



A Tutorial on Graph-Based SLAM

はじめに

- SLAMへのアプローチ

- フィルタリングアプローチ (オンラインSLAMと呼ばれる)

システムの状態が現在のロボットの位置と地図で構成されるオンライン状態推定として問題をモデル化する

例) カルマン情報フィルタ

パーティクルフィルタ

- 平滑化アプローチ

測定した情報セットからロボットの完全な軌道を推定する。これらの手法は **最小二乗問題**として扱われる。



はじめに

- グラフベースSLAM
 - 頂点: ロボットの姿勢・ランドマーク
 - 辺 : センサデータ

しかし、センサデータは誤差を含む

⇒ **誤差最小化問題**となる



SLAMの定式化

$$p(x_{1:T}, m | z_{1:T}, u_{1:T}, x_0)$$

$x_{1:t} = \{x_1, \dots, x_t\}$: ロボットの軌道

$u_{1:t} = \{u_1, \dots, u_t\}$ オドメトリ

$z_{1:t} = \{z_1, \dots, z_t\}$: 環境情報

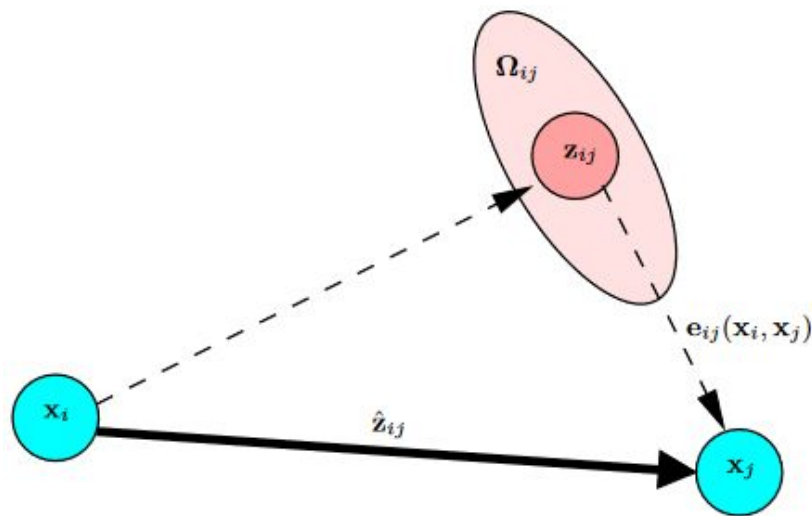
m : 地図

オドメトリ・環境情報・現在地が与えられたときの軌道と地図を推定する



グラフベースSLAM

グラフのエッジを定義するための関数とその量



x_i と x_j を接続する図

$x = (x_1, \dots, x_T)$: ロボットの姿勢

z_{ij}, Ω_{ij} : ノード i, j を接続するための仮定の測定値の入った情報行列

$\hat{z}_{ij}(x_i, x_j)$: ノード x_i, x_j から測定値を予測する

$e(x_i, x_j, z_{ij})$ 誤差関数



目的関数

$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} \underbrace{\mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}}_{\mathbf{F}_{ij}},$$

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \mathbf{F}(\mathbf{x}).$$

- 目的
 - 観測値の負の対数尤度 $\mathbf{F}(\mathbf{x})$ を最小にするノード \mathbf{x}^* の構成を見つけることである
- 手法
 - 反復的局所線形化による誤差最小化 を行う
例) ガウス・ニュートン法
Levenberg-Marquardtアルゴリズム
※ロボットの姿勢の初期推定値 $\check{\mathbf{x}}$
既知である必要がある



反復的局所線形化による誤差最小化（ 1 / 5 ）

現在の初期推定値 $\check{\mathbf{x}}$ を中心とした一次テイラー展開によって誤差関数を近似する

$$\mathbf{e}_{ij}(\check{\mathbf{x}}_i + \Delta \mathbf{x}_i, \check{\mathbf{x}}_j + \Delta \mathbf{x}_j) = \mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x}) \quad (6)$$

$$\simeq \mathbf{e}_{ij} + \mathbf{J}_{ij} \Delta \mathbf{x}. \quad (7)$$




反復的局所線形化による誤差最小化 (2 / 5)

