Riemannian Optimization with its Application to Clustering Problems

Speaker: Wen Huang

Xiamen University

August 23, 2023

Guilin University Of Electronic Technology

Outline

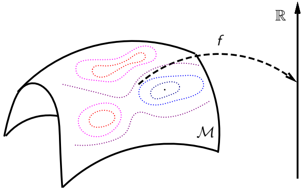
- Problem statement
- Motivation
- Smooth optimization framework
- Literature review
- A Riemannian optimization approach to clustering problems
- Riemannian proximal gradient methods
- Numerical experiments

Problem Statement

Problem: Given $f(x) : \mathcal{M} \to \mathbb{R}$, solve

$$\min_{x \in \mathcal{M}} f(x)$$

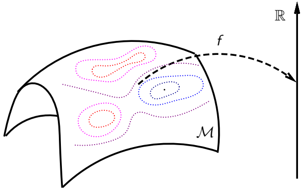
where $\mathcal{M}$ is a Riemannian manifold.



Problem: Given $f(x) : \mathcal{M} \to \mathbb{R}$, solve

$$\min_{x \in \mathcal{M}} f(x)$$

where $\mathcal{M}$ is a Riemannian manifold.

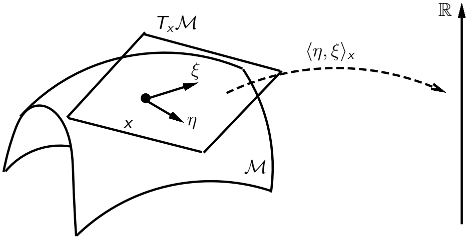


Manifolds:

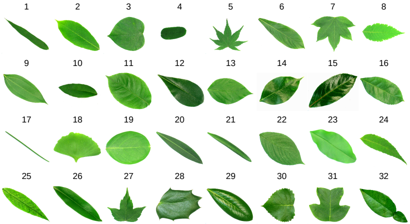
- Stiefel: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\}$;
- Grassmann: the set of p dimensional linear spaces in $\mathbb{R}^n$;
- Fixed rank: $\mathbb{R}_r^{m \times n} = \{X \in \mathbb{R}^{m \times n} : \mathrm{rank}(X) = r\}$ or tensor;
- Symmetric positive definite: $\mathcal{S}_{++}^n = \{X \in \mathbb{R}^{n \times n} : X \succ 0\}$;
- And many more;

Riemannian Manifolds

Roughly, a Riemannian manifold $\mathcal{M}$ is a smooth set with a smoothly-varying inner product on the tangent spaces.



Riemannian manifold $=$ Manifold $+$ Riemannian metric (inner products)



- Classification [LKS+12, HGSA15]
- Face recognition [DBS+13]

- Elastic shape analysis invariants:
    - Rescaling
    - Translation
    - Rotation
    - Reparametrization
- The shape space is a quotient space

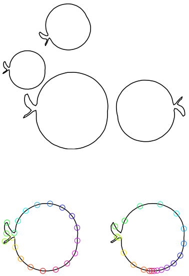
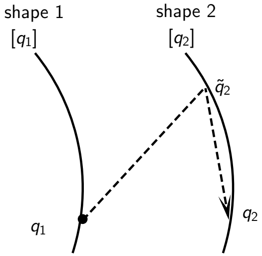
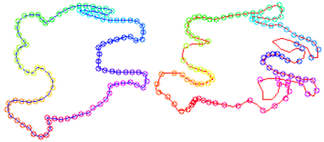


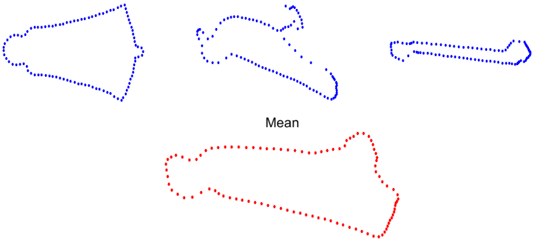
Figure: All are the same shape.





- Optimization problem $\min_{q_2 \in [q_2]} \mathrm{dist}(q_1, q_2)$ is defined on a Riemannian manifold

- Computation of a geodesic between two shapes
- Interpolation in shape space



- Computation of Karcher mean of a population of shapes

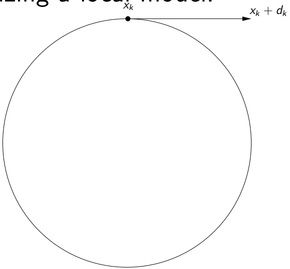
- Role model extraction
- Computations on SPD matrices
- Blind source separation
- Phase retrieval problem
- Blind deconvolution
- Synchronization of rotations
- Computations on low-rank tensor
- Low-rank approximate solution for Lyapunov equation

Consider the following generic update for an iterative Euclidean optimization algorithm:

$$x_{k+1} = x_k + \Delta x_k = x_k + \alpha_k s_k \ .$$

This iteration is implemented in numerous ways, e.g.:

- Steepest descent: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- Newton's method: $x_{k+1} = x_k - \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$
- Trust region method: Δx_k is set by optimizing a local model.



Riemannian Manifolds Provide

- Riemannian concepts describing directions and movement on the manifold
- Riemannian analogues for gradient and Hessian

Definition

The Riemannian gradient of f at x is the unique tangent vector in $\mathrm{T}_x \mathcal{M}$ satisfying $\forall \eta \in \mathrm{T}_x \mathcal{M}$, the directional derivative

$$\mathrm{D} f(x)[\eta] = \langle \operatorname{grad} f(x), \eta \rangle$$

and $\operatorname{grad} f(x)$ is the direction of steepest ascent.

Definition

The Riemannian Hessian of f at x is a symmetric linear operator from $\mathrm{T}_x \mathcal{M}$ to $\mathrm{T}_x \mathcal{M}$ defined as

$$\operatorname{Hess} f(x) : \mathrm{T}_x \mathcal{M} \to \mathrm{T}_x \mathcal{M} : \eta \to \nabla_\eta \operatorname{grad} f,$$

where ∇ is the affine connection.

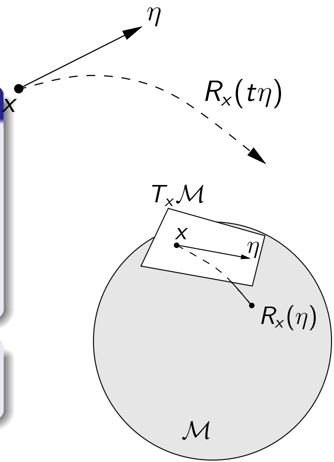
Euclidean	Riemannian
$x_{k+1} = x_k + \alpha_k d_k$	$x_{k+1} = R_{x_k}(\alpha_k \eta_k)$

Definition

A retraction is a mapping R from $\mathrm{T}\mathcal{M}$ to $\mathcal{M}$ satisfying the following:

- R is continuously differentiable
- $R_x(0) = x$
- $\mathrm{D}\, R_x(0)[\eta] = \eta$

- maps tangent vectors back to the manifold
- defines curves in a direction



Retraction-based: local information only

Line search-based: use local tangent vector and $R_x(t\eta)$ to define line

- Steepest decent
- Newton

Local model-based: series of flat space problems

- Riemannian trust region Newton (RTR)
- Riemannian adaptive cubic overestimation (RACO)

Retraction and transport-based: information from multiple tangent spaces

- Nonlinear conjugate gradient: multiple tangent vectors
- Quasi-Newton e.g. Riemannian BFGS: transport operators between tangent spaces

Additional element required for optimizing a cost function;

- formulas for combining information from multiple tangent spaces.

