Optimal Approximation of Doubly Stochastic Matrices

Nikitas Rontsis
Department of Engineering Science
University of Oxford, OX1 3PJ, UK

Paul J. Goulart
Department of Engineering Science
University of Oxford, OX1 3PJ, UK

Abstract

We consider the least-squares approximation of a matrix C in the set of doubly stochastic matrices with the same sparsity pattern as C. Our approach is based on applying the well-known Alternating Direction Method of Multipliers (ADMM) to a reformulation of the original problem. Our resulting algorithm requires an initial Cholesky factorization of a positive definite matrix that has the same sparsity pattern as $C + I$ followed by simple iterations whose complexity is linear in the number of nonzeros in C, thus ensuring excellent scalability and speed. We demonstrate the advantages of our approach in a series of experiments on problems with up to 82 million nonzeros; these include normalizing large scale matrices arising from the 3D structure of the human genome, clustering applications, and the SuiteSparse matrix library. Overall, our experiments illustrate the outstanding scalability of our algorithm; matrices with millions of nonzeros can be approximated in a few seconds on modest desktop computing hardware.

1 INTRODUCTION

Consider the following optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|X - C\|^2 \\ \text{subject to} & X \text{ nonnegative} \\ & X_{i,j} = 0 \;\forall\; (i,j) \text{ with } C_{i,j} = 0 \\ & X\mathbf{1} = \mathbf{1},\;\; X^T\mathbf{1} = \mathbf{1}, \end{array} \qquad (\mathcal{P})$$

which approximates the symmetric real, $n \times n$ matrix C in the Frobenius norm by a *doubly stochastic* matrix $X \in \mathbb{R}^{n\times n}$, i.e., a matrix with nonnegative elements whose columns and rows sum to one, that has the same sparsity pattern as C. Problem $\mathcal{P}$ is a *matrix nearness* problem, i.e., a problem of finding a matrix with certain properties that is close to some given matrix; see Higham (1989) for a survey on matrix nearness problems.

Adjusting a matrix so that it becomes doubly stochastic is relevant in many fields, e.g., for preconditioning linear systems (Knight, 2008), as a normalization tool used in spectral clustering (Zass and Shashua, 2007), optimal transport (Cuturi, 2013), and image filtering (Milanfar, 2013), or as a tool to estimate a doubly stochastic matrix from incomplete or approximate data used e.g., in longitudinal studies in life sciences (Diggle et al., 2002), or to analyze the 3D structure of the human genome (Rao et al., 2014)

A related and widely used approach is to search for a diagonal matrix D such that DCD is doubly stochastic. This is commonly referred to as the *matrix balancing* problem, or the Sinkhorn's algorithm (Knight, 2008). Such a scaling matrix D exists and is unique whenever C has total support. Perhaps surprisingly, when C has only nonnegative elements, then the matrix balancing problem can be considered as a matrix nearness problem. This is because, DCD has been shown to minimize the relative entropy measure (Idel, 2016, Observation 3.19) (Benamou et al., 2015), i.e., it is the solution of the following convex problem

$$\begin{array}{ll} \text{minimize} & \sum_{i,j} X_{i,j} \log X_{i,j}/C_{i,j} \\ \text{subject to} & X \text{ doubly stochastic} \\ & X_{i,j} = 0 \;\forall\; (i,j) \text{ with } C_{i,j} = 0, \end{array} \qquad (1)$$

where we define $0 \cdot log(0) = 0$, $0 \cdot log(0/0) = 0$, and $1 \cdot log(1/0) = \infty$. Note, however, that the relative entropy is not a proper distance metric since, it is not symmetric, does not satisfy the triangular inequality and can take the value ∞.

The *matrix balancing problem* can be solved by iterative methods with remarkable scalability. Standard and simple iterative methods exist that exhibit linear per-iteration complexity w.r.t. the number of nonze-

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

ros in C and linear convergence rate (Knight, 2008), (Idel, 2016). More recent algorithms can exhibit a super-linear convergence rate and increased robustness in many practical situations (Knight and Ruiz, 2013).

