Manifold optimization for k-means clustering

Timothy Carson*, Dustin G. Mixon†, Soledad Villar*, Rachel Ward*
*Department of Mathematics, University of Texas at Austin
{tcarson, mvillar, rward}@math.utexas.edu
†Department of Mathematics and Statistics, Air Force Institute of Technology
dustin.mixon@gmail.com

***Abstract*—We introduce a manifold optimization relaxation for k-means clustering that generalizes spectral clustering. We show how to implement it as gradient descent in a compact manifold. We also present numerical simulations of the algorithm using Manopt [5]. An extended version of this article, with further theory and numerical simulations will be available as [8].**

I. Introduction

In unsupervised machine learning, clustering problems consist of partitioning a given finite set of data points $\{x_i\}_{i=1}^n$ into subsets such that a dissimilarity cost function is minimized. The standard k-means clustering problem assumes the data points are in Euclidean space, the number of subsets is k, and its objective is to minimize the sum of the squared distance of each data point to the centroid of its cluster:

$$\begin{aligned} \text{minimize} \quad & \sum_{t=1}^{k} \sum_{i\in A_t} \left\| x_i - \frac{1}{|A_t|} \sum_{j\in A_t} x_j \right\|_2^2 \\ \text{subject to} \quad & A_1 \sqcup \cdots \sqcup A_k = \{1,\dots,n\}. \end{aligned} \tag{1}$$

Using a simple computation, one can rewrite the k-means problem as

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\text{Trace}(DX) \\ \text{subject to} \quad & X := \sum_{t=1}^{k} \frac{1}{|A_t|} 1_{A_t} 1_{A_t}^T, \end{aligned} \tag{2}$$

where D is an $n\times n$ matrix such that $D_{ij} = \|x_i - x_j\|^2$, and X is a projection matrix into the span of the indicator vectors of each cluster.

An equivalent formulation for k-means is the following optimization in the set of rank k matrices:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\text{Trace}(DYY^T) \\ \text{subject to} \quad & Y \in \mathbb{R}^{n\times k},\ YY^T 1 = 1, \\ & Y^T Y = I_k,\ Y \geq 0. \end{aligned} \tag{3}$$

Here the constraint $Y^TY = I_k$ means that Y has orthonormal columns. Using that $Y \geq 0$ entry wise, we obtain that $Y_{ij} \neq 0$ implies that $Y_{ik} = 0$ for all $k \neq j$, so Y has exactly one nonnegative entry per row. The constraint $YY^T1 = 1$ implies that the vector $1 \in \mathbb{R}^n$ belongs to the span of the columns of Y. Therefore if $Y_{ij} \neq 0$ and $Y_{lj} \neq 0$ then $Y_{ij} = Y_{lj}$. This shows that every feasible matrix for (3) is feasible for (2).

A. Relaxations of the k-means problem

Optimization problems (1)-(3) are equivalent to k-means, which is NP-hard [2]. A typical way to tackle such hard problems is to relax the discrete feasible set to a larger set, then use analytic tools to solve the larger problem, and finally round a solution of the larger problem into a feasible solution for the original problem.

For instance, the *spectral clustering* technique is based on the following relaxation of (3):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\text{Trace}(DYY^T) \\ \text{subject to} \quad & Y \in \mathbb{R}^{n\times k},\ Y^TY = I_k. \end{aligned} \tag{4}$$

Note that the solution of (4) is a matrix with columns consisting of the top k eigenvectors of D.

In general, spectral clustering algorithms replace the matrix D by a matrix $-K$, where K corresponds to the Gram matrix of the points mapped to a higher dimensional space (i.e.: $K_{ij} = \langle \phi(x_i), \phi(x_j)\rangle$ for $\phi : \mathbb{R}^n \to \mathbb{R}^N$.) One particularly common implementation uses the Gaussian kernel: $K_{ij} = \exp(-\|x_i - x_j\|^2/\sigma^2)$.

