Optimization on Stiefel Sets by Linear Search

A brief introduction to a typical gradient-based algorithm

Weijia Zheng [1]

Department of Information Engineering
The Chinese University of Hong Kong

wjzheng@link.cuhk.edu.hk

Sept. 27, 2024

[1] I mainly take reference to: "A feasible method for optimization with orthogonality constraints" by Z. Wen, and W. Yin. 2010; and "Notes on Optimization on Stiefel Manifold" by H. D. Tagare, 2011. The latter is the most accessible document I've found about this topic.

Overview

Overview

Stiefel manifold

Optimization on Stiefel sets

People aimed to develop a procedure to tackle this (non-convex for sure) optimization problem.

$$\min_{X_k \in \mathbb{R}^{n \times p}} F(X_1, X_2, ..., X_K) \qquad \text{subject to } X_k^T X_k = I_p$$

[2] Copied from the first sentence of "Manifold" Wiki-page.

Stiefel manifold

Optimization on Stiefel sets

People aimed to develop a procedure to tackle this (non-convex for sure) optimization problem.

$$\min_{X_k \in \mathbb{R}^{n \times p}} F(X_1, X_2, ..., X_K) \qquad \text{subject to } X_k^T X_k = I_p$$

Definition of Stiefel manifold

Stiefel manifold, or sometimes called Stiefel set, is a set of semi-orthogonal matrices of size $n \times p$, denoted by

$$\mathcal{V}_p(\mathbb{R}^n) \triangleq \{X \in \mathbb{R}^{n \times p} | X^T X = I_p\},$$

where I_p is the identity matrix of size $p \times p$.

[2] Copied from the first sentence of "Manifold" Wiki-page.

Stiefel manifold

Optimization on Stiefel sets

People aimed to develop a procedure to tackle this (non-convex for sure) optimization problem.

$$\min_{X_k \in \mathbb{R}^{n\times p}} F(X_1, X_2, ..., X_K) \qquad \text{subject to } X_k^T X_k = I_p$$

Definition of Stiefel manifold

Stiefel manifold, or sometimes called Stiefel set, is a set of semi-orthogonal matrices of size $n \times p$, denoted by

$$\mathcal{V}_p(\mathbb{R}^n) \triangleq \{X \in \mathbb{R}^{n\times p} | X^T X = I_p\},$$

where I_p is the identity matrix of size $p \times p$.

We have nothing to do with the formal definition(s) of manifold, and it suffices to just consider the above thing as a special set. However, it is good to regard a "manifold" as a space that locally resembles Euclidean space near each point.[2]

[2] Copied from the first sentence of "Manifold" Wiki-page.

Why do we care about it? – For useful applications.

Low-rank nearest correlation estimation

Let $C \in \mathbb{S}^n$ be a given be a given symmetric matrix and $H \geqslant 0$ be a nonnegative (weight) matrix. The low-rank nearest correlation estimation problem is given by

$$\min_{Q \in \mathbb{S}_+^n} \|H \odot (Q - C)\|_F^2, \qquad \text{subject to } Q_{ii} = 1,\ \forall i \in [n];\ \ \operatorname{rank}(Q) \leqslant r,$$

where $\mathbb{S}_+^n$ denotes the set of $n \times n$ positive semi-definite matrices.

The problem tries to find a correlation matrix[3] (which is Q) from an estimate C.
C may not be PSD, since it's usually a noisy estimate.
The rank constraint makes the problem not convex.

[3] Correlation matrix should be PSD.

Why do we care about it? – For useful applications.

Low-rank nearest correlation estimation

Let $C \in \mathbb{S}^n$ be a given be a given symmetric matrix and $H \geqslant 0$ be a nonnegative (weight) matrix. The low-rank nearest correlation estimation problem is given by

$$\min_{Q \in \mathbb{S}^n_+} \|H \odot (Q - C)\|_F^2, \qquad \text{subject to } Q_{ii} = 1,\ \forall i \in [n];\ \ \operatorname{rank}(Q) \leqslant r,$$

where $\mathbb{S}^n_+$ denotes the set of $n \times n$ positive semi-definite matrices.

The problem tries to find a correlation matrix[3] (which is Q) from an estimate C. C may not be PSD, since it's usually a noisy estimate.
The rank constraint makes the problem not convex.

