Information Flow Optimization in Inference Networks

Aditya Deshmukh[1], Jing Liu [1], Venugopal Veeravalli[1], Gunjan Verma[2]

[1]Coordinated Science Laboratory, University of Illinois at Urbana-Champaign

[2]Army Research Laboratory, Adelphi, MD

ICASSP 2020 Poster Presentation
April 18, 2020

Introduction

- Sensor nodes: take observations, compress and send through capacitated network
- Relay nodes: route messages towards fusion center
- Fusion center: infer state of nature
- Objective: optimize information flow for inference task (for maximizing relevant information content at the Fusion center)

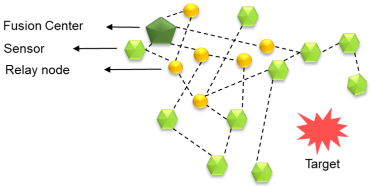
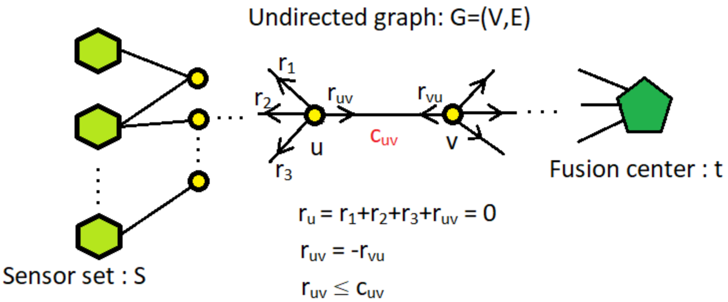


Figure: An example of sensor network

Problem setup



- Concave utility function $g_s : \mathbb{R}^+ \to \mathbb{R}$ for each $s \in S$
- Optimization problem:

$$\max_{r_{uv}} \sum_{s \in S} g_s(r_s)$$

$$\begin{aligned} \text{subject to } & \forall (u,v) \in E,\ r_{uv} \leq c_{uv} \\ & \forall (u,v) \in E,\ r_{uv} = -r_{vu} \\ & \forall u \in V \setminus \{S \cup t\},\ r_u = 0. \end{aligned}$$

Problem solution

- Real-valued rates $\Longrightarrow$ convex program
- Problem lies in category of Network Utility Maximization problems
- Existing algorithms, for e.g. Dual Decomposition method

Problem solution

- Real-valued rates $\implies$ convex program
- Problem lies in category of Network Utility Maximization problems
- Existing algorithms, for e.g. Dual Decomposition method
- If rates and capacities are integer-valued, solution of real-valued relaxation can be used to obtain integral solution, such that integral rates $r_{uv}^{(I)}$ are close to real-valued rates r_{uv} in the sense that $\sum_{s \in S} r_s^{(I)} = \left\lfloor \sum_{s \in S} r_s \right\rfloor$ (Lee et. al. '13)

Flow Optimization for Parameter Estimation

- Sensor network with N spatially distributed sensors
- Each sensor i:

$$y_i = \boldsymbol{a}_i^T \boldsymbol{x} + \eta_i, i = 1, ..., N,$$

jointly written as:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta}$$

where
$\boldsymbol{y} \in \mathbb{R}^N$: measurements result from all the N sensors;
$\boldsymbol{A} \in \mathbb{R}^{N \times q}$: known;
$\boldsymbol{\eta} \in \mathbb{R}^N$: i.i.d. bounded noise with zero mean and variance σ^2.

Quantization

- Uniform Quantizer, fine quantization
- For each y_i, denote its quantized version as d_i, and corresponding quantization noise is $\epsilon_i = d_i - y_i$. So

$$d_i = \boldsymbol{a}_i^T \boldsymbol{x} + \eta_i + \epsilon_i, i = 1, ..., N,$$

  For small quantization intervals Δ_i, ϵ_i is approximately uncorrelated with y_i and has zero-mean with variance $\Delta_i^2/12$.
- Fusion center performs least squares $\hat{\boldsymbol{x}} = \boldsymbol{A}^\dagger \boldsymbol{d}$.

Quantization

- Uniform Quantizer, fine quantization
- For each y_i, denote its quantized version as d_i, and corresponding quantization noise is $\epsilon_i = d_i - y_i$. So

$$d_i = \boldsymbol{a}_i^T \boldsymbol{x} + \eta_i + \epsilon_i, i = 1, ..., N,$$

  For small quantization intervals Δ_i, ϵ_i is approximately uncorrelated with y_i and has zero-mean with variance $\Delta_i^2/12$.
- Fusion center performs least squares $\hat{\boldsymbol{x}} = \boldsymbol{A}^\dagger \boldsymbol{d}$.
- **Q:** *Given the limited capacity of the network, to minimize the estimation error at the fusion center, how should we optimize the network flows and how many bits should we allocate to each sensor?*

