Non-Convex Matrix Factorization for Semidefinite Programming: Geometry and Optimization

CSE 597: Large-Scale Machine Learning

Rohan Shah

Abstract

To understand how non-convex reformulation of rank constrained convex positive semi-definite (PSD) matrix can be efficiently solved using matrix factorization as a proxy and investigate the non-linear programming algorithm to solve SDP which is practically efficient (i.e. less memory and time intensive). Secondly, to understand the non-convex geometry of low-rank matrix optimization and how converging to any local minima will approximately yield the global minimum or a strict saddle point (where the hessian matrix has a strictly negative eigenvalue).

1 Introduction

1.1 Project Statement

To understand the applications of matrix factorization to solve semidefinite programming (SDP) problems by analyzing its geometry and optimization.

1.2 Motivation

Combinatorial optimization problems like max-cut or minimum spanning tree problems are known to be NP-hard, but the application of SDP helps in obtaining bounds towards its solution. On the other hand, the theoretically efficient algorithms for solving SDP are actually time and memory-intensive in practice, especially while dealing with large-scale combinatorial problems. So, I will take advantage of the idea that non-convex reformulations of convex optimization problems involve fewer variables compare to their convex formulations, thus, allowing for scalability and efficiency. Therefore, this paper will analyze the parameterization technique from Burer's paper [1] in optimizing the non-convex SDP and discuss the optimization and geometrical structures of the non-convex reformulation. Geometrical structures of convex problems are simple, since converging to a local minima will converge to the global minima, but the same cannot be inferred for non-convex reformulation. So, to understand and analyze the landscape better, this paper will then focus on Li's paper [2]. It turns out that for a range of convex optimization problems (matrix completion and sensing problems), there corresponding non-convex reformulations have nice geometric structures that converging to a local minima guarantees either its a global minima or a strict saddle point.

It should also be noted that the famous Netflix prize problem was solved using number of SVD models, where the researchers were able to achieve around 10 percent increase in accuracy over Netflix's recommending algorithm, and since then, SVD and in-turn Matrix Factorization, have become very popular in the field of collaborative filtering; where collaborative filtering is one of the techniques in recommender systems.

1.3 Outline of the paper

Section 2.1 of this paper discuses the non-convex reformulation of SDP, then focuses on developing non-linear programming algorithm for practical use, by exploiting sparsity. Next, section 2.2 studies the geometrical structures of non-convex reformulations of SDP, by answering the key question: given the non-convex problem, is it possible to converge to global minimum? It should be noted that, understanding of these structures is an active research area.

2 Discussion

2.1 Optimization

Burer's [1] paper discusses the optimization of SDP using low-rank factorization by replacing the positive semidefinite variable X with RR^T in the SDP. The goal of this section is to develop practically efficient algorithm by exploiting sparsity in large-scale SDPs and relying on gradient based information only (first order). The standard-form primal SDP is as follows:

$$\min\{C \cdot X \colon A_i \cdot X = b_i, i = 1, ..., m, X \succeq 0\} \tag{1}$$

where data matrices C and A_i are nxn real symmetric matrices, the data vector b is m-dimensional.

Replacing X with RR^T:

$$\min\{C \cdot RR^T \colon A_i \cdot RR^T = b_i, i = 1, ..., m\} \tag{2}$$

where matrix R is nxn real matrix. It can be observed that since X is replaced with RR^T, computing this algorithm will now be cheaper since computing positive semidefinite X is difficult and (2) has eliminated X. Although, the objective function and constraints are now quadratic (nonconvex) in nature, i.e. they are nonlinear and therefore, the replacement of positive semidefinite variable X with factor RR^T now bears the cost. But, the algorithm was developed with the central idea of better practical performance compare to theoretical guarantees of SDP's. So, in order to compare the practical performances of (1) and (2), the paper addresses following important points: efficiently managing n^2 variables in R, optimization method that can exploit sparsity in the problem data, and finding a global solution for a nonconvex programming problem, since the problem is now nonconvex in nature.