Reformulation of the above

One can express the rank constraints explicitly as $Q = U^T U$ with $U = [U_1, ..., U_n] \in \mathbb{R}^{r \times n}$ with $\|U_i\|_2 = 1, \quad \forall i \in [n]$. And we have

$$\min_{Q \in \mathbb{S}^n_+} \|H \odot (U^T U - C)\|_F^2, \quad \text{subject to } \|U_i\|_2 = 1,\ \forall i \in [n] \iff U_i \in \mathcal{V}_1(\mathbb{R}^r)\ \forall i \in [n]$$

[3] Correlation matrix should be PSD.

Gradient descent?

Let's ease ourselve with the (easier, though still hard) problem of

$$\min_{X \in \mathbb{R}^{n \times p}} F(X) \qquad \text{subject to } X^T X = I_p \iff X \in \mathcal{V}_p(\mathbb{R}^n).$$

General gradient-based methods in Euclidean spaces

We usually use this iteration to reach a minimizer:

$$X_{t+1} \leftarrow X_t - \gamma_t \nabla F(X_t)$$

Gradient descent?

Let's ease ourselve with the (easier, though still hard) problem of

$$\min_{X \in \mathbb{R}^{n\times p}} F(X) \qquad \text{subject to } X^T X = I_p \iff X \in \mathcal{V}_p(\mathbb{R}^n).$$

General gradient-based methods in Euclidean spaces

We usually use this iteration to reach a minimizer:

$$X_{t+1} \leftarrow X_t - \gamma_t \nabla F(X_t)$$

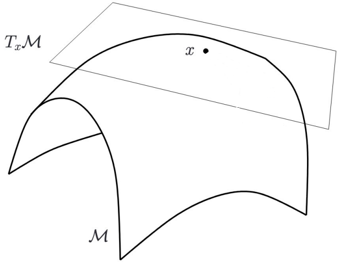
Difficulties come with the feasible set $\mathcal{V}_p(\mathbb{R}^n)$:

- Non-convexity: $\mathcal{V}_p(\mathbb{R}^n)$ is not convex, since $\pm I_{n,p} \in \mathcal{V}_p(\mathbb{R}^n)$, but $0 \notin \mathcal{V}_p(\mathbb{R}^n)$.
- (In)feasibility: hard to stay inside the constraint set: $\mathcal{V}_p(\mathbb{R}^n)$.

Hopefully, we want to adapt such successful way in Euclidean spaces into our new $\mathcal{V}_p(\mathbb{R}^n)$, after some modification if needed.

And, we are not aiming to find a global-minimizer-approaching algorithm, because doing so on a manifold $\mathcal{M}$ is really hard. Instead, what we are trying to obtain is an algorithm to generate a decreasing sequence of feasible X_t's.

Tangent space of an $X \in \mathcal{M}(= \mathcal{V}_p(\mathbb{R}^n))$

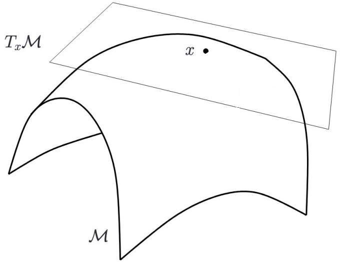


Tangent space of $X \in \mathcal{M}$

The tangent space (of X) is a "bridge" connecting the ambient $\mathbb{R}^{n\times p}$ and $\mathcal{M}$. Before taking a step from current $X = X_t$ towards some X_{t+1}, we first want to find the deepest decreasing direction at the tangent plane of X_t.[a]

[a] By the informal definition of manifold.

Tangent space of an $X \in \mathcal{M}(= \mathcal{V}_p(\mathbb{R}^n))$



Tangent space of $X \in \mathcal{M}$

The tangent space (of X) is a "bridge" connecting the ambient $\mathbb{R}^{n\times p}$ and $\mathcal{M}$. Before taking a step from current $X = X_t$ towards some X_{t+1}, we first want to find the deepest decreasing direction at the tangent plane of X_t.[a]

[a] By the informal definition of manifold.