The aim of the paper is to show that the direct minimization of a least squares objective in doubly stochastic approximation, which has a long history dating back to the influential paper of (Deming and Stephan, 1940)[1], can also be solved efficiently. This gives practitioners a new solution to a very important problem of doubly stochastic matrix approximation, which might prove useful for cases where the relative entropy metric is not suitable to their problem.

The approach we present is also flexible in the sense that it can handle other interesting cases, for example, where C is asymmetric or rectangular, $\|\cdot\|$ is a weighted Frobenius norm, and $X\mathbf{1}$ and $X^T\mathbf{1}$ are required to sum to an arbitrary, given vector. We discuss these generalizations in §3.2.

Related work Zass and Shashua (2007) consider the problem $\mathcal{P}$ in the case when C is fully dense. They suggest an alternating projections algorithm that has linear per-iteration complexity. The approach of Zass and Shashua (2007) resembles the results of §3.2, but as we will see in the experimental section, it is not guaranteed to converge to an optimizer.

The paper is organized as follows: In Section 2, we introduce a series of reformulations to Problem 1, resulting in a problem that is much easier to solve.In Section 3, we suggest a solution method that reveals a particular structure in the problem. Finally, in Section 4, we present a series of numerical results that highlight the scalability of the approach.

Notation used The symbol $\otimes$ denotes the Kronecker product, vec$(\cdot)$ the operator that stacks a matrix into a vector in a column-wise fashion and mat$(\cdot)$ the inverse operator to vec$(\cdot)$ (see (Golub and Van Loan, 2013, 12.3.4) for a rigorous definition). Given two matrices, or vectors, with equal dimensions $\odot$ denotes the Hadamard (element-wise) product. $\|\cdot\|$ denotes the 2-norm of a vector and the Frobenius norm of a matrix, while card$(\cdot)$ the number of nonzero elements in a vector or a matrix. Finally ϵ_{p} denotes the machine precision.

[1]Deming and Stephan (1940) consider the weighted least squares cost $\frac{1}{2}\sum_{i,j}[X_{i,j} - C_{i,j}]^2/C_{i,j}$ which we treat in §3.2.

2 MODELING $\mathcal{P}$ EFFICIENTLY

In this section, we present a reformulation of the doubly stochastic approximation problem $\mathcal{P}$ suitable for solving large-scale problems. One of the difficulties with the original formulation, $\mathcal{P}$, is that it has n^2 variables and $2n^2 + 2n$ constraints. Attempting to solve $\mathcal{P}$ with an off-the-shelf QP solver, such as Gurobi (Gurobi Optimization LLC, 2018), can result in out-of-memory issues just by representing the problem's data even for medium-sized problems.

In order to avoid this issue, we will perform a series of reformulations that will eliminate variables from $\mathcal{P}$. The final problem, i.e., $\mathcal{P}_2$, will have significantly fewer variables and constraints while maintaining a remarkable degree of structure, which will be revealed and exploited in the next section.

We first take the obvious step of eliminating all variables in $\mathcal{P}$ that are constrained to be zero, as prescribed by the constraint

$$X_{i,j} = 0 \text{ for all } (i,j) \text{ such that } C_{i,j} = 0. \tag{2}$$

Indeed, consider vec$(\cdot)$, the operator that stacks a matrix into a vector in a column-wise fashion and H an $n_{\text{nz}} \times n^2$ matrix, where $n_{\text{nz}} := \text{card}(C)$, that selects all the nonzero elements of vec(C). Note that H depends on the sparsity pattern S of C, defined as the $0-1$ $n \times n$ matrix

$$S_{i,j} := \begin{cases} 0 & C_{i,j} = 0 \\ 1 & \text{otherwise,} \end{cases} \tag{3}$$

but we will not denote this dependence explicitly. We can now isolate the nonzero variables contained in X and C in $n_{\text{nz}}-$dimensional vectors defined as

