Scalable Algorithms for Distributed Statistical Inference

Animashree Anandkumar

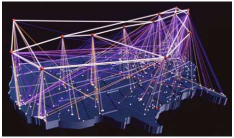
School of Electrical and Computer Engineering
Cornell University, Ithaca, NY 14853

Currently visiting EECS, MIT, Cambridge, MA 02139

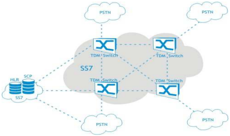
PhD Committee: Lang Tong, Aaron Wagner, Kevin Tang
David Williamson, Ananthram Swami.

Supported by ARL-CTA, ARO, IBM PhD Fellowship

Introduction



Internet



PSTN

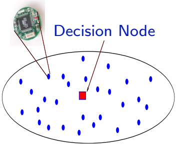
Traditional Wire-line Networks

- Fixed networks
- Over-provisioned links
- Layered architecture

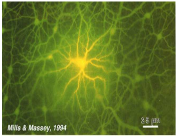
Emerging Networks

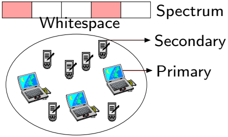
- Large, complex, ubiquitous
- Resource constraints
  e.g., Energy, Bandwidth
- Heterogeneous nodes
- Interaction between different networks

Sensor Networks

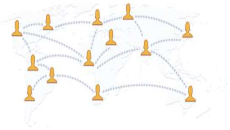


Biological Networks





Cognitive Networks

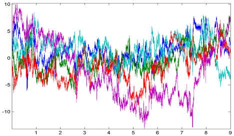


Social Networks

Network Data: Integrated View

Characteristics

- Large number of samples, multi-modal
- Noisy, imperfect or missing data

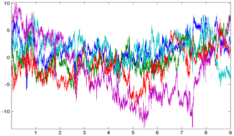


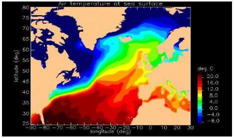
Network Data: Integrated View

Characteristics

- Large number of samples, multi-modal
- Noisy, imperfect or missing data
- Data Locality: relationship between data at nearby nodes

  e.g., Temperature & other environmental data



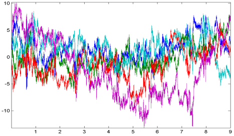


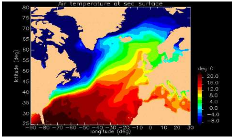
Network Data: Integrated View

Characteristics

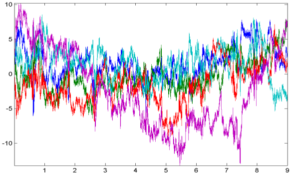
- Large number of samples, multi-modal
- Noisy, imperfect or missing data
- Data Locality: relationship between data at nearby nodes

  e.g., Temperature & other environmental data





Data to Knowledge: Specific Goals of Networks

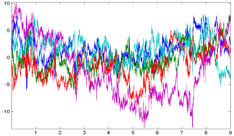


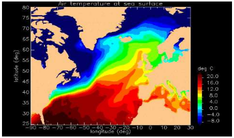
Network Data: Integrated View

Characteristics

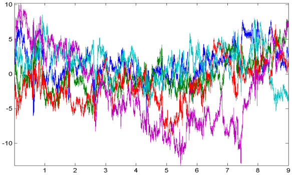
- Large number of samples, multi-modal
- Noisy, imperfect or missing data
- Data Locality: relationship between data at nearby nodes

  e.g., Temperature & other environmental data





Data to Knowledge: Specific Goals of Networks



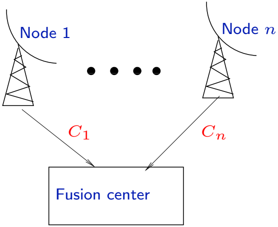
Distributed Statistical Inference

Distributed Statistical Inference

Inference about a random population made from its samples

Distributed Statistical Inference

Inference about a random population made from its samples

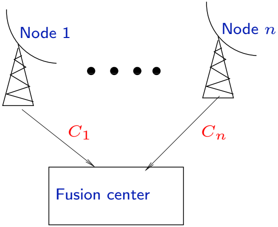


Classical Inference

- Quantization and inference rules
- Fixed configuration (one hop)
- Independent data at nodes

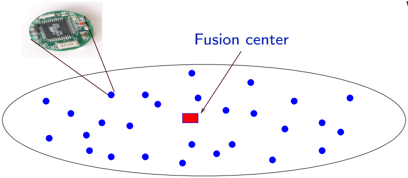
Distributed Statistical Inference

Inference about a random population made from its samples



Classical Inference

- Quantization and inference rules
- Fixed configuration (one hop)
- Independent data at nodes

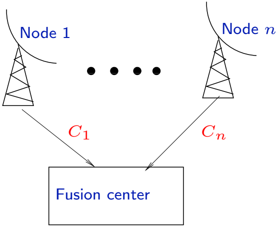


Wireless Sensor Networks for Inference

- Multihop data fusion
- Constraints on fusion costs
- Transmission and fusion policies
- Correlated data: local dependence

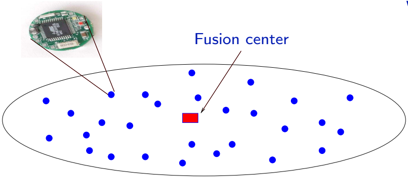
Distributed Statistical Inference

Inference about a random population made from its samples



Classical Inference

- Quantization and inference rules
- Fixed configuration (one hop)
- Independent data at nodes



Wireless Sensor Networks for Inference

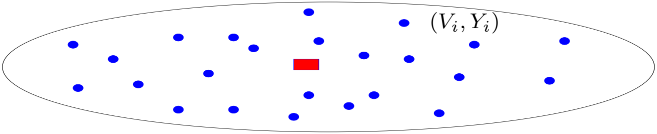
- Multihop data fusion
- Constraints on fusion costs
- Transmission and fusion policies
- Correlated data: local dependence

Scaling of Fusion Costs & Inference Accuracy with Network Size

Setup: Fusion of Sensor Data & Fusion Cost

Setup

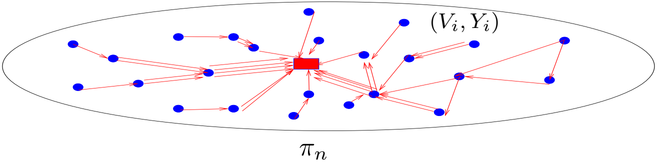
- Consider n randomly distributed sensors $V_i \in \mathbf{V}_n$ making random observations $\mathbf{Y}_{\mathbf{V}_n}$.
- Fusion center makes decision on underlying hypothesis using data
- The fusion policy π_n schedules transmissions and computations at sensor nodes in $\mathbf{V}_n$



Setup: Fusion of Sensor Data & Fusion Cost

Setup

- Consider n randomly distributed sensors $V_i \in \mathbf{V}_n$ making random observations $\mathbf{Y}_{\mathbf{V}_n}$.
- Fusion center makes decision on underlying hypothesis using data
- The fusion policy π_n schedules transmissions and computations at sensor nodes in $\mathbf{V}_n$



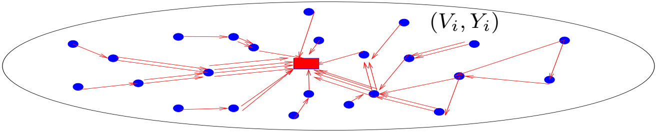
Setup: Fusion of Sensor Data & Fusion Cost

Setup

- Consider n randomly distributed sensors $V_i \in \mathbf{V}_n$ making random observations $\mathbf{Y}_{\mathbf{V}_n}$.
- Fusion center makes decision on underlying hypothesis using data
- The fusion policy π_n schedules transmissions and computations at sensor nodes in $\mathbf{V}_n$

Cost of a Fusion Policy

The average fusion cost $\bar{\mathcal{E}}(\pi_n) \stackrel{\Delta}{=} \frac{1}{n} \sum_{V_i \in \mathbf{V}_n} \mathcal{E}_i(\pi_n)$



π_n

Scaling of Fusion Cost & Lossless Fusion

Cost of a Fusion Policy

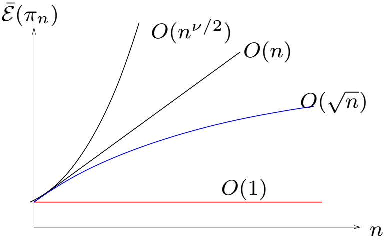
- The fusion policy π_n schedules transmissions of sensor nodes
- The average fusion cost $\bar{\mathcal{E}}(\pi_n) \stackrel{\Delta}{=} \frac{1}{n} \sum_{V_i \in \mathbf{V}_n} \mathcal{E}_i(\pi_n)$
- How does $\bar{\mathcal{E}}(\pi_n)$ behave?

A. Anandkumar, J.E. Yukich, L. Tong, A. Swami, "Energy scaling laws for distributed inference in random networks," accepted to *IEEE JSAC: Special Issues on Stochastic Geometry and Random Graphs for Wireless Networks*, Dec. 2008 (on ArXiv)

Scaling of Fusion Cost & Lossless Fusion

Cost of a Fusion Policy

- The fusion policy π_n schedules transmissions of sensor nodes
- The average fusion cost $\bar{\mathcal{E}}(\pi_n) \stackrel{\Delta}{=} \frac{1}{n} \sum_{V_i \in \mathbf{V}_n} \mathcal{E}_i(\pi_n)$
- How does $\bar{\mathcal{E}}(\pi_n)$ behave?

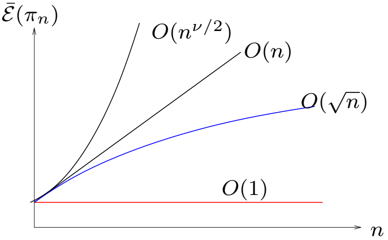


A. Anandkumar, J.E. Yukich, L. Tong, A. Swami, "Energy scaling laws for distributed inference in random networks," accepted to *IEEE JSAC: Special Issues on Stochastic Geometry and Random Graphs for Wireless Networks*, Dec. 2008 (on ArXiv)

Scaling of Fusion Cost & Lossless Fusion

Cost of a Fusion Policy

- The fusion policy π_n schedules transmissions of sensor nodes
- The average fusion cost
  $$\bar{\mathcal{E}}(\pi_n) \stackrel{\Delta}{=} \frac{1}{n} \sum_{V_i \in \mathbf{V}_n} \mathcal{E}_i(\pi_n)$$
- How does $\bar{\mathcal{E}}(\pi_n)$ behave?



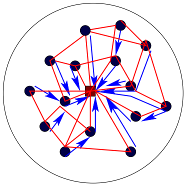
Constraint: No Loss in Inference Performance

A fusion policy is lossless if it results in no loss of inference performance at fusion center- as if all raw data available at fusion center

---

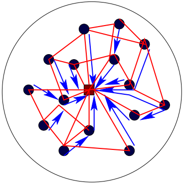
A. Anandkumar, J.E. Yukich, L. Tong, A. Swami, "Energy scaling laws for distributed inference in random networks," accepted to *IEEE JSAC: Special Issues on Stochastic Geometry and Random Graphs for Wireless Networks*, Dec. 2008 (on ArXiv)

Problem Statement-I : Energy Scaling Laws

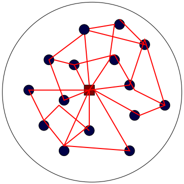


Fusion policy graph

Problem Statement-I : Energy Scaling Laws

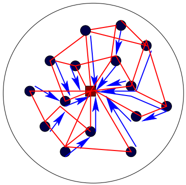


Fusion policy graph

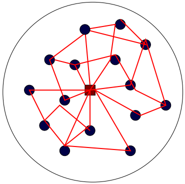


Network graph

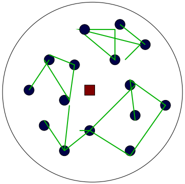
Problem Statement-I : Energy Scaling Laws



Fusion policy graph

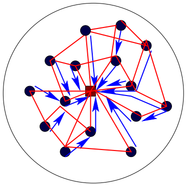


Network graph

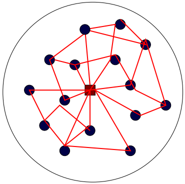


Dependency graph

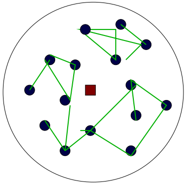
Problem Statement-I : Energy Scaling Laws



Fusion policy graph



Network graph



Dependency graph

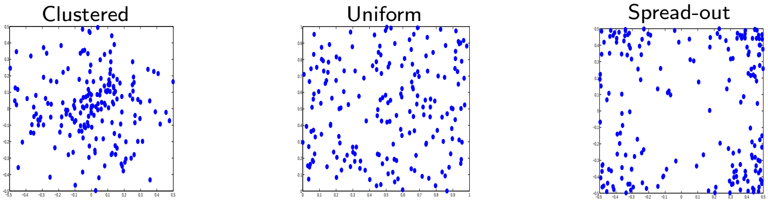
Scalable Lossless Fusion Policy

Find a sequence of scalable policies $\{\pi_n\}$, i.e.,

$$\limsup_{n\to\infty} \frac{1}{n} \sum_{V_i \in \mathbf{V}_n} \mathcal{E}_i(\pi_n) \overset{L^2}{=} \bar{\mathcal{E}}^{\pi}_{\infty} < \infty,$$

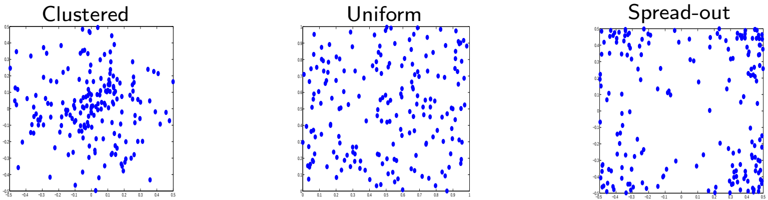
with small scaling constant $\bar{\mathcal{E}}^{\pi}_{\infty}$ such that optimal inference is achieved at fusion center (lossless) for a class of node configurations.

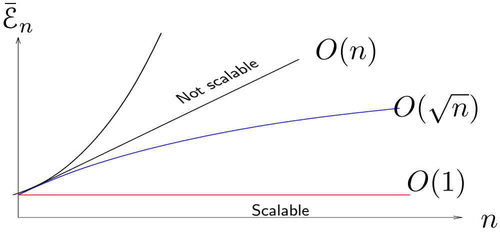
Problem II: Optimal Node Placement Distribution



Goal: what placement strategy has best asymptotic average energy $\bar{\mathcal{E}}^{\pi}_{\infty}$?

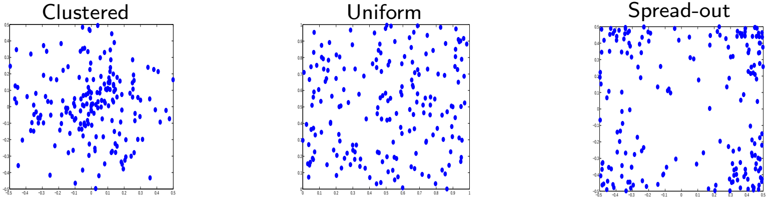
Problem II: Optimal Node Placement Distribution

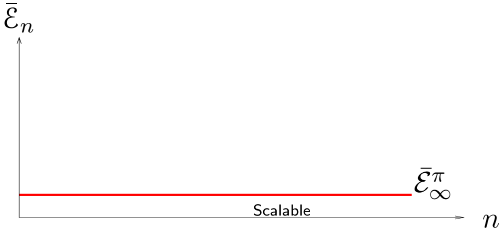




Goal: what placement strategy has best asymptotic average energy $\bar{\mathcal{E}}^{\pi}_{\infty}$?

