Minimum Cost Data Aggregation with Localized Processing for Statistical Inference

A. Anandkumar[1] L. Tong[1] A. Swami[2] A. Ephremides[3]

[1]ECE Dept., Cornell University, Ithaca, NY 14853

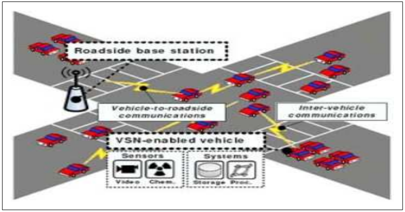
[2]Army Research Laboratory, Adelphi MD 20783

[3]EE Dept., University of Maryland College Park, MD 20742

IEEE INFOCOM 2008

Supported by Army Research Laboratory CTA

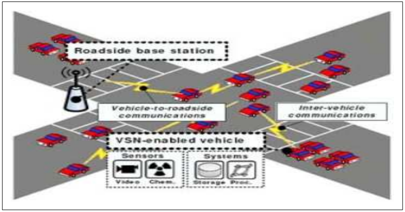
Distributed Statistical Inference



Sensor Network Applications

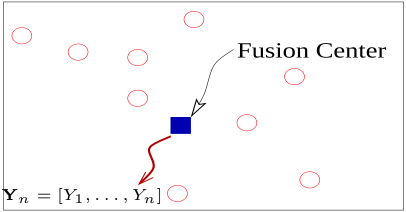
- Detection
- Estimation

Distributed Statistical Inference



Sensor Network Applications

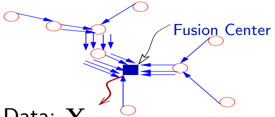
- Detection
- Estimation



Classical Distributed Inference

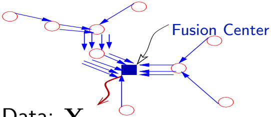
- Sensors: take measurements
- Fusion Center: Final decision
- Statistical Model

Routing for Inference

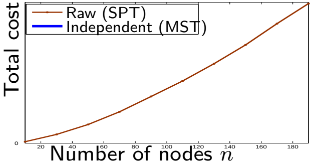


Raw Data: $\mathbf{Y}_n$

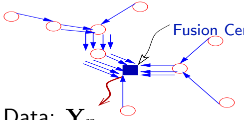
Routing for Inference



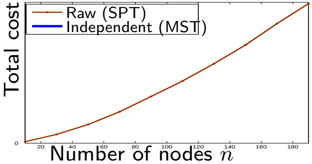
Raw Data: $\mathbf{Y}_n$

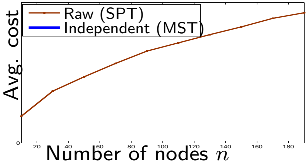


Routing for Inference

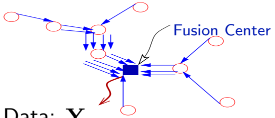


Raw Data: $\mathbf{Y}_n$

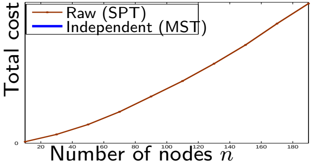


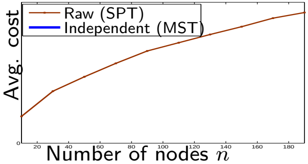


Routing for Inference



Raw Data: $\mathbf{Y}_n$

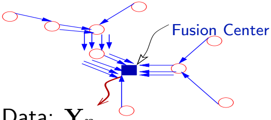




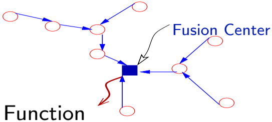
Sufficient Statistics for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

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

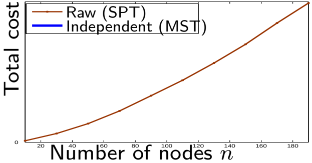
Routing for Inference

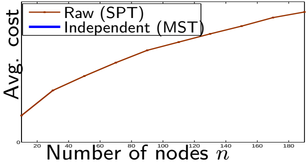


Raw Data: $\mathbf{Y}_n$



Sum Function

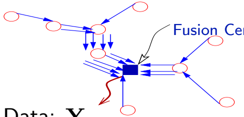




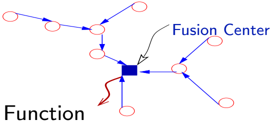
Sufficient Statistics for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

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

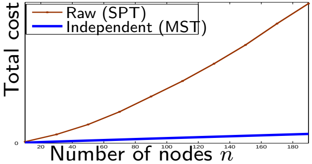
Routing for Inference

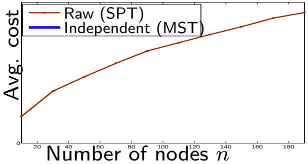


Raw Data: $\mathbf{Y}_n$



Sum Function

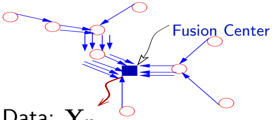




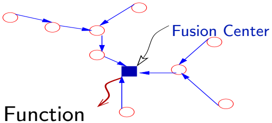
Sufficient Statistics for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

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

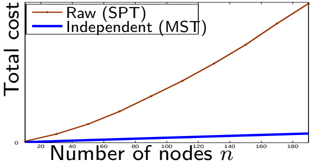
Routing for Inference

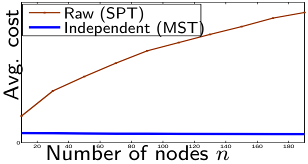


Raw Data: $\mathbf{Y}_n$



Sum Function

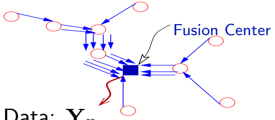




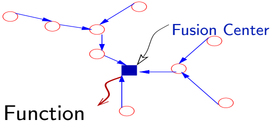
Sufficient Statistics for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

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

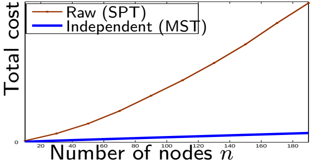
Routing for Inference

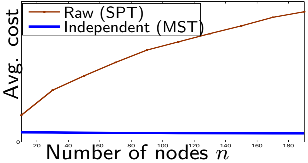


Raw Data: $\mathbf{Y}_n$



Sum Function



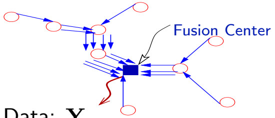


Sufficient Statistics for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

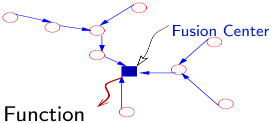
$\sum_i Y_i$ sufficient to estimate θ: no performance loss

Binary Hypothesis Test: $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} f(y; \mathcal{H}_0)$ or $f(y; \mathcal{H}_1)$

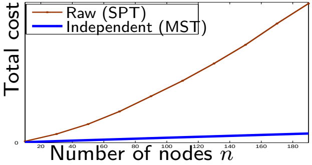
Routing for Inference

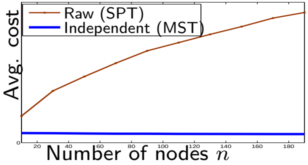


Raw Data: $\mathbf{Y}_n$



Sum Function





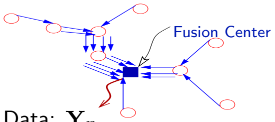
Sufficient Statistics for Mean Estimation $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, 1)$

