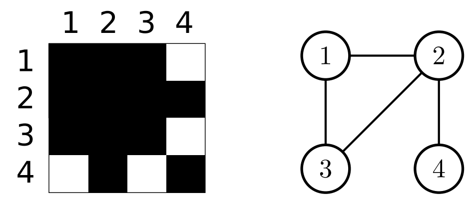
Conservative Sparsification for Efficient and Consistent Approximate Estimation

John Vial*, Hugh Durrant-Whyte† and Tim Bailey*

*Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney
2006 NSW, Australia
j.vial@acfr.usyd.edu.au

†National ICT Australia
Australian Technology Park
Level 5, 13 Garden Street
Eveleigh NSW 2015, Australia
hugh@nicta.com.au

***Abstract*— This paper presents a new technique for sparsification of the information matrix of a multi-dimensional Gaussian distribution. We call this technique Conservative Sparsification (CS) and show that it produces estimates which are consistent with respect to an optimal filter. This technique was applied to the Simultaneous Localisation and Mapping (SLAM) problem, and compared with two existing sparsification approaches; the Sparse Extended Information Filter (SEIF) and the Data Discarding Sparse Extended Information Filter (DDSEIF). Simulation demonstrates that CS is a consistent approach and provides a tighter upper bound than existing conservative methods.**



(a) Information Matrix (b) Graphical Model

Fig. 1. Example of mapping from information matrix (a) to equivalent graphical model (b), a black square indicates nonzero elements in the information matrix which correspond to conditional dependency links in the graph.

I. Introduction

The probability density function (pdf) of a Gaussian distribution is characterized by two moments, its mean (μ) and covariance ($\mathbf{\Sigma}$). A Gaussian may also be expressed using the canonical form (or information form) that specifies the information matrix $\mathbf{Y} = \mathbf{\Sigma}^{-1}$ and information vector $\mathbf{y} = \mathbf{Y}\mu$. The sparsity pattern of $\mathbf{Y}$corresponds to the conditional dependency structure of the underlying distribution. This pattern can be used to generate a graphical model which depicts the conditional relationships of the pdf. An example of this mapping is shown in Figure 1.

If a particular estimate contains very few conditional dependency links, then the graphical model and the information matrix for this estimate is sparse. Sparse estimation provides two main advantages over a dense representation; the memory required to store a sparse matrix is lower and algorithms can be designed to exploit sparsity and thus perform much faster.

One such problem which can benefit from sparsity in the information matrix is the Simultaneous Localisation and Mapping (SLAM) algorithm. Using an Extended Information Filter (EIF) for SLAM is advantageous as the update step is constant rather than quadratic in complexity. However, marginalisation can induce off-diagonal elements in the information matrix which can greatly reduce the efficiency of the filter. Thrun et. al. [1] showed that by selectively deleting these off-diagonal elements the efficiency of the filter could be retained. The process of selectively setting off-diagonal elements to zero is called sparsification.

This paper develops a new approach to sparsifying the information matrix called Conservative Sparsification (CS). This method allows any off-diagonal element in the information matrix to be set to zero, while guaranteeing that the resulting distribution is consistent with respect to the original distribution. Consistency allows the estimate to be used in future Bayesian updates without the result becoming overconfident. The problems of data association and model linearisation are not going to be addressed by this paper. This paper uses simple heuristics to decide which links to remove, since link selection is not the primary focus. The interested reader is directed to [2] for some good approaches to this problem. Developing a more principled theory with regard to good link selection for CS remains as future work.

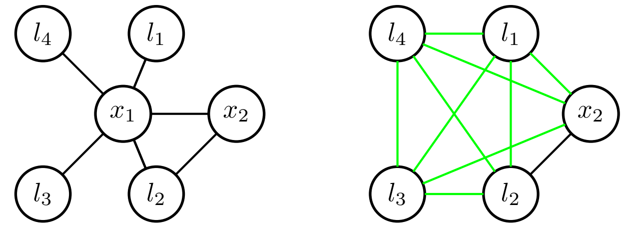
The outline of this paper is as follows; Section II describes the SLAM problem and previous approaches to sparsification. Section III discusses consistency for a Gaussian pdf, formulates the CS problem and describes some properties of CS. Section IV outlines the results of a simple SLAM simulation with respect to accuracy and consistency.

II. Background and Related Work

This section describes the SLAM problem and then several existing sparse methods.

A. *Simultaneous Localization and Mapping*

Simultaneous Localization and Mapping (SLAM) is the problem of a robot estimating its own location using estimates of its motion while also generating a map of landmarks



(a) Graph of SLAM Problem, x denotes a robot state node, while l denotes a landmark node.

(b) After marginalisation. Green links represent new connections

Fig. 2. Graphical model for the SLAM predict step. State x_1 is predicted forward to a new state, x_2. x_1 is then marginalised away, resulting in new non-zero links being created.

which it observes in its environment. A key feature of SLAM is that the robots location and landmarks are estimated jointly. The SLAM problem has a direct interpretation as a graphical model, the robot state at each time instant being a single node, while each landmark is also a single node. Links between the nodes represent condition dependency relationships. Figure 2 shows a graphical model interpretation of the SLAM problem.

The SLAM problem can be formulated in such a way that it is naturally sparse [3][4]. A graphical model of the sparse SLAM problem can be seen in Figure 2(a). This exploits the fact that landmarks are conditionally independent if we maintain estimates of all the robot poses. Unfortunately, the requirement to store a state estimate for every robot pose results in a memory footprint that grows linearly. This either requires the filter to be eventually halted or that some of the robot poses must be marginalised. Marginalisation, which is the process of removing a state from the estimation scheme, results in an effect known as fill-in. Figure 2(b) demonstrates the effect of fill-in for the SLAM problem. Here we see that previously zero off-diagonal terms have been added to the system (green links). The number of links after the marginalisation step is greater than existed beforehand, thus the problem has become less sparse. Marginalisation is explored in further detail in [5]. The current pose SLAM problem, where only the latest pose estimate is stored in the state vector, is a special case of the SLAM problem. Figure 2(b) shows that this approach will always result in a densely connected system. In this paper, we shall address the current pose SLAM problem, for which the optimal graph is dense.

Two common approaches to sparsification exploit the observation that links connecting robot poses to distant landmarks tend to have smaller values. Thrun et. al. [1] proposed a method called Sparse Extended Information Filters (SEIF) which sets the smallest of these links to zero. SEIF is discussed in detail in Section II-D.1. The Thin Tree Junction Tree by Paskin [6] also imposes sparsity by thinning cliques of states. Both of these approaches are not guaranteed to produce consistent estimates.

