Optimization Problem with Orthogonality Constraints

Haotao Wang

Sep 9, 2017

1 Problem Introduction

1.1 Numeric Optimization Problem

For a traditional unconstrained optimization problem

$$\min_{\mathbf{X}\in\mathbb{R}^{m*n}} F(\mathbf{X}) \tag{1}$$

there are basically three most popular numeric methods to find a minimal, steepest descend, Newton method and Quasi-Newton method. The update schemes in those methods can be formulated in the same way as

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \delta^{(k)}\mathbf{D}^{(k)}, \tag{2}$$

where $\mathbf{X}^{(k)}$, δ and $\mathbf{D}^{(k)}$ are correspondingly the value of $\mathbf{X}$, the step size and the direction of update on the kth step. The three methods different in their ways to determine the update directions, but they share the same algorithms to find out the step size. There are tons of different self adaptive ways to determine the step size at each step, but they mainly falls into two buckets, monotone and non-monotone. In the traditional unconstrained optimization problem, the determination of the update direction and the determination of the step size are totally uncorrelated, which means most step size algorithms can be used in all three methods mentioned above.

1.1.1 Update directions

The update directions of the three methods are shown as follows:
1. Steepest Descend Method

$$\mathbf{D}^{(k)} = -\nabla F(\mathbf{X})|_{\mathbf{X}=\mathbf{X}^{(k)}}, \tag{3}$$

2. Newton Method

$$\mathbf{D}^{(k)} = -\mathbf{H}^{-1}(\mathbf{X})\nabla F(\mathbf{X})|_{\mathbf{X}=\mathbf{X}^{(k)}}, \tag{4}$$

where $\mathbf{H}(\mathbf{X})$ is the Hessian Matrix of the objective function. Compared with steepest descend method, Newton method converge faster, since it makes uses of the second order derivative information of the objective function. While the draw back is the calculation of Hessian matrix is too costly and sometimes error-prone.
3. Quasi-Newton
Find a way to approximate Hessian matrix with another low calculation cost matrix. See [1] formula (2.17) and (2.18) for more details.

1.1.2 Step size

The algorithms to determine the step size are summarized as follows:
1. Monotone line search
The main idea of this kind of methods is to search for a step size that satisfies certain conditions to guaranty the value of the energy function decreases after each iteration. Two most popular conditions are the Goldstein condition([1] formula (3.11)) and the (strong) Wolfe condition([1] formula (3.6) and (3.7)). All these monotone line search conditions contain a same part called Armijo condition [2], or the sufficient decrease condition, which guarantees the decrease of energy value at each iteration. The other parts of the conditions aim to make the chosen step size more efficient, such as punishing it from being too small or too big.
2. Nonmonotone line search
The most popular nonmonotone line search method is called BB(Barzilai-Borwein) method [3]. The Wolfe condition has different nonmonotone variations [4]. Nonmonotone methods allow the energy value to increase at some steps, but can guarantee a global convergence if the energy function satisfies some certain conditions [4].

1.2 Sphere Constrains

Optimization problems with sphere constrains (e.g. $\mathbf{x}^T\mathbf{x} = 1$), or more generally speaking orthogonal constrains (e.g. $\mathbf{X}^T\mathbf{X} = \mathbf{I}$), have wide applications in polynomial optimization, combinatorial optimization, eigenvalue problems, sparse PCA, p-harmonic flows, 1-bit compressive sensing, matrix rank minimization, etc [5]. [5] proposed a novel method to solve the optimization problem under orthogonal constrains, and the following of this article aims to give a more detailed explanation to that method.

2 A feasible method for optimization with orthogonality constraints

$$\min_{\mathbf{X}\in\mathbb{R}^{m*n}} F(\mathbf{X}) s.t. \mathbf{X}^T\mathbf{X} = \mathbf{I} \tag{5}$$

define

$$G := \frac{dF(X)}{dX} \tag{6}$$

$$A := GX^T - XG^T \tag{7}$$

Lemma A is screw symmetric
proof

$$A^T = XG^T - GX^T = -A$$

define update schemes:

$$Y(\tau) = (I + \frac{\tau}{2}A)^{-1}(I - \frac{\tau}{2}A)X \tag{8}$$

where X is the current step, $Y(\tau)$ is the next step, τ is the step size, which can be calculated by the monotone or nonmonotone step size algorithms. Note that equation (8) is equavalent to the following form:

$$Y(\tau) = X - \frac{\tau}{2}AX - \frac{\tau}{2}AY(\tau) \tag{9}$$

Lemma Y is well defined and still satisfy the constrain, i.e. $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}$
proof
for $\forall \mathbf{x} \neq \mathbf{0}$