Vector Transport

- Vector transport: Transport a tangent vector from one tangent space to another
- $\mathcal{T}_{\eta_x}\xi_x$, denotes transport of ξ_x to tangent space of $R_x(\eta_x)$. R is a retraction associated with $\mathcal{T}$

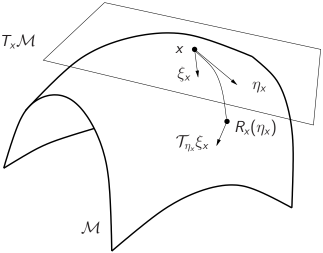


Figure: Vector transport.

Given a retraction and a vector transport, we can generalize many Euclidean methods to the Riemannian setting. Do the Riemannian versions of the methods work well?

Given a retraction and a vector transport, we can generalize many Euclidean methods to the Riemannian setting. Do the Riemannian versions of the methods work well?

No

- Lose many theoretical results and important properties;
- Impose restrictions on retraction/vector transport;

Elements required for optimizing a cost function $(\mathcal{M}, g)$:

- an representation for points x on $\mathcal{M}$, for tangent spaces $\mathrm{T}_x \mathcal{M}$, and for the inner products $g_x(\cdot, \cdot)$ on $\mathrm{T}_x \mathcal{M}$;
- choice of a retraction $R_x : \mathrm{T}_x \mathcal{M} \to \mathcal{M}$;
- formulas for $f(x), \mathrm{grad}\; f(x)$ and $\mathrm{Hess}\; f(x)$ (or its action);
- Computational and storage efficiency;

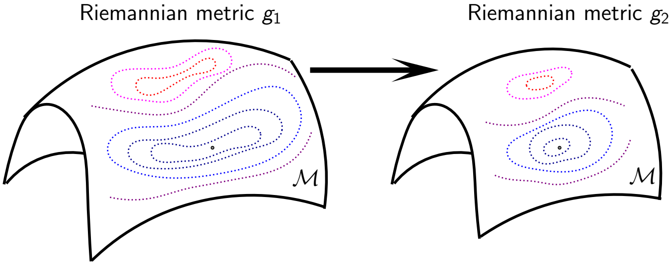
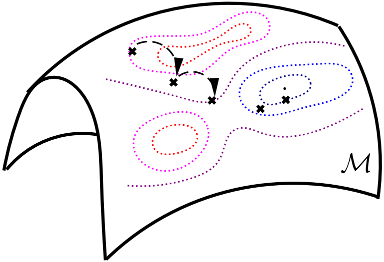


Figure: Changing metric may influence the difficulty of a problem.

Riemannian metric influences

- Riemannian gradient
- Riemannian Hessian

- All iterates on the manifold
- Convergence properties of unconstrained optimization algorithms
- No need to consider Lagrange multipliers or penalty functions
- Exploit the structure of the constrained set



Benefits

- Increased generality does not compromise the important theory
- Less expensive than or similar to previous approaches
- May provide theory to explain behavior of algorithms specifically developed for a particular application – or closely related ones

Possible Problems

- May be inefficient compared to algorithms that exploit application details

- Smooth unconstrained problems
  - Steepest descent: Smith 1994; Helmke-Moore 1994; Iannazzo-Porcelli 2019;
  - Conjugate gradient: Smith 1994; Gallivan-Absil 2010; Ring-Wirth 2012; Sato-Iwai 2015;
  - Quasi-Newton: Ring-Wirth 2012; Huang-Absil-Gallivan 2018; Huang-Gallivan 2022
  - Trust region Newton: Absil-Baker-Gallivan 2007;
- Nonsmooth unconstrained problems
  - Proximal point method: Ferreira-Oliveira 2002;
  - Optimality conditions: Yang-Zhang-Song 2014;
  - Gradient sampling: Huang 2013; Hosseini and Uschmajew 2017;
  - ϵ-subgradient-based methods: Grohs-Hosseini 2015;
  - Proximal gradient methods: Huang-Wei 2022;
- Constrained problems:
  - Augmented Lagrangian methods: Boumal-Liu 2019;

A non-exhaustive review

Some History of Optimization On Manifolds

- Smooth unconstrained problems:
  - Stiefel manifold: Wen-Yin 2012; Jiang-Dai 2014; Xiao-Liu-Yuan 2020; Dai-Wang-Zhou 2020
  - Symmetric positive definite manifold: Bini-Iannazzo 2013; Zhang 2017; Yuan-Huang-Absil-Gallivan 2020;
  - Fixed rank manifold: Wen-Yin-Zhang 2012; Mishra 2014; Sutti-Vandereycken 2021; Levin-Kileel-Boumal 2022

- Nonsmooth unconstrained problems:
  - Stiefel Manifold: Huang-Wei 2019; Chen-Ma-So-Zhang 2020; Xiao-Liu-Yuan 2020;
  - Fixed rank manifold: Cambier-Absil 2016;
  - Matrix manifolds: Zhou-Bao-Ding-Zhu 2022

- Constrained problems:
  - Stiefel + non-negativity: Jiang-Meng-Wen-Chen 2019;
  - Symmetric positive definite + zeros: Phan-Menickelly 2020;

Riemannian optimization libraries for general problems:

- Boumal, Mishra, Absil, Sepulchre(2014)
  Manopt (Matlab library)
- Townsend, Koep, Weichwald (2016)
  Pymanopt (Python version of manopt)
- Bergmann (2019)
  Manoptjl (Julia, nonsmooth methods)
- Huang, Absil, Gallivan, Hand (2018)
  ROPTLIB (C++ library, interfaces to Matlab and Julia)
- Martin, Raim, Huang, Adragni (2018)
  ManifoldOptim (R wrapper of ROPTLIB)
- Meghawanshi, Jawanpuria, Kunchukuttan, Kasai, Mishra (2018)
  McTorch (Python, GPU acceleration)

- Smooth unconstrained problems
  - Broyden family including BFGS method [HGA15, HAG17, HAG18]
  - Trust-region symmetric rank-one method [HAG15]
  - Their limited-memory versions [HG22]
- Nonsmooth unconstrained problems
  - ϵ-subgradient with quasi-Newton method [HHY18]
  - Proximal gradient methods [HW21a]
  - Proximal Newton method [SAH$^+$23]
- Applications:
  - Elastic shape analysis [HGSA15]
  - Blind deconvolution [HH18]
  - Phase retrieval [HGZ16]
  - Sparse principal component analysis [HW21c]
  - Gray/color image completion [CH23, PH23]
- Library: ROPTLIB [HAGH18]

Problem Statement

Clustering problems

The task of clustering is to group a set of objects such that the objects in the same group are more similar or closely connected under certain criterion to each other than to those in other groups.

Problem Statement

Clustering problems

The task of clustering is to group a set of objects such that the objects in the same group are more similar or closely connected under certain criterion to each other than to those in other groups.

Clustering problems that can be formulated as

$$\min_{X \in \mathcal{A}_{n,k}} f(X),$$

where $\mathcal{A}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, X \geq 0, \mathbf{1}_n \in \mathrm{span}(X)\}$.