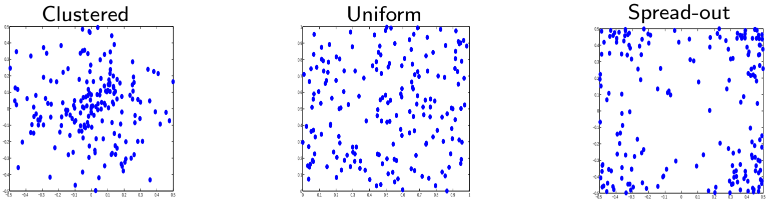
Problem II: Optimal Node Placement Distribution

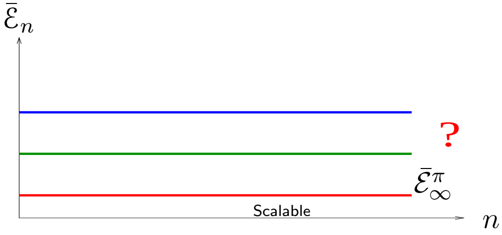




Goal: what placement strategy has best asymptotic average energy $\bar{\mathcal{E}}_\infty^\pi$?

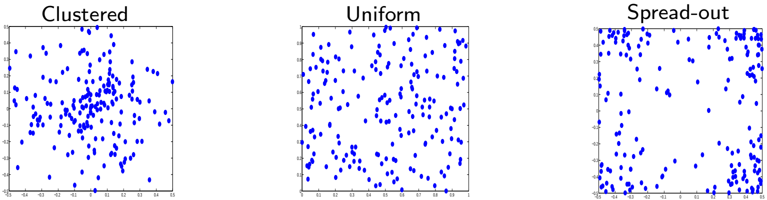
Problem II: Optimal Node Placement Distribution

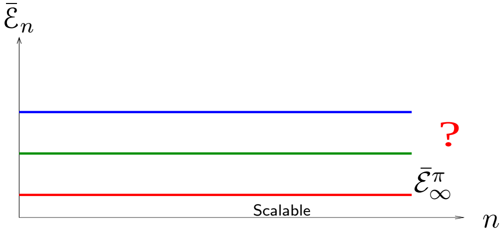




Goal: what placement strategy has best asymptotic average energy $\bar{\mathcal{E}}^{\pi}_{\infty}$?

Problem II: Optimal Node Placement Distribution





Goal: what placement strategy has best asymptotic average energy $\bar{\mathcal{E}}^{\pi}_{\infty}$?

Challenge: Network & dependency graphs influenced by node locations

Related Work: Scaling Laws in Networks

Capacity Scaling in Wireless Networks (Gupta & Kumar, IT '00)

- Information flow between nodes, $O(\frac{1}{\sqrt{n \log n}})$ scaling

Routing Correlated Data

- Algorithms for gathering correlated data (Cristescu, B. Beferull-Lozano & Vetterli, TON '06)

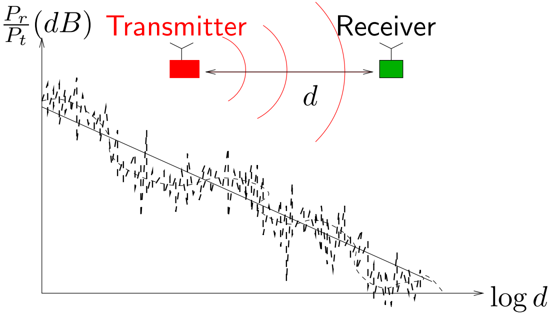
Function Computation

- Rate scaling for Computation of separable functions at a sink (Giridhar & Kumar, JSAC '05)
- Bounds on time required to achieve a distortion level for distributed computation (Ayaso, Dahleh & Shah, ISIT '08)

Outline

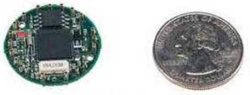
- Models, assumptions, and problem formulations
  - Propagation, network, and inference models
- Insights from special cases
- Markov random fields
- Scalable data fusion for Markov random field
- Some related problems
- Conclusion and future work

Propagation Model and Assumptions



- Cost for perfect reception: $\mathcal{E}_T = O(d^\nu)$. ν: path-loss exponent.
- Scheduling to avoid interference.
- Quantization effects ignored.

Berkeley Mote



Characteristics

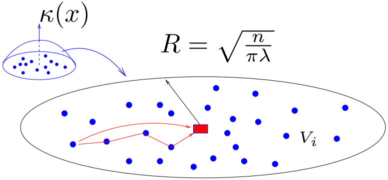
- ☐ Transmission range: 500-1000 ft.
- ☐ Current draw: 25mA (tx), 8mA (rx)
- ☐ Rate: 38.4 Kbaud.

A. Ephremides, "Energy concerns in wireless networks," *IEEE Wireless Comm.*, no. 4, Aug. 2002

Network Graph Model For Communication

Random Node Placement

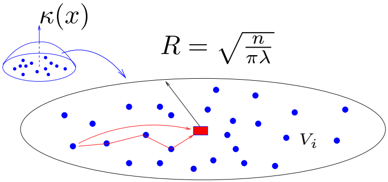
- Points $X_i \overset{\text{i.i.d.}}{\sim} \kappa(x)$ on unit ball Q_1
- $\kappa(x)$ bounded away from 0 and ∞
- Network scaled to a fixed density λ: $V_i = \sqrt{\frac{n}{\lambda}} X_i$



Network Graph Model For Communication

Random Node Placement

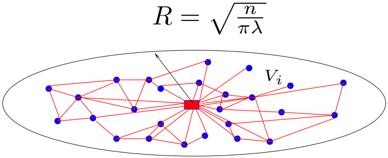
- Points $X_i \overset{\text{i.i.d.}}{\sim} \kappa(x)$ on unit ball Q_1
- $\kappa(x)$ bounded away from 0 and ∞
- Network scaled to a fixed density λ: $V_i = \sqrt{\frac{n}{\lambda}} X_i$



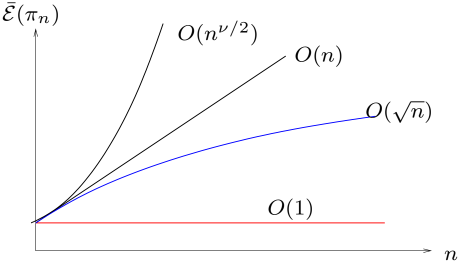
Network Graph for Communication

- Connected set of comm. links
- Energy & interference constraints

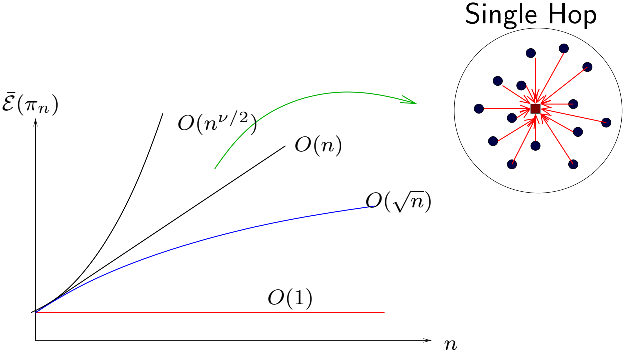
  Disc graph above critical radius
- Adjustable transmission power



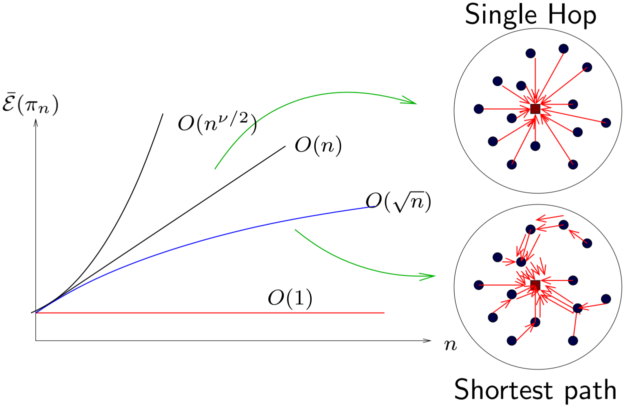
Routing Strategies With No Fusion Are Not Scalable



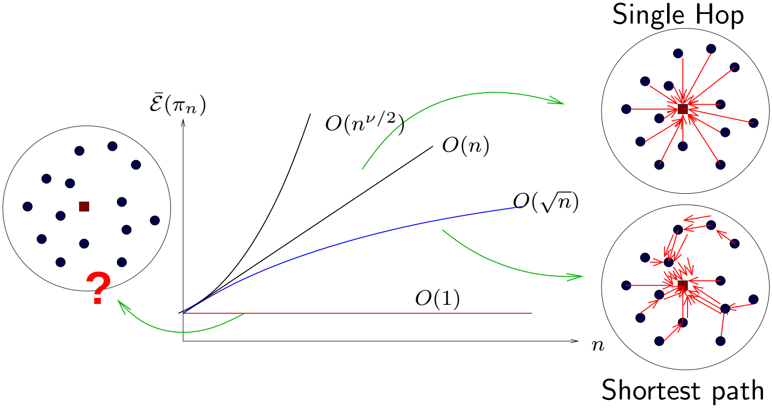
Routing Strategies With No Fusion Are Not Scalable



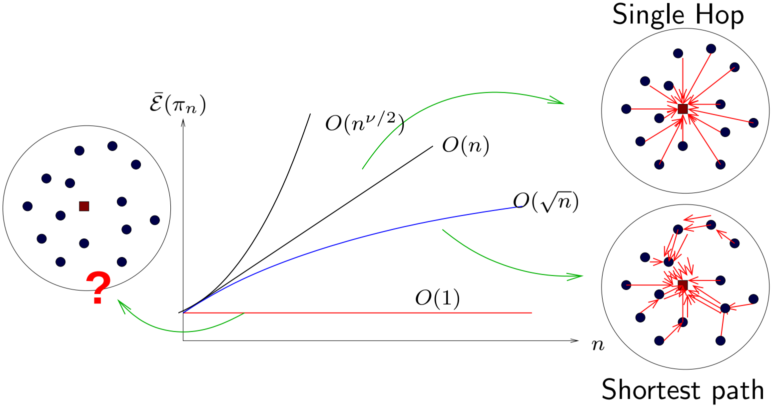
Routing Strategies With No Fusion Are Not Scalable



Routing Strategies With No Fusion Are Not Scalable



Routing Strategies With No Fusion Are Not Scalable



Incorporate inference model (dependency graph) for scalable fusion policy

Distributed Computation of Sufficient Statistic

Example: Sufficient Statistic for Mean Estimation $Y_1, \ldots, Y_n \stackrel{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

$\sum_i Y_i$ sufficient to estimate θ: no performance loss

E. Dynkin, "Necessary and sufficient statistics for a family of probability distributions," *Tran. Math, Stat. and Prob.*, vol. 1, pp. 23-41, 1961

Distributed Computation of Sufficient Statistic

Example: Sufficient Statistic for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

$\sum_i Y_i$ sufficient to estimate θ: no performance loss

Sufficient Statistic For Inference: No Performance Loss

- Dimensionality reduction: lower communication costs
- Minimal Sufficiency: Maximum dimensionality reduction

E. Dynkin, "Necessary and sufficient statistics for a family of probability distributions," *Tran. Math, Stat. and Prob.*, vol. 1, pp. 23-41, 1961

Distributed Computation of Sufficient Statistic

Example: Sufficient Statistic for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

$\sum_i Y_i$ sufficient to estimate θ: no performance loss

Sufficient Statistic For Inference: No Performance Loss

- Dimensionality reduction: lower communication costs
- Minimal Sufficiency: Maximum dimensionality reduction

Binary Hypothesis Testing: Decide $Y_1, \ldots, Y_n \sim f_0(\mathbf{Y}_n)$ or $f_1(\mathbf{Y}_n)$

E. Dynkin, "Necessary and sufficient statistics for a family of probability distributions," *Tran. Math, Stat. and Prob.*, vol. 1, pp. 23-41, 1961

Distributed Computation of Sufficient Statistic

Example: Sufficient Statistic for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

$\sum_i Y_i$ sufficient to estimate θ: no performance loss

Sufficient Statistic For Inference: No Performance Loss

- Dimensionality reduction: lower communication costs
- Minimal Sufficiency: Maximum dimensionality reduction

Binary Hypothesis Testing: Decide $Y_1, \ldots, Y_n \sim f_0(\mathbf{Y}_n)$ or $f_1(\mathbf{Y}_n)$

Minimal Sufficient Statistic for Binary Hypothesis Testing (Dynkin 61)

$$\text{Log Likelihood Ratio: } L_{\mathcal{G}}(\mathbf{Y}_n) = \log \frac{f_0(\mathbf{Y}_n)}{f_1(\mathbf{Y}_n)}$$

E. Dynkin, "Necessary and sufficient statistics for a family of probability distributions," *Tran. Math, Stat. and Prob.*, vol. 1, pp. 23-41, 1961

Distributed Computation of Sufficient Statistic

Example: Sufficient Statistic for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

$\sum_i Y_i$ sufficient to estimate θ: no performance loss

Sufficient Statistic For Inference: No Performance Loss

- Dimensionality reduction: lower communication costs
- Minimal Sufficiency: Maximum dimensionality reduction

Binary Hypothesis Testing: Decide $Y_1, \ldots, Y_n \sim f_0(\mathbf{Y}_n)$ or $f_1(\mathbf{Y}_n)$

Minimal Sufficient Statistic for Binary Hypothesis Testing (Dynkin 61)

$$\text{Log Likelihood Ratio: } L_{\mathcal{G}}(\mathbf{Y}_n) = \log \frac{f_0(\mathbf{Y}_n)}{f_1(\mathbf{Y}_n)}$$

Is there a scalable fusion policy for computing likelihood ratio?

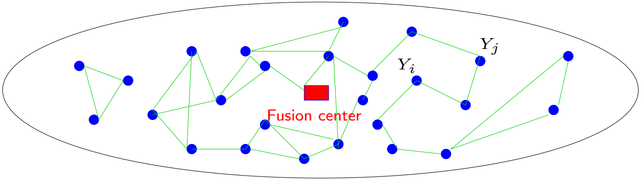
E. Dynkin, "Necessary and sufficient statistics for a family of probability distributions," *Tran. Math, Stat. and Prob.*, vol. 1, pp. 23-41, 1961

Inference Model and Assumptions

- Random location $\mathbf{V}_n \stackrel{\Delta}{=} (V_1, \cdots, V_n)$ and sensor data $\mathbf{Y}_{\mathbf{V}_n}$.
- Binary hypothesis: $\mathcal{H}_0$ vs. $\mathcal{H}_1$: $\quad \mathcal{H}_k : \mathbf{Y}_{\mathbf{V}_n} \sim f(\mathbf{y}_{\mathbf{v}_n} | \mathbf{V}_n = \mathbf{v}_n; \mathcal{H}_k)$

Inference Model and Assumptions

- Random location $\mathbf{V}_n \stackrel{\Delta}{=} (V_1, \cdots, V_n)$ and sensor data $\mathbf{Y}_{\mathbf{V}_n}$.
- Binary hypothesis: $\mathcal{H}_0$ vs. $\mathcal{H}_1$: $\quad \mathcal{H}_k : \mathbf{Y}_{\mathbf{V}_n} \sim f(\mathbf{y}_{\mathbf{v}_n} | \mathbf{V}_n = \mathbf{v}_n; \mathcal{H}_k)$
- $\mathbf{Y}_{\mathbf{V}_n}$: Markov random field with dependency graph $\mathcal{G}_k(\mathbf{V}_n)$



Dependency neighbor condition: No direct "interaction" between two nodes unless they are neighbors in dependency graph

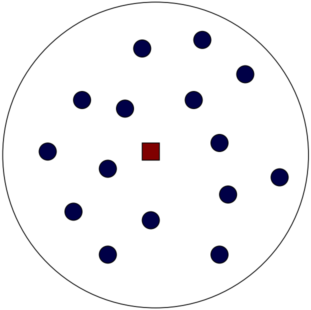
Outline

- Models, assumptions, and problem formulations
  - Propagation, network, and inference models
- Insights from special cases
- Markov random fields
- Scalable data fusion for Markov random field
- Some related problems
- Conclusion and future work

Optimal Fusion: the IID Case

Consider i.i.d. observations

$$\mathcal{H}_k : \mathbf{Y}_{\mathbf{V}} \sim \prod_{i \in \mathbf{V}} f_k(Y_i)$$



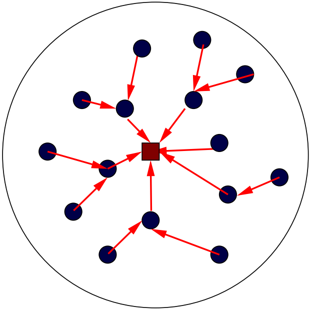
Sufficient statistic

$$L(\mathbf{Y}_{\mathbf{V}}) = \log \frac{f_0(\mathbf{Y}_{\mathbf{V}})}{f_1(\mathbf{Y}_{\mathbf{V}})} = \sum_{i \in \mathbf{V}} L(Y_i)$$

Optimal Fusion: the IID Case

Consider i.i.d. observations

$$\mathcal{H}_k : \mathbf{Y_V} \sim \prod_{i \in \mathbf{V}} f_k(Y_i)$$



Sufficient statistic

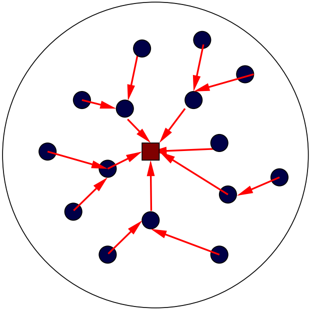
$$L(\mathbf{Y_V}) = \log \frac{f_0(\mathbf{Y_V})}{f_1(\mathbf{Y_V})} = \sum_{i \in \mathbf{V}} L(Y_i)$$

The optimal data fusion is the LLR aggregation over the MST (why?)

