大規模疎行列と共役勾配法に関するノート

平成 19 年 5 月 30 日

1 疎行列

先ず, 疎行列とは何かを明確にしよう. 素朴に言って疎行列とはその行列のサイズに比べて非零成分の数が非常に少ない行列である.

例えば $N \times N$ 行列 $\mathbf{A}$ は当然 N^2 の成分を持つが, その非零成分の数が N に比例する様な行列を疎行列と言う.

この様な言及では特定の行列が疎か否かと言う問いはナンセンスである. つまり, 疎行列とは (例えばある種の問題の過程に現れる) 行列の集まりについての概念である. そこで行列の集合 $\mathcal{A}$ を考え, 以下の様に疎行列の概念を定義しよう. $A \in \mathcal{A}$ のサイズを $N(A)$ で表すとする.

Definition 1 ある定数 K が有って, 任意の $(a_{ij}) = A \in \mathcal{A}$ に対し

$$\#\{a_{ij} \neq 0\} \leq KN(A)$$

なる時, $\mathcal{A}$ は K-疎行列集合であると言う. [1]

1.0.1 計算機と疎行列

明らかに通常の行列として扱う方が容易であり, アルゴリズムの複雑性を考慮して[2] なお利益がある場合のみ疎行列としての特殊な扱いをするべきである. つまり, 計算機において疎行列が有用であるのは, 巨大なサイズの行列を扱う場合である.

計算機において, 疎行列を疎行列として扱う事による利点は 2 点ある. メモリの節約とアルゴリズムの速度である.

例えば, サイズ $1,000,000$ の行列[3]は 10^{12} 個の要素を含み, 要素毎に 2 バイト使う事にすれば, ”1 つの行列変数” に 2 テラバイトのメモリを要する. ところが, 扱う行列集合が静的な 20-疎行列集合であると判っていれば 40 メガバイトに節約できる.

一方, 上記のサイズの行列を係数行列とする連立方程式を解くためには, ガウスの方法で凡そ $4 * 10^{17}$ 回の乗算を要する. 毎秒 1 ギガ回の乗算が可能なマシンでも $4 * 10^8$ 秒 (12 年以上) 要する. 扱う行列集合が, 自己共役正定値な 20-疎行列集合であると判っていれば, 共役勾配法により凡そ 10^{13} 回の乗算になり, 同じマシンで 10^4 秒 (3 時間弱) で済む.

この様な場合, もはや疎行列は避けて通れないと言えよう.

---

[1] 明らかに $\max_{A \in \mathcal{A}} N(A) \leq K$ のときはナンセンスであり, 実際上は, 殆どの $A \in \mathcal{A}$ が $N(A) >> K$ なる時に興味がある.

[2] バグの可能性, メンテナンス性等. また, あまりに小さな行列の場合, 単純にメモリや速度の面でさえ通常のコードに劣る場合もある.

[3] この数字はそれ程驚くべきものではない. 例えば各方向に 100 刻みの格子法による 3 次元有限要素法に現れる行列のサイズは凡そ $5,000,000$ である.

1.1 疎行列の分類

疎行列集合は, その非零成分の位置が”既知”か否かによって 2 種に大別できる. 例えば,

$$a_{ij} = 0 \quad \text{if } |i-j| > L$$

を満たすとき, $A = (a_i j)$ を $2L+1$ 重対角行列と呼ぶ事にすれば, K 重対角行列の集まりは K-疎行列集合である. この場合, 非零成分の位置は既知であると言えよう. もっと一般に

Definition 2 ある関数 $f : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ が与えられて, 任意の $(a_{ij}) = A \in \mathcal{A}$ に対し

$$a_{ij} = 0 \quad \textit{if } f(i, j) = 0$$

なる時, $\mathcal{A}$ は f によって静的であると言う.

現実的には, 静的に扱うためには判定関数 f が十分容易に計算できる事が要求される.

扱う行列集合が静的と見なせるか否かは, 計算機における行列の表現に大きく影響する.

1.2 疎行列集合の例

最も重要な疎行列集合は K 重対角行列の成すものであろう. [4] この疎行列集合はガウスの消去法の過程の操作において K 重対角行列である事を保つという点で著しく重要である.