Problem Statement

Clustering problems

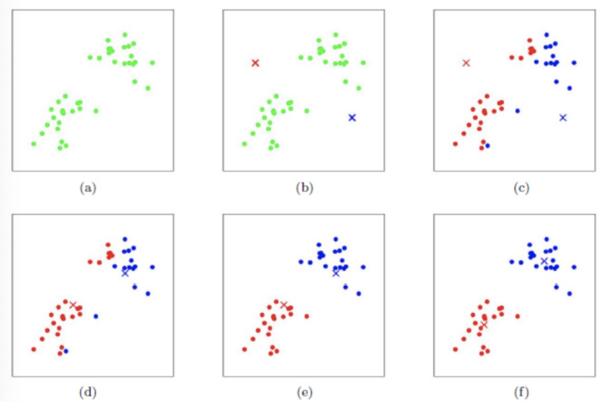
The task of clustering is to group a set of objects such that the objects in the same group are more similar or closely connected under certain criterion to each other than to those in other groups.

Clustering problems that can be formulated as

$$\min_{X \in \mathcal{A}_{n,k}} f(X),$$

where $\mathcal{A}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, X \geq 0, \mathbf{1}_n \in \mathrm{span}(X)\}$.

- Spectral clustering
- Normalized cuts
- k-means
- Community detection
- Etc



0 Initial estimations for the means

1 Assign points to their closest means and creates groups

2 Means are updated by computing the means of the new groups

[0]The figure is from `https://www.cnblogs.com/xiaxuexiaoab/p/10211279.html`

0 Initial estimations for the means

---

n points a_i in $\mathbb{R}^d$ represented by $A = [a_1, a_2, \ldots, a_n]^T \in \mathbb{R}^{n \times d}$, k clusters;

0 initial k means, $M = [m_1, m_2, \ldots, m_k]^T \in \mathbb{R}^{k \times d}$;

0 Initial estimations for the means

1 Assign points to their closest means and creates groups

---

n points a_i in $\mathbb{R}^d$ represented by $A = [a_1, a_2, \ldots, a_n]^T \in \mathbb{R}^{n \times d}$, k clusters;

0 initial k means, $M = [m_1, m_2, \ldots, m_k]^T \in \mathbb{R}^{k \times d}$;

1 Find an indicator matrix $Y \in \mathbb{R}^{n \times k}$ such that
$Y = \mathrm{argmin}_Y \|A - YM\|_F^2$;

0 Initial estimations for the means
1 Assign points to their closest means and creates groups
2 Means are updated by computing the means of the new groups

---

n points a_i in $\mathbb{R}^d$ represented by $A = [a_1, a_2, \ldots, a_n]^T \in \mathbb{R}^{n \times d}$, k clusters;

0 initial k means, $M = [m_1, m_2, \ldots, m_k]^T \in \mathbb{R}^{k \times d}$;
1 Find an indicator matrix $Y \in \mathbb{R}^{n \times k}$ such that $Y = \mathrm{argmin}_Y \|A - YM\|_F^2$;
2 The new means: $M_+ = \mathrm{argmin}_{M \in \mathbb{R}^{k \times d}} \|A - YM\|_F^2 \Rightarrow M_+ = (Y^T Y)^{-1} Y^T A$

Problem Statement
A clustering problem: k-means

0 Initial estimations for the means
1 Assign points to their closest means and creates groups
2 Means are updated by computing the means of the new groups

---

n points a_i in $\mathbb{R}^d$ represented by $A = [a_1, a_2, \dots, a_n]^T \in \mathbb{R}^{n \times d}$, k clusters;

0 initial k means, $M = [m_1, m_2, \dots, m_k]^T \in \mathbb{R}^{k \times d}$;
1 Find an indicator matrix $Y \in \mathbb{R}^{n \times k}$ such that $Y = \mathrm{argmin}_Y \|A - YM\|_F^2$;
2 The new means: $M_+ = \mathrm{argmin}_{M \in \mathbb{R}^{k \times d}} \|A - YM\|_F^2 \Rightarrow M_+ = (Y^T Y)^{-1} Y^T A$

Optimization problem [BDM09]:

$$\min_Y \|A - Y(Y^T Y)^{-1} Y^T A\|_F^2,$$

where Y is an indicator matrix

Optimization problem:

$$\min_{Y} \|A - Y(Y^T Y)^{-1} Y^T A\|_F^2 \Longleftrightarrow \min_{X \in \mathcal{A}_{n,k}} \|A - XX^T A\|_F^2$$

where $\mathcal{A}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, X \geq 0, \mathbf{1}_n \in \mathrm{span}(X)\}$.

For $X \in \mathcal{A}_{n,k}$,

- Only one entry is nonzero in each row
- All positive entries in a column have the same value
- $X_{ij} \neq 0$ implies that point i is in the cluster j

Optimization problem:

$$\min_{Y} \|A - Y(Y^T Y)^{-1} Y^T A\|_F^2 \Longleftrightarrow \min_{X \in \mathcal{A}_{n,k}} \|A - XX^T A\|_F^2$$

where $\mathcal{A}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, X \geq 0, \mathbf{1}_n \in \mathrm{span}(X)\}$.

For $X \in \mathcal{A}_{n,k}$,

- Only one entry is nonzero in each row
- All positive entries in a column have the same value
- $X_{ij} \neq 0$ implies that point i is in the cluster j

The optimization problem is in the form of

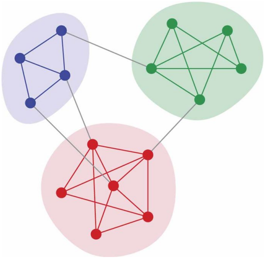
$$\min_{X \in \mathcal{A}_{n,k}} f(X),$$

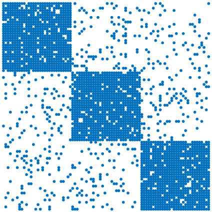
where f is smooth.

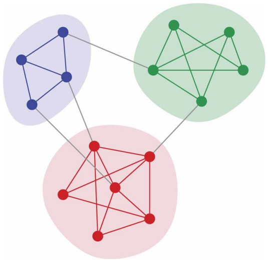
Problem Statement

A clustering problem: community detection

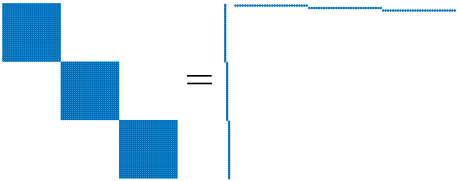
- Adjacency matrix $A \in \mathbb{R}^{n \times n}$ (Undirected)







- Adjacency matrix $A \in \mathbb{R}^{n \times n}$ (Undirected)
- Ideal adjacency matrix $A = ZZ^T$
- $Z \in \mathbb{R}^{n \times k}$ defines the communities



Existing methods:

- The GN algorithm [New04]
- The spectral modularity maximization algorithm [New06]
- The Louvain method [BGLL08]
- The infomap algorithm [RB08]
- Statistical inference [NL07]
- Deep learning [YCH+16].

Existing methods:

- The GN algorithm [New04]
- The spectral modularity maximization algorithm [New06]
- The Louvain method [BGLL08]
- The infomap algorithm [RB08]
- Statistical inference [NL07]
- Deep learning [YCH+16].

Modularity optimization approaches have shown to be highly effective [For10]

Problem Statement

A clustering problem: community detection

Maximize modularity:

$$\tilde{f} : \tilde{\mathcal{A}}_{n,k} \to \mathbb{R} : Y \mapsto \mathrm{trace}(Y^T M Y),$$

where $M = A - \frac{A\mathbf{1}_n\mathbf{1}_n^T A}{\mathbf{1}_n^T A \mathbf{1}_n}$ and $\tilde{\mathcal{A}}_{n,k}$ is the set of indicator matrices.