Optimal Fusion: the IID Case

Consider i.i.d. observations

$$\mathcal{H}_k : \mathbf{Y_V} \sim \prod_{i \in \mathbf{V}} f_k(Y_i)$$



Sufficient statistic

$$L(\mathbf{Y_V}) = \log \frac{f_0(\mathbf{Y_V})}{f_1(\mathbf{Y_V})} = \sum_{i \in \mathbf{V}} L(Y_i)$$

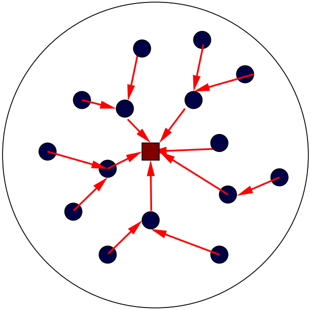
The optimal data fusion is the LLR aggregation over the MST (why?)

- each node must transmit at least once
- MST minimizes power-weighted edge sum: $\min \sum_i |e_i|^\nu$

Optimal Fusion: the IID Case

Consider i.i.d. observations

$$\mathcal{H}_k : \mathbf{Y_V} \sim \prod_{i \in \mathbf{V}} f_k(Y_i)$$



Sufficient statistic

$$L(\mathbf{Y_V}) = \log \frac{f_0(\mathbf{Y_V})}{f_1(\mathbf{Y_V})} = \sum_{i \in \mathbf{V}} L(Y_i)$$

The optimal data fusion is the LLR aggregation over the MST (why?)

- each node must transmit at least once
- MST minimizes power-weighted edge sum: $\min \sum_i |e_i|^\nu$

Assume network graph contains MST

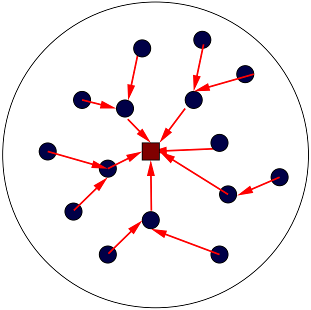
Optimal Fusion: Energy Analysis

Energy per node is

$$\bar{\mathcal{E}}(\pi_n^{\mathsf{MST}}) = \frac{1}{n} \sum_{e \in \mathsf{MST}_n} |e|^\nu$$

Steele'88, Yukich'00

$$\frac{1}{n} \sum_{e \in \mathsf{MST}_n} |e|^\nu \xrightarrow{L^2} \bar{\mathcal{E}}_\infty^{\mathsf{MST}} < \infty$$



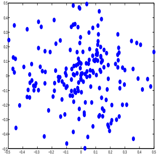
Scalable fusion along MST for independent data

J. E. Yukich, "Asymptotics for weighted minimal spanning trees on random points," *Stochastic Processes and their Applications*, vol. 85, No. 1, pp. 123-138, Jan. 2000.

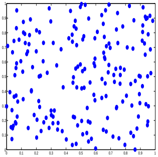
Role of Sensor Location Distribution

Better scaling constant $\bar{\mathcal{E}}_{\infty}^{\mathsf{MST}} = \zeta(\nu; \mathsf{MST}) \int_{Q_1} \kappa(x)^{1-\frac{\nu}{2}} dx$?

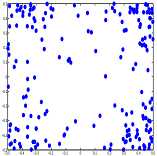
Clustered



Uniform



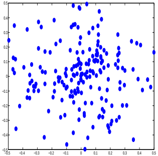
Spread-out



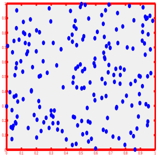
Role of Sensor Location Distribution

Better scaling constant $\bar{\mathcal{E}}_{\infty}^{\mathsf{MST}} = \zeta(\nu; \mathsf{MST}) \int_{Q_1} \kappa(x)^{1-\frac{\nu}{2}} dx$?

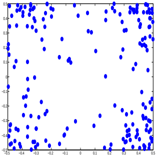
Clustered



Uniform



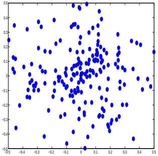
Spread-out



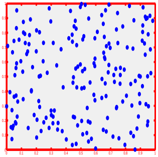
Role of Sensor Location Distribution

Better scaling constant $\bar{\mathcal{E}}_\infty^{\text{MST}} = \zeta(\nu; \text{MST}) \int_{Q_1} \kappa(x)^{1-\frac{\nu}{2}} dx$?

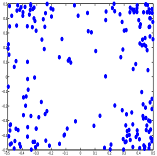
Clustered



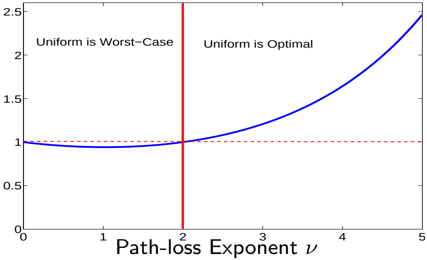
Uniform



Spread-out



Ratio of $\bar{\mathcal{E}}_\infty^{\text{MST}}$ of clustered and spread-out placements with respect to uniform



Outline

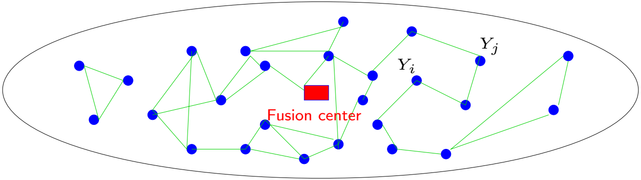
- Models, assumptions, and problem formulations
  - Propagation, network, and inference models
- Insights from special cases
- Markov random fields
  - Conditional-independence Relationships
  - Hammersley-Clifford Theorem
  - Form of Likelihood Ratio
- Scalable data fusion for Markov random field
- Some related problems
- Conclusion and future work

Inference Model and Assumptions

- Random location $\mathbf{V}_n \stackrel{\Delta}{=} (V_1, \cdots, V_n)$ and samples $\mathbf{Y}_{\mathbf{V}_n}$.
- Binary hypothesis: $\mathcal{H}_0$ vs. $\mathcal{H}_1$: $\quad \mathcal{H}_k : \mathbf{Y}_{\mathbf{V}_n} \sim f(\mathbf{y}_{\mathbf{v}_n} | \mathbf{V}_n = \mathbf{v}_n; \mathcal{H}_k)$

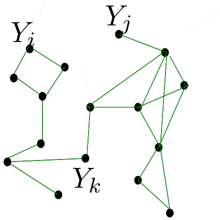
Inference Model and Assumptions

- Random location $\mathbf{V}_n \stackrel{\Delta}{=} (V_1, \cdots, V_n)$ and samples $\mathbf{Y}_{\mathbf{V}_n}$.
- Binary hypothesis: $\mathcal{H}_0$ vs. $\mathcal{H}_1$: $\quad \mathcal{H}_k : \mathbf{Y}_{\mathbf{V}_n} \sim f(\mathbf{y}_{\mathbf{v}_n} | \mathbf{V}_n = \mathbf{v}_n; \mathcal{H}_k)$
- $\mathbf{Y}_{\mathbf{V}_n}$: Markov random field with dependency graph $\mathcal{G}_k(\mathbf{V}_n)$



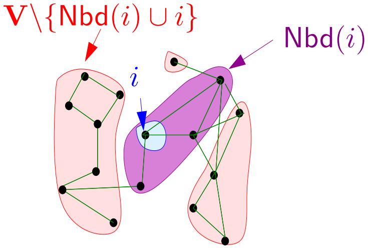
Dependency Graph and Markov Random Field

- Consider an undirected graph $\mathcal{G}(\mathbf{V})$, each vertex $V_i \in \mathbf{V}$ is associated with a random variable Y_i



Dependency Graph and Markov Random Field

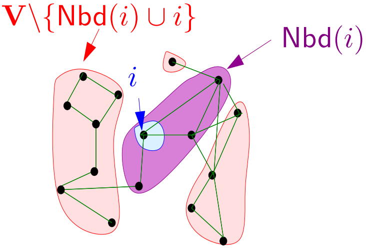
- Consider an undirected graph $\mathcal{G}(\mathbf{V})$, each vertex $V_i \in \mathbf{V}$ is associated with a random variable Y_i



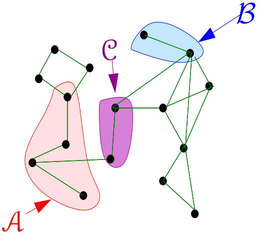
$$\mathbf{Y}_i \perp\!\!\!\perp \mathbf{Y}_{\mathbf{V}\backslash\{\text{Nbd}(i)\cup i\}} \big| \mathbf{Y}_{\text{Nbd}(i)}$$

Dependency Graph and Markov Random Field

- Consider an undirected graph $\mathcal{G}(\mathbf{V})$, each vertex $V_i \in \mathbf{V}$ is associated with a random variable Y_i
- For any disjoint sets $\mathcal{A}, \mathcal{B}, \mathcal{C}$ such that $\mathcal{C}$ separates $\mathcal{A}$ and $\mathcal{B}$,

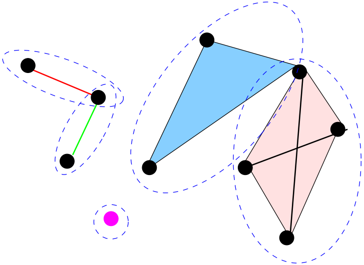


$$\mathbf{Y}_i \perp\!\!\!\perp \mathbf{Y}_{\mathbf{V}\backslash\{\text{Nbd}(i) \cup i\}} | \mathbf{Y}_{\text{Nbd}(i)}$$



$$\mathbf{Y}_{\mathcal{A}} \perp\!\!\!\perp \mathbf{Y}_{\mathcal{B}} | \mathbf{Y}_{\mathcal{C}}$$

Likelihood Function of MRF



Hammersley-Clifford Theorem'71

Let f be joint pdf of MRF with graph $\mathcal{G}(\mathbf{V})$,

$$-\log f(\mathbf{Y}_{\mathbf{V}}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$

where $\mathcal{C}$ is the set of maximal cliques.

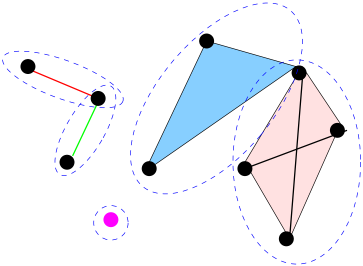
Likelihood Function of MRF

Hammersley-Clifford Theorem'71

Let f be joint pdf of MRF with graph $\mathcal{G}(\mathbf{V})$,

$$-\log f(\mathbf{Y_V}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$

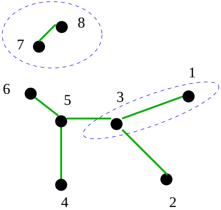
where $\mathcal{C}$ is the set of maximal cliques.



Gaussian MRF:

$$-\log f(\mathbf{Y_V}) = \frac{1}{2}\Big(-n \log 2\pi - \log|\mathbf{\Sigma_V}| + \sum_{(i,j)\in\mathcal{G}} \Sigma_{\mathbf{V}}^{-1}(i,j) Y_i Y_j + \sum_{i \in \mathbf{V}} \Sigma_{\mathbf{V}}^{-1}(i,i) Y_i^2\Big)$$

Dependency Graph



$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & X & & X & & & & & \\ 2 & & X & X & & X & & & \\ 3 & X & X & X & & & & & \\ 4 & & & & X & & & & \\ 5 & & & X & & X & X & & \\ 6 & & & & & X & X & & \\ 7 & & & & & & & X & X \\ 8 & & & & & & & X & X \end{array}$$

Inverse of Covariance Matrix

Inference Model and Assumptions

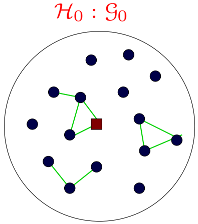
- Random location $\mathbf{V}_n \stackrel{\Delta}{=} (V_1, \cdots, V_n)$ and samples $\mathbf{Y}_{\mathbf{V}_n}$.
- Binary hypothesis: $\mathcal{H}_0$ vs. $\mathcal{H}_1$: $\quad \mathcal{H}_k : \mathbf{Y}_{\mathbf{V}_n} \sim f(\mathbf{y}_{\mathbf{v}_n} | \mathbf{V}_n = \mathbf{v}_n, \mathcal{H}_k)$

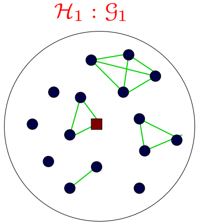
Inference Model and Assumptions

- Random location $\mathbf{V}_n \stackrel{\Delta}{=} (V_1, \cdots, V_n)$ and samples $\mathbf{Y}_{\mathbf{V}_n}$.
- Binary hypothesis: $\mathcal{H}_0$ vs. $\mathcal{H}_1$: $\quad \mathcal{H}_k : \mathbf{Y}_{\mathbf{V}_n} \sim f(\mathbf{y}_{\mathbf{v}_n} | \mathbf{V}_n = \mathbf{v}_n, \mathcal{H}_k)$
- $\mathbf{Y}_{\mathbf{V}_n}$: Markov random field with dependency graph $\mathcal{G}_k(\mathbf{V}_n)$

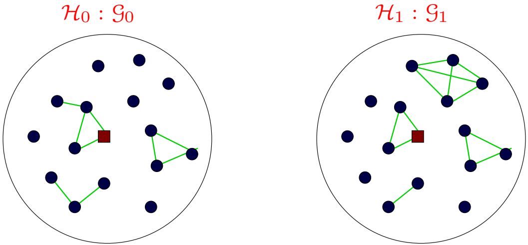
$$-\log f(\mathbf{Y}_{\mathbf{V}_n} | \mathcal{G}_k, \mathcal{H}_k) = \sum_{c \in \mathcal{C}_k} \Psi_{k,c}(\mathbf{Y}_c)$$

where $\mathcal{C}_{n,k}$ is the collection of maximal cliques $\Psi_{k,c}$ clique potentials.





