6H-01　自動微分によるヤコビ行列計算における乗算回数の削減

武政 英男　　久保田 光一
中央大学大学院理工学研究科*

1　はじめに

自動微分法 (Automatic Differentiation, 以下 AD) は n 変数 m 次元ベクトル値関数 f について，f の計算の手間を $\mathrm{L}(f)$ として，高々$6\cdot\max(n,m)L(f)$ の手間でそのヤコビ行列 J の全成分を計算できる事が知られている [1, 3]. Uwe Naumann はこの AD について計算グラフ上の鎖律の応用を考えることで乗算の数を削減するアルゴリズムを提案した [2].

本稿ではそのアルゴリズムを実装し，鎖律の応用方法の違いや計算グラフに対する前処理による乗算回数の削減の違いを報告する.

2　アルゴリズム

Naumann の方法は，AD の Top Down 算法，Bottom Up 算法を計算グラフの頂点除去操作及び辺除去操作の列としてみなすことに基づいている.

以下に Naumann の方法の基本手順を示す.

1. 与えられたアルゴリズム (プログラムリスト) から計算グラフを導出する (図 1).

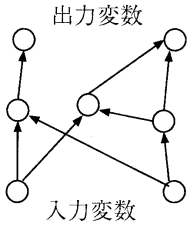

図 1　計算グラフの例

2. 各辺の要素的偏導関数を求める.
3. 辺や点の削除順を決定しそれらを削除する.

この方法により最終的に入力変数と出力変数の完全二部グラフ (図 2) をつくり，Jacobi 行列を導く．点や辺の削除法については次節参照.

3　基本操作

以下は，与えられたアルゴリズムの計算グラフ $G=(V,A)$ は既に構築済みであるとし，$v\in V$, X を入力変数の集合，Y を出力変数の集合，c_{ji} を辺 $(v_i,v_j)\in A$ に対応する要素的偏導関数 $\frac{\partial v_j}{\partial v_i}$ とする.

*Reduction of number of multiplications for computing Jacobian matrix with automatic differentiation, Hideo TAKEMASA and Koichi KUBOTA, Graduate School of Science and Engineering, Chuo University 1–13–27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan.

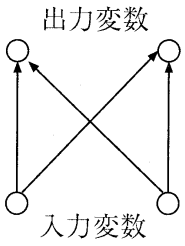

図 2　最終形

3.1　Forward edge elimination

以下の鎖律表現を考える：

$$\dot{v}_j=\sum_{v_l\in\Gamma^- v_j} c_{jl}\cdot\dot{v}_l,\quad v_j\notin X. \tag{1}$$

ただし，$\dot{v}_p\equiv\frac{\partial v_p}{\partial v_q},(v_q\in X)$ とする．式 (1) から変数 $v_i\in\Gamma^- v_j$ に関する項を削除することを考える．このことは計算グラフから辺 (v_i,v_j) を削除することに等しい．まず，$\dot{v}_i$ を含んでいる項を分離する：

$$\dot{v}_j=c_{ji}\cdot\dot{v}_i+\sum_{v_i\neq v_l\in\Gamma^- v_j} c_{jl}\cdot\dot{v}_l. \tag{2}$$

その後，式 (2) で $\dot{v}_i$ に対応する表現に置き換える：

$$\dot{v}_j=\sum_{v_l\in\Gamma^- v_i} c_{ji}\cdot c_{il}\cdot\dot{v}_l+\sum_{v_i\neq v_l\in\Gamma^- v_j} c_{jl}\cdot\dot{v}_l. \tag{3}$$

このように，$\dot{v}_i$ の項を含まない $\dot{v}_j$ の新しい計算式を得る：

$$\dot{v}_j=\sum_{v_l\in\Gamma^- v_i\cup\Gamma^- v_j\setminus\{v_i\}}\tilde{c}_{jl}\cdot\dot{v}_l,\quad \tilde{c}_{jl}\equiv c_{jl}+c_{ji}\cdot c_{il}. \tag{4}$$

直感的には，辺 (v_i,v_j) の forward edge elimination は $\Gamma^- v_i$ と v_j を接続し，その新しい辺に対応する要素的偏導関数を計算し，最後に (v_i,v_j) を削除することである.

3.2　Backward edge elimination

以下の鎖律表現を考える：

$$\bar{v}_l=\sum_{v_j\in\Gamma^+ v_l} c_{jl}\cdot\bar{v}_j,\quad v_l\notin Y. \tag{5}$$

ただし，$\bar{v}_q\equiv\frac{\partial v_p}{\partial v_q},(v_p\in Y)$ とする．式 (5) から変数 $v_i\in\Gamma^+ v_l$ に関する項を削除することを考える．このことは計算グラフから辺 (v_l,v_i) を削除することに等しい．まず，v_i を含んでいる項を分離する：

$$\bar{v}_l=c_{il}\cdot\bar{v}_i+\sum_{v_i\neq v_j\in\Gamma^+ v_l} c_{jl}\cdot\bar{v}_j. \tag{6}$$

次に，式 (6) で v_i に対応する表現に置き換える：

$$\bar{v}_l = \sum_{v_j \in \Gamma^+ v_i} c_{il} \cdot c_{ji} \cdot \bar{v}_j + \sum_{v_i \neq v_j \in \Gamma^+ v_l} c_{jl} \cdot \bar{v}_j. \quad (7)$$