For ideal graph:

- $A = ZZ^T$
- The global minimizer of $\tilde{f}$ is Z
- $Z_{ij} = 1$ implies that node i is in the community j

Problem Statement

A clustering problem: community detection

Maximize modularity with modifications [WHGVD21]:

$$\tilde{f} : \mathcal{A}_{n,k} \to \mathbb{R} : X \mapsto \mathrm{trace}(X^T M X),$$

where $\mathcal{A}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, X \geq 0, \mathbf{1}_n \in \mathrm{span}(X)\}$ and $M = A - \frac{A \mathbf{1}_n \mathbf{1}_n^T A}{\mathbf{1}_n^T A \mathbf{1}_n}$

For idea graph, i.e., $A = ZZ^T$, it can be proven that the maximizer $\tilde{Z}$ of f is given by normalizing the columns of Z. Therefore, $\tilde{Z}$ defines the same communities.

Maximize modularity with modifications [WHGVD21]:

$$\tilde{f} : \mathcal{A}_{n,k} \to \mathbb{R} : X \mapsto \mathrm{trace}(X^T M X),$$

where $\mathcal{A}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, X \geq 0, \mathbf{1}_n \in \mathrm{span}(X)\}$ and $M = A - \frac{A \mathbf{1}_n \mathbf{1}_n^T A}{\mathbf{1}_n^T A \mathbf{1}_n}$

For idea graph, i.e., $A = ZZ^T$, it can be proven that the maximizer $\tilde{Z}$ of f is given by normalizing the columns of Z. Therefore, $\tilde{Z}$ defines the same communities.

The optimization problem is also in the form of

$$\min_{X \in \mathcal{A}_{n,k}} f(X) = -\tilde{f}(X),$$

where f is smooth.

Normalized cut:

$$\min_{Y^T D Y = I_q, Y \geq 0, \mathbf{1}_n \in \mathrm{span}(Y)} \mathrm{trace}(Y^T L Y),$$

where $L \in \mathbb{R}^{n \times n}$ is the Laplacian matrix of a graph and $D \in \mathbb{R}^{n \times n}$ is the diagonal matrix of the node degrees.

Normalized cut reformulation: (Let $D^{1/2} Y = x$)

$$\min_{X^T X = I_q, X \geq 0, v \in \mathrm{span}(X)} \mathrm{trace}(X^T D^{-1/2} L D^{-1/2} X),$$

where $v = \mathrm{diag}(D^{1/2}) > 0$.

Normalized cut:

$$\min_{Y^T D Y = I_q, Y \geq 0, \mathbf{1}_n \in \mathrm{span}(Y)} \mathrm{trace}(Y^T L Y),$$

where $L \in \mathbb{R}^{n \times n}$ is the Laplacian matrix of a graph and $D \in \mathbb{R}^{n \times n}$ is the diagonal matrix of the node degrees.

Normalized cut reformulation: (Let $D^{1/2} Y = x$)

$$\min_{X^T X = I_q, X \geq 0, v \in \mathrm{span}(X)} \mathrm{trace}(X^T D^{-1/2} L D^{-1/2} X),$$

where $v = \mathrm{diag}(D^{1/2}) > 0$.

Note that it is required here that $v \in \mathrm{span}(X)$ instead of $\mathbf{1}_n \in \mathrm{span}(X)$. We only discuss $\mathbf{1}_n \in \mathrm{span}(X)$ for simplicity. But the following derivations still work for $v \in \mathrm{span}(X)$ and $v > 0$.

Problem Statement

Reformulation of the optimization problem

k-means: $\min_{X \in \mathcal{A}_{n,k}} \|A - XX^T A\|_F^2$ com. det.: $\min_{X \in \mathcal{A}_{n,k}} -\mathrm{trace}(X^T M X)$

Expression:

$$\min_{X \in \mathcal{A}_{n,k}} f(X),$$

where $\mathcal{A}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, X \geq 0, \mathbf{1}_n \in \mathrm{span}(X)\}$

Variant:

$$\min_{X \in \mathcal{B}_{n,k}} f(X),$$

where $\mathcal{B}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, \|X\|_0 = n, \mathbf{1}_n \in \mathrm{span}(X)\}$

Problem Statement

Reformulation of the optimization problem

k-means: $\min_{X \in \mathcal{A}_{n,k}} \|A - XX^T A\|_F^2$ com. det.: $\min_{X \in \mathcal{A}_{n,k}} -\mathrm{trace}(X^T M X)$

Expression:

$$\min_{X \in \mathcal{A}_{n,k}} f(X),$$

where $\mathcal{A}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, X \geq 0, \mathbf{1}_n \in \mathrm{span}(X)\}$

Variant:

$$\min_{X \in \mathcal{B}_{n,k}} f(X),$$

where $\mathcal{B}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, \|X\|_0 = n, \mathbf{1}_n \in \mathrm{span}(X)\}$

Variant:

$$\min_{X \in \mathcal{F}_{n,k}} f(X) + \lambda \|X\|_1,$$

where $\mathcal{F}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, \mathbf{1}_n \in \mathrm{span}(X)\}$

k-means: $\min_{X \in \mathcal{A}_{n,k}} \|A - XX^T A\|_F^2$ com. det.: $\min_{X \in \mathcal{A}_{n,k}} -\mathrm{trace}(X^T M X)$

Expression:

$$\min_{X \in \mathcal{A}_{n,k}} f(X),$$

where $\mathcal{A}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, X \geq 0, \mathbf{1}_n \in \mathrm{span}(X)\}$

Variant:

$$\min_{X \in \mathcal{B}_{n,k}} f(X),$$

where $\mathcal{B}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, \|X\|_0 = n, \mathbf{1}_n \in \mathrm{span}(X)\}$

Variant:

$$\min_{X \in \mathcal{F}_{n,k}} f(X) + \lambda \|X\|_1,$$

where $\mathcal{F}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, \mathbf{1}_n \in \mathrm{span}(X)\}$

$$\min_{X \in \mathcal{F}_{n,k}} f(X) + \lambda \|X\|_1,$$

where $\mathcal{F}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, \mathbf{1}_n \in \mathrm{span}(X)\}$

Riemannian proximal gradient methods consider

$$\min_{x \in \mathcal{M}} F(x) = f(x) + g(x),$$

- $\mathcal{M}$ is a Riemannian manifold;
- f is continuously differentiable and may be nonconvex; and
- g is continuous, but may be not differentiable.

$$\min_{X \in \mathcal{F}_{n,k}} f(X) + \lambda \|X\|_1,$$

where $\mathcal{F}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, \mathbf{1}_n \in \mathrm{span}(X)\}$

Riemannian proximal gradient methods consider

$$\min_{x \in \mathcal{M}} F(x) = f(x) + g(x),$$

- Prove that $\mathcal{F}_{n,k}$ is a manifold
- Use a Riemannian proximal gradient method

Riemannian Manifold Structure of $\mathcal{F}_{n,q}$

Theorem

The set $\mathcal{F}_{n,q}$ is an embedded submanifold of $\mathrm{St}(q,n)$ with dimension $\dim(\mathrm{St}(q,n)) - (n-q) = nq - q(q+1)/2 - n + q$. Furthermore, $\mathcal{F}_{n,q}$ is also an embedded submanifold of $\mathbb{R}^{n \times q}$ with the same dimension and $\mathcal{F}_{n,q}$ is compact.