As mentioned before, the crux of the story is to exploit sparsity in the data while optimizing nonlinear reformulation of the standard SDP (2). The non-linear programming method chosen to optimize this problem, say N_r (non-linear program) is Lagrangian method. The reason for choosing Lagrangian is because Lagrangian method is good with optimizing constrained non-convex problem. Lagrangian function is defined as:

$$\mathcal{L}(R, y) = C \cdot RR^T - \sum_{n=1}^{m} y_i(A_i \cdot RR^T - b_i), \tag{3}$$

where y is m-dimensional vector of unrestricted Lagrange multipliers for the equality constraints of N_r (i.e. of rank r).

Another reason for choosing this non-linear programming method is because it ignores the constraints (introduced by replacing $X \succeq 0$ with $X = A_i \cdot RR^T$) all together. So, it works on the idea of penalization and since penalization is itself not enough, augmented Lagrangian function is adapted in this paper, which introduces Lagrangian multipliers yi for each constraint. Augmented Lagrangian function used in this paper is as follows:

$$\mathcal{L}(R, y) = C \cdot RR^T - \sum_{i=1}^{m} y_i(A_i \cdot RR^T - b_i) + \frac{\sigma}{2}\sum_{i=1}^{m} y_i(A_i \cdot RR^T - b_i)^2, \tag{4}$$

Adapted Lagrangian function has an additional last term which computes euclidean norm of the infeasibility of R with respect to N_r and its scaled by penalty parameter σ. So, the algorithm to converge to a local minima is as follows:

```
i. compute $v = \sum_{n=1}^{m}(A_i \cdot (R^k \cdot (R^k))^T - b_i)^2$
ii. if $v < \eta v_k$,
    $y_i^{k+1} = y_i^k - \sigma_k(A_i \cdot (R^k \cdot (R^k))^T - b_i)$ for all i
    $\sigma_{k+1} = \sigma_k$
    $v_{k+1} = v$
iii. else
    $y_i^{k+1} = y_i^k$ for all i
    $\sigma_{k+1} = \sigma_k$
    $v_{k+1} = \gamma v_k$
```

where, v_k is best infeasibility obtained for some R. So, in a nutshell, the algorithm is trying to obtain best infeasibility and therefore, if $v_k < \eta v_k$, R_k has found a better infeasibility compare to previous iterations. Also, one achievement in this algorithm is its time-complexity is O(nr) compare to $O(n^2)$, where rank(R)=r (R: factored matrix) and $r << n$.

2.2 Geometry of non-convex reformulation of convex optimization problems

In this section, the goal is to understand the geometry of non-convex reformulation of PSD described in (2), by answering the question: given non-convex reformulation of PSD (2), is it possible to converge to the global optimum?

Definition 1 (Condition Number) It measures how much the output value of the function can change for a small change in the input argument. A problem is considered to be well-conditioned if its condition number is low.
So, for (2) to converge to a global minimum, the following equation must be satisfied (i.e. the function must be restricted well-conditioned):

$$\alpha||D||_F^2 \leq [\bigtriangledown^2 f(X)](D,D) \leq \beta||D||_F^2 \text{ with} \frac{\beta}{\alpha} \leq 1.5 \tag{5}$$

whenever $rank(X) \leq 2r$ and $rank(D) \leq 4r$. Also, $[\bigtriangledown^2 f(X)](D,D)$ is a directional curvature (or bilinear form of hessian), which is computed as follow:

$$[\bigtriangledown^2 f(X)](G,H) = \sum_{i,j,k,l} \frac{\bigtriangledown^2 f(X)}{\bigtriangledown Z_{ij} \delta Z_{kl}} G_{ij} H_{kl} \tag{6}$$

for any $G, H \in R^{mxn}$.

Definition 2 (RIP) A linear operator $A : R^{nxn} \rightarrow R^m$ satisfies the r-RIP with constant δ_r if

$$(1-\delta_r)||D||_F^2 \leq (||A(D)||_2^2) \leq ((1+\delta_r)||D||_F^2) \tag{7}$$

Above equation infers that the condition number of Hessian matrix $\bigtriangledown^2[f(X)](D,D)$ should be small at least in the directions of the low rank matrices D, since the directional curvature form of f(X) is $\bigtriangledown^2[f(X)](D,D) = ||A(D)||_F^2$
It should be noted that (6) and (7) are similar, since both measure the condition number and condition number is measured in different forms depending on the problem formulation. For instance, to compute condition number for weighted PCA and matrix sensing problems, (6) and (7) can be used, respectively.