(7)の式をF(x)の式に代入する

$$\mathbf{F}_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x})$$
$$= \mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x})^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x}) \quad (8)$$

$$\simeq (\mathbf{e}_{ij} + \mathbf{J}_{ij} \Delta \mathbf{x})^T \boldsymbol{\Omega}_{ij} (\mathbf{e}_{ij} + \mathbf{J}_{ij} \Delta \mathbf{x}) \quad (9)$$

$$= \underbrace{\mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}}_{c_{ij}} + 2 \underbrace{\mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{b}_{ij}} \Delta \mathbf{x} + \Delta \mathbf{x}^T \underbrace{\mathbf{J}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{H}_{ij}} \Delta \mathbf{x} \quad (10)$$

$$= c_{ij} + 2\mathbf{b}_{ij} \Delta \mathbf{x} + \Delta \mathbf{x}^T \mathbf{H}_{ij} \Delta \mathbf{x} \quad (11)$$


反復的局所線形化による誤差最小化 (3 / 5)

局所近似により次のように書き換えることが出来る。

そして、 $c = \sum c_{ij}, b = \sum b_{ij}, H = \sum H_{ij}$ することで (13) \Rightarrow (14) と置き換わる

$$\mathbf{F}(\check{\mathbf{x}} + \Delta \mathbf{x}) = \sum_{\langle i, j \rangle \in \mathcal{C}} \mathbf{F}_{ij}(\check{\mathbf{x}} + \Delta \mathbf{x}) \quad (12)$$

$$\simeq \sum_{\langle i, j \rangle \in \mathcal{C}} c_{ij} + 2\mathbf{b}_{ij} \Delta \mathbf{x} + \Delta \mathbf{x}^T \mathbf{H}_{ij} \Delta \mathbf{x} \quad (13)$$

$$= c + 2\mathbf{b}^T \Delta \mathbf{x} + \Delta \mathbf{x}^T \mathbf{H} \Delta \mathbf{x}. \quad (14)$$



反復的局所線形化による誤差最小化（ 4 / 5 ）

Λx を最小化することで解くことが出来る

$$\mathbf{H} \Delta \mathbf{x}^* = -\mathbf{b}. \quad (15)$$



反復的局所線形化による誤差最小化（ 5 / 5 ）

行列Hは測定誤差をヤコビ行列を通して軌道の空間に投影することが出来る。構造上、Hは疎な行列であり、制約によって接続されたポーズには0がない。非ゼロのブロック数は制約の数の2倍である。これにより式 (15)を**Sparse Cholesky**により解くことが出来る。線形化された解は、計算された増分を初期推測に加えることで得られる。

$$\mathbf{x}^* = \check{\mathbf{x}} + \Delta \mathbf{x}^*. \quad (16)$$

一般的なガウス・ニュートン法は (14)の線形化、(15)の解、(16)の更新の繰り返しである。各反復では前回の結果が用いられる。上記の手順は一般的なアプローチであり、パラメータ \mathbf{x} がユークリッドであることを前提としている。そのため、SLAMでは有効でない可能性もある。



線形化されたシステムへの考察（ 1 / 3 ）

式(14)から、行列Hとベクトルbは、制約ごとに1つずつの行列とベクトルを合計することで得られる。全ての制約条件はシステムに負荷項をもたらす。この加算による構造は誤差関数のヤコビ行列に依存する。制約の誤差関数は 2つのノードの値にのみ依存するため、(7)のヤコビ行列は次のような形になる。

$$\mathbf{J}_{ij} = \begin{pmatrix} 0 \cdots 0 & \underbrace{\mathbf{A}_{ij}}_{\text{node } i} & 0 \cdots 0 & \underbrace{\mathbf{B}_{ij}}_{\text{node } j} & 0 \cdots 0 \end{pmatrix}. \quad (17)$$