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

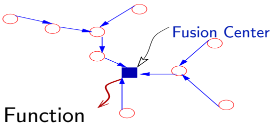
Binary Hypothesis Test: $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} f(y; \mathcal{H}_0)$ or $f(y; \mathcal{H}_1)$

- $[\sum_i \log f(Y_i; \mathcal{H}_0), \sum_i \log f(Y_i; \mathcal{H}_1)]$ sufficient to decide hypothesis

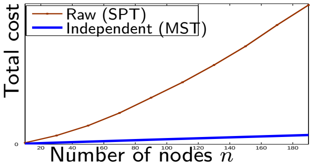
Routing for Inference

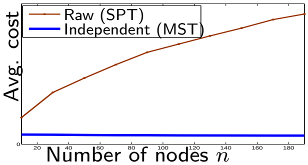


Raw Data: $\mathbf{Y}_n$



Sum Function





Sufficient Statistics 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

Binary Hypothesis Test: $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} f(y; \mathcal{H}_0)$ or $f(y; \mathcal{H}_1)$

- $[\sum_i \log f(Y_i; \mathcal{H}_0), \sum_i \log f(Y_i; \mathcal{H}_1)]$ sufficient to decide hypothesis
- LLR$= \sum_i \frac{\log f(Y_i;\mathcal{H}_1)}{\log f(Y_i;\mathcal{H}_1)}$ minimally sufficient to decide hypothesis

Minimum Cost In-Network Processing for Inference

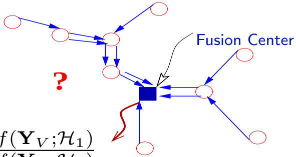
Minimal Sufficient Statistic for Binary Hypothesis Testing

Log Likelihood Ratio: $\mathrm{LLR}(\mathbf{Y}_V) = \log \dfrac{f(\mathbf{Y}_V; \mathcal{H}_1)}{f(\mathbf{Y}_V; \mathcal{H}_0)}$

Minimum Cost In-Network Processing for Inference

Minimal Sufficient Statistic for Binary Hypothesis Testing

Log Likelihood Ratio: $\text{LLR}(\mathbf{Y}_V) = \log \frac{f(\mathbf{Y}_V; \mathcal{H}_1)}{f(\mathbf{Y}_V; \mathcal{H}_0)}$



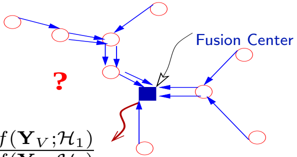
$\text{LLR}(\mathbf{Y}_V) = \frac{\log f(\mathbf{Y}_V; \mathcal{H}_1)}{\log f(\mathbf{Y}_V; \mathcal{H}_0)}$

Extent of Processing?

Minimum Cost In-Network Processing for Inference

Minimal Sufficient Statistic for Binary Hypothesis Testing

Log Likelihood Ratio: $\text{LLR}(\mathbf{Y}_V) = \log \frac{f(\mathbf{Y}_V; \mathcal{H}_1)}{f(\mathbf{Y}_V; \mathcal{H}_0)}$



$\text{LLR}(\mathbf{Y}_V) = \frac{\log f(\mathbf{Y}_V; \mathcal{H}_1)}{\log f(\mathbf{Y}_V; \mathcal{H}_0)}$

Extent of Processing?

Fusion Scheme?

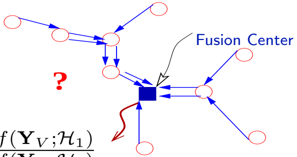
Minimum Cost Data Fusion for Inference

Min total costs s.t. LLR is delivered to fusion center

Minimum Cost In-Network Processing for Inference

Minimal Sufficient Statistic for Binary Hypothesis Testing

Log Likelihood Ratio: $\text{LLR}(\mathbf{Y}_V) = \log \frac{f(\mathbf{Y}_V; \mathcal{H}_1)}{f(\mathbf{Y}_V; \mathcal{H}_0)}$



$\text{LLR}(\mathbf{Y}_V) = \frac{\log f(\mathbf{Y}_V; \mathcal{H}_1)}{\log f(\mathbf{Y}_V; \mathcal{H}_0)}$

Extent of Processing?

Fusion Scheme?

Minimum Cost Data Fusion for Inference

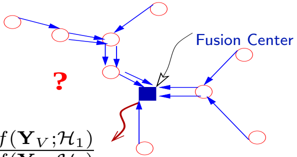
Min total costs s.t. LLR is delivered to fusion center

Spatial Correlation Model: Should Capture Full Correlation Range

Minimum Cost In-Network Processing for Inference

Minimal Sufficient Statistic for Binary Hypothesis Testing

$$\text{Log Likelihood Ratio: } \mathsf{LLR}(\mathbf{Y}_V) = \log \frac{f(\mathbf{Y}_V; \mathcal{H}_1)}{f(\mathbf{Y}_V; \mathcal{H}_0)}$$



$$\mathsf{LLR}(\mathbf{Y}_V) = \frac{\log f(\mathbf{Y}_V; \mathcal{H}_1)}{\log f(\mathbf{Y}_V; \mathcal{H}_0)}$$

Extent of Processing?

Fusion Scheme?

Minimum Cost Data Fusion for Inference

Min total costs s.t. LLR is delivered to fusion center

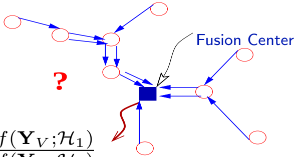
Spatial Correlation Model: Should Capture Full Correlation Range

- Markov random field with dependency graph

Minimum Cost In-Network Processing for Inference

Minimal Sufficient Statistic for Binary Hypothesis Testing

Log Likelihood Ratio: $\text{LLR}(\mathbf{Y}_V) = \log \frac{f(\mathbf{Y}_V; \mathcal{H}_1)}{f(\mathbf{Y}_V; \mathcal{H}_0)}$



$\text{LLR}(\mathbf{Y}_V) = \frac{\log f(\mathbf{Y}_V; \mathcal{H}_1)}{\log f(\mathbf{Y}_V; \mathcal{H}_0)}$

Extent of Processing?

Fusion Scheme?

Minimum Cost Data Fusion for Inference

Min total costs s.t. LLR is delivered to fusion center

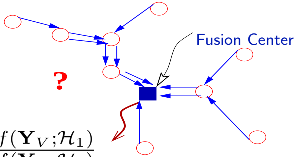
Spatial Correlation Model: Should Capture Full Correlation Range

- Markov random field with dependency graph
- Structured LLR: sum over dependency graph cliques

Minimum Cost In-Network Processing for Inference

Minimal Sufficient Statistic for Binary Hypothesis Testing

Log Likelihood Ratio: $\mathrm{LLR}(\mathbf{Y}_V) = \log \frac{f(\mathbf{Y}_V;\mathcal{H}_1)}{f(\mathbf{Y}_V;\mathcal{H}_0)}$



$\mathrm{LLR}(\mathbf{Y}_V) = \frac{\log f(\mathbf{Y}_V;\mathcal{H}_1)}{\log f(\mathbf{Y}_V;\mathcal{H}_0)}$

Extent of Processing?