$$(\mathbf{x}^T\mathbf{A}\mathbf{x})^T = \mathbf{x}^T\mathbf{A}^T\mathbf{x} = -\mathbf{x}^T\mathbf{A}\mathbf{x}$$

so this is a n=-n equation, which directly gives that n=0, i.e. $\mathbf{x}^T\mathbf{A}\mathbf{x} = 0$. Consider

$$\mathbf{x}^T(\mathbf{I} + \mathbf{A})\mathbf{x} = \mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{x}^T\mathbf{I}\mathbf{x} = 0 + \mathbf{x}^T\mathbf{x} \neq 0$$

so we have $\mathbf{I} + \mathbf{A}$ is always invertible. Multiplying matrix A by a constant doesn't change the screw symmetric property, so we know that $\mathbf{I} + \frac{\tau}{2}\mathbf{A}$ is invertible for any τ and thus $Y(\tau)$ is well defined. Next we prove that $Y(\tau)$ still satisfy the constrain. Consider

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

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

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

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

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

The third and fifth equations are using $(AB)^{-1} = B^{-1}A^{-1}$ if square matrix A, B are invertible and are of the same size. The forth equation is using the fact that $(I+B)(I-B) = I - B*B = (I-B)(I+B)$.

We consider τ as the step size at X. Fixing X, we can take $Y(\tau)$ as a parameterized curve function, where τ is the curve parameter. We can use the traditional line search methods on this curve line to find a proper step size, i.e. the proper value of τ, then we can get the value of $Y(\tau)$. Until now, we have defined the overall update scheme in this curvilinear search method, which guarantees every step always satisfies the orthogonal constrain: with the old step X and a step size τ, we can get the next step $Y(\tau)$.

But this is far from finished, since we still need to prove this curve $Y(\tau)$ is a descent curve, i.e. the energy will decrease along this curve.

Lemma $Y(\tau)$ is a descent curve, i.e. $\frac{\partial F(Y(\tau))}{\partial \tau}|_{\tau=0} \leq 0$

proof

To prove this lemma, first we need to know $\frac{dY(\tau)}{d\tau}$. Do $\frac{d}{d\tau}$ on both sides of equation (9),

$$\frac{dY(\tau)}{d\tau} = -\frac{AX}{2} - \frac{A}{2}Y(\tau) - \frac{\tau}{2}A\frac{dY(\tau)}{d\tau}$$

so

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

Using the chain rule, we have

$$\frac{\partial F(Y(\tau))}{\partial \tau} = tr(\frac{dF(X)}{dX}^T\frac{dY(\tau)}{d\tau})$$

At τ=0, we have $Y(0) = X$, $\frac{dY(\tau)}{d\tau} = -AX$, and $\frac{dF(X)}{dX} = G(X)$ as defined in (6). So we have

$$\frac{\partial F(Y(\tau))}{\partial \tau}|_{\tau=0} = -tr(G^TAX) = -tr(G^T(GX^T - XG^T)X)$$
$$= -\frac{1}{2}tr(A^TA) = -\frac{1}{2}||A||_F^2 \leq 0$$

The second to last equation is not so straight forward. So I'll explain it a little bit more.

$$A'A = (XG' - GX')(GX' - XG') = XG'GX' + GG' - GX'GX' - XG'XG'$$

so

$$tr(A'A) = tr(XG'GX') + tr(GG') - tr(GX'GX') - tr(XG'XG')$$

$$= tr(X'XG'G) + tr(GG') - tr(X'GX'G) - tr(G'XG'X)$$
$$= tr(G'G) + tr(GG') - tr(X'GX'G) - tr(G'XG'X)$$

The second equation is simply using the fact that $tr(AB) = tr(BA)$.
Since $tr(G'G) = tr(GG')$ and $tr(X'GX'G) = tr(G'XG'X)$, which's because $tr(MM) = tr((MM)') = tr(M'M')$, where $M := G'X$, we have

$$tr(A'A) = 2(tr(G'G) - tr(G'XG'X)) = 2tr(G'(GX' - XG')X)$$

Now we have finished explaining the basic parts of this curvilinear search algorithm.