Definition 5 (Critical point): A point x is a critical point of a function, if the gradient of this function vanishes at x.

Definition 6 (Strict saddles or ridable saddles): For a twice differentiable function, a strict saddle is one of its critical points whose Hessian matrix has at least one strictly negative eigenvalue. So, a twice differentiable function satisfies strict saddle property if each critical point either corresponds to the local minima or is a strict saddle.

Theorem 2.1: Suppose the objective function of (2) is twice continuously differentiable and is restricted well-conditioned assumption in (5). Assume X^* is

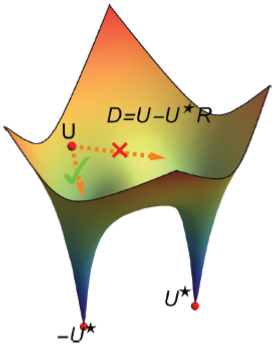

Figure 1: The matrix $D = U - U^*R$

an optimal solution of (2) with $\text{rank}(X^*) = r^*$. Set $r \geq r^*$ for the factored variables U and V . Then any critical point U either corresponds to the global optimum X^* such that $X^* = UU^T$ for (7) or is a strict saddle point (which includes a local maximum) of g.
Proof:
The goal is to prove each critical point U of g(U) either corresponds to the optimal solution X^* or its Hessian matrix $\bigtriangledown^2 g(U)$ has at least one strictly negative eigenvalue (a strict saddle point), i.e. to show $[\bigtriangledown^2 g(U)] = -\tau||D||_F^2$, for some $\tau > 0$.

For any U, f(U) = f(UR) where $R \in O_r$ (where O_r represents orthogonal matrix of dimension rxr), consider the distance between two points U1 and U2:

$$d(U_1, U_2) = \min_{R_1 \in O_r, R_2 \in O_r} = ||U_1R_1 - U_2R_2||_F = \min_{R \in O_r} ||U_1 - U_2R||_F. \tag{8}$$

So, $d(U, U^*)$ then represents the distance between the class containing a critical point $U \in R^{nxr}$ and the optimal factor class U^*. Then, it is evident from the figure below that the direction from U to $-U^*$ has more negative curvature compared to the direction from U to U^* (i.e. U is closer to $-U^*$ compare to U from $+U^*$). Therefore, we choose direction $D = U - U^*R$, since it will produce a strictly negative curvature for each critical point U not corresponding to X^*.

By definition,

$$\bigtriangledown^2[g(U)](D, D) = 2\langle \bigtriangledown f(X), DD^T \rangle + [\bigtriangledown^2 f(X)](DU^T + UD^T, DU^T + UD^T) \tag{9}$$

As stated before, let $D = U - U^*R$, then
DD^T:

$$\begin{aligned} &= (U - U^*R)(U - U^*R)^T \\ &= (U - U^*R)(U^T - (U^*R)^T) \\ &= U^*U^{*T} - U^*RU^T - U(U^*R)^T + UU^T \qquad (10) \end{aligned}$$

For $D = U - U^*R$ and by critical point property, $\bigtriangledown f(UU^T)U = 0$,

$2\langle \bigtriangledown f(UU^T), DD^T\rangle$ becomes:

$$\begin{aligned} &= 2\langle f(UU^T), U^*U^{*T} - U^*RU^T - U(U^*R)^T + UU^T\rangle \qquad (11) \\ &= 2\langle f(UU^T), U^*U^{*T}\rangle \qquad (12) \\ &= 2\langle f(UU^T), U^*U^{*T} - UU^T\rangle \qquad (13) \end{aligned}$$

i.e. last three terms of (12) are zero.
Therefore, by replacing $X = UU^T$ back into (13), we get

$\bigtriangledown^2[g(U)](D, D)$

$$= 2\langle \bigtriangledown f(X), X^* - X\rangle + [\bigtriangledown^2 f(X)](DU^T + UD^T, DU^T + UD^T) \qquad (14)$$

Since we know $\bigtriangledown f(X^*) \succeq 0$, then $\bigtriangledown f(X^*)X^* = 0$ for $X^* \succeq 0$.