Fusion Scheme?

Minimum Cost Data Fusion for Inference

Min total costs s.t. LLR is delivered to fusion center

Spatial Correlation Model: Should Capture Full Correlation Range

- Markov random field with dependency graph
- Structured LLR: sum over dependency graph cliques
- Local processing of clique data

Summary of Results

Minimum Cost Data Fusion for Inference

Min total routing costs s.t. likelihood ratio is delivered to fusion center

Summary of Results

Minimum Cost Data Fusion for Inference

Min total routing costs s.t. likelihood ratio is delivered to fusion center

AggMST: MST-based Heuristic

- Separation of local processor selection and aggregation
- Approximation Ratio of 2 for Nearest-Neighbor Dependency

Summary of Results

Minimum Cost Data Fusion for Inference

Min total routing costs s.t. likelihood ratio is delivered to fusion center

AggMST: MST-based Heuristic

- Separation of local processor selection and aggregation
- Approximation Ratio of 2 for Nearest-Neighbor Dependency

Steiner Tree Reduction

- Joint design of local processors and aggregation
- Optimal Cost is given by Steiner tree on expanded graph
- Approximation-factor preserving reduction: best known is 1.55

Summary of Results

Minimum Cost Data Fusion for Inference

Min total routing costs s.t. likelihood ratio is delivered to fusion center

AggMST: MST-based Heuristic

- Separation of local processor selection and aggregation
- Approximation Ratio of 2 for Nearest-Neighbor Dependency

Steiner Tree Reduction

- Joint design of local processors and aggregation
- Optimal Cost is given by Steiner tree on expanded graph
- Approximation-factor preserving reduction: best known is 1.55

Constant Average Cost Scaling (Allerton '07)

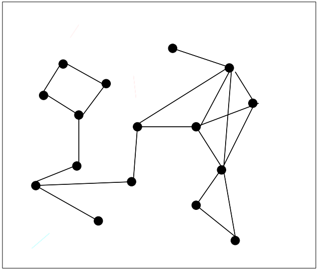
k-NNG Dependency in Random Large Constant Density Networks

Outline

Outline

Markov Random Field (MRF)

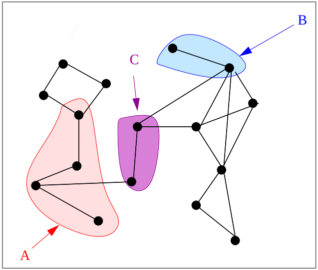
Dependency Graph



Markov Random Field (MRF)

Dependency Graph

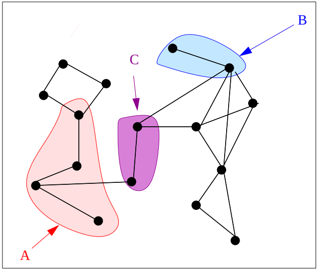
$\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$



Markov Random Field (MRF)

Dependency Graph

$$\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$$



Hammersley-Clifford Theorem '71

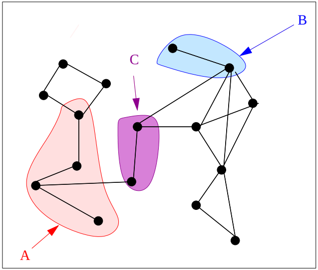
Log-Likelihood: sum of potentials of maximal cliques of dependency graph

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

Markov Random Field (MRF)

Dependency Graph

$\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$



Hammersley-Clifford Theorem '71

Log-Likelihood: sum of potentials of maximal cliques of dependency graph

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

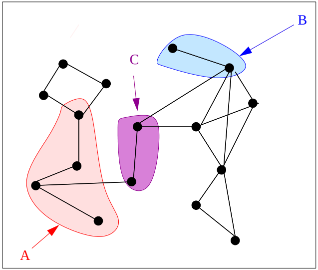
Independent: Cliques=Nodes

$$-\log f(\mathbf{Y}_V; G_d) = \sum_{i \in V} \Psi_i(Y_i)$$

Markov Random Field (MRF)

Dependency Graph

$\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$



Hammersley-Clifford Theorem '71

Log-Likelihood: sum of potentials of maximal cliques of dependency graph

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

Independent: Cliques=Nodes

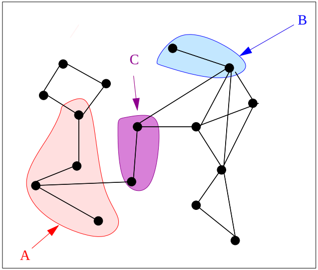
$$-\log f(\mathbf{Y}_V; G_d) = \sum_{i \in V} \Psi_i(Y_i)$$

Chain Dependency Graph

Markov Random Field (MRF)

Dependency Graph

$\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$



Hammersley-Clifford Theorem '71

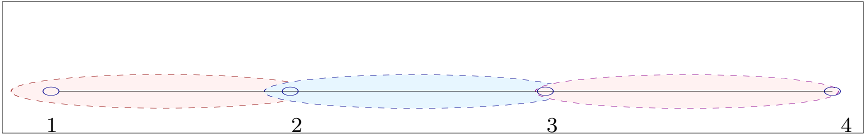
Log-Likelihood: sum of potentials of maximal cliques of dependency graph

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

Independent: Cliques=Nodes

$$-\log f(\mathbf{Y}_V; G_d) = \sum_{i \in V} \Psi_i(Y_i)$$

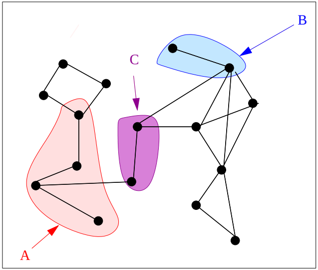
Chain Dependency Graph



Markov Random Field (MRF)

Dependency Graph

$\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$



Hammersley-Clifford Theorem '71

Log-Likelihood: sum of potentials of maximal cliques of dependency graph

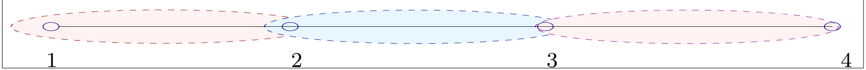
$$-\log f(\mathbf{Y}_V; G_d) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$

Independent: Cliques=Nodes

$$-\log f(\mathbf{Y}_V; G_d) = \sum_{i \in V} \Psi_i(Y_i)$$

Chain Dependency Graph

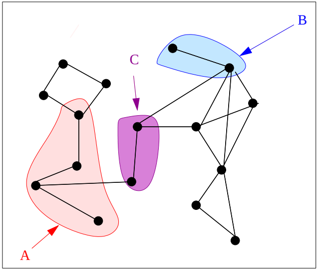
$-\log f(\mathbf{Y}_V; G_d) =$



Markov Random Field (MRF)

