Optimal Registration in Networks

Doug Cochran

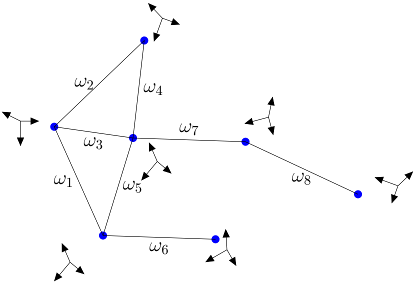
School of Mathematical & Statical Sciences
Arizona State University

(Joint work with Stephen Howard and Bill Moran)

20 October 2010

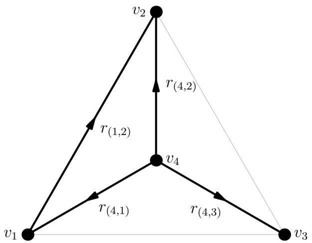
Introduction
Tenets

- The purpose of sensor networks is to *sense*; i.e., to enable detection, estimation, classification, and tracking
- To exploit network structure, data at the nodes must be *registered*
  - Intrinsic data; e.g., clocks, platform orientation
  - Extrinsic data: collected by sensors



Introduction
Registration on a Graph

- *Vertex labels:* An element of a Lie group G is associated with each node of a graph modeling a network
- *Edge labels:* An element of G representing a noisy measurement of the difference of the values on the vertices at either end
- *Goal:* Make the best estimate of the *connection* – the true relative offsets between the vertex values



- Denote by $f(x|\theta)$ a family of probability density functions parameterized by elements θ on a manifold M
- For given data x, $f(x|\theta)$ defines the likelihood function on M
- A maximum-likelihood (ML) estimate for θ is defined by

$$\hat{\theta} = \arg\max_{\theta \in M} f(x|\theta)$$

- The log-likelihood function $\ell(x|\theta) : M \to \mathbb{R}$ is defined by

$$\ell(x|\theta) = \log f(x|\theta)$$

- A natural measure of discrimination on M is the Kullback-Leibler divergence

$$D(f_\theta || f'_\theta) = \int f(x|\theta) \log \frac{f(x|\theta)}{f(x|\theta')}\, dx$$

- Although the KL divergence is not symmetric, locally it gives a metric on M called the Fisher information metric
  - $F = \mathsf{E}[d\ell \otimes d\ell]$
  - F can also be computed as $F = -\mathsf{E}[\nabla^2 \ell]$, for any connection
- The corresponding volume form

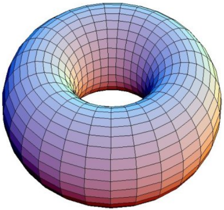
$$\mathrm{vol}_F = \sqrt{\det F}\, d\theta_1 \wedge \cdots \wedge d\theta_m$$

is the Jeffreys prior

- The state of the network is a G-valued function x on $V(\Gamma)$ but it is never directly observed
  - If the network were aligned, the function would be constant (flat connection)
- What is observed is the connection $\omega \in \Omega = G^m$
  - In the absence of noise, for an aligned network this is the identity connection
  - If the network is not aligned, a gauge transformation can be found that takes ω to the identity

$$x_v \mapsto g_v x_v \qquad \omega_e \mapsto g_{t(e)} \omega_e g_{s(e)}^{-1}$$

- Gauge transformations that fix the identity form a normal subgroup that can be quotiented out
- The noise on the measurements should be a random variable taking values $\varepsilon \in \Omega$ and whose density $p(\varepsilon)$ is defined with respect to the normalized Haar measure on Ω



- Directed graph Γ with vertex set $V(\Gamma)$ and edge set $E(\Gamma)$

$$|V(\Gamma)| = n \qquad |E(\Gamma)| = m$$

- Vertex space $C_0(\Gamma)$ – functions from $V(\Gamma)$ to $\mathbb{R}$; a real vector space:

$$\boldsymbol{f} = \sum_{j=1}^{n} f_j \boldsymbol{v}_j \qquad \boldsymbol{v}_j(v_\ell) = \delta_{j\ell}$$

- Edge space $C_1(\Gamma)$ – functions from $E(\Gamma)$ to $\mathbb{R}$

$$\boldsymbol{\omega} = \sum_{j=1}^{m} \omega_j \boldsymbol{e}_j \qquad \boldsymbol{\omega}_j(\omega_\ell) = \delta_{j\ell}$$

