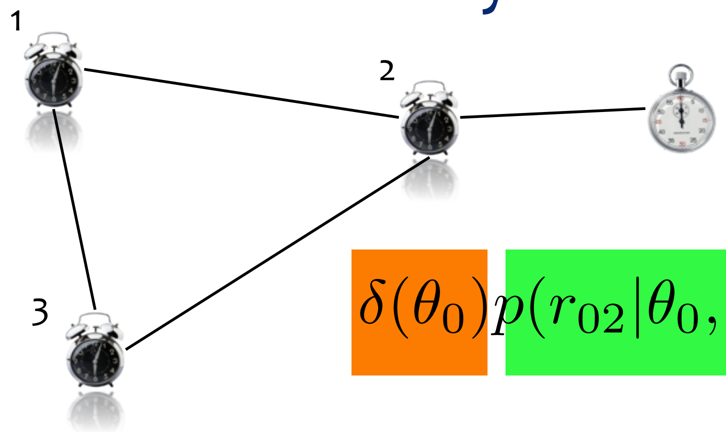
Measurement model:

$$r_{ij} = (\theta_i - \theta_j) + n_{ij}$$

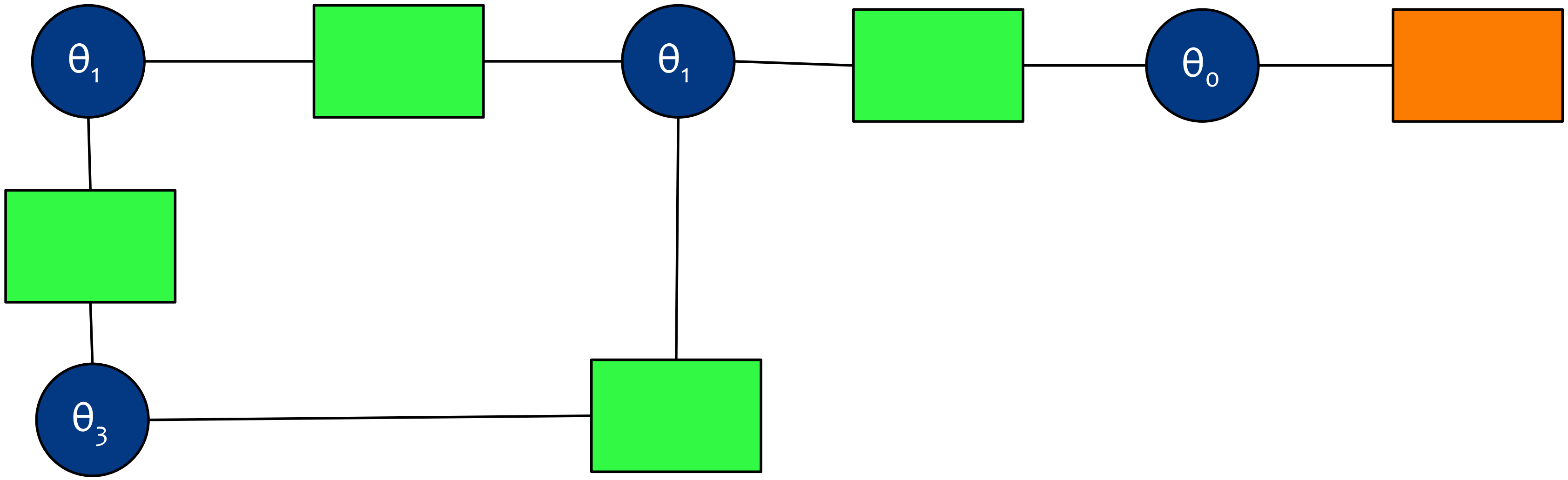
Goal:
Synchronize all nodes to master clock

Leng, Wu, TSP 2011

Network synchronization

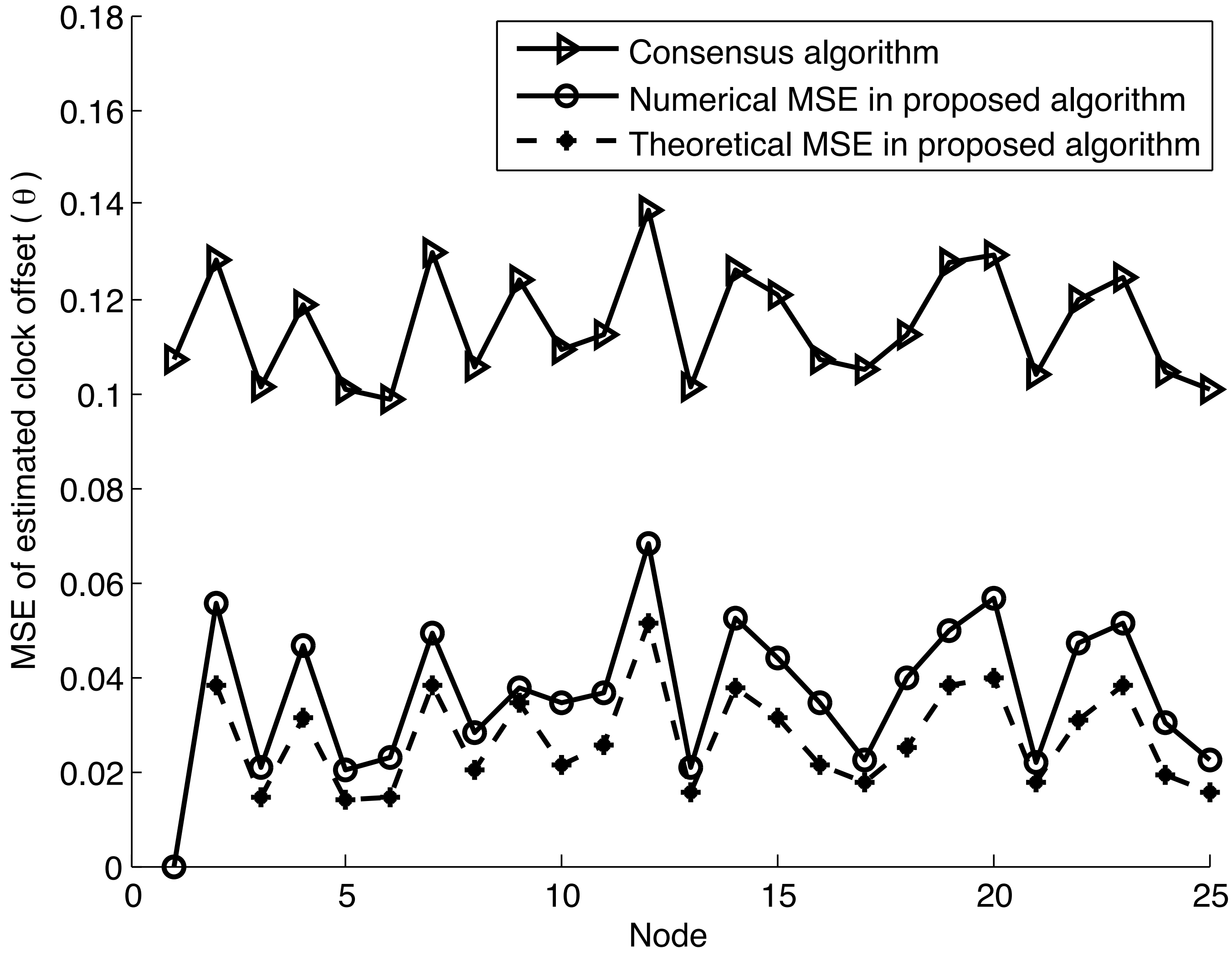


$$p(\theta_{0:4}|r_{02}, r_{12}, r_{23}, r_{13}) =$$
$$\delta(\theta_0) p(r_{02}|\theta_0, \theta_2) p(r_{12}|\theta_1, \theta_2) p(r_{23}|\theta_2, \theta_3) p(r_{13}|\theta_1, \theta_3)$$



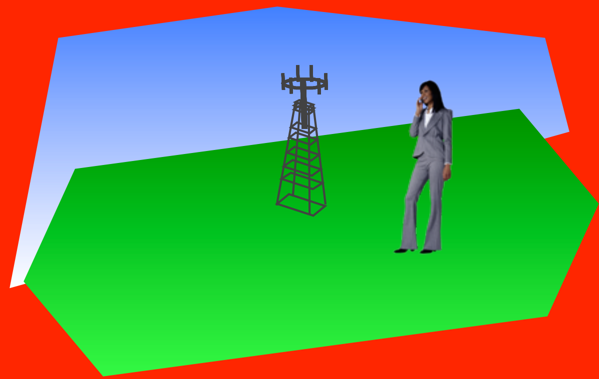
Performance example

Leng, Wu, TSP 2011

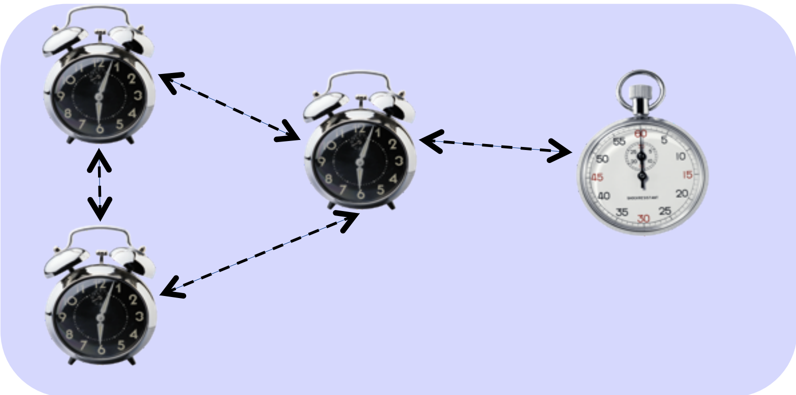


Four applications

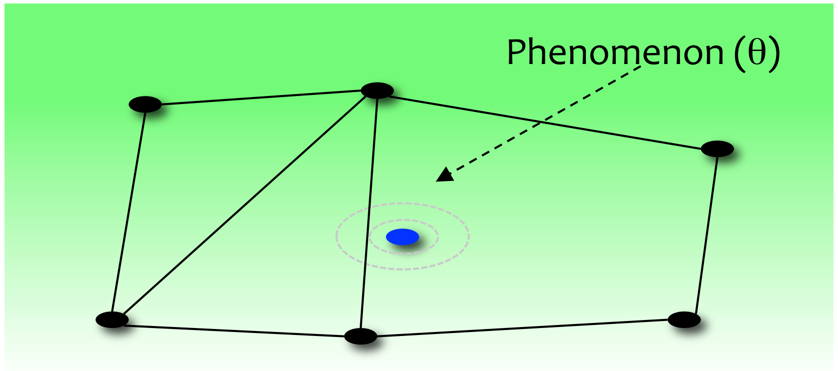
cooperative localization



distributed beamforming

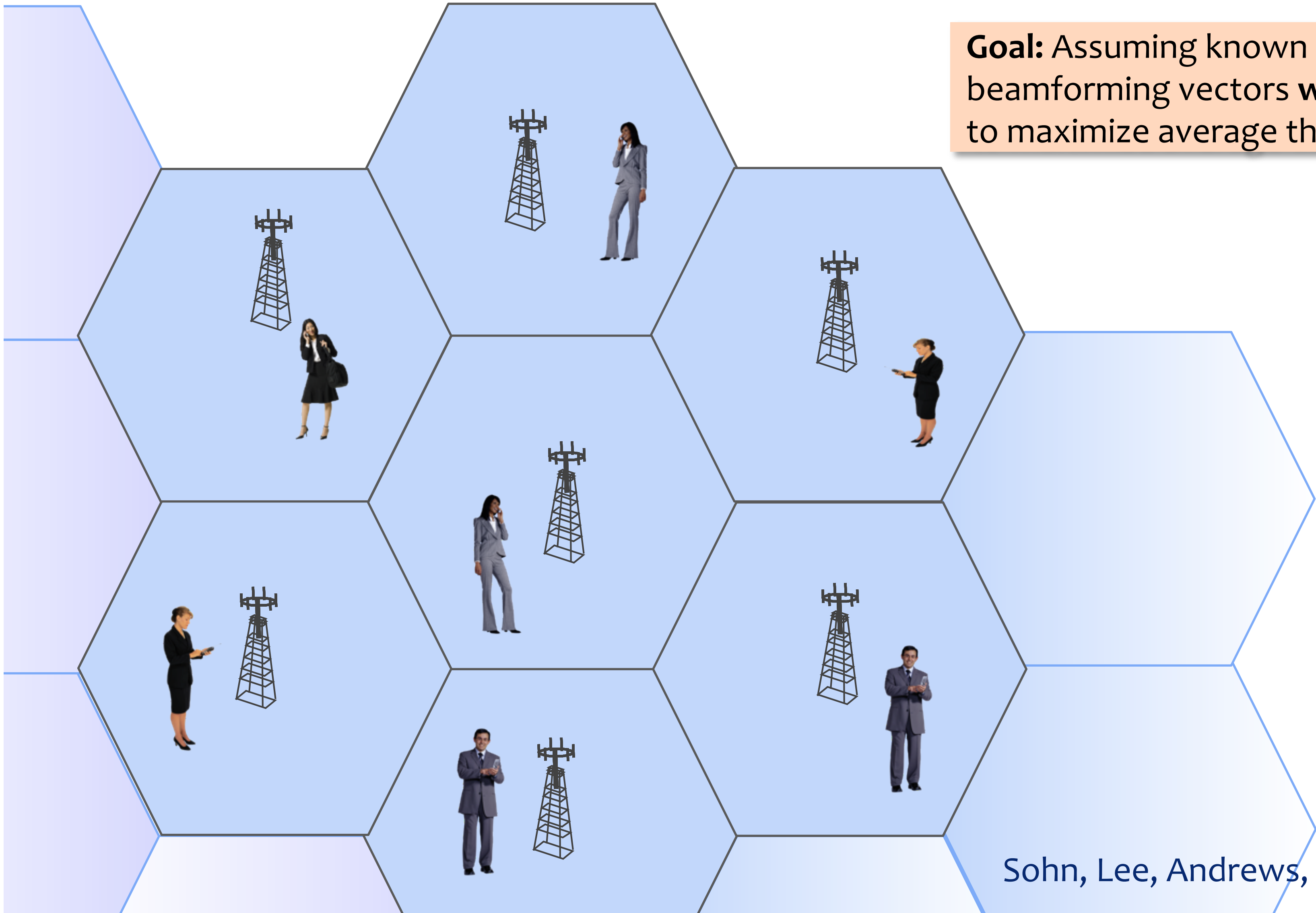


cooperative synchronization



distributed estimation

Distributed beamforming



Goal: Assuming known CSI, design beamforming vectors $\mathbf{w}_j$ for every BS to maximize average throughput.

