The Mixing method for Maxcut-SDP problem

Po-Wei Wang
Machine Learning Department
Carnegie-Mellon University
Pittsburgh, PA 15213
poweiw@cs.cmu.edu

J. Zico Kolter
Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA 15213
zkolter@cs.cmu.edu

Abstract

We propose a fast, simple low-rank method for solving the SDP relaxation of Maxcut problem, in which the complexity per iteration is linear in the number of edges. Experiments suggest that it always converges to global optima and is orders of magnitude faster than existing state-of-the-art low-rank methods. Specially, the proposed method is capable of solving problems involving millions of vertices.

1 Introduction

Although solving nonconvex optimization problems can be NP-hard, several such problems have well-known approximate solutions. One of the canonical examples of such problem is Maxcut problem, the task of finding a graph cut that maximizes the weight of the cut. There is a well-known semidefinite relaxation of the Maxcut problem, which requires solving the semidefinite program

$$\underset{X \succeq 0,\ x_{ii}=1}{\text{minimize}} \quad \langle C, X \rangle, \tag{MAXCUT–SDP}$$

where C is the adjacency matrix of the graph. [GW95] showed that the solution of the above problem gives a strong (0.878) approximation for the Maxcut problem by randomized rounding.

However, solving this SDP can still be impractical. For example, typical interior point solvers [e.g. BY05] for SDP problems require $O(n^3)$ time per iteration, plus $O(n^2)$ memory for storing X. Fortunately, [Bar95] and [Pat98] show that there is always a low-rank solution for MAXCUT–SDP. [BM03] use this fact to create a low-rank method for general SDP problems, solving the corresponding MAXCUT–VEC of sufficient rank with augmented Lagrangian method. In this paper, we proposed a simple algorithm called MIXING that further exploits this low-rank structure. The MIXING method is strictly decreasing and its steps are feasible in contrast to [BM03]. Experiments suggest that it typically converges to global optima in practice, and is orders of magnitude faster than existing state-of-the-art methods.

2 The Mixing Method

By factorizing $X = V^T V$ for some $V \in \mathbb{R}^{k \times n}$, we see that the Maxcut SDP is equivalent to the (nonconvex) optimization problem

$$\underset{V \in \mathbb{R}^{k \times n},\ \|v_i\|=1}{\text{minimize}} \quad \langle C, V^\top V \rangle, \tag{MAXCUT–VEC}$$

Our MIXING method is simply the following. Observe that, if we minimize one v_i (column of V) at a time for MAXCUT–VEC, there is a closed-form solution

$$v_i^{\text{next}} = \text{normalize}\left(- \sum_j c_{ij} v_j \right). \tag{1}$$

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Thus, we can simply cycle through all i and update each v_i sequentially using (1). The algorithm is strictly decreasing, feasible, and the complexity per iteration is $O(\#\text{edges} \cdot k)$.

Despite its simplicity, this algorithm substantially outperforms all alternative approaches for solving the Maxcut SDP that we can find in the literature. Although the current best known bound for k is simply $k = n$ (rendering the algorithm less practical), in practice much lower values still converge to global optima (as measured by globally convergent methods). Further, we prove the following theoretical property.

Theorem 1. *The* MIXING *method converges to a local optimum of the* MAXCUT–VEC *problem. Further, for sufficiently large k and good initialization (namely, where the relaxed objective is lower that the non-relaxed optimal Maxcut solution),* MIXING *converges to a global optimum.*

2.1 Proof of Theorem 1

We first formalize the MIXING method before starting the proof. Let L be the lower triangular part of C so that $C = L + L^T$ (because $C_{ii} = 0$). Then the update of MIXING can be written as

$$-V^r L = V^{r+1}\left(L^T + \operatorname{diag}(y^{r+1})\right), \tag{2}$$

in which V^r is the solution at the r-th iteration and

$$y_i^{r+1} = \|\sum_{j<i} c_{ij} v_j^{r+1} + \sum_{j>i} c_{ij} v_j^r\|. \tag{3}$$

This formulation is similar to the analysis of block coordinate descent in [SH15]. Specifically, note that y^{r+1} is not a constant and thus the evolution is not linear. Further, we show that our method admit sufficient decrease for every cycle.