Verify [Bou20, Definition 8.70]

- Any $X \in \mathcal{F}_{n,q}$, find a function $h : \mathcal{U} \subseteq \mathrm{St}(q,n) \to \mathbb{R}^{n-q}$ such that
  - $h^{-1}(0) = \mathcal{F}_{n,q} \cap \mathcal{U}$
  - $\mathrm{rank}\, \mathrm{D}\, h(X) = n - q$
- h is constructed from the exponential mapping on $\mathrm{St}(q,n)$

Riemannian Manifold Structure of $\mathcal{F}_{n,q}$

- Riemannian metric: $\langle U, V \rangle = \mathrm{trace}(U^T V), \quad \forall U, V \in \mathbb{R}^{n \times q}$
- Tangent space:

$$\mathrm{T}_X \mathcal{F}_{n,q} = \{X\Omega + X_\perp K : \Omega^T = -\Omega, K \in \mathbb{R}^{(n-q)\times q}, KX^T \mathbf{1}_n = 0\}$$

and orthogonal projection is

$$P_{\mathrm{T}_X}(Z) = X \frac{X^T Z - Z^T X}{2} + (I - XX^T) Z (I - \hat{\alpha}\hat{\alpha}^T)$$

where $\hat{\alpha} = X^T \mathbf{1}_1 / \|X^T \mathbf{1}_n\|$

Riemannian Manifold Structure of $\mathcal{F}_{n,q}$

Retractions on $\mathcal{F}_{n,q}$ are given by

$$R_X(\eta_x) = \mathbf{1}_n q_*^T / \sqrt{n} + R_X^{\mathrm{St}}(\eta_x)(I - q_* q_*^T)$$

where $q_* = R_X^{\mathrm{St}}(\eta_x)^T \mathbf{1}_n / \|R_X^{\mathrm{St}}(\eta_x)^T \mathbf{1}_n\|$ and R_X^{St} is a retraction on the Stiefel manifold $\mathrm{St}(q, n)$.

- For any $X \in \mathrm{St}(q, n)$ with $X^T \mathbf{1}_n \neq 0$:

$$\mathbf{1}_n q_*^T / \sqrt{n} + X(I - q_* q_*^T) = \operatorname*{argmin}_{Y \in \mathcal{F}} \|X - Y\|^2 \qquad (1)$$

- Combine a retraction on $\mathrm{St}(q, n)$ with the orthogonal projection (1)
- If $X \notin \mathrm{St}(q, n)$, the closed form solution of (1) is unknown

Optimization with Structure: $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x), \tag{2}$$

1

Optimization with Structure: $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x), \tag{2}$$

A proximal gradient method[1]:

initial iterate: x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

Optimization with Structure: $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x), \tag{2}$$

A proximal gradient method[1]:

initial iterate: x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

Optimization with Structure: $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x), \tag{2}$$

A proximal gradient method[1]:

initial iterate: x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;
- L: greater than the Lipschitz constant of ∇f;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

Optimization with Structure: $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x), \qquad (2)$$

A proximal gradient method[1]:

initial iterate: x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;
- L: greater than the Lipschitz constant of ∇f;
- Proximal mapping: easy to compute;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

Optimization with Structure: $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x), \qquad (2)$$

A proximal gradient method[1]:

$$\text{initial iterate:} x_0,$$

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;
- L: greater than the Lipschitz constant of ∇f;
- Proximal mapping: easy to compute;
- Any limit point is a critical point;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

Optimization with Structure: $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x), \qquad (2)$$

A proximal gradient method[1]:

initial iterate: x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;
- L: greater than the Lipschitz constant of ∇f;
- Proximal mapping: easy to compute;
- Any limit point is a critical point;
- $O(1/k)$ sublinear convergence rate for convex f and g;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

Optimization with Structure: $\mathcal{M} = \mathbb{R}^n$

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x), \quad (2)$$

A proximal gradient method[1]:

initial iterate: x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;
- L: greater than the Lipschitz constant of ∇f;
- Proximal mapping: easy to compute;
- Any limit point is a critical point;
- $O(1/k)$ sublinear convergence rate for convex f and g;
- Local convergence rate by KL property;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n\times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2}\|p\|_F^2 + g(x_k + p)$$

A Riemannian proximal mapping [CMSZ20]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + g(x_k + \eta)$;

- Only works for a manifold with a linear ambient space;

[1][CMSZ18]: S. Chen, S. Ma, M. C. So, and T. Zhang, Proximal gradient method for nonsmooth optimization over the Stiefel manifold. SIAM Journal on Optimization, 30(1):210-239, 2020

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

A Riemannian proximal mapping [CMSZ20]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;

- Only works for a manifold with a linear ambient space;
- Proximal mapping is defined in tangent space;

[1][CMSZ18]: S. Chen, S. Ma, M. C. So, and T. Zhang, Proximal gradient method for nonsmooth optimization over the Stiefel manifold. SIAM Journal on Optimization, 30(1):210-239, 2020

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

A Riemannian proximal mapping [CMSZ20]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;

- Only works for a manifold with a linear ambient space;
- Proximal mapping is defined in tangent space;
- Convex programming;

[1] [CMSZ18]: S. Chen, S. Ma, M. C. So, and T. Zhang, Proximal gradient method for nonsmooth optimization over the Stiefel manifold. SIAM Journal on Optimization, 30(1):210-239, 2020

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

ManPG [CMSZ20]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta);$

- Only works for a manifold with a linear ambient space;
- Proximal mapping is defined in tangent space;
- Convex programming;
- Solved for the Stiefel manifold by a semi-Newton algorithm [XLWZ18b];

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

ManPG [CMSZ20]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;
2. $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$ with an appropriate step size α_k;

- Only works for a manifold with a linear ambient space;
- Proximal mapping is defined in tangent space;
- Convex programming;
- Solved for the Stiefel manifold by a semi-Newton algorithm [XLWZ18b];
- Convergence to a stationary point;

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

ManPG [CMSZ20]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;
2. $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$ with an appropriate step size α_k;

- Only works for a manifold with a linear ambient space;
- Proximal mapping is defined in tangent space;
- Convex programming;
- Solved for the Stiefel manifold by a semi-Newton algorithm [XLWZ18b];
- Convergence to a stationary point;
- No convergence rate results;

ManPG [CMSZ20]

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$$

RPG [HW21a]

1. $\eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad} f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta))$;
2. $x_{k+1} = R_{x_k}(\eta_k)$;

A Riemannian Proximal Gradient Method

A Riemannian Proximal Gradient Method in [HW21a]

ManPG [CMSZ20]

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$$

RPG [HW21a]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad} f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta))$;
2. $x_{k+1} = R_{x_k}(\eta_k)$;

- General framework for Riemannian optimization;

ManPG [CMSZ20]

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$$

RPG [HW21a]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad} f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta))$;
2. $x_{k+1} = R_{x_k}(\eta_k)$;

- General framework for Riemannian optimization;
- Any limit point is a critical point;

ManPG [CMSZ20]

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$$

RPG [HW21a]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad} f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta))$;
2. $x_{k+1} = R_{x_k}(\eta_k)$;

- General framework for Riemannian optimization;
- Any limit point is a critical point;
- $O(1/k)$ sublinear convergence rate for retraction-convex f and g;

ManPG [CMSZ20]

$$\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + g(x_k + \eta)$$

RPG [HW21a]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \mathrm{grad} f(x_k), \eta \rangle_{x_k} + \frac{L}{2}\|\eta\|_{x_k}^2 + g(R_{x_k}(\eta))$;
2. $x_{k+1} = R_{x_k}(\eta_k)$;

- General framework for Riemannian optimization;
- Any limit point is a critical point;
- $O(1/k)$ sublinear convergence rate for retraction-convex f and g;
- Local convergence rate by Riemannian KL property;

ManPG [CMSZ20]