Sohn, Lee, Andrews, TCOM, 2011

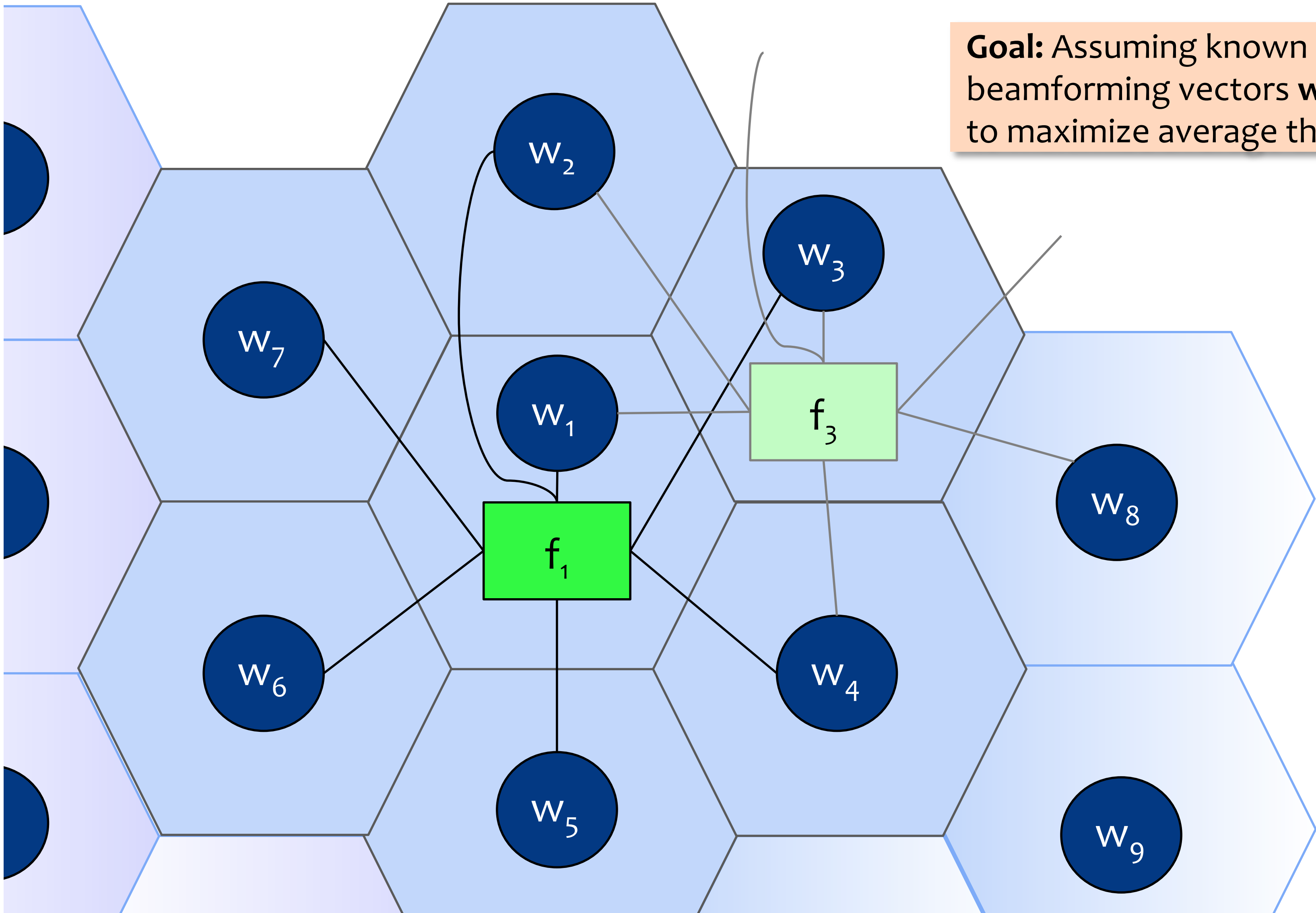
Distributed beamforming

- Maximize $$R(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N) = \frac{1}{N}\sum_{j=1}^{N} \mathrm{SINR}_j(\mathbf{w}_j, \mathbf{w}_{k\in\mathcal{N}_j})$$ $$\doteq \sum_{j=1}^{N} \psi_j(\mathbf{w}_j, \mathbf{w}_{k\in\mathcal{N}_j})$$
- Equivalent: for any β>0, maximize

$$\exp(\beta R(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N)) = \prod_{j=1}^{N} \exp(\beta\psi_j(\mathbf{w}_j, \mathbf{w}_{k\in\mathcal{N}_j}))$$
$$\doteq \prod_{j=1}^{N} f_j(\mathbf{w}_j, \mathbf{w}_{k\in\mathcal{N}_j})$$

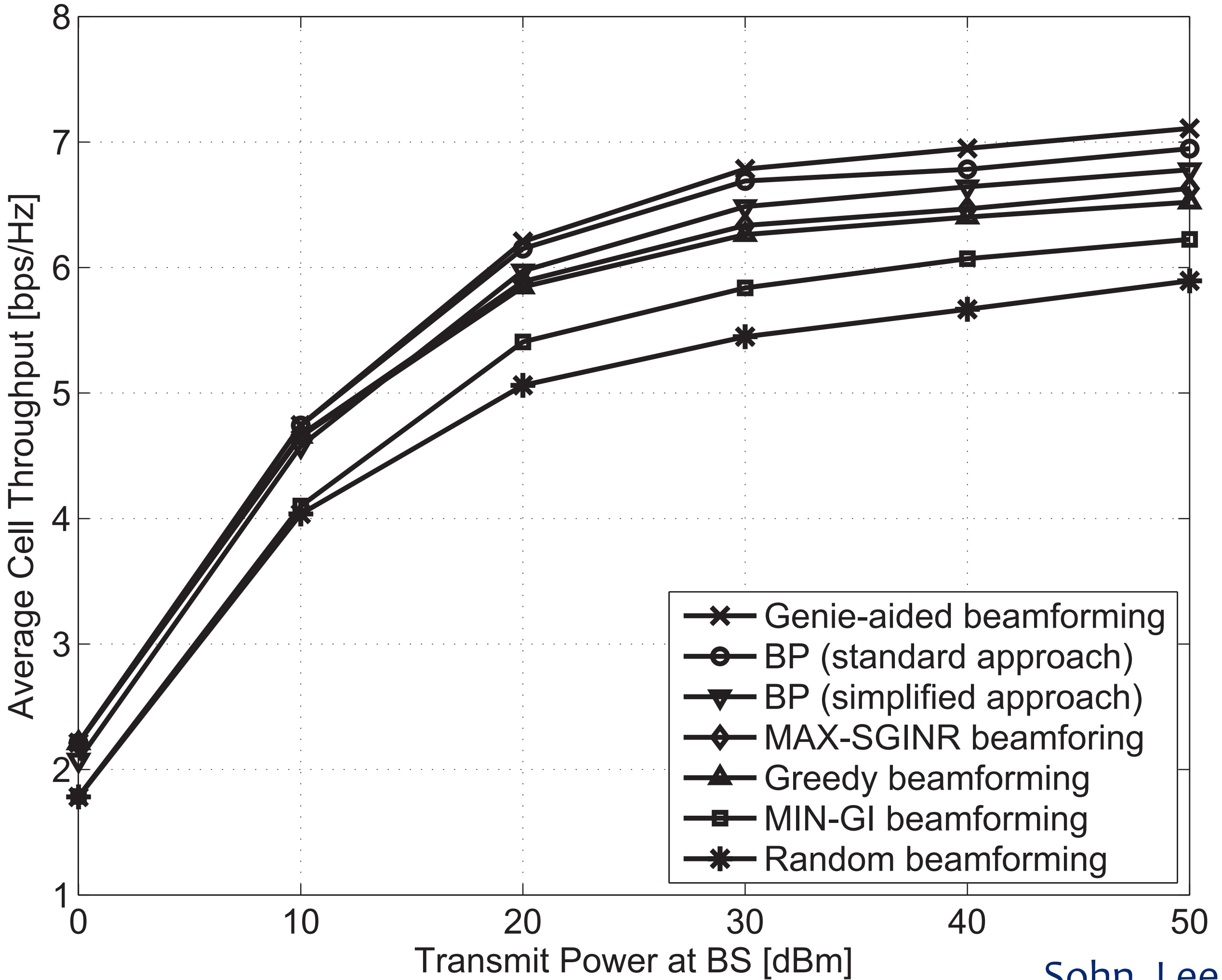
- Comment
  - Large β: more peaky around max.; chance of oscillations/misconvergence
  - small β: more flat, slower convergence

Distributed beamforming



Goal: Assuming known CSI, design beamforming vectors $\mathbf{w}_j$ for every BS to maximize average throughput.