Therefore, $\bigtriangledown^2[g(U)](D, D)$

$$\leq \underbrace{2\langle \bigtriangledown f(X) - \bigtriangledown f(X^*), X^* - X\rangle}_{\text{Term1}} + \underbrace{[\bigtriangledown^2 f(X)](DU^T + UD^T, DU^T + UD^T)}_{\text{Term2}} \qquad (15)$$

In order to bound term1, let's use Taylor's theorem (as stated below), by inspecting gradient and hessian of term1.
Theorem 2.2 (Taylor's theorem): Suppose that $f : R^n \to R$ is continuously differentiable and that $p \in R^n$. Then we have that

$$f(x + p) = f(x) + \bigtriangledown f(x + tp)^T p,$$

for some t $\in (0, 1)$.

Moreover, if f is twice continuously differentiable, we have that

$$\bigtriangledown f(x + p) = \bigtriangledown f(x) + \int_0^1 \bigtriangledown^2 f(x + tp)p\, dt,$$

and that

$$f(x+p) = f(x) + \bigtriangledown f(x)^T p + \frac{1}{2} p^T \bigtriangledown^2 f(x+tp)p, \tag{16}$$

for some t $\in (0,1)$.

Therefore, by applying taylor's theorem [4] to term1, we get:

$$\begin{aligned} term1 &= -2\langle \int_0^1 [\bigtriangledown^2 f(tX + (1-t)X^*)](X^* - X)dt, X^* - X\rangle \\ &= -2\int_0^1 [\bigtriangledown^2 f(tX + (1-t)X^*)](X^* - X, X^* - X)dt \\ &= -2\alpha ||X^* - X||_F^2 \end{aligned} \tag{17}$$

where (19) is obtained by applying restricted well-conditioned assumption defined in (5), since $rank(tX + (1-t)X^*) \leq 2r$ and $rank(X^* - X) \leq 2r$ (note: $tX + (1-t)X^*$ is PSD matrix).

Now, to compute bound on term2, let us apply restricted well-conditioned assumption (5) on term2, then

$$\begin{aligned} term2 &= [\bigtriangledown^2 f(X)](DU^T + UD^T, DU^T + UD^T) \\ &= \beta ||DU^T + UD^T||_F^2 \end{aligned} \tag{18}$$

In order to separate the above term, let's consider the following lemma [5]
Lemma 1. Let U and Z be any two matrices in $R^{n\times r}$ such that $U^T Z = Z^T U$ is PSD. Assume Q is an orthogonal matrix whose columns span Range(U), then

$$||(U-Z)U^T||_F^2 = \frac{1}{8}||UU^T - ZZ^T||_F^2 + (3 + \frac{1}{2\sqrt{2}-2})||(UU^T - ZZ^T)QQ^T||_F^2 \tag{19}$$

where QQ^T being the projection matrix on Range(U).
So, applying lemma 1 to restricted well-conditioned term2 obtained in (21), we get

$$\begin{aligned} term2 &= 4\beta ||DU^T||_F^2 \\ &= 4\beta[\frac{1}{8}||X - X^*||_F^2 + (3 + \frac{1}{2\sqrt{2}-2})||(X - X^*)QQ^T||_F^2] \end{aligned} \tag{20}$$

To find upper bound on $||(UU^T - ZZ^T)QQ^T||_F$ term, we will define lemma 2
Lemma 2: Suppose the objective function f(X) is is twice differentiable and

satisfies the restricted well-conditioned assumption. Further, let U be any critical point of $f(UU^T)$ and Q be the orthonormal basis spanning Range(U). Then

$$||(UU^T - U^*U^{*T})QQ^T||_F \leq \frac{\beta-\alpha}{\beta+\alpha}||(UU^T - U^*U^{*T})||_F \tag{21}$$