Covariance Intersection (CI) [7] is a method for data fusion which can be used to solve the SLAM problem [8]. This approach estimates each landmark and robot pose independently, and does so in a manner which is guaranteed to be consistent, but significantly inflates the uncertainty for the state estimates. CI is not considered further as the other approaches examined in this paper have better performance.

Walter et. al. developed another method that we call the Data Discarding Sparse Extended Information Filter (DDSEIF) [5] which selectively discards motion observations to induce sparsity, this method is described in Section II-D.2

B. Consistent Estimation

If we wish to estimate the mean ($\hat{a}$) and covariance ($\mathbf{A}$) of a Gaussian random variable, with true state ($\bar{a}$) and true error covariance $\bar{\mathbf{A}} = \mathbf{E}[(\hat{a}-\bar{a})(\hat{a}-\bar{a})]$. An estimate is considered consistent if,

$$\mathbf{E}[\hat{a} - \bar{a}] = 0, \tag{1}$$

$$\mathbf{A} \succeq \bar{\mathbf{A}}. \tag{2}$$

Here the symbol $\succeq$ implies the result of $\mathbf{A} - \bar{\mathbf{A}}$ is positive semi-definite. This definition of consistency is used by the Covariance Intersection algorithm [7], it is also used by Bar-Shalom [9, p. 237].

C. Covariance Selection and Max-Det

An approach closely related to the method used in this paper is Covariance Selection. A covariance is desired which is the *closest* to a problem-specific symmetric matrix which may have unknown elements. The resulting matrix may have some sparsity constraints imposed on its inverse covariance (information matrix). The closest matrix is measured using a loss function. Typical loss functions are the Kullback-Leibler Divergence (KLD) or the L2 norm [10]. Covariance Selection problems are not concerned with consistency, and only attempt to minimise the loss function subject to equality constraints. Therefore Covariance Selection solutions produce inconsistent estimates.

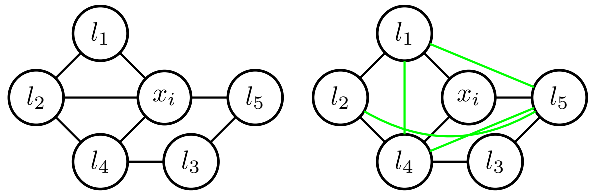
If a covariance selection problem uses only a KLD loss function then it can be written in the form of a MAXDET problem. Boyd et. al. have a solver [11] which can solve this problem in $\mathcal{O}(n^{2.5}m^2)$, where m is the number of free parameters and the size the matrices are $n \times n$. Note that for dense matrices this cost is $\mathcal{O}(n^{6.5})$.

D. Methods examined in this paper

This section describes the methods which CS will be compared to in the experimental section of this work.

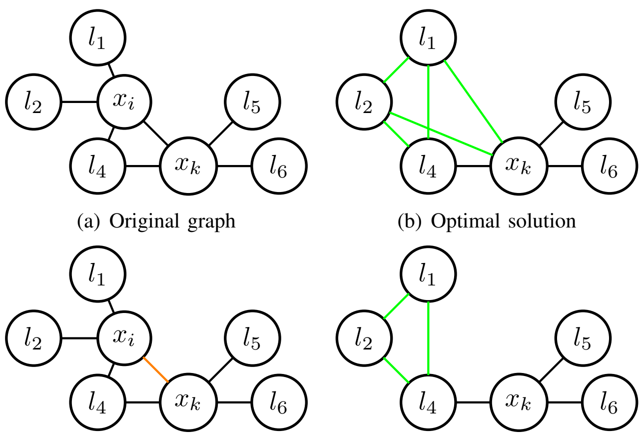
1) SEIF: The Sparse Extended Information Filter was first proposed by Thrun et. al. and applied to the SLAM problem [1]. Their approach allowed the deletion of links connecting the robot state to the set of landmarks. Although this approach allows for constant time prediction steps, it produces inconsistent estimates [12].

SEIF splits landmarks in its graph into three distinct sets. The first set are all nodes which are currently not neighbours to the robot pose, denoted by Y^0. The second set contains all the nodes which are to be disconnected from the robot pose, denoted by Y^- . The last set is the landmarks which are to remain connected to the pose, denoted by Y^+. SEIF's



(a) Original graph (b) Sparsified graph

Fig. 3. Sparsifying link between x_i and l_2 using SEIF approach, this link was induced through marginalisation of a prior state. Green links are induced by SEIF. The set Y^0 contains landmark l_3, while the set Y^- contains l_2, finally set Y^+ contains the active landmarks l_1, l_4 and l_5. Fill-in has occurred between the sparsified set Y^- and the active set Y^+. After this operation, the robot state has a reduced number of connections.



(a) Original graph (b) Optimal solution

(c) DDSEIF discards orange link (d) DDSEIF solution

Fig. 4. The optimal approach is shown in (a) and (b), while the DDSEIF approach is shown in (c) and (d). In this example the robot predicts its state forward and then observes two new landmarks (l_5 and l_6) and a previously surveyed landmark, l_4. The graph in (b) shows the network after the state x_i has been marginalised and has formed a large clique. In (c) the DDSEIF algorithm chooses to discard the motion update (represented by an orange link). Upon marginalising x_i in (d) landmarks l_1 and l_2 are not connected to state x_k. DDSEIF has induced sparsity by ignoring a motion observation.

approach allows fill-in to occur between the sets Y^+ and Y^-, while the nodes in Y^0 are completely unaffected by the operation. Although large amounts of fill-in can occur, the approach does indeed reduce the connectivity of the robot pose. Fig. 3 shows the SEIF algorithm applied to an example graph.

A SEIF implementation will only attempt to sparsify when the number of links connected to the robot pose reaches a predetermined threshold. SEIF then removes links to ensure that the number of connections to the robot pose is reduced below the threshold. Weaker links are chosen for deletion first. This approach tends to disconnect landmarks that are further away from the robot [1].

2) DDSEIF: The Data Discarding Sparse Extended Information filter approach by Walter et. al. [5] uses the insight that by selectively choosing to discard odometry data, as used by the vehicles prediction step, the sparsity of the graph can be improved. The algorithm also requires the robot to re-observe a landmark in order to reinitialise the robot pose.