- Source and target maps $s, t : C_1(\Gamma) \to C_0(\Gamma)$; if $e = (v_i, v_j)$,

$$s(\boldsymbol{e}) = \boldsymbol{v}_i \qquad t(\boldsymbol{e}) = \boldsymbol{v}_j$$

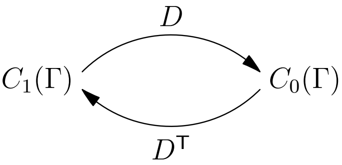
- Vertex space $C_0(\Gamma)$ – functions from $V(\Gamma)$ to $\mathbb{R}$; a real vector space:

$$\boldsymbol{f} = \sum_{j=1}^{n} f_j \boldsymbol{v}_j \qquad \boldsymbol{v}_j(v_\ell) = \delta_{j\ell}$$

Mathematical Preliminaries
Graphs

- Boundary map (incidence matrix) $D : C_1(\Gamma) \to C_0(\Gamma)$ by $D(\boldsymbol{e}) = t(\boldsymbol{e}) - s(\boldsymbol{e})$
- Coboundary map $D^{\mathsf{T}} : C_0(\Gamma) \to C_1(\Gamma)$ by

$$D^{\mathsf{T}}(\boldsymbol{v}) = \sum_{t(e_j)=v} \boldsymbol{e}_j - \sum_{s(e_j)=v} \boldsymbol{e}_j$$



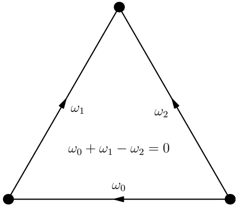
- Cycle space $Z(\Gamma)$: A *cycle* is a closed path in Γ; i.e., a sequence of vertices $\mathfrak{L} = v_1 v_2 \ldots v_q$ where v_i is adjacent to v_{i+1} for $i = 1, \ldots, q$ and v_q is adjacent to v_1

$$z_{\mathfrak{L}}(e_j) = \begin{cases} 1 & \text{if } e_j \in \mathfrak{L} \text{ and } e_j \text{ is oriented as } \mathfrak{L} \\ -1 & \text{if } e_j \in \mathfrak{L} \text{ and } e_j \text{ is oriented opposite to } \mathfrak{L} \\ 0 & \text{otherwise} \end{cases}$$

- $Z(\Gamma)$ is the linear subspace of $C_1(\Gamma)$ spanned by the $z_{\mathfrak{L}}$
- *Kirchoff's current law (KCL):* $Z(\Gamma) = \ker D$; i.e., $D\boldsymbol{z} = 0$ for all $\boldsymbol{z} \in Z(\Gamma)$

$$\sum_{t(e_j)=v} \boldsymbol{e}_j - \sum_{s(e_j)=v} \boldsymbol{e}_j = \boldsymbol{0} \quad \forall v \in V(\Gamma)$$

- Cocycle space $B(\Gamma)$: The orthogonal complement of $Z(\Gamma)$ in $C_!(\Gamma)$
  - For any $\boldsymbol{z} \in Z(\Gamma)$ and $\boldsymbol{\omega} \in B(\Gamma)$, $\langle \boldsymbol{z}, \boldsymbol{\omega} \rangle_{C_1} = 0$
- $B(\Gamma)$ is the image of $C_0(\Gamma)$ under the coboundary operator; i.e., every $\boldsymbol{\omega} \in B(\Gamma)$ can be written as $\boldsymbol{\omega} = D^{\mathsf{T}}\boldsymbol{x}$ for some $\boldsymbol{x} \in C_0(\Gamma)$
- This gives *Kirchhoff's voltage law (KVL)*



Gaussian noise on $\mathbb{R}$
Data Model

- The data has the form $\boldsymbol{r} \in C_1(\Gamma)$

$$\boldsymbol{r} = \boldsymbol{\omega} + \boldsymbol{\varepsilon}$$

- The true offsets satisfy $\boldsymbol{\omega} \in B(\Gamma)$ (KVL)
- The noise $\boldsymbol{\varepsilon} \in C_1(\Gamma)$ has probability density

$$f(\boldsymbol{\varepsilon}) = \frac{1}{\sqrt{(2\pi)^m \det R}} \exp\left(-\frac{1}{2}\langle \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}\rangle_R\right)$$

where

$$\langle \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}\rangle_R = \boldsymbol{\varepsilon}^{\mathsf{T}} R^{-1} \boldsymbol{\varepsilon}$$