Minimize the mean squared error $\mathbb{E}[\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2]$

$$\begin{aligned}
\mathbb{E}[\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2] &= \mathbb{E}[\|\boldsymbol{A}^\dagger \boldsymbol{d} - \boldsymbol{x}\|_2^2] \\
&= \mathbb{E}[\|\boldsymbol{A}^\dagger (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta} + \boldsymbol{\epsilon}) - \boldsymbol{x}\|_2^2] \\
&\approx \mathrm{Tr}\{\boldsymbol{A}^\dagger \mathrm{diag}(\sigma^2 + \Delta_1^2/12, ..., \sigma^2 + \Delta_N^2/12)(\boldsymbol{A}^\dagger)^T\} \\
&= \sum_{i=1}^{N} \left[(\sigma^2 + \Delta_i^2/12)\|\boldsymbol{A}^\dagger(:, i)\|_2^2\right]
\end{aligned}$$

$\Delta_i \propto \frac{1}{2^{r_i}}$, where r_i is #bits allocated to sensor i. Equivalent to maximizing:

$$\sum_{i=1}^{N} -\|\boldsymbol{A}^\dagger(:, i)\|_2^2 / 4^{r_i},$$

Now overall problem is cast as the proposed optimization problem

Simulations for Parameter Estimation

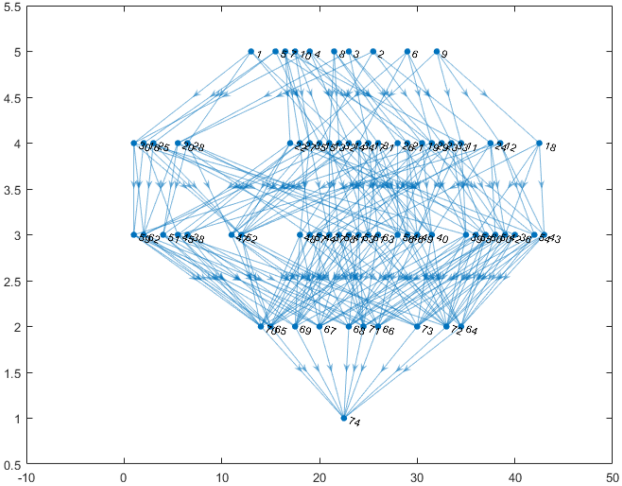


Figure: An instance of a generated network

Simulations for Parameter Estimation

- Some sensors close to the target $\longrightarrow$ strong signal
  The rest are away from to the target$\longrightarrow$ weak signal
- Generated $\boldsymbol{A} \in \mathbb{R}^{10\times 3}$ with entries $\sim U(0,1)$, then multiplied its first 4 rows by α

Table: MSE of the estimated $\hat{\boldsymbol{x}}$

MSE	α=1	α=0.3	α=0.1
Max-Flow	0.2673	1.1939	1.5068
Proposed	0.0148	0.0230	0.0241

Flow Optimization for Detection

Classical results (Tsitsiklis '93):

- Problem: design optimal quantizer to perform binary hypothesis testing given number of quantization levels
- Under hypothesis H_i, $i = 0, 1$, Y has the distribution P_i
- Deterministic n-level quantizer is measurable function that maps $\mathbb{R}$ to $\{1, 2, \ldots, n\}$
- Γ_n : set of all randomized n-level quantizers
- $Q_i(\gamma)$: distribution of $\gamma(Y)$ under hypothesis H_i, where γ is a randomized quantizer

Utility function for Detection

- Surrogate problem: find quantizer to maximize Kullback-Leibler(KL) divergence between $Q_1(\gamma)$ and $Q_0(\gamma)$
- Neyman-Pearson testing: error exponent is KL divergence
- $f(n) := \sup_{\gamma \in \Gamma_n} D(Q_1(\gamma) || Q_0(\gamma))$
- Maximized by likelihood ratio quantizer. Likelihood ratio quantizer $\gamma_{\boldsymbol{t}}$ with thresholds $\boldsymbol{t} = (t_0, t_1, \ldots, t_n) \in [0, \infty]^{n+1}$ satisfies:

  $$P_i(\gamma_{\boldsymbol{t}}(Y) = \ell \text{ and } L(Y) \notin [t_{\ell-1}, t_\ell]) = 0,$$

  $L(Y)$: likelihood ratio between $\mathbb{P}_1$ and $\mathbb{P}_0$
- $f(n) = \max_{\boldsymbol{t}} D(Q_1(\gamma_{\boldsymbol{t}}) || Q_0(\gamma_{\boldsymbol{t}}))$
- Utility function: $g(r) = f(2^r)$