In the case of the current pose SLAM problem, DDSEIF begins by observing that maintaining the most recent motion prediction step will result in the number of links to the robot state rising above a predefined threshold. This motion observation is then discarded, the previous robot pose is marginalised into the landmarks and the robot pose is reinitialised through a landmark re-observation. An example of this process is given in Fig. 4.

III. Conservative Sparsification

A. Problem Statement

We wish to find a sparse representation of the multi-dimensional Gaussian distribution $\mathcal{N}(\mathbf{y}_{tr}, \mathbf{Y}_{tr})$, the new approximate distribution will be of the form $\mathcal{N}(\mathbf{y}_{sp}, \mathbf{Y}_{sp})$. The problem is supplied with a set of links in $\mathbf{Y}_{tr}$ which are to be deleted in $\mathbf{Y}_{sp}$, and another set containing the current sparsity pattern of $\mathbf{Y}_{tr}$ which must be maintained. The new distribution associated with $\mathbf{Y}_{sp}$ must be consistent with respect to the original distribution associated with $\mathbf{Y}_{tr}$. The matrix $\mathbf{Y}_{sp}$ must also be as close as possible to the original matrix. We will also assume that $\mathbf{Y}_{tr}$ is a consistent estimate of the underlying random variables. Finally we will ensure that the mean is unchanged:

$$\mathbf{Y}_{tr}^{-1}\mathbf{y}_{tr} = \mathbf{Y}_{sp}^{-1}\mathbf{y}_{sp}. \quad (3)$$

We will approach this as a convex optimisation problem, using techniques described in [13], [14] and [15]. This problem is designed to ensure that the new covariance will represent an upper bound on the original covariance. Thus the approach is named Conservative Sparsification.

B. Problem Formulation

We choose to minimise the Kullback-Leibler Divergence (KLD) as a measure of similarity between $\mathbf{Y}_{tr}$ and $\mathbf{Y}_{sp}$ (with coincident means). The form of the KLD for multi-dimensional Gaussian distributions with coincident means is (where n is the number of dimensions),

$$\left\langle \mathbf{Y}_{tr}^{-1} || \mathbf{Y}_{sp}^{-1} \right\rangle = \frac{1}{2}\left(\ln\left(\frac{\det(\mathbf{Y}_{sp}^{-1})}{\det(\mathbf{Y}_{tr}^{-1})} \right) + \mathrm{tr}(\mathbf{Y}_{sp}\mathbf{Y}_{tr}^{-1}) - n \right). \quad (4)$$

If $\mathbf{Y}_{tr}$ remains fixed and equal to our original distribution's information matrix, then we can construct a cost function that will minimise the KLD;

$$f(\mathbf{Y}_{sp}) = \mathrm{tr}(\mathbf{Y}_{sp}\mathbf{Y}_{tr}^{-1}) - \log\,\det(\mathbf{Y}_{sp}). \quad (5)$$

Note that this is a convex function over the space of positive definite matrices. Using the definition of consistency in Section II-B, we have the consistency constraint,

$$\mathbf{Y}_{tr} - \mathbf{Y}_{sp} \succeq \mathbf{0}.$$

Using the above equations and adding equality constraints to enforce sparsity we have the following problem:

$$\text{minimise } \mathrm{tr}(\mathbf{Y}_{sp}\mathbf{Y}_{tr}^{-1}) - \log\,\det(\mathbf{Y}_{sp}),$$

subject to:

$$\mathbf{Y}_{tr} - \mathbf{Y}_{sp} \succeq 0$$
$$\mathbf{Y}_{sp_{ij}} = 0 \text{ where } (i,j) \in \mathcal{I} \cup \mathcal{J}.$$

Let $\mathcal{I}$ be the set of indices which are to be set to zero and let $\mathcal{J}$ include the indices of all the zero valued elements in $\mathbf{Y}_{tr}$. Its important to note that after a CS sparsification, the total number of non-zero elements in the information matrix will always decrease. This problem can be formulated as a MAXDET problem [11].

C. Problem Reduction

The above approach requires an expensive convex optimisation step over all elements of the matrix to be performed. This section gives a brief description of a problem reduction approach, which allows only a subset of the matrix elements to be modified while guaranteeing a solution to the full problem. The Appendix to this paper proves two key theorems required for this approach.

We require that $\mathbf{Y}_{tr}$ be partitioned as,

$$\mathbf{Y}_{tr} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{0} \\ \mathbf{B}_{12}^T & \mathbf{B}_{22} & \mathbf{B}_{23} \\ \mathbf{0} & \mathbf{B}_{23}^T & \mathbf{B}_{33} \end{bmatrix}. \quad (6)$$

We also require that the sparsification constraints only effect elements of $\mathbf{B}_{11}$. We can now marginalise the problem to the Markov blanket of the sparsification operation, setting:

$$\mathbf{Y}_{tr} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{12}^T & \mathbf{B}_{22} - \mathbf{B}_{23}\mathbf{B}_{33}^{-1}\mathbf{B}_{23}^T \end{bmatrix}. \quad (7)$$

We now define the following notation; $(\mathbf{A})_{ij}$ accesses the ith element and jth row of the matrix expression $\mathbf{A}$. We show in the Appendix, Theorem 1, that if we can solve a reduced problem, with $\mathbf{Y}_{tr}$ is as above, $\mathcal{I}$ entirely contained in $\mathbf{B}_{11}$ and the additional equality constraint:

$$(\mathbf{Y}_{sp})_{ij} = (-\mathbf{B}_{23}\mathbf{B}_{33}^{-1}\mathbf{B}_{23}^T)_{ij} \text{ for all } (i,j) \in \mathcal{J}_{22}. \quad (8)$$

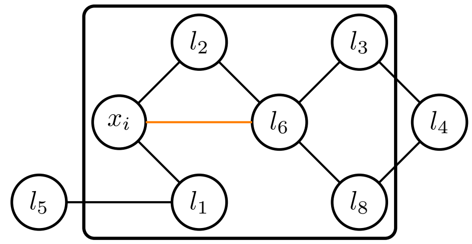
Where $\mathcal{J}_{22}$ is the subset of $\mathcal{J}$ contained in $\mathbf{B}_{22}$.

