UNCONVENTIONAL ITERATIVE METHODS FOR NONCONVEX OPTIMIZATION IN A MATRIX-FREE ENVIRONMENT

sas THE POWER TO KNOW.

ICML @ NYC
June 24, 2016

Josh Griffin
Alireza Yektamaram
Wenwen Zhou

1. Why second order?
2. Background
3. Iterative Solvers
4. Line-search method
5. Trust-region method

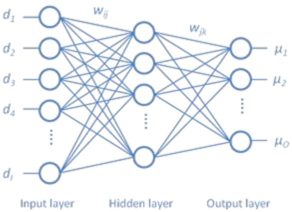
WHY SECOND ORDER?

DEEP LEARNING CONTEXT

Neural Network Loss Function:

$$\min_{w \in \mathbb{R}^n} f(w) = \frac{1}{|\mathcal{T}|} \sum_{(d,y) \in \mathcal{T}} L(\mu(d,w), y)$$

- $\mu(d,w) : \mathbb{R}^I \to \mathbb{R}^O$
- $\hat{y} = \mu(d,w)$ denotes model prediction
- y observed from data $d \in \mathcal{T}$
- $\nabla f(w)$, $\nabla^2 f(w)s$ obtainable

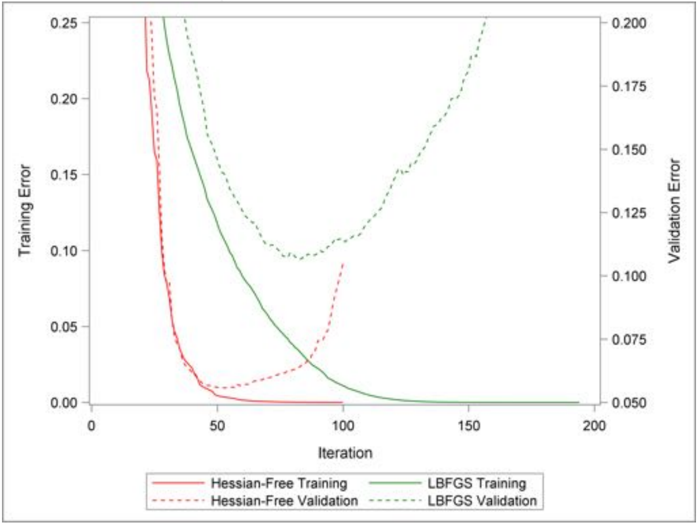


Observations

- No bad local minimums
  - (Kawaguchi, 2016), (Soudry and Carmon, 2016)
- Example: w_0 + MNIST + LBFGS $\Rightarrow f(w^*) = 0$

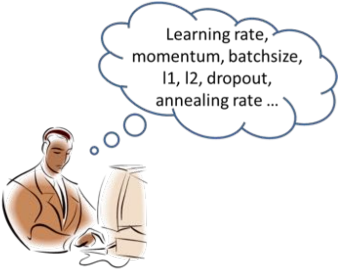
WHY SECOND ORDER?

MNIST 60K TRAINING, 10K TEST



WHY SECOND ORDER?

HYPER-PARAMETER OPTIMIZATION

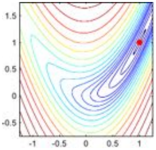


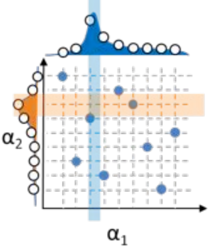
Two ways to dive deep when tuning:

1. solver fixed, tune model
2. model fixed, tune solver

How to minimize total user time?

1. Parallel autotune and SGD
2. Second-order methods





Random Latin Hypercube

BACKGROUND | THE DEEP LEARNING HESSIAN

The Hessian for deep learning problems has form:

$$H = \frac{1}{|\mathcal{T}|} \sum_{(d,y)\in\mathcal{T}} \underbrace{J_\mu^T H_L J_\mu}_{G_\mu \succeq 0} + N(d, y)$$

Where J_μ is the Jacobian of $\mu(d, w)$, H_L is the Hessian for the loss function $L(z, y)$ with respect to z, and

$$N(d, y) = \sum_{o=1}^{O} [\nabla_z L(\mu(d, w), y)]_o \nabla^2 [\mu(d, w)]_o.$$

Note that $N(d, y) = 0$ if training error is 0, or $\mu(d, w)$ is linear.