Dependency Graph Model



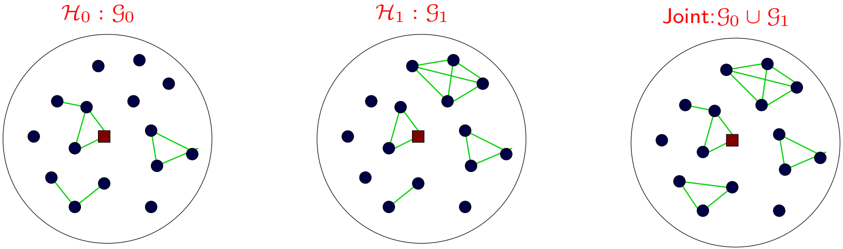
Recall Hammersley-Clifford Theorem

$$-\log f(\mathbf{Y}_{\mathbf{V}_n} | \mathcal{G}_k, \mathcal{H}_k) = \sum_{c \in \mathcal{C}_k} \Psi_{k,c}(\mathbf{Y}_c)$$

Minimal sufficient statistic

$$L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}}) = \log \frac{f(\mathbf{Y}_{\mathbf{V}} | \mathcal{G}_0, \mathcal{H}_0)}{f(\mathbf{Y}_{\mathbf{V}} | \mathcal{G}_1, \mathcal{H}_1)}$$

Dependency Graph Model



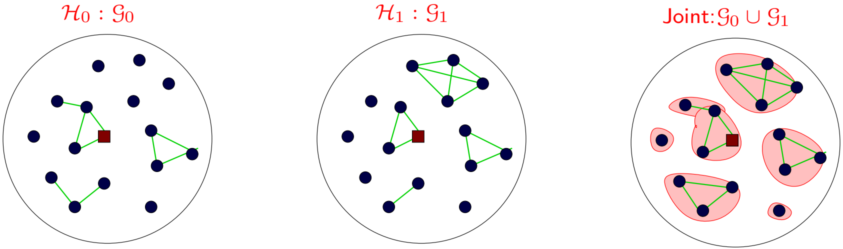
Recall Hammersley-Clifford Theorem

$$-\log f(\mathbf{Y}_{\mathbf{V}_n} | \mathcal{G}_k, \mathcal{H}_k) = \sum_{c \in \mathcal{C}_k} \Psi_{k,c}(\mathbf{Y}_c)$$

Minimal sufficient statistic

$$L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}}) = \log \frac{f(\mathbf{Y}_{\mathbf{V}} | \mathcal{G}_0, \mathcal{H}_0)}{f(\mathbf{Y}_{\mathbf{V}} | \mathcal{G}_1, \mathcal{H}_1)}$$

Dependency Graph Model



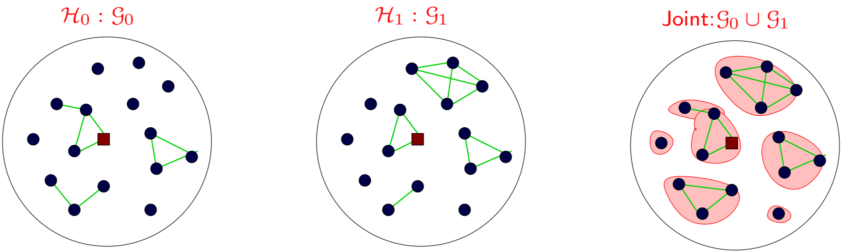
Recall Hammersley-Clifford Theorem

$$-\log f(\mathbf{Y}_{\mathbf{V}_n}|\mathcal{G}_k, \mathcal{H}_k) = \sum_{c \in \mathcal{C}_k} \Psi_{k,c}(\mathbf{Y}_c)$$

Minimal sufficient statistic

$$L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}}) = \log \frac{f(\mathbf{Y}_{\mathbf{V}}|\mathcal{G}_0, \mathcal{H}_0)}{f(\mathbf{Y}_{\mathbf{V}}|\mathcal{G}_1, \mathcal{H}_1)}$$

Dependency Graph Model



Recall Hammersley-Clifford Theorem

$$-\log f(\mathbf{Y}_{\mathbf{V}_n} | \mathcal{G}_k, \mathcal{H}_k) = \sum_{c \in \mathcal{C}_k} \Psi_{k,c}(\mathbf{Y}_c)$$

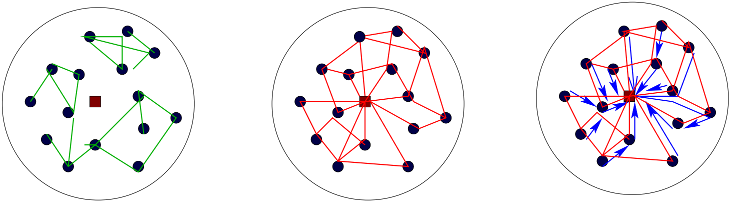
Minimal sufficient statistic

$$L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}}) = \log \frac{f(\mathbf{Y}_{\mathbf{V}} | \mathcal{G}_0, \mathcal{H}_0)}{f(\mathbf{Y}_{\mathbf{V}} | \mathcal{G}_1, \mathcal{H}_1)} = \sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c)$$

Outline

- Models, assumptions, and problem formulations
  - Propagation, network, and inference models
- Insights from special cases
- Markov random fields
- Scalable data fusion for Markov random field
  - A suboptimal scalable policy
  - Effects of sparsity on scalability
  - Energy scaling analysis
- Some related problems
- Conclusion and future work

Fusion for Markov Random Field



Dependency graph Network graph Fusion policy graph

Lossless Fusion Policies

Given the network and dependency graphs $(\mathcal{N}, \mathcal{G})$,

$$\mathfrak{F}_{\mathcal{G},\mathcal{N}} \stackrel{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y_V}) = \sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c) \text{ computable at the fusion center}\}.$$

Optimal fusion Policy: $\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}_n,\mathcal{N}_n}} \sum_i \mathcal{E}_i(\pi_n)$

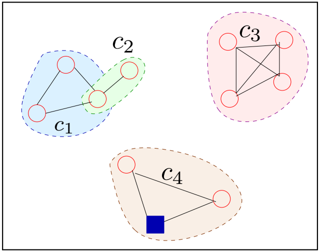
NP-hard: Steiner-tree reduction (INFOCOM '08)

Data Fusion for Markov Random Field (DFMRF)

Log-likelihood Ratio $L_{\mathcal{G}}(\mathbf{Y_V}) = \sum_{c\in\mathcal{C}} \phi(\mathbf{Y}_c)$

Step I: Data forwarding and local computation:

- Given dependency graph $\mathcal{G}$ and network graph $\mathcal{N}$.
- Randomly select a representative (processor) in each clique of $\mathcal{G}$.
- Clique members forward data to processor via SPR on $\mathcal{N}$

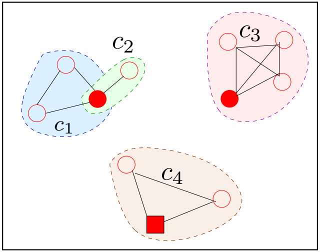


Data Fusion for Markov Random Field (DFMRF)

Log-likelihood Ratio $L_{\mathcal{G}}(\mathbf{Y_V}) = \sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c)$

Step I: Data forwarding and local computation:

- Given dependency graph $\mathcal{G}$ and network graph $\mathcal{N}$.
- Randomly select a representative (processor) in each clique of $\mathcal{G}$.
- Clique members forward data to processor via SPR on $\mathcal{N}$

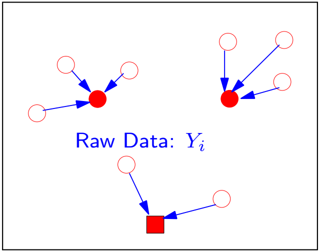


Data Fusion for Markov Random Field (DFMRF)

Log-likelihood Ratio $L_{\mathcal{G}}(\mathbf{Y_V}) = \sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c)$

Step I: Data forwarding and local computation:

- Given dependency graph $\mathcal{G}$ and network graph $\mathcal{N}$.
- Randomly select a representative (processor) in each clique of $\mathcal{G}$.
- Clique members forward data to processor via SPR on $\mathcal{N}$

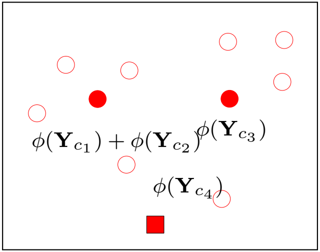


Data Fusion for Markov Random Field (DFMRF)

Log-likelihood Ratio $L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}}) = \sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c)$

Step I: Data forwarding and local computation:

- Given dependency graph $\mathcal{G}$ and network graph $\mathcal{N}$.
- Randomly select a representative (processor) in each clique of $\mathcal{G}$.
- Clique members forward data to processor via SPR on $\mathcal{N}$



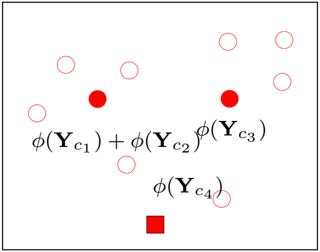
Data Fusion for Markov Random Field (DFMRF)

Log-likelihood Ratio $L_{\mathcal{G}}(\mathbf{Y_V}) = \sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c)$

Step I: Data forwarding and local computation:

- Given dependency graph $\mathcal{G}$ and network graph $\mathcal{N}$.
- Randomly select a representative (processor) in each clique of $\mathcal{G}$.
- Clique members forward data to processor via SPR on $\mathcal{N}$

Step II: aggregating LLR over MST



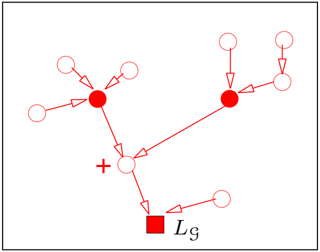
Data Fusion for Markov Random Field (DFMRF)

Log-likelihood Ratio $L_{\mathcal{G}}(\mathbf{Y_V}) = \sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c)$

Step I: Data forwarding and local computation:

- Given dependency graph $\mathcal{G}$ and network graph $\mathcal{N}$.
- Randomly select a representative (processor) in each clique of $\mathcal{G}$.
- Clique members forward data to processor via SPR on $\mathcal{N}$

Step II: aggregating LLR over MST



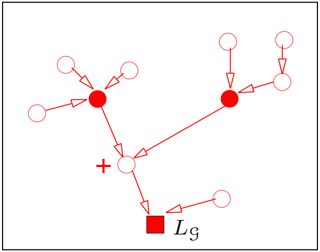
Data Fusion for Markov Random Field (DFMRF)

Log-likelihood Ratio $L_{\mathcal{G}}(\mathbf{Y_V}) = \sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c)$

Step I: Data forwarding and local computation:

- Given dependency graph $\mathcal{G}$ and network graph $\mathcal{N}$.
- Randomly select a representative (processor) in each clique of $\mathcal{G}$.
- Clique members forward data to processor via SPR on $\mathcal{N}$

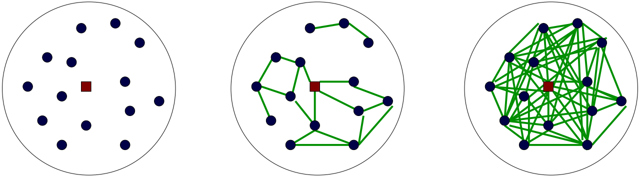
Step II: aggregating LLR over MST



Total energy consumption= Data Forwarding + MST Aggregation

Effects of Dependency Graph Sparsity on Scalability

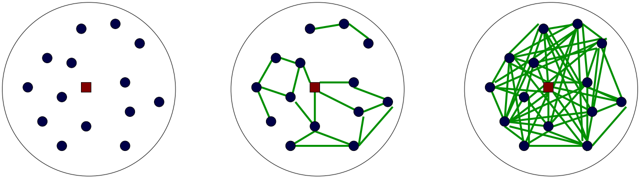
Sparsity of Dependency Graph



$\sum_{i \in \mathbf{V}} \phi(Y_i)$ $\sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c)$ $\phi(\mathbf{Y_V})$

Effects of Dependency Graph Sparsity on Scalability

Sparsity of Dependency Graph



$\sum_{i \in \mathbf{V}} \phi(Y_i)$ $\sum_{c \in \mathcal{C}} \phi(\mathbf{Y}_c)$ $\phi(\mathbf{Y_V})$

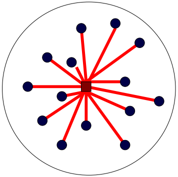
Stabilizing graph (Penrose-Yukich)

Local graph structure not affected by far away points (k-NNG, Disk)

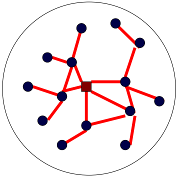
M. D. Penrose and J. E. Yukich, "Weak Laws Of Large Numbers In Geometric Probability," *Annals of Applied probability,* vol. 13, no. 1, pp. 277-303, 2003

Effects of Network Graph Sparsity on Scalability

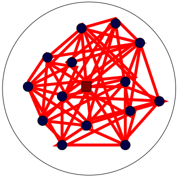
Sparsity of Network Graph



Single Hop



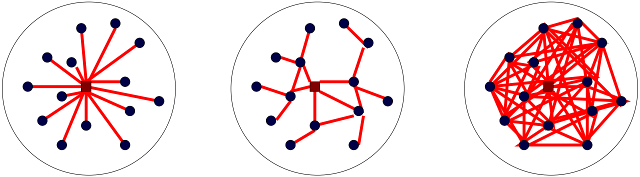
u-Spanner



Complete ($\overline{\mathcal{N}}_n$)

Effects of Network Graph Sparsity on Scalability

Sparsity of Network Graph



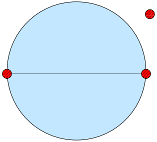
Single Hop u-Spanner Complete ($\overline{\mathcal{N}}_n$)

u-Spanner

Given network graph $\mathcal{N}_n$ and its completion $\overline{\mathcal{N}}_n$, $\mathcal{N}_n$ is a u-spanner if

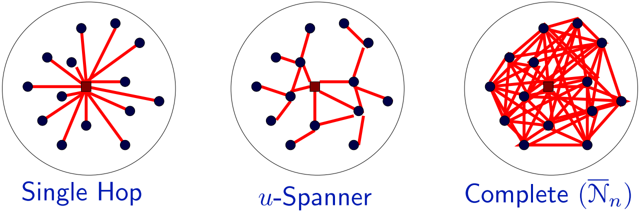
$$\max_{V_i, V_j \in \mathbf{V}_n} \frac{\mathcal{E}(V_i \to V_j; \text{SP on } \mathcal{N}_n)}{\mathcal{E}(V_i \to V_j; \text{SP on } \overline{\mathcal{N}}_n)} \leq u$$

Gabriel: $u = 1$ for $\nu \geq 2$



Effects of Network Graph Sparsity on Scalability

Sparsity of Network Graph

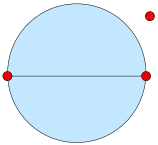


u-Spanner

Given network graph $\mathcal{N}_n$ and its completion $\overline{\mathcal{N}}_n$, $\mathcal{N}_n$ is a u-spanner if

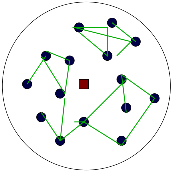
$$\max_{V_i, V_j \in \mathbf{V}_n} \frac{\mathcal{E}(V_i \to V_j; \text{SP on } \mathcal{N}_n)}{\mathcal{E}(V_i \to V_j; \text{SP on } \overline{\mathcal{N}}_n)} \leq u$$

