A C++ Riemannian Optimization Library

Wen Huang[1], P.-A. Absil[1], Kyle A. Gallivan[2]

1- Université Catholique de Louvain 2- Florida State University

19 May 2015

Goal

Develop a library to find an optimum of a real-valued function f on a Riemannian manifold, i.e.,

$$\min f(x), x \in \mathcal{M}.$$

Many libraries exist, e.g. ManOpt [BMAS14].

- Reliable computational time
- Interfaces for various languages users
- Can be built in other packages

C++ Package

Available on `http://www.math.fsu.edu/~whuang2/ROPTLIB`

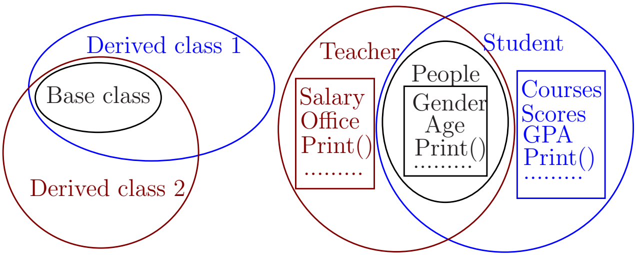
- C++ is a popular and object-oriented programming language
    - Not difficult to maintain
    - Built in other packages
    - Reliable computational time
- Use standard linear algebra packages, BLAS and LAPACK

Framework

The framework partly inspires by ManOpt [BMAS14] and GenRTR [ABG07], and include four parts:

- Solvers: State-of-the-art algorithms
- Space: Storing elements on manifolds, tangent vectors and linear operators
- Manifold: Operations of manifolds
- Problem: Cost function, gradient, etc.

Inheritance



- Multiple base classes $\rightarrow$ a derived class
- Make it easy to maintain the code
- Overwrite a function, e.g. Print()

Solvers

Table: Riemannian algorithms in the package

Riemannian trust-region Newton (RTRNewton)	[ABG07]
Riemannian trust-region symmetric rank-one update (RTRSR1)	[HAG15]
Limited-memory RTRSR1 (LRTRSR1)	[HAG15]
Riemannian trust-region steepest descent (RTRSD)	[AMS08]
Riemannian line-search Newton (RNewton)	[AMS08]
Riemannian Broyden family (RBroydenFamily)	[HGA14]
Riemannian BFGS (RWRBFGS and RBFGS)	[RW12, HGA14]
Limited-memory RBFGS (LRBFGS)	[HGA14]
Riemannian conjugate gradients (RCG)	[NW06, AMS08, SI13]
Riemannian steepest descent (RSD)	[AMS08]

Solvers

- Line search based methods
  - $x_{k+1} = R_{x_k}(t_k \eta_k)$
  - Line search algorithm is used to find a step size t_k
  - Different algorithms use different search direction η_k
- Trust region based methods
  - Approximately solve the local model $\eta_k = \mathrm{argmin}_{\|\eta\| \in \mathbb{D}} f(x_k) + g(\mathrm{grad}\, f(x_k), \eta) + \frac{1}{2} g(\mathcal{B}_k \eta, \eta)$ and accept or reject $\tilde{x}_{k+1} = R_{x_k}(\eta_k)$ based on the quality of the approximation
  - $\mathcal{B}_k$ is the Hessian approximation
  - Different algorithms use different Hessian approximation

Solvers

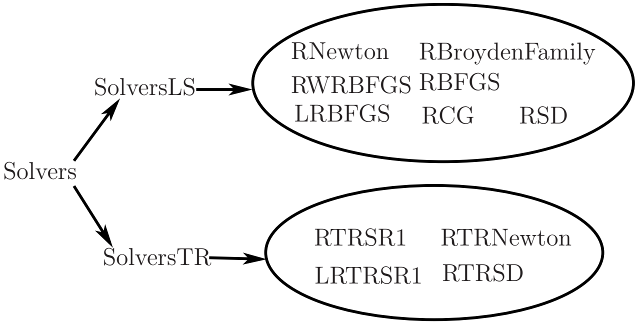


Figure: Relationships among classes of solvers in the package. Arrows are from base class to derived class.