Dependency Graph

$\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$



Hammersley-Clifford Theorem '71

Log-Likelihood: sum of potentials of maximal cliques of dependency graph

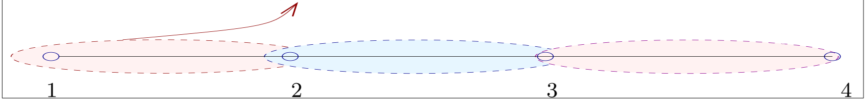
$$-\log f(\mathbf{Y}_V; G_d) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$

Independent: Cliques=Nodes

$$-\log f(\mathbf{Y}_V; G_d) = \sum_{i \in V} \Psi_i(Y_i)$$

Chain Dependency Graph

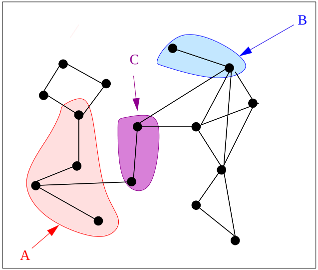
$$-\log f(\mathbf{Y}_V; G_d) = \quad \Psi_{1,2}(Y_1, Y_2)+$$



Markov Random Field (MRF)

Dependency Graph

$\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$



Hammersley-Clifford Theorem ‘71

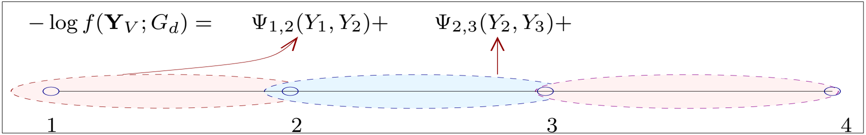
Log-Likelihood: sum of potentials of maximal cliques of dependency graph

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

Independent: Cliques=Nodes

$$-\log f(\mathbf{Y}_V; G_d) = \sum_{i \in V} \Psi_i(Y_i)$$

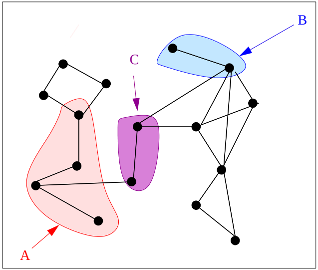
Chain Dependency Graph



Markov Random Field (MRF)

Dependency Graph

$\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$



Hammersley-Clifford Theorem '71

Log-Likelihood: sum of potentials of maximal cliques of dependency graph

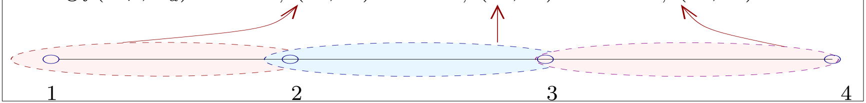
$$-\log f(\mathbf{Y}_V; G_d) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$

Independent: Cliques=Nodes

$$-\log f(\mathbf{Y}_V; G_d) = \sum_{i \in V} \Psi_i(Y_i)$$

Chain Dependency Graph

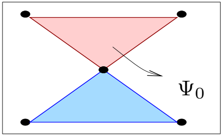
$$-\log f(\mathbf{Y}_V; G_d) = \Psi_{1,2}(Y_1, Y_2) + \Psi_{2,3}(Y_2, Y_3) + \Psi_{3,4}(Y_3, Y_4)$$



Binary Hypothesis Testing of MRFs

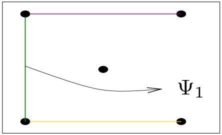
Null Hypothesis

$\mathcal{H}_0 : \{G_0(V), \mathcal{C}_0, \Psi_0\}$



Alternative Hypothesis

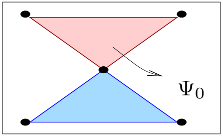
$\mathcal{H}_1 : \{G_1(V), \mathcal{C}_1, \Psi_1\}$



Binary Hypothesis Testing of MRFs

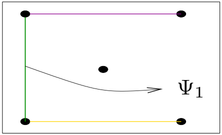
Null Hypothesis

$\mathcal{H}_0 : \{G_0(V), \mathcal{C}_0, \Psi_0\}$



Alternative Hypothesis

$\mathcal{H}_1 : \{G_1(V), \mathcal{C}_1, \Psi_1\}$



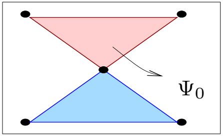
Minimal Sufficient Statistic for Binary Hypothesis Testing

$$\text{Log Likelihood Ratio: } \mathsf{LLR}(\mathbf{Y}_V) = \log \frac{f(\mathbf{Y}_V; \mathcal{H}_1)}{f(\mathbf{Y}_V; \mathcal{H}_0)}$$

Binary Hypothesis Testing of MRFs

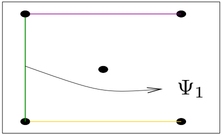
Null Hypothesis

$\mathcal{H}_0 : \{G_0(V), \mathcal{C}_0, \Psi_0\}$



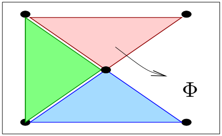
Alternative Hypothesis

$\mathcal{H}_1 : \{G_1(V), \mathcal{C}_1, \Psi_1\}$



Effective MRF For LLR

$\{G_0(V) \cup G_1(V), \max(\mathcal{C}_0 \cup \mathcal{C}_1), \Phi\}$



Minimal Sufficient Statistic for Binary Hypothesis Testing

Log Likelihood Ratio: $\text{LLR}(\mathbf{Y}_V) = \log \dfrac{f(\mathbf{Y}_V; \mathcal{H}_1)}{f(\mathbf{Y}_V; \mathcal{H}_0)}$

LLR in MRF= Log-Likelihood of Effective MRF

$$\text{LLR}(\mathbf{Y}_V) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$$

Outline

Minimum Cost Fusion for Inference

Problem Statement

Minimize sum routing costs s.t. $\text{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$ is delivered

Minimum Cost Fusion for Inference

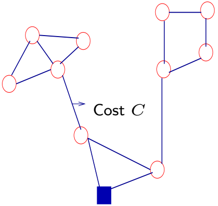
Problem Statement

Minimize sum routing costs s.t. $\mathsf{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$ is delivered

Network & Communication Model

Connected Network, Bidirectional Links, Unicast Mode

Comm. Graph with Link Costs



Minimum Cost Fusion for Inference

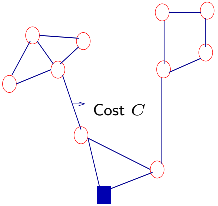
Problem Statement

Minimize sum routing costs s.t. $\text{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$ is delivered

Network & Communication Model

Connected Network, Bidirectional Links, Unicast Mode

Comm. Graph with Link Costs



Minimum Cost Fusion for Inference

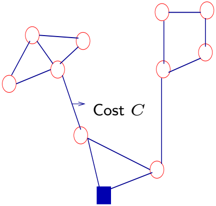
Problem Statement

Minimize sum routing costs s.t. $\mathsf{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$ is delivered