線形化されたシステムへの考察（ 2 / 3 ）

ここで、 A_{ij} と B_{ij} は、 x_i と x_j に対する誤差関数の微分である。式 (10)から、ブロック行列 H_{ij} の構造は次のようになる。

(見やすさの為に0は省略している。)

$$\mathbf{H}_{ij} = \begin{pmatrix} \ddots & & & & \\ & \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij} & \cdots & \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{B}_{ij} & \\ & \vdots & \ddots & \vdots & \\ & \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij} & \cdots & \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{B}_{ij} & \\ & & & & \ddots \end{pmatrix} \quad (18)$$

$$\mathbf{b}_{ij} = \begin{pmatrix} \vdots \\ \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij} \\ \vdots \\ \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij} \\ \vdots \end{pmatrix} \quad (19)$$



線形化されたシステムへの考察 (3 / 3)

アルゴリズム1

アルゴリズム1は、ロボットの姿勢に関する事後情報の平均値と情報行列を決定する反復的なガウス・ニュートン法を要約したものである。システムのほとんどの構造は疎であるため、**メモリ効率の良い表現を使用したほうが良い**。ヘッセ行列の構造は、グラフの接続性から事前に知られているので、反復の最初に一度だけヘッセ行列を事前に割り当て、新たな線形化が必要なときに、すべてのエッジをループしてその場で更新したほうが良い。各エッジは、ブロック $H_{[ii]}$ 、 $H_{[ij]}$ 、 $H_{[ji]}$ 、 $H_{[jj]}$ と、係数ベクトルのブロック $b_{[i]}$ および $b_{[j]}$ に寄与する。さらに、 H の上三角部は対称的なので、上三角部のみを計算するという最適化も行われている。アルゴリズム1では、一般性を損なわないように、最初のノード x_1 を固定する。ポーズ・グラフの特定のノードを固定する別の方法は、式15の線形システムの k 番目のブロックの行と k 番目のブロックの列を抑制することである。

```
Require:  $\check{x} = \check{x}_{1:T}$ : 初期予測,  $\mathcal{C} = \{ \langle e_{ij}(\cdot), \Omega_{ij} \rangle \}$ :  
制約  
Ensure:  $x^*$ : 解,  $H^*$ : 新たな情報行列  
// 尤度が最大の解を求める  
while  $\neg \text{converged}$  do  
   $b \leftarrow 0$    $H \leftarrow 0$   
  for all  $\langle e_{ij}, \Omega_{ij} \rangle \in \mathcal{C}$  do  
    // ヤコビ行列  $A_{ij}$ ,  $B_{ij}$  及び誤差関数を計算する  
     $A_{ij} \leftarrow \left. \frac{\partial e_{ij}(x)}{\partial x_i} \right|_{x=\check{x}}$    $B_{ij} \leftarrow \left. \frac{\partial e_{ij}(x)}{\partial x_j} \right|_{x=\check{x}}$   
    // この制約条件の線形システムへの寄与度を計算  
     $H_{[ii]} += A_{ij}^T \Omega_{ij} A_{ij}$    $H_{[ij]} += A_{ij}^T \Omega_{ij} B_{ij}$   
     $H_{[ji]} += B_{ij}^T \Omega_{ij} A_{ij}$    $H_{[jj]} += B_{ij}^T \Omega_{ij} B_{ij}$   
    // 係数ベクトルの算出  
     $b_{[i]} += A_{ij}^T \Omega_{ij} e_{ij}$    $b_{[j]} += B_{ij}^T \Omega_{ij} e_{ij}$   
  end for  
  // 最初のノードを固定する  
   $H_{[11]} += I$   
  // Sparse Cholesky を用いて線形システムを解く  
   $\Delta x \leftarrow \text{solve}(H \Delta x = -b)$   
  // パラメータを更新  
   $\check{x} += \Delta x$   
end while  
 $x^* \leftarrow \check{x}$   
 $H^* \leftarrow H$   
// 最初のノードを解放(?) release the first node  
 $H_{[11]} -= I$   
return  $\langle x^*, H^* \rangle$ 
```