Solvers

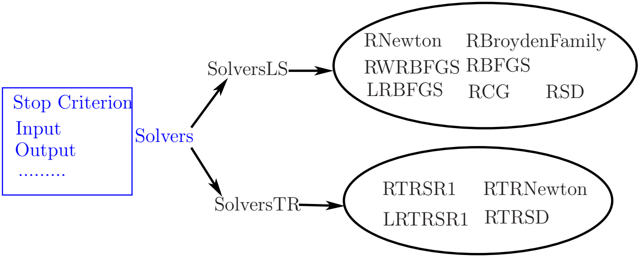


Figure: Relationships among classes of solvers in the package. Arrows are from base class to derived class.

Solvers

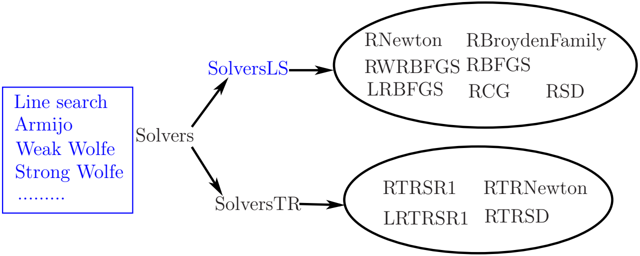


Figure: Relationships among classes of solvers in the package. Arrows are from base class to derived class.

Solvers

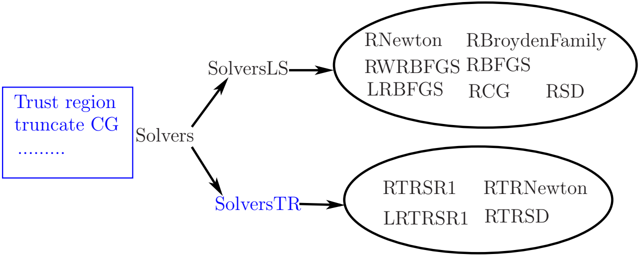


Figure: Relationships among classes of solvers in the package. Arrows are from base class to derived class.

Solvers

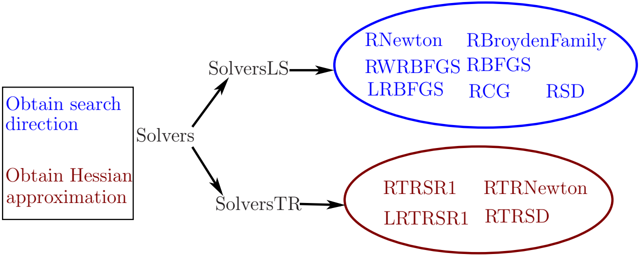


Figure: Relationships among classes of solvers in the package. Arrows are from base class to derived class.

Space

- Copy-on-Write strategy is used

```
>> A = randn(1000, 1000);
>> tic; B = A; toc          %% 0.000006 seconds
>> tic; B(1,1) = 1; toc %% 0.006373 seconds.
```

- Elements on Product of manifolds
  - Consecutive memory
  - Spatial locality
- Shared memory

Memory of $x \in \mathcal{M}$

x	x related temp data

Space

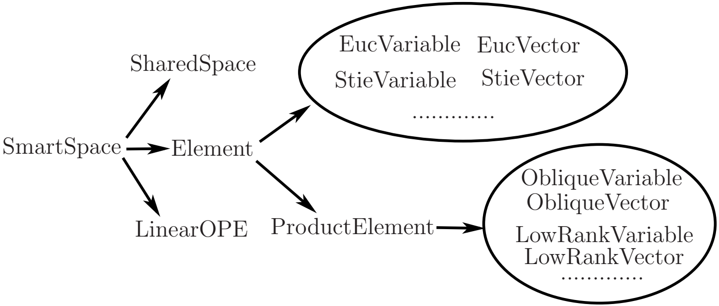


Figure: Relationships among classes of variables and tangent vectors in the package. Arrows are from base class to derived class.

