ANPL | Autonomous Navigation and Perception Lab

Computationally Efficient Belief Space Planning via Augmented Matrix Determinant Lemma and Re-Use of Calculations

Dmitry Kopitkov and Vadim Indelman

TECHNION Israel Institute of Technology

Information-Theoretic BSP - Problem Types

- Unfocused BSP – reduce uncertainty of all variables

$$J_{H}(a) = dim.const - \frac{1}{2}\ln\left|\Lambda_{k+L}\right| \rightarrow O(N^3) \text{ complexity!!!}$$

- Focused BSP – reduce uncertainty of a subset of variables

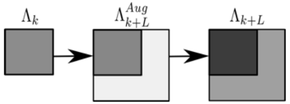
$$J_{H}^{F}(a) = H(X_{k+L}^{F}) = dim.const + \frac{1}{2}\ln\left|\Sigma_{k+L}^{F}\right|$$

Requires Schur complement. Even more expensive!!!

- Existing approaches: per **each** action propagate **posterior belief**, compute determinants of **huge** matrices
- **Expensive** for high-dimensional state spaces!

Addressed Setting – Augmented BSP

- Candidate actions may introduce new factors and **new state variables** (e.g. new robot poses)
- Propagation of posterior information matrix:



$$\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + A^T \cdot A$$

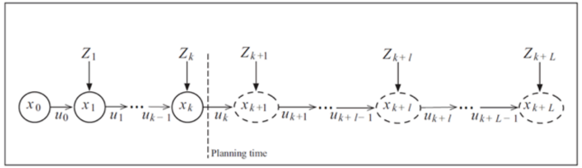
- **More complex** scenario w.r.t not-augmented case where action introduces only new factors (handled in previous work)

Problem Formulation

- Consider state vector $X_k \in \mathbb{R}^n$ at time t_k
  - e.g. history of robot poses, landmarks, etc.
  - n can be huge (> 10000), for example..
- Consider its belief $b[X_k] = N(X_k^*, \Sigma_k)$
- Consider candidate actions $\mathcal{A} \doteq \{a_1, a_2, \ldots, a_N\}$
- Each candidate a_i provides different posterior belief $b[X_{k+L} | a_i]$
- The goal is to choose optimal action according to some objective:

$$a^* = \underset{a \in A}{\operatorname{argmin}} J(a)$$

- Example from mobile robotics domain:
  - Given action $a = u_{k:k+L-1}$ and new observations $Z_{k+1:k+L}$, future belief is:



(Image is taken from Indelman15ijrr)

$$b[X_{k+L}] = p(X_{k+L}|Z_{0:k+L}, u_{0:k+L-1}) \propto p(X_k|Z_{0:k}, u_{0:k-1}) \prod_{l=k+1}^{k+L} \underbrace{p(x_l|x_{l-1}, u_{l-1})}_{\text{motion model}} \underbrace{p(Z_l|X_l^o)}_{\text{measurement model}}$$

» L is planning horizon

Key Ideas

- Develop and Use **Augmented Matrix Determinant Lemma (AMDL)**
  - A new variant of known Matrix Determinant Lemma for augmented matrices
- Use **Information Gain (IG)**
  - Mathematically identical to posterior entropy
  - Usually can be calculated more efficiently than posterior entropy
- Exploit **Sparsity of Jacobian** matrix of new factors
- **Re-use** of calculations between actions

Contributions

- Computationally-efficient **information-theoretic BSP** approach
  - **Without** posterior propagation for each candidate action
  - **Avoid** calculating determinants of large matrices
  - Calculation **Re-use**
- Per-action evaluation **does not depend on state** dimension
- **Exact** and **general** solution
- Approach addresses both *unfocused* and *focused* cases

ANPL Autonomous Navigation and Perception Lab

Computationally Efficient Belief Space Planning via Augmented Matrix Determinant Lemma and Re-Use of Calculations

Dmitry Kopitkov and Vadim Indelman

TECHNION Israel Institute of Technology

Problem Formulation

- This work – information-theoretic objectives
  - (Differential) Entropy – measures uncertainty of estimation
    $$H(X) = -\int_X p(x)\cdot \log p(x)dx$$
  - BSP Information term (Unfocused):
    - (Differential) Entropy:
      $$J_H(a) = H(b[X_{k+L}]) \quad a^* = \underset{a\in A}{\text{argmin}}\, J_H(a)$$
    - Information Gain:
      $$J_{IG}(a) = H(b[X_k]) - H(b[X_{k+L}]) \quad a^* = \underset{a\in A}{\text{argmax}}\, J_{IG}(a)$$
    - Mathematically identical
    - Each can be computationally preferable in different scenarios

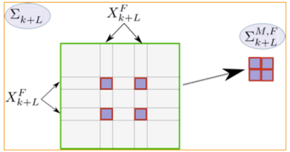
- Assuming Gaussian Distributions
- Objectives for Unfocused BSP:

$$J_H(a) = dim.const - \frac{1}{2}\ln|\Lambda_{k+L}|, \quad J_{IG}(a) = dim.const + \frac{1}{2}\ln\frac{|\Lambda_{k+L}|}{|\Lambda_k|}$$

$O(N^3)$ complexity!!!