多様体における最小二乗法（ 1 / 3 ）

- SLAMの地図が扱う空間の要素

- 並進ベクトル
 - 回転成分
- 回転成分は非ユークリッド空間の存在である

➡ 多様体上で最適化を行う

多様体: 大局的には必ずしもユークリッドではないが、局所的にはユークリッドとみなすことができる数学的空間

基礎となる空間を多様体と考え、ユークリッド空間の局所的な変化 Δx を多様体上の変化 $\Delta x \mapsto x \boxplus \Delta x$ にマッピングする演算子 \boxplus を定義する。



多様体における最小二乗法（ 2 / 3 ）

演算子 \boxplus を用いて、新しい誤差関数を次のように定義することができる。

$$\check{e}_{ij}(\Delta\tilde{\mathbf{x}}_i, \Delta\tilde{\mathbf{x}}_j) \stackrel{\text{def.}}{=} \mathbf{e}_{ij}(\check{\mathbf{x}}_i \boxplus \Delta\tilde{\mathbf{x}}_i, \check{\mathbf{x}}_j \boxplus \Delta\tilde{\mathbf{x}}_j) \quad (20)$$

$$= \mathbf{e}_{ij}(\check{\mathbf{x}} \boxplus \Delta\tilde{\mathbf{x}}) \simeq \check{e}_{ij} + \tilde{\mathbf{J}}_{ij} \Delta\tilde{\mathbf{x}}, \quad (21)$$

式(9)に式(21)を挿入することができ、次のような連立方程式が導かれる。

$$\tilde{\mathbf{H}} \Delta\tilde{\mathbf{x}}^* = -\tilde{\mathbf{b}}. \quad (26)$$

増分 $\Delta\tilde{\mathbf{x}}^*$ は、初期推測値 $\check{\mathbf{x}}$ の局所的なユークリッド環境で計算されるため、 \boxplus 演算子によって元のオーバーパラメトリック空間に再マッピングする必要がある。したがって、式 (16)は次のようになる。

$$\mathbf{x}^* = \check{\mathbf{x}} \boxplus \Delta\tilde{\mathbf{x}}^*. \quad (27)$$



多様体における最小二乗法 (3 / 3)

アルゴリズム2

多様体上での最小化問題の形式化

1. 式(26)によって初期推測の周りの局所ユークリッド近似で増分のセットを計算する。
2. 式(27)によって大域的な非ユークリッド空間で増分を累積することである。アルゴリズム 2は、SLAMのためのガウス・ニュートン法の多様体バージョンである。

多様体問題のヘッセ行列 \tilde{H} は、もはや軌道の情報行列ではなく、軌道の増分 $\tilde{\Delta}x$ の情報行列を表している。軌道の情報行列を得るために、アルゴリズム 2はポーズ x の元の空間で H を計算する

Require: $\tilde{x} = \tilde{x}_{1:T}$: 初期予測 $\mathcal{C} = \{(\mathbf{e}_{ij}(\cdot), \Omega_{ij})\}$:

制約

Ensure: x^* : new solution, \tilde{H}^* : new information matrix

尤度が最大となる解を探索する

while $\neg \text{converged}$ **do**

多様体上の補助ヤコビ行列 M を計算する

for all $\tilde{x}_i \in \tilde{x}$ **do**

$M_i \leftarrow \frac{\tilde{x}_i \oplus \Delta \tilde{x}_i}{\Delta \tilde{x}_i} \Big|_{\Delta \tilde{x}=0}$

end for

$\tilde{b} \leftarrow 0$ $\tilde{H} \leftarrow 0$

for all $(\mathbf{e}_{ij}, \Omega_{ij}) \in \mathcal{C}$ **do**

任意関数の計算に用いるヤコビ行列 A_i の計算

$A_{ij} \leftarrow \frac{\partial \mathbf{e}_{ij}(x)}{\partial x_i} \Big|_{x=\tilde{x}}$ $B_{ij} \leftarrow \frac{\partial \mathbf{e}_{ij}(x)}{\partial x_j} \Big|_{x=\tilde{x}}$