Sensor network setting

- Observations from sensors are conditionally independent
- $P_i^{(j)}$: distribution of observation $Y_j \in \mathcal{Y}_j$ from sensor j under the hypotheses H_i, $i = 0, 1$
- $Q_i^{(j)} = Q_i^{(j)}(\gamma_j)$: distribution of the quantized observation $\gamma_j(Y_j)$ from j^{th} sensor
- Objective:

$$\begin{aligned}
&\max_{n_j, \gamma_j \in \Gamma_{n_j}} \sum_{j=1}^{N} D\Big(Q_1^{(j)}(\gamma_j) || Q_0^{(j)}(\gamma_j)\Big) \\
&= \max_{n_j, \boldsymbol{t_j}} \sum_{j=1}^{N} D\Big(Q_1^{(j)}(\gamma_{\boldsymbol{t_j}}) || Q_0^{(j)}(\gamma_{\boldsymbol{t_j}})\Big) \\
&= \max_{r_j} \sum_{j=1}^{N} f^{(j)}(2^{r_j}) = \max_{r_j} \sum_{j=1}^{N} g^{(j)}(r_j)
\end{aligned}$$

Concavity of utility functions

- Utility function: $g(r) = f(2^r)$, where $r = \log_2(n)$ and n is the number of quantization levels
- Linear interpolation to get surrogate utility function for real-valued relaxation
- It is difficult to show that $g(r)$ is concave for general distributions
- We verified via simulations that it is indeed concave for various distributions

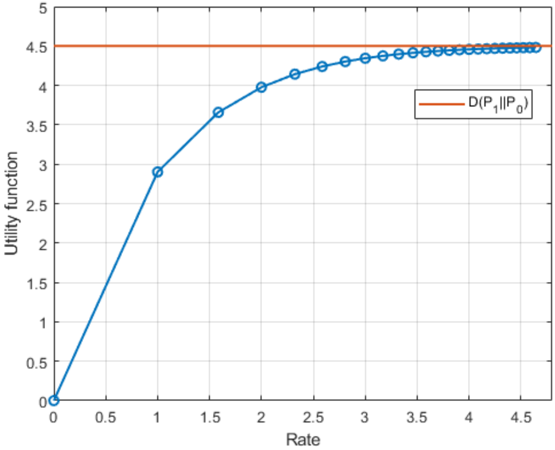


Figure: Plot of $g(r)$: P_0 is $\mathcal{N}(0, 1)$ and P_1 is $\mathcal{N}(3, 1)$.

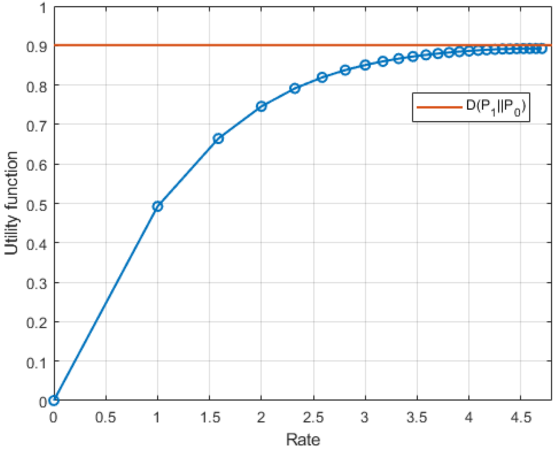
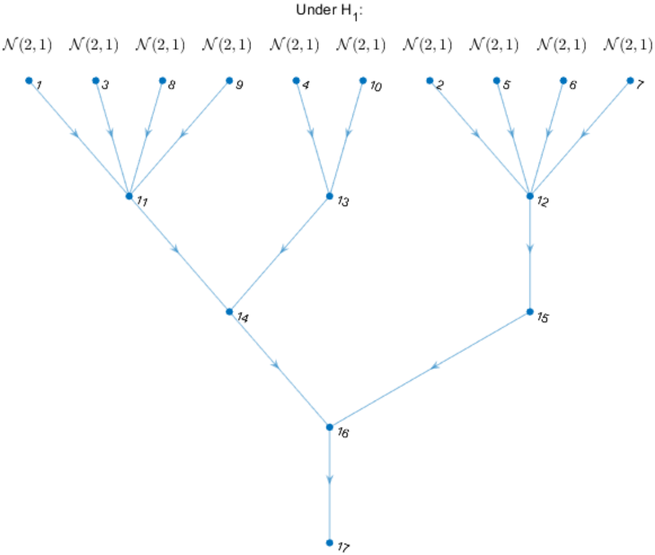


Figure: Plot of $g(r)$: P_0 is $\text{Exp}\left(\frac{1}{2}\right)$ and P_1 is $\text{Exp}\left(\frac{1}{6}\right)$.