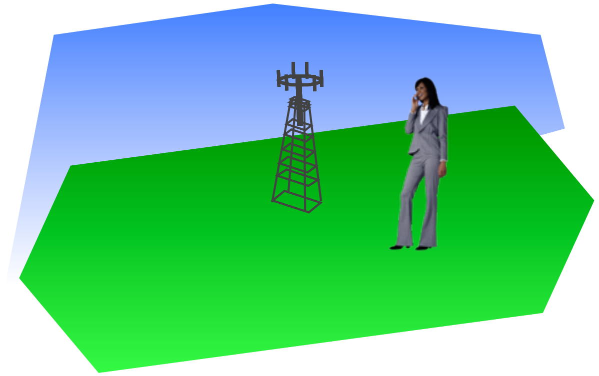
Performance example



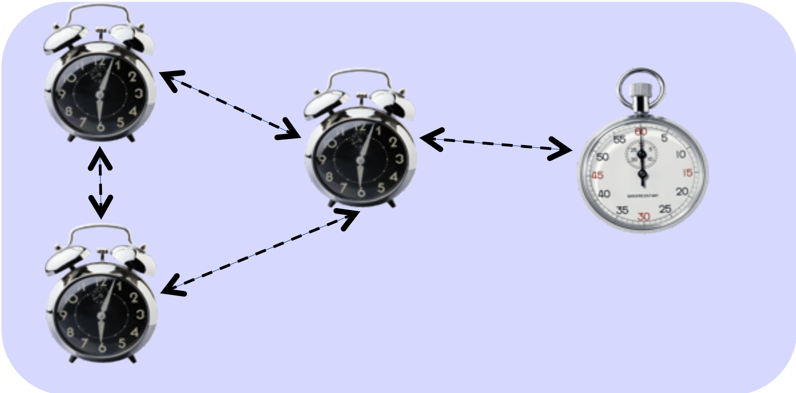
Sohn, Lee, Andrews, TCOM, 2011

Four applications

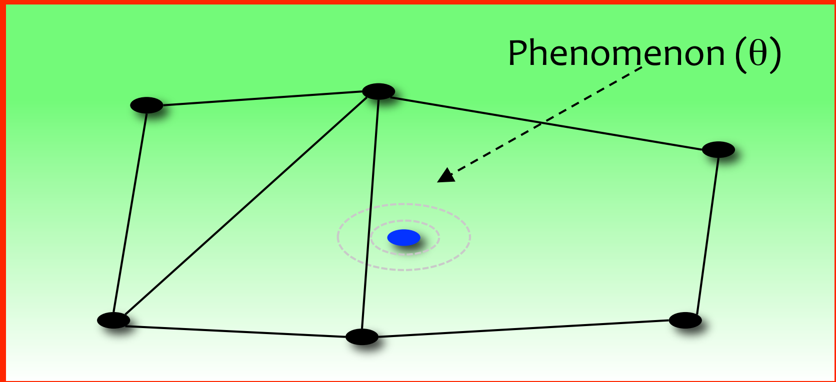
cooperative localization



distributed beamforming

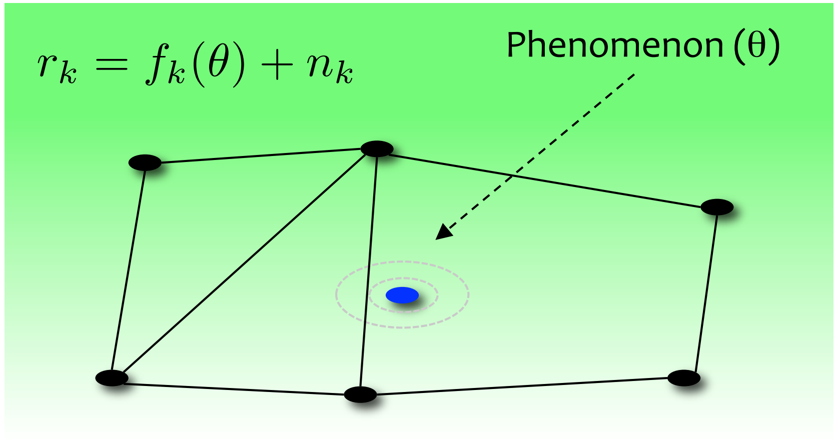


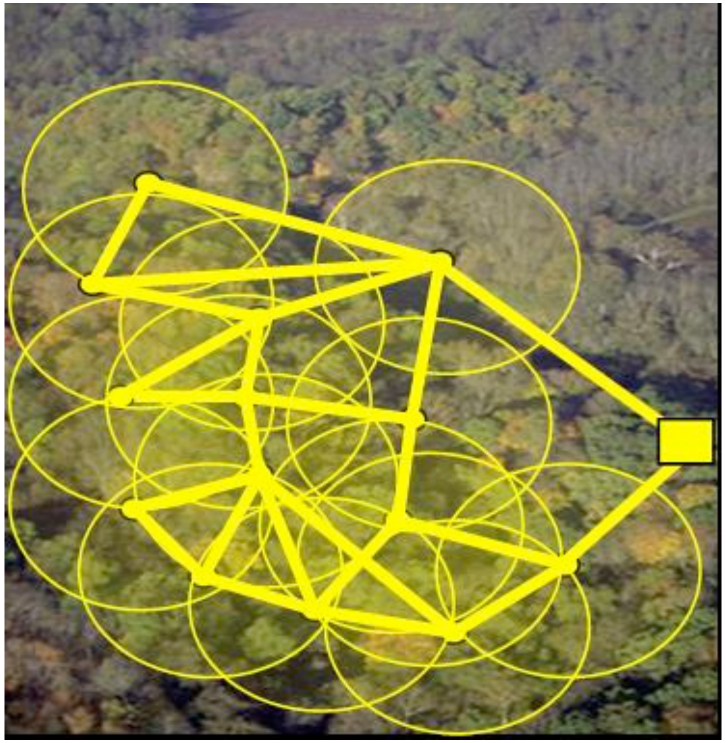
cooperative synchronization



distributed estimation

Wireless sensor network

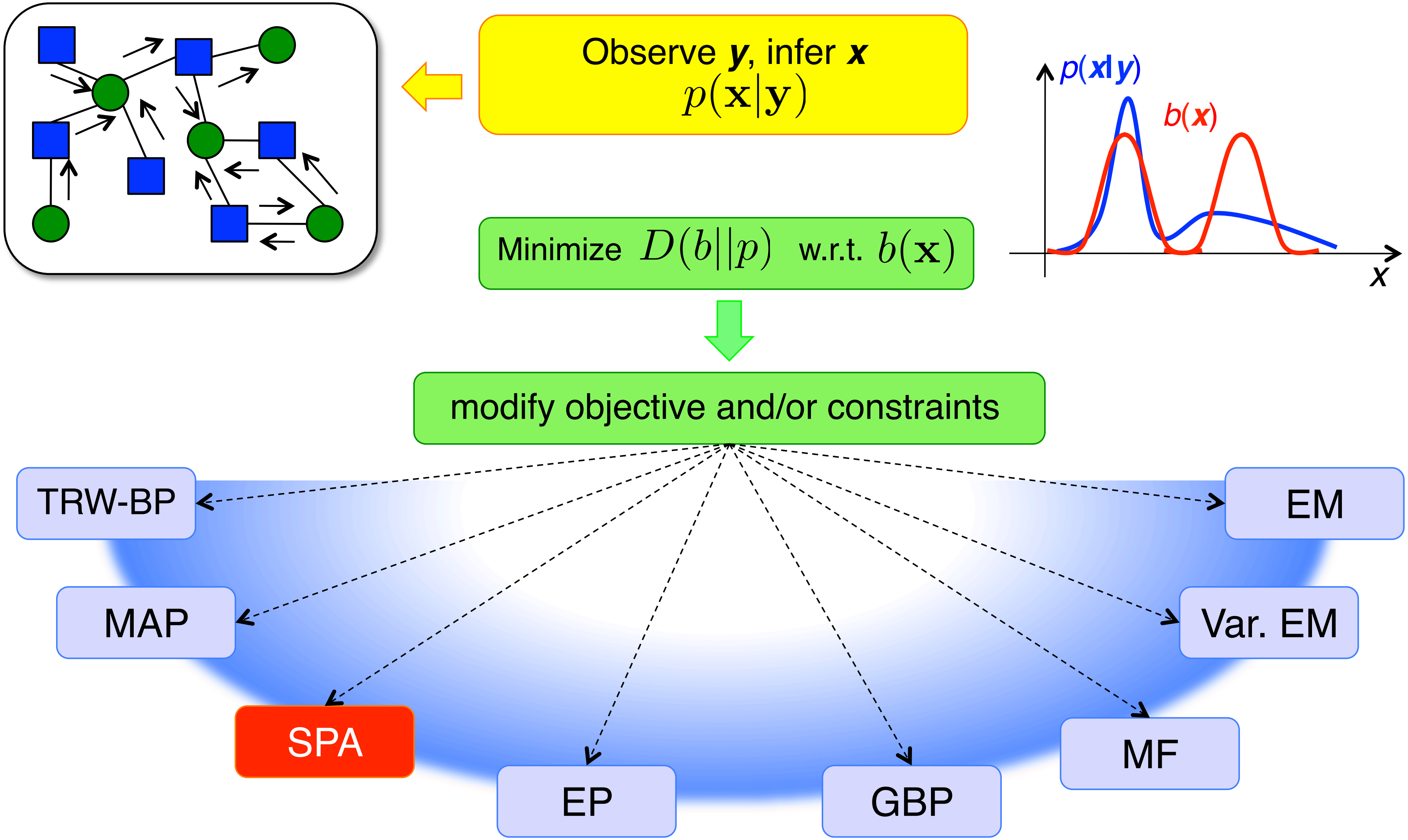




Goal:
Jointly compute estimate of θ

Outline

- Applications
- Background and terminology
- Bayesian detection
- Tool 1: Bayesian graphical models
  - Basics
  - Recipe for Bayesian inference
  - Practicalities
  - A worked example: a digital receiver
- Tool 2: Belief consensus
  - Basics
  - Convergence
- Applications revisited
- Variational interpretation
- Summary and conclusions


Observe *y*, infer *x*
$p(\mathbf{x}|\mathbf{y})$
$p(\boldsymbol{x}|\boldsymbol{y})$
$b(\boldsymbol{x})$
x
Minimize $D(b||p)$ w.r.t. $b(\mathbf{x})$
modify objective and/or constraints
TRW-BP
MAP
SPA
EP
GBP
MF
Var. EM
EM

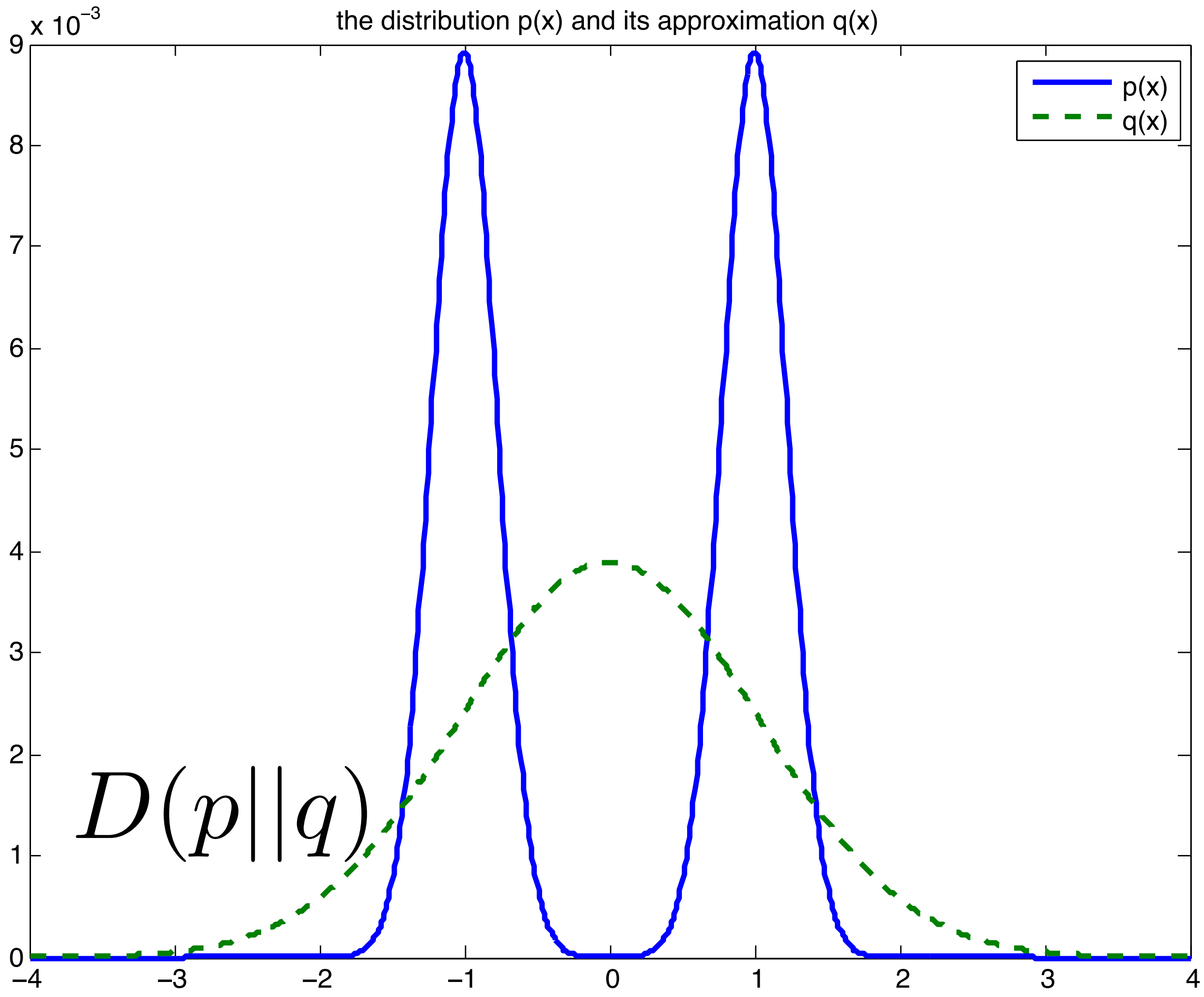
Kullback Leibler divergence (KLD)

- Measure of "distance" between distributions

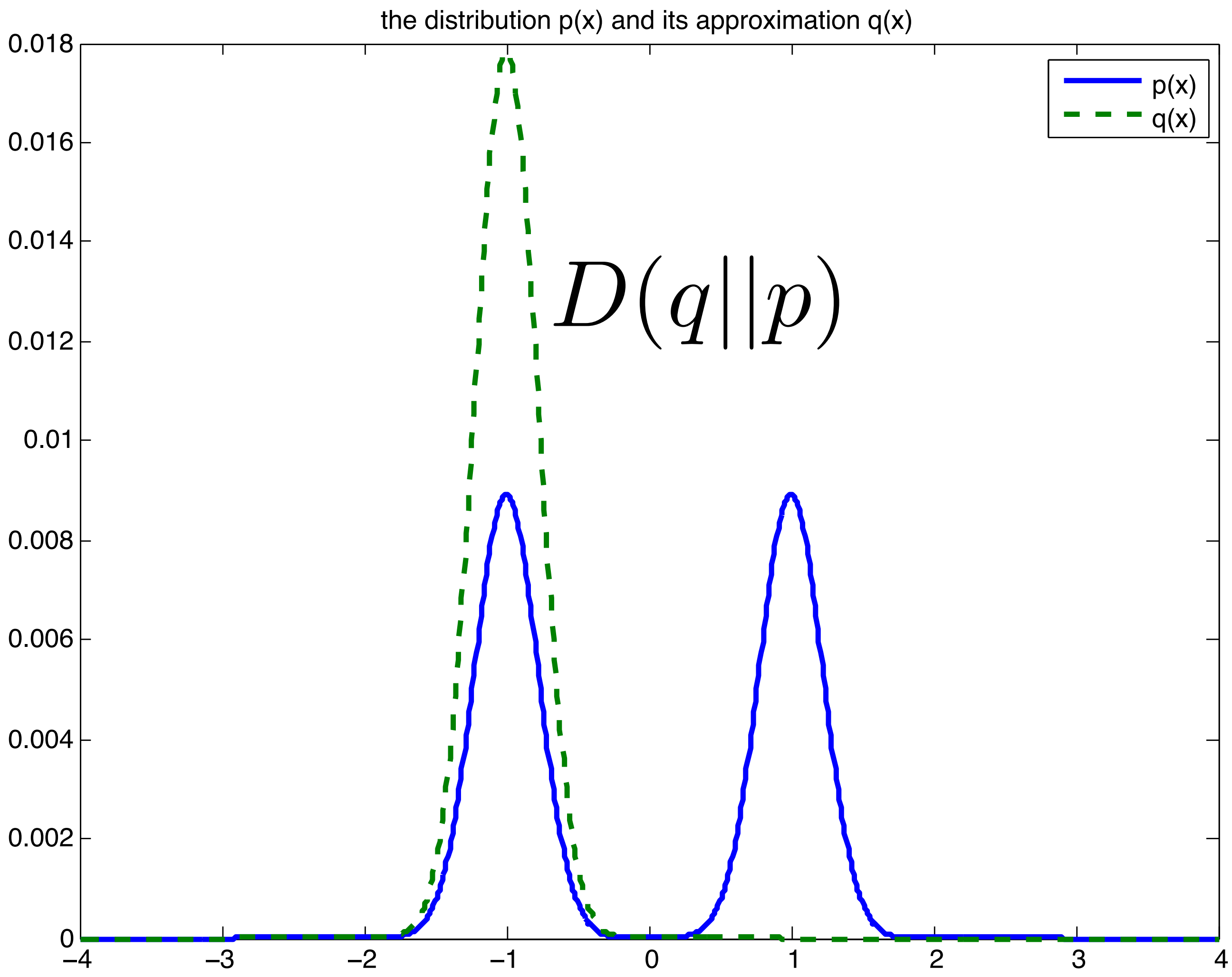
$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \geq 0$$
$$= \sum_x p(x) \log p(x) - \sum_x p(x) \log q(x)$$

- Convex in (p, q)
- Generally not known in closed form, except for Gaussian
- D(p||q) behaves different from D(q||p)
- minimize D(p||q) w.r.t. q
  - Does not like to be small when p is large
  - find q that matches moments
- minimize D(q||p) w.r.t. q
  - Locks on to one of the modes

Example: which KLD did we minimize?



Example: which KLD did we minimize?



Revisiting the likelihood

- Recall the important problems
  - MAP configuration
    $$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y})$$
  - Marginal posteriors
    $$p(x_i|\mathbf{y}) = \sum_{\sim x_i} p(\mathbf{x}|\mathbf{y})$$
  - Probability of y: $p(\mathbf{y})$

- We start from expression for p(**x**,**y**)

KLD and the likelihood function

- We start from $p(\mathbf{x},\mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = \prod_k f_k(\mathbf{x}_k)$
- And $p(\mathbf{x}|\mathbf{y}) = p(\mathbf{x},\mathbf{y})/p(\mathbf{y})$
- We are looking for a more “simple” q(x) to approximate p(x|y)

$$D(q||p) = \sum_{\mathbf{x}} q(\mathbf{x}) \log q(\mathbf{x}) - \sum_{\mathbf{x}} q(\mathbf{x}) \log p(\mathbf{x}|\mathbf{y}) \geq 0$$

$$= -\mathcal{H}(q) - \sum_{\mathbf{x}} q(\mathbf{x}) \log \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{y})}$$

$$= -\mathcal{H}(q) + \log p(\mathbf{y}) - \sum_{\mathbf{x}} q(\mathbf{x}) \sum_k \log f_k(\mathbf{x}_)$$

$$= -\mathcal{H}(q) + \log p(\mathbf{y}) - \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$

- Some re-arranging gives

$$\log p(\mathbf{y}) \geq \mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$

$$= \max_q \mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$

Example

- $p(x,y)=f(x_1,x_2)g(x_3)$
  - How should q(x) look like?
  - Evaluate
    $$\mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$
  - What can you say about the complexity?

- $p(x,y)=f(x_1)g(x_2)h(x_3)$
  - How should q(x) look like?
  - Evaluate
    $$\mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$
  - What can you say about the complexity?