- Where
  - Λ_k is prior information matrix
  - $\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + A^T \cdot A$ is posterior information matrix (A: Action Jacobian)

- Focused setting:
  - Consider focused variables $X_{k+L}^F \subseteq X_{k+L}$
  - Its posterior marginal covariance: $(\Sigma_{k+L} = \Lambda_{k+L}^{-1})$

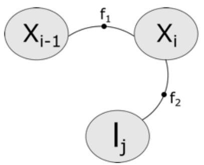


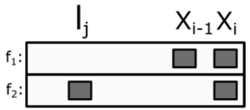
  - Measure the posterior information (entropy, IG) for these variables:

$$J_H^F(a) = H(X_{k+L}^F) = dim.const + \frac{1}{2}\ln|\Sigma_{k+L}^{M,F}|$$
$$J_{IG}^F(a) = H(X_k^F) - H(X_{k+L}^F) = \frac{1}{2}\ln\frac{|\Sigma_k^{M,F}|}{|\Sigma_{k+L}^{M,F}|}$$

Jacobian Structure Sparsity

- Matrix A is Jacobian of **new** factors, with dimension $m \times N$
- Its rows represent **new** factors (measurements)
- Its columns represent state variables (old and new)
- Only variables *involved* in new factors will have non-zero columns in A
- Typically m and number of *involved* variables is very small





Key Ideas

- Develop and Use **Augmented Matrix Determinant Lemma (AMDL)**
  - A new variant of known Matrix Determinant Lemma for augmented matrices
- Use **Information Gain (IG)**
  - Mathematically identical to posterior entropy
  - Usually can be calculated more efficiently than posterior entropy
- Exploit **Sparsity of Jacobian** matrix of new factors
- **Re-use** of calculations between actions

Contributions

- Computationally-efficient **information-theoretic BSP** approach
  - **Without** posterior propagation for each candidate action
  - **Avoid** calculating determinants of large matrices
  - Calculation **Re-use**
- Per-action evaluation **does not depend on state** dimension
- **Exact** and **general** solution
- Approach addresses both *unfocused* and *focused* cases

ANPL Autonomous Navigation and Perception Lab

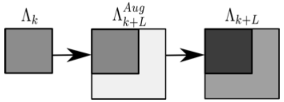
Computationally Efficient Belief Space Planning via Augmented Matrix Determinant Lemma and Re-Use of Calculations

Dmitry Kopitkov and Vadim Indelman

TECHNION Israel Institute of Technology

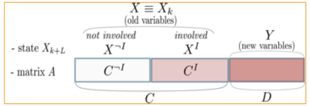
Augmented Matrix Determinant Lemma (AMDL)

- Augmented case (new variables were introduced by action):



$$\Lambda_{k+L} = \Lambda^{Aug}_{k+L} + A^T \cdot A$$

- Partitioning of:

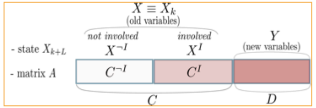


- We developed AMDL:

$$\frac{|\Lambda_{k+L}|}{|\Lambda_k|} = |\Delta| \cdot |D^T \cdot \Delta^{-1} \cdot D|, \quad \Delta = I_m + C \cdot \Sigma_k \cdot C^T$$

Unfocused IG – Final Expression

- Consider state partitioning:



- Final expression for IG:

$$J_{IG}(a) = dim.const + \frac{1}{2}\ln|P| + \frac{1}{2}\ln|D^T \cdot P^{-1} \cdot D|$$

$$P = I_m + C^I \cdot \Sigma_k^{X^I} \cdot (C^I)^T$$

- Calculation complexity depends on m, $dim(X^I)$ and $dim(Y)$
- Given $\Sigma_k^{X^I}$, does not depend on state dimension N
- Only **few entries** from the prior covariance are actually required!
- Very cheap

Calculation Re-use

$$J_{IG}(a) = dim.const + \frac{1}{2}\ln|P| + \frac{1}{2}\ln|D^T \cdot P^{-1} \cdot D|, \quad P = I_m + C^I \cdot \Sigma_k^{X^I} \cdot (C^I)^T$$