BACKGROUND

GENERALIZED GUASS-NEWTON MATRIX

Martens 2010 seminal work show great results by

1. Approximating H with $G \succeq 0$

$$G = \frac{1}{|\mathcal{T}|} \sum_{(d,y) \in \mathcal{T}} J_\mu^T H_L J_\mu$$

2. Using Levenberg-Marquardt modifications

$$(G + \lambda I)s = -g$$

   where λ is modified based on past performance

3. Applying the conjugate gradient algorithm

Why not use H directly? (Martens 2012)

BACKGROUND

THE PROBLEM WITH NEWTON'S METHOD

Suppose we simply solve (where $H = \nabla^2 f(w)$ and $g = \nabla f(w)$)

$$Hs = -g, \text{ where we need } s^T g < 0$$

Using spectral decomposition $H = V\Lambda V^T$:

$$s^T g = \underbrace{\sum_{\lambda_i < 0} \frac{(v_i^T g)^2}{|\lambda_i|}}_{\geq 0} - \underbrace{\sum_{\lambda_i > 0} \frac{(v_i^T g)^2}{\lambda_i}}_{\geq 0}$$

In general $s = s_n + s_p$ where

- s_n maximizes, depends on negative eigenspace
- s_p minimizes, depends on positive eigenspace

All it takes is one small negative eignenvalue!

BACKGROUND

IMPLICATIONS FOR ITERATIVE METHODS

1. Classical iterative methods solve equations as is:

$$Hs = -g, \text{ unconcerned if } s^T g \geq \text{ or } \leq 0.$$

2. Need to *implicitly* or *explicitly* work with $\hat{H} \approx H$ such that

$$\hat{H} \succ 0 \quad \Rightarrow s^T g < 0$$

3. Line-search methods use explicit modifications
4. Trust-region methods use implicit modifications

ITERATIVE SOLVERS | CURRENT STATE-OF-ART NONCONVEX ITERATIVE METHODS

- Steihaug-Toint
- GLTR
- Saddle-free Newton (Dauphin et al. 2014)

CG METHOD OVERVIEW

1. Generate $\{p_0, \ldots, p_k\}$ such that

$$p_k^T H p_j = 0 \text{ if } i \neq j.$$

2. Recursively obtain approximate solution s_{k+1} as

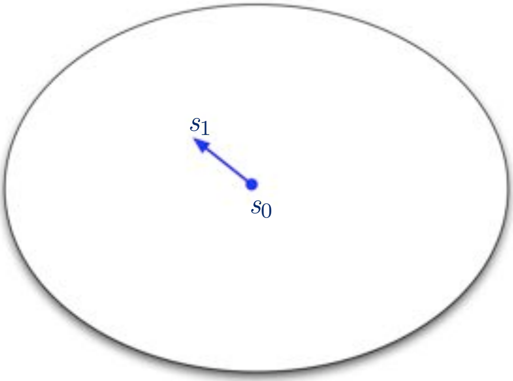
$$s_{k+1} = s_k + \alpha_k p_k$$

   - $\alpha_k = \arg\min_\alpha Q(s_k + \alpha p_k)$, if $p_k^T H p_k > 0$
   - $\alpha_k = \arg\max_\alpha Q(s_k + \alpha p_k)$, if $p_k^T H p_k < 0$
   - Here $Q(s) = s^T g + \frac{1}{2} s^T H s$

3. While $p_k^T H p_k > 0$
   - $\|s_k\|_P \geq \|s_{k-1}\|_P$ assuming $s_0 = 0$
   - s_{k+1} minimizes quadratic model $Q(s)$ in span$\{p_0, \ldots, p_k\}$.