Upper bounds the log-likelihood

- Solve the following convex problem

$$\log p(\mathbf{y}) = \max_{q \in \mathcal{F}_p} \mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$

- Note that
  - entropy is concave
  - cross-term is linear
  - family to search over is related to structure of p(x|y)

Solving optimization problems

- Problem formulation

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \ldots, m \\ & h_i(x) = 0 & i = 1, \ldots, p \end{array}$$

- Lagrangian

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

- Stationary points: define set of equations

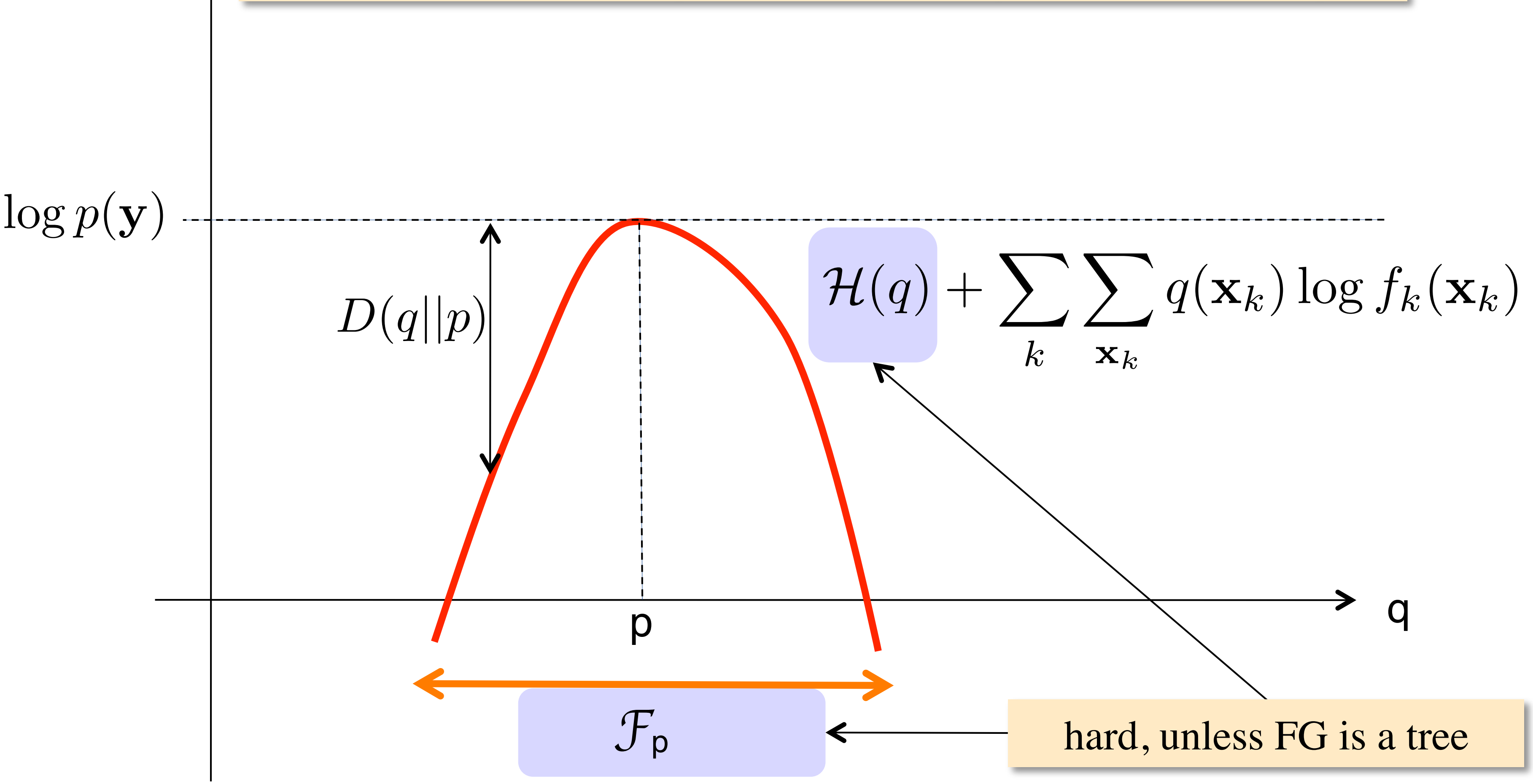
$$\frac{\partial L(x, \lambda, \nu)}{\partial x} = 0; \frac{\partial L(x, \lambda, \nu)}{\partial \lambda} = 0; \frac{\partial L(x, \lambda, \nu)}{\partial \nu} = 0$$

- For certain (convex) problems, stationary points correspond to global solution

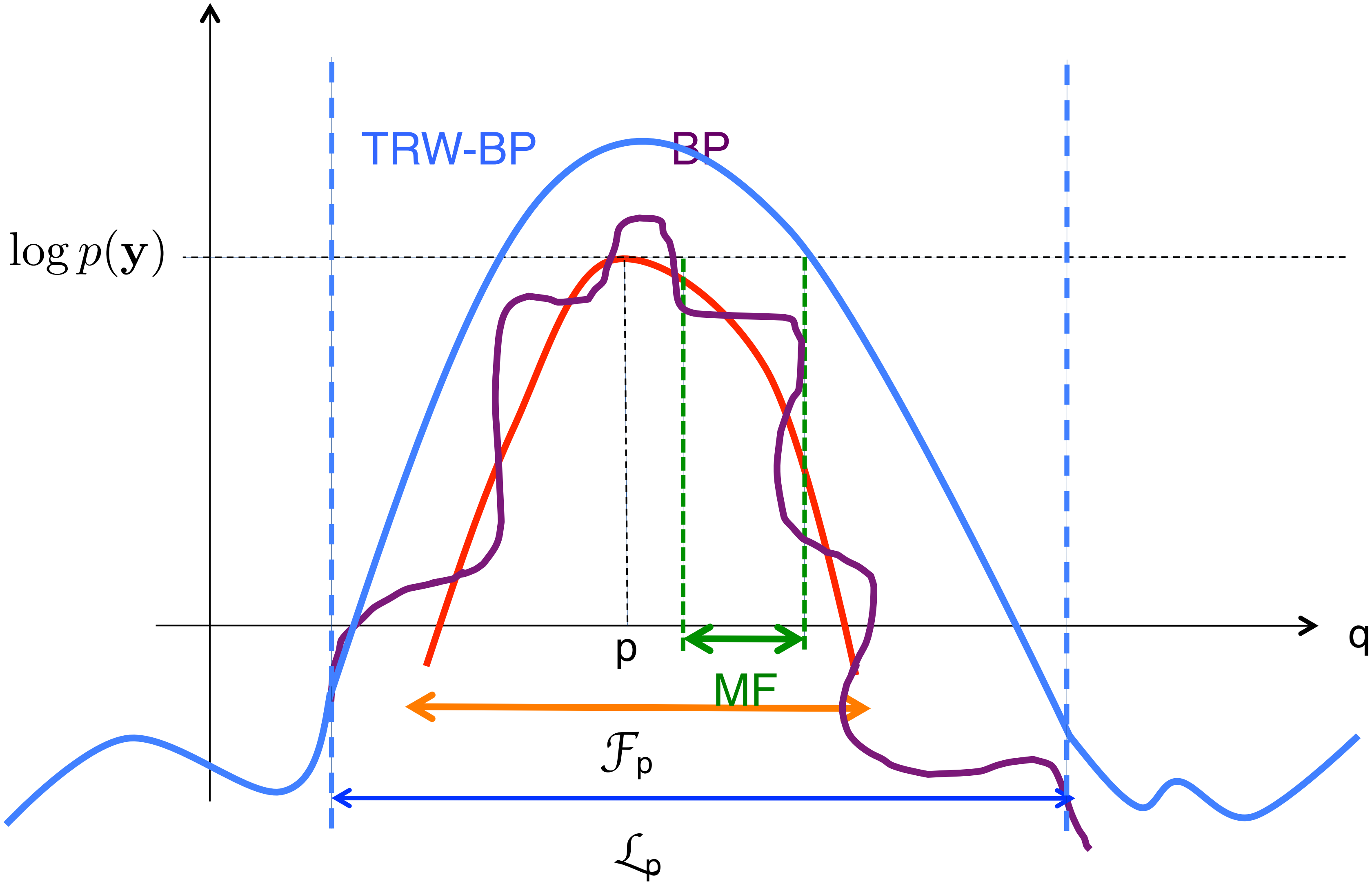
Graphical interpretation

~~“Once a probler is formulated as a co optimization problem, it is relatively straight e it.”~~

~~*Convex Optimization – S. oyd & L. Vandenberghe*~~



Approximations to the variational problem



Special case: when FG of p(x,y) is a tree

- Suppose $p(\mathbf{x},\mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = \prod_k f_k(\mathbf{x}_k)$ with $\mathbf{x}_k$ containing at least two variables (using grouping)
- For trees (d_n is the degree of variable node X_n) it can be shown that

$$p(\mathbf{x}|\mathbf{y}) = \frac{\prod_k p(\mathbf{x}_k|\mathbf{y})}{\prod_{n=1}^{N} p(x_n|\mathbf{y})^{d_n - 1}}$$

- So we can limit ourselves to

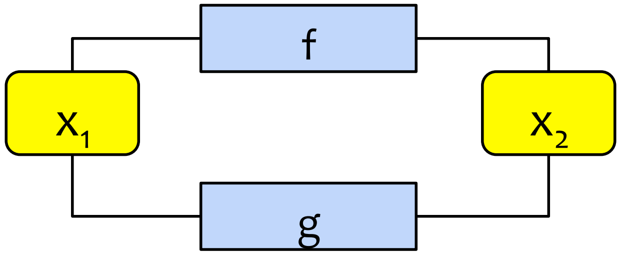
$$q(\mathbf{x}) = \frac{\prod_k q(\mathbf{x}_k)}{\prod_{n=1}^{N} q(x_n)^{d_n - 1}}$$

- Why does the problem become easy?

$$\log p(\mathbf{y}) = \max_{q \in \mathcal{F}_p} \mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$

A simple example when you don’t have a tree

- Suppose $p(x_1,x_2)=f(x_1,x_2)g(x_1,x_2)$ with factor graph



- Then we can try to write

$$p(x_1, x_2) = \frac{p(x_1, x_2)p(x_1, x_2)}{p(x_1)p(x_2)} = p(x_1|x_2)p(x_2|x_1)$$

- It is easy to find conditional distributions so that the right hand side is not normalized

Solving the optimization problem explicitly

Again $\log p(\mathbf{y}) = \max_{q \in \mathcal{F}_p} \mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$

Approach*

1. Express constraint set
2. Express entropy
3. Express Lagrangian
4. Set derivatives w.r.t. variables to zero

Jonathan Yedidia, 2005

Surprising result

- stationary points correspond to SPA rules
- provides rigorous interpretation of SPA
- even when we don't have a tree, when we approximate $\mathcal{F}_p$ using functions (not distributions!) of the form

$$q(\mathbf{x}) = \frac{\prod_k q(\mathbf{x}_k)}{\prod_{n=1}^{N} q(x_n)^{d_n - 1}}$$

Related results

- MAP configuration: same problem without entropy
- MPA not equivalent to solving variational problem

*under some technical conditions

Simplified case

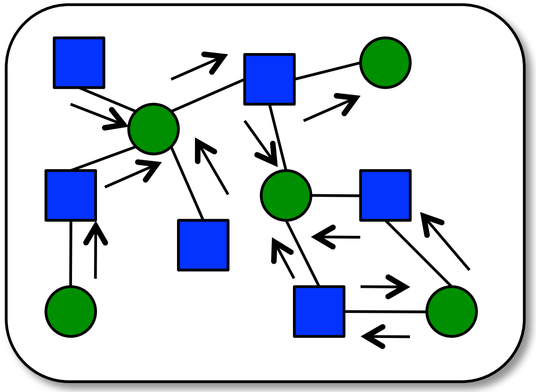
- Suppose p(x|y) factorizes completely $p(\mathbf{x},\mathbf{y}) = \prod_n f_n(x_n)$
- Approximation will be $q(\mathbf{x}) = \prod_n q_n(x_n)$
- Then

$$\mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k) = \sum_n \sum_{x_n} q_n(x_n) \left(\log f_n(x_n) - \log q_n(x_n)\right)$$

- Leads to the following problem

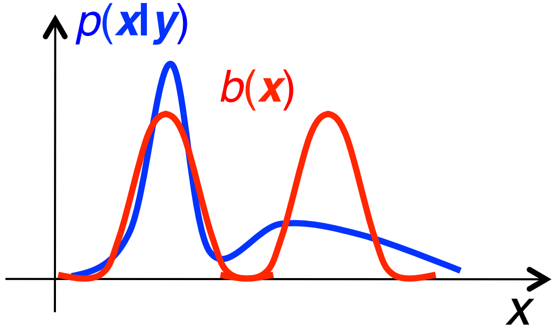
$$\begin{aligned} \text{maximize} \quad & \sum_n \sum_{x_n} q_n(x_n) \left(\log f_n(x_n) - \log q_n(x_n)\right) \\ \text{s.t.} \quad & q_n(x_n) \geq 0, \forall n, x_n \\ & \sum_{x_n} q_n(x_n) = 1, \forall n \end{aligned}$$

- Take the derivative of the objective w.r.t. $q_m(x_m)$



Observe ***y***, infer ***x***

$$p(\mathbf{x}|\mathbf{y})$$



Minimize $D(b||p)$ w.r.t. $b(\mathbf{x})$

modify objective and/or constraints

TRW-BP

EM

MAP

Var. EM

SPA

MF

EP

GBP

Outline

- Applications
- Background and terminology
- Bayesian detection
- Tool 1: Bayesian graphical models
  - Basics
  - Recipe for Bayesian inference
  - Practicalities
  - A worked example: a digital receiver
- Tool 2: Belief consensus
  - Basics
  - Convergence
- Applications revisited
- Variational interpretation
- Summary and conclusions

Goals - revisited

At the end of this tutorial, you should be able to:

1. *Recognize and formulate* inference problems ✔
2. *Solve* inference problems using factor graphs ✔
3. *Design* algorithms for centralized and distributed processing ✔
4. *Describe* the bigger picture ✔

Conclusions

- Factor graphs: powerful tool to solve problems related to
  - Estimation
  - Detection
  - Optimization
  - Distributed processing
- Not a silver bullet, but often a good start
- Many open problems, many potential applications

Thank You

More information:
tinyurl.com/hwymeers
www.youtube.com/hwymeers

Answer Keys

- The following slides contains solutions to the problems in the tutorial

STOP

Example

- Consider the following distribution over binary variables

$$p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} p(x_i) \times \prod_{i=2}^{4} p(\mathbf{y}_i | x_1, x_i)$$

- Solve the three problems

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}, \mathbf{y})$$

$$\hat{x}_i = \arg\max_{x_i} \underbrace{\max_{\sim x_i} p(\mathbf{x}, \mathbf{y})}_{g(x_i)}$$

$$p(x_i, \mathbf{y}) = \sum_{\sim x_i} p(\mathbf{x}, \mathbf{y})$$

Example

- Consider the following distribution over binary variables

$$p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} p(x_i) \times \prod_{i=2}^{4} p(\mathbf{y}_i | x_1, x_i)$$