- **Note**:
  - We can **avoid** posterior propagation and determinants of **large** matrices
  - Calculation of action impact **does not depend** on N
  - Still, we need $\Sigma_k^{X^I}$
  - Different candidate actions often **share** many *involved* variables X^I

- **We propose re-use of calculation**:
  - Combine variables *involved* in all candidate actions into set $X_{All} \subseteq X_k$
  - Perform one-time calculation of $\Sigma_k^{X_{All}}$ (depends on N)
  - Calculate $J_{IG}(a)$ for each action, using $\Sigma_k^{X_{All}}$

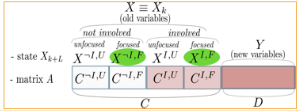
Focused Setting

- Different cases:
  1. $X^F_{k+L} \subseteq Y$, for example robot last pose
  2. $X^F_{k+L} \subseteq X$, for example mapped landmarks
  3. $X^F_{k+L} \subseteq \{X \cup Y\}$, hard to find example

- We handle first 2 cases

Focused Setting $X^F_{k+L} \subseteq X$

- Consider state partitioning:



- IG of X^F_{k+L}:

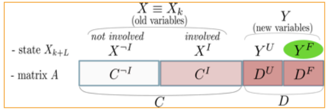
$$J^F_{IG}(a) = \frac{1}{2}(\ln|P| + \ln|D^T \cdot P^{-1} \cdot D| - \ln|S| - \ln|D^T \cdot S^{-1} \cdot D|)$$

$$P = I_m + C^I \cdot \Sigma_k^{X^I} \cdot (C^I)^T, \quad S = I_m + C^{I,U} \cdot \Sigma_k^{X^{I,U}|F} \cdot (C^{I,U})^T$$

- Calculation complexity depends on m, $dim(X^I)$ and $dim(Y)$
- Given $\Sigma_k^{X^I}$ and $\Sigma_k^{X^{I,U}|F}$, does not depend on state dimension N

Focused Setting $X^F_{k+L} \subseteq Y$

- Consider state partitioning:



- Posterior Entropy of X^F_{k+L}:

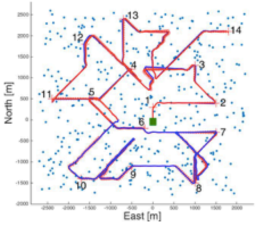
$$J^F_{H}(a) = dim.const + \frac{1}{2}\ln|(D^U)^T \cdot P^{-1} \cdot D^U| - \frac{1}{2}\ln|D^T \cdot P^{-1} \cdot D|$$

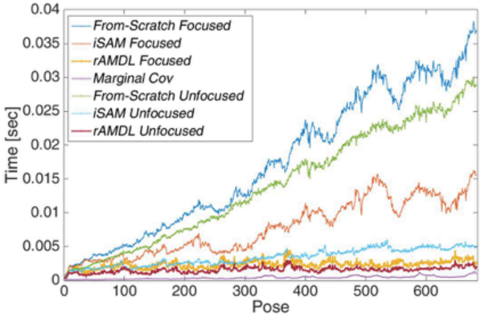
$$P = I_m + C^I \cdot \Sigma_k^{X^I} \cdot (C^I)^T$$

- Calculation complexity depends on m, $dim(X^I)$ and $dim(Y)$
- Given $\Sigma_k^{X^I}$, does not depend on state dimension N
- Only **few entries** from the prior covariance are actually required!
- Very cheap

Application to Autonomous Navigation in Unknown Environment

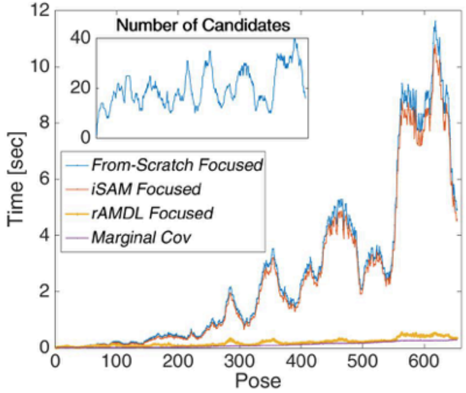
- Significant time reduction in *Focused* case – focus on robot's last pose x_{k+L}





Application to Autonomous Navigation in Unknown Environment

- Significant time reduction in *Focused* case – focus on mapped landmarks



Conclusions

rAMDL (**Re**-use with **A**ugmented **M**atrix **D**eterminant **L**emma):

- Exact (identical to original objectives)
- General (any measurement model)
- Per-candidate complexity does not depend on state dimension
- Unfocused and Focused problem formulations
- Applicable to Sensor Deployment, Measurement Selection, Graph Sparsification, Active SLAM and many more..