しかしながら, 如何なる行列集合を扱うかは問題の要請するところであり, 通常我々には選択権が無い事が多い. 不幸にして問題の与える行列集合が K 重対角行列でない場合, 例えばガウスの消去法等は使えない (過程で疎でない行列が出現し, 疎行列である利点が無い). そのような場合でも, 正定値自己共役であることが保証できれば, [5] 疎行列である事を生かして効率化する事の出来る連立方程式の解法アルゴリズムが存在する. それこそが**共役勾配法**と呼ばれるものであり, 本稿のもう一つの主題である.

2 計算機上の扱い

以下, $\mathcal{A}$ を K-疎行列集合, A は $A \in \mathcal{A}$ なる $N \times N$ 行列とする.

2.1 大規模疎行列の保持とメモリ

$N \times N$ 行列 $\mathbf{A}$ の保持には通常 N^2 単位のメモリが必要であるが, 疎行列の場合, 予め $\mathbf{A}$ の殆どの成分がゼロである事が判っているので非零成分の位置と値を保持すればよい.

特に, 静的である場合 $KN\times$ 要素辺りのメモリのサイズで良い.

```
double A[N][K];
```

動的である場合, 非零成分の位置を記憶するメモリを要する.

```
struct matrixI{
    double val[KN]; // kn 番目の非零成分の値
    int    i[KN];   // kn 番目の非零成分の行
    int    j[KN];   // kn 番目の非零成分の列
}
```

行列としての操作性を良くするには

```
struct matrixII{
    double (*val)[N]; // n 行目, k 番目の非零成分の値
    int (*j)[N];      // n 行目, k 番目の非零成分の列
    int K[N];         // n 行目の非零成分の数
}
```

[4]例えば 1 次元有限要素法が帰着される連立方程式の係数行列は 3-重対角である.

[5]ある種の方程式に対する有限要素法が帰着される連立方程式の係数行列は正定値自己共役である. そして 2 次元以上の場合 K-重対角とは言いがたい.

2.2 疎行列とベクトルの積

通常 $\mathbf{y} \leftarrow \mathbf{Ax}$ の計算には N^2 回の乗算が必要である.

```
for(i=0;i<N;i++){
    y[i] = 0;
    for(j=0;j<N;j++) y[i] += A[i][j] * x[j];
}
```

然るに, 疎行列の場合, y[i] += 0 * x[j]; という計算は省略することができるので, 計算量を著しく減らすことができる. 例えば 3-重対角行列の場合は

```
y[0] = A[0][0] * x[0] + A[0][1] * x[1];
for(i=1;i<N-1;i++){
    y[i] = A[i][i-1]*x[i-1] + A[i][i]*x[i] + A[i][i+1]*x[i+1];
}
y[N-1] = A[N-1][N-2] * x[N-2] + A[N-1][N-1] * x[N-1];
```

凡そ $3N$ 回の乗算で $\mathbf{y} \leftarrow \mathbf{Ax}$ の計算ができる. 一般に K-重対角行列の場合は凡そ KN 回の乗算で済む.

この事情は k 重対角行列でなくても同じで, 少々汚いが

```
struct matrixII A;
for(i=0;i<N;i++){
   y[i] = 0;
   for(l=0;l<A.K[i];l++) y[i] += *(A.val[i]+A.j[l]) * x[A.j[l]];
}
```

とできる.[6]

2.3 連立方程式の解法 (共役勾配法)

K 重対角の場合は他稿に任せるとして,[7] ここでは, 不幸にして K 重対角では無いが, 正定値自己共役が保証されている場合を扱う.

