Hessian Compression

Uwe Naumann

Informatik 12:
Software and Tools for Computational Engineering (STCE)

RWTH Aachen University

Contents

Outline

Objective and Learning Outcomes

Recall: Accumulation of Hessians
- Finite Differences
- Algorithmic Differentiation

Accumulation of Sparse Hessians

Star-Coloring
- Definition
- Proof of Correctness
- Implementation

Summary and Next Steps

Objective

- Introduction to direct Hessian compression by star-coloring of the adjacency graph

Learning Outcomes

- You will understand
  - HESSIAN COMPRESSION
  - star-coloring of adjacency graph
  - proof of correctness of star-coloring.
- You will be able to
  - star-color adjacency graph
  - derive corresponding seed matrices for second-order adjoints.

Outline

The Hessian

$$f'' = f''(\mathbf{x}) \equiv \frac{d^2 f}{d\mathbf{x}^2}(\mathbf{x}) = \left(\frac{d^2 f}{dx_i dx_j}(\mathbf{x}) \right) \in \boldsymbol{R}^{n \times n}$$

of a twice continuously differentiable multivariate scalar function $f : \boldsymbol{R}^n \rightarrow \boldsymbol{R}$ can be approximated at a given point $\tilde{\mathbf{x}} \in \boldsymbol{R}^n$ as a (central) finite difference approximation of the Jacobian of a (central) finite difference approximation of the gradient

$$f' = f'(\mathbf{x}) \equiv \frac{df}{d\mathbf{x}}(\mathbf{x}) = \left(\frac{df}{dx_i}(\mathbf{x}) \right) \in \boldsymbol{R}^n$$

of f :

$$\frac{d^2 f}{dx_i dx_j}(\tilde{\mathbf{x}}) \approx \frac{\frac{df}{dx_i}(\tilde{\mathbf{x}} + \mathbf{e}_j \cdot \Delta x_j) - \frac{df}{dx_i}(\tilde{\mathbf{x}} - \mathbf{e}_j \cdot \Delta x_j)}{2 \cdot \Delta x_j} .$$

$\mathbf{e}_j$ denotes the j-th Cartesian basis vector in $\boldsymbol{R}^n$.

Accumulation of Hessians

[Approximate] Tangents of [Approximate] Tangents

Software and Tools for Computational Engineering
RWTH AACHEN UNIVERSITY

A second derivative code $f^{(1,2)} : \boldsymbol{R}^n \times \boldsymbol{R}^n \times \boldsymbol{R}^n \times \boldsymbol{R}^n \rightarrow \boldsymbol{R} \times \boldsymbol{R} \times \boldsymbol{R} \times \boldsymbol{R}$, generated in tangent-of-tangent mode of Algorithmic Differentiation (AD) computes

$$\begin{pmatrix} y \\ y^{(2)} \\ y^{(1)} \\ y^{(1,2)} \end{pmatrix} = f^{(1,2)}(\mathbf{x}, \mathbf{x}^{(2)}, \mathbf{x}^{(1)}, \mathbf{x}^{(1,2)})$$

as

$$\begin{pmatrix} y \\ y^{(2)} \\ y^{(1)} \\ y^{(1,2)} \end{pmatrix} := \begin{pmatrix} f(\mathbf{x}) \\ f'(\mathbf{x}) \cdot \mathbf{x}^{(2)} \\ f'(\mathbf{x}) \cdot \mathbf{x}^{(1)} \\ {\mathbf{x}^{(1)}}^T \cdot f''(\mathbf{x}) \cdot \mathbf{x}^{(2)} + f'(\mathbf{x}) \cdot \mathbf{x}^{(1,2)} \end{pmatrix} .$$

Note: In context of chain rule both $y^{(1)}$ and $y^{(2)}$ required and non-vanishing $\mathbf{x}^{(1,2)}$; $f''(\mathbf{x})^T = f''(\mathbf{x})$ as f twice continuously differentiable

The computational cost of accumulating the Hessian in tangent-of-tangent mode is $O(n^2) \cdot \text{Cost}(f)$.

Software and Tools for Computational Engineering | RWTH AACHEN UNIVERSITY

A second derivative code

$$f^{(2)}_{(1)} : \boldsymbol{R}^n \times \boldsymbol{R}^n \times \boldsymbol{R} \times \boldsymbol{R} \to \boldsymbol{R} \times \boldsymbol{R} \times \boldsymbol{R}^{1\times n} \times \boldsymbol{R}^{1\times n},$$

generated by algorithmic differentiation in tangent-of-adjoint mode computes

$$\begin{pmatrix} y \\ y^{(2)} \\ \mathbf{x}_{(1)} \\ \mathbf{x}^{(2)}_{(1)} \end{pmatrix} = f^{(2)}_{(1)}\left(\mathbf{x}, \mathbf{x}^{(2)}, y_{(1)}, y^{(2)}_{(1)}\right) = \begin{pmatrix} f(\mathbf{x}) \\ f'(\mathbf{x}) \cdot \mathbf{x}^{(2)} \\ y_{(1)} \cdot f'(\mathbf{x}) \\ \mathbf{x}^{(2)\,T} \cdot y_{(1)} \cdot f''(\mathbf{x}) + y^{(2)}_{(1)} \cdot f'(\mathbf{x}) \end{pmatrix}.$$

Finite differences applied to adjoints yield approximate second-order adjoints.