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$$

RPG [HW21a]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad} f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta))$;
2. $x_{k+1} = R_{x_k}(\eta_k)$;

- General framework for Riemannian optimization;
- Any limit point is a critical point;
- $O(1/k)$ sublinear convergence rate for retraction-convex f and g;
- Local convergence rate by Riemannian KL property;
- Solving the proximal mapping by exploring the manifold structure or using the semi-smooth Newton iteratively;

Both ManPG and RPG require the Riemannian proximal mapping to be solved exactly

- Theoretically, but not practical numerically
- Can we relax this requirement and still preserve desired convergence properties?
- ManPG (yes)
- RPG [HW21b]

Both ManPG and RPG require the Riemannian proximal mapping to be solved exactly

- Theoretically, but not practical numerically
- Can we relax this requirement and still preserve desired convergence properties?
- ManPG (yes)
- RPG [HW21b]

The Riemannian proximal mapping in [CMSZ20] can be rewritten as

$$\arg \min_{B_x^T \eta = 0} \langle \xi_x, \eta \rangle + \frac{1}{2\mu} \|\eta\|_F^2 + g(x + \eta)$$

where $B_x^T \eta = (\langle b_1, \eta \rangle, \langle b_2, \eta \rangle, \ldots, \langle b_m, \eta \rangle)^T$, and $\{b_1, \ldots, b_m\}$ forms an orthonormal basis of $\mathrm{N}_x \mathcal{M}$.

The Lagrangian function:

$$\mathcal{L}(\eta, \Lambda) = \langle \xi_x, \eta \rangle + \frac{1}{2\mu} \langle \eta, \eta \rangle + g(X + \eta) - \langle \Lambda, B_x^T \eta \rangle.$$

Therefore

$$\text{KKT: } \left\{ \begin{array}{c} \partial_\eta \mathcal{L}(\eta, \Lambda) = 0 \\ B_x^T \eta = 0 \end{array} \right. \Longrightarrow \left\{ \begin{array}{c} \eta = \mathrm{Prox}_{\mu g} (x - \mu(\xi_x - B_x \Lambda)) - x \\ B_x^T \eta = 0 \end{array} \right.$$

where $\mathrm{Prox}_{\mu g}(z) = \mathrm{argmin}_{v \in \mathbb{R}^{n \times p}} \frac{1}{2} \|v - z\|_F^2 + \mu g(v)$.

Semi-smooth Newton method finds the Λ such that

$$\Psi(\Lambda) := B_x^T \left(\mathrm{Prox}_{\mu g} \left(x - \mu(\xi_x - B_x \Lambda) \right) - x \right) = 0$$
$$\eta_* = \mathrm{Prox}_{\mu g} \left(x - \mu(\xi_x - B_x \Lambda) \right) - x$$

- Ψ is not differentiable everywhere but semi-smooth;
- Semi-smooth Newton:
  1. $J_\Psi(\Lambda_k)[d] = -\Psi(\Lambda_k)$, where J_Ψ is the generalized Jacobian of Ψ;
  2. $\Lambda_{k+1} = \Lambda_k + d_k$
- Regularized semi-smooth Newton [XLWZ18a]

Semi-smooth Newton method finds the Λ such that

$$\Psi(\Lambda) := B_x^T (\mathrm{Prox}_{\mu g}(x - \mu(\xi_x - B_x \Lambda)) - x) \approx 0$$

- Ψ is not differentiable everywhere but semi-smooth;
- Semi-smooth Newton:
  1. $J_\Psi(\Lambda_k)[d] = -\Psi(\Lambda_k)$, where J_Ψ is the generalized Jacobian of Ψ;
  2. $\Lambda_{k+1} = \Lambda_k + d_k$
- Regularized semi-smooth Newton [XLWZ18a]
- Solving the equation inexactly

Solving the equation inexactly implies:

$$\Psi(\Lambda) = \epsilon \neq 0.$$

If $\Psi(\Lambda) = \epsilon$,

- $\eta_* = \mathrm{Prox}_{\mu g}\left(x - \mu(\xi_x - B_x\Lambda)\right) - x$ is not even in the tangent space $\mathrm{T}_x \mathcal{M}$ in this case
- Use $\hat{v}(\Lambda) = P_{\mathrm{T}_x \mathcal{M}}(\mathrm{Prox}_{\mu g}\left(x - \mu(\xi_x - B_x\Lambda)\right) - x)$ instead
- How small does ϵ need to be?

Solving the equation inexactly implies:

$$\Psi(\Lambda) = \epsilon \neq 0.$$

If $\Psi(\Lambda) = \epsilon$,

- $\eta_* = \mathrm{Prox}_{\mu g}\left(x - \mu(\xi_x - B_x\Lambda)\right) - x$ is not even in the tangent space $\mathrm{T}_x \mathcal{M}$ in this case
- Use $\hat{v}(\Lambda) = P_{\mathrm{T}_x \mathcal{M}}(\mathrm{Prox}_{\mu g}\left(x - \mu(\xi_x - B_x\Lambda)\right) - x)$ instead
- How small does ϵ need to be?

$$\|\epsilon\|_F \leq \sqrt{4\mu^2 L_g^2 + \|\hat{v}(\Lambda)\|_F^2 / 2} - 2\mu L_g,$$

Algorithm 1 ManPG without solving the subproblem exactly

Given x_0, $\nu \in (0,1)$, $\sigma \in (0, 1/(8\mu))$, $\mu > 0$;
for $k = 0, 1, \ldots$ **do**
    Approximately solve

$$\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad}\, f(x_k), \eta \rangle + \frac{1}{2\mu} \|\eta\|_F^2 + g(x_k + \eta)$$

    such that $\|\Psi_k(\Lambda)\|_F \leq \sqrt{4\mu^2 L_g^2 + \|\hat{v}_k(\Lambda)\|_F^2/2} - 2\mu L_g$;
    Set $\eta_k = \hat{v}_k(\Lambda)$ and set $\alpha = 1$;
    while $F(R_{x_k}(\alpha \eta_{x_k})) > F(x_k) - \sigma \alpha \|\eta_{x_k}\|_F^2$ **do**
        $\alpha = \nu \alpha$;
    end while
    $x_{k+1} = R_{x_k}(\alpha \eta_{x_k})$;
end for

Assumption

The function f is Lipschitz continuously differentiable on $\mathcal{M}$ and g is Lipschitz continuous on $\mathcal{M}$.

Theorem

Suppose the assumption holds. Then for any $\mu > 0$, there exists a constant $\bar{\alpha} \in (0, 1]$ such that for any $0 < \alpha < \bar{\alpha}$, the sequence $\{x_k\}$ generated by Algorithm 1 satisfies

$$F(R_{x_k}(\alpha \eta_{x_k})) - F(x_k) \leq -\frac{\alpha}{8\mu} \|\eta_{x_k}\|_F^2.$$

Moreover, the step size $\alpha > \rho\bar{\alpha}$ for all k.

Theorem

Suppose the assumption holds. Then any accumulation point of the sequence $\{x_k\}$ *generated by Algorithm 1 is a stationary point, i.e., if* x_* *is an accumulation point of the above sequence, then* $0 \in P_{\mathrm{T}_{x_*}\mathcal{M}} \partial F(x_*)$.

Theorem

Suppose the assumption holds. Then any accumulation point of the sequence $\{x_k\}$ generated by Algorithm 1 is a stationary point, i.e., if x_* is an accumulation point of the above sequence, then $0 \in P_{\mathrm{T}_{x_*}\mathcal{M}} \partial F(x_*)$.