Space

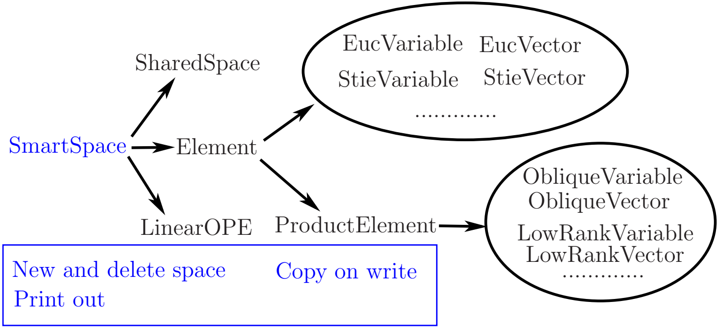


Figure: Relationships among classes of variables and tangent vectors in the package. Arrows are from base class to derived class.

Space

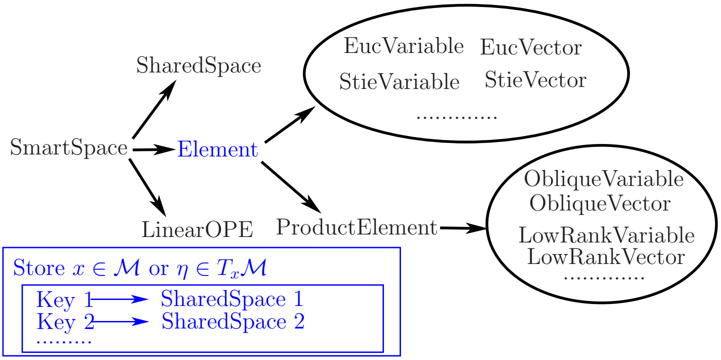


Figure: Relationships among classes of variables and tangent vectors in the package. Arrows are from base class to derived class.

Space

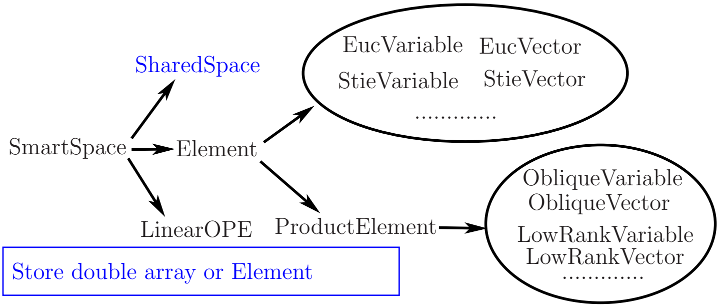


Figure: Relationships among classes of variables and tangent vectors in the package. Arrows are from base class to derived class.

Space

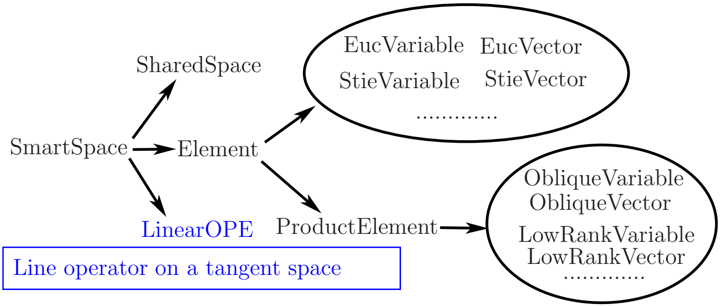


Figure: Relationships among classes of variables and tangent vectors in the package. Arrows are from base class to derived class.