The computational cost of accumulating the Hessian in either finite difference-of-adjoint or tangent-of-adjoint modes is $O(n) \cdot \text{Cost}(f)$.

A second derivative code

$$f_{(2)}^{(1)} : \boldsymbol{R}^n \times \boldsymbol{R}^n \times \boldsymbol{R} \times \boldsymbol{R} \rightarrow \boldsymbol{R} \times \boldsymbol{R} \times \boldsymbol{R}^{1\times n} \times \boldsymbol{R}^{1\times n}$$

generated by algorithmic differentiation in adjoint-of-tangent mode computes

$$\begin{pmatrix} y \\ y^{(1)} \\ \mathbf{x}_{(2)} \\ \mathbf{x}_{(2)}^{(1)} \end{pmatrix} = f_{(2)}^{(1)}\left(\mathbf{x}, \mathbf{x}^{(1)}, y_{(2)}, y_{(2)}^{(1)}\right) = \begin{pmatrix} f(\mathbf{x}) \\ f'(\mathbf{x}) \cdot \mathbf{x}^{(1)} \\ y_{(2)}^{(1)} \cdot \mathbf{x}^{(1)\,T} \cdot f''(\mathbf{x}) + y_{(2)} \cdot f'(\mathbf{x}) \\ y_{(2)}^{(1)} \cdot f'(\mathbf{x}) \end{pmatrix} .$$

An adjoint of a finite difference approximation of the first-order tangent yields an approximate second-order adjoint.

The computational cost of accumulating the Hessian in either adjoint-of-finite-difference or adjoint-of-tangent modes is $O(n) \cdot \text{Cost}(f)$ ($O(n^2) \cdot \text{Cost}(f)$ if implemented naively).

A second derivative code

$$f_{(1,2)} : \boldsymbol{R}^n \times \boldsymbol{R}^n \times \boldsymbol{R} \times \boldsymbol{R} \to \boldsymbol{R} \times \boldsymbol{R}^{1\times n} \times \boldsymbol{R}^{1\times n} \times \boldsymbol{R},$$

generated by algorithmic differentiation in adjoint-of-adjoint mode computes

$$\begin{pmatrix} y \\ \mathbf{x}_{(1)} \\ \mathbf{x}_{(2)} \\ y_{(1,2)} \end{pmatrix} = f_{(1,2)}\left(\mathbf{x}, \mathbf{x}_{(1,2)}, y_{(1)}, y_{(1,2)}\right) = \begin{pmatrix} f(\mathbf{x}) \\ y_{(1)} \cdot f'(\mathbf{x}) \\ y_{(2)} \cdot f'(\mathbf{x}) + \mathbf{x}_{(1,2)}^T \cdot y_{(1)} \cdot f''(\mathbf{x}) \\ f'(\mathbf{x}) \cdot \mathbf{x}_{(1,2)} \end{pmatrix}$$

The computational cost of accumulating the Hessian in adjoint-of-adjoint mode is $O(n) \cdot \mathrm{Cost}(f)$.

Outline

Let $F'' \in \boldsymbol{R}^{n \times n}$ have the symmetric sparsity pattern $P \in \{0,1\}^{n \times n}$.

Sparsity can be exploited in [approximate] tangent of [approximate] tangent mode by computing the nonzero entries exclusively.

Exploitation of sparsity is useful if the computation of P followed by the computation of of F'' undercuts cost of evaluating F'' without taking sparsity into account.

For example, dense second-order adjoint mode might defeat sparse second-order tangent mode.

Let $F'' \in \mathbf{R}^{n \times n}$ have the symmetric sparsity pattern $P \in \{0,1\}^{n \times n}$.

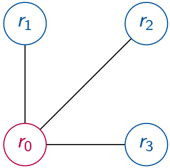
Find $B^f \in \mathbf{R}^{n \times l^f}$ s.th. F'' can be recovered from $C^f = F'' \cdot B^f \in \mathbf{R}^{n \times l^f}$ and the cost of

- computation of P
- computation of B^f
- computation of C^f
- recovery of F''

undercuts the cost of evaluating F'' without taking sparsity into account.