Ideas in the proofs (Suppose $\Psi(\Lambda) = \epsilon \neq 0$)

- Consider the nearby optimization problem:

$$\arg \min_{B_x^T \eta = \epsilon} \langle \xi_x, \eta \rangle + \frac{1}{2\mu} \|\eta\|_F^2 + g(x + \eta)$$

- Its minimizer is given by $v(\Lambda) = \mathrm{Prox}_{\mu g}\left(x - \mu(\xi_x - B_x \Lambda)\right) - x$
- Show that $\hat{v}(\Lambda) = P_{\mathrm{T}_x \mathcal{M}} v(\Lambda)$ satisfies the same properties as η_*
- The vein of the remaining proofs follows [CMSZ20, HW21c]

$$\min_{X \in \mathcal{F}_{n,k}} -\mathrm{trace}(X^T M X) + \lambda \|X\|_1,$$

where $\mathcal{F}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, \mathbf{1}_n \in \mathrm{span}(X)\}$.

Comparing models and effectiveness

		μ_{LFR}								
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Lou.(k)	NMI	1.000	1.000	1.000	1.000	1.000	0.998	0.980	0.298	0.084
	AMI	1.000	1.000	1.000	1.000	1.000	0.997	0.965	0.238	0.039
	Mod.	0.949	0.849	0.750	0.650	0.549	0.449	0.347	0.209	0.196
	time	0.544	0.747	1.033	1.204	1.700	2.076	2.767	5.452	5.506
	k	20	20	20	20	20	20	19	12	12
New.(k)	NMI	0.998	0.683	0.678	0.667	0.549	0.391	0.280	0.134	0.049
	AMI	0.998	0.599	0.599	0.602	0.470	0.307	0.209	0.090	0.023
	Mod.	0.948	0.474	0.446	0.400	0.305	0.237	0.191	0.157	0.146
	time	0.645	0.466	0.437	0.452	0.423	0.341	0.365	0.321	0.311
	k	20	18	17	18	15	9	7	6	6
I-A.	NMI	1.000	1.000	1.000	1.000	1.000	0.999	0.960	0.451	0.129
	AMI	1.000	1.000	1.000	1.000	1.000	0.999	0.953	0.403	0.056
	Mod.	0.949	0.849	0.750	0.650	0.549	0.449	0.341	0.173	0.111
	time	0.635	0.469	0.587	0.949	0.674	0.472	1.033	1.630	1.675
	k	20	20	20	20	20	20	20	20	20

- Louvain method [BGLL08]
- Newman algorithm [New06]
- I-AManPG (With cceleration)

Comparing models and effectiveness

		μ_{LFR}								
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Lou.(k)	NMI	1.000	1.000	1.000	1.000	1.000	0.998	0.980	0.298	0.084
	AMI	1.000	1.000	1.000	1.000	1.000	0.997	0.965	0.238	0.039
	Mod.	0.949	0.849	0.750	0.650	0.549	0.449	0.347	0.209	0.196
	time	0.544	0.747	1.033	1.204	1.700	2.076	2.767	5.452	5.506
	k	20	20	20	20	20	20	19	12	12
New.(k)	NMI	0.998	0.683	0.678	0.667	0.549	0.391	0.280	0.134	0.049
	AMI	0.998	0.599	0.599	0.602	0.470	0.307	0.209	0.090	0.023
	Mod.	0.948	0.474	0.446	0.400	0.305	0.237	0.191	0.157	0.146
	time	0.645	0.466	0.437	0.452	0.423	0.341	0.365	0.321	0.311
	k	20	18	17	18	15	9	7	6	6
I-A.	NMI	1.000	1.000	1.000	1.000	1.000	0.999	0.960	0.451	0.129
	AMI	1.000	1.000	1.000	1.000	1.000	0.999	0.953	0.403	0.056
	Mod.	0.949	0.849	0.750	0.650	0.549	0.449	0.341	0.173	0.111
	time	0.635	0.469	0.587	0.949	0.674	0.472	1.033	1.630	1.675
	k	20	20	20	20	20	20	20	20	20

- The generalized LFR benchmark graphs [LF09]
- The larger μ is, the more difficult the community detection is
- An average of 10 random runs
- NMI: normalized mutual information [DDGDA05], AMI: adjusted mutual information [VEB10]

Comparing efficiency of ManPG with/without solving the subproblem exactly

K.	$q = 2$		$q = 3$		$q = 4$		$q = 5$	
Measurements	Exactly	Approx	Exactly	Approx	Exactly	Approx	Exactly	Approx
NMI	1	1	0.811	0.811	0.687	0.687	0.542	0.542
AMI	1	1	0.672	0.672	0.505	0.505	0.364	0.364
Mod.	0.372	0.372	0.373	0.373	0.420	0.420	0.382	0.382
time(s)	6.568	6.170	6.278	3.675	3.520	2.735	5.394	2.137

Less computational time, same effectiveness

$$\min_{X \in \mathcal{F}_{n,k}} -\mathrm{trace}(X^T D^{-1/2} W D^{-1/2} X) + \lambda \|X\|_1,$$

where W is the weight/affinity matrix,
$\mathcal{F}_{n,k} = \{X \in \mathbb{R}^{n \times k} : X^T X = I_k, v \in \mathrm{span}(X)\}$.

Numerical experiments

Normalized cut for image segmentation

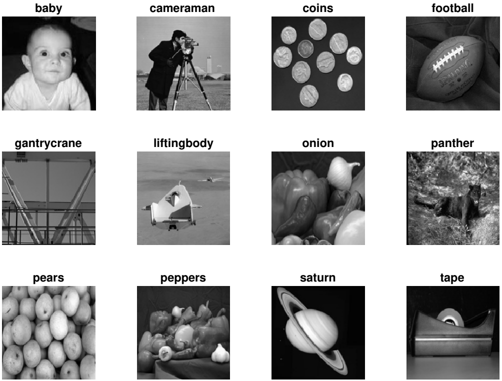
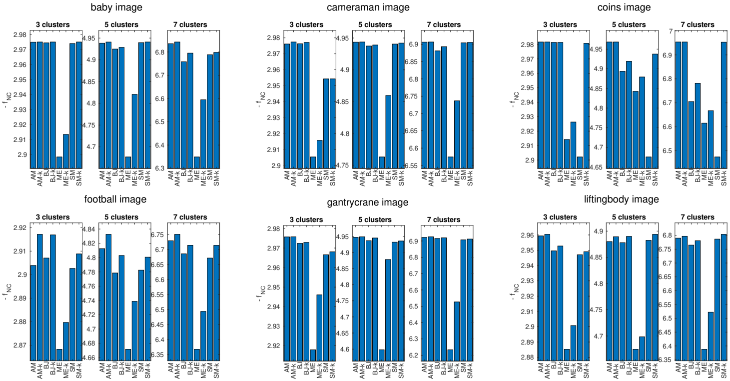
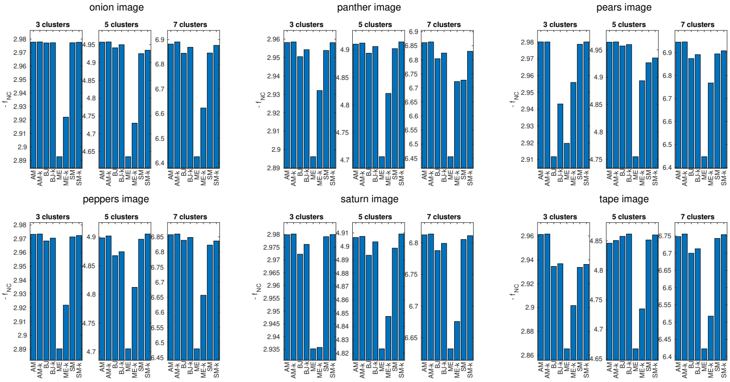


Figure: The tested images



Compare four methods and their combination with kernel k-means:

- Bach and Jordan [BJ03] (BJ), Shi and Malik [SM00] (SM), Karypis and Kumar [KK98] (ME), our method (AM)
- Their combination with kernel k-means, denoted by BJ-k, SM-k, ME-k, and AM-k respectively



Compare four methods and their combination with kernel *k*-means:

- Bach and Jordan [BJ03] (BJ), Shi and Malik [SM00] (SM), Karypis and Kumar [KK98] (ME), our method (AM)
- Their combination with kernel *k*-means, denoted by BJ-k, SM-k, ME-k, and AM-k respectively

Numerical experiments

Normalized cut for image segmentation

The segmentations by the Riemannian approach look more intuitive, especially for 7 clusters.