以下, $\mathbf{A}$ は正定値, 自己共役 (正則) であるとする. このとき,

$$\mathbf{Ax} = \mathbf{b}$$

の解 $\mathbf{x}$ は $\mathrm{span}[\mathbf{b}, \mathbf{Ab}, \mathbf{A}^2\mathbf{b}, ...] =: \mathcal{K}$ なる空間に含まれる[8]ことに注意する. 任意の内積 $(\cdot,\cdot)_*$ に対し,

グラムシュミットの直交化

$$\begin{aligned} \mathbf{p}_0 &:= \mathbf{b} \\ \mathbf{p}_k &:= \mathbf{Ap}_{k-1} - \sum_{l<k} \frac{(\mathbf{Ap}_{k-1}, \mathbf{p}_l)_*}{(\mathbf{p}_l, \mathbf{p}_l)_*} \mathbf{p}_l \quad \text{for } 0 < k < \dim \mathcal{K} \end{aligned}$$

によって, $\mathcal{K}$ の $(\cdot,\cdot)_*$-直交基底を求める事ができ, [9]

[6] ここは, どうしても少々ダーティに成らざるを得ない. 判定関数 f の結果をべた書きする必要がある (毎回判定していたのでは乗算するよりも遥かに時間がかかる!!) からである. (それでも少々汚すぎる気もするが, まあそこは偉い人に任せよう)

[7] ガウスの消去法が可能な場合はそちらを使うべきである. ガウスと共役勾配ではオーダーこそ等しいが係数ではガウスの方が遥かに早い. 共役勾配法の利点は”K 重対角でない”行列にも適用できる事である.

[8] $\mathcal{K}$ は高々N 次元であるので, ある $\alpha_1, \ldots, \alpha_N$ があって $\mathbf{b} = \alpha_1\mathbf{Ab} + \cdots + \alpha_{N^2}\mathbf{A}^N\mathbf{b}$ とでき, 故に $\mathbf{x} = \alpha_1\mathbf{b} + \alpha_2\mathbf{Ab} + \cdots + \alpha_N\mathbf{A}^{N-1}\mathbf{b}$.

[9] $\mathbf{p}_{\dim\mathcal{K}} = \mathbf{0}$ となる

パーセバルの等式

$$\mathbf{x} = \sum_{k=0}^{M-1} \frac{(\mathbf{x}, \mathbf{p}_k)_*}{(\mathbf{p}_k, \mathbf{p}_k)_*} \mathbf{p}_k$$

が成り立つ．ただし $M := \dim \mathcal{K} \leq N$.

特に $\mathbf{A}$ が正定値である事を仮定すれば $(\cdot, \cdot)_{\mathbf{A}} := (\mathbf{A}\cdot, \cdot)$ は内積を定め,

$$\mathbf{x} = \sum_{k=0}^{M-1} \frac{(\mathbf{x}, \mathbf{p}_k)_{\mathbf{A}}}{(\mathbf{p}_k, \mathbf{p}_k)_{\mathbf{A}}} \mathbf{p}_k = \sum_{k=0}^{M-1} \frac{(\mathbf{b}, \mathbf{p}_k)}{(\mathbf{A}\mathbf{p}_k, \mathbf{p}_k)} \mathbf{p}_k \tag{1}$$

更に $\mathbf{A}$ に通常の内積 $(\cdot, \cdot)$ に対する自己共役性を仮定すれば $(\cdot, \cdot)_{\mathbf{A}}$-内積に対しても自己共役になるので,

$$\mathbf{p}_k = \mathbf{A}\mathbf{p}_{k-1} - \sum_{l<k} \frac{(\mathbf{A}\mathbf{p}_{k-1}, \mathbf{p}_l)_{\mathbf{A}}}{(\mathbf{p}_l, \mathbf{p}_l)_{\mathbf{A}}} \mathbf{p}_l = \mathbf{A}\mathbf{p}_{k-1} - \sum_{l<k} \frac{(\mathbf{p}_{k-1}, \mathbf{A}\mathbf{p}_l)_{\mathbf{A}}}{(\mathbf{p}_l, \mathbf{p}_l)_{\mathbf{A}}} \mathbf{p}_l \tag{2}$$