Calculation complexity and low rank form of matrix A

In the practical problems, the data we got $\mathbf{X} \in \mathbb{R}^{m*n}$ usually have high dimensions. Say we are training on 4k SIFT/GIST features, than we will have m=128 and n=4000. Thus A will be a 128*128 matrix. Finding $Y(\tau)$ requires doing inverse on such a huge matrix is time consuming, so we want to find a way to reduce the complexity to find $Y(\tau)$.
Here is a way to do low rank decomposition on A:

$$\hat{A} = G_{(q)}X_{(q)}^T - X_{(q)}G_{(q)}^T \tag{10}$$

where q is a index:

$$q := argmax_{i=1,\dots,n}G_{(i)}X_{(i)}^T - X_{(i)}G_{(i)}^T \tag{11}$$

We can easily prove that the $Y(\tau)$ we got using this $\hat{A}$ instead of A is still a descend curve, i.e. $\frac{\partial F(Y(\tau))}{\partial \tau}|_{\tau=0} \leq -\frac{1}{2n}||A||_F^2 \leq 0$
After doing this low rank decomposition, we get a rank-2 matrix $\hat{A}$, we can then use equation (20) in [5] to calculate $Y(\tau)$, which is much more time saving.
While in the geometry problem, such as finding the conformal mapping, the inputs are simply coordinates $\mathbf{x} \in \mathbb{R}^3$ which is restricted to the sphere constrains $||\mathbf{x}||^2 = 1$, so we needn't do the low rank decomposition, since it is rank one vector itself.

Extension to multiple orthogonal constrains and the geometry view

As we have discussed before, in geometry problems, the input is a bunch of vertex (coordinates). If written into a matrix form, it should be $\mathbf{X} \in \mathbb{R}^{3*n}$. We need to notice that this matrix X doesn't satisfy the orthogonal constrain, since different vertex's coordinate's inner product is not 0, i.e. the off-diagonal elements of X^TX are not 0. This is actually the multi-constrain problem in [5]:

$$\min_{\mathbf{x_1}\in\mathbb{R}^3,...,\mathbf{x_n}\in\mathbb{R}^3} \quad F(\mathbf{x_1},...,\mathbf{x_n}) \quad s.t. \quad \mathbf{x_i}^T\mathbf{x_i} = 1 \quad for \quad i \in 1,..,n \tag{12}$$

Now we need to generalize the algorithm to multiple constrains. Since the $\mathbf{x}_i$ are totally uncorrelated with each other, we can define three updating directions on their own:

$$A_i = G_iX_i^T - X_iG_i^T \quad for \quad i \in 1,..,n \tag{13}$$

If we write it into matrix format,

$$G := [G_1, ...G_n]$$

$$X := [X_1, ...X_n]$$

we will get:

$$GX^T - XG^T = \Sigma_{i=1}^n G_iX_i^T - X_iG_i^T = \Sigma_{i=1}^n A_i \tag{14}$$

We still define matrix A as:

$$A := GX^T - XG^T$$

then we have $A = \Sigma_{i=1}^n A_i$. Since A_i are all screw symmetric matrix and $||A_i||_F^2 >= 0$, we can easily prove that A is also screw symmetric and that

$$||A||_F^2 = \Sigma||A_i||_F^2 >= 0 \tag{15}$$

which guarantees the $Y(\tau)$ we got from equation (8) is still a descend curve under the multiple orthogonal or spherical constrains.

3 Application to Finding the Conformal Mapping

3.1 Introduction to conformal mapping

Conformal mapping between two subjects maintains the shape. That means after the deformation of a surface, which is represented by a set of vertex and faces, the angles of the triangle faces are unspoiled. A most simple usage of conformal mapping is to draw the maps.

ARCTIC OCEAN
Greenland
U.S.A.
CANADA
UNITED STATES OF AMERICA
RUSSIAN FEDERATION
KAZAKHSTAN
MONGOLIA
CHINA
ATLANTIC OCEAN
PACIFIC OCEAN
BRAZIL
INDIAN OCEAN
AUSTRALIA
SOUTHERN OCEAN
ANTARCTICA
Political Map of the World
Your logo here

We can see that the longitudes and latitudes are still orthogonal on the 2D rectangle as they actually are on the real sphere(the earth surface). That means the angles are preserved during the shape deformation from a sphere to a 2D rectangle. However, the area is not preserved. For example, the Russia looks even larger than the whole Africa, which is not true. The goal of angle preserving is motivated by early navigation. When traveling on the sea, it is more important to know the directions than to know the distance. So it is more wise to make maps that preserve the angle (direction) while not the area (distance), considering a mapping that can keep both unchanged does not exist.