Pick any $X \in \mathcal{V}_p(\mathbb{R}^n)$, the tangent space of X is denoted as $T_X(\mathcal{V}_p(\mathbb{R}^n))$. Our next goal is to investigate the (directional) derivative starting from X towards the direction $Z \in T_X(\mathcal{V}_p(\mathbb{R}^n))$. Before that, we recall the usual case for $Z \in \mathbb{R}^{n\times p}$:

Directional derivative of F at $X \in \mathcal{M}$ towards direction $Z \in \mathbb{R}^{n\times p}$

$$\mathcal{D}F_X[Z] \triangleq \lim_{t\to 0^+} \frac{F(X+tZ)-F(X)}{t} = \langle \mathcal{D}F_X \cdot Z \rangle_E$$

Overview

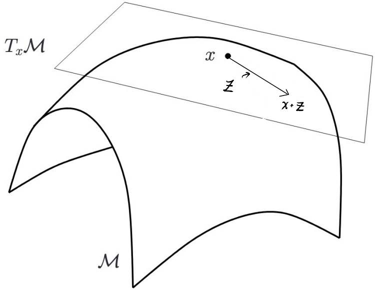
Characterising $T_X(\mathcal{V}_p(\mathbb{R}^n))$

For any $X \in \mathcal{V}_p(\mathbb{R}^n)$, we know $X = [x_1, ..., x_p]$ is made up of p orthonormal columns. Pick another $n - p$ new orthonormal vectors: $v_1, ..., v_{n-p}$ such that $\{x_1, ..., x_p, v_1, ..., v_{n-p}\}$ is a basis of $\mathbb{R}^n$. Define $X_\perp \triangleq [v_1, ..., v_{n-p}] \in \mathbb{R}^{n\times(n-p)}$.

Lemma

$Z = XA + X_\perp B \in T_X(\mathcal{V}_p(\mathbb{R}^n))$, if and only if $A = -A^T$, aka, A is a skew-symmetric matrix.

Side note: $0 \in T_X(\mathcal{V}_p(\mathbb{R}^n))$. The tangent space $T_X(\mathcal{V}_p(\mathbb{R}^n))$ takes X as origin.



Inner product representation of directional derivative

In previous slides, we stated that fix any $X \in \mathcal{V}_p(\mathbb{R}^n)$, for any $Z \in \mathbb{R}^{n\times p}$ (whatever direction),

$$\mathcal{D}F_X[Z] = \sum_{i,j} \frac{\partial F}{\partial X_{i,j}} Z_{i,j} = \mathbf{tr}(G^T Z) = \langle G \cdot Z \rangle_E \in \mathbb{R},$$

where $G = [\frac{\partial F}{\partial X_{i,j}}] \in \mathbb{R}^{n\times p}$. We always use G to denote the usual (Euclidean) gradient of F at X.

Inner product representation of directional derivative

In previous slides, we stated that fix any $X \in \mathcal{V}_p(\mathbb{R}^n)$, for any $Z \in \mathbb{R}^{n\times p}$ (whatever direction),

$$\mathcal{D}F_X[Z] = \sum_{i,j} \frac{\partial F}{\partial X_{i,j}} Z_{i,j} = \mathbf{tr}(G^T Z) = \langle G \cdot Z \rangle_E \in \mathbb{R},$$

where $G = [\frac{\partial F}{\partial X_{i,j}}] \in \mathbb{R}^{n\times p}$. We always use G to denote the usual (Euclidean) gradient of F at X.

Inner product representation

Fix any $X \in \mathcal{V}_p(\mathbb{R}^n)$, we say G represents[a] the (linear) operator $\mathcal{D}F_X : \underbrace{\mathbb{R}^{n\times p}}_{Z\in} \to \mathbb{R}$ – which is the directional derivative – under Euclidean standard inner product $\langle \cdot \rangle_E$.

[a] Because $\mathcal{D}F_X[Z] = \langle G \cdot Z \rangle_E$.

Inner product representation of directional derivative

In previous slides, we stated that fix any $X \in \mathcal{V}_p(\mathbb{R}^n)$, for any $Z \in \mathbb{R}^{n\times p}$ (whatever direction),

$$\mathcal{D}F_X[Z] = \sum_{i,j} \frac{\partial F}{\partial X_{i,j}} Z_{i,j} = \mathbf{tr}(G^T Z) = \langle G \cdot Z \rangle_E \in \mathbb{R},$$

where $G = [\frac{\partial F}{\partial X_{i,j}}] \in \mathbb{R}^{n\times p}$. We always use G to denote the usual (Euclidean) gradient of F at X.