Gabriel: $u = 1$ for $\nu \geq 2$



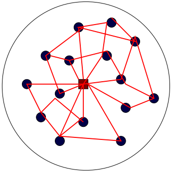
Longest edge $O(\sqrt{\log n})$

Main Result: Scalability of DFMRF



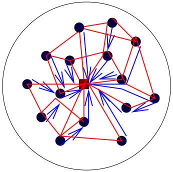
Dependency graph

Stabilizing



Network graph

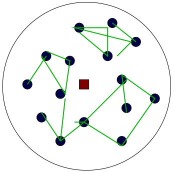
u-Spanner

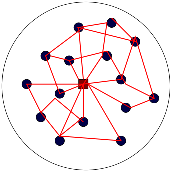


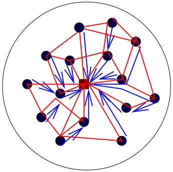
Fusion policy graph

DFMRF

Main Result: Scalability of DFMRF







Dependency graph — Stabilizing

Network graph — u-Spanner

Fusion policy graph — DFMRF

Scaling Constant for Scale-Invariant Graphs (k-NNG)

$$\limsup_{n\to\infty} \frac{\mathcal{E}(\pi_n^{\text{DFMRF}})}{n} \leq \lambda^{-\frac{\nu}{2}} \Big[\underbrace{u\,\zeta(\nu;\mathcal{G})}_{\text{data forward}} + \underbrace{\zeta(\nu;\ \text{MST})}_{\text{MST aggregation}} \Big] \int_{Q_1} \kappa(x)^{1-\frac{\nu}{2}}\, dx,$$

$$\zeta(\nu;\mathcal{G}) \stackrel{\Delta}{=} \mathbb{E} \sum_{(\mathbf{0},j)\in\mathcal{G}(\mathcal{P}_1\cup\{\mathbf{0}\})} |\mathbf{0},j|^{\nu}$$

Approximation Ratio for DFMRF

Recall $\mathfrak{F}_{\mathcal{G}} \stackrel{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y_V}) \text{ computable at the fusion center}\}$

$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

Approximation Ratio for DFMRF

Recall $\mathfrak{F}_{\mathcal{G}} \stackrel{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y_V})$ computable at the fusion center$\}$

$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

Lower and Upper Bounds For Optimal Fusion Policy

$$\mathcal{E}(\pi_n^{\mathsf{MST}}) \leq \mathcal{E}(\pi_n^*) \leq \mathcal{E}(\pi_n^{\mathsf{DFMRF}})$$

Approximation Ratio for DFMRF

Recall $\mathfrak{F}_{\mathcal{G}} \overset{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}})$ computable at the fusion center$\}$

$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

Lower and Upper Bounds For Optimal Fusion Policy

$$\mathcal{E}(\pi_n^{\text{MST}}) \leq \mathcal{E}(\pi_n^*) \leq \mathcal{E}(\pi_n^{\text{DFMRF}})$$

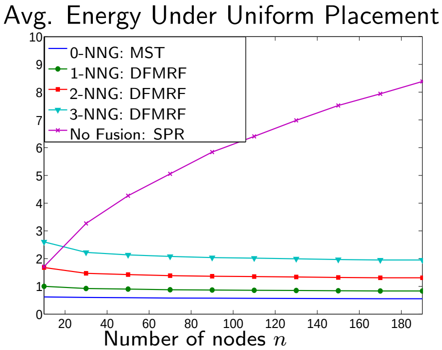
Approximation Ratio of DFMRF for k-NNG Dependency

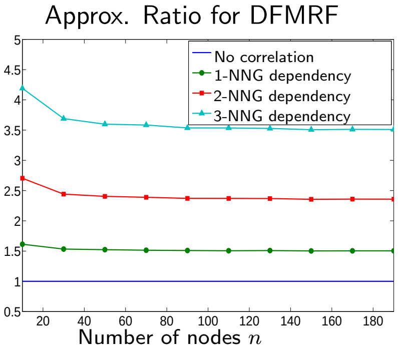
$$\limsup_{n \to \infty} \frac{\mathcal{E}(\pi_n^{\text{DFMRF}})}{\mathcal{E}(\pi_n^*)} \leq \left(1 + u \frac{\zeta(\nu; \mathcal{G})}{\zeta(\nu; \text{ MST})}\right)$$

Constant factor approximation for DFMRF for large networks

Approximation ratio independent of node placement for k-NNG

Simulation Results for k-NNG Dependency





What Have We Done and Left Out....

- Energy scaling laws
  - Assumed stabilizing dependency graph and u-spanner network graph
  - Defined a fusion policy π_n^{DFMRF} (DFMRF)
  - Scalability analysis: $\limsup_{n\to\infty} \frac{1}{n}\sum_i \mathcal{E}_i(\pi_n^{\text{DFMRF}}) \leq \bar{\mathcal{E}}_\infty^{\text{DFMRF}}$

$$\alpha \leq \bar{\mathcal{E}}_\infty^{\pi_*} \leq \bar{\mathcal{E}}_\infty^{\text{DFMRF}} \leq \beta < \infty$$

  - Asymptotic approximation ratio: $\frac{\beta}{\alpha}$.

What Have We Done and Left Out....

- Energy scaling laws
  - Assumed stabilizing dependency graph and u-spanner network graph
  - Defined a fusion policy π_n^{DFMRF} (DFMRF)
  - Scalability analysis: $\limsup_{n\to\infty} \frac{1}{n}\sum_i \mathcal{E}_i(\pi_n^{\text{DFMRF}}) \leq \bar{\mathcal{E}}_\infty^{\text{DFMRF}}$

$$\alpha \leq \bar{\mathcal{E}}_\infty^{\pi_*} \leq \bar{\mathcal{E}}_\infty^{\text{DFMRF}} \leq \beta < \infty$$

  - Asymptotic approximation ratio: $\frac{\beta}{\alpha}$.
- Remarks
  - Energy consumption is a key parameter for large sensor networks.
  - Sensor location is a new source of randomness in distributed inference
  - Asymptotic techniques are useful in overall network design.

What Have We Done and Left Out....

- Energy scaling laws
  - Assumed stabilizing dependency graph and u-spanner network graph
  - Defined a fusion policy π_n^{DFMRF} (DFMRF)
  - Scalability analysis: $\limsup_{n\to\infty} \frac{1}{n}\sum_i \mathcal{E}_i(\pi_n^{\text{DFMRF}}) \leq \bar{\mathcal{E}}_\infty^{\text{DFMRF}}$

$$\alpha \leq \bar{\mathcal{E}}_\infty^{\pi_*} \leq \bar{\mathcal{E}}_\infty^{\text{DFMRF}} \leq \beta < \infty$$

  - Asymptotic approximation ratio: $\frac{\beta}{\alpha}$.
- Remarks
  - Energy consumption is a key parameter for large sensor networks.
  - Sensor location is a new source of randomness in distributed inference
  - Asymptotic techniques are useful in overall network design.
- We have ignored several issues:
  - one-shot inference
  - quantization of measurements and link capacity constraints
  - perfect transmission/reception and scheduling
  - computation cost and overheads

Outline

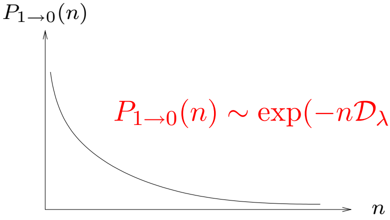
- Models, assumptions, and problem formulations
  - Propagation, network, and inference models
- Insights from special cases
- Markov random fields
- Scalable data fusion for Markov random field
- Some related problems
  - Error exponents on random graph
  - Cost performance tradeoff
  - Inference in finite networks
- Conclusion and future work

Design for Energy Constrained Inference

Error Exponent (IT '09, ISIT '09)

For MRF hypothesis with node density λ and distribution $\kappa(x)$,

$$-\frac{1}{n}\log P_{1\to 0}(n) \xrightarrow{?} \mathcal{D}_{\lambda,\kappa}$$



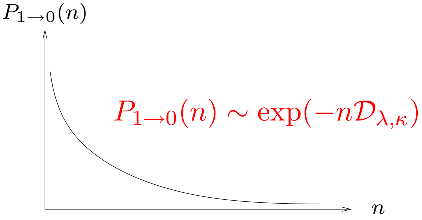
$$P_{1\to 0}(n) \sim \exp(-n\mathcal{D}_{\lambda,\kappa})$$

Design for Energy Constrained Inference

Error Exponent (IT '09, ISIT '09)

For MRF hypothesis with node density λ and distribution $\kappa(x)$,

$$-\frac{1}{n}\log P_{1\to 0}(n) \xrightarrow{?} \mathcal{D}_{\lambda,\kappa}$$



Design for Energy Constrained Inference (SP '08)

$$\max_{\lambda,\kappa,\pi} \mathcal{D}_{\lambda,\kappa} \quad \text{subject to } \bar{\mathcal{E}}^{\pi}_{\lambda,\kappa} \leq \bar{\mathcal{E}}_o$$

---

(1) A. Anandkumar, L. Tong, A. Swami, "Detection of Gauss-Markov Random Fields with Nearest-Neighbor Dependency," *IEEE Tran. on Information Theory*, Feb. 2009

(2) A. Anandkumar, J.E. Yukich, L. Tong, A. Willsky, "Detection Error Exponent for Spatially Dependent Samples in Random Networks," *Proc. of IEEE ISIT*, Jun. 2009

(3) A. Anandkumar, L. Tong, and A. Swami, "Optimal Node Density for Detection in Energy Constrained Random Networks," *IEEE Tran. Signal Proc.*, pp. 5232-5245, Oct. 2008.

Inference In Finite Fusion Networks

We have so far considered

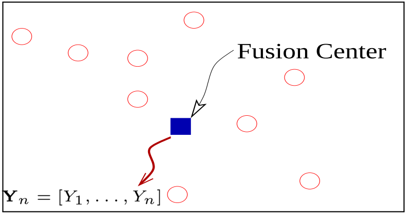
- Random node placement
- Scaling as $n \to \infty$

Harder problem

- Arbitrary node placement
- Finite n

Results (INFOCOM '08 & '09)

- Fusion scheme has a Steiner tree reduction
- Cost-performance tradeoff

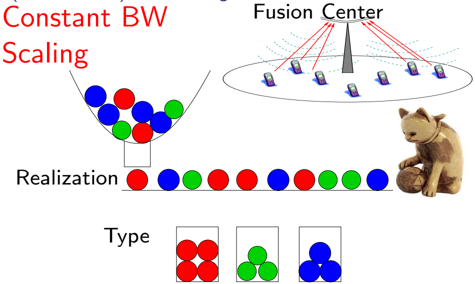


(1) A. Anandkumar, L. Tong, A. Swami, and A. Ephremides, "Minimum Cost Data Aggregation with Localized Processing for Statistical Inference," in *Proc. of INFOCOM*, April 2008

(2) A. Anandkumar, M. Wang, L. Tong, and A. Swami, "Prize-Collecting Data Fusion for Cost- Performance Tradeoff in Distributed Inference," in *Proc. of IEEE INFOCOM*, April 2009.

Medium Access Control

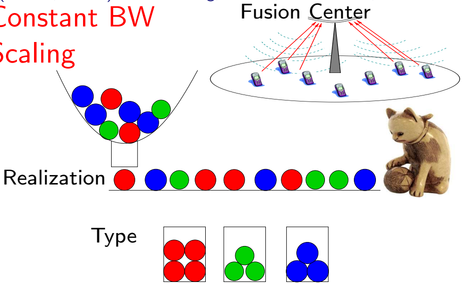
(SP '07, IT '08) With L. Tong, Cornell, & A. Swami, ARL



Medium Access Control

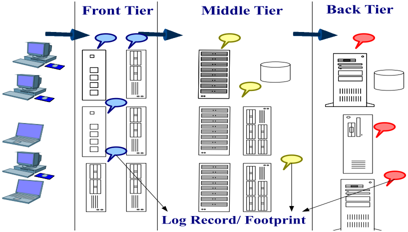
(SP '07, IT '08) With L. Tong, Cornell, & A. Swami, ARL

Constant BW Scaling



Transaction Monitoring

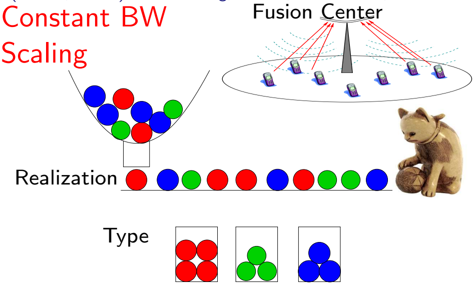
(Sigmetrics '08) With C. Bisdikian & D. Agrawal, IBM Research



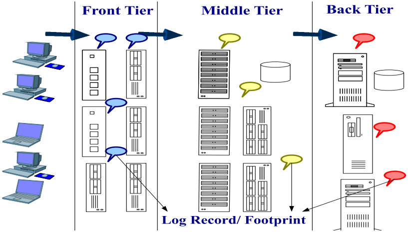
Decentralized Bipartite Matching

Medium Access Control
(SP '07, IT '08) With L. Tong, Cornell, & A. Swami, ARL

Constant BW Scaling



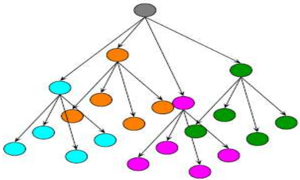
Transaction Monitoring
(Sigmetrics '08) With C. Bisdikian & D. Agrawal, IBM Research



Decentralized Bipartite Matching

Learning dependency models
(ISIT '09) With V. Tan, A. Willsky, MIT, & L. Tong, Cornell

SNR for learning

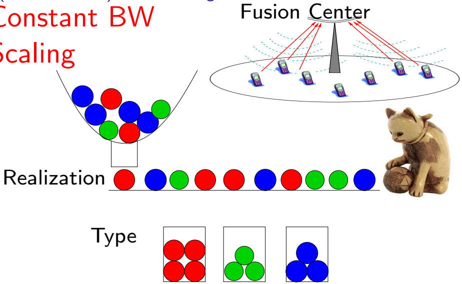


Medium Access Control

(SP '07, IT '08) With L. Tong, Cornell, & A. Swami, ARL

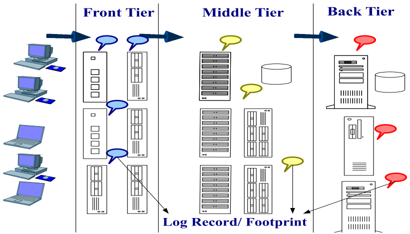
Constant BW

Scaling



Transaction Monitoring

(Sigmetrics '08) With C. Bisdikian & D. Agrawal, IBM Research

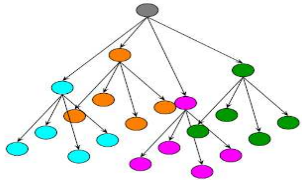


Decentralized Bipartite Matching

Learning dependency models

(ISIT '09) With V. Tan, A. Willsky, MIT, & L. Tong, Cornell

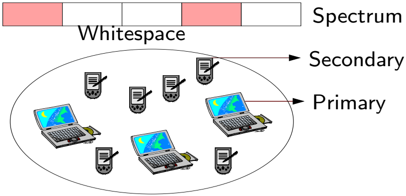
SNR for learning



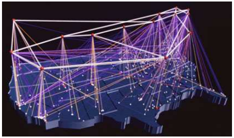
Competitive Learning

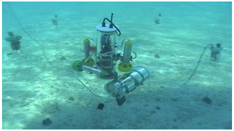
With A.K. Tang, Cornell Univ.

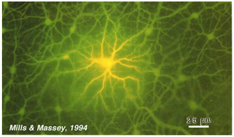
Regret-free under interference

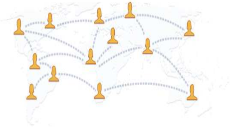


Holy Grail...





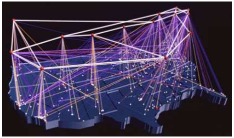


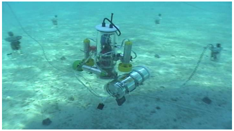


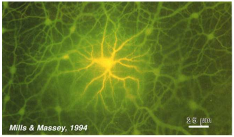
Networks

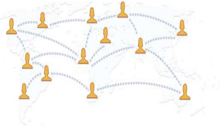
- Seamless operation
- Efficient resource utilization
- Unified theory: feasibility of large networks under different applications

Holy Grail...