3.2 The harmonic energy

To find the conformal mapping, we need to minimize the harmonic energy. It has been proven that if the harmonic energy is 0, then the mapping is strictly conformal. While in practical problems, it is difficult to find a mapping decreasing the energy to 0. So people just use numeric methods to decrease the energy as low as much as possible, finding an acceptable local minimal. That kind of mapping is called quasi-conformal mapping, since it is not strictly conformal but near conformal. But even finding a quasi-conformal mapping is not easy, since the optimization problem is highly non-linear and the spherical constraint is hard to maintain during the optimization. So this algorithm provides a good way to keep the constraints during optimization, thus achieving better results. The harmonic energy looks like this:

$$E(F) = tr(F\Delta_M F^T) \quad s.t. \quad ||F_i||^2 = 1 \quad for \quad i \in 1,...,n \tag{16}$$

where Δ_M is called the discrete Laplace operator, which is actually a $n \times n$ matrix containing the information of the original shape. $F \in \mathbb{R}^{3n}$ is the vertex coordinate matrix.

4 Discussion

4.1 Compare with traditional constrained optimization methods

Traditional methods to solve constrained optimization problems falls mainly into two buckets:
1. Lagrange multiple method/ Augment Lagrange method

$$L(x, \lambda) = L_0(x) + \lambda c(x) \tag{17}$$

$$\nabla_x L(x) = \nabla L_0(x) + \lambda \nabla c(x) \tag{18}$$

$$\nabla_\lambda L(x) = c(x) \tag{19}$$

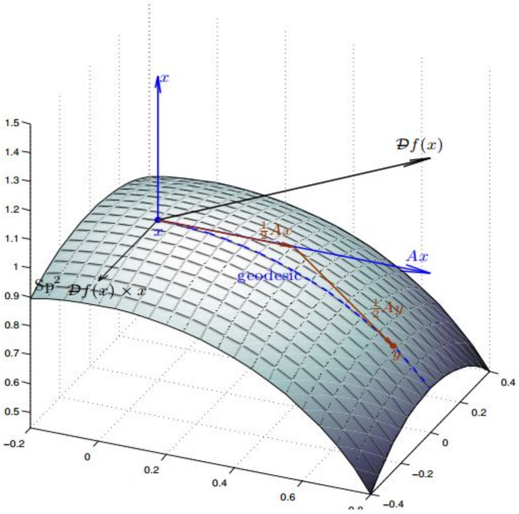
These two methods update the parameter X together with the Lagrange multipier λ. According to the theory, the constraint is satisfied only when $\frac{\partial}{\partial \lambda} L(X, \lambda) = 0$. However, this equation is never really satisfied since we are using a numeric method. It is just going in a direction that tend to satisfy the constrain.
2. Punishment method
In this method, the constrain is taken as a punishment term in the loss function, where λ is taken as a prefixed constant.

$$L(x) = L_0(x) + \lambda c(x) \tag{20}$$

$$\nabla L(x) = \nabla L_0(x) + \lambda \nabla c(x) \tag{21}$$

4.2 What's matrix A

Fig. 1 An illustration using $\mathrm{Sp}^2 := \{x \in \mathbb{R}^3 : \|x\|_2 = 1\}$. The point $x + \tau A x$ is not feasible. The curve $y(\tau)$ satisfying $y = x + \frac{\tau}{2} A(x + y)$ is feasible and a geodesic.



Lemma
$A(X)X$ is in the tangent surface of the sphere.

proof

$$< X, A(X)X >= tr(X'A(X)X) = tr(X'(GX' - XG')X)$$

$$= tr(X'G - GX') = tr(X'G) - tr(GX') = 0$$

where $< \cdot, \cdot >$ means the inner product. What's more, consider the first order condition of Lagrange problem:

$$\nabla_X L(X, \lambda) = \nabla_X F(X) - X\lambda^T = 0 \tag{22}$$

$$\nabla_\lambda L(X, \lambda) = X^T X - I = 0 \tag{23}$$

Multiply X_T on both sides of equation (22), we get

$$\lambda = G'X \tag{24}$$

substitute (24) into (22), we get

$$\nabla_X L(X, \lambda) = (GX' - XG')X = AX$$

Reference

[1] J. Nocedal and S. J. Wright, Numerical Optimization, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.

[2] Armijo, Larry, Minimization of functions having Lipschitz continuous first partial derivatives, Pacific J. Math, 1966.

[3] J. Barzilai and J. M. Borwein, Two-point step size gradient methods, IMA J. Numer. Anal., 8 (1988), pp. 141-148.

[4] H. Zhang and W. W. Hagar, A Nonmonotone Line Search Technique and its Application to Unconstrained Optimization, SIAM, 2004

[5] Wen Z., Yin W.: A feasible method for optimization with orthogonality constraints, Math. Program, (2013)

[6] R. Lai, et al.: Folding-Free Global Conformal Mapping for Genus-0 Surfaces by Harmonic Energy Minimization, J Sci Comput, (2014)