$$x := H\,\text{vec}(X), \quad c := H\,\text{vec}(C). \tag{4}$$

Note that x is simply a re-writing of X in $\mathbb{R}^{n_{\text{nz}}}$, since for any $X \in \mathcal{S} := \{X \in \mathbb{R}^{n\times n} \mid X_{i,j} = 0 \text{ for all } C_{i,j} = 0\}$ we have

$$H^T H\,\text{vec}(X) = \text{vec}(X) \Leftrightarrow H^T x = \text{vec}(X),$$

due to the fact that $H^T H = \text{diag}(\text{vec}(S))$ Thus every x defines a unique $X \in \mathcal{S}$ and vice versa.

We can now describe the constraints of $\mathcal{P}$, i.e., $X \geq 0$, $X\mathbf{1}_n = \mathbf{1}_n$ and $X^T\mathbf{1}_n = \mathbf{1}_n$, on the "$x-$space". Obviously $X \geq 0$ trivially maps to $x \geq 0$. Furthermore, recalling the standard Kronecker product identity

$$\text{vec}(LMN) = (N^T \otimes L)\,\text{vec}(M) \tag{5}$$

for matrices of compatible dimension, the constraints $X\mathbf{1}_n = \mathbf{1}_n$ and $X^T\mathbf{1}_n = \mathbf{1}_n$ of $\mathcal{P}$ can be rewritten in

an equivalent vectorized form as

$$\begin{bmatrix}\mathbf{1}_n^T \otimes I_n \\ I_n \otimes \mathbf{1}_n^T\end{bmatrix} \operatorname{vec}(X) = \mathbf{1}_{2n} \tag{6}$$

or, by noting that $H^T x = \operatorname{vec}(X)$, as

$$\begin{bmatrix}\mathbf{1}_n^T \otimes I_n \\ I_n \otimes \mathbf{1}_n^T\end{bmatrix} H^T x = \mathbf{1}_{2n}. \tag{7}$$

Thus $\mathcal{P}$ can be rewritten in the "$x-$space" as

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}\|x - c\|^2 \\ \text{subject to} & x \geq 0 \\ & \begin{bmatrix}\mathbf{1}_n^T \otimes I_n \\ I_n \otimes \mathbf{1}_n^T\end{bmatrix} H^T x = \mathbf{1}_{2n}.\end{array} \tag{$\mathcal{P}_1$}$$

A further reduction of the variables and constraints of $\mathcal{P}_1$ can be achieved by exploiting the symmetry of C. To this end, note that when C is symmetric the optimal doubly stochastic approximation will also be symmetric according to the following proposition:

Proposition 2.1. *If C is symmetric, then the optimal solution X^* of $\mathcal{P}$ is also symmetric.*

Proof. Assume the contrary, so that X^* is optimal but asymmetric. Then the matrix X^{*T} is a feasible solution for $\mathcal{P}$ since it remains element-wise negative when its row and column sums are exchanged, and has an identical objective value since C is assumed symmetric. Then define the symmetric matrix $\tilde{X}$ as the convex combination

$$\tilde{X} := \frac{1}{2}(X^* + X^{*T})$$

which is also a feasible point for $\mathcal{P}$. Since the objective function is strictly convex (at least on the subset of elements of X not constrained to be zero), the objective function evaluated at $\tilde{X}$ will be strictly lower than that for both X^*, contradicting the optimality of X^*. □

Note that the above proof, like the one of the following Theorem 2.2, are simply algebraic calculations.