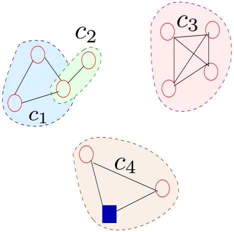
Network & Communication Model

Connected Network, Bidirectional Links, Unicast Mode

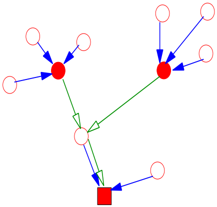
Comm. Graph with Link Costs



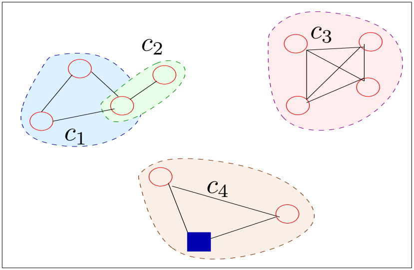
Cliques of Dependency Graph

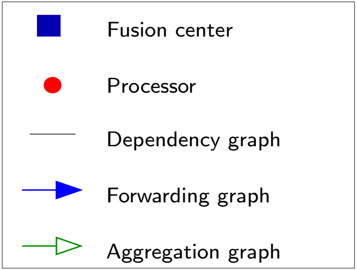


Min. Cost Fusion

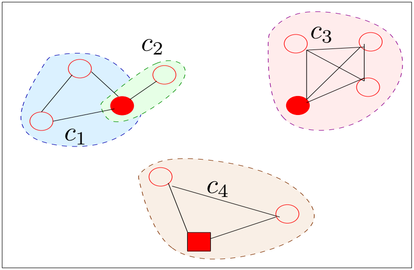


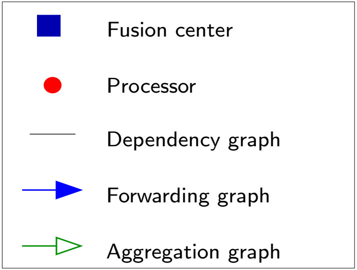
Stages of LLR Computation: $\text{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



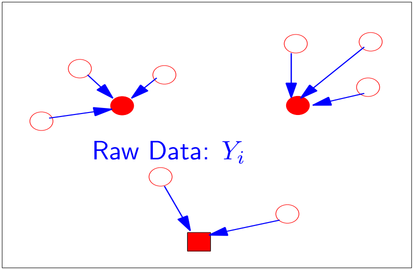


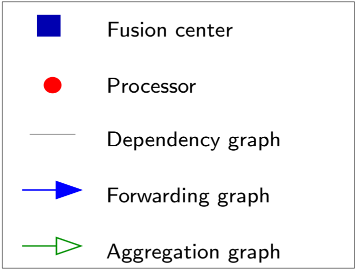
Stages of LLR Computation: $\mathbf{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



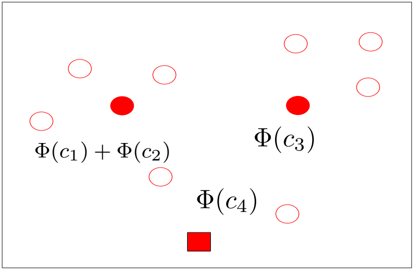


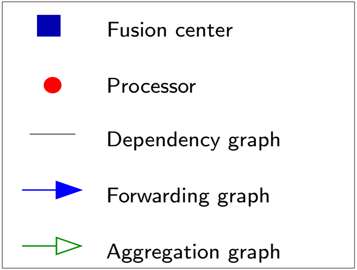
Stages of LLR Computation: $\mathrm{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



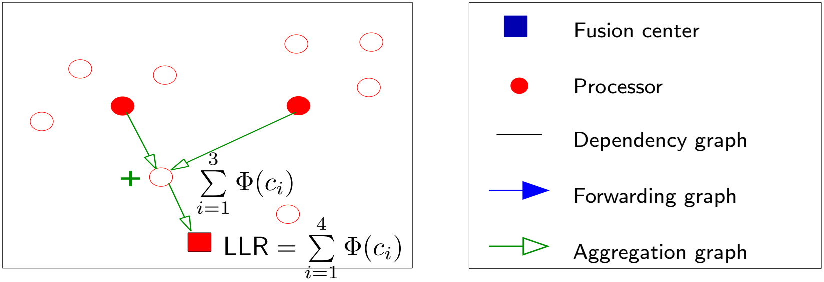


Stages of LLR Computation: LLR$(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$

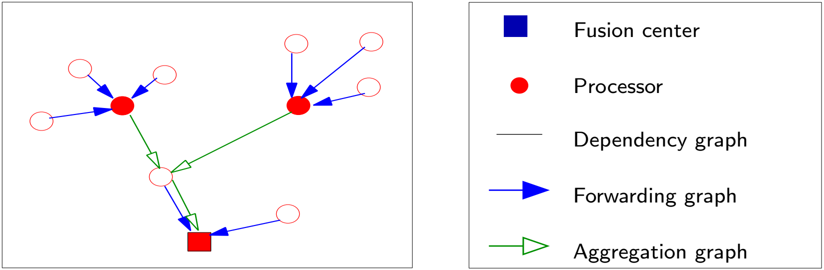




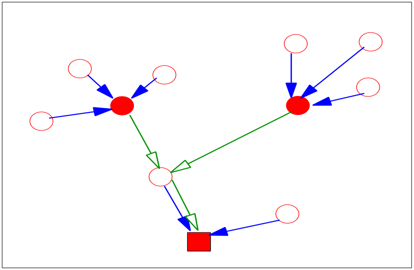
Stages of LLR Computation: $\mathsf{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$

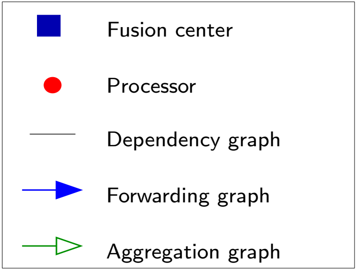


Stages of LLR Computation: LLR$(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



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

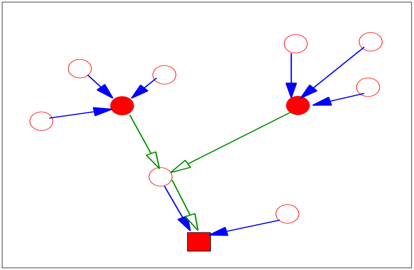


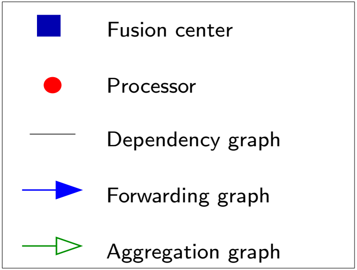


Local Computation of Clique Potentials: Processor is a Clique Member

- Simplifies optimization problem
- Local knowledge of function parameters

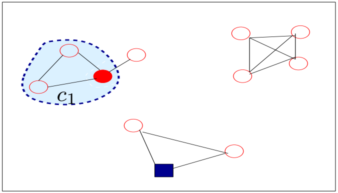
Stages of LLR Computation: $\text{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