ヤコビ行列を多様体上に投影

$\tilde{A}_{ij} \leftarrow A_{ij} M_i$ $\tilde{B}_{ij} \leftarrow B_{ij} M_j$

非ゼロブロックのヘッセ行列を計算

$\tilde{H}_{[ii]} += \tilde{A}_{ij}^T \Omega_{ij} \tilde{A}_{ij}$ $\tilde{H}_{[ij]} += \tilde{A}_{ij}^T \Omega_{ij} \tilde{B}_{ij}$

$\tilde{H}_{[ji]} += \tilde{B}_{ij}^T \Omega_{ij} \tilde{A}_{ij}$ $\tilde{H}_{[jj]} += \tilde{B}_{ij}^T \Omega_{ij} \tilde{B}_{ij}$

任意ベクトルを計算する

$\tilde{b}_{[i]} += \tilde{A}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}$ $\tilde{b}_{[j]} += \tilde{B}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}$

end for

最初のノードを固定する

$H_{[11]} += I$

線形最小二乗問題を解く

$\Delta \tilde{x} \leftarrow \text{solve}(\tilde{H} \Delta \tilde{x} = -\tilde{b})$

バリエーションを更新

for all $\tilde{x}_i \in \tilde{x}$ **do**

$\tilde{x}_i \leftarrow \tilde{x}_i \oplus \Delta \tilde{x}_i$

end for

end while

$x^* \leftarrow \tilde{x}$

最大値が見つかったら、今度は元の空間でヘッセ行列を計算する

$H^* \leftarrow 0$

for all $(\mathbf{e}_{ij}, \Omega_{ij}) \in \mathcal{C}$ **do**

$H_{[ii]} += A_{ij}^T \Omega_{ij} A_{ij}$ $H_{[ij]} += A_{ij}^T \Omega_{ij} B_{ij}$

$H_{[ji]} += B_{ij}^T \Omega_{ij} A_{ij}$ $H_{[jj]} += B_{ij}^T \Omega_{ij} B_{ij}$

end for

return $\langle x^*, H^* \rangle$



实践

2D レーザーベースの地図生成(1 / 2)

アルゴリズム

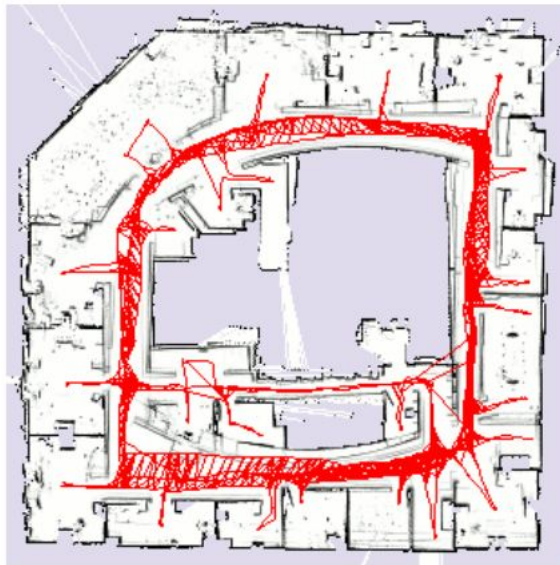
1. ロボットが0.5メートル以上移動したり、0.5ラジアン(28.6度)以上回転したりするたびにグラフに新しい頂点を追加し、現在のレーザー観測値をラベル付けする。
2. レーザースキャンを先に取得したオドメトリ推定値と照合し、対応するエッジをグラフに追加する。
3. **ループ閉じ込み**を行い、スキャンと過去の測定値の一致を求める。現在の観測値と他のノードの観測値との間のマッチングが成功した場合、グラフに新しいエッジを追加する。このエッジには、2つのスキャンが最もよく重なり合うような相対的な変換がラベル付けされている。
4. ループ閉じ込みが検知されるたびに最適化を行う。