- Solve

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}, \mathbf{y})$$

$$\hat{\mathbf{x}} = \arg \max_{x_1, x_2, x_3, x_4} p(x_1)p(x_2)p(x_3)p(x_4)p(\mathbf{y}_2|x_1, x_2)p(\mathbf{y}_3|x_1, x_3)p(\mathbf{y}_4|x_1, x_4)$$

- Complexity $2^{\text{number of variables}}$

Complexity
- try all 16 combinations
- General: 2^N function evaluations

Example

- Consider the following distribution over binary variables

$$p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} p(x_i) \times \prod_{i=2}^{4} p(\mathbf{y}_i | x_1, x_i)$$

- Solve

$$\hat{x}_i = \arg\max_{x_i} \underbrace{\max_{\sim x_i} p(\mathbf{x}, \mathbf{y})}_{g(x_i)}$$

Complexity
- 3+3 O(2) problems
- General: 2N small problems

$$g_1(x_1) = p(x_1) \left\{ \max_{x_2} p(x_2) p(\mathbf{y}_2 | x_1, x_2) \right\} \left\{ \max_{x_3} p(x_3) p(\mathbf{y}_3 | x_1, x_3) \right\} \left\{ \max_{x_4} p(x_4) p(\mathbf{y}_4 | x_1, x_4) \right\}$$

$$g_2(x_2) = p(x_2) \left\{ \max_{x_1} p(x_1) p(\mathbf{y}_2 | x_1, x_2) \left\{ \max_{x_3} p(x_3) p(\mathbf{y}_3 | x_1, x_3) \right\} \left\{ \max_{x_4} p(x_4) p(\mathbf{y}_4 | x_1, x_4) \right\} \right\}$$

$$g_3(x_3) = p(x_3) \left\{ \max_{x_1} p(x_1) p(\mathbf{y}_3 | x_1, x_3) \left\{ \max_{x_2} p(x_2) p(\mathbf{y}_2 | x_1, x_2) \right\} \left\{ \max_{x_4} p(x_4) p(\mathbf{y}_4 | x_1, x_4) \right\} \right\}$$

$$g_4(x_4) = p(x_4) \left\{ \max_{x_1} p(x_1) p(\mathbf{y}_4 | x_1, x_4) \left\{ \max_{x_2} p(x_2) p(\mathbf{y}_2 | x_1, x_2) \right\} \left\{ \max_{x_3} p(x_3) p(\mathbf{y}_3 | x_1, x_3) \right\} \right\}$$

Example

- Consider the following distribution over binary variables

$$p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} p(x_i) \times \prod_{i=2}^{4} p(\mathbf{y}_i | x_1, x_i)$$

- Solve the three problems

$$p(x_i, \mathbf{y}) = \sum_{\sim x_i} p(\mathbf{x}, \mathbf{y})$$

Complexity

- 3+3 O(2) problems
- General: 2N small problems

$$p(x_1, \mathbf{y}) = \quad p(x_1) \left\{ \sum_{x_2} p(x_2) p(\mathbf{y}_2 | x_1, x_2) \right\} \left\{ \sum_{x_3} p(x_3) p(\mathbf{y}_3 | x_1, x_3) \right\} \left\{ \sum_{x_4} p(x_4) p(\mathbf{y}_4 | x_1, x_4) \right\}$$

$$p(x_2, \mathbf{y}) = \quad p(x_2) \left\{ \sum_{x_1} p(x_1) p(\mathbf{y}_2 | x_1, x_2) \left\{ \sum_{x_3} p(x_3) p(\mathbf{y}_3 | x_1, x_3) \right\} \left\{ \sum_{x_4} p(x_4) p(\mathbf{y}_4 | x_1, x_4) \right\} \right\}$$

$$p(x_3, \mathbf{y}) = \quad p(x_3) \left\{ \sum_{x_1} p(x_1) p(\mathbf{y}_3 | x_1, x_3) \left\{ \sum_{x_2} p(x_2) p(\mathbf{y}_2 | x_1, x_2) \right\} \left\{ \sum_{x_4} p(x_4) p(\mathbf{y}_4 | x_1, x_4) \right\} \right\}$$

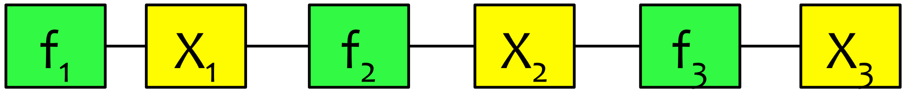
$$p(x_4, \mathbf{y}) = \quad p(x_4) \left\{ \sum_{x_1} p(x_1) p(\mathbf{y}_4 | x_1, x_4) \left\{ \sum_{x_2} p(x_2) p(\mathbf{y}_2 | x_1, x_2) \right\} \left\{ \sum_{x_3} p(x_3) p(\mathbf{y}_3 | x_1, x_3) \right\} \right\}$$

Example

1. Draw the FG of the following distribution

$$p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} p(x_i) \times \prod_{i=2}^{4} p(\mathbf{y}_i | x_1, x_i)$$

2. Write down *a* function corresponding to the following FG

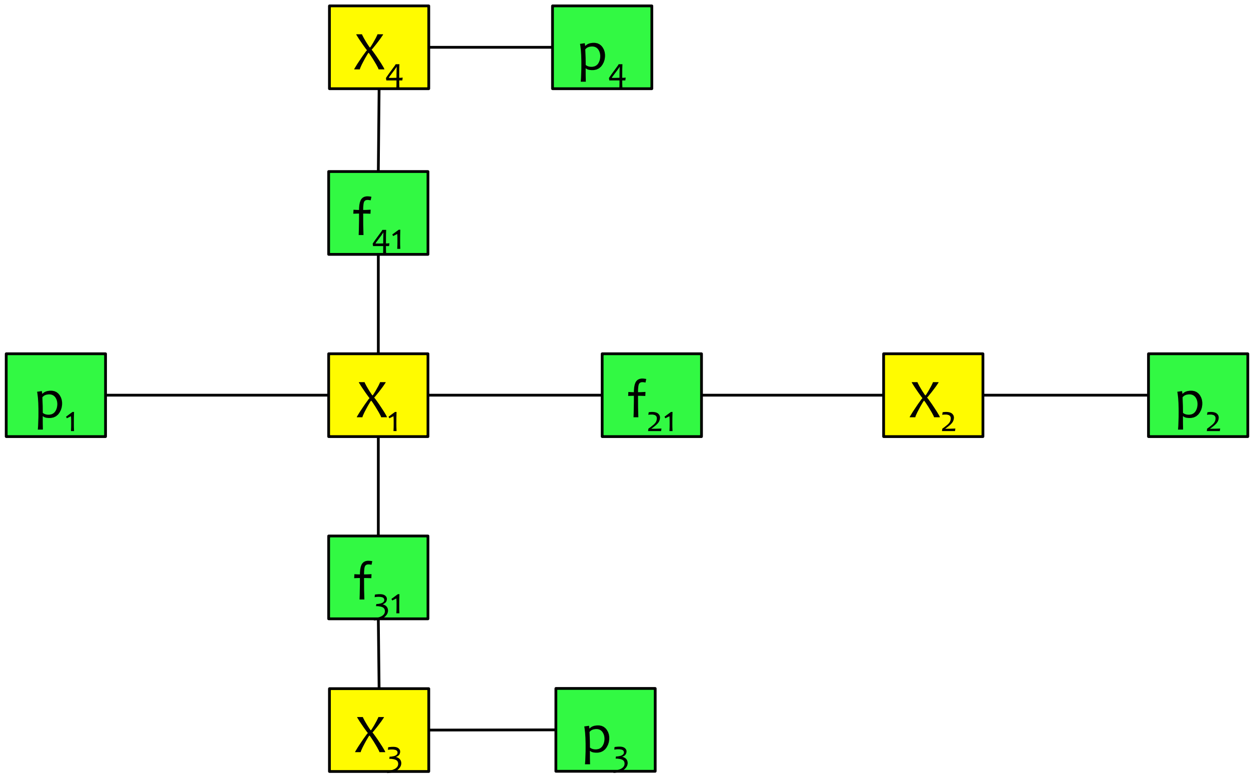


3. Can you come up with a reasonable distribution?

Example

1. Draw the FG of the following distribution

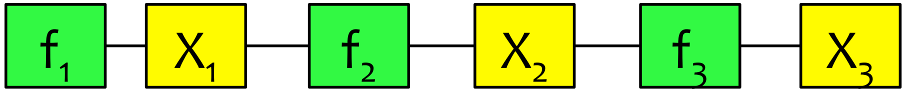
$$p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} p(x_i) \times \prod_{i=2}^{4} p(\mathbf{y}_i | x_1, x_i)$$



Example

Write down *a* function corresponding to the following FG

Can you come up with a reasonable distribution?



Solution

$f_1(X_1)f_2(X_1,X_2)f_3(X_2,X_3)$

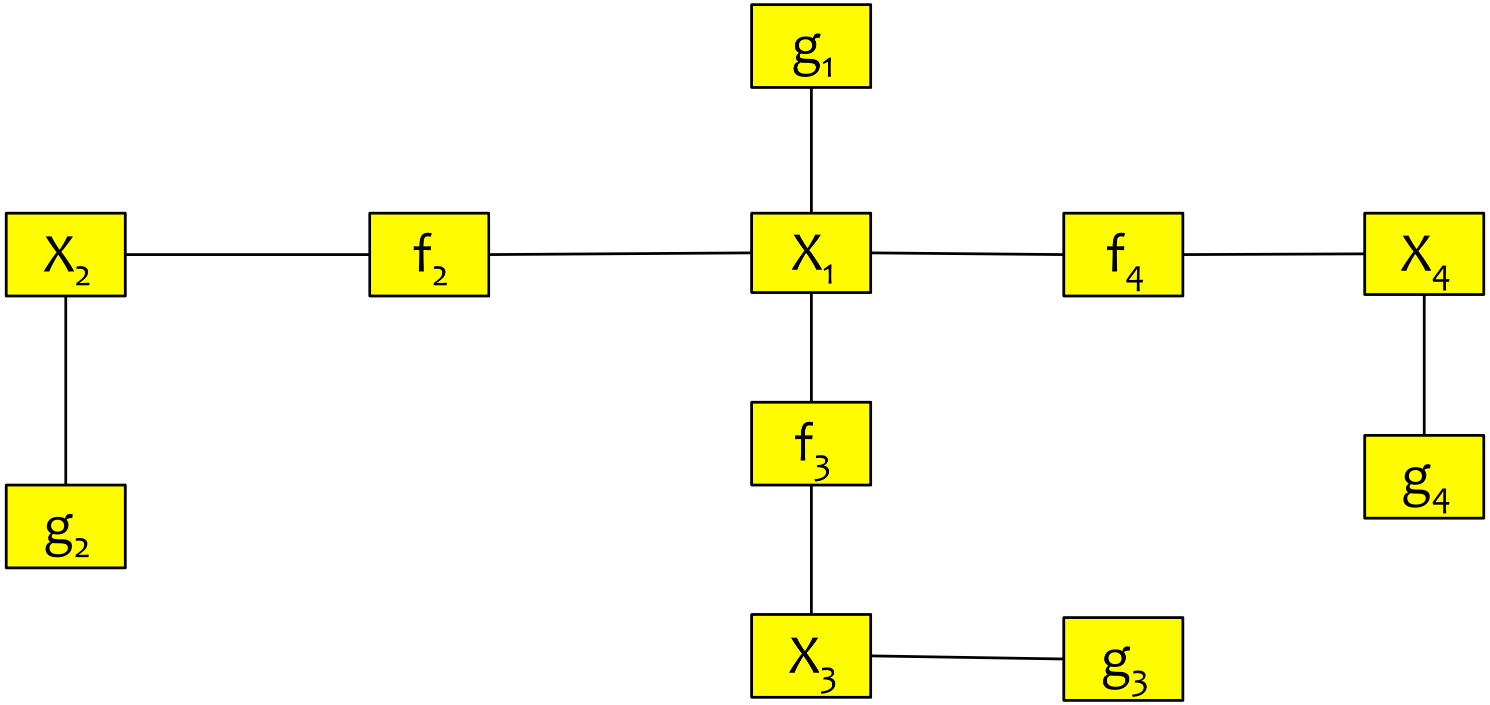
structure is given, but variables are not, nor are function details

distribution:

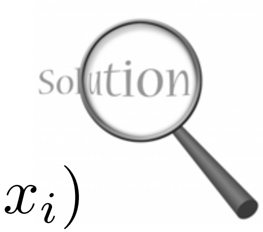
- $p(X_1)p(X_2|X_1)p(X_3|X_1)$
- $p(X_3)p(X_2|X_3)p(X_1|X_2)p(y|X_1)$, with uniform $p(X_3)$

Example

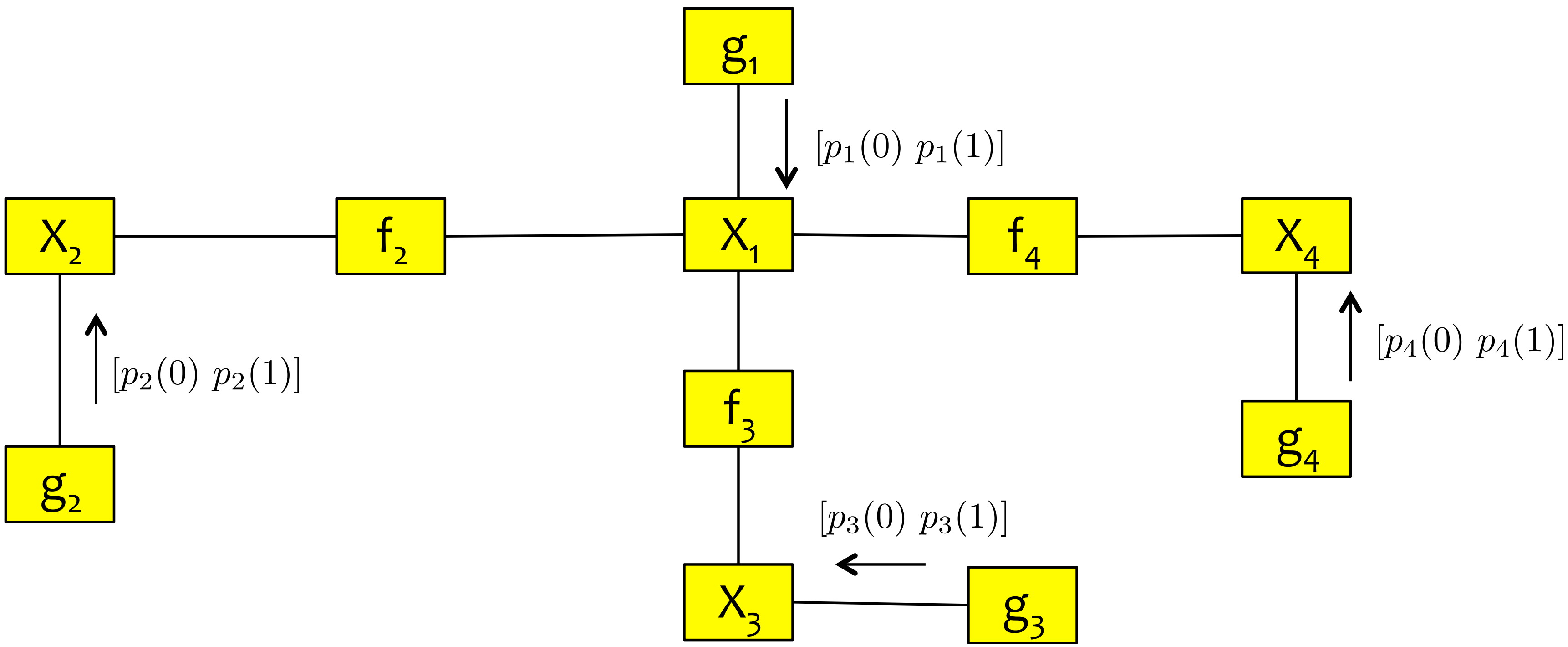
- Consider $p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} \underbrace{p(x_i)}_{g_i(x_i)} \times \prod_{i=2}^{4} \underbrace{p(\mathbf{y}_i | x_1, x_i)}_{f_i(x_1, x_i)}$
- Write down the sum-product rules
- Verify the solution