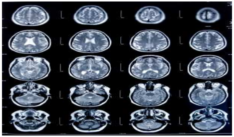
Networks

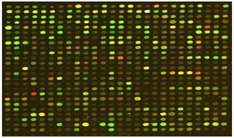
- Seamless operation
- Efficient resource utilization
- Unified theory: feasibility of large networks under different applications

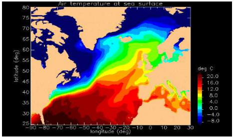
Network Data

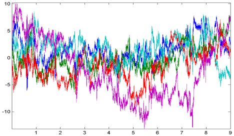
- Data-centric paradigms
- Unifying computation and communication.

  e.g., inference
- Fundamental limits and scalable algorithms

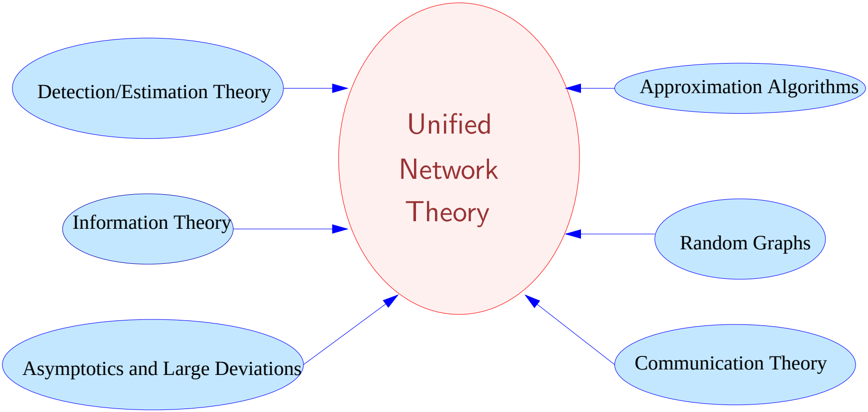






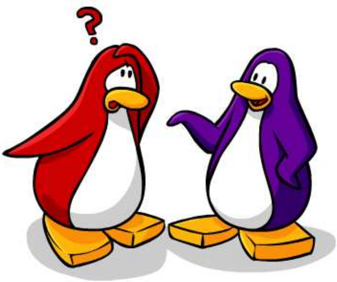


Multidisciplinary Research



http://acsp.ece.cornell.edu/members/anima.html

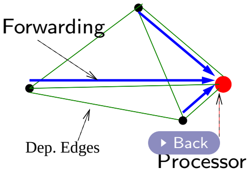
Thank You!



Appendix

Illustration of Stabilization: 1-NNG

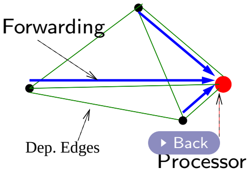
For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



Back

Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



▸ Back

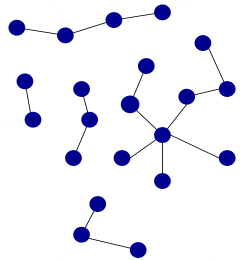
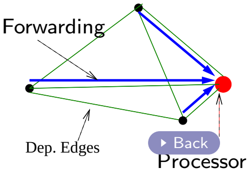


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



▸ Back

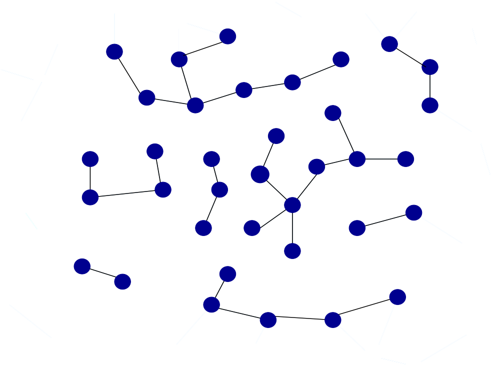
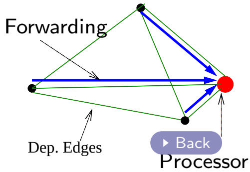


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



Back

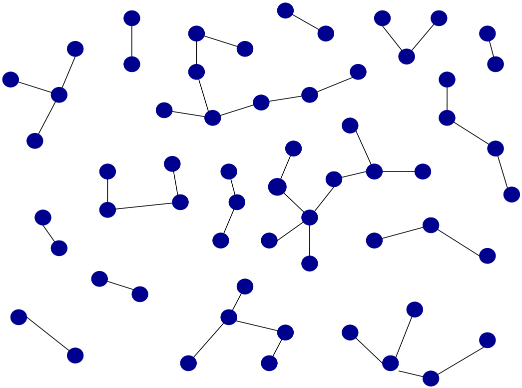
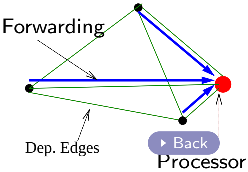


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



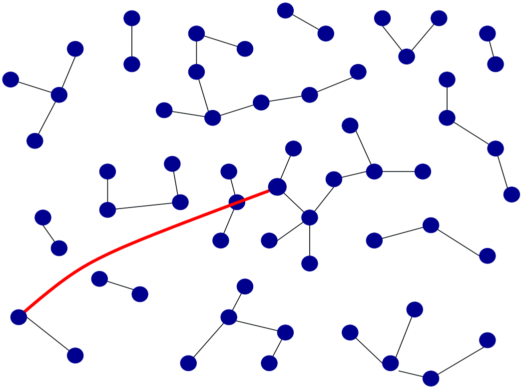
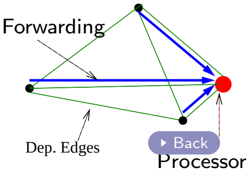


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



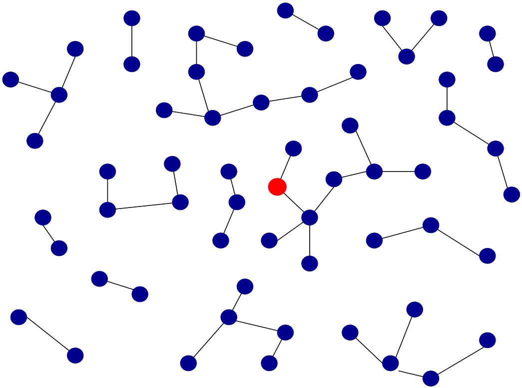
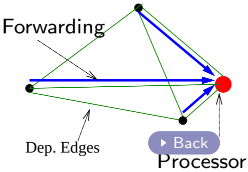


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



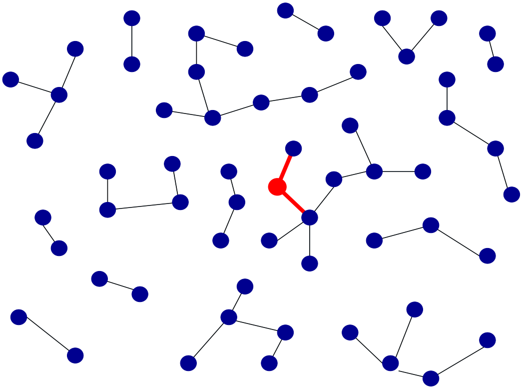
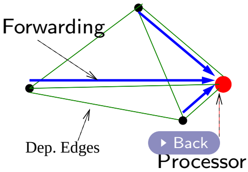


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



▸ Back

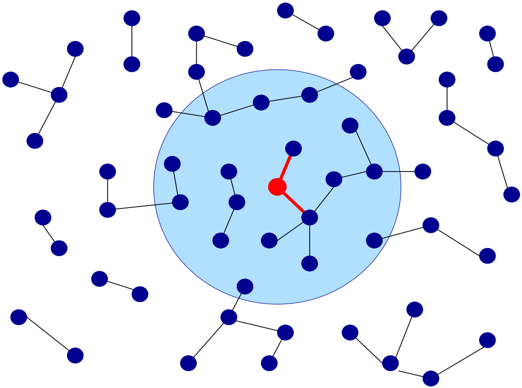
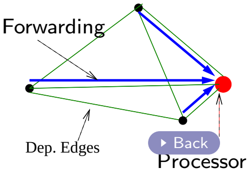


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



▸ Back

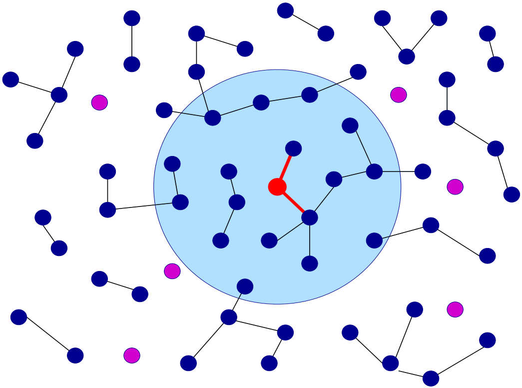
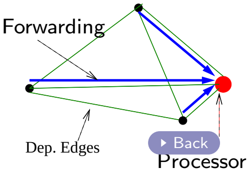


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



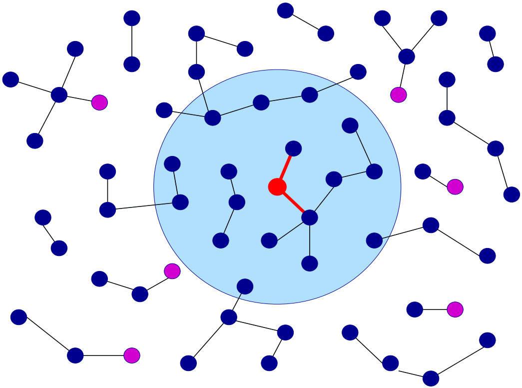
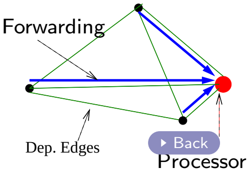


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



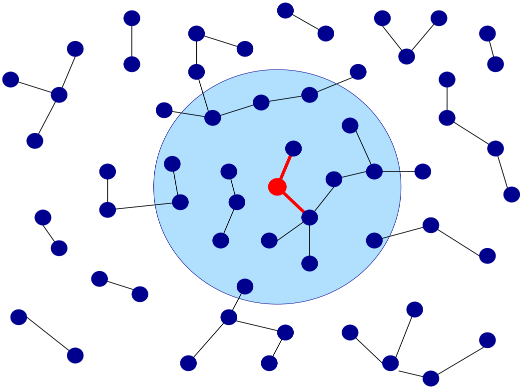
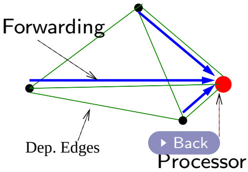


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs, computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication



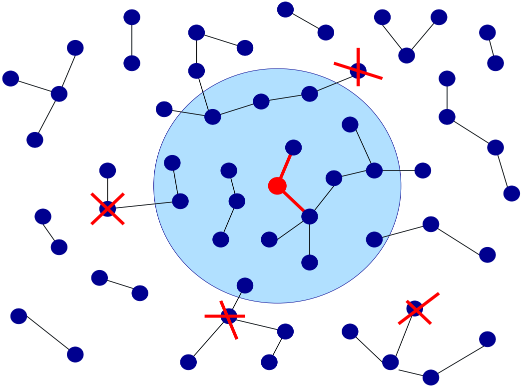
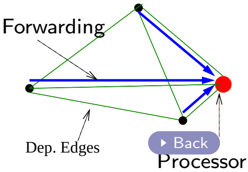
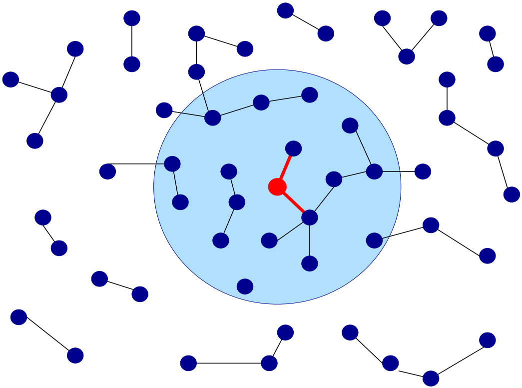


Illustration of Stabilization: 1-NNG

For stabilizing dependency graphs,
computation of clique potentials $\phi_c(\mathbf{Y}_c)$ does not require long distance communication





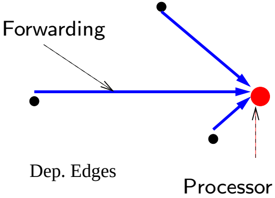
Key ideas

Bound on Forwarding

$$\mathcal{E}(\text{Forward}) = \sum_{c \in \mathcal{C}(\mathbf{V})} \sum_{i \subset c} \text{SP}(i, \text{Proc}(c))$$

$$\leq u \sum_{c \in \mathcal{C}(\mathbf{V})} \sum_{i \subset c} \underbrace{|i, \text{Proc}(c)|^{\nu}}_{\text{Direct Tx.}} \leq u \sum_{e \in \mathcal{G}} |e|^{\nu}$$

In Each Clique



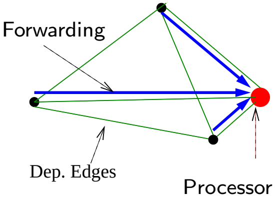
Key ideas

Bound on Forwarding

$$\mathcal{E}(\text{Forward}) = \sum_{c \in \mathcal{C}(\mathbf{V})} \sum_{i \subset c} \text{SP}(i, \text{Proc}(c))$$

$$\leq u \sum_{c \in \mathcal{C}(\mathbf{V})} \sum_{i \subset c} \underbrace{|i, \text{Proc}(c)|^{\nu}}_{\text{Direct Tx.}} \leq u \sum_{e \in \mathcal{G}} |e|^{\nu}$$

In Each Clique



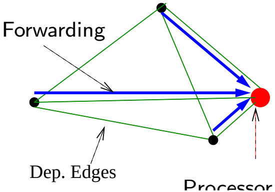
Key ideas

Bound on Forwarding

$$\mathcal{E}(\text{Forward}) = \sum_{c \in \mathcal{C}(\mathbf{V})} \sum_{i \subset c} \text{SP}(i, \text{Proc}(c))$$

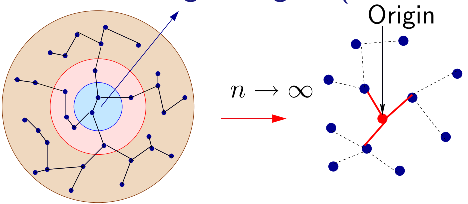
$$\leq u \sum_{c \in \mathcal{C}(\mathbf{V})} \sum_{i \subset c} \underbrace{|i, \text{Proc}(c)|^{\nu}}_{\text{Direct Tx.}} \leq u \sum_{e \in \mathcal{G}} |e|^{\nu}$$

In Each Clique



LLN for Normalized Sum of Edge Weights (Penrose-Yukich)

Back



$$\frac{1}{n} \sum_{e \in \mathcal{G}(\mathbf{V}_n)} |e|^{\nu} \longrightarrow \frac{1}{2}\mathbb{E}[\sum_{(\mathbf{0},j) \in \mathcal{G}(\mathcal{P}_1 \cup \mathbf{0})} |\mathbf{0}, j|^{\nu}] \int_{Q_1} \kappa(x)^{1-\frac{\nu}{2}} dx$$

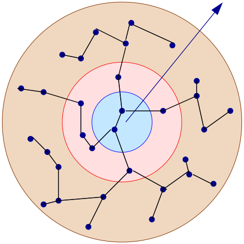
of origin of Poisson process

Scaling Constant via Poissonization

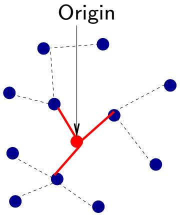
$$\frac{1}{n}\sum_{e\in \mathsf{MST}_n}|e|^\nu \quad \xrightarrow{L^2} \quad \bar{\mathcal{E}}_\infty^{\mathsf{MST}}$$

$$\bar{\mathcal{E}}_\infty^{\mathsf{MST}}(\kappa) = \zeta(\nu;\mathsf{MST})\int_{Q_1}\kappa(x)^{1-\frac{\nu}{2}}dx,$$

$$\zeta(\nu;\mathsf{MST}) = \frac{1}{2}\mathbb{E}\left\{\sum_{(\mathbf{0},j)\in\mathsf{MST}(\mathcal{P}_1\cup\mathbf{0})}|\mathbf{0},j|^\nu\right\}$$



$n\to\infty$



Back IID

Back MRF

Optimal Fusion: Lower Bound

Recall $\mathfrak{F}_{\mathcal{G}} \overset{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y_V})$ computable at the fusion center$\}$

$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

Optimal Fusion: Lower Bound

Recall $\mathfrak{F}_{\mathcal{G}} \stackrel{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y_V})$ computable at the fusion center$\}$

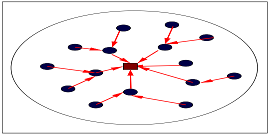
$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

Lower Bound

For any dependency graph $\mathcal{G}$

$$\frac{1}{n}\mathcal{E}(\pi_n^*) \geq \frac{1}{n}\mathcal{E}(\pi_n^{\mathsf{MST}}) \stackrel{L^2}{\rightarrow} \zeta(\nu; \mathsf{MST}) \int_{Q_1} \kappa(x)^{1-\frac{\nu}{2}} dx$$

- Each node must transmit at least once.
- The fusion graph needs to be connected.