Inner product representation

Fix any $X \in \mathcal{V}_p(\mathbb{R}^n)$, we say G represents[a] the (linear) operator $\mathcal{D}F_X : \underbrace{\mathbb{R}^{n\times p}}_{Z\in} \to \mathbb{R}$

– which is the directional derivative – under Euclidean standard inner product $\langle\cdot\rangle_E$.

[a] Because $\mathcal{D}F_X[Z] = \langle G \cdot Z \rangle_E$.

We then restrict $Z \in T_X \subset \mathbb{R}^{n\times p}$. Because we only move inside the T_X. We want to find (1) a matrix W, and (2) an inner product $\langle\cdot\rangle_C$, such that W represents $\mathcal{D}F_X|_{T_X} : T_X \to \mathbb{R}$ under that inner product. Aka, $\forall Z \in T_X$,

$$\mathcal{D}F_X[Z] = \langle W \cdot Z \rangle_C$$

Theorem (Representation of gradient only wrt directions in tangent space.)

Pick any $X \in \mathcal{M} = \mathcal{V}_p(\mathbb{R}^n)$. Analog to G represents the action of $\mathcal{D}F_X$ on $\mathbb{R}^{n\times p}$ under the standard matrix inner product $\langle\cdot\rangle_E$;
W, with $W \triangleq (GX^T - XG^T)X$, represents the action of $\mathcal{D}F_X$ on T_X under another inner product $\langle\cdot\rangle_C$ defined as $\langle Q \cdot H\rangle_C \triangleq \mathbf{tr}(Q^T(I - \frac{1}{2}XX^T)H)$.

Theorem (Representation of gradient only wrt directions in tangent space.)

Pick any $X \in \mathcal{M} = \mathcal{V}_p(\mathbb{R}^n)$. Analog to G represents the action of $\mathcal{D}F_X$ on $\mathbb{R}^{n\times p}$ under the standard matrix inner product $\langle\cdot\rangle_E$;
W, with $W \triangleq (GX^T - XG^T)X$, represents the action of $\mathcal{D}F_X$ on T_X under another inner product $\langle\cdot\rangle_C$ defined as $\langle Q \cdot H\rangle_C \triangleq \mathbf{tr}(Q^T(I - \frac{1}{2}XX^T)H)$.

Proof.

Write $G = XG_1 + X_\perp G_2$. Suppose $Z \in T_X$, then $Z = XZ_1 + X_\perp Z_2$, $Z_1 = -Z_1^T$. $\mathcal{D}F_X[Z] = \mathbf{tr}(G^TZ) = \mathbf{tr}(G_1^TZ_1) + \mathbf{tr}(G_2^TZ_2)$. Write $G_1 = G_{1,sym} + G_{1,skew}$. We proceed by $\mathbf{tr}(G_1^TZ_1) + \mathbf{tr}(G_2^TZ_2) = \mathbf{tr}(G_{1,skew}^TZ_1) + \mathbf{tr}(G_2^TZ_2)$.

We want to find some $U = XU_1 + X_\perp U_2 \in T_X$ (by previous lemma $U_1 = -U_1^T$) such that $\langle U \cdot Z\rangle_C = \mathcal{D}F_X[Z]$.

$\langle U \cdot Z\rangle_C = \mathbf{tr}((XU_1 + X_\perp U_2)^T(I - \frac{1}{2}XX^T)(XZ_1 + X_\perp Z_2)) = \frac{\mathbf{tr}(U_1^TZ_1)}{2} + \mathbf{tr}(U_2^TZ_2)$.
Set $U_1 \leftarrow 2G_{1,skew}$ and $U_2 \leftarrow G_2$. We surprisingly achieve that.

Last, rewrite U with X and $G = [\frac{\partial F}{\partial X_{ij}}]$. Using $G_{1,skew} = \frac{G_1 - G_1^T}{2} = \frac{X^TG - G^TX}{2}$.

$$U = 2XG_{1,skew} + X_\perp G_2 = G - XG^TX = (GX^T - XG^T)X.$$

□

Overview

Descent curve

Suppose now we are at some initial guess $X_0 \in \mathcal{M}$. And by previous analysis, we know the direction we should move (on the tangent plane) is $-W$.
Should we move to $\hat{X}_1 \leftarrow X_0 - \gamma_0 W = X_0 - \gamma_0(GX_0^T - X_0 G^T)X_0$, where γ_0 is some stepsize?