Example

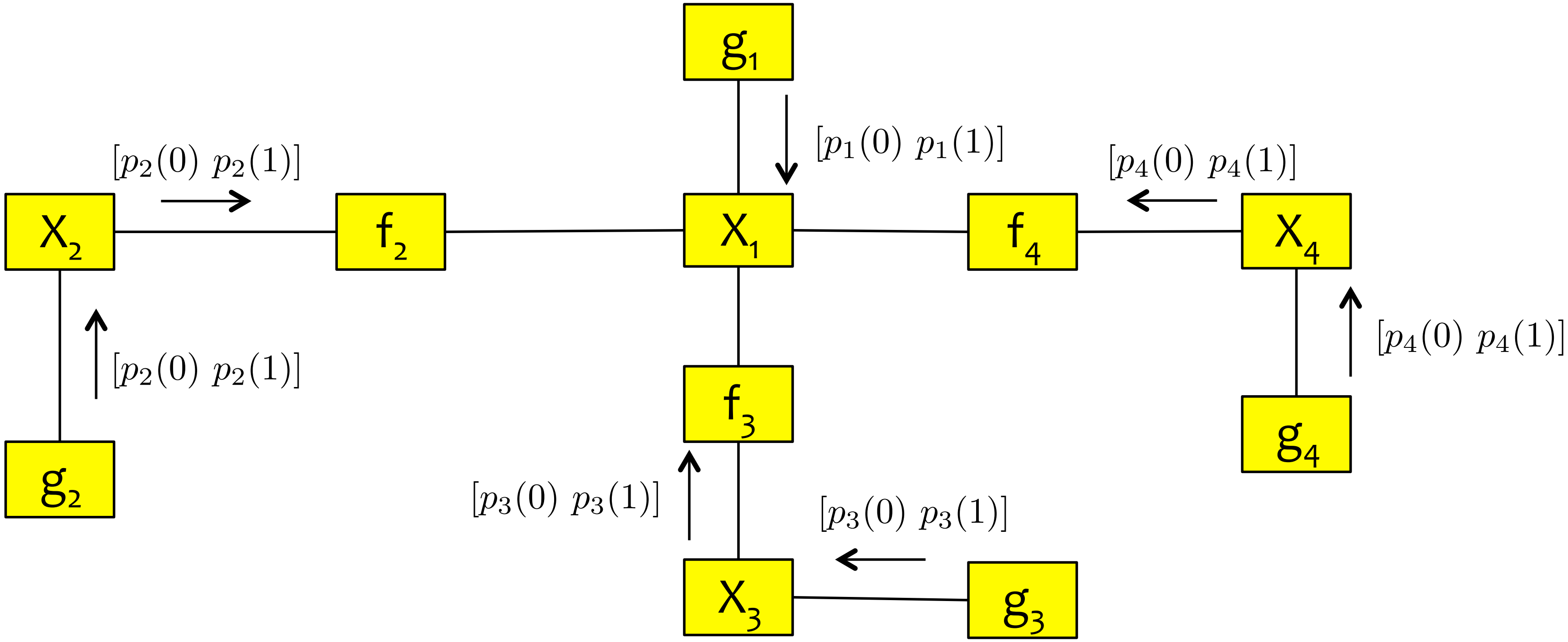


- Consider $p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} \underbrace{p(x_i)}_{g_i(x_i)} \times \prod_{i=2}^{4} \underbrace{p(\mathbf{y}_i | x_1, x_i)}_{f_i(x_1, x_i)}$



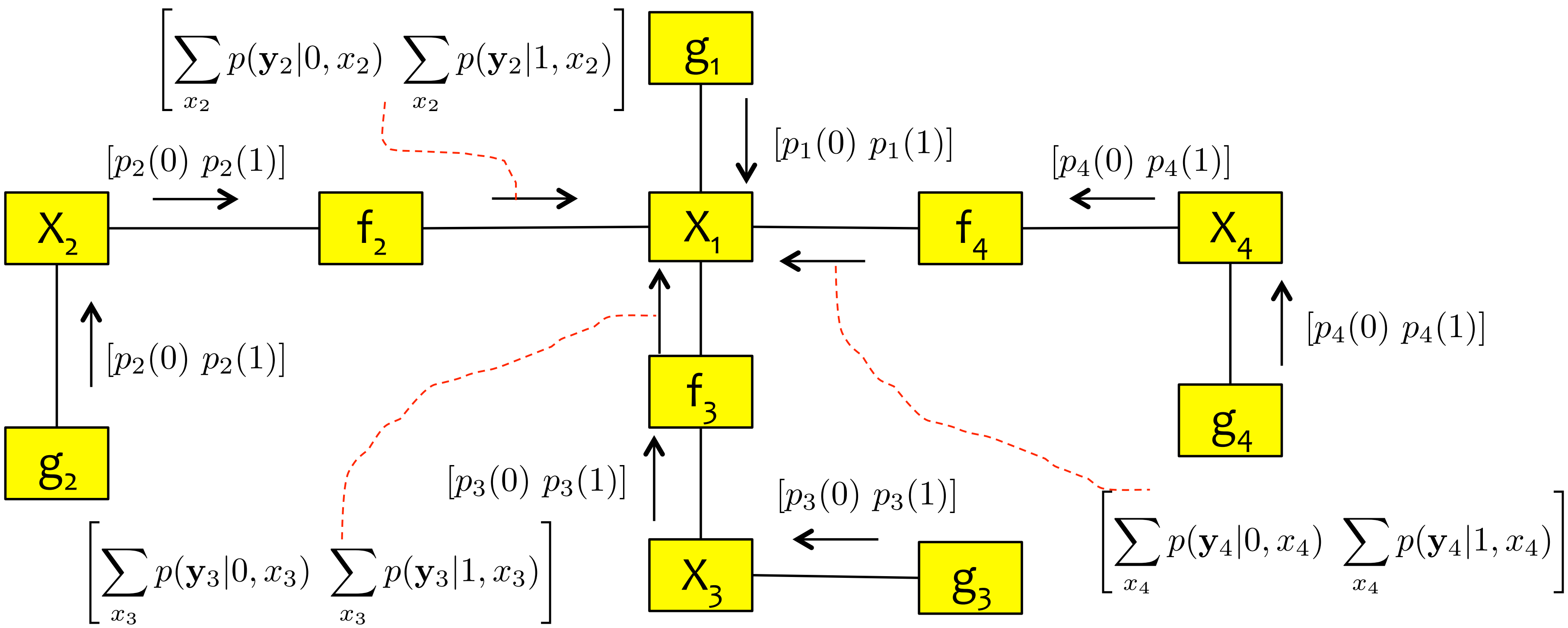
Example

- Consider $p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} \underbrace{p(x_i)}_{g_i(x_i)} \times \prod_{i=2}^{4} \underbrace{p(\mathbf{y}_i | x_1, x_i)}_{f_i(x_1, x_i)}$



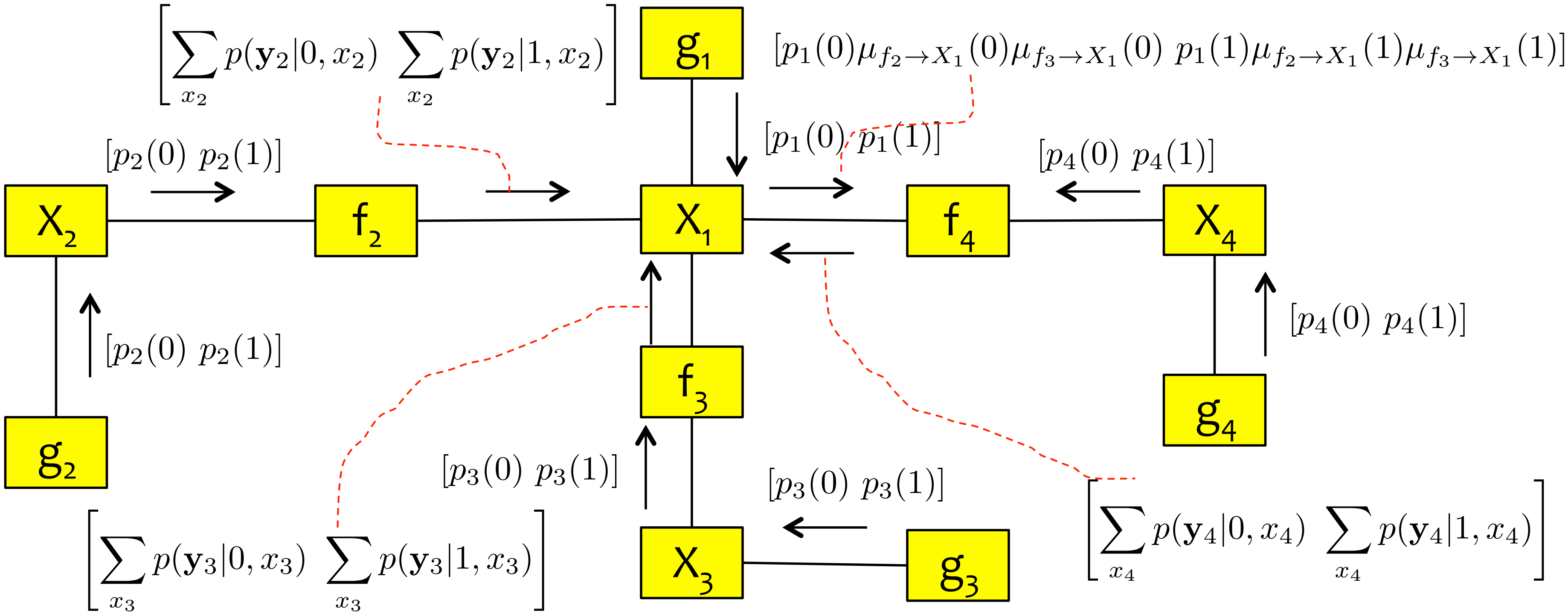
Example

- Consider $p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} \underbrace{p(x_i)}_{g_i(x_i)} \times \prod_{i=2}^{4} \underbrace{p(\mathbf{y}_i | x_1, x_i)}_{f_i(x_1, x_i)}$



Example

- Consider $p(x_1,x_2,x_3,x_4,\mathbf{y}) = \prod_{i=1}^{4} \underbrace{p(x_i)}_{g_i(x_i)} \times \prod_{i=2}^{4} \underbrace{p(\mathbf{y}_i|x_1,x_i)}_{f_i(x_1,x_i)}$

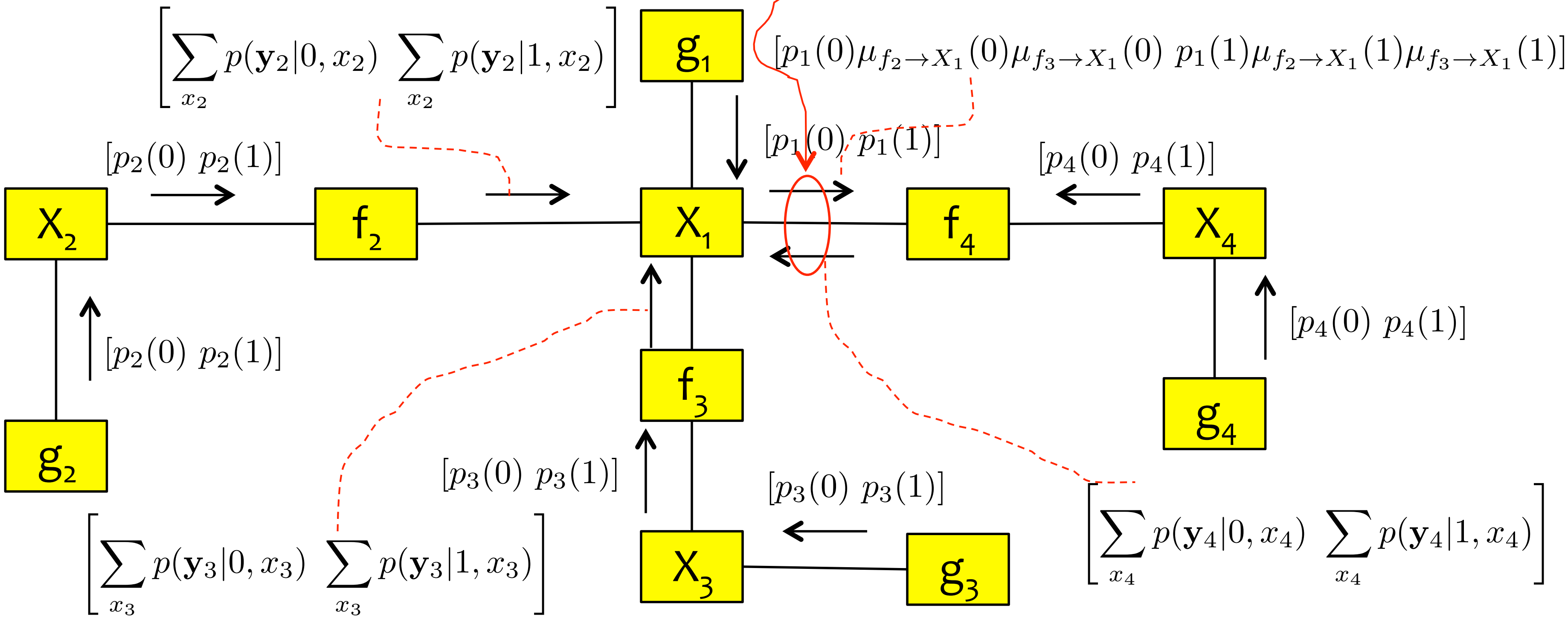


Example

- Consider $p(x_1, x_2, x_3, x_4, \mathbf{y}) = \prod_{i=1}^{4} p(x_i) \times \prod_{i=2}^{4} p(\mathbf{y}_i | x_1, x_i)$