Lower bound is tight (achieved for independent data).

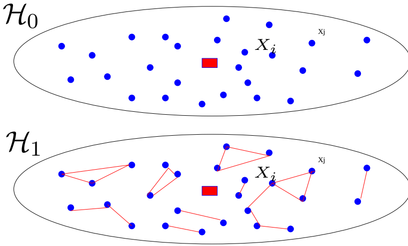
Back

Example: Gauss-Markov random field

- Test on GMRF:

$$\mathcal{H}_0 \quad : \quad X_V \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I})$$
$$\mathcal{H}_1 \quad : \quad X_V \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$$

- Nearest neighbor graph.

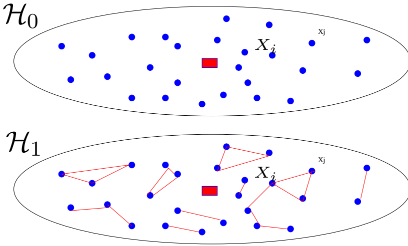


Example: Gauss-Markov random field

- Test on GMRF:

$$\mathcal{H}_0 \quad : \quad X_V \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I})$$
$$\mathcal{H}_1 \quad : \quad X_V \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$$

- Nearest neighbor graph.



- Tradeoff between exploiting signal strength and exploiting correlation:

$$K = \frac{\sigma_1^2}{\sigma_0^2} \quad \text{vs.} \quad g(R_{ij}) \stackrel{\Delta}{=} \frac{\mathbf{\Sigma}(i,j)}{\sigma_1^2}$$

where $\mathbf{\Sigma}[i,i] = \sigma_1^2$ and $g(\cdot)$ a decreasing function.

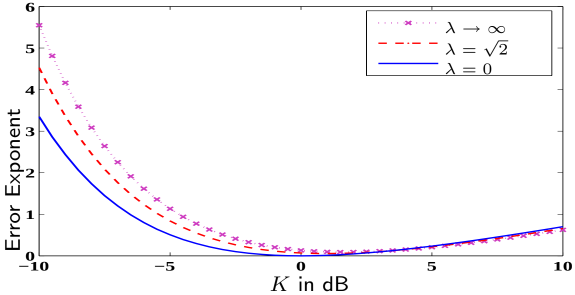
- Sparse deployment: independent samples, costly data fusion.
- Dense deployment: correlated samples, require less energy.

Error exponent behavior

- Closed-form error exponent

$$
\begin{aligned}
-\lim_{n\to\infty} \log P_M(n) &= \mathcal{D}(\lambda, K; g) \\
&= \frac{1}{2}\mathbb{E}_\lambda\, h\big(Z\lambda^{-0.5}, K; g\big) + \mathcal{D}_{\text{IID}}(K)
\end{aligned}
$$

- The error exponent reverse its behavior at a threshold K_τ.



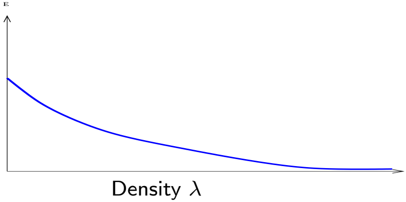
Design for Energy Constrained Inference

- Energy constrained network for inference

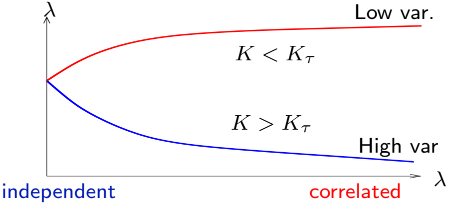
$$\lambda_* \stackrel{\Delta}{=} \arg\max_{\lambda>0} \mathcal{D}(\lambda, K; g) \quad \text{subject to } \bar{E} \leq \bar{E}_{\text{max}}$$

- Energy and performance scaling laws: $K \stackrel{\Delta}{=} \frac{\sigma_1^2}{\sigma_0^2}$

Energy vs. λ

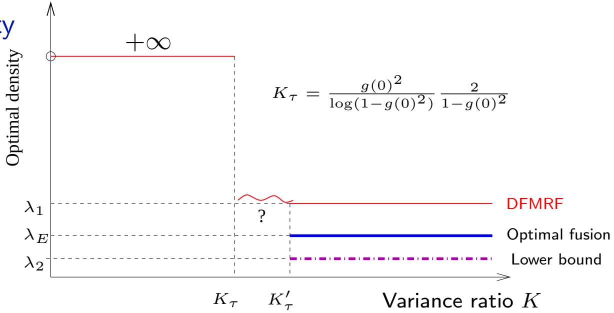


Exponent vs. λ



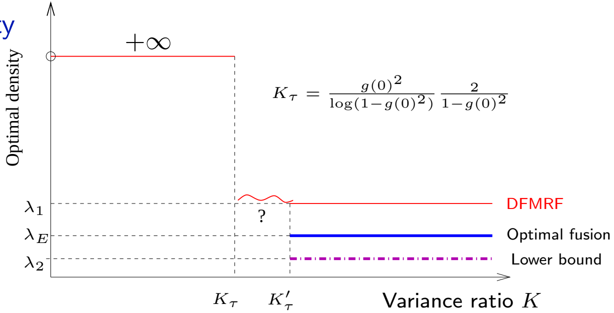
Design for Energy Constrained Inference

Optimal density



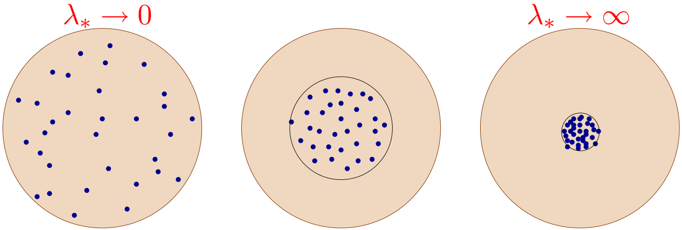
Design for Energy Constrained Inference

Optimal density

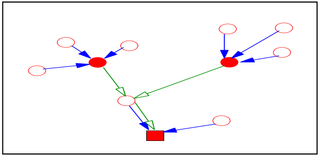


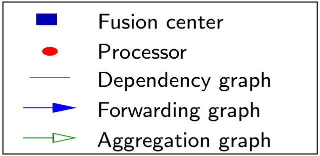
Deployment implications

Back



Stages of LLR Computation: $L_{\mathcal{G}}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$

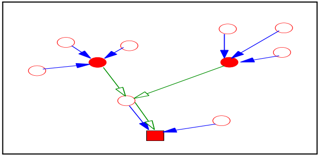


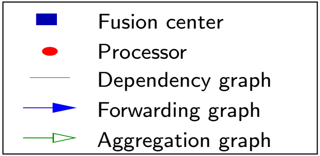


Recall $\mathfrak{F}_{\mathcal{G}} \stackrel{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}})$ computable at the fusion center$\}$

$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

Stages of LLR Computation: $L_{\mathcal{G}}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



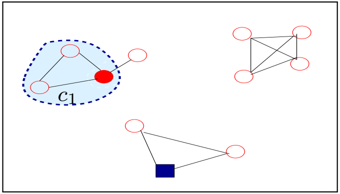


Recall $\mathfrak{F}_{\mathcal{G}} \stackrel{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}})$ computable at the fusion center$\}$

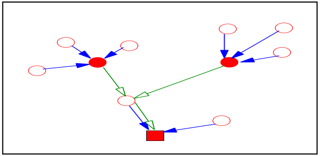
$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

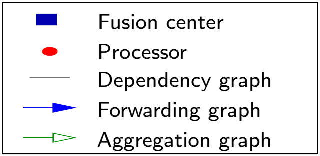
Local Computation of Clique Potentials: Processor is a Clique Member

- Simplifies optimization problem
- Local knowledge of function parameters



Stages of LLR Computation: $L_{\mathcal{G}}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



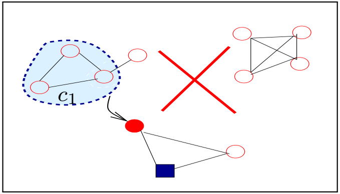


Recall $\mathfrak{F}_{\mathcal{G}} \stackrel{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}})$ computable at the fusion center$\}$

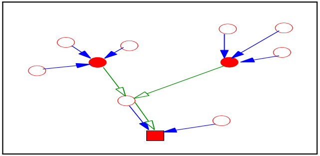
$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

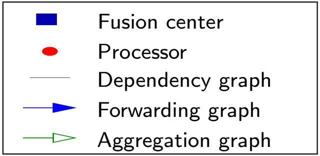
Local Computation of Clique Potentials: Processor is a Clique Member

- Simplifies optimization problem
- Local knowledge of function parameters



Stages of LLR Computation: $L_{\mathcal{G}}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



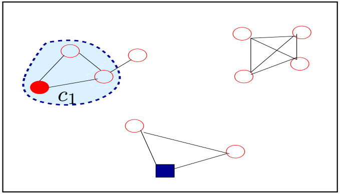


Recall $\mathfrak{F}_{\mathcal{G}} \stackrel{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}})$ computable at the fusion center$\}$

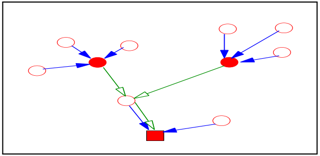
$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

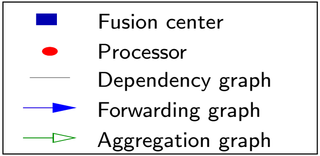
Local Computation of Clique Potentials: Processor is a Clique Member

- Simplifies optimization problem
- Local knowledge of function parameters



Stages of LLR Computation: $L_{\mathcal{G}}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



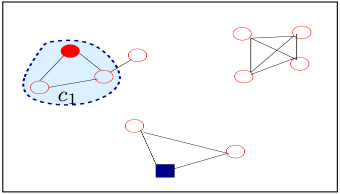


Recall $\mathfrak{F}_{\mathcal{G}} \stackrel{\Delta}{=} \{\pi : L_{\mathcal{G}}(\mathbf{Y}_{\mathbf{V}})$ computable at the fusion center$\}$

$$\mathcal{E}(\pi_n^*) = \min_{\pi \in \mathfrak{F}_{\mathcal{G}}} \sum_i \mathcal{E}_i(\pi_n)$$

Local Computation of Clique Potentials: Processor is a Clique Member

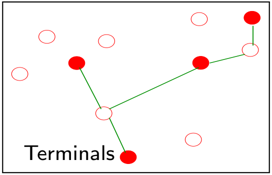
- Simplifies optimization problem
- Local knowledge of function parameters



Steiner-Tree Reduction

Steiner Tree

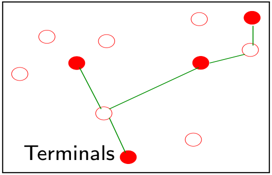
- Minimum cost tree containing a required set of nodes called terminals
- NP-hard problem, currently the best approximation is 1.55



Steiner-Tree Reduction

Steiner Tree

- Minimum cost tree containing a required set of nodes called terminals
- NP-hard problem, currently the best approximation is 1.55



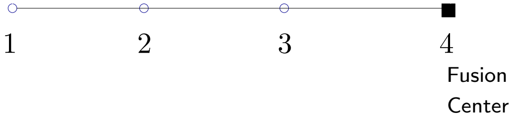
Main result

Min cost fusion has approx. ratio preserving Steiner tree reduction

Implications

- Any approximation for Steiner tree has same ratio for fusion
- Best approximation for min cost fusion: 1.55

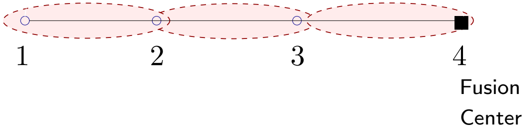
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

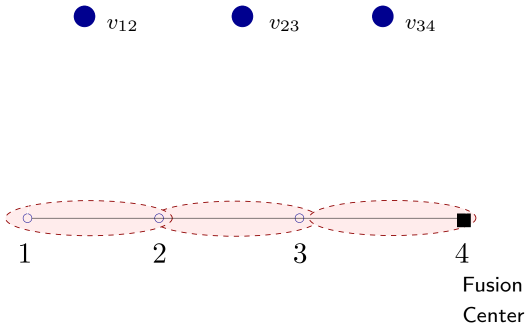
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

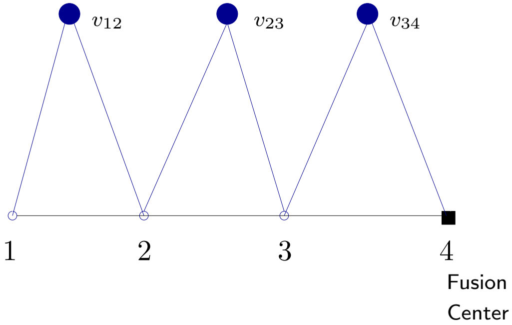
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

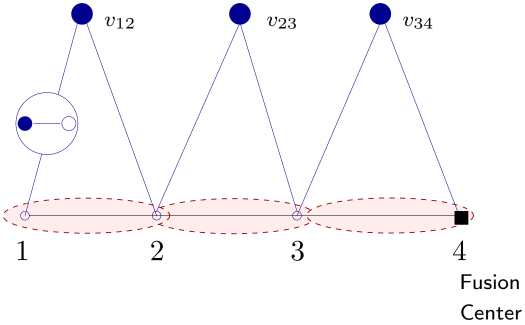
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

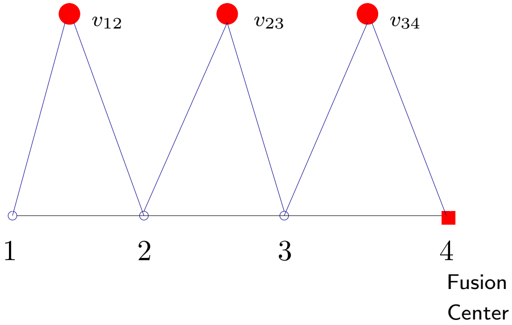
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