Not really, because ${\hat{X}_1}^T \hat{X}_1 \notin \mathcal{M}$. We are moving out of the manifold!

[4] Thanks to the smoothness of $\mathcal{M}$.

Descent curve

Suppose now we are at some initial guess $X_0 \in \mathcal{M}$. And by previous analysis, we know the direction we should move (on the tangent plane) is $-W$.
Should we move to $\hat{X}_1 \leftarrow X_0 - \gamma_0 W = X_0 - \gamma_0(GX_0^T - X_0 G^T)X_0$, where γ_0 is some stepsize?

Not really, because $\hat{X}_1{}^T \hat{X}_1 \notin \mathcal{M}$. We are moving out of the manifold!
We don't need to worry. Some genius came up with good ideas.
When we are at X, we can consider a curve parameterized by $\tau \geqslant 0$:

$$Y(\tau) = (I + \frac{\tau}{2}A)^{-1}(I - \frac{\tau}{2}A)X,$$

where $A = GX^T - XG^T$. Such curve has some good properties:

(1) $\forall \tau \geqslant 0$, $Y(\tau)^T Y(\tau) = I_p$. (one-line proof using $A = -A^T$)

(2) $Y'(0) = -AX$. It starts towards the deepest step on the tangent plane.

(3) $\frac{dF(Y(\tau))}{d\tau}|_{\tau=0} = -\frac{1}{2}\|A\|_F^2 < 0$ if $A \neq 0$.
Hence[4] $\exists$ some $\tau > 0$ such that $F(Y(\tau)) < F(X)$. (Such τ can be found via linear search.)

[4] Thanks to the smoothness of $\mathcal{M}$.

Retraction in general

In general, **retraction** is some map that sends out-of-manifold elements back to the manifold $\mathcal{M}$.

Cayley-transform based retraction

The retraction map we just discussed is termed as a "Cayley-transform" based one. Because "Cayley-transform" means something like:

$$X \to (I - U)(I + U)^{-1}X$$

with U being skew-symmetric.

Retraction in general

In general, **retraction** is some map that sends out-of-manifold elements back to the manifold $\mathcal{M}$.

Cayley-transform based retraction

The retraction map we just discussed is termed as a "Cayley-transform" based one. Because "Cayley-transform" means something like:

$$X \to (I - U)(I + U)^{-1} X$$

with U being skew-symmetric.

Of course, there are other kinds of retractions, e.g.,

Projection based retraction

Let $\xi \leftarrow -\gamma G$ (G always denote the usual gradient $G = [\frac{\partial F}{\partial X_{ij}}]$). One can also consider $X \to (X + \xi)(I + \xi^T \xi)^{-\frac{1}{2}}$. One can actually prove[a] $(X + \xi)(I + \xi^T \xi)^{-\frac{1}{2}} = \Pi_{\mathcal{V}_p(\mathbb{R}^n)}(X + \xi)$.

[a] Lemma 1 of "Weakly convex optimization over Stiefel Manifold using Riemannian subgradient-type methods", 2021

Overview

Simulation results on the "nearest correlation" problem

TABLE 5.6
Computational results for the Low-Rank nearest correlation problem.