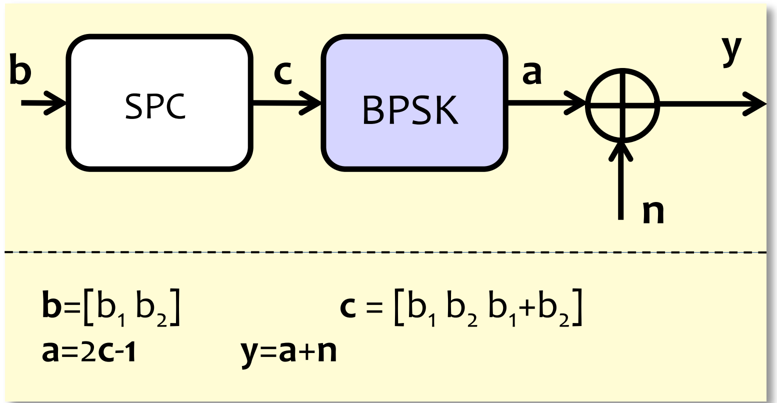
$g_i(x_i)$ $\quad$ $f_i(x_1, x_i)$

$$\boxed{\begin{aligned} p(X_1 = 0, \mathbf{y}) &= p_1(0)\mu_{f_2 \to X_1}(0)\mu_{f_3 \to X_1}(0) \sum_{x_4} p(\mathbf{y}_4 | 0, x_4) \\ p(X_1 = 1, \mathbf{y}) &= p_1(1)\mu_{f_2 \to X_1}(1)\mu_{f_3 \to X_1}(1) \sum_{x_4} p(\mathbf{y}_4 | 1, x_4) \end{aligned}}$$



Normal factor graph

- Transmitter

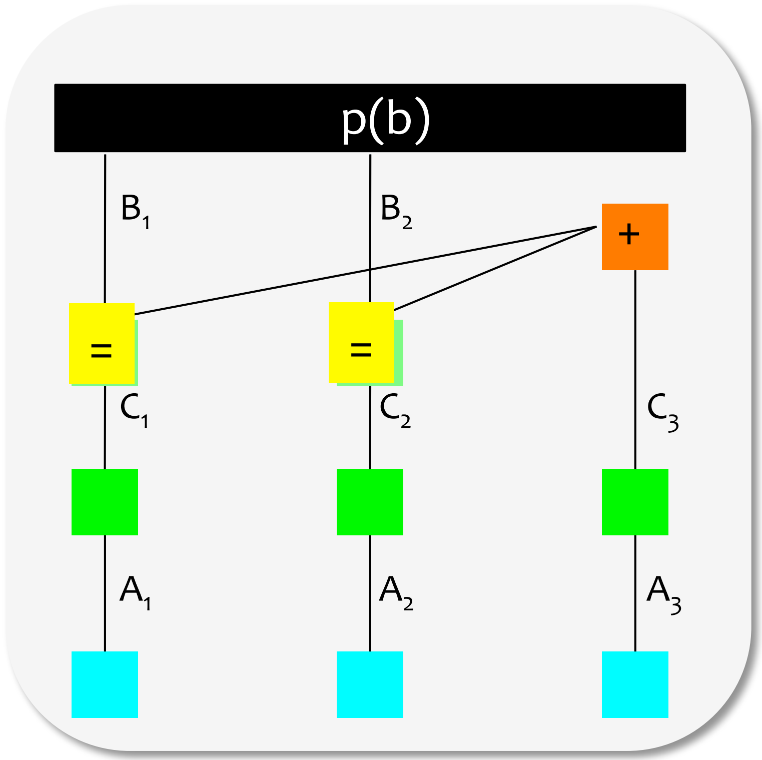


$$p(\mathbf{b},\mathbf{c},\mathbf{a},\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a})p(\mathbf{a}|\mathbf{c})p(\mathbf{c}|\mathbf{b})p(\mathbf{b})$$

$$p(\mathbf{y}|\mathbf{a}) = \prod_{k=1}^{3} p(y_k|a_k)$$

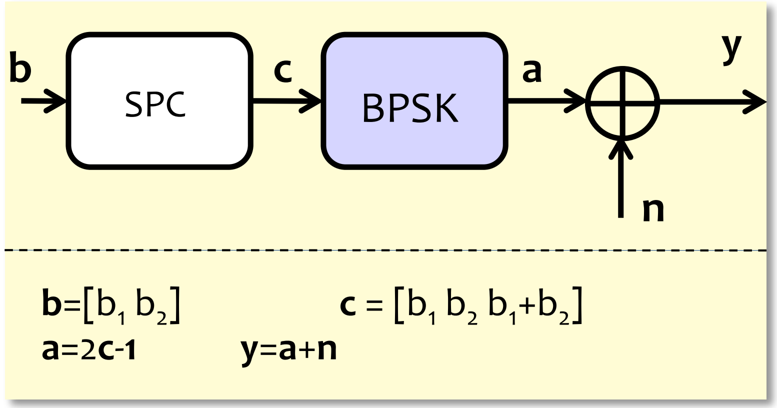
$$p(\mathbf{a}|\mathbf{c}) = \prod_{k=1}^{3} p(a_k|c_k)$$

$$p(\mathbf{c}|\mathbf{b}) = \prod_{k=1}^{2} \mathbb{I}\{c_k = b_k\} \times \mathbb{I}\{c_3 + b_1 + b_2 = 0\}$$



Conventional factor graph

- Transmitter

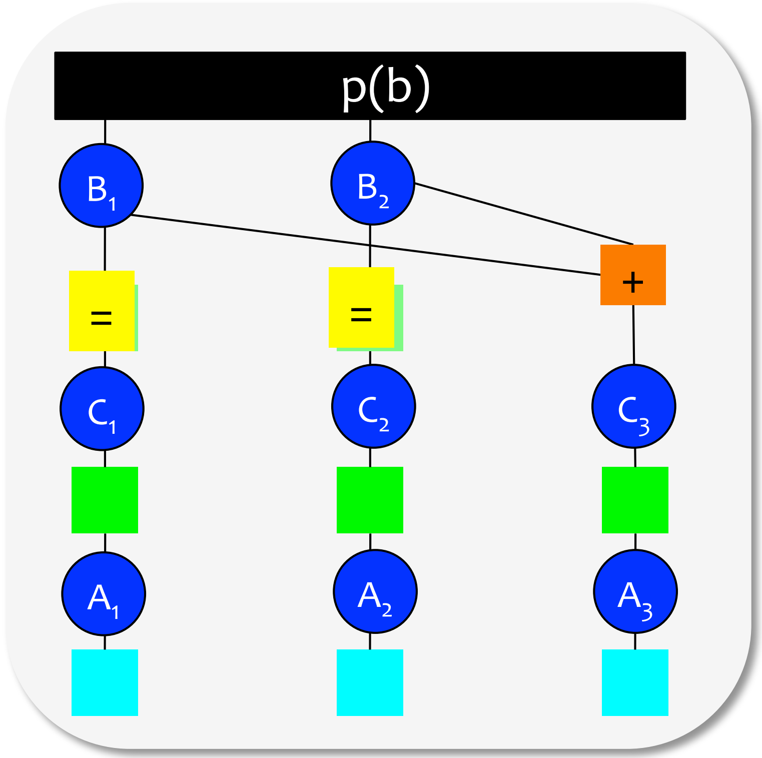


$$p(\mathbf{b},\mathbf{c},\mathbf{a},\mathbf{y}) \propto p(\mathbf{y}|\mathbf{a})p(\mathbf{a}|\mathbf{c})p(\mathbf{c}|\mathbf{b})p(\mathbf{b})$$

$$p(\mathbf{y}|\mathbf{a}) = \prod_{k=1}^{3} p(y_k|a_k)$$

$$p(\mathbf{a}|\mathbf{c}) = \prod_{k=1}^{3} p(a_k|c_k)$$

$$p(\mathbf{c}|\mathbf{b}) = \prod_{k=1}^{2} \mathbb{I}\{c_k = b_k\} \times \mathbb{I}\{c_3 + b_1 + b_2 = 0\}$$



Example: messages for normal FG

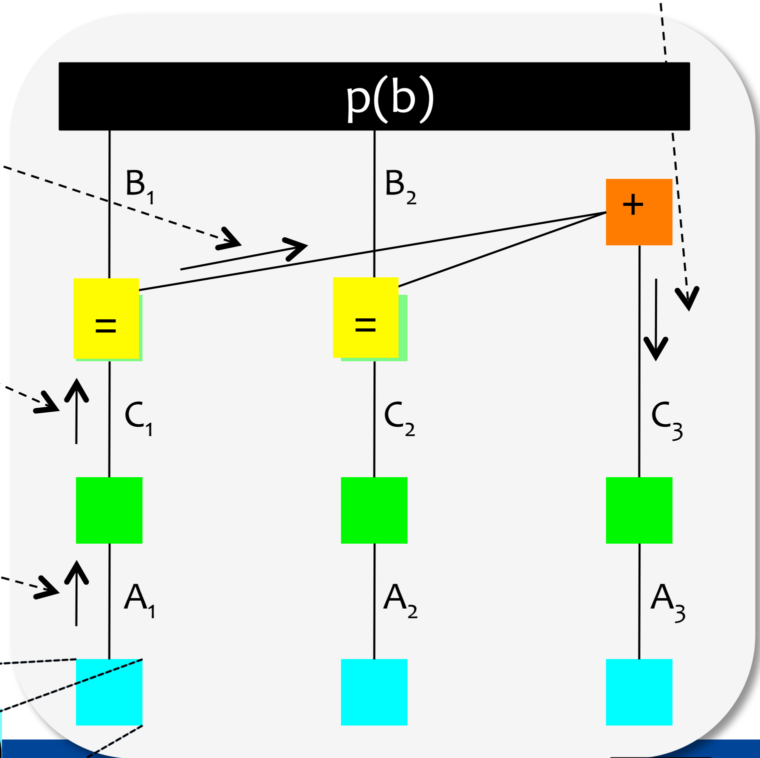
Solution

$$\mu^{\downarrow}_{C_3}(c_3) \propto \sum_{b_1,b_2} \mathbb{I}\{b_1+b_2+c_3=0\}\mu_{B_1\to[+]}(b_1)\mu_{B_2\to[+]}(b_2)$$
$$= \sum_{b_1=0}^{1} \mu_{B_1\to[+]}(b_1)\mu_{B_2\to[+]}(b_1+c_3)$$
$$= \mu_{B_1\to[+]}(0)\mu_{B_2\to[+]}(c_3)+\mu_{B_1\to[+]}(1)\mu_{B_2\to[+]}(1+c_3)$$

$$\mu_{B_1\to[+]}(b_1) = \mu^{\uparrow}_{C_1}(b_1)$$

$$\mu^{\uparrow}_{C_1}(c_1) \propto \sum_{a_1} \mathbb{I}\{a_1 = 2c_1 - 1\}\mu^{\uparrow}_{A_1}(a_1)$$
$$= \mu^{\uparrow}_{A_1}(2c_1-1)$$

$$\mu^{\uparrow}_{A_1}(a_1) \propto \exp(-\frac{1}{2\sigma^2}|a_1-y_1|^2)$$



$$p(y_1|a_1) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(a_1-y_1)^2\right)$$

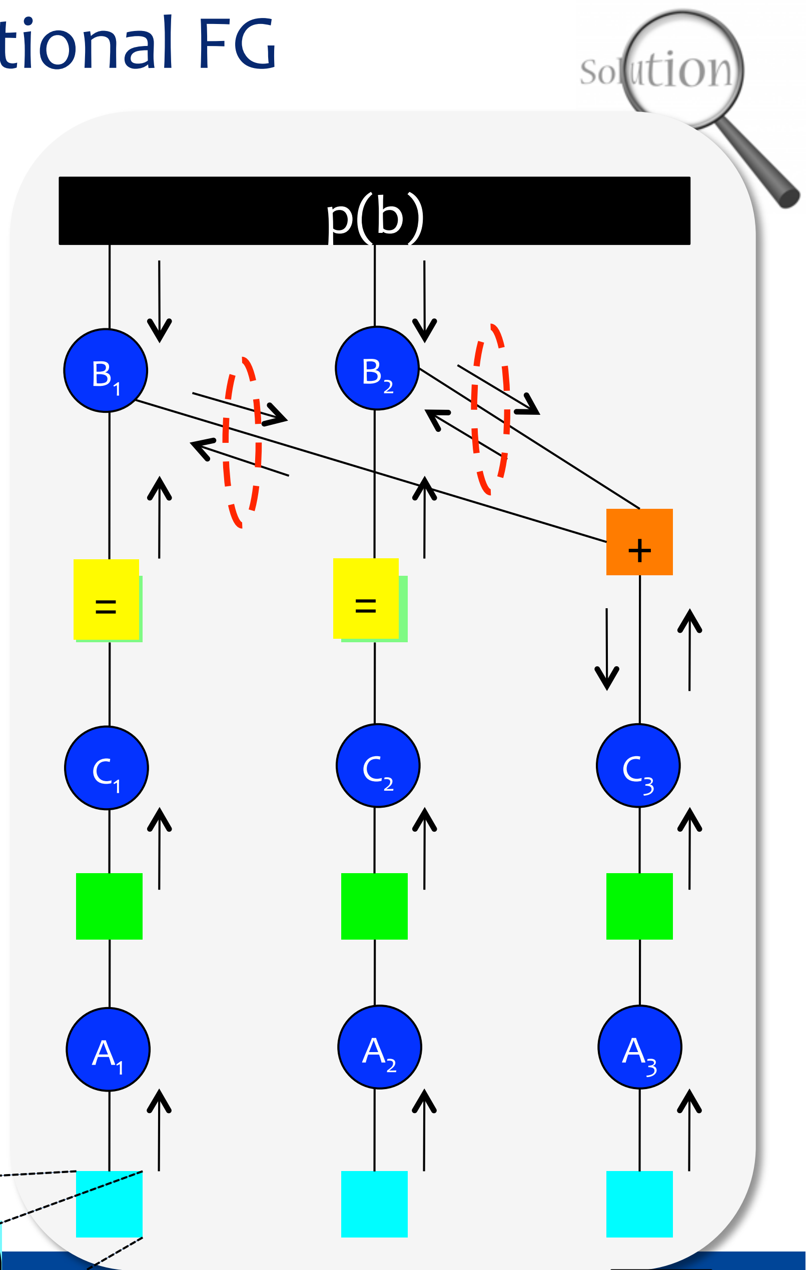
Example: messages for conventional FG

$$\mu^{\downarrow}_{C_3}(c_3) \propto \sum_{b_1,b_2} \mathbb{I}\{b_1+b_2+c_3=0\}\mu_{B_1\to[+]}(b_1)\mu_{B_2\to[+]}(b_2)$$
$$= \sum_{b_1=0}^{1} \mu_{B_1\to[+]}(b_1)\mu_{B_2\to[+]}(b_1+c_3)$$
$$= \mu_{B_1\to[+]}(0)\mu_{B_2\to[+]}(c_3) + \mu_{B_1\to[+]}(1)\mu_{B_2\to[+]}(1+c_3)$$