Simulations for binary hypothesis testing

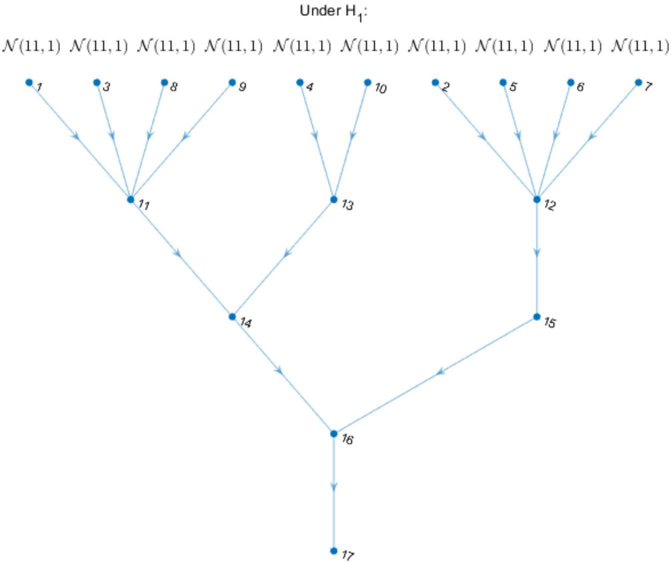
- Small network to illustrate the performance
- We compare performance with that of Max-Flow
- Four different settings of distributions
- In all settings, under H_0, all sensors follow $\mathcal{N}(0, 1)$
- Gain over Max-Flow depends on factors such as distance between the distributions under the two hypothesis and variability across sensors

Setting 1



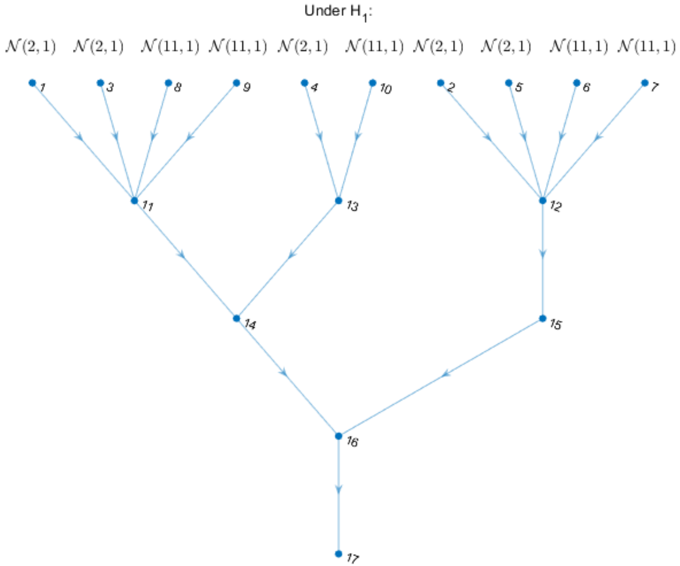
- Proposed framework: 5.1152, Max-FLow: 5.1152

Setting 2



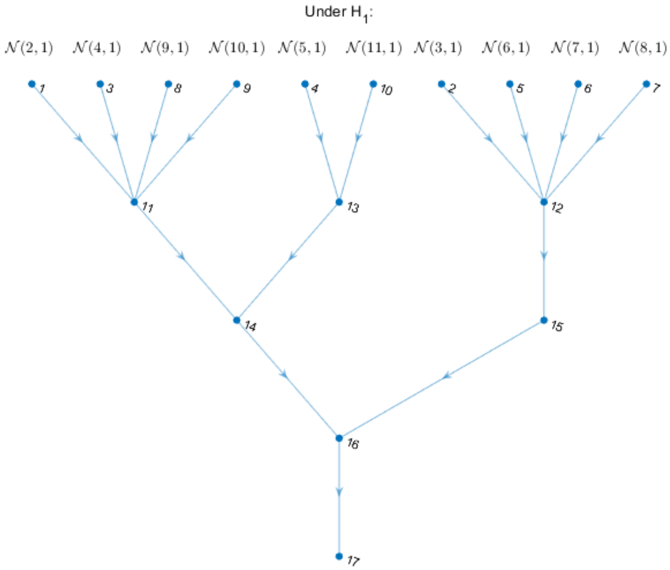
- Proposed framework: 124.0653, Max-FLow: 99.3528

Setting 3



- Proposed framework: 103.7984, Max-FLow: 65.3699

Setting 4



- Proposed framework: 96.9858, Max-FLow: 57.3427

Conclusion

- Showed that problem to find optimal rates in sensor networks tasked with inference objectives, can be cast as optimization problem
- Problem lies in the category of Network Utility Maximization problems, if appropriate utility functions are assigned to each sensor
- Showed existence of utility functions in common parameter estimation setting and detection setting
- Simulations illustrate the gain of the proposed framework over the traditional Max-Flow algorithm

Thank You!