- The conditional density for $\boldsymbol{r}$ given $\boldsymbol{\omega}$ is

$$f(\boldsymbol{\varepsilon}|\boldsymbol{\omega}) = \frac{1}{\sqrt{(2\pi)^m \det R}} \exp\left(-\frac{1}{2}\langle \boldsymbol{r}-\boldsymbol{\omega}, \boldsymbol{r}-\boldsymbol{\omega}\rangle_R\right)$$

- The maximum-likelihood estimate of $\boldsymbol{\omega}$ is

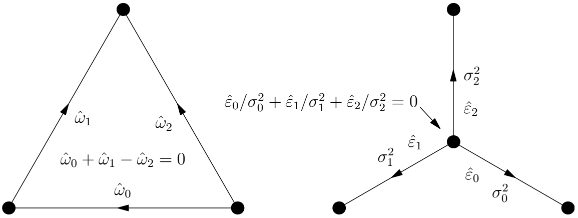
$$\hat{\boldsymbol{\omega}} = \arg\min_{\boldsymbol{\omega}\in B(\Gamma)} \langle \boldsymbol{r}-\boldsymbol{\omega}, \boldsymbol{r}-\boldsymbol{\omega}\rangle_R$$

- The data may be written as $\boldsymbol{r} = \hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\varepsilon}}$ with $\hat{\boldsymbol{\omega}} \in B(\Gamma)$ (KVL) and

$$\langle \hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\varepsilon}} \rangle_R = 0$$

i.e., the "residual" $\hat{\boldsymbol{\varepsilon}}$ satisfies

$$R^{-1}(\hat{\boldsymbol{\varepsilon}}) \in Z(\Gamma) \quad \text{(KCL)}$$



Gaussian noise on $\mathbb{R}$
ML Estimator

- Weighted graph Laplacian $L : C_0(\Gamma) \to C_0(\Gamma)$

$$L = DR^{-1}D^{\mathsf{T}}$$

- Not invertible since $\ker L$ contains constant functions in $C_0(\Gamma)$
- Choosing a reference vertex, the offset estimator $\hat{\boldsymbol{x}} \in W$ is

$$\hat{\boldsymbol{x}} = L_W^{-1} R^{-1} \boldsymbol{r}$$

where L_W (and D_W below) are defined with respect to the reduced vertex set
- The connection estimate is $\hat{\boldsymbol{\omega}} = D_W^{\mathsf{T}} \hat{\boldsymbol{x}}$

$$\hat{\boldsymbol{\omega}} = D_W^{\mathsf{T}} L_W^{-1} D_W R^{-1} \boldsymbol{r}$$

$D_W^{\mathsf{T}} L_W^{-1} D_W R^{-1}$ is the R-orthogonal projection into $B(\Gamma)$

Gaussian noise on $\mathbb{R}^d$

Performance of the ML Estimator with $R = \sigma^2 \mathbb{I}$

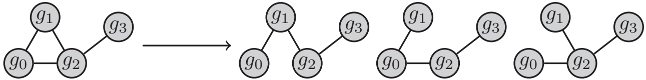
- For $R = \sigma^2 \mathbb{I}$, the Fisher information is

$$\frac{1}{\sigma^2} L_W$$

- Its determinant is

$$\det F = t(\Gamma)/\sigma^{2(n-1)}$$

- $t(\Gamma)$ is the number of spanning trees in Γ



Gaussian noise on $\mathbb{R}$

Performance of the ML Estimator with $R = \sigma^2 \mathbb{I}$

- The ML estimator $\hat{\boldsymbol{x}}$ is unbiased
- Its covariance is

$$C_{\hat{\boldsymbol{x}}} = \mathsf{E}[(\boldsymbol{x} - \hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}})^{\mathsf{T}}] = \sigma^2 L_W^{-1}$$