It follows that restricting the feasible set of $\mathcal{P}$ to symmetric matrices does not affect its solution. We will exploit this by eliminating all the variables embedded into X that are below its main diagonal. Define an upper triangular matrix X_u consisting of scaled elements of X such that $X = X_u + X_u^T$, and likewise for C, i.e.

$$X_u := U \odot X, \quad C_u := U \odot C \tag{8}$$

where

$$U := \begin{bmatrix}\frac{1}{2} & 1 & \cdots & 1 \\ & \ddots & & \vdots \\ & & \frac{1}{2} & 1 \\ 0 & & & \frac{1}{2}\end{bmatrix}. \tag{9}$$

As in the previous definitions, define H_u as the matrix that stacks all the nonzeros of C_u in an column-wise fashion, which is used to extract the nonzero elements of X_u and C_u, i.e., scaled nonzero elements of the upper triangular part of X, to the vectors

$$x_u := H_u \operatorname{vec}(X_u), \quad c_u := H_u \operatorname{vec}(X_u). \tag{10}$$

We can now write down our reduced optimization problem. Note that, although it might not be directly evident, the following problem possesses a remarkable degree of internal structure that is exploited in the suggested solution algorithm of the next section.

Theorem 2.2. *Solving $\mathcal{P}$ for a symmetric C is equivalent to solving*

$$\begin{array}{ll}\textit{minimize} & \frac{1}{2}\|p \odot (x_u - c_u)\|^2 \\ \textit{subject to} & x_u \geq 0 \\ & Ax_u = \mathbf{1}_n,\end{array} \tag{$\mathcal{P}_2$}$$

where

$$p := H_u \operatorname{vec}\left(\begin{bmatrix}2 & \sqrt{2} & \cdots & \sqrt{2} \\ & \ddots & & \vdots \\ & & 2 & \sqrt{2} \\ 0 & & & 2\end{bmatrix}\right)$$

$$A_1 := \mathbf{1}_n^T \otimes I_n, \quad A_2 := I_n \otimes \mathbf{1}_n^T$$

and $A := (A_1 + A_2)H_u^T$, in the sense that $\mathcal{P}$ is feasible iff $\mathcal{P}_2$ is, and the optimizer X^* of $\mathcal{P}$ can be constructed from the optimizer x_u^* of $\mathcal{P}_2$ using (10) *and* (8)*.*

Proof. We will first show that every feasible x_u of $\mathcal{P}_2$ defines a feasible X for $\mathcal{P}$, where $\operatorname{vec}(X_u) := H_u^T x_u$, with the same objective value. Similarly to S, define $S_u \in \mathbb{R}^{n \times n}$ as an $0-1$ matrix that represents the sparsity of C and the upper triangular of C respectively, i.e.

$$S_{u_{i,j}} = \begin{cases}0 & C_{i,j} = 0, \text{ or } i < j \\ 1 & \text{otherwise.}\end{cases}$$

The equality of the objective value can be shown as follows:

$$\begin{aligned}&\|p \odot (x_u - c_u)\| \\ =&\left\|(\mathbf{1}_{n\times n}\sqrt{2} + (2-\sqrt{2})I_n) \odot (X_u - C_u)\right\| \\ =&\left\|(\mathbf{1}_{n\times n}\sqrt{2} + (2-\sqrt{2})I_n) \odot U \odot (X - C)\right\| \\ =&\left\|(\mathbf{1}_{n\times n}\sqrt{2} + (1-\sqrt{2})I_n) \odot S_U \odot (X - C)\right\| \\ =&\|S \odot (X - C)\| = \|X - C\|.\end{aligned} \tag{11}$$

Furthermore, similarly to (5)-(7), we have

$$\begin{aligned}X\mathbf{1}_n &= (X_u + X_u^T)\mathbf{1}_n \\ &= (\mathbf{1}_n^T \otimes I_n + I_n \otimes \mathbf{1}_n^T) \operatorname{vec}(X_u) \\ &= (\mathbf{1}_n^T \otimes I_n + I_n \otimes \mathbf{1}_n^T)H_u^T x_u = Ax_u,\end{aligned} \tag{12}$$

resulting in $X\mathbf{1}_n = \mathbf{1}_n$ due to the feasibility of x_u for $\mathcal{P}_2$. Due to the symmetry of X we also get $X^T\mathbf{1}_n$. Finally, X is nonnegative construction and has sparsity pattern S. Therefore X is feasible for $\mathcal{P}$.