$$\mu_{B_1\to[+]}(b_1) = \mu^{\uparrow}_{C_1}(b_1)$$

$$\mu^{\uparrow}_{C_1}(c_1) \propto \sum_{a_1} \mathbb{I}\{a_1 = 2c_1 - 1\}\mu^{\uparrow}_{A_1}(a_1)$$
$$= \mu^{\uparrow}_{A_1}(2c_1-1)$$

$$\mu^{\uparrow}_{A_1}(a_1) \propto \exp(-\frac{1}{2\sigma^2}|a_1-y_1|^2)$$

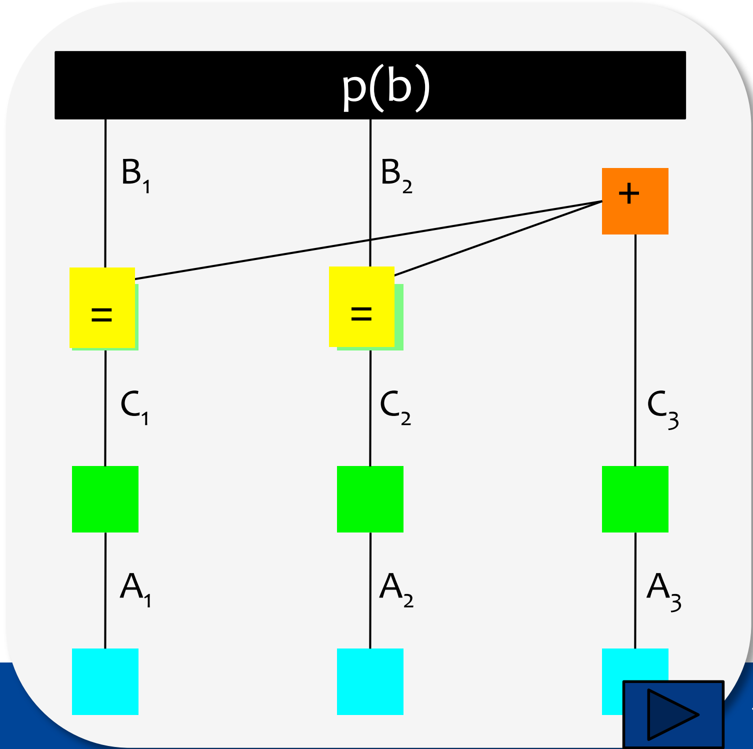


$$p(y_1|a_1) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(a_1-y_1)^2\right)$$

MATLAB code

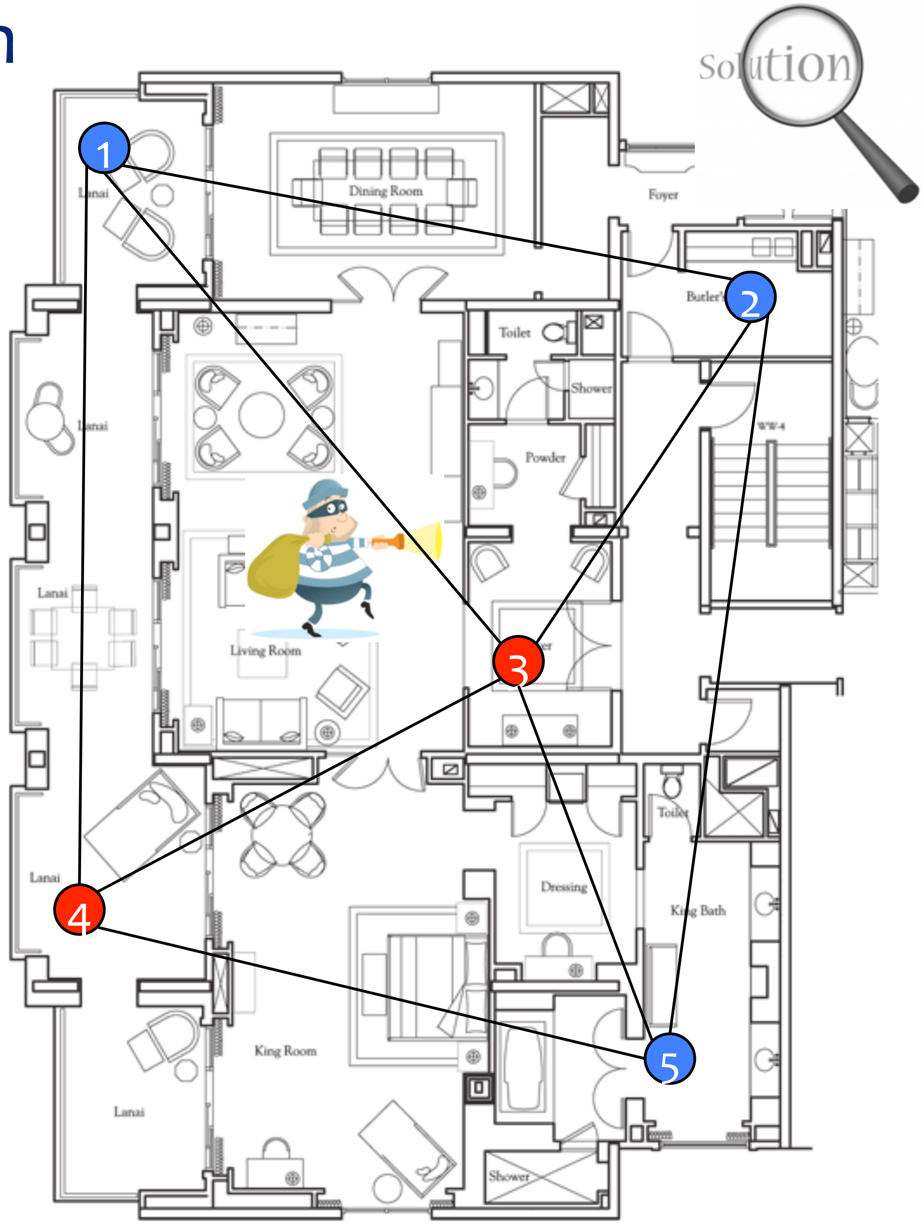
```
b1=round(rand(1)); b2=round(rand(1)); c=[b1 b2 mod(b1+b2,2)];
a=2*c-1;
y=a+randn(1,3)*sigma; % sigma is preset noise st.dev.
L_a_up=2*y/sigma^2;
L_c_up=L_a_up;
L_b1_plus=L_c_up(1); L_b2_plus=L_c_up(2);
L_plus_b1=max_star(L_c_up(3),L_b2_plus)-max_star(L_b2_plus+L_c_up(3),0);
L_plus_b2=max_star(L_c_up(3),L_b1_plus)-max_star(L_b1_plus+L_c_up(3),0);
L_belief_b1=L_plus_b1+L_b1_plus;
L_belief_b2=L_plus_b2+L_b2_plus;
b1_hat=(sign(L_belief_b1)+1)/2
b2_hat=(sign(L_belief_b2)+1)/2
```

```
function r=max_star(a,b)
   r=max(a,b)+log(1+exp(-abs(a-b)));
```



Example: intruder detection

- Goal: detect presence of thief $x \in \{0,1\}$, uniform prior
- Every node makes a measurement $y_i \in \{0=\text{blue}, 1=\text{red}\}$, with missed detection prob. q_{MD}/ false alarm prob. q_{FA} (with $q_{MD} >> q_{FA}$)
- Determine
  - a local likelihood
  - the initial states
  - good value for ε
  - consensus rules
  - final state
- Can you do consensus without the logs?



Intruder detection

- Local likelihoods, observation **y**=[00110]
  1. [p(y_1=0|x=0) p(y_1=0|x=1)]=[1-q_{FA} q_{MD}]
  2. [p(y_2=0|x=0) p(y_2=0|x=1)]=[1-q_{FA} q_{MD}]
  3. [p(y_3=1|x=0) p(y_3=1|x=1)]=[q_{FA} 1-q_{MD}]
  4. [p(y_4=1|x=0) p(y_4=1|x=1)]=[q_{FA} 1-q_{MD}]
  5. [p(y_5=0|x=0) p(y_5=0|x=1)]=[1-q_{FA} q_{MD}]
- Initial states
  - No intruder: $\mathbf{z}^{(0)}(0)$=[log(1-q_{FA}) log(1-q_{FA}) log(q_{FA}) log(q_{FA}) log(1-q_{FA})]
  - Intruder: $\mathbf{z}^{(1)}(0)$=[log(q_{MD}) log(q_{MD}) log(1-q_{MD}) log(1-q_{MD}) log(1-q_{MD})]
- Value for ε: max degree is 4, so ε=0.25
- Consensus rules (for node 1)

$$z_1^{(0)}(k) = z_1^{(0)}(k-1) + \varepsilon \sum_{j\in\{2,3,4\}} (z_j^{(0)}(k-1) - z_1^{(0)}(k-1))$$

$$z_1^{(1)}(k) = z_1^{(1)}(k-1) + \varepsilon \sum_{j\in\{2,3,4\}} (z_j^{(1)}(k-1) - z_1^{(1)}(k-1))$$

Intruder detection

- Final state, for any node i

$$z_i^{(0)}(k) \to \frac{1}{5}(3\log(1-q_{\mathrm{FA}}) + 2\log(q_{\mathrm{FA}}))$$
$$z_i^{(1)}(k) \to \frac{1}{5}(3\log(q_{\mathrm{MD}}) + 2\log(1-q_{\mathrm{MD}}))$$

Example

- $p(x,y)=f(x_1,x_2)g(x_3)$
  - How should q(x) look like?
  - Evaluate
    $$\mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$
  - What can you say about the complexity?

- $p(x,y)=f(x_1)g(x_2)h(x_3)$
  - How should q(x) look like?
  - Evaluate
    $$\mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$
  - What can you say about the complexity?

Example

- $p(x,y)=f(x_1,x_2)g(x_3)$
  - $q(x)=q(x_1,x_2)q(x_3)$
  - Evaluate

$$\mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$

$$\mathcal{H}(q) = -\sum_{x_1,x_2} q(x_1,x_2) \log q(x_1,x_2) - \sum_{x_3} q(x_3) \log q(x_3)$$

$$\sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k) = \sum_{x_1,x_2} q(x_1,x_2) \log f(x_1,x_2) + \sum_{x_3} q(x_3) \log g(x_3)$$

  - Complexity: depends on structure of p(x,y)

Special case: when FG of p(x,y) is a tree

- Suppose $p(\mathbf{x},\mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = \prod_k f_k(\mathbf{x}_k)$ with $\mathbf{x}_k$ containing at least two variables (using grouping)
- For trees (d_n is the degree of variable node X_n) it can be shown that

$$p(\mathbf{x}|\mathbf{y}) = \frac{\prod_k p(\mathbf{x}_k|\mathbf{y})}{\prod_{n=1}^{N} p(x_n|\mathbf{y})^{d_n-1}}$$

- So we can limit ourselves to

$$q(\mathbf{x}) = \frac{\prod_k q(\mathbf{x}_k)}{\prod_{n=1}^{N} q(x_n)^{d_n-1}}$$

- Why does the problem become easy?

$$\log p(\mathbf{y}) = \max_{q \in \mathcal{F}_p} \mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k)$$

Special case: when FG of p(x,y) is a tree

1. Entropy term is easy

$$
\begin{aligned}
\mathcal{H}(q) &= -\sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k)\log q(\mathbf{x}_k) + \sum_n (d_n - 1)\sum_{x_n} q(x_n)\log q(x_n) \\
&= \sum_n \mathcal{H}(q_n) - \Big(\sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k)\log q(\mathbf{x}_k) - \sum_n d_n \sum_{x_n} q(x_n)\log q(x_n)\Big) \\
&= \sum_n \mathcal{H}(q_n) - \sum_k \mathcal{I}(q(\mathbf{x}_k))
\end{aligned}
$$

$$
\mathcal{I}(q(\mathbf{x}_k)) = \sum_{\mathbf{x}_k} q(\mathbf{x}_k)\log \frac{q(\mathbf{x}_k)}{\prod_{n:X_n \in \mathbf{X}_k} q(x_n)}
$$

2. Constraint set fully defined by $q(\mathbf{x}_k), q(x_n), \forall n, k$

Simplified case

- Suppose p(x|y) factorizes completely $p(\mathbf{x},\mathbf{y}) = \prod_n f_n(x_n)$
- Approximation will be $q(\mathbf{x}) = \prod_n q_n(x_n)$
- Then

$$\mathcal{H}(q) + \sum_k \sum_{\mathbf{x}_k} q(\mathbf{x}_k) \log f_k(\mathbf{x}_k) = \sum_n \sum_{x_n} q_n(x_n) \left(\log f_n(x_n) - \log q_n(x_n)\right)$$

- Leads to the following problem

$$\begin{aligned} \text{maximize} \quad & \sum_n \sum_{x_n} q_n(x_n) \left(\log f_n(x_n) - \log q_n(x_n)\right) \\ \text{s.t.} \quad & q_n(x_n) \geq 0, \forall n, x_n \\ & \sum_{x_n} q_n(x_n) = 1, \forall n \end{aligned}$$

- Take the derivative of the objective w.r.t. $q_m(x_m)$

Simplified case

- Problem

$$\begin{aligned} \text{maximize} \quad & \sum_{n} \sum_{x_n} q_n(x_n) \left(\log f_n(x_n) - \log q_n(x_n) \right) \\ \text{s.t.} \quad & q_n(x_n) \geq 0, \forall n, x_n \\ & \sum_{x_n} q_n(x_n) = 1, \forall n \end{aligned}$$

- Partial Lagrangian

$$\mathcal{L}(q, \boldsymbol{\lambda}) = \sum_{n} \sum_{x_n} q_n(x_n)(\log f_n(x_n - \log q_n(x_n))) + \sum_{n} \lambda_n (\sum_{x_n} q_n(x_n) - 1)$$

$$\text{s.t.} \quad q_n(x_n) \geq 0, \forall n, x_n$$

- Derivative w.r.t. $q_m(x_m)$, set to zero

$$\log f_m(x_m) - \log q_m(x_m) - q_m(x_m) \frac{1}{q_m(x_m)} + \lambda_m = 0$$

$$\log q_m(x_m) = \log f_m(x_m) - \lambda_m$$

$$q_m(x_m) \propto f_m(x_m)$$