Applying lemma 2 to (20) then results in

$$term2 \leq 4\beta[\frac{1}{8} + (3 + \frac{1}{2\sqrt{2}-2})\frac{(\beta-\alpha)^2}{(\beta+\alpha)^2}]||(X - X^*)||_F^2 \tag{22}$$

By setting $\frac{\beta}{\alpha} \leq 1.5$ (from restricted well-conditioned assumption)

$$term2 \leq 1.76\alpha||(X - X^*)||_F^2 \tag{23}$$

Combining term1 and term2 from (17) and (22) results in:

$$term1 + term2 = \bigtriangledown^2[g(U)](D, D) \leq -0.24\alpha||(X - X^*)||_F^2 \tag{24}$$

Finally, let us relate the lifted distance $||X - X^*||_F^2$ with the factored distance $||U - U^*R||$ using following lemmas 3 and 4 for $r > r^*$ and $r = r^*$: Lemma 3: Assume that $U_1, U_2 \in R^{n\times r}$. Then

$$||U_1U_1^T - U_2U_2^T||_F \geq \min\{\rho(U_1), \rho(U_2)\}\; d(U_1, U_2).$$

where $\rho(U_1)$ denotes the largest singular value of $\rho(U_1)$.
Using lemma 3, for $r > r^*$:

$$\begin{aligned}\bigtriangledown^2[g(U)](D, D) &\leq -0.24\alpha \min\{\rho(U)^2, \rho(U^*)^2\}||D||_F^2 \\ &= -0.24\alpha \min\{\rho(U)^2, \rho(X^*)^2\}||D||_F^2\end{aligned} \tag{25}$$

Lemma 4[6]: Assume that $U_1, U_2 \in R^{n\times r}$ and rank(U_1) = r. Then

$$||U_1U_1^T - U_2U_2^T||_F \geq 2(\sqrt{2}-1)\rho(U_1)\; d(U_1, U_2).$$

Therefore, by using lemma 4, when $r = r^*$, we get

$$\begin{aligned}\bigtriangledown^2[g(U)](D, D) &\leq -0.19\alpha\rho(U^*)^2||D||_F^2 \\ &= -0.19\alpha\rho(U^*)^2||D||_F^2\end{aligned} \tag{26}$$

As it can be observed from (25) and (26), the hessian will always be negative, since the right-hand side terms of both equations has a constant factor (0.19 or 0.24), $\alpha, \rho(U)^2$, and frobenius norm. Since α is a positive constant, $\rho(U)^2$ is highest singular value i.e. positive, and frobenius norm is always positive, the resulting product will therefore be always be positive but with the negative sign, it will make the product negative. Therefore, the proof concludes here, since it was proved that for all convex functions satisfying restricted well-condition property, the hessian is always negative, and therefore the local optimum will either converge to global optimum or be a saddle point.

3 Conclusion

This study focuses on Burer-Monteiro's re-paremetrization to improve computational efficiency of general convex functions, by factoring $X = UU^T$ for symmetric convex functions. Under the restricted well-condition, f(X) will have each critical point either corresponding to a global optimum of the original convex programs, or is a strict saddle where the Hessian matrix has a strictly negative eigenvalue. Such a benign landscape then allows many iterative optimization methods to escape from all the saddle points and converge to a global optimum with even random initialization.

References

[1] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming.* 95(2):329–357, 2003.

[2] Q. Li and G. Tang. The nonconvex geometry of low-rank matrix optimizations with general objective functions. *In 2017 IEEE Global Conference onSignal and Information Processing (GlobalSIP),* pages 1235–1239, Nov 2017.

[3] Samuel Burer and Renato DC Monteiro. Local Minima and Convergence in Low-Rank Semidefinite Programming. *Mathematical Programming.* 103: 427, 2005.

[4] Jorge Nocedal and Stephen Wright. *Numerical optimization.* Springer Science & Business Media, 2 edition, 2006.

[5] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems,* pages 3873–3881, 2016.

[6] Stephen Tu, Ross Boczar, Mahdi Soltanolkotabi, and Benjamin Recht. Low-rank solutions of linear matrix equations via procrustes flow. *arXiv preprint arXiv:1507.03566*, 2015.