Another relaxation of k-means that has become quite popular lately is Peng and Wei's k-means SDP [15], [3], which solves

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\text{Trace}(DX) \\ \text{subject to} \quad & \text{Trace}X = k,\ X1 = 1,\ X \geq 0,\ X \succeq 0, \end{aligned} \tag{5}$$

where $X \succeq 0$ means that X is symmetric and positive semidefinite. Note that the results from [9] indicate that the constraint $X \geq 0$ is strictly weaker than the constraint $Y \geq 0$.

In this paper we consider the relaxation of (3) where we include the non-negative constraint $Y \geq 0$ as a penalization in the objective, and restrict the minimization to $Y \in M$ where M is a smooth submanifold of $\mathbb{R}^{n\times k}$:

$$\begin{aligned} \text{minimize} \quad & \text{Trace}(DYY^T) + \lambda \|Y_-\|_F^2 \\ \text{subject to} \quad & Y \in M. \end{aligned} \tag{6}$$

Here λ is a non-negative parameter, Y_- indicates the negative entries of Y, and M is the submanifold

$$M = \{Y \in \mathbb{R}^{n\times k} : Y^T Y = I_k,\ YY^T 1 = 1\}. \tag{7}$$

By removing the constraint $Y \geq 0$, our discrete feasible set becomes a smooth manifold without boundary, which allows us to use manifold optimization algorithms.

Also note that adding the constraint $YY^T 1 = 1$ to spectral clustering is simple and doesn't change its spectral nature (in particular, if $\lambda = 0$ the solution can be computed from the top $k-1$ eigenvectors of the projection of D onto $\{1\}^\perp \subset \mathbb{R}^n$). What makes this optimization significantly different from spectral clustering is the term $\lambda\|Y_-\|_F^2$ in the objective.

B. Manifold optimization

Our relaxation (6) is a constrained optimization where the set of constraints is a Riemannian manifold. See Section III for more details of the algorithm. For this kind of problems there is a beautiful theory [1] that allows us to think of our problem as an unconstrained optimization where we replace the usual Euclidean ambient space by a Riemannian manifold M.

Let M be a Riemannian manifold and $f : M \to \mathbb{R}$ a smooth function. The objective is to solve $\min_{Y\in M} f(Y)$. The basic gradient descent algorithm relies on gradient and retraction functions,

$$\operatorname{grad}_f : M \to TM, \tag{8}$$

$$\operatorname{retr}_Y : T_Y M \to M. \tag{9}$$

The gradient is computable using the Riemannian structure. The retraction is a choice of map which should satisfy

$$\operatorname{retr}_Y(0) = 0, \quad \left.\frac{d}{dt}\right|_{t=0} \operatorname{retr}_Y(tV) = V. \tag{10}$$

A canonical choice of retraction map is the exponential map for M, but this is not always computationally feasible. If M is a submanifold of euclidean space then $\operatorname{retr}_Y(V)$ will be a first order approximation to $Y+V$.

The algorithm consists of iteratively following the gradient of f in the tangent space and then retracting back into the manifold:

$$Y_{n+1} = \operatorname{retr}_{Y_n}(-\alpha_n \operatorname{grad}_f(Y_n)).$$

The stepsize α_n can be set to be a small constant or adaptively chosen through a line search. Second order algorithms like trust regions have also been adapted to the manifold optimization setting [1].

In this paper we restrict ourselves to first order methods, where gradient descent methods with backtracking Amijo line-search are proven to converge to a stationary point under mild hypotheses [4].

Theorem 6 in [4]. *Let M a Riemannian manifold and $f : M \to \mathbb{R}$ bounded from below. Assume that $f \circ \operatorname{retr}_Y$ is Lipchitz with constant L independent of Y. Then a gradient descent on M with backtracking Amijo line-search initialized at Y_0 returns Y_* such that*

$$f(Y_*) \leq f(Y_0) \quad \text{and} \quad \|\operatorname{grad}_f(Y_*)\| \leq \varepsilon$$

in $O(1/\varepsilon^2)$ iterations.