We can then transform the solution of the reduced problem to the full solution:

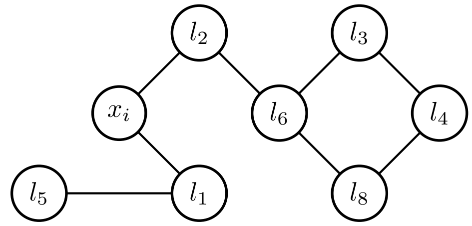
$$\mathbf{Y}_{sp}^* = \begin{bmatrix} \mathbf{B}_{11}^* & \mathbf{B}_{12}^* & \mathbf{0} \\ \mathbf{B}_{12}^{*T} & \mathbf{B}_{22}^* + \mathbf{B}_{23}\mathbf{B}_{33}^{-1}\mathbf{B}_{23}^T & \mathbf{B}_{23} \\ \mathbf{0} & \mathbf{B}_{23}^T & \mathbf{B}_{33} \end{bmatrix}, \quad (9)$$

where, $\mathbf{B}^*$ is the optimal solution for the reduced problem. Theorem 2 states that if we choose to include the Markov blanket of the sparsification in the reduced problem set, a solution will always be found. However, if a set smaller than the Markov blanket is chosen, then the reduced problem may not have a solution. Also, if the structure in Eqn. 6 cannot be found, then the reduced problem coincides with the full problem.

The marginal projection $\mathbf{B}_{23}\mathbf{B}_{33}^{-1}\mathbf{B}_{23}^T$ is required for this initial problem reduction. If it has already been computed using a sparse solve (see [16] for sparse solving algorithms) then the computation complexity for CS reduces to $\mathcal{O}(k^{2.5}l^2)$. Where, k is the number of states inside the Markov blanket and l is the number of non-zeros inside the Markov blanket. Thus the problem complexity is bounded by the size of the Markov blanket. For a large sparse matrix where $k << n$, CS will approach constant time. Fig. 5 demonstrates the effect of performing CS on the problem's structure and the zero fill in characteristic of CS.



(a) Graph before CS. Orange link to be sparsified. Box indicates Markov blanket.



(b) Graph after CS

Fig. 5. CS's approach to removing link x_i to l_6. This link was induced through fill-in after marginalising a prior state. CS has produced no additional fill-in associated with its operation. Landmarks l_4 and l_5 lie outside the Markov blanket and are unaffected by the sparsification operation. Note that choosing a set smaller than the Markov blanket is not in general guaranteed to produce a feasible reduced problem.

IV. EXPERIMENTS

A SLAM simulation was designed to compare CS to DD-SEIF and SEIF. A robot with a linear process model and linear observation model was simulated moving through a 2-dimensional environment containing 20 stationary landmarks. Linear models were chosen so that linearisation errors would not effect the results, and so that a Kalman Filter (KF) would be optimal for this simulation. The process model (Q) and observation model (R) are shown below,

$$\mathbf{Q} = \begin{bmatrix} 8.3 & 9.8 \\ 9.8 & 14.6 \end{bmatrix} \times 10^{-3}, \mathbf{R} = \begin{bmatrix} 4.5 & 2.4 \\ 2.4 & 2.7 \end{bmatrix} \times 10^{-4}.$$

Consistency is measured after each data fusion step by computing the minimum eigenvalue of the sparsity constraint $\mathbf{Y}_{tr} - \mathbf{Y}$. If the minimum eigenvalue is less than 0 then the constraint is not positive definite and thus the filter is inconsistent.

The robot performed current pose SLAM, and thus marginalised its pose estimate at every time step. During each run, the CS algorithm was called after 5 data fusion steps to sparsify the current estimate. The CS algoirthm replicated the sparsity pattern of DD-SEIF at each sparsity step. DD-SEIF's algorithm was called when the robot was linked to more than 2 landmarks, while SEIF's algorithm was called when the robot was connected to more than 3 landmarks.

The simulation was ended when the robot had moved 200 times, resulting in 259 landmark observations. Each filter's performance was recorded after each data fusion step.

A. Results

Figure 6(a) shows that all of the sparse approaches maintained a similar number of non-zeros during the simulation.

This allows a fair comparison of the KLD between the algorithms since each filter modeled a system with similar complexity.

Figure 6(b) shows the KLD between the Kalman Filter and each of the sparse approaches. CS maintains a smaller KLD than DD-SEIF throughout the simulation, and thus is the most accurate consistent sparse filter in this simulation. SEIF is clearly the most accurate filter in this simulation, however from Figure 6(c) we see that its minimum eigenvalue quickly drops below zero, and it is thus inconsistent. Peaks in this graph are due to sparsification steps increasing the KLD with subsequent data fusion steps quickly reducing the KLD.

V. Conclusion

Conservative Sparsification is a new approach for sparsely approximating multi-dimensional Gaussian distributions, enabling cheaper storage and faster inference. Approximations by CS always represent an upper bound on the optimal covariance, while leaving the mean unchanged. In sparse systems where the marginal projections are readily available, the cost of computing CS is bounded by the size of the Markov blanket, and for very large, sparse matrices, this cost is constant time. This paper has demonstrated CS for SLAM. Our results show that CS provides a tighter upper bound than existing conservative methods such as DD-SEIF. In future work we will investigate principled link deletion strategies and the application of CS to large-scale decentralised estimation problems, such as cooperative localisation, distributed tracking and sensor networks.

Acknowledgements

This work is supported by the Australian Centre for Field Robotics and the Australian New South Wales State Government.

Appendix

This appendix proves two key theorems which are vital for the problem reduction described in Section III-B. We define the following notation; $(\mathbf{A})_{ij}$ accesses the ith element and jth row of the matrix expression $\mathbf{A}$. Let the matrix $\mathbf{E}_{nm}$ be known as the indicator matrix, defined as:

$$(\mathbf{E}_{nm})_{ij} = \begin{cases} \sqrt{2}, & \text{if } (i \neq j) \text{ and} \\ & ((n,m) = (i,j) \text{ or } (n,m) = (j,i)) \\ 1, & \text{if } (i = j) \text{and} (n = i) \text{and} (m = j) \\ 0, & \text{otherwise.} \end{cases}$$

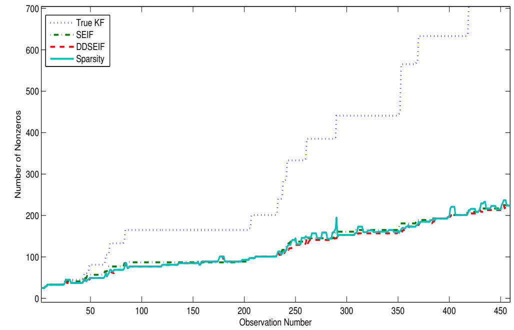
We will also define a mapping from $\mathbb{R}^n$ to $\mathbb{S}^{m\times m}$ (where $\mathbb{S}^{m\times m}$ is the set of real symmetric matrices):

$$F(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^{n} x_i \mathbf{E}_{qp}, \text{where } (q,p) \in \mathcal{Z} \qquad (10)$$