$\{\mathbf{p}_k\}$ の定義より $l+1 < k-1$ なる l について $\mathbf{p}_{k-1} \perp_{\mathbf{A}} \mathrm{span}[\mathbf{p}_0, \ldots, \mathbf{p}_{l+1}] = \mathrm{span}[\mathbf{p}_0, \ldots, \mathbf{p}_l, \mathbf{A}\mathbf{p}_l]$ であるから上式右辺の $l < k-2$ なる項は全て 0. 故に $k \geq 2$ で漸化式

$$\begin{aligned}
\mathbf{p}_k &= \mathbf{A}\mathbf{p}_{k-1} - \frac{(\mathbf{p}_{k-1}, \mathbf{A}\mathbf{p}_{k-1})_{\mathbf{A}}}{(\mathbf{p}_{k-1}, \mathbf{p}_{k-1})_{\mathbf{A}}} \mathbf{p}_{k-1} - \frac{(\mathbf{p}_{k-1}, \mathbf{A}\mathbf{p}_{k-2})_{\mathbf{A}}}{(\mathbf{p}_{k-2}, \mathbf{p}_{k-2})_{\mathbf{A}}} \mathbf{p}_{k-2} \\
&= \mathbf{A}\mathbf{p}_{k-1} - \frac{(\mathbf{A}\mathbf{p}_{k-1}, \mathbf{A}\mathbf{p}_{k-1})}{(\mathbf{A}\mathbf{p}_{k-1}, \mathbf{p}_{k-1})} \mathbf{p}_{k-1} - \frac{(\mathbf{A}\mathbf{p}_{k-1}, \mathbf{A}\mathbf{p}_{k-2})}{(\mathbf{A}\mathbf{p}_{k-2}, \mathbf{p}_{k-2})} \mathbf{p}_{k-2} \qquad (3)
\end{aligned}$$

を満たす．式 (1), 式 (3) 及び $k=1$ についての式 (3) と同様の議論から

共役勾配法

$$\begin{aligned}
\mathbf{p}_0 &= \mathbf{b} \\
\mathbf{p}_1 &= \mathbf{A}\mathbf{p}_0 - \frac{(\mathbf{A}\mathbf{p}_0, \mathbf{A}\mathbf{p}_0)}{(\mathbf{A}\mathbf{p}_0, \mathbf{p}_0)} \mathbf{p}_0 \\
\mathbf{p}_k &= \mathbf{A}\mathbf{p}_{k-1} - \frac{(\mathbf{A}\mathbf{p}_{k-1}, \mathbf{A}\mathbf{p}_{k-1})}{(\mathbf{A}\mathbf{p}_{k-1}, \mathbf{p}_{k-1})} \mathbf{p}_{k-1} - \frac{(\mathbf{A}\mathbf{p}_{k-1}, \mathbf{A}\mathbf{p}_{k-2})}{(\mathbf{A}\mathbf{p}_{k-2}, \mathbf{p}_{k-2})} \mathbf{p}_{k-2} \quad \text{for } 1 < k < M \\
\mathbf{x} &= \sum_{k=0}^{M-1} \frac{(\mathbf{b}, \mathbf{p}_k)}{(\mathbf{A}\mathbf{p}_k, \mathbf{p}_k)} \mathbf{p}_k
\end{aligned}$$

簡単のためにダミー変数 $\mathbf{p}_{-1} = \mathbf{0}$ を用意すれば $\mathbf{p}_1$ も $k \geq 2$ の漸化式で書ける．記号 $\mathbf{q}_k := \mathbf{A}\mathbf{p}_k$ を導入すれば

共役勾配法

$$\begin{aligned}
\mathbf{p}_{-1} &= \mathbf{0} \\
\mathbf{p}_0 &= \mathbf{b} \\
\mathbf{p}_k &= \mathbf{q}_{k-1} - \frac{(\mathbf{q}_{k-1}, \mathbf{q}_{k-1})}{(\mathbf{q}_{k-1}, \mathbf{p}_{k-1})} \mathbf{p}_{k-1} - \frac{(\mathbf{q}_{k-1}, \mathbf{q}_{k-2})}{(\mathbf{q}_{k-2}, \mathbf{p}_{k-2})} \mathbf{p}_{k-2} \quad \text{for } 0 < k < M \\
\mathbf{x} &= \sum_{k=0}^{M-1} \frac{(\mathbf{b}, \mathbf{p}_k)}{(\mathbf{q}_k, \mathbf{p}_k)} \mathbf{p}_k
\end{aligned}$$