このように，v_i の項を含まない v_l の新しい計算式を得る：

$$\bar{v}_l = \sum_{v_j \in \Gamma^+ v_i \cup \Gamma^+ v_l \setminus \{v_i\}} \tilde{c}_{jl} \cdot \bar{v}_j, \quad \tilde{c}_{jl} \equiv c_{jl} + c_{il} \cdot c_{ji}. \quad (8)$$

直感的には，辺 (v_l, v_i) の backward edge elimination は v_l と $\Gamma^+ v_i$ を接続し，その新しい辺に対応する要素的偏導関数を計算し，最後に辺 (v_l, v_i) を削除することである．

3.3 Vertex elimination

計算グラフ $G = (V, A)$ から $v_j \notin X \cup Y$ を削除することを考える．

$$\dot{v}_j = \sum_{v_i \in \Gamma^- v_j} c_{ji} \cdot \dot{v}_i, \quad (9)$$

全ての $v_k \in \Gamma^+ v_j$ について，

$$\dot{v}_k = c_{kj} \cdot \dot{v}_j + \sum_{v_l \in \Gamma^- v_k \setminus \{v_j\}} c_{kl} \cdot \dot{v}_l. \quad (10)$$

式 (10) で v_j の表現を式 (9) を用いて置き換える．

$$\dot{v}_k = c_{kj} \sum_{v_i \in \Gamma^- v_j} c_{ji} \cdot \dot{v}_i + \sum_{v_l \in \Gamma^- v_k \setminus \{v_j\}} c_{kl} \cdot \dot{v}_l \quad (11)$$

直感的には，点 v_j の vertex elimination は $v_i \in \Gamma^- v_j, v_k \in \Gamma^+ v_j$ の完全二部グラフを作成し，その新しい辺に対応する要素的偏導関数を計算し，最後に点 v_j を削除することである．また，全ての辺 $(v_i, v_j), (v_i \in \Gamma^- v_j)$ の forward edge elimination と等しい．点削除は $\bar{v}_i$ を用いても表現することができる．その場合，全ての辺 $(v_j, v_k), (v_k \in \Gamma^+ v_j)$ の backward edge elimination と等しい．

4 前処理

計算グラフ $G = (V, A)$ 上の鎖律の応用により Jacobi 行列を効率良く計算するには，$|V|, |A|$ の指数関数に比例する手間がかかる．$v \in V$ について $|\delta^- v| \cdot |\delta^+ v| = 1$ となるような点は，前処理として削除する (辺 $(\Gamma^- v, \Gamma^+ v)$ を作ることと等しい) ことにより，次節で述べる点の削除順を決定する際の比較対象となる点の数を容易に減らすことができる．これ以外にも要素的偏導関数値が 1 になるもの等，前処理として削除することが出来る点や辺の条件は存在する．

5 削除順の判断

5.1 点の削除順の判断

計算グラフ $G = (V, A)$ から点を削除することを考える．点 $v_i, v_j \in V$ が候補として挙がっているときに削除順の判断基準を 2 つ示す．但し，$v_i < v_j$ は v_j を削除する前に v_i を削除することを表す．また，一般的に点削除のみによる削除では最小の乗算回数を導かない．

基準 1 $v_i < v_j \Leftrightarrow |\Gamma^- v_i| \cdot |\Gamma^+ v_i| \leq |\Gamma^- v_j| \cdot |\Gamma^+ v_j|$

特徴： 削除に伴う乗算の回数の少ない順に削除する．

基準 2 $v_i < v_j \Leftrightarrow \mathrm{dd}_i - |\Gamma^- v_i| \cdot |\Gamma^+ v_i| \leq \mathrm{dd}_j - |\Gamma^- v_j| \cdot |\Gamma^+ v_j|$

但し，$\mathrm{dd}_l = \{v_l$ が依存する入力変数の数 $\} \cdot \{v_l$ に依存している出力変数の数 $\}$

特徴 ：基準 1 より手間がかかるが，効率が上がる．多くの場合についてほぼ最適な計算手順を計算する．

5.2 辺の削除順の判断

点の削除と同様に $(v_i, v_j) < (v_i', v_j')$ は (v_i', v_j') を削除する前に (v_i, v_j) を削除することを表す．削除の基準の 1 つとして：

基準 1 $(v_i, v_j) < (v_i', v_j') \Leftrightarrow \min\{|\Gamma^- v_i|, |\Gamma^+ v_j|\} \leq \min\{\Gamma^- v_i'|, |\Gamma^+ v_j'|\}$,

が提案されている．辺削除のみによる削除は最小の乗算回数を導くが，点 v_i を削除するときの $|\Gamma^- v_i|, |\Gamma^+ v_i|$ に比例する回数の判定が必要になる．

6 まとめ

点や辺の削除順を判断する基準により乗算回数は大きく変化する．また計算グラフに前処理を行うことによっても乗算回数は変化する．近年の計算機は乗算，加算はほぼ等しい速さで計算できるが，除算には時間がかかる．よって除算回数を少なくするような削除方法も考えられる．また，特定の点削除基準順に依存した辺削除基準もある．

前処理の方法，削除順を判断する基準の改良，点と辺を削除する基準の組合せに効率の向上が今後の課題である．

参考文献

[1] 久保田光一，伊理正夫：アルゴリズムの自動微分，コロナ社 (1998).

[2] Uwe Naumann：*Efficient Calculation of Jacobian Matrices by Optimized Application of the Chain Rule to Computational Graphs*, dissertation, der Fakultät Mathematik und Naturwissenschaften der Technischen Universität Dresden (1999).

[3] 吉田利信： グラフの変形を用いた偏導関数の計算過程の導出, 情報処理学会論文誌, Vol. 28, No. 11, pp. 1112–1120 (1987) .