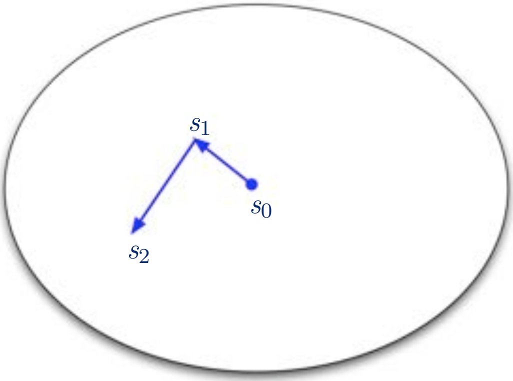
ITERATIVE SOLVERS

STEIHAUG-TOINT ALGORITHM



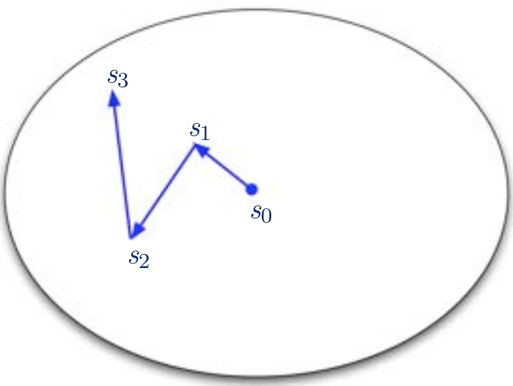
ITERATIVE SOLVERS

STEIHAUG-TOINT ALGORITHM

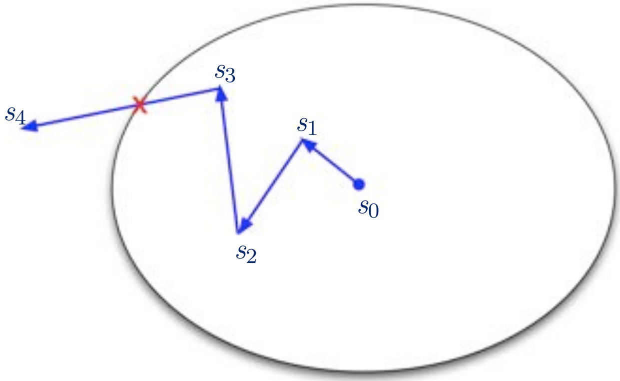


ITERATIVE SOLVERS

STEIHAUG-TOINT ALGORITHM



STEIHAUG-TOINT ALGORITHM



ITERATIVE SOLVERS | TRUNCATED NEWTON ERROR BOUND NOT TRUE WHEN NONCONVEX

Consider the $2D$ trust-region problem

$$\underset{\substack{s\in\mathbb{R}^2 \\ \|s\|_2 \le 1,}}{\text{minimize}} \quad s^T\begin{bmatrix}-1\\1\end{bmatrix} + \frac{1}{2}s^T\begin{bmatrix}-10^6 & 0\\ 0 & 10^6\end{bmatrix}s$$

We can show that $Q(s^*) < -\frac{10^6}{2}$. However, because $g^TBg = 0$, the Steihaug-Toint algorithm would exit immediately, with

$$s_{ST} = \begin{bmatrix}1/\sqrt{2}\\ -1/\sqrt{2}\end{bmatrix} \quad \Rightarrow \quad Q(s_{ST}) = -2/\sqrt{2} \gg Q(s^*).$$

Note: In deep learning, we need accuracy early on, not asymptotically

ITERATIVE SOLVERS | GENERALIZED LANCZOS TRUST-REGION (GLTR) METHOD

1. Starts where Steihaug-Toint stops
2. Searches for boundary solution in span of Lanczos vectors
3. Subspaces are nested
4. Updates are not recursive
5. Uses Moré and Sorensen on tri-diagonal system:

$$y^* = \arg\min_y \quad \gamma y^T e_1 + \tfrac{1}{2} y^T T y, \quad \text{s.t.} \|y\| \leq \delta$$

6. To obtain the direction s_k we need all Lanczos vectors

$$s_k = [q_1, q_2, \ldots, q_{ST}, \ldots, q_{ST+1}, \ldots q_k] \begin{bmatrix} y_1^* \\ \vdots \\ y_k^* \end{bmatrix}$$

7. Storage cost: kn, k is matrix multiplies, n is dimension of g.

ITERATIVE SOLVERS

IDEAL ITERATIVE SOLVER FOR DL (MARTENS 2012)

1. Accuracy controlled by solver not problem geometry
2. Recursive updates, low overhead
3. Warm-starts, $s_0^j = s_k^{j-1}$
4. Preconditioner not tied to elliptic norm/matrix shift

$$\hat{H} = H + \lambda I, \text{ where } I \neq P.$$

Additionally want:

- Descent direction guaranteed: $s_k^T \nabla f(w) < 0$
- Naturally reduces to CG on Newton's method