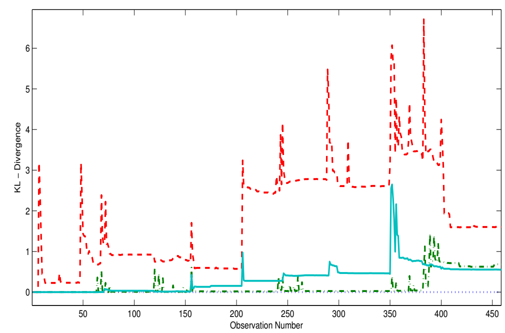
where, $\mathcal{Z}$ is a set of indices in a $m \times m$ matrix space, $\mathbf{E}_{qp}, \mathbf{F}_0 \in \mathbb{S}^{m\times m}$ and $(\mathbf{F}_0)_{ij} = 0$ for all $(i,j) \in \mathcal{Z}$.

We now define the CS problem:

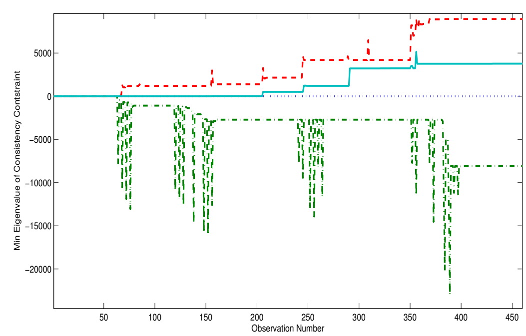
$$\underset{\mathbf{Y}_{sp}}{\arg\min} \;\; \text{tr}(\mathbf{Y}_{sp}\mathbf{Y}_{tr}^{-1}) - \log\, \det(\mathbf{Y}_{sp})$$



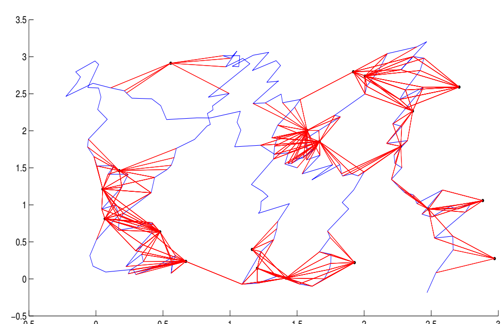
(a) Sparsity of each filter



(b) Kullback-Leibler Divergence for each estimation scheme.



(c) Minimum eigenvalue of consistency constraint $(\mathbf{Y}_{tr} - \mathbf{Y})$, a value below zero indicates an inconsistent filter.



(d) Robot true path during experiment shown in blue. Black dots indicate landmarks, red lines indicate robot-landmark observations.

Fig. 6. Results from experiment. Here we see CS (light blue Sparsity plot) compared with DD-SEIF and SEIF

subject to:

$$\mathbf{Y}_{tr} - \mathbf{Y}_{sp} \succeq 0$$
$$\mathbf{Y}_{sp_{ij}} = 0 \text{ for all } (i,j) \in \mathcal{I} \cup \mathcal{J},$$

where, $\mathbf{Y}_{tr}, \mathbf{Y}_{sp} \in \mathbb{S}_{+}^{m \times m}$, $\mathcal{I}$ contains the set of indices which are to be sparsified, while $\mathcal{J}$ contains the set of indices which satisfy $(\mathbf{Y}_{tr})_{ij} = 0$. Here $\mathbb{S}_{+}^{m \times m}$ is the cone of $m \times m$ positive definite matrices.

We will also define another set of indices $\mathcal{K}$, which contains the indicies of all the elements which are to have a nonzero value in the final solution $\mathcal{K} \notin \mathcal{I} \cup \mathcal{J}$.

We can remove the equality constraints by performing a parameterisation using Eqn. 10. Let,

$$\begin{aligned}\mathbf{Y}_{sp} &= F(\mathbf{x}) \\ &= \mathbf{F}_0 + \sum_{i=1}^{n} x_i \mathbf{E}_{qp}, \text{where } (q,p) \in \mathcal{K}\end{aligned}$$

and, $\mathbf{F}_0 = \mathbf{0}$.

We can now define the CS problem using the new parameterisation:

$$\arg\min_{F(\mathbf{x})} \ \operatorname{tr}(F(\mathbf{x})\mathbf{Y}_{tr}^{-1}) - \log\det(F(\mathbf{x}))$$

subject to:

$$\mathbf{Y}_{tr} - F(\mathbf{x}) \succeq 0.$$

The Lagrange dual function for this problem is:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \begin{aligned}&\operatorname{tr}(F(\mathbf{x})\mathbf{Y}_{tr}^{-1}) - \log\det(F(\mathbf{x})) \\ &\quad + \operatorname{tr}(\boldsymbol{\lambda}(F(\mathbf{x}) - \mathbf{Y}_{tr})).\end{aligned} \quad (11)$$

Using the Lagrange function we can form the first KKT equation,

$$\begin{aligned}\frac{d}{d\mathbf{x}}\big(\operatorname{tr}(F(\mathbf{x})\mathbf{Y}_{tr}^{-1}) - \log\det(F(\mathbf{x}))\big) \\ + \frac{d}{d\boldsymbol{\lambda}}(\operatorname{tr}(\boldsymbol{\lambda}(F(\mathbf{x}) - \mathbf{Y}_{tr})))\end{aligned} = \mathbf{F}_{tr}^{-1} - \mathbf{F}_{sp}^{-1} + \mathbf{F}_{\lambda}. \quad (12)$$

where,

$$\mathbf{F}_{tr}^{-1} = \begin{cases} (\mathbf{Y}_{tr}^{-1})_{ij} & \text{for all } (i,j) \in \mathcal{K} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

$$\mathbf{F}_{sp}^{-1} = \begin{cases} (F(\mathbf{x})^{-1})_{ij} & \text{for all } (i,j) \in \mathcal{K} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

$$\mathbf{F}_{\lambda} = \begin{cases} (\boldsymbol{\lambda})_{ij} & \text{for all } (i,j) \in \mathcal{K} \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