As the objective function we consider is locally Lipschitz in $\mathbb{R}^{n\times k}$ and our M is compact, the hypotheses of this theorem are satisfied in our case.

II. The k-means manifold.

In order to implement the manifold optimization relaxation of k-means we need to explicitly construct the gradient and retraction maps (8) and (9). The tangent space to M at Y is given by

$$T_Y M = \{V \in \mathbb{R}^{n\times k} : V^T Y + Y^T V = 0, \\ (VY^T + YV^T)1 = 0\}. \tag{11}$$

Our manifold is a submanifold of a Euclidean space, and our objective function is defined on the entirety of this Euclidean space. As such, we may compute the gradient of the objective function on our manifold by orthogonally projecting its gradient in Euclidean space onto the tangent space to our manifold. That is, from the orthogonal projection $\Pi_{T_Y M} : T_Y \mathbb{R}^{n\times k} \to T_Y M$ we can compute

$$\operatorname{grad}_f^M(Y) = \Pi_{T_Y M} \circ \nabla f(Y)$$

where ∇f is the gradient of f in the ambient Euclidean space $\mathbb{R}^{n\times k}$. For our objective function (with parameter λ),

$$f_\lambda(Y) = \operatorname{Trace}(DYY^T) + \lambda\|Y_-\|^2,$$

the gradient is computed in Section II-A to be

$$\nabla f_\lambda(Y) = 2DY + 2\lambda(Y_-). \tag{12}$$

In Section II-B we compute the orthogonal projection. It is:

$$\Pi_{T_Y M}(W) = W - 2Y\Omega - (x1^T + 1x^T)Y, \tag{13}$$

where

$$x = \frac{1}{n} WY^T 1 \in \mathbb{R}^n,$$

$$\Omega = \frac{1}{4}(W^T Y + YW^T - 2Y^T(x1^T + 1x^T)Y) \in \mathbb{R}^{k\times k}.$$

We use the following retraction:

$$\operatorname{retr}_Y(V) = \exp(B)\exp(A')Y, \tag{14}$$

where

$$A = Y^T V \in \mathbb{R}^{k\times k},$$

$$A' = YAY^T \in \mathbb{R}^{n\times n},$$

$$B = VY^T - YV^T - 2A' \in \mathbb{R}^{n\times n}.$$

Here exp denotes the matrix exponential. We explain this retraction in Section II-E.

A. Gradient of the objective function

We compute the ambient space gradient $\nabla f_\lambda(Y)$ of f_λ. By definition we know $\nabla f_\lambda(Y) = W$ if and only if for all $V \in T_Y M$ we have

$$\begin{aligned}\langle V, W\rangle &= D_Y f_\lambda(V) \\ &= \operatorname{Trace}(D(VY^T + YV^T)) \\ &\quad + \lambda \operatorname{Trace}(V(Y_-^T) + (Y_-)V^T)\end{aligned}$$

where $D_Y f_\lambda(V)$ is the directional derivative of f_λ. Equivalently,

$$\operatorname{Trace}(WV^T) = \operatorname{Trace}\left(((D+D^T)Y + 2\lambda(Y_-))V^T\right).$$

Since D is symmetric we find (12).

B. Projection of a vector onto $T_Y M$

Let $L_1 : \mathbb{R}^{n\times k} \to \mathbb{R}^{k\times k}_{sym}$ be $L_1(W) = W^T Y + Y^T W$ and let $L_2 : \mathbb{R}^{n\times k} \to \mathbb{R}^n$ be $L_2(W) = (WY^T + YW^T)1$. We can write the tangent space as $T_Y M = \ker(L)$ where $L = L_1 \oplus L_2 : \mathbb{R}^{n\times k} \to \mathbb{R}^{k\times k}_{sym} \times \mathbb{R}^n$.