LINE-SEARCH METHOD | ADAPTING CG TO NEGATIVE CURVATURE

1. Generate $\{p_0, \dots, p_k\}$ such that

$$p_k^T H p_j = 0 \text{ if } i \neq j.$$

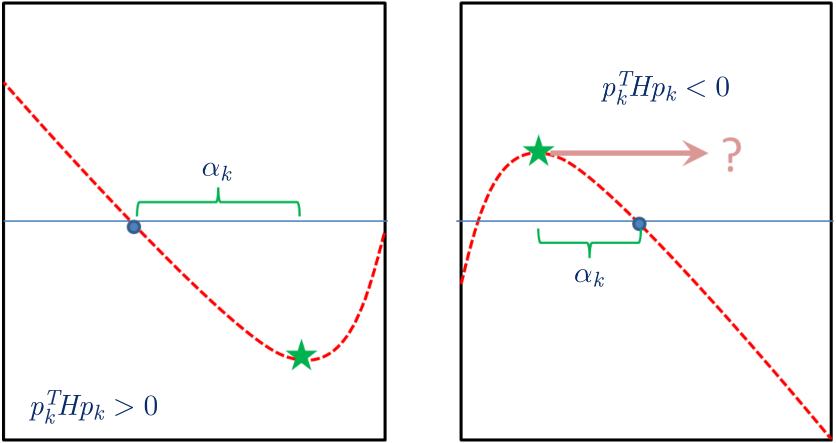
2. Recursively obtain approximate solution s_{k+1} as

$$s_{k+1} = s_k + \alpha_k p_k$$

- $\alpha_k = \arg\min_\alpha Q(s_k + \alpha p_k)$, if $p_k^T H p_k > 0$
- $\alpha_k = \arg\max_\alpha Q(s_k + \alpha p_k)$, if $p_k^T H p_k < 0$
- Here $Q(s) = s^T g + \frac{1}{2} s^T H s$

LINE-SEARCH METHOD

MODIFYING CG



LINE-SEARCH METHOD | EARLY MODIFICATIONS FOR NEWTON'S METHOD

Set $\hat{H} = V|\Lambda|V^T$, where $H = V\Lambda V^T$ and solve:

$$\hat{H}s = -g$$

Then

$$s = \sum_{i=1}^{n} \frac{-v_i^T g}{|\lambda_i|} v_i \quad \Rightarrow \quad s^T g < 0.$$

The problem:

$$\lim_{|\lambda_i| \to 0} \frac{|v_i^T s|}{||v_i|| \, ||s||} = 1$$

Singular vectors optimized before directions of greatest negative curvature.

RECENT MODIFICATIONS FOR NEWTON'S METHOD

Set $\hat{H} = V(|\Lambda| + \sigma I)\,V^T$, where $H = V\Lambda\,V^T$ and solve:

$$\hat{H}s = -g$$

Then

$$s = \sum_{i=1}^{n} \frac{-v_i^T g}{|\lambda_i| + \sigma I} v_i \quad \Rightarrow \quad s^T g < 0.$$

Compare to trust-region solution

$$s = \sum_{i=1}^{n} \frac{-v_i^T g}{\lambda_i + \sigma I} v_i \quad \Rightarrow \quad s^T g < 0.$$

where $\sigma > \lambda_i$.

Emphasis on v_i corresponding to $\min|\lambda_i|$ versus $\min \lambda_i$.

LINE-SEARCH METHOD

MODIFIED CG OBSERVATIONS (ZHOU 2009)

- Class of modifications that avoid restarts:

$$\hat{H} = H + \sigma_k r_k r_k^T$$

where $r_k = Hs_k + g$. (O'Leary 1982, Nash 1984)