Space

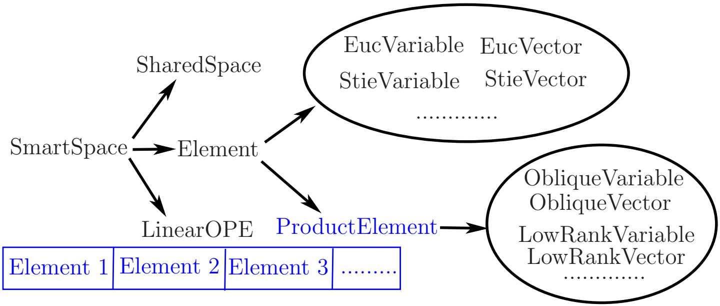


Figure: Relationships among classes of variables and tangent vectors in the package. Arrows are from base class to derived class.

Manifold

Define basic operations on Manifold

- Metric
- Retraction
- Vector transport
- Projection onto tangent space
- Euclidean gradient to Riemannian gradient
- Euclidean Hessian to Riemannian Hessian
- etc

Provide functions to check correctness of operations.

Manifold

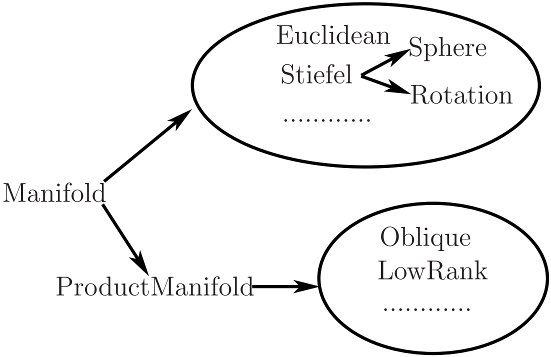
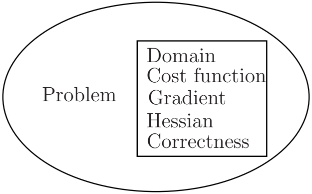


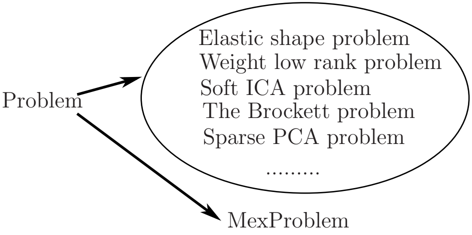
Figure: Relationships among classes of Manifolds in the package. Arrows are from base class to derived class.

Problem



- Define cost function, gradient and action of Hessian
- Convert Euclidean gradient and action of Euclidean Hessian to Riemannian gradient and action of Riemannian Hessian
- Check correctness of gradient and action of Hessian

Problem



- "MexProblem" is the bridge between C++ and Matlab
- It converts function handles of Matlab to C++ functions

Installation

Only Matlab environment is shown.

- Set up the mex environment properly. Follow the webpage: `www.mathworks.com/support/compilers/R2014b/index.html`
- Run "GenerateMyMex.m"
- Use "MyMex.m" to compile code

```
>> GenerateMyMex
 Generate MyMex.m file ...
>> MyMex TestStieBrockett
 Building with 'g++-4.7'.
MEX completed successfully.
```

Interface of Matlab

- Run "MyMex DriverMexProb" to obtain the driver "DriverMexProb" for Matlab
- "DriverMexProb" is wrapped by the matlab script "DriverOPT.m"
- "DriverOPT.m" can be called by

```
[Xf, fv, gfv, gf0, iter, nf, ng, nR, nV, nVp, nH, time, FS, GFS, TS] =
DriverOPT(fh, gfh, Hh, SolverParams, ManiParams, HasHHR, initialX)
```

An Example

- The Brockett cost function: Minimize

$$\mathrm{trace}(X^T BXD) \tag{1}$$

  such that $x \in \mathrm{St}(p, n)$, where $B \in \mathbb{R}^{n\times n}$, $B = B^T$, $D = \mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_p)$ and $\mu_1 > \mu_2 > \ldots > \mu_p$.
- The columns of a global minimizer, $X^* e_i$, are eigenvectors for the p smallest eigenvalues, λ_i, ordered so that $\lambda_1 \leq \cdots \leq \lambda_p$ [AMS08, §4.8].