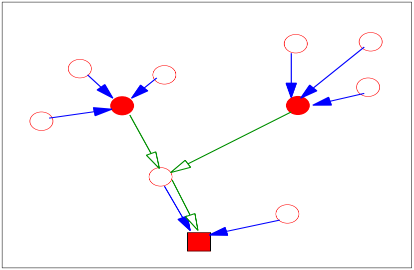


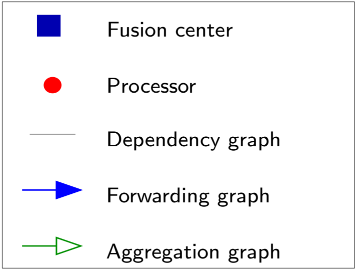
Local Computation of Clique Potentials: Processor is a Clique Member

- Simplifies optimization problem
- Local knowledge of function parameters



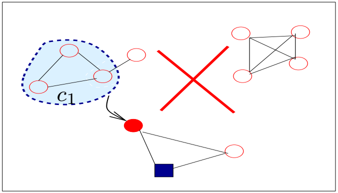
Stages of LLR Computation: $\text{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



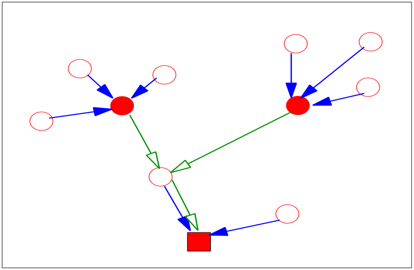


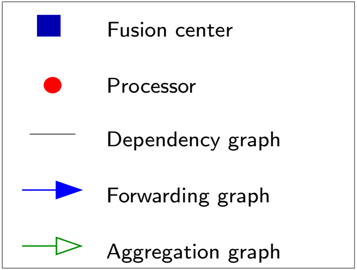
Local Computation of Clique Potentials: Processor is a Clique Member

- Simplifies optimization problem
- Local knowledge of function parameters



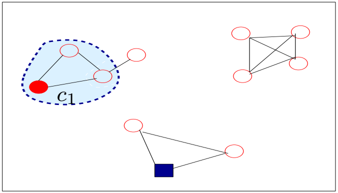
Stages of LLR Computation: $\mathrm{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



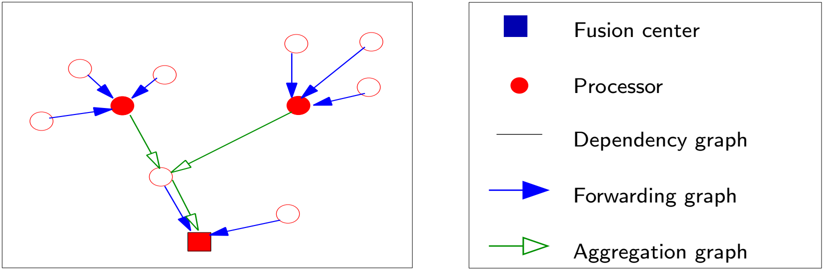


Local Computation of Clique Potentials: Processor is a Clique Member

- Simplifies optimization problem
- Local knowledge of function parameters

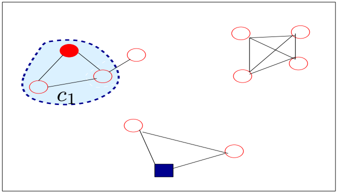


Stages of LLR Computation: $\mathrm{LLR}(\mathbf{Y}_n) = \sum_{c \in \mathcal{C}} \Phi_c(\mathbf{Y}_c)$



Local Computation of Clique Potentials: Processor is a Clique Member

- Simplifies optimization problem
- Local knowledge of function parameters

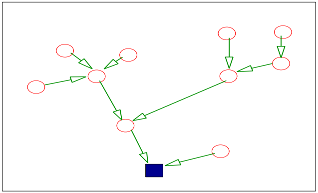


Outline

Lower Bound of MST & AggMST Heuristic

Minimum Spanning Tree: Lower Bound for Min Routing Cost

Independent data: achieves bound ⇒ Correlation increases cost

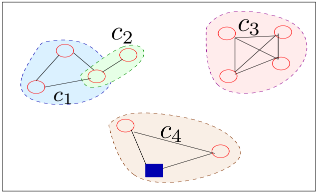


Lower Bound of MST & AggMST Heuristic

Minimum Spanning Tree: Lower Bound for Min Routing Cost

Independent data: achieves bound ⇒ Correlation increases cost

AggMST Heuristic



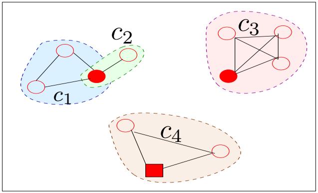
Lower Bound of MST & AggMST Heuristic

Minimum Spanning Tree: Lower Bound for Min Routing Cost

Independent data: achieves bound ⇒ Correlation increases cost

AggMST Heuristic

- Processor Assignment: Any clique member



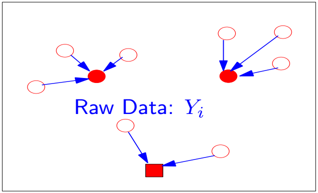
Lower Bound of MST & AggMST Heuristic

Minimum Spanning Tree: Lower Bound for Min Routing Cost

Independent data: achieves bound ⇒ Correlation increases cost

AggMST Heuristic

- Processor Assignment: Any clique member
- Forwarding: Other members to processor



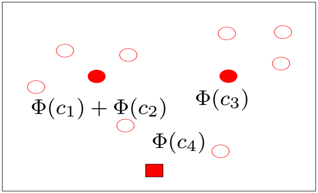
Lower Bound of MST & AggMST Heuristic

Minimum Spanning Tree: Lower Bound for Min Routing Cost

Independent data: achieves bound ⇒ Correlation increases cost

AggMST Heuristic

- Processor Assignment: Any clique member
- Forwarding: Other members to processor



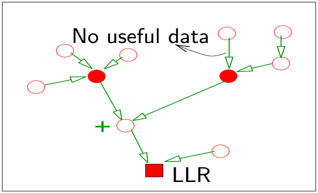
Lower Bound of MST & AggMST Heuristic

Minimum Spanning Tree: Lower Bound for Min Routing Cost

Independent data: achieves bound ⇒ Correlation increases cost

AggMST Heuristic

- Processor Assignment: Any clique member
- Forwarding: Other members to processor
- Aggregation: MST, towards fusion center



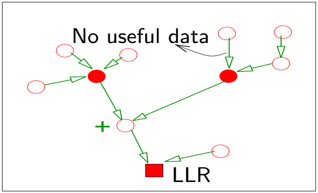
Lower Bound of MST & AggMST Heuristic

Minimum Spanning Tree: Lower Bound for Min Routing Cost

Independent data: achieves bound $\Rightarrow$ Correlation increases cost

AggMST Heuristic

- Processor Assignment: Any clique member
- Forwarding: Other members to processor
- Aggregation: MST, towards fusion center



Approximation Algorithm with Ratio ρ

Routing cost no worse than ρ times optimal, runs in polynomial time

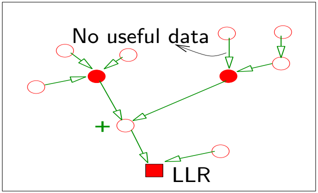
Lower Bound of MST & AggMST Heuristic

Minimum Spanning Tree: Lower Bound for Min Routing Cost

Independent data: achieves bound $\Rightarrow$ Correlation increases cost

AggMST Heuristic

- Processor Assignment: Any clique member
- Forwarding: Other members to processor
- Aggregation: MST, towards fusion center