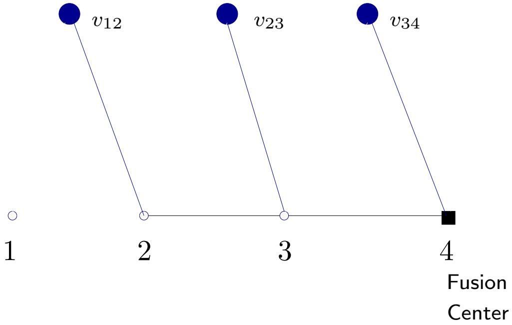
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

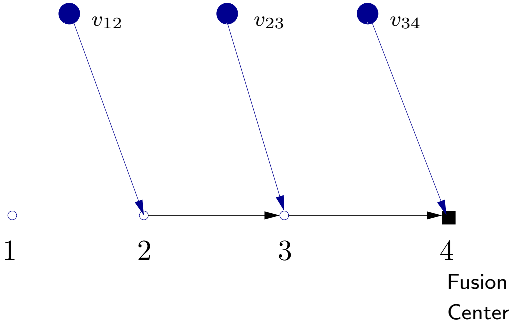
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

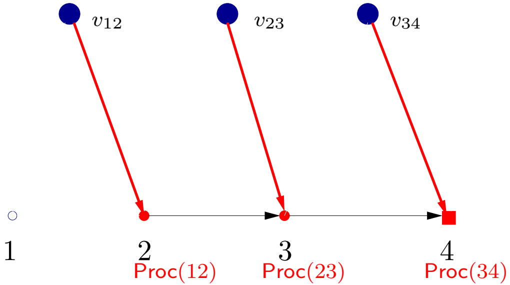
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

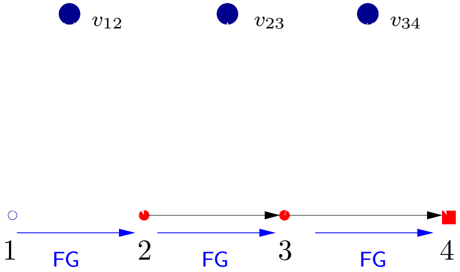
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

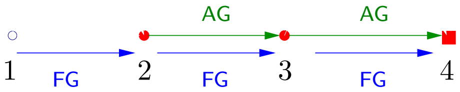
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

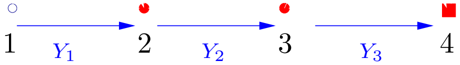
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

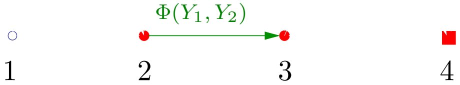
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

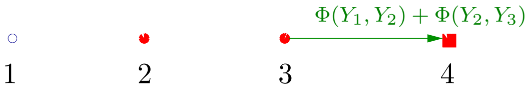
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

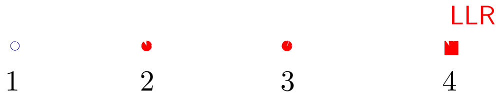
Example : Chain dependency graph



Graph transformation and building Steiner tree.

Back

Example : Chain dependency graph



Graph transformation and building Steiner tree.

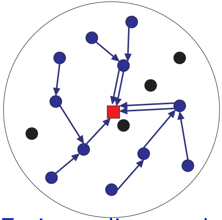
Back

Optimal Cost-Performance Tradeoff

Problem Statement

- Select $V_s \subset V$ and design a fusion scheme $\Gamma(V_s)$.
- Minimize the total routing costs $\mathcal{C}(\Gamma(V_s))$ plus a penalty π based on the error prob. $P_M(V_s)$.

$$\pi(V \backslash V_s) \stackrel{\Delta}{=} \log \frac{P_M(V_s)}{P_M(V)} > 0$$

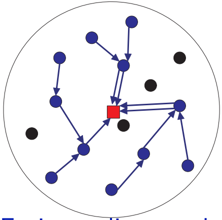


Fusion policy graph

Optimal Cost-Performance Tradeoff

Problem Statement

- Select $V_s \subset V$ and design a fusion scheme $\Gamma(V_s)$.
- Minimize the total routing costs $\mathcal{C}(\Gamma(V_s))$ plus a penalty π based on the error prob. $P_M(V_s)$.



Fusion policy graph

$$\pi(V \backslash V_s) \stackrel{\Delta}{=} \log \frac{P_M(V_s)}{P_M(V)} > 0$$

$$\min_{V_s \subset V, \Gamma(V_s)} \Big[\mathcal{C}(\Gamma(V_s)) + \mu \pi(V \backslash V_s) \Big], \ \mu > 0$$

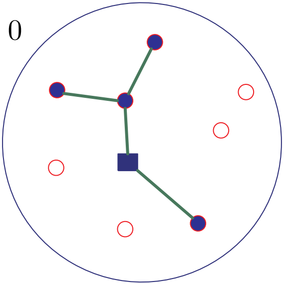
Prize-Collecting Data Fusion

Main Results

$$\min_{V_s \subset V, \Gamma(V_s)} \left[\mathcal{C}(\Gamma(V_s)) + \mu \log \frac{P_M(V_s)}{P_M(V)}) \right], \ \mu > 0$$

IID measurements

$2 - (|V| - 1)^{-1}$ approximation via
Prize-Collecting Steiner Tree



PCST

Correlated data: component and clique selection heuristics

- Provable approximation guarantee for special dependency graphs.
- Substantially better than no data fusion.
- Performance under different node placements.

Back

PCDF: IID case

$$\min_{V_s \subset V, \Gamma(V_s)} \left[\mathcal{C}(\Gamma(V_s)) + \mu \log \frac{P_M(V_s)}{P_M(V)} \right], \ \mu > 0$$

Simplifications of IID measurements

- $\mathcal{H}_k : \mathbf{Y_V} \sim \prod_{i \in \mathbf{V}} f_k(Y_i)$
- $L_{\mathcal{G}}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} \log \frac{f(Y_i; \mathcal{H}_0)}{f(Y_i; \mathcal{H}_1)} = \sum_{i \in V_s} L_{\mathcal{G}}(\mathbf{Y}_i)$
- Error exponent $\mathcal{D} = D(f(Y; \mathcal{H}_0) || f(Y; \mathcal{H}_1))$

PCDF: IID case

$$\min_{V_s \subset V, \Gamma(V_s)} \Big[\mathcal{C}(\Gamma(V_s)) + \mu \log \frac{P_M(V_s)}{P_M(V)}\Big], \ \mu > 0$$

Simplifications of IID measurements

- $\mathcal{H}_k : \mathbf{Y_V} \sim \prod_{i \in \mathbf{V}} f_k(Y_i)$
- $L_{\mathcal{G}}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} \log \frac{f(Y_i;\mathcal{H}_0)}{f(Y_i;\mathcal{H}_1)} = \sum_{i \in V_s} L_{\mathcal{G}}(\mathbf{Y}_i)$
- Error exponent $\mathcal{D} = D(f(Y;\mathcal{H}_0)||f(Y;\mathcal{H}_1))$

Modified cost-performance tradeoff for IID

$$\min_{V_s \subset V, \Gamma(V_s)} \Big[\mathcal{C}(\Gamma(V_s)) + \mu[|V| - |V_s|]D\Big]$$

- Asymptotic convergence to the original problem.
- The optimal solution is the Prize Collecting Steiner Tree.

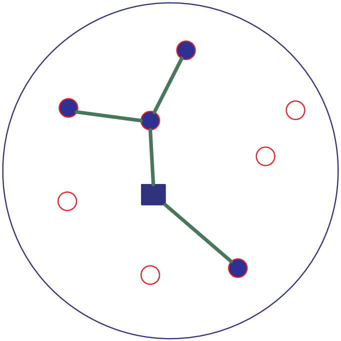
Prize Collecting Steiner Tree (PCST)

Definition

- Tree with minimum sum edge costs plus node penalties not spanned

$$T_* = \arg \min_{T=(V',E')} \Big[\sum_{e \in E'} c_e + \sum_{i \notin V'} \pi_i \Big].$$

- NP-hard, Goemans-Williamson algorithm has approx. ratio of $2 - \frac{1}{|V|-1}$



Approx. PCST

Back

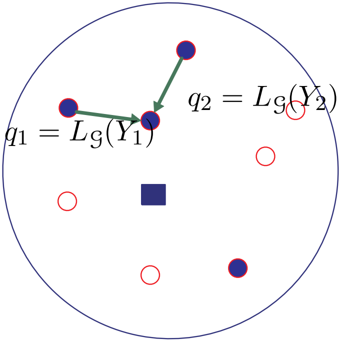
Prize Collecting Steiner Tree (PCST)

Definition

- Tree with minimum sum edge costs plus node penalties not spanned

$$T_* = \arg \min_{T=(V',E')} \Big[\sum_{e \in E'} c_e + \sum_{i \notin V'} \pi_i \Big].$$

- NP-hard, Goemans-Williamson algorithm has approx. ratio of $2 - \frac{1}{|V|-1}$



Fusion of IID measurements

$$L_{\mathcal{G}}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} L_{\mathcal{G}}(\mathbf{Y}_i)$$

Back

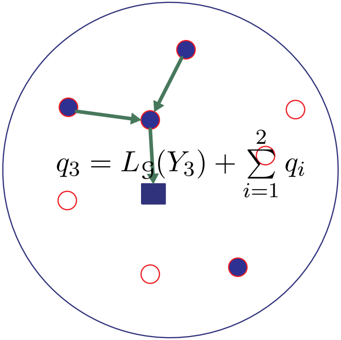
Prize Collecting Steiner Tree (PCST)

Definition

- Tree with minimum sum edge costs plus node penalties not spanned

$$T_* = \arg \min_{T=(V',E')} \Big[\sum_{e \in E'} c_e + \sum_{i \notin V'} \pi_i \Big].$$

- NP-hard, Goemans-Williamson algorithm has approx. ratio of $2 - \frac{1}{|V|-1}$



Fusion of IID measurements

$$L_{\mathcal{G}}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} L_{\mathcal{G}}(\mathbf{Y}_i)$$

Back

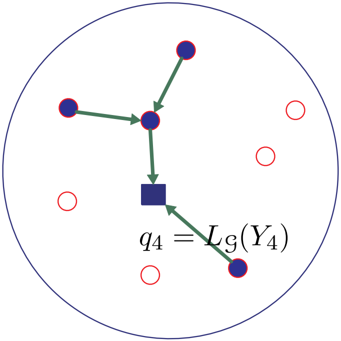
Prize Collecting Steiner Tree (PCST)

Definition

- Tree with minimum sum edge costs plus node penalties not spanned

$$T_* = \arg \min_{T=(V',E')} \Big[\sum_{e \in E'} c_e + \sum_{i \notin V'} \pi_i \Big].$$

- NP-hard, Goemans-Williamson algorithm has approx. ratio of $2 - \frac{1}{|V|-1}$

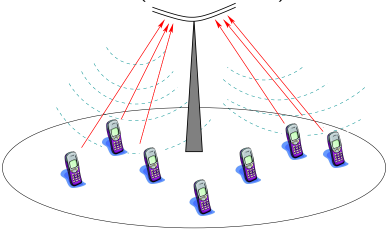


Fusion of IID measurements

$$L_{\mathcal{G}}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} L_{\mathcal{G}}(\mathbf{Y}_i)$$

Back

Medium Access Control (MAC) For Inference



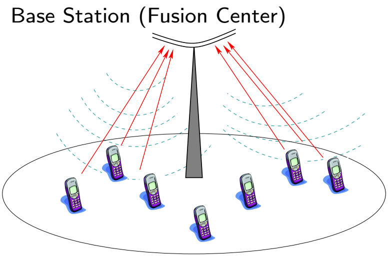
Design of MAC (Single Hop)

Back

(1) A. Anandkumar and L. Tong, "Type-based Random Access for Distributed Detection over Multi-access Fading Channels," *IEEE Tran. on Signal Processing*, vol.55, no.10, pp.5032-5043, Oct. 2007 *(2008 IEEE SPS Young Author Best Paper Award)*

(2) A. Anandkumar, L. Tong and A. Swami, " Distributed Estimation via Random Access," in *IEEE Tran. on Information Theory*, vol. 54, pp. 3175-3181, July 2008.

Medium Access Control (MAC) For Inference



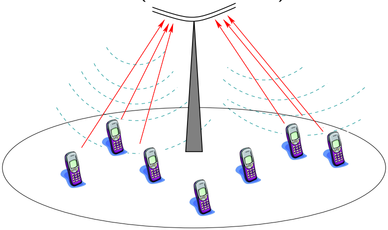
Design of MAC (Single Hop)

Back

(1) A. Anandkumar and L. Tong,"Type-based Random Access for Distributed Detection over Multi-access Fading Channels," *IEEE Tran. on Signal Processing*, vol.55, no.10, pp.5032-5043, Oct. 2007 *(2008 IEEE SPS Young Author Best Paper Award)*

(2) A. Anandkumar, L. Tong and A. Swami," Distributed Estimation via Random Access," in *IEEE Tran. on Information Theory*, vol. 54, pp. 3175-3181, July 2008.

Medium Access Control (MAC) For Inference



Design of MAC (Single Hop)

Back

Classical Design

Orthogonal Division

Proposed Design

Type-Based Random Access

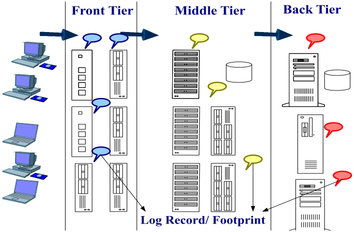
- Sensor encoding based on data level
- Optimal spatio-temporal allocation based on channel conditions

(1) A. Anandkumar and L. Tong,"Type-based Random Access for Distributed Detection over Multi-access Fading Channels," *IEEE Tran. on Signal Processing*, vol.55, no.10, pp.5032-5043, Oct. 2007 *(2008 IEEE SPS Young Author Best Paper Award)*

(2) A. Anandkumar, L. Tong and A. Swami," Distributed Estimation via Random Access," in *IEEE Tran. on Information Theory*, vol. 54, pp. 3175-3181, July 2008.

Inference of Transaction Paths in Distributed Systems

Transactions & Log Records



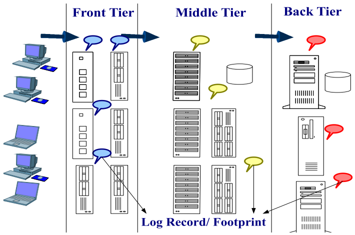
Back

(1) A. Anandkumar, C. Bisdikian, and D. Agrawal, Tracking in a Spaghetti Bowl: Monitoring Transactions Using Footprints, in *Proc. ACM Intl. Conf. on Measurement & Modeling of Computer Systems* (Sigmetrics), June 2008

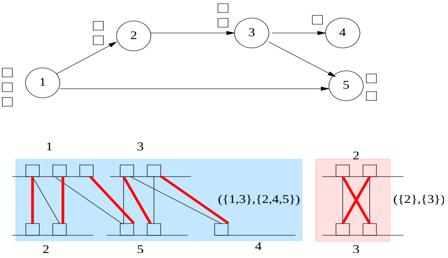
(2) A. Anandkumar, C. Bisdikian, T. He, and D. Agrawal, Designing A Fine Tooth Comb Frugally: Selectively Retrofitting Monitoring in Distributed Systems. *Workshop on Mathematical Performance Modeling and Analysis* June, 2009.

Inference of Transaction Paths in Distributed Systems

Transactions & Log Records



State Transition Model



Maximum Likelihood Tracking $\equiv$ Series of Bipartite Matches

Back

(1) A. Anandkumar, C. Bisdikian, and D. Agrawal, Tracking in a Spaghetti Bowl: Monitoring Transactions Using Footprints, in *Proc. ACM Intl. Conf. on Measurement & Modeling of Computer Systems* (Sigmetrics), June 2008

(2) A. Anandkumar, C. Bisdikian, T. He, and D. Agrawal, Designing A Fine Tooth Comb Frugally: Selectively Retrofitting Monitoring in Distributed Systems. *Workshop on Mathematical Performance Modeling and Analysis* June, 2009.