Likewise, following (11)-(12) in reverse order, we can show that every symmetric feasible matrix X of $\mathcal{P}$ defines an $x_u := H_u \operatorname{vec}(X_u)$, where $X_u := U \odot X$, that is feasible for $\mathcal{P}_2$ and has identical objective value. Since only symmetric optimizers exist for $\mathcal{P}$ (Lemma 2.1) this concludes the proof. □

Unlike $\mathcal{P}$ which has n^2 and $2n$ constraints, $\mathcal{P}_2$ has approximately $n_{\text{nz}}/2$ and n constraints. Furthermore, it possesses a specific internal structure that we exploit in the solution algorithm presented in §3.

3 SOLUTION METHOD

In this section, we describe how the reduced problem $\mathcal{P}_2$ can be solved with ADMM. We begin with a brief introduction to the ADMM algorithm in the general setting, which follows (Boyd et al., 2011), and then describe how ADMM can be applied efficiently for $\mathcal{P}_2$.

Several optimization problems, including reformulations of $\mathcal{P}_2$ (Stellato et al., 2017), are concerned with the minimization of a function q that can be decomposed into two parts $q = f + g$ such that optimizing independently f or g is tractable. Ideally, if f and g operate on disjoint variables, i.e., if $q(\chi, \psi) = f(\chi)+g(\psi)$, then q can also be optimized efficiently by merely minimizing q over χ and ψ independently. However, it is often that case that there is some coupling between χ and ψ which we assume to be in the form $A\chi+B\psi = d$, resulting in the following optimization problem

$$\begin{array}{ll}\text{minimize} & f(\chi) + g(\psi) \\ \text{subject to} & A\chi + B\psi = d\end{array} \tag{13}$$

where χ, ψ denote the decision variables, f, g are proper lower-semicontinuous convex functions, and A, B, d are matrices of appropriate dimensions.

The Alternating Direction Method of Multipliers (ADMM) is a first-order (i.e., "gradient-based") algorithm that solves (13) while exploiting the assumption that f and g can be easily optimized independently. Indeed, ADMM iterates as follows,

$$\begin{aligned}\chi^{k+1} &= \inf_{\chi} L_\rho(\chi, \psi^k, y^k) \\ \psi^{k+1} &= \inf_{\psi} L_\rho(\chi^{k+1}, \psi, y^k) \\ y^{k+1} &= y^k + \rho(A\chi^{k+1} + B\psi^{k+1} - d)\end{aligned}$$

where

$$\begin{aligned}L_\rho(\chi, \psi, y) := f(\chi) + h(\psi) + y^T(A\chi + B\psi - d) \\ +(\rho/2)\|A\chi + B\psi - d\|^2,\end{aligned}$$

is the augmented Lagrangian of (13) and ρ is a positive penalty parameter.

Recalling that $(\mathcal{P}_2)$ is a Quadratic Problem, we note that solving QPs with ADMM has been widely studied in the literature (Stellato et al., 2017), (Garstka et al., 2019), (O'Donoghue et al., 2016). We will follow the approach of (Stellato et al., 2017) which can solve $\mathcal{P}_2$ by applying ADMM to the following splitting

$$\begin{array}{ll}\text{minimize} & f(\tilde{x}, \tilde{z}) + g(x, z) \\ \text{subject to} & (\tilde{x}, \tilde{z}) = (x, z)\end{array} \tag{14}$$

where f and g, are defined as

$$\begin{aligned}f(\tilde{x}, \tilde{z}) &= \frac{1}{2}x^T P x - P c_u + \mathcal{I}_{A\tilde{x}=\tilde{z}}(\tilde{x}, \tilde{z}) \\ g(x, z) &= \mathcal{I}_{x\geq 0}(x) + \mathcal{I}_{z=\mathbf{1}_{2n}}(z)\end{aligned}$$