Interface of Matlab

```
function output = testBrockett()
  n = 5; p = 2;                    % size of the Stiefel manifold
  B = randn(n, n); B = B + B';     % data matrix
  D = sparse(diag(p : -1 : 1));    % data matrix
  fhandle = @(x)f(x, B, D);        % cost function handle
  gfhandle = @(x)gf(x, B, D);      % gradient
  Hesshandle = @(x, eta)Hess(x, eta, B, D); % Hessian

  SolverParams.method = 'RSD';     % Use RSD solver
  ManiParams.name = 'Stiefel';     % Domain is the Stiefel manifold
  ManiParams.n = n;                % assign size to manifold parameter
  ManiParams.p = p;                % assign size to manifold parameter
  initialX.main = orth(randn(n, p));  % initial iterate
  % call the driver
  output = DriverOPT(fhandle, gfhandle, Hesshandle, SolverParams, ManiParams, initialX);
end

function [output, x] = f(x, B, D)
  x.BUD = B * x.main * D;
  output = x.main(:)' * x.BUD(:);
end

function [output, x] = gf(x, B, D)
  output.main = 2 * x.BUD;
end

function [output, x] = Hess(x, eta, B, D)
  output.main = 2 * B * eta.main * D;
end
```

An Example

- Summation of three Brockett cost functions: Minimize

$$\mathrm{trace}(X_1^T B_1 X_1 D_1) + \mathrm{trace}(X_2^T B_2 X_2 D_2) + \mathrm{trace}(X_3^T B_3 X_3 D_3) \tag{2}$$

such that $(X_1, X_2, X_3) \in \mathrm{St}(p, n) \times \mathrm{St}(p, n) \times \mathrm{St}(q, m)$, where $B_1, B_2 \in \mathbb{R}^{n\times n}$, $B_3 \in \mathbb{R}^{m\times m}$, $B_1 = B_1^T$, $B_2 = B_2^T$, $B_3 = B_3^T$, $D_1 = \mathrm{diag}(\mu_1, \mu_2, \dots, \mu_p)$, $\mu_1 \geq \mu_2 \geq \dots \geq \mu_p$, $D_2 = \mathrm{diag}(\nu_1, \nu_2, \dots, \nu_p)$, $\nu_1 \geq \nu_2 \geq \dots \geq \nu_p$, $D_3 = \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_q)$, and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_q$.

Future Work

- Interface with ManOpt
- More manifolds
- Interfaces for other languages, e.g., python
- Automatic differentiation

References I

P.-A. Absil, C. G. Baker, and K. A. Gallivan.
Trust-region methods on Riemannian manifolds.
Foundations of Computational Mathematics, 7(3):303–330, 2007.

P.-A. Absil, R. Mahony, and R. Sepulchre.
Optimization algorithms on matrix manifolds.
Princeton University Press, Princeton, NJ, 2008.

N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre.
Manopt, a Matlab toolbox for optimization on manifolds.
Journal of Machine Learning Research, 15(1455-1459), 2014.

W. Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian symmetric rank-one trust-region method.
Mathematical Programming, 150(2):179–216, February 2015.

W. Huang, K. A. Gallivan, and P.-A. Absil.
A Broyden class of quasi-Newton methods for Riemannian optimization.
Submitted for publication, 2014.

J. Nocedal and S. J. Wright.
Numerical optimization.
Springer, second edition, 2006.

References II

W. Ring and B. Wirth.
Optimization methods on Riemannian manifolds and their application to shape space.
SIAM Journal on Optimization, 22(2):596–627, January 2012.
doi:10.1137/11082885X.

H. Sato and T. Iwai.
A new, globally convergent riemannian conjugate gradient method.
Optimization, page 22, February 2013.
1302.0125.