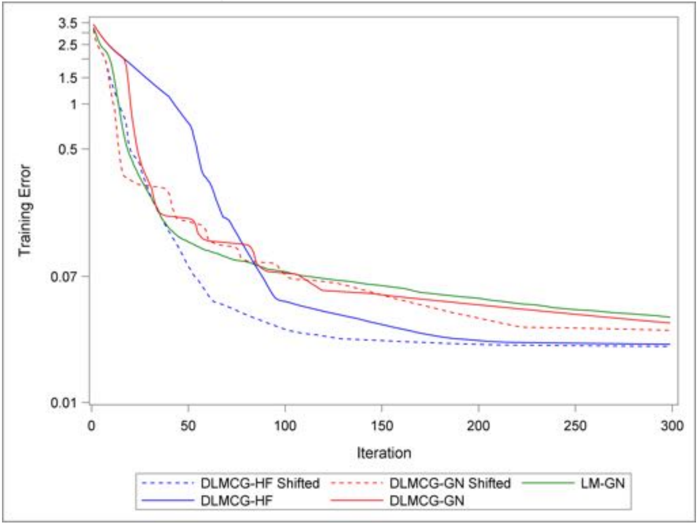
- Choose σ_k so that

$$\frac{p_k^T \hat{H} p_k}{p_k^T p_k} \leq \lambda \|g\|$$

- Can then show trust-region strength convergence
- No need to store $\{r_k \mid \sigma_k \neq 0\}$
- Works seamlessly in Levenberg-Marquardt framework