and $P := \operatorname{diag}(p \odot p)$, $\mathcal{I}_{A\tilde{x}=\tilde{z}}$, $\bar{n}_{\text{nz}}$ are the number of nonzeros in the upper triangular of C and $\mathcal{I}_{x\geq 0}$, $\mathcal{I}_{z=\mathbf{1}_{2n}}$ denote the indicator functions of the sets $\{(x, z) \in \mathbb{R}^{\bar{n}_{\text{nz}}} \times \mathbb{R}^{2n} \mid Ax = z\}$, $\{x \in \mathbb{R}^{\bar{n}_{\text{nz}}} \mid x \geq 0\}$ and $\{x \in \mathbb{R}^{2n} \mid x = \mathbf{1}_{2n}\}$ respectively.

Applying ADMM for the problem (14) results[2] in Algorithm 1, where Π_+ denotes the projection of a vector to the nonnegative orthant and $\Pi_1(x) := \mathbf{1}$ for any vector x. Refer to (Stellato et al., 2017, §3) for details.

Algorithm 1: Solving $\mathcal{P}_2$ with ADMM

given initial values x^0, z^0, y^0 and parameters $\rho > 0$, $\sigma > 0$, and $\alpha \in (0, 2)$;
repeat
  $(\tilde{x}^{k+1}, \tilde{z}^{k+1}) \leftarrow$ solution of the linear system
  $$\begin{bmatrix}(P + \sigma I_{\bar{n}_{\text{nz}}}) & \rho A^T \\ \rho A & -\rho I_{2n}\end{bmatrix}\begin{bmatrix}\tilde{x}^{k+1} \\ \tilde{z}^{k+1}\end{bmatrix} = \begin{bmatrix}\sigma x^k - w^k + A^T(\rho z^k - y^k) + P c_u \\ 0\end{bmatrix};$$
  $x^{k+1} \leftarrow \Pi_+(\alpha\tilde{x}^{k+1} + (1-\alpha)x^k)$;
  $z^{k+1} \leftarrow \Pi_{\mathbf{1}}(\alpha\tilde{z}^{k+1} + (1-\alpha)z^k + \rho^{-1}y^k)$;
  $w^{k+1} \leftarrow w^k + \sigma(\alpha\tilde{x}^{k+1} + (1-\alpha)x^k - x^{k+1})$;
  $y^{k+1} \leftarrow y^k + \rho(\alpha\tilde{z}^{k+1} + (1-\alpha)z^k - z^{k+1})$;
until *termination condition is satisfied*;

[2] Algorithm 1 includes two extensions over the simple ADMM algorithm discussed in §3: it uses *over-relaxation*, a commonly used variation that can increase the speed of convergence (Eckstein and Bertsekas, 1992, Figure 2), and two different step sizes ρ, σ for the update of Lagrange multipliers w and y respectively. The parameters ρ, σ and α are chosen according to (Stellato et al., 2017) wherein exhaustive numerical testing was done to identify the best choices of these parameters when solving QPs across a wide variety of problem structures.

The most computationally intensive operation of Algorithm 1 is the solution of the $(\bar{n}_{\text{nz}}+n)\times(\bar{n}_{\text{nz}}+n)$ linear system in line 3. We will show that its solution can be obtained by solving instead a reduced $n\times n$ linear system.

Fact 3.1. *Consider the following linear system*

$$\begin{bmatrix} P+\sigma I_{\bar{n}_{nz}} & \rho A^T \\ \rho A & -\rho I_{2n} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} \tag{15}$$

where $x, r \in \mathbb{R}^{\bar{n}_{nz}}, z \in \mathbb{R}^{\bar{n}_{nz}}$ and $A \in \mathbb{R}^{2n\times\bar{n}_{nz}}$. Its solution can be obtained as follows