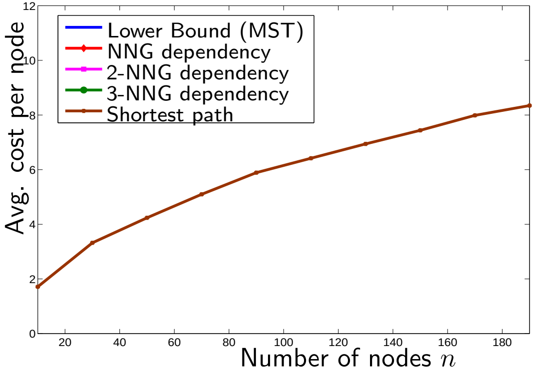
Approximation Algorithm with Ratio ρ

Routing cost no worse than ρ times optimal, runs in polynomial time

Approximation Ratio of AggMST $= 2$ for Nearest-Neighbor Graph

$$\frac{\mathsf{C}(\text{AggMST})}{\mathsf{C}(G^*)} \leq \frac{\mathsf{C}(\text{AggMST})}{\mathsf{C}(\text{MST})} \leq 2$$

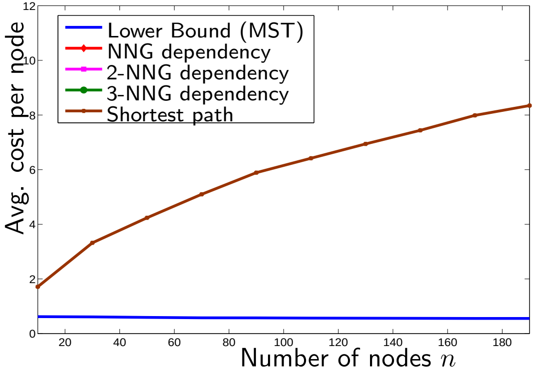
Simulation Results



Implications

- Scalable in network size for k-NNG dependency
- Fusion cost sensitive to No. of cliques in dependency graph

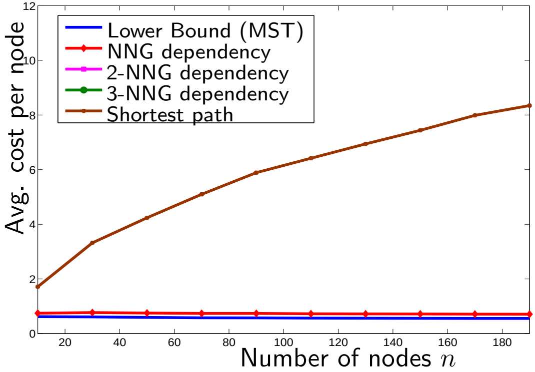
Simulation Results



Implications

- Scalable in network size for k-NNG dependency
- Fusion cost sensitive to No. of cliques in dependency graph

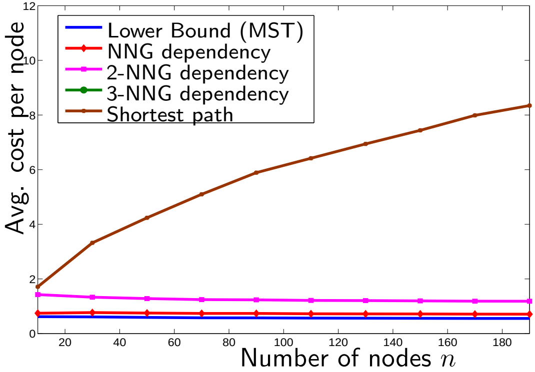
Simulation Results



Implications

- Scalable in network size for k-NNG dependency
- Fusion cost sensitive to No. of cliques in dependency graph

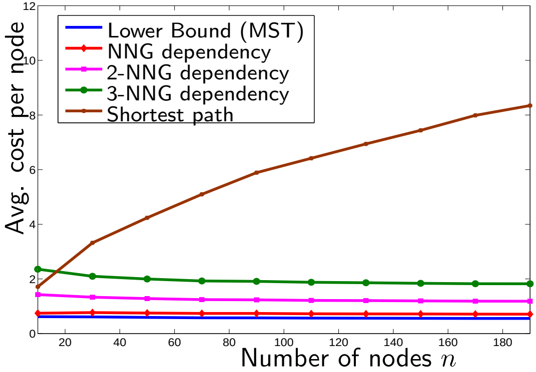
Simulation Results



Implications

- Scalable in network size for k-NNG dependency
- Fusion cost sensitive to No. of cliques in dependency graph

Simulation Results



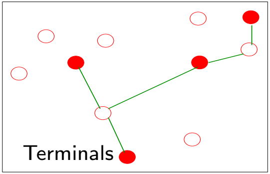
Implications

- Scalable in network size for k-NNG dependency
- Fusion cost sensitive to No. of cliques in dependency graph

Steiner-Tree Reduction

Steiner Tree

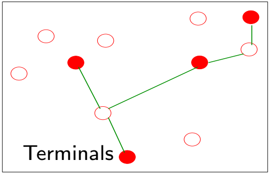
- Minimum cost tree containing a required set of nodes called terminals
- NP-hard problem, currently the best approximation is 1.55



Steiner-Tree Reduction

Steiner Tree

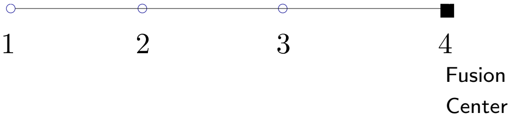
- Minimum cost tree containing a required set of nodes called terminals
- NP-hard problem, currently the best approximation is 1.55



Main result

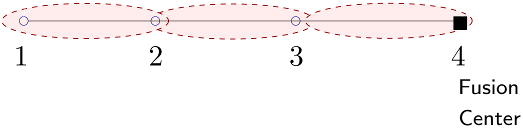
Min cost fusion has approx. ratio preserving Steiner tree reduction

Example : Chain dependency graph



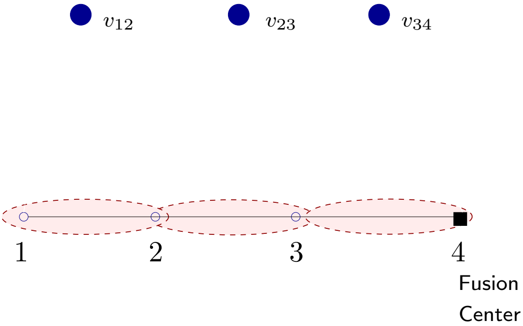
Graph transformation and building Steiner tree.

Example : Chain dependency graph



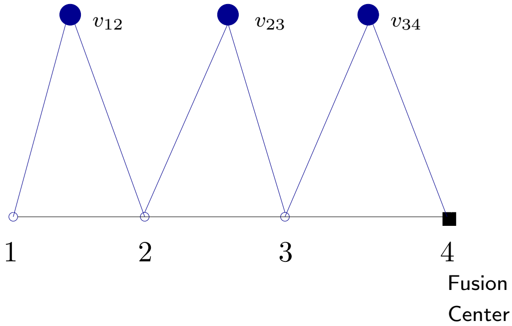
Graph transformation and building Steiner tree.