We can use $\ker(L)^\perp = \operatorname{im}(L^*)$ to compute a parameterization for $(T_Y M)^\perp$. Then we will solve $W - L^*(\Omega, x) \in \ker(L)$ for (Ω, x) to find the projection $\Pi_{T_Y M}(W) = W - L^*(\Omega, x)$.

We calculate that for Ω symmetric:

$$\begin{aligned}\langle L_1 W, \Omega\rangle &= \langle W^T Y, \Omega\rangle + \langle Y^T W, \Omega\rangle \\ &= \langle W^T, \Omega Y^T\rangle + \langle W, Y\Omega\rangle \\ &= 2\langle W, Y\Omega\rangle,\end{aligned}$$

from which we see $L_1^*\Omega = 2Y\Omega$. Now calculate for $x \in \mathbb{R}^n$:

$$\begin{aligned}\langle L_2 W, x\rangle &= \langle (WY^T + YW^T)1, x\rangle \\ &= \langle WY^T + YW^T, x1^T\rangle \\ &= \langle W, x1^T Y + 1x^T Y\rangle,\end{aligned}$$

so $L_2^* x = (x1^T + 1x^T)Y$

Now we can find Ω and x so that $W - L_1^*\Omega - L_2^* x \in \ker(L)$ by solving the system of equations:

$$\begin{cases} L_1(W - L_1^*\Omega - L_2^* x) = 0 \\ L_2(W - L_1^*\Omega - L_2^* x) = 0 \end{cases}$$

The first equation reads

$$W^T Y + Y^T W - 4\Omega - 2Y^T(x1^T + 1x^T)Y = 0$$

We can use this to substitute Ω in the second equation to get:

$$u + Bx = 0$$

where $u = (I_n - YY^T)WY^T 1$ and $B = -n(I_n - YY^T)$. In particular we can choose x and Ω as below (13). (There is nonuniqueness in x and Ω because the image of L is not the full stated range, but of course the projection is unique.)

C. Homogenous structure of M

Recall the Definition (7) of M. Let M_0 be the manifold

$$M_0 = \{YY^T : Y \in M\} \subset \mathbb{R}^{n\times n}_{sym}.$$

The manifold M_0 is the set of orthogonal projections onto a k dimensional subspace of $\mathbb{R}^n$ including the vector 1_n, and as such each member of M_0 is determined by its image (i.e. the span of its columns). A point in the manifold M has the additional information of a choice of basis of the image of $X = YY^T$.

For a subspace $A \subset \mathbb{R}^n$, $O(A)$ is the group of orthogonal matrices for which $Av = v$ for all $v \in A^\perp$. Let $\mathbb{P} = \{1_n\}^\perp \subset \mathbb{R}^n$. We can see M as a homogenous space; it has a transitive action by $O(\mathbb{P}) \times O(\mathbb{R}^k)$ given by multiplication by the first factor on the left and the $O(\mathbb{R}^k)$ factor on the right:

$$\begin{aligned}M \times O(\mathbb{P}) \times O(\mathbb{R}^k) &\to M \\ (Y, Q, R) &\mapsto QYR.\end{aligned}$$

The multiplication on the right by an element of $O(\mathbb{R}^k)$ controls changes which change Y but not X, which may be seen directly from the computation $(YR)(YR)^T = YRR^T Y^T = YY^T$. The multiplication on the left by $Q \in O(\mathbb{P})$ allows for any change in $X \in M_0$.

Multiplication of $Y \in M$ on the right by $R \in O(\mathbb{R}^k)$ is always equivalent to multiplication of Y on the left by $R' = YRY^T$:

$$R'Y = (YRY^T)Y = YR(Y^T Y) = YR.$$

The matrix (YRY^T) is an orthogonal projection onto $\operatorname{im}(X)$ composed with an orthogonal transformation of $\operatorname{im}(X)$, which may also be shown by computing

$$R'(I - X) = 0, \quad R(R')^T = X.$$

Recalling that X is an orthogonal projection, the first equality shows that R' annihilates $\operatorname{im}(X)^\perp$ and the second equality shows that R' acts as an orthogonal transformation of $\operatorname{im}(X)$ (on which X is the identity).