以上から,

```
last_p = zerovector; p = b;
last_q = zerovector; q = A*p;
last_alpha = 1; alpha = inner_product(q,p);
x = ( inner_product(b,p) / alpha ) * p
while(true){
  new_p = q - inner_product(q,q)/alpha * p
            - inner_product(q,last_q)/last_alpha * last_p;
  last_p = p; p = new_p;
  last_q = q; q = A*p;
  last_alpha = alpha; alpha = inner_product(q,p);
  if( abs(alpha) < epsilon ) return x;
  x += ( inner_product(b,p) / alpha ) * p
}
```

によって $\mathbf{x}$ を求める事ができる．上記のアルゴリズムに要する乗算は, $\mathbf{Ap}_k$ の計算に前述の疎行列による効率化を用いれば, 凡そ $(K+7)NM$ 回であり, $M \leq N$ より $(K+7)N^2$ 回となって N^2 のオーダーで済む.

3 おまけ (共役勾配法に関する雑記)

3.1 鍵

共役勾配法の鍵は 2 つある.

1 つは $(\cdot,\cdot)_{\mathbf{A}}$-内積であり, 式 (1) である．$(\cdot,\cdot)_{\mathbf{A}}$-内積を用いる事によって $\mathbf{x}$ の値を知る事無く $\mathbf{x}$ に関する"内積"が計算可能であり, パーセバルの等式を用いる事ができる.

もう 1 つは自己共役性であり, 式 (3) である．自己共役故にグラムシュミットの直交化を漸化式に落とす事が出来る．式 (2) のままでは, 直交化に N^3 のオーダーの乗算を要し, $\mathbf{Ap}_k$ の計算を疎行列である事を利用して効率化しても殆ど意味が無い.

3.2

一般に共役勾配法として検索される方法は上述の方法と見かけ上異なるが, この記述が数学的には (少なくとも私的に) 最も判りやすいと思う．というか, 自分が理解できるように書いた.

3.3 その他の方法

共役勾配法において, 疎行列が利点を持つのは, ベクトルへの積の計算量が節約できるからである．故に, ベクトルへの積できるような他の方法でも効率化可能である．具体的には多種の反復法がこれに当てはまる．しかしながら, これらは本質的に近似法であり, 収束誤差を持つ間接的手段である．直接法で在りながら, (K-重対角でない) 疎行列で効率化可能であると言う点で共役勾配法は特異である.

3.4　正定値自己共役でなかったら?

$\mathbf{A}$ が正定値自己共役であることはかなり強い要求に見えるが実はそうでもない．任意の正則行列 $\mathbf{B}$ に対し $\mathbf{B}\mathbf{B}^t$ は正定値自己共役である．そして, 方程式 $\mathbf{B}\mathbf{x} = \mathbf{b}$ の解は方程式 $\mathbf{B}\mathbf{B}^t\mathbf{y} = \mathbf{b}$ が解ければ $\mathbf{x} = \mathbf{B}^t\mathbf{y}$ としてリーズナブルに計算できる．

一般には $\mathbf{B}$ が疎であっても $\mathbf{B}\mathbf{B}^t$ が疎であるとは限らない．しかしながら $\mathbf{B}$ の添え字が幾何学的背景を持ち, $\mathbf{B}$ の非零成分が局所的作用を表している, 即ち添え字が"近い"時のみ値を持つならば. $\mathbf{B}\mathbf{B}^t$ も"近い"(倍の距離) 時のみ値を持つ故に疎となる．

3.5　クリロフ空間

$\mathcal{K}$ をクリロフ空間と言う．実際のオーダーは $(K+7)MN$ であるので $M = \dim\mathcal{K}$ が N より小さければ更に効率化できそうに見える．が, しかし, これは余り期待できない, $\mathcal{K}$ は, 概ね"$\mathbf{A}$ の一般固有空間の直和空間の中で $\mathbf{b}$ を含む最小のもの" である．著しい例を挙げれば, $\mathbf{b}$ が $\mathbf{A}$ の固有ベクトルである時, 共役勾配法は 1 段で終了する．

然るに $\mathbf{A}$ や $\mathbf{b}$ は"与えられる"値であり, 固有ベクトル等は式変形で保存される物であるからして, 妥当な工夫で M を小さくする事は無理なように思われる．