Lemma 2. *The function difference for each cycle of the* MIXING *method is*

$$\langle C, X^r\rangle - \langle C, X^{r+1}\rangle = \sum_{i=1}^n y_i^{r+1}\|v_i^r - v_i^{r+1}\|^2. \tag{4}$$

Proof. Left-multiply $V^{r\top}$ and $V^{r+1\top}$ to (2) and take the difference, we have

$$X^r L - X^{r+1} L = V^{r+1\top} V^r L - V^{r\top} V^{r+1} L^\top + (X^{r+1} - V^{r\top} V^{r+1}) \operatorname{diag}(y^{r+1}). \tag{5}$$

Because

$$\operatorname{tr}(XL) = \frac{1}{2}\operatorname{tr}(XC) \quad \text{and} \quad \operatorname{tr}(V^{r+1\top} V^r L) = \operatorname{tr}(V^{r\top} V^{r+1} L^\top),$$

taking trace on (5) gives

$$\frac{1}{2}(\operatorname{tr}(CX^r) - \operatorname{tr}(CX^{r+1})) = 0 + \sum_i y_i^{r+1}(1 - v_i^{r\top} v_i^{r+1}). \tag{6}$$

The result follows from $1 - v_i^{r\top} v_i^{r+1} = \frac{1}{2}\|v_i^r - v_i^{r+1}\|^2$. □

Lemma 2 means that the MIXING method admits a unique limit point. Let the limit be $\bar{V}$ and the corresponding limit of y^r be $\bar{y}$. Then $\bar{V}$ being a fix point of (2) implies

$$\bar{V}(C + \operatorname{diag}(\bar{y})) = 0, \tag{7}$$

which also means

$$\bar{X}(C + \operatorname{diag}(\bar{y})) = 0 \tag{8}$$

if we let $\bar{X} = \bar{V}^\top \bar{V}$. Remember that the KKT condition of MAXCUT–SDP is

$$X^* \succeq 0, \quad X^*_{ii} = 1 \qquad \text{prime feasibility} \tag{9}$$

$$X^*(C + \operatorname{diag}(y^*)) = 0 \qquad \text{complementary slackness} \tag{10}$$

$$C + \operatorname{diag}(y^*) \succeq 0 \qquad \text{dual feasibility.} \tag{11}$$

Thus, together with the feasibility of the MIXING method, the limit $\bar{V}$ satisfies (9) and (10). Now we show the local convergence to global optima by proving (11) also holds if we start from a neighborhood of the optima.

Theorem 3. *The* MIXING *method converges to global optima when the limit* $\bar{X}$ *satisfies*

$$\min_{u\in\{-1,1\}^n} u^\top C u \geq \langle C, \bar{X}\rangle. \tag{12}$$

Proof. By condition (12), we have

$$u^\top C u \geq \langle C, \bar{X}\rangle, \quad \forall u \in \{-1,1\}^n. \tag{13}$$

Further, property (8) for the limit point implies

$$\langle C, \bar{X}\rangle = -1^\top \bar{y}. \tag{14}$$

Together with

$$1^\top \bar{y} = u' \operatorname{diag}(\bar{y}) u, \quad \forall u \in \{-1,1\}^n, \tag{15}$$

we have

$$u^\top (C + \operatorname{diag}(\bar{y})) u \geq 0, \quad \forall u \in \{-1,1\}^n. \tag{16}$$

This means that $C + \operatorname{diag}(\bar{y}) \succeq 0$ and the KKT condition holds. □

Because the MIXING method is strictly decreasing, Theorem 3 implies that if the initial V^0 satisfies

$$\min_{u\in\{-1,1\}^n} u^\top C u \geq \langle C, V^{0\top} V^0\rangle, \tag{17}$$

the method converges to global optima. However, experiments suggest that the MIXING method converges globally without such assumption. It remains an open problem to prove the global convergence for the method.

3 Experiments

In this section we compare the proposed MIXING method with the following algorithms: 1) **DSDP 5.8** [BY05], which implements an interior point method with dual scaling. It is very mature and supports sparse matrices. We report its primal objective and measure only the solving time in the comparison. 2) **SDPLR 1.03-beta** [BM03], which implements a low-rank augmented Lagrangian method with L-BFGS line search. Note that its solutions may be infeasible by the nature of Lagrangian method. To make a fair comparison, we project the solution to feasible set and recalculate the function value at every data point. We do not include this projection or function evaluation time in the running time of the algorithm.

We evaluate the mixing methods on several datasets. Among them, the GSET [HR00] and SP3DL [FPRR02] are the standard dataset for comparing Maxcut algorithms. We also consider larger dataset in the SNAP collection [LK14] downloaded from University of Florida sparse matrix collection [DH11]. In particular, we compare the MIXING method with SDPLR on ca-HepPh [LKF07] and email-Enron [LKF05]. To test the limit of our proposed method, we also try the MIXING method on the roadNet.CA dataset [LLDM09], which consists of near 2 million vertices and 5.5 million edges. Also, for theoretical purpose, we test these methods on noisy hypercube and noisy sphere, because they are known to be hard [KV15] for the MAXCUT–SDP problem. In particular, we create the noisy hypercube dataset by connecting all the edges between $x, y \in \{-1,1\}^{10}$ satisfying $x^\top y \leq 0.7 \cdot 10$, and the noisy sphere by sampling $4 \cdot 2^{10}$ samples uniformly from the unit sphere and connecting edges the same way as the noisy hypercube. We set $k = \sqrt{n}$ on all the experiments except for roadNet.CA, which we set $k = 500$ to avoid memory issues.