- Its determinant is

$$C_{\hat{\boldsymbol{x}}} = \sigma^{2(n-1)}/t(\Gamma)$$

Gaussian noise on $\mathbb{R}^d$

Performance of the ML Estimator with correlated edge data

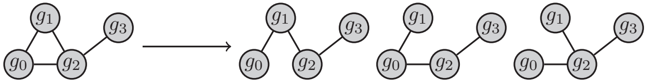
- For $R = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_m^2)$, the Fisher information determinant is

$$\det F = \sum_S \prod_{e_j \in S} \frac{1}{\sigma_j^2}$$

where the sum is over all spanning trees S in Γ

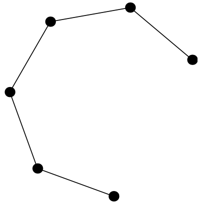
- The estimator $\hat{\boldsymbol{x}}$ is unbiased and

$$\det C_{\hat{\boldsymbol{x}}} = \left(\sum_S \prod_{e_j \in S} \frac{1}{\sigma_j^2} \right)^{-1}$$

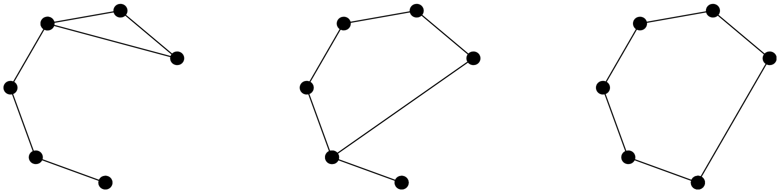


Gaussian Noise on $\mathbb{R}$
Network design

If one link can be added to...



Which of these is the best choice?



Local Estimation

One-dimensional Gaussian noise case

- For $R = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_m^2)$, the ML estimator residual satisfies KCL; i.e.,

$$DR^{-1}(\boldsymbol{r} - \hat{\boldsymbol{\omega}}) = \mathbf{0}$$

- Thus $\hat{\boldsymbol{\omega}} = D^{\mathsf{T}}\boldsymbol{x}$ for some $\boldsymbol{x} \in C_0(\Gamma)$ (KVL)
- So $L\boldsymbol{x} = DR^{-1}\boldsymbol{r}$ where $L = DR^{-1}D^{\mathsf{T}}$ is the weighted graph Laplacian
- *Any solution to this equation will serve for the local estimation algorithm*

Local Estimation
One-dimensional Gaussian noise case

- Jacobi's Method gives a locally implementable solution
- The Laplacian can be written as $L = N - A$ where N is the degree map and A is the adjacency map for Γ
- Jacobi's method entails the iteration

$$\boldsymbol{x}_{n+1} = N^{-1}(DR^{-1}\boldsymbol{r} + A\boldsymbol{x}_n)$$

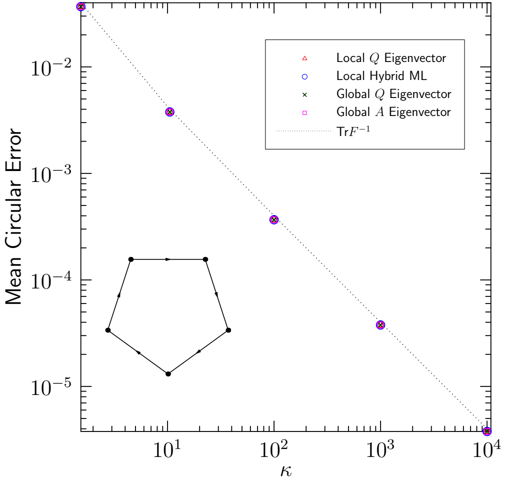
  i.e., "be the average of what your neighbors think you are"
- Jacobi's method is guaranteed to converge if L is diagonally dominant; i.e., $|L_{ii}| > \sum_{i \neq j} |L_{ij}|$
- Unfortunately $|L_{ii}| = \sum_{i \neq j} |L_{ij}|$, so convergence is not guaranteed

Other Results
Summary

- Extension to $G = \mathbb{R}^d$ – solutions have a tensor structure
- Von Mises noise on $\mathbb{T}$ – oscillator phase alignment
  - Includes localizable fast algorithms that converge to near-ML estimates in all cases tested
  - Algorithm: "Be the weighted circular mean of what your neighbors think you are" where the weights come from how well the neighbors are aligned with their neighbors
- Compact connected abelian Lie groups follow the form of $\mathbb{R}$ and $\mathbb{T}$, but suitable noise models are a problem
- Making progress in the non-abelian Lie group setting, including the important special case $G = SO(3)$
- New work focusing more on estimation of parameters from extrinsic data on a sensor network

Other Results

Performance of Fast Estimators on $\mathbb{T}$



Other Results

Performance of Fast Estimators on $\mathbb{T}$