For each Y_0 the action by $O(\mathbb{P}) \times O(\mathbb{R}^k)$ has a stabilizer which is determined by $X_0 = Y_0 Y_0^T$. The action by $O(\mathbb{R}^k)$ generates all $Y \in M$ with the same X_0:

$$\{Y_0 R : R \in O(\mathbb{R}^k)\} = \{Y \in M : YY^T = Y_0 Y_0^T\},$$

but there are also elements of $O(\mathbb{P})$ which fix X_0 namely,

$$\begin{aligned}&\{Q \in SO(\mathbb{R}^n) : QX_0 = X_0 Q, Q1_n = 1_n\} \\ &= O(\operatorname{im}(X)) \oplus O(\ker(X)) \\ &\subset O(\mathbb{P}) = \{Q \in O(\mathbb{R}^n) : Q1_n = 1_n\}.\end{aligned}$$

D. Splitting the tangent space to M

We may use our understanding of M as a homogenous space to compute a splitting of the tangent space $T_Y M$ into two orthogonal parts: those which generate changes

that fix X, and its perpendicular space. Let $\mathfrak{so}(\mathbb{R}^n)$ be the set of antisymmetric matrices in $\mathbb{R}^n$.

The matrices in $\{CY : C \in \mathfrak{so}(\mathbb{R}^n)\}$ which are tangent to the direction of fixed X (generated by $O(\mathrm{im}(X)) \oplus O(\ker(X))$) are

$$\{C \in \mathfrak{so}(\mathbb{R}^n) : CX = XC\}.$$

This is the kernel of the linear map $\mathfrak{so}(n) \to \mathbb{R}^{n\times n}_{sym}$ given by $L(C) = CX - XC$. The adjoint map $L^* : \mathbb{R}^{n\times n}_{sym} \to \mathfrak{so}(n)$ is given by $L^*(\Omega) = \Omega X - X\Omega$. Therefore

$$\{C \in \mathfrak{so}(\mathbb{R}^n) : CX = XC\}^{\perp} = \{\Omega X - X\Omega : \Omega \in \mathbb{R}^{n\times n}_{sym}\}.$$

Given $V \in T_Y M$ we aim to write $V = BY + YA$ where $A \in \mathfrak{so}(k)$, $B \in \mathfrak{so}(\mathbb{P})$ and furthermore $B = \Omega X - X\Omega$ for $\Omega \in \mathbb{R}^{n\times n}_{sym}$. Under this ansantz, we compute $Y^T V$, VY^T, and YV^T and use that $Y^T Y = I_k$ to find

$$A = Y^T V, \tag{15}$$

$$B = \Omega X - X\Omega = VY^T - YV^T - 2YAY^T. \tag{16}$$

Using the formula for $T_Y M$ (11) one can check that we actually recover V as $V = BY + YA$ and that $A \in \mathfrak{so}(\mathbb{R}^k)$ and $B \in \mathfrak{so}(\mathbb{P})$, i.e.

$$A + A^T = 0, B + B^T = 0, B1_n = 0. \tag{17}$$

E. A retraction map

Given $V \in T_Y M$ we aim to find a retraction

$$\mathrm{retr}_Y(V) \in M$$

satisfying (10). Write V as $V = BY + YA$ as in (15), (16). Note we may also see V as $BY + (YAY^T)Y = (B + (YAY^T))Y$; this is using the equivalence between multiplication on the right by $O(\mathbb{R}^k)$ and multiplication on the left by $O(\mathrm{im}(X))$, mentioned in Section II-C. Let $A' = YAY^T$ so $V = (B + A')Y$. Now set

$$\mathrm{retr}_Y(V) = \exp(B)\exp(A')Y. \tag{18}$$

The property (10) is straightforward given the differential equation satisfied by exp, but it is not as obvious that $\tilde{Y} = \mathrm{retr}_Y(V) \in M$. The condition $\tilde{Y}^T\tilde{Y} = I$ follows because we are performing left multiplication by orthogonal matrices. To check that $\tilde{Y}\tilde{Y}^T 1_n = 1_n$ we may compute,