Summary

- Riemannian optimization problem statement
- Motivation
- Smooth optimization framework
- Literature review
- Clustering Problem

References I

Christos Boutsidis, Petros Drineas, and Michael W Mahoney.
Unsupervised feature selection for the k-means clustering problem.
In *Advances in Neural Information Processing Systems*, pages 153–161, 2009.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre.
Fast unfolding of communities in large networks.
Journal of statistical mechanics: theory and experiment, 2008(10):P10008, 2008.

Francis R. Bach and Michael I. Jordan.
Learning spectral clustering.
In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, page 305–312, Cambridge, MA, USA, 2003. MIT Press.

Nicolas Boumal.
An introduction to optimization on smooth manifolds.
Available online, Nov 2020.

Jianheng Chen and Wen Huang.
An iterative algorithm for low-rank tensor completion problem with sparse noise and missing values, 2023.

Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang.
Proximal gradient method for nonsmooth optimization over the Stiefel manifold.
SIAM Journal on Optimization, 30(1):210–239, 2020.

H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi, and R. Slama.
3D face recognition under expressions, occlusions, and pose variations.
Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(9):2270–2283, 2013.

Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas.
Comparing community structure identification.
Journal of Statistical Mechanics: Theory and Experiment, 2005(09):P09008, 2005.

Santo Fortunato.
Community detection in graphs.
Physics Reports, 486(3-5):75–174, 2010.

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, P.-A. Absil, and K. A. Gallivan.
Intrinsic representation of tangent vectors and vector transport on matrix manifolds.
Numerische Mathematik, 136(2):523–543, 2017.

Wen Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian BFGS method without differentiated retraction for nonconvex optimization problems.
SIAM Journal on Optimization, 28(1):470–495, 2018.

W. Huang, P.-A. Absil, K. A. Gallivan, and P. Hand.
ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds.
ACM Transactions on Mathematical Software, 4(44):43:1–43:21, 2018.

W. Huang and K. A. Gallivan.
A Limited-Memory Riemannian Symmetric Rank-One Trust-Region Method with a Restart Strategy.
Journal of Scientific Computing, 93(1), 2022.

W. Huang, K. A. Gallivan, and P.-A. Absil.
A Broyden Class of Quasi-Newton Methods for Riemannian Optimization.
SIAM Journal on Optimization, 25(3):1660–1685, 2015.

W. Huang, K. A. Gallivan, Anuj Srivastava, and P.-A. Absil.
Riemannian optimization for registration of curves in elastic shape analysis.
Journal of Mathematical Imaging and Vision, 54(3):320–343, 2015.
DOI:10.1007/s10851-015-0606-8.

References III

Wen Huang, K. A. Gallivan, and Xiangxiong Zhang.
Solving phaselift by low rank Riemannian optimization methods.
In *Proceedings of the International Conference on Computational Science (ICCS2016), accepted*, 2016.

Wen Huang and Paul Hand.
Blind deconvolution by a steepest descent algorithm on a quotient manifold.
SIAM Journal on Imaging Sciences, 11(4):2757–2785, 2018.

S. Hosseini, W. Huang, and R. Yousefpour.
Line search algorithms for locally Lipschitz functions on Riemannian manifolds.
SIAM Journal on Optimization, 28(1):596–619, 2018.

W. Huang and K. Wei.
Riemannian proximal gradient methods.
Mathematical Programming, 2021.
published online, DOI:10.1007/s10107-021-01632-3.

Wen Huang and Ke Wei.
An Inexact Riemannian Proximal Gradient Method, 2021.

Wen Huang and Ke Wei.
An extension of fast iterative shrinkage-thresholding algorithm to Riemannian optimization for sparse principal component analysis.
Numerical Linear Algebra with Applications, page e2409, 2021.

George Karypis and Vipin Kumar.
A fast and high quality multilevel scheme for partitioning irregular graphs.
SIAM Journal on Scientific Computing, 20(1):359–392, 1998.

Andrea Lancichinetti and Santo Fortunato.
Community detection algorithms: a comparative analysis.
Physical Review E, 80(5):056117, 2009.

References IV

H. Laga, S. Kurtek, A. Srivastava, M. Golzarian, and S. J. Miklavcic.
A Riemannian elastic metric for shape-based plant leaf classification.
2012 International Conference on Digital Image Computing Techniques and Applications (DICTA), pages 1–7, December 2012. doi:10.1109/DICTA.2012.6411702.

Mark EJ Newman.
Fast algorithm for detecting community structure in networks.
Physical review E, 69(6):066133, 2004.

Mark EJ Newman.
Modularity and community structure in networks.
Proceedings of the national academy of sciences, 103(23):8577–8582, 2006.

Mark EJ Newman and Elizabeth A Leicht.
Mixture models and exploratory analysis in networks.
Proceedings of the National Academy of Sciences, 104(23):9564–9569, 2007.

Qiangwei Peng and Wen Huang.
An image inpainting algorithm using exemplar matching and low-rank sparse prior, 2023.

Martin Rosvall and Carl T Bergstrom.
Maps of random walks on complex networks reveal community structure.
Proceedings of the National Academy of Sciences, 105(4):1118–1123, 2008.

Wutao Si, P. A. Absil, Wen Huang, Rujun Jiang, and Simon Vary.
A riemannian proximal newton method, 2023.

Jianbo Shi and J. Malik.
Normalized cuts and image segmentation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000.

References V

Nguyen Xuan Vinh, Julien Epps, and James Bailey.
Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance.
The Journal of Machine Learning Research, 11:2837–2854, 2010.

M. Wei, W. Huang, K. A. Gallivan, and P. Van Dooren.
Community Detection by a Riemannian Projected Proximal Gradient Method.
In *Proceedings of the 24th Internaltional Symposium on Mathematical Theory of Networks and Systems*, 2021.
accepted.

X. Xiao, Y. Li, Z. Wen, and L. Zhang.
A regularized semi-smooth Newton method with projection steps for composite convex programs.
Journal of Scientific Computing, 76(1):364–389, Jul 2018.

Xiantao Xiao, Yongfeng Li, Zaiwen Wen, and Liwei Zhang.
A regularized semi-smooth newton method with projection steps for composite convex programs.
Journal of Scientific Computing, 76(1):364–389, Jul 2018.

Liang Yang, Xiaochun Cao, Dongxiao He, Chuan Wang, Xiao Wang, and Weixiong Zhang.
Modularity based community detection with deep learning.
In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, volume 16, pages 2252–2258, 2016.

Thank you

Thank you!