	Major			PenCorr			Algorithm 2		
p	resi	CPU	feasi	resi	CPU	feasi	resi	CPU	feasi
Ex1, $H = I$									
2	1.564201e+02	8.34	1.2e-13	1.564172e+02	53.20	4.3e-09	1.686832e+02	0.25	3.5e-15
5	7.883390e+01	4.54	1.4e-13	7.883423e+01	10.34	3.1e-08	7.882875e+01	1.12	3.7e-15
10	3.868525e+01	5.25	5.2e-15	3.868518e+01	6.30	2.9e-07	3.868258e+01	1.01	4.0e-15
20	1.570825e+01	12.64	5.6e-15	1.570796e+01	5.31	7.8e-08	1.570688e+01	1.26	5.1e-15
50	4.140789e+00	142.30	7.1e-15	4.139455e+00	2.29	5.2e-07	4.139235e+00	5.71	5.9e-15
100	1.471204e+00	928.58	9.5e-15	1.466498e+00	2.56	2.4e-07	1.467395e+00	19.46	8.0e-15
125	1.055070e+00	1731.39	9.4e-15	1.048114e+00	2.85	3.0e-08	1.049154e+00	24.65	8.1e-15
Ex1, random H									
2	9.106583e+02	14.64	1.4e-13	9.109902e+02	163.18	4.6e-07	9.778999e+02	0.76	3.6e-15
5	4.536099e+02	31.42	1.2e-13	4.537458e+02	91.15	7.7e-07	4.535966e+02	2.04	3.9e-15
10	2.204165e+02	67.21	5.1e-15	2.204421e+02	69.79	5.4e-07	2.204043e+02	2.08	4.2e-15
20	8.812054e+01	218.99	5.4e-15	8.812887e+01	69.65	3.6e-07	8.851307e+01	3.36	4.9e-15
50	2.203931e+01	2022.31	7.1e-15	2.191649e+01	94.46	8.6e-07	2.188864e+01	27.36	6.0e-15
100	7.110542e+00	9649.30	9.2e-15	6.389784e+00	121.00	2.6e-07	6.457456e+00	39.97	8.0e-15
125	5.030963e+00	13801.88	1.0e-14	4.179018e+00	151.09	9.6e-07	4.348972e+00	48.05	8.8e-15
Ex2, $H = I$									
5	4.127677e+02	10.63	1.6e-13	4.128428e+02	172.70	6.5e-08	4.127542e+02	2.72	5.3e-15
10	3.265122e+02	46.72	6.2e-15	3.263352e+02	185.19	1.2e-07	3.262344e+02	2.73	5.6e-15
20	2.887315e+02	27.11	7.6e-15	2.887228e+02	88.19	3.2e-08	2.886782e+02	9.87	6.6e-15
50	2.763023e+02	78.99	1.0e-14	2.762742e+02	51.40	7.5e-08	2.762733e+02	11.74	8.8e-15
100	2.758067e+02	260.08	1.4e-14	2.757853e+02	9.44	1.7e-08	2.757854e+02	6.35	1.1e-14
150	2.758095e+02	925.63	1.7e-14	2.757853e+02	9.44	1.7e-08	2.757854e+02	7.40	1.3e-14
250	2.758087e+02	1676.32	2.1e-14	2.757853e+02	9.44	1.7e-08	2.757854e+02	11.11	1.6e-14
Ex2, H given by T. Fushiki									
5	1.141113e+04	407.47	2.6e-13	1.147849e+04	2480.31	5.7e-07	1.140483e+04	35.89	5.1e-15
10	7.586522e+03	1299.88	6.1e-15	7.638036e+03	2305.20	2.8e-07	7.586921e+03	33.17	5.8e-15
20	5.200671e+03	1733.57	7.3e-15	5.219117e+03	2062.14	2.7e-07	5.194958e+03	30.54	6.1e-15
50	3.712916e+03	4154.29	9.0e-15	3.718507e+03	1356.16	2.0e-09	3.711896e+03	21.99	7.8e-15
100	3.503209e+03	6426.36	1.2e-14	3.507128e+03	906.02	3.3e-07	3.502541e+03	22.05	9.7e-15
150	3.501124e+03	8734.17	1.6e-14	3.505351e+03	824.69	3.1e-07	3.500676e+03	27.78	1.1e-14
250	3.501170e+03	15595.92	2.0e-14	3.505351e+03	864.67	3.1e-07	3.500654e+03	40.59	1.4e-14

Remarks

"Algorithm 2" is the proposed method.

resi: objective value.
CPU: running time (sec).
feasi: infeasibility.

All 3 metrics are "the lower the better".

Improvement is achieved in running time, especially for the nontrivial H cases.[a]

[a] Thanks to an efficient implementation of the original Cayley transform.

Codebase repository

The authors put their MatLab codes – with quite high-quality implementation – of this problem to GitHub.[5] There are some more extensions, but the usage are similar.

$$\min_{X \in \mathbb{R}^{n \times p}} F(X) \qquad \text{subject to } X^T X = I_p$$

Anyone can use their API in the following way:

```
X0 = 1/np.sqrt(n) * np.ones((n, p));
opts.record = 0;
opts.mxitr  = 2000;
opts.xtol = 1e-20;
opts.gtol = 1e-20;
opts.ftol = 1e-20;

[X, out]= OptStiefelGBB(X0, @fun, opts, arg1, arg2, arg3);


    function [F, G] = fun(X, arg1, arg2, arg3)
            F = ... ## Define your function here
            G = ... ## Calculate the usual gradient here
    end
```

[5] https://github.com/optsuite/OptM