$$\begin{aligned}
&(\exp(B)\exp(A')Y)(\exp(B)\exp(A')Y)^T 1_n \\
&= \exp(B)\exp(A')YY^T\exp(-A')\exp(-B)1_n \\
&= \exp(B)YY^T\exp(-B)1_n \\
&= \exp(B)YY^T 1_n = \exp(B)1_n = 1_n.
\end{aligned}$$

We have used, successively, that $A'(YY^T) = (YY^T)A' = A'$ (so $\exp(-tA')$ commutes with YY^T), that $B1_n = 0$ (so $\exp(-tB)$ fixes 1_n) and that $YY^T 1_n = 1_n$. Note that the order of the matrix exponentials matters. For example,

$$V \mapsto \exp(A')\exp(B)Y \text{ and } V \mapsto \exp(B)Y\exp(A)$$

are paths satisfying (10) but will not lie on the manifold if $A'1 \neq 0$.

III. Numerical algorithm

The projection and retraction functions from the previous section allow us to implement gradient descent algorithms in Manopt. In order to tackle the k-means problem, the algorithm we propose entails iteratively solving the manifold optimization relaxation of k-means (6), increasing the penalty λ until convergence to a k-means feasible Y. See Algorithm 1.

Algorithm 1 Manifold optimization iteration for k-means clustering

1: $\lambda_0 \leftarrow 0$
2: **repeat**
3: $\quad Y_{n+1} \leftarrow \mathrm{GradientDescent}(f_\lambda)$ {Initialized at Y_n}
4: $\quad \lambda_{n+1} \leftarrow 2\lambda_n + 1$
5: **until** $\|Y_-\|_F < \varepsilon$

Theorem 6 in [4] guarantees that step 3 in the algorithm finds a stationary point of the objective. The fact that $\lambda\,\mathrm{Trace}(DYY^T)$ is bounded for $Y \in M$ suggests the algorithm may converge to a feasible clustering. It would be very interesting to show that Algorithm 1 converges to the actual k-means solution for *sufficiently nice* data, provided a *good initialization*.

A. Numerical simulations

Due space constraints we only present a simple numerical simulation. We refer the reader to [8] for further numerical evaluation of the algorithm. For the experiment we sample points uniformly from 4 unit balls in $\mathbb{R}^4$ with centers separated by 2.05 (following the stochastic ball model introduced in [12] and further studied in [3], [10]). We sample 22, 18, 19 and 21 points from each ball respectively. We run Algorithm 1 using Manopt to implement step 3. In Figure 1 we plot the results.

IV. Discussion

Before manifold optimization became popular, Burer and Monteiro [7] introduced the idea of using a low rank factorization of a matrix in order to solve a semidefinite program of the form.

$$\begin{aligned}
&\text{minimize } \mathrm{Trace}(CX) \\
&\text{subject to } \mathrm{Trace}(A_i X) = b_i \; 1 \leq i \leq m,\; X \succeq 0
\end{aligned} \tag{19}$$

According to the Pataki bound [14], the solution of (19) is a matrix $X = YY^T$ for some $Y \in \mathbb{R}^{n\times p}$ with $\frac{p(p+1)}{2} \leq m$. Therefore if we replace the positive semidefinite constraint in (19) by $X = YY^T$ for $Y \in R^{n\times p}$ the global minimizer of both problems coincide. In their paper, Burer and Monteiro propose an augmented lagrangian