It is sufficient for the optimal solution to this problem to satisfy the following equations (known as the KKT conditions):

$$\mathbf{F}_{tr}^{-1} - \mathbf{F}_{sp}^{-1} + \mathbf{F}_{\lambda} = 0 \quad (16)$$
$$\mathbf{Y}_{tr} - F(\mathbf{x}) \succeq 0 \quad (17)$$
$$\boldsymbol{\lambda} \succeq 0 \quad (18)$$
$$\operatorname{tr}(\boldsymbol{\lambda}(F(\mathbf{x}) - \mathbf{Y}_{tr})) = 0. \quad (19)$$

The KKT conditions only hold under the assumption of strong duality. Slater's constraint states that a convex problem has strong duality if its inequality constraints hold strictly. For this problem, Slater's constraint holds if there exists a positive number ϵ, which satisfies the following (where $\mathbf{I}$ is the identity matrix):

$$\mathbf{Y}_{tr} - \epsilon\mathbf{I} \succ \mathbf{0}. \quad (20)$$

Since this is true for any matrix as long as $\mathbf{Y}_{tr} \succ 0$, the assumption of strong duality is valid for this problem.

Defining a Graph Cut: Consider a new operation, the *CS Graph Cut* which can operate on any CS problem. This operation splits states in the problem into two disjoint sets of arbitrary size. The only constraint on a CS Graph Cut is that the full set of sparsification indices $\mathcal{I}$ must be included in only one of the sets. We can express this using matrix partitions as below:

$$\mathbf{Y}_{tr} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12}^T \\ \mathbf{A}_{12} & \mathbf{A}_{22} \end{bmatrix}, F(\mathbf{x}) = \begin{bmatrix} \mathbf{A}_{11}^* & \mathbf{A}_{12}^{*\,T} \\ \mathbf{A}_{12}^* & \mathbf{A}_{22}^* \end{bmatrix},$$

where $\mathbf{A}_{11}$ contains the set of indices $\mathcal{I}$. We will also define another set of indices $\mathcal{L}$, which is the set of indices which satisfies,

$$(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T)_{ij} \neq 0.$$

We will define two sets of indices related to the cut. Let $\mathcal{J}_{11}$ be the subset of $\mathcal{J}$ which is contained in the cut $\mathbf{A}_{11}$, and let $\mathcal{K}_{11}$ be the subset of indices contained in the cut $\mathbf{A}_{11}$. It is important to recognise that the CS Graph Cut is not unique, there may be many valid cuts, all that is required is that the split is disjoint and indices $\mathcal{I}$ are entirely included in one of the sets.

Finally we will define the mapping from $\mathbb{R}^t$ to $\mathbb{S}^{u \times u}$:

$$G(\mathbf{x}_{11}) = \mathbf{G}_0 + \sum_{p=1}^{t} x_p \mathbf{E}_{i,j}, \quad (21)$$

where, $(i,j) \in \mathcal{K}_{11}, \mathbf{E} \in \mathbb{S}^{u \times u}, u$ is the size of the graph cut, and t is the number of free parameters in the cut, $\mathbf{x}_{11} \in \mathbb{R}^t$.

Theorem 1: [1] For any CS problem, choose a valid CS graph cut for which the following problem is feasible.
Let

$$\tilde{\mathbf{Y}}_{tr} = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T \quad (22)$$

$$(\mathbf{G}_0)_{ij} = \begin{cases} (-\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T)_{ij} & \text{for } (i,j) \in \mathcal{L} \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

Now solve the problem (known as the reduced problem):

$$\arg\min_{\mathbf{x}_{11}} \ \operatorname{tr}(G(\mathbf{x}_{11})\tilde{\mathbf{Y}}_{tr}^{\,-1}) - \log\det(G(\mathbf{x}_{11}))$$

[1]Note that Theorem 1 is similar to the result in [2, Sect. 6.2] as both show that the KLD can be computed locally. Our theorem differs, as we show that the consistency constraint can also be optimised locally, and we do not require our final solution to be a Junction Tree (which requires a matrix ordering to be chosen and can cause fill in).

subject to:

$$\tilde{\mathbf{Y}}_{tr} - \mathbf{G}(\tilde{\mathbf{x}}_{11}) \succeq 0$$

Let $\tilde{\mathbf{x}}_{11}$ and $\tilde{\boldsymbol{\lambda}}_{11}$ be the optimal solutions to the reduced problem. We will then consider the optimal solution to the full problem to be:

$$F^*(\tilde{\mathbf{x}}) = \begin{bmatrix} \mathbf{G}(\tilde{\mathbf{x}}_{11}) + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \quad (24)$$

$$\boldsymbol{\lambda}^* = -\mathbf{Y}_{tr}^{-1} + F^*(\tilde{\mathbf{x}})^{-1} \quad (25)$$

Proof Outline: This proof will show that the KKT conditions hold for the proposed solutions.

Proof: If $\tilde{\mathbf{x}}_{11}$ and $\tilde{\boldsymbol{\lambda}}_{11}$ are the optimal solutions to the reduced problem, then the following KKT equations hold.

$$\frac{d}{d\mathbf{x}}\begin{pmatrix} \text{tr}(\mathbf{G}(\tilde{\mathbf{x}}_{11})\mathbf{Y}_{tr}^{-1}) \\ -\log\det(\mathbf{G}(\tilde{\mathbf{x}}_{11})) \\ +\text{tr}(\tilde{\boldsymbol{\lambda}}(\mathbf{G}(\tilde{\mathbf{x}}_{11}) - \mathbf{Y}_{tr})) \end{pmatrix} = \mathbf{G}_{tr}^{-1} - \mathbf{G}_{sp}^{-1} + \mathbf{G}_{\lambda}.$$

$$\frac{d}{d\mathbf{x}}(\boldsymbol{\lambda}) = \mathbf{F}_{\lambda}$$

$$= -\mathbf{F}_{tr}^{-1} + \mathbf{F}_{sp}^{-1} \quad (26)$$

$$\mathbf{G}_{tr}^{-1} = \begin{cases} (\tilde{\mathbf{Y}}_{tr}{}^{-1})_{ij} \text{ for all } (i,j) \in \mathcal{K} \\ 0, \text{otherwise} \end{cases} \quad (27)$$

$$\mathbf{G}_{sp}^{-1} = \begin{cases} (\mathbf{G}(\tilde{\mathbf{x}}_{11})^{-1})_{ij} \text{ for all } (i,j) \in \mathcal{K} \\ 0, \text{otherwise} \end{cases} \quad (28)$$

$$\mathbf{G}_{\lambda} = \begin{cases} (\boldsymbol{\lambda})_{ij} \text{ for all } (i,j) \in \mathcal{K} \\ 0, \text{otherwise.} \end{cases} \quad (29)$$