For example, dense second-order adjoint mode might defeat sparse second-order adjoint mode.

$$F'' = \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{0,1} & a_{1,1} & & \\ a_{0,2} & & a_{2,2} & \\ a_{0,3} & & & a_{3,3} \end{pmatrix}$$



Note distance-1 coloring of the adjacency graph $G_a(F'')$ as special case of star-coloring of $G_a(F'')$.

$$F'' \cdot B^f = \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{0,1} & a_{1,1} & & \\ a_{0,2} & & a_{2,2} & \\ a_{0,3} & & & a_{3,3} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a_{0,0} & \sum_{i=1}^{3} a_{0,i} \\ a_{0,1} & a_{1,1} \\ a_{0,2} & a_{2,2} \\ a_{0,3} & a_{3,3} \end{pmatrix}$$

... works due to symmetry

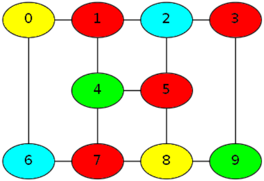
Outline

A star-coloring of $G_a(F'')$ is a distance-1 coloring of $G_a(F'')$ such that every path of (vertex-)length four has at least three colors.

Example:

$$
\cdot \begin{pmatrix}
a_{00} & a_{01} & & & & & a_{06} & & & \\
a_{01} & a_{11} & a_{12} & & a_{14} & & & & & \\
 & a_{12} & a_{22} & a_{23} & & a_{25} & & & & \\
 & & a_{23} & a_{33} & & & & & & a_{39} \\
 & a_{14} & & & a_{44} & a_{45} & & a_{47} & & \\
 & & a_{25} & & a_{45} & a_{55} & & & a_{58} & \\
a_{06} & & & & & & a_{66} & a_{67} & & \\
 & & & & a_{47} & & a_{67} & a_{77} & a_{78} & \\
 & & & & & a_{58} & & a_{78} & a_{88} & a_{89} \\
 & & & a_{39} & & & & & a_{89} & a_{99}
\end{pmatrix}
\cdot
\begin{pmatrix}
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{pmatrix}
=
\begin{pmatrix}
a_{01} & a_{06} & & a_{00} \\
a_{11} & {\color{red}a_{12}} & {\color{green}a_{14}} & a_{01} \\
{\color{red}a_{12} + a_{23} + a_{25}} & a_{22} & & \\
a_{33} & {\color{red}a_{23}} & a_{39} & \\
{\color{green}a_{14} + a_{45} + a_{47}} & & a_{44} & \\
a_{55} & {\color{red}a_{25}} & {\color{green}a_{45}} & {\color{blue}a_{58}} \\
a_{67} & a_{66} & & a_{06} \\
a_{77} & a_{67} & {\color{green}a_{47}} & {\color{blue}a_{78}} \\
{\color{blue}a_{58} + a_{78}} & & a_{89} & a_{88} \\
a_{39} & & a_{99} & a_{89}
\end{pmatrix}
$$

All values can be recovered directly as every value $a_{i,j} = a_{j,i}$ is given explicitly at least once, i.e, either $a_{i,j}$ or $a_{j,i}$ is available.



See lecture for sequential coloring with largest first vertex ordering and colors ordered as red, blue, green, yellow.

See A. Gebremedhin et al.: What Color is your Jacobian? SIAM, 2005 for sequential coloring algorithm and heuristics for ordering vertices.

In the following we focus on proving correctness of star-coloring of $G_a(F'')$ as a feasible technique for Hessian compression.

Consider a non-distance-1 coloring of $G_a(F'')$.

Let $(i,j) \in E_a$, that is, $a_{i,j} = a_{j,i} \neq 0$. Suppose same color for $i \in V_a$ and $j \in V_a$.

$$\begin{pmatrix} a_{i,i} & \dots & a_{i,j} \\ & \vdots & \\ a_{j,i} & \dots & a_{j,j} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} a_{i,i} + a_{i,j} \\ \vdots \\ a_{j,i} + a_{j,j} \end{pmatrix}$$

$\Rightarrow$ neither $a_{i,j}$ nor $a_{j,i}$ available.

Consider necessity of three colors per path of (vertex-)length four.

There are $\binom{4}{2} = 6$ ways to bi-color a path of (vertex-)length four, namely

1. o o o o
2. o o o o
3. o o o o
4. o o o o
5. o o o o
6. o o o o

Options 1-4 are non-distance-1 colorings (hence out). Options 5 and 6 are structurally equivalent. Hence, only one of them needs to be investigated further; w.l.o.g. option 5: o o o o.

Consider a corresponding two-coloring for a path of (vertex-)length four: o o o o.