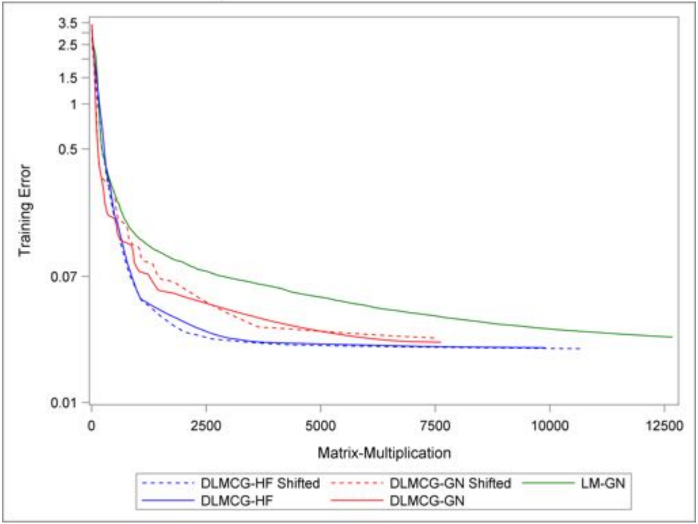
LINE-SEARCH METHOD

MNIST WITH 784-400-150-10 NETWORK



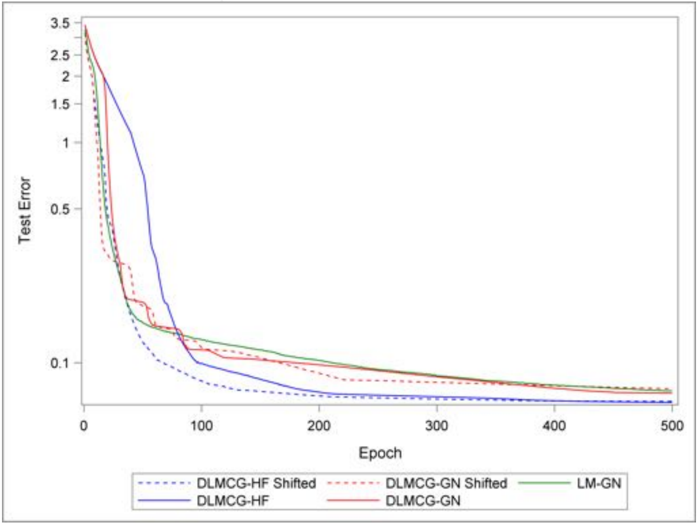
LINE-SEARCH METHOD

MNIST WITH 784-400-150-10 NETWORK



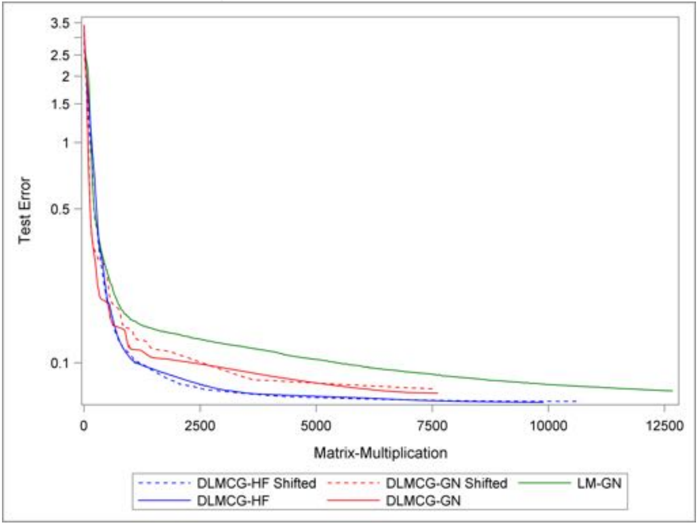
LINE-SEARCH METHOD

MNIST WITH 784-400-150-10 NETWORK



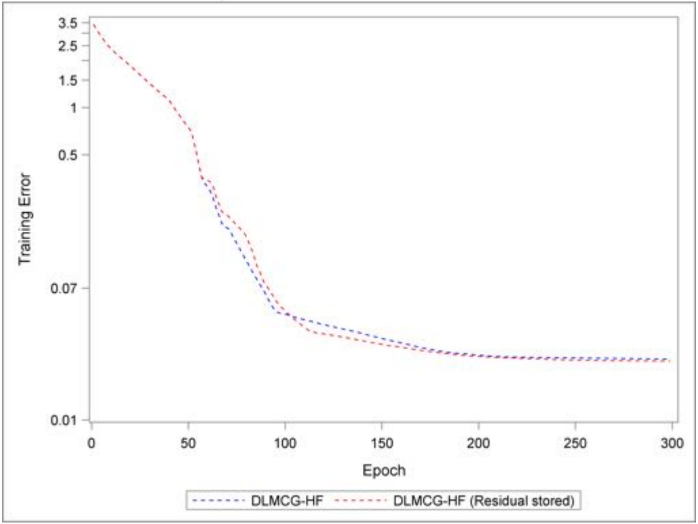
LINE-SEARCH METHOD

MNIST WITH 784-400-150-10 NETWORK



LINE-SEARCH METHOD

MNIST WITH 784-400-150-10 NETWORK



TRUST-REGION METHOD

SUCCESSIVE SUBSPACE METHODS (SSM)

- Starts where the Steihaug-Toint (ST) method stops
- Small overhead compared to CG after ST point
- Use evolving small dimensional subspaces

$$\{W_1, W_2, \ldots\} \text{ where } W_j \in \mathbb{R}^{n\times k}, k \le 4.$$

- Uses Moré and Sorensen on

$$\underset{\|Wu\|_2 \le \delta_k}{\underset{u}{\text{minimize}}} \quad u^T(W^Tg) + \frac{1}{2}u^T(W^THW)u, \tag{1}$$

- Use LAPACK to solve

$$\underset{\|Wz\|_2 = 1}{\underset{z}{\text{minimize}}} \quad z^T(W^THW)z, \tag{2}$$

TRUST-REGION METHOD

FUNDAMENTAL SSM THEOREM

Theorem (Convergence Hager)

Suppose at each iteration

$$\operatorname{span}(s_k, Hs_k + g, v^*) \subset \operatorname{span}(W_k)$$

where

$$v^* = \arg\min \frac{v^T H v}{v^T v}$$

then $s \to s^*$, *the global trust-region subproblem solution!*

Approximating v on the fly typically more than sufficient

Implementations: (Hager 2001), (G. 2005), (Erway, Gill, G. 2007), (Erway, Gill 2008)

TRUST-REGION METHOD

CONCLUSION

Trust-region line-search methods suggested that:

1. Accuracy controlled by solver not problem geometry
2. Recursive updates, low overhead
3. Warm-starts, $s_0^j = s_k^{j-1}$
4. Preconditioner not tied to elliptic norm/matrix shift

$$\hat{H} = H + \lambda I, \text{ where } I \neq P.$$

5. Descent direction guaranteed: $s_k^T \nabla f(w) < 0$
6. Naturally reduces to CG on Newton's method

FUTURE WORK

- Numerical results for SSM method class
- Mini-batching
- Hybrids: only need second-order for initial iterations
- New class of algorithms for "symmetric linear" functions:
  - $H(w) : R^n \to R^n$ does not always behave like a matrix
  - $|w^T H(y) - y^T H(w)| \gg \epsilon$
  - $H(w) = H + \text{noise}$
  - Not all book-keeping tricks may be applicable
  - MCG-LS may have advantage over SSM-TR
  - Is it a bug?

http://support.sas.com/or

Unconventional iterative methods for nonconvex optimization in a matrix-free environment