Example : Chain dependency graph



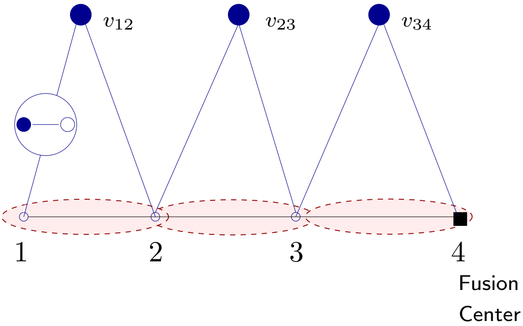
Graph transformation and building Steiner tree.

Example : Chain dependency graph



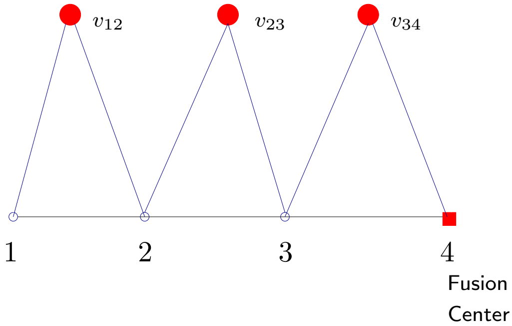
Graph transformation and building Steiner tree.

Example : Chain dependency graph



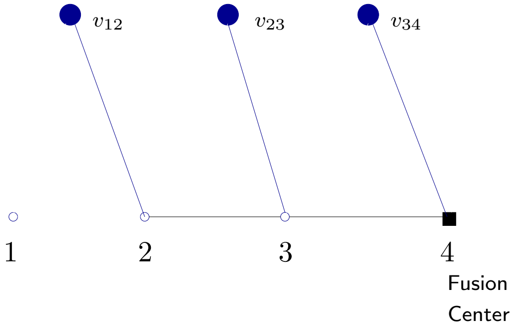
Graph transformation and building Steiner tree.

Example : Chain dependency graph



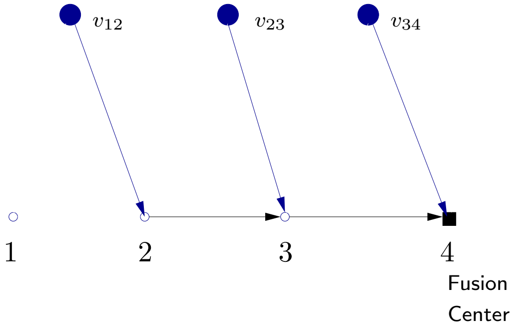
Graph transformation and building Steiner tree.

Example : Chain dependency graph



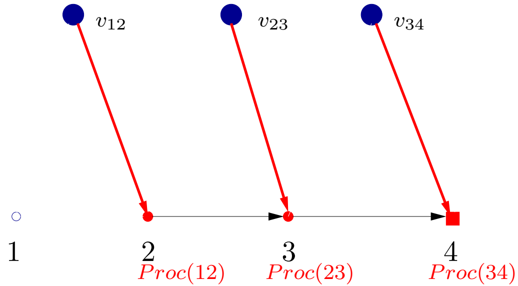
Graph transformation and building Steiner tree.

Example : Chain dependency graph



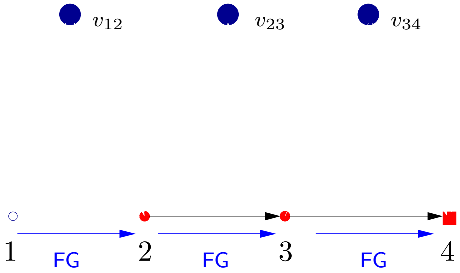
Graph transformation and building Steiner tree.

Example : Chain dependency graph



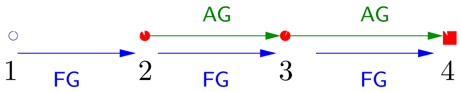
Graph transformation and building Steiner tree.

Example : Chain dependency graph



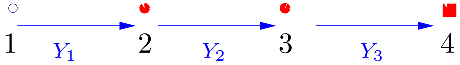
Graph transformation and building Steiner tree.

Example : Chain dependency graph



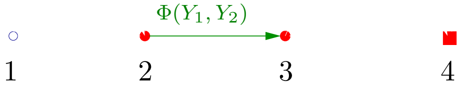
Graph transformation and building Steiner tree.

Example : Chain dependency graph



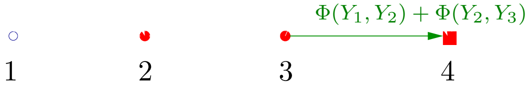
Graph transformation and building Steiner tree.

Example : Chain dependency graph



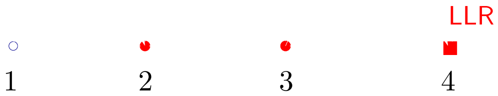
Graph transformation and building Steiner tree.

Example : Chain dependency graph



Graph transformation and building Steiner tree.

Example : Chain dependency graph



Graph transformation and building Steiner tree.

Outline

Conclusion

Summary

- Minimum cost data fusion for inference
- Concept of dependency graph based routing
  - Exploit correlation structure to fuse data efficiently
- Proposed MST-based heuristic: AggMST
  - 2-approximation for NNG, simple construction
- Steiner-tree reduction
  - 1.55-approximation, Use of Steiner tree algorithms

Conclusion

Summary

- Minimum cost data fusion for inference
- Concept of dependency graph based routing
  - Exploit correlation structure to fuse data efficiently
- Proposed MST-based heuristic: AggMST
  - 2-approximation for NNG, simple construction
- Steiner-tree reduction
  - 1.55-approximation, Use of Steiner tree algorithms

Outlook

- Incorporating physical layer issues
  - Effect of interference, Broadcast nature of wireless medium
- Spatial probability approach for large random networks (Allerton 07)
- Tradeoff between routing costs and inference performance

Previous Works on Correlated Data Routing

Spatial Correlation Models: All incorporated under MRF framework

- Joint-Gaussian, distance based correlation (Marco et al. 03, Yoon & Shahabi 07)
- Joint entropy (Pattem et al. 04), spl. MRF (Jindal & Psounis 06)

Correlated Data Gathering (Cristescu et al. 06, Scaglione&Servetto 02)

Raw data not needed at fusion center, only the likelihood function for optimal inference

In-network Function Computation (Giridar & Kumar 06)

Valid for symmetric functions, likelihood function may not have this form

Routing for Inference: For Special Correlation Models

- Independent Measurements: (Yang & Blum 07, Yu & Ephremides 06)
- 1-D Gauss-Markov process: (Sung et al. 06, Chamberland & Veeravalli 06)

Routing for Belief Propagation (Kreidl & Willsky 06, Williams et al. 05)

Local MAP estimate of raw data at each node: not global decision at fusion center

Thank You !