1. *Obtain z by solving the following $n\times n$ positive definite linear system*
$$(\rho A(P+\sigma I_{\bar{n}_{nz}})^{-1}A^T + I_n)z = A(P+\sigma I_{\bar{n}_{nz}})^{-1}r \tag{16}$$
2. *Obtain x as $(P+\sigma I_{\bar{n}_{nz}})^{-1}(r-\rho A^T z)$.*

Proof. The first block row gives $(P+\sigma I_{\bar{n}_{\text{nz}}})x+\rho A^T z = r \Leftrightarrow x = (P+\sigma I_{\bar{n}_{\text{nz}}})^{-1}(r-\rho A^T z)$. Reducing the variable x from (15) results in (16). □

Thus solving (15) can be reduced to solving a linear system with left hand side

$$(\rho A(P+\sigma I_{\bar{n}_{\text{nz}}})^{-1}A^T + I_n). \tag{17}$$

Fortunately, the reduced matrix (17), which is equal to $A\,\text{diag}(H_u\,\text{vec}(U))A^T$ with

$$U := \begin{bmatrix} (4+\sigma)^{-1} & (2+\sigma)^{-1} & \cdots & (2+\sigma)^{-1} \\ & \ddots & & \vdots \\ & & (4+\sigma)^{-1} & (2+\sigma)^{-1} \\ 0 & & & (4+\sigma)^{-1} \end{bmatrix},$$

turns out to be positive definite with a sparsity pattern matching that of $C+I$:

Theorem 3.2. *The following relation holds*

$$\begin{aligned} &A\,\text{diag}(H_u\,\text{vec}(D))A^T \\ =&S\odot(D+D^T) + \text{diag}(S\odot(D+D^T)\mathbf{1}) \end{aligned}$$

for any upper triangular $n\times n$ matrix D.

Proof. The proof is in the supplementary material. □

The solution of the reduced linear system (16) can be obtained given an initial Cholesky factorization of $\rho A(P+\sigma I_{\bar{n}_{\text{nz}}})^{-1}A^T + I_n$, or even with a factorization free algorithm e.g., Conjugate Gradients which primarily consists of repeated matrix-vector multiplications with $\rho A(P+\sigma I_{\bar{n}_{\text{nz}}})^{-1}A^T + I_n$. The fact that the linear system to be solved has the same sparsity pattern of $S+I$ can be particularly beneficial in cases where a fill-in reducing permutation is already known for the matrix under approximation C (and thus for S), since the same permutation could be used before the factorization of $\rho A(P+\sigma I_{\bar{n}_{\text{nz}}})^{-1}A^T + I_n$ resulting in reduced fill-in and thus significant speedup.

3.1 Convergence and Feasibility

We terminate Algorithm 1 when the primal and dual residuals of $\mathcal{P}_2$

$$\begin{aligned} r_{\text{prim}} &:= \|Au-\mathbf{1}\|_\infty, \\ r_{\text{dual}} &:= \left\|Px-c_u+A^Ty+w\right\|_\infty \end{aligned}$$

become smaller than some acceptable tolerance. Algorithm 1 is guaranteed to converge to the solution of $\mathcal{P}_2$ whenever $\mathcal{P}_2$, or equivalently $\mathcal{P}$, is feasible.

We next establish conditions that characterize the feasibility of $\mathcal{P}$:

Lemma 3.3. *Problem $\mathcal{P}$ is feasible if and only if there exists a set of indices $\mathcal{I} = \{(i_1,j_1)\ldots,(i_n,i_n)\}$ corresponding to exactly one nonzero element from each row and column of C.*

Proof. Regarding the "if" part, the $n\times n$ matrix

$$X_{i,j} = \begin{cases} 1 & \text{if} \quad i,j\in I \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

is a feasible point for $\mathcal{P}$. Regarding the "only if" part, since $\mathcal{P}$ has a feasible point X, then according to (Perfect and Mirsky, 1965, Theorem 1) there exists a set of indices $\mathcal{I} = \{(i_1,j_1)\ldots,(i_n,i_n)\}$ corresponding to exactly one nonzero element from each row and column of X. Due to the second constraint of $\mathcal{P}$, the same argument also holds for C. □

Note that Lemma 3.3 and (Perfect and Mirsky, 1965, Theorem 1) imply that $\mathcal{P}$ is feasible whenever the matrix balancing problem is feasible, i.e. when the matrix C has total support (Knight and Ruiz, 2013).