Where,

$$\mathbf{G}_{tr}^{-1} - \mathbf{G}_{sp}^{-1} + \mathbf{G}_{\lambda} = 0 \quad (30)$$

$$\tilde{\boldsymbol{\lambda}}_{11} \succeq 0 \quad (31)$$

$$\text{tr}(\tilde{\boldsymbol{\lambda}}(\mathbf{G}(\tilde{\mathbf{x}}_{11}) - \mathbf{Y}_{tr})) = 0 \quad (32)$$

$$\mathbf{Y}_{tr} - \mathbf{G}(\tilde{\mathbf{x}}_{11}) \succeq 0. \quad (33)$$

Also, a solution to the reduced problem must lie in its domain. That is, the following must be true,

$$\mathbf{G}(\tilde{\mathbf{x}}_{11}) \succ \mathbf{0}. \quad (34)$$

Show that $F^*(\tilde{\mathbf{x}}) \succ 0$:
Using partition in Eqn. 24 and since $\mathbf{G}(\tilde{\mathbf{x}}_{11}) \succ 0$ and $\mathbf{A}_{22}^{-1} \succ \mathbf{0}$ then,

$$\mathbf{G}(\tilde{\mathbf{x}}_{11}) + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T \succ \mathbf{0}. \quad (35)$$

Also,

$$\mathbf{G}(\tilde{\mathbf{x}}_{11}) + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T = \mathbf{G}(\tilde{\mathbf{x}}_{11}) \succ \mathbf{0}, \quad (36)$$

which implies $F^*(\tilde{\mathbf{x}}) \succ 0$.

Show KKT Condition (Equation 16):

$$\mathbf{F}_{tr}^{-1} - \mathbf{F}_{sp}^{-1} + \mathbf{F}_{\lambda} = \mathbf{F}_{tr}^{-1} - \mathbf{F}_{sp}^{-1} + (-\mathbf{F}_{tr}^{-1} + \mathbf{F}_{sp}^{-1}) \quad (37)$$

$$= \mathbf{0}, \quad (38)$$

where the above has used Eqn. 25

Show KKT Condition (Equation 17):

$$\begin{aligned} -F^*(\mathbf{x}_{11}) + \mathbf{Y}_{tr} &= -\begin{bmatrix} \mathbf{F}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12}^T \\ \mathbf{A}_{12} & \mathbf{A}_{22} \end{bmatrix} \\ &= \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T - \mathbf{G}(\tilde{\mathbf{x}}_{11})) \\ &= \tilde{\mathbf{Y}}_{tr} - \mathbf{G}(\tilde{\mathbf{x}}_{11})) \\ &\succeq \mathbf{0}, \end{aligned}$$

where $\mathbf{F}_{11} = \mathbf{G}(\tilde{\mathbf{x}}_{11}) + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T$.

Show KKT Condition (Equation 18):

$\mathbf{Y}_{tr} - F(\mathbf{x}) \succeq \mathbf{0} \iff F(\mathbf{x})^{-1} - \mathbf{Y}_{tr}^{-1} \succeq 0$,
therefore,

$$\boldsymbol{\lambda} = F(\mathbf{x})^{-1} - \mathbf{Y}_{tr}^{-1} \quad (39)$$

$$\boldsymbol{\lambda} \succeq 0. \quad (40)$$

Show KKT Condition (Equation 19)

$$\text{tr}\left(\boldsymbol{\lambda}\begin{pmatrix} F^*(\mathbf{x}_{11}) \\ -\mathbf{Y}_{tr} \end{pmatrix}\right) = \text{tr}\left(\boldsymbol{\lambda}\begin{pmatrix} \mathbf{G}(\tilde{\mathbf{x}}_{11}) + \\ \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T \\ -\mathbf{A}_{11} \end{pmatrix}\right) \quad (41)$$

$$= \text{tr}\left(\boldsymbol{\lambda}_{11}(\mathbf{G}(\tilde{\mathbf{x}}_{11}) - \tilde{\mathbf{Y}}_{tr})\right). \quad (42)$$

Where $\boldsymbol{\lambda}_{11}$ is the top left block of $\boldsymbol{\lambda}$. From Eqn. 30 we have

$$\mathbf{G}_{\lambda} = \mathbf{G}_{sp}^{-1} - \mathbf{G}_{tr}^{-1}.$$

Decomposing $\boldsymbol{\lambda}_{11}$:

$$\boldsymbol{\lambda}_{11} = \begin{cases} (\mathbf{G}_{\lambda})_{ij} \text{ for } (i,j) \notin \mathcal{J}_{11} \cup \mathcal{K}_{11} \\ (\boldsymbol{\lambda}_0)_{ij} \text{ for } (i,j) \in \mathcal{J}_{11} \cup \mathcal{K}_{11}, \end{cases}$$

where,

$$\boldsymbol{\lambda}_0 = \begin{cases} (\tilde{\boldsymbol{\lambda}})_{ij} \text{for} (i,j) \in \mathcal{J}_{11} \cup \mathcal{K}_{11} \\ 0 \text{ otherwise.} \end{cases}$$

Also, note that

$$(\tilde{\boldsymbol{\lambda}}(\mathbf{G}(\tilde{\mathbf{x}}_{11}) - \tilde{\mathbf{Y}}_{tr})))_{ij} = (\mathbf{G}_{\lambda}(\mathbf{G}(\tilde{\mathbf{x}}_{11}) - \tilde{\mathbf{Y}}_{tr}))_{ij} = 0, \quad (43)$$

for all $(i,j) \in \mathcal{J}_{11} \cup \mathcal{K}_{11}$.
Now, Eqn 42 becomes:

$$\begin{aligned} \text{tr}(\boldsymbol{\lambda}(F^*(\mathbf{x}_{11}) - \mathbf{Y}_{tr}) &= \text{tr}((\boldsymbol{\lambda}_0 + \mathbf{G}_{\lambda})(\mathbf{G}(\tilde{\mathbf{x}}_{11}) - \tilde{\mathbf{Y}}_{tr})) \\ &= \text{tr}\begin{pmatrix} \boldsymbol{\lambda}_0\begin{pmatrix} \mathbf{G}_0 + \mathbf{F}_0 \\ +\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T \end{pmatrix} \\ +\text{tr}(\mathbf{G}_{\lambda}(\mathbf{G}(\tilde{\mathbf{x}}_{11}) - \tilde{\mathbf{Y}}_{tr})) \end{pmatrix} \\ &= \text{tr}\left(\boldsymbol{\lambda}_0\begin{pmatrix} \mathbf{0} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T \\ +\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T \end{pmatrix}\right) \\ &= \mathbf{0}, \end{aligned}$$

where the above has decomposed the problem into matrices based on the elements which are in or not in $\mathcal{J}_{11} \cup \mathcal{K}_{11}$ respectively. Then using Eqn. 43, 23 and 32, we have the above result.