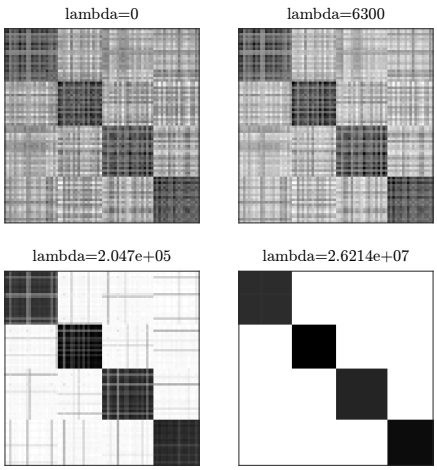

Fig. 1. Illustrations of YY^T for iterations of Algorithm 1 with successive values of λ. Note the top left image is equivalent to a purely spectral method while the bottom right image coincides with the planted solution of the k-means problem. The algorithm is oblivious to the planted order of points. We choose the order where the first points belong to the first cluster, and so on, to simplify visualization.

iteration in $X = YY^T$. They prove it converges to a stationary point of their objective. Since the objective is not convex there is a priori no guarantee that it won't converge to some spurious stationary point.

Later work by Journée and collaborators [11] introduced a manifold optimization algorithm and proved that, under somewhat restrictive conditions (not satisfied by our clustering problem), it converges to the global optimizer.

Recent work by Boumal, Voroninski and Bandeira [6] extend Journée's work by showing that the Bourer-Monteiro problem (i.e. the minimization in matrices of the form YY^T) is equivalent to respective SDP for some specific problems. They actually show that for those problems there are no spurious stationary points.

Some natural questions arise: (a) How small can p be chosen with still no spurious stationary point? In their original paper Burer and Monteiro suggested that if the rank of the planted solution is k one should be able to choose $p = k+1$ or $p = k+2$. (b) Is it possible to adapt manifold optimization methods to singular manifolds? And in particular, (c) can a theory like this be developed for manifolds with boundary?

To the best of our knowledge the best algorithms that can deal efficiently and reliably with semidefinite programs with non-negative constraints are based on interior point methods [13]. As far as we know, there is no theory that provides convergence guarantees for matrix factorization based algorithms in presence of non-negative constraints; nor even successful implementation for algorithms like that for generic SDPs with non-negative constraints.

ACKNOWLEDGEMENTS

The authors thank Afonso Bandeira and Dmitri Chklovskii for a conversation about different relaxations of k-means that inspired the introduction of this paper.

TC was partially supported by NSF DMS 1148490. SV and RW were partially supported by NSF CAREER grant #1255631. DGM was partially supported by NSF DMS 1321779, AFOSR F4FGA05076J002, and an AFOSR Young Investigator Research Program award. The views expressed in this article are those of the authors and do not reflect the official policy or position of the U.S. Air Force, Department of Defense, or the U.S. Government.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[2] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of euclidean sum-of-squares clustering. *Mach. Learn.*, 75(2):245–248, May 2009.

[3] P. Awasthi, A. S. Bandeira, M. Charikar, R. Krishnaswamy, S. Villar, and R. Ward. Relax, no need to round: Integrality of clustering formulations. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 191–200. ACM, 2015.

[4] N. Boumal, P.-A. Absil, and C. Cartis. Global rates of convergence for nonconvex optimization on manifolds. *arXiv preprint arXiv:1605.08101*, 2016.

[5] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

[6] N. Boumal, V. Voroninski, and A. Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.

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

[8] T. Carson, D. G. Mixon, S. Villar, and R. Ward. Manifold optimization for semidefinite programs with non-negative constraints, (in preparation) 2017.

[9] H. Dong and K. Anstreicher. Separating doubly nonnegative and completely positive matrices. *Mathematical Programming*, pages 1–23, 2013.

[10] T. Iguchi, D. G. Mixon, J. Peterson, and S. Villar. Probably certifiably correct k-means clustering. *Mathematical Programming*, pages 1–38, 2016.

[11] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

[12] A. Nellore and R. Ward. Recovery guarantees for exemplar-based clustering. *Information and Computation*, 245:165–180, 2015.

[13] Y. Nesterov, A. Nemirovskii, and Y. Ye. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994.

[14] G. 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.

[15] J. Peng and Y. Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 18(1):186–205, 2007.