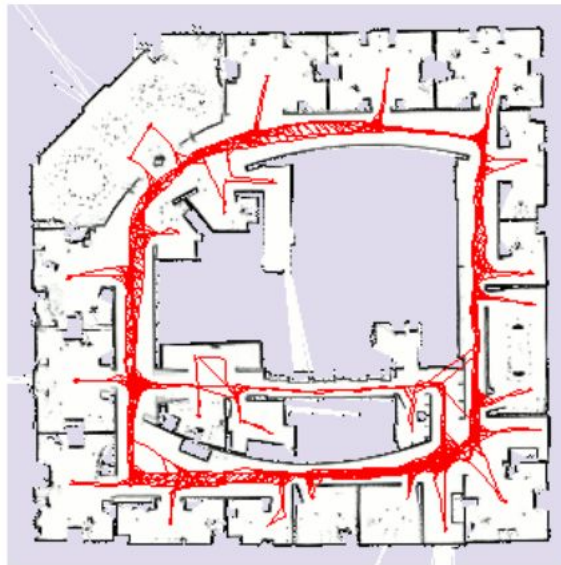
2D レーザーベースの地図生成(2 / 2)

実行結果

- グラフは1,802個のノードと3,546個のエッジで構成
- Intel Core2@2.4 GHz で100ミリ秒で最適化を行うことができた



左:最適化前



右:最適化後

3D レーザーベースの地図生成(1 / 2)

SLAMアルゴリズムの3Dへの拡張

- SE(3)の要素を表現するのに必要な最小のパラメータ数は6である(3次元の並進ベクトルと3つのオイラー角)。
 - このパラメータ化を利用すると、アルゴリズムAが得られる。
 - しかし、この最小の表現には特異点があり、パラメータ化された状態空間を利用する必要がある。

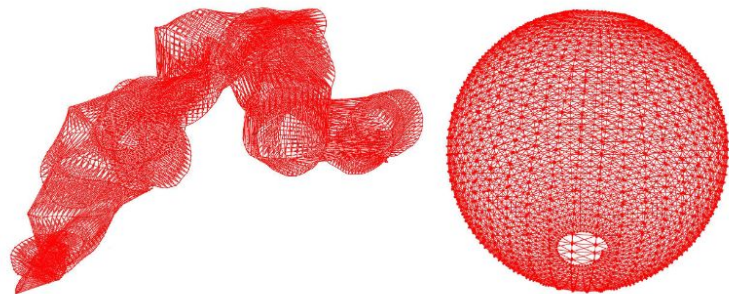
➡ アルゴリズム1とアルゴリズム2を比較する



3D レーザーベースの地図生成(2 / 2)

球体表面の走行シミュレーション

測定値には大きな誤差があり、オドメトリ情報を使用してシステムを初期化した結果、左部分に示すグラフが得られた。この初期推測から出発して、ガウス・ニュートン・アルゴリズムを、多様体の線形化を行った場合と行わなかった場合(すなわちオイラー角を使用するか否か)比較しながら実行した。まず、どちらのアプローチも誤差を減らすことができた。しかし、特異点を適切に考慮しないと、アルゴリズム1は発散してしまうが、アルゴリズム2は正しい解に収束する。



左: 初期設定

右: 最適化を行った



結論

結論

本稿では、グラフベースのSLAMに関するチュートリアルを紹介した。その目的は、提案された手法を簡単に実装できるよう、読者に十分な詳細と洞察を提供することである。本論文で紹介したアルゴリズムでも、十分なパフォーマンスを発揮するが、これらのアルゴリズムを更に最適化すると、更に驚くほど大きな問題を扱うことができる。