3.2 Special cases and generalizations

The case where C is almost or fully dense: In the special case where C is fully dense we have $S = \mathbf{1}_n\mathbf{1}_n^T$, and Theorem 3.2 gives

$$\rho A(P+\sigma I_{\bar{n}_{\text{nz}}})^{-1}A^T + I_n = \alpha I + \beta\mathbf{1}_{n\times n}$$

where

$$\alpha := \frac{\sigma\rho}{(2+\sigma/2)(2+\sigma)} + \frac{n\rho}{2+\sigma} + 1 \quad \text{and} \quad \beta := \frac{\rho}{2+\sigma}.$$

Using the Sherman-Morrison formula, we can explicitly calculate the inverse of (16) as

$$(\rho A(P+\sigma I_{\bar{n}_{\text{nz}}})^{-1}A^T + I_n)^{-1} = \frac{1}{\alpha}\left(I - \frac{\beta\mathbf{1}_n\mathbf{1}_n^T}{\alpha+\beta n}\right). \tag{19}$$

We can then solve (15) and perform ADMM on $\mathcal{P}_2$ without the need to perform an initial matrix factorization.

This approach can also be extended to cases where C has a relatively small number of zero elements. Indeed, by avoiding the elimination of the zero variables of $\mathcal{P}$ we get the following variant of $\mathcal{P}_2$:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|p \odot x_u - c_u\|^2 \\ \text{subject to} & x_u \geq 0 \\ & x_{u_i} = 0 \; \forall \; i \text{ with } c_{u_i} = 0 \\ & Ax_u = \mathbf{1}_{2n} \end{array} \qquad (\mathcal{P}_3)$$

where p, A, x_u, and c_u are defined according to Section 2, but with H_u an $\frac{(n+1)n}{2} \times n^2$ linear map that extracts *all* the upper triangular elements of a vectorized $n \times n$ matrix. $\mathcal{P}_3$ can then be solved with Algorithm 1, with the following two changes. First, we replace Π_+ (line 2, Algorithm 1) with $\Pi_{\mathcal{S}}$, where $\mathcal{S} := \{x \in \mathbb{R}^{(n^2+n)/2} \mid x \geq 0 \text{ and } x_i = 0 \text{ for all } i \text{ such that } c_i = 0\}$. Secondly, the solution of the linear system of line 3 of Algorithm 1 is trivially solved using Fact 3.1 and (19).

Solving variants of $\mathcal{P}$ with Algorithm 1: Algorithm 1 can be easily adjusted to the case where $\|\cdot\|$ in the objective of $\mathcal{P}$ is a weighted Frobenius norm, i.e. $\|X\| = \|W \odot X\|_F$ where W is a given symmetric matrix. The only thing that has to change is the definition of p, and thus of $P := \text{diag}(p \odot p)$, to:

$$p := H_u \,\text{vec}\left(\begin{bmatrix} 2 & \sqrt{2} & \cdots & \sqrt{2} \\ & \ddots & & \vdots \\ & & 2 & \sqrt{2} \\ 0 & & & 2 \end{bmatrix} \odot W \right).$$

Theorem 3.2 could then be used to solve the linear system of Algorithm 1 (line 3) efficiently. Similarly, we can allow for general constraints $X\mathbf{1} = r$ and $X^T\mathbf{1} = r$ in $\mathcal{P}$, where r is a given nonnegative vector, by simply changing $\Pi_{\mathbf{1}}$ in line 5 of Algorithm 1 to $\Pi_r(x) := r$. Finally, non-square or asymmetric matrices C can be solved via use of $\mathcal{P}$ for the symmetric matrix $\begin{bmatrix} 