Since the KKT equations are satisfied, the proposed solution is a optimal. ■

Theorem 2: If the Markov blanket is wholly included in one of the sets of a CS Graph cut, then the associated reduced problem will always have a feasible solution.

Proof Outline: This proof first describes the inequalities which can result in an infeasible solution. It then describes a choice of the reduction which will always result in a feasible solution. A problem is known as infeasible if there are no solutions which satisfy the equality and inequality constraints.

Proof: We denote a solution to the reduced problem by, $\mathbf{G}(\tilde{\mathbf{x}}_{11}) = \tilde{\mathbf{A}}_{11}$. The inequality for the reduced problem corresponding to the consistency constraint is:

$$\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T - \tilde{\mathbf{A}}_{11} \succeq \mathbf{0} \quad (44)$$

$$\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T \succeq \tilde{\mathbf{A}}_{11} \succ \mathbf{0}. \quad (45)$$

The subtraction of the term $(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T)$ reduces the upper bound on the solution $\tilde{\mathbf{A}}_{11}$, if $\mathcal{I} \cap \mathcal{L} \neq \emptyset$, then it is possible that the problem will have no solution (i.e. infeasible).

Exploiting Structure: Consider a matrix $\mathbf{Y}_{tr}$ with the following structure:

$$\mathbf{Y}_{tr} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{0} \\ \mathbf{B}_{12}^T & \mathbf{B}_{22} & \mathbf{B}_{23} \\ \mathbf{0} & \mathbf{B}_{23}^T & \mathbf{B}_{33} \end{bmatrix}. \quad (46)$$

If we choose to reduce the problem so that $\tilde{\mathbf{A}}_{11}$ encompasses $\mathbf{B}_{11}$ and $\mathbf{B}_{22}$ (The Markov blanket of the equality constraints contained in $\mathbf{B}_{11}$), then the above inequality becomes:

$$\begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{12}^T & \mathbf{B}_{22} - \mathbf{B}_{23}\mathbf{B}_{33}^{-1}\mathbf{B}_{23}^T \end{bmatrix} \succeq \begin{bmatrix} \tilde{\mathbf{B}}_{11} & \tilde{\mathbf{B}}_{12} \\ \tilde{\mathbf{B}}_{12}^T & \tilde{\mathbf{B}}_{22} \end{bmatrix} \quad (47)$$

$$\succ \mathbf{0}. \quad (48)$$

If this structure exists for a particular reduction choice, then this problem will always have a feasible solution as long as $\mathcal{I}$ is contained in $\mathbf{B}_{11}$, and $\mathcal{L}$ is contained in $\mathbf{B}_{22}$. A feasible solution is demonstrated below (where ϵ satisfies the $\mathbf{B}_{11} \succ \epsilon\mathbf{I}$)

$$\tilde{\mathbf{A}}_{11} = \begin{bmatrix} \epsilon\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{22} - \mathbf{B}_{23}\mathbf{B}_{33}^{-1}\mathbf{B}_{23}^T \end{bmatrix}. \quad (49)$$

This satisfies the equality and inequality constraints, the inequality constraint is shown below:

$$\begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{12}^T & \mathbf{B}_{22} - \mathbf{B}_{23}\mathbf{B}_{33}^{-1}\mathbf{B}_{23}^T \end{bmatrix} \succeq \begin{bmatrix} \epsilon\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{22} - \mathbf{B}_{23}\mathbf{B}_{33}^{-1}\mathbf{B}_{23}^T \end{bmatrix} \quad (50)$$

$$\succeq \mathbf{0}. \quad (51)$$

Note that we cannot choose $\tilde{\mathbf{B}}_{22} = \epsilon\mathbf{I}$ as the equality constraints associated with $\mathcal{L}$ require off diagonal terms to be non-zero.

Therefore, the reduced problem will always have a feasible solution if the Markov blanket is included in the reduced problem.

■

References

[1] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *The International Journal of Robotics Research*, vol. 23, no. 7-8, p. 693, 2004.

[2] U. Kjærulff, "Approximation of bayesian networks through edge removals," Institute for Electronic Systems, Department of Mathematics and Computer Science, Aalborg University, Tech. Rep. IR-93-2007, 1993.

[3] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, p. 1181, 2006.

[4] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1100–1114, 2006.

[5] M. Walter, R. Eustice, and J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *The International Journal of Robotics Research*, vol. 26, no. 4, p. 335, 2007.

[6] M. Paskin, "Thin junction tree filters for simultaneous localization and mapping," *Computer*, 2002.

[7] S. Julier and J. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *American Control Conference, 1997. Proceedings of the 1997*, vol. 4. IEEE, 2002, pp. 2369–2373.

[8] ——, "Using covariance intersection for SLAM," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 3–20, 2007.

[9] Y. Bar-Shalom and X. Li, "Estimation and tracking- Principles, techniques, and software," *Norwood, MA: Artech House, Inc, 1993.*, 1993.

[10] E. Cripps, C. Carter, and R. Kohn, "Variable selection and covariance selection in multivariate regression models," *Handbook of Statistics*, vol. 25, pp. 519–552, 2005.

[11] S. Wu, L. Vandenberghe, and S. Boyd, "MAXDET: software for determinant maximization problems," *Information Systems Laboratory, Stanford University*, 1996.

[12] R. Eustice, M. Walter, and J. Leonard, "Sparse extended information filters: Insights into sparsification," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3281–3288.

[13] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.

[14] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of semidefinite programming: theory, algorithms, and applications*. Springer Netherlands, 2000.

[15] J. Dattorro, *Convex optimization & Euclidean distance geometry*. Meboo Publishing USA, 2005.

[16] A. Björk, "Numerical methods for least squares problems." *SIAM*, 1996.