From

$$\begin{pmatrix} a_{i,i} & a_{i,j} & & \\ a_{j,i} & a_{j,j} & a_{j,k} & \\ & a_{k,j} & a_{k,k} & a_{k,l} \\ & & a_{l,k} & a_{l,l} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a_{i,i} & a_{i,j} \\ a_{j,i} + a_{j,k} & a_{j,j} \\ a_{k,k} & a_{k,j} + a_{k,l} \\ a_{l,k} & a_{l,l} \end{pmatrix}$$

follows that neither $a_{j,k}$ nor $a_{k,j}$ are available.

We conclude that three colors per path of (vertex-)length four is a necessary condition for Hessian compression.

Consider sufficiency of three colors per path of (vertex-)length four.

There are $\binom{4}{2} \cdot 2! = 12$ ways to three-color a path of (vertex-)length four.

Six are distance-1 colorings. The remaining six are pair-wise symmetric leaving the following three scenarios to be investigate in detail:

1. o o o o
2. o o o o
3. o o o o

Consider o o o o.

$$\begin{pmatrix} a_{i,i} & a_{i,j} & & \\ a_{j,i} & a_{j,j} & a_{j,k} & \\ & a_{k,j} & a_{k,k} & a_{k,l} \\ & & a_{l,k} & a_{l,l} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a_{i,i} & a_{i,j} & 0 \\ a_{j,i} + a_{j,k} & a_{j,j} & 0 \\ a_{k,k} & a_{k,j} & a_{k,l} \\ a_{l,k} & 0 & a_{l,l} \end{pmatrix}$$

$\Rightarrow a_{i,j} = a_{j,i}$ and $a_{k,j} = a_{j,k}$ are available.

Consider o o o o

$$\begin{pmatrix} a_{i,i} & a_{i,j} & & \\ a_{j,i} & a_{j,j} & a_{j,k} & \\ & a_{k,j} & a_{k,k} & a_{k,l} \\ & & a_{l,k} & a_{l,l} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} a_{i,i} & a_{i,j} & 0 \\ a_{j,i} & a_{j,j} & a_{j,k} \\ a_{k,l} & a_{k,j} & a_{k,k} \\ a_{l,l} & 0 & a_{l,k} \end{pmatrix}$$

$\Rightarrow$ all nonzero entries are available.

Consider o o o o

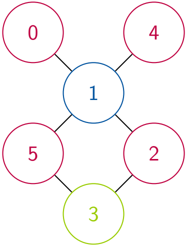
$$\begin{pmatrix} a_{i,i} & a_{i,j} & & \\ a_{j,i} & a_{j,j} & a_{j,k} & \\ & a_{k,j} & a_{k,k} & a_{k,l} \\ & & a_{l,k} & a_{l,l} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} a_{i,i} & a_{i,j} & 0 \\ a_{j,i} & a_{j,j} & a_{j,k} \\ 0 & a_{k,j} + a_{k,l} & a_{k,k} \\ 0 & a_{l,l} & a_{l,k} \end{pmatrix}$$

$\Rightarrow a_{j,k} = a_{k,j}$ and $a_{l,k} = a_{k,l}$ are available.

Star-Coloring

Example

$$
\begin{pmatrix}
* & * & & & & \\
* & * & * & & * & * \\
& * & * & * & & \\
& & * & * & & * \\
& * & & & * & \\
& * & & * & & *
\end{pmatrix}
$$



... sequential coloring with lowest-degree first ordering and colors ordered as red, green, blue.

$$
\begin{pmatrix}
a_{0,0} & a_{0,1} & & & & \\
a_{1,0} & a_{1,1} & a_{1,2} & & a_{1,4} & a_{1,5} \\
& a_{2,1} & a_{2,2} & a_{2,3} & & \\
& & a_{3,2} & a_{3,3} & & a_{3,5} \\
& a_{4,1} & & & a_{4,4} & \\
& a_{5,1} & & a_{5,3} & & a_{5,5}
\end{pmatrix}
\cdot
\begin{pmatrix}
1 & & \\
& 1 & \\
1 & & \\
& & 1 \\
1 & & \\
1 & &
\end{pmatrix}
=
\begin{pmatrix}
a_{0,0} & a_{0,1} & \\
\sum \ldots & a_{1,1} & \\
a_{2,2} & a_{2,1} & a_{2,3} \\
\sum \ldots & & a_{3,3} \\
a_{4,4} & a_{4,1} & \\
a_{5,5} & a_{5,1} & a_{5,3}
\end{pmatrix}
$$

ColPack

https://github.com/CSCsw/ColPack

implements a range of coloring methods for Hessian compression.

See A. Gebremedhin et al.: What Color is your Jacobian? SIAM, 2005 further details.

Outline

Software and Tools for Computational Engineering
RWTH AACHEN UNIVERSITY

Summary

- Direct Hessian compression
- Star-coloring of adjacency graph
- Proof of correctness of star-coloring
- Seed matrices for second-order adjoints

Next Steps

- Practice star-coloring.
- Get familiar with ColPack.
- Continue the course to find out more ...