The experiments (Figure 1) indicate that the MIXING method is in different order of complexity compared to DSDP, and is consistently orders of magnitude faster than SDPLR. Note that DSDP is excluded from the SNAP experiments because it is too slow for comparison, and only the MIXING method survives in the roadNet.CA dataset due to the memory consumption.

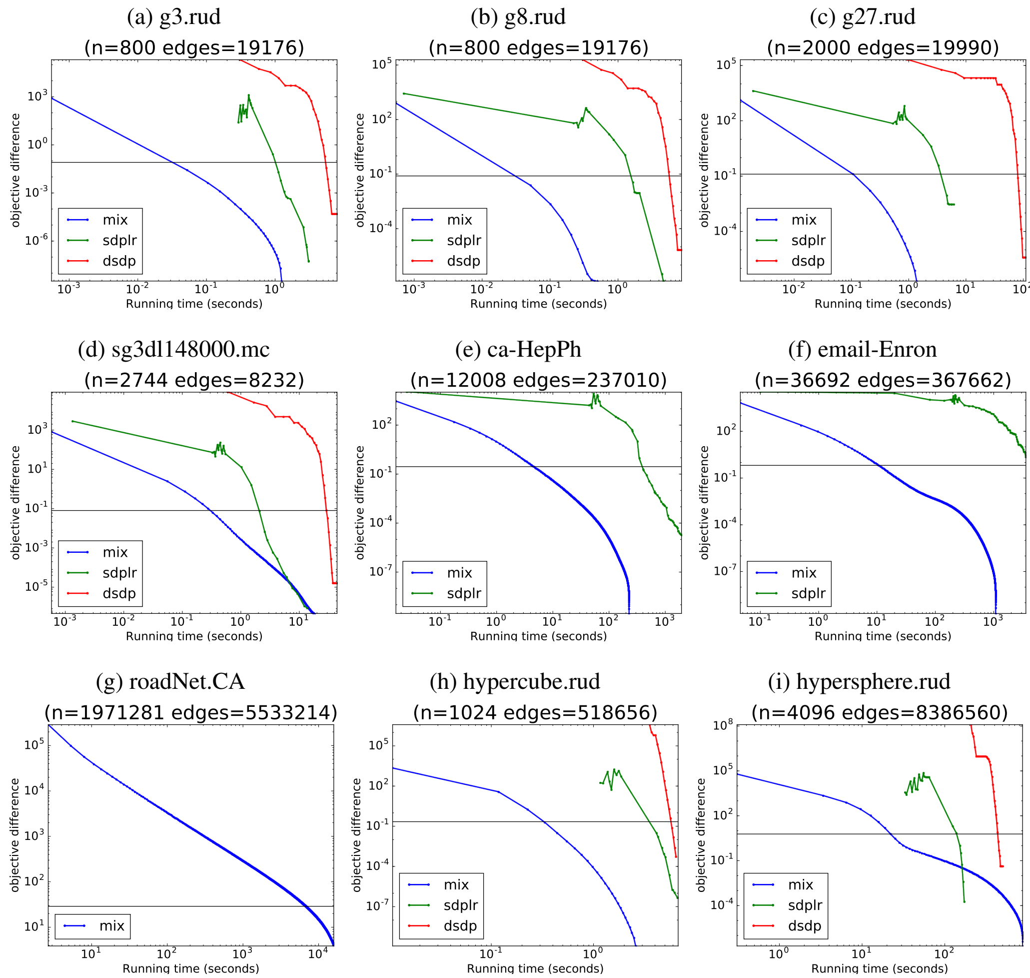

Figure 1: The log-log plots of function difference $f(x) - f^*$ v.s. running time in seconds. The horizontal line marks $1e-4$ times the initial function difference of the MIXING method.

References

[Bar95] Alexander I. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete & Computational Geometry*, 13(2):189–202, 1995.

[BM03] 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.

[BY05] Steven J. Benson and Yinyu Ye. DSDP5: Software for semidefinite programming. Technical Report ANL/MCS-P1289-0905, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, September 2005. Submitted to ACM Transactions on Mathematical Software.

[DH11] Timothy A Davis and Yifan Hu. The university of florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1, 2011.

[FPRR02] Paola Festa, Panos M Pardalos, Mauricio GC Resende, and Celso C Ribeiro. Randomized heuristics for the max-cut problem. *Optimization methods and software*, 17(6):1033–1058, 2002.

[GW95] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[HR00] Christoph Helmberg and Franz Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[KV15] Subhash A Khot and Nisheeth K Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative-type metrics into ℓ 1. *Journal of the ACM (JACM)*, 62(1):8, 2015.

[LK14] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[LKF05] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

[LKF07] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[LLDM09] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[Pat98] Gábor Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of operations research*, 23(2):339–358, 1998.

[SH15] Ruoyu Sun and Mingyi Hong. Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *Advances in Neural Information Processing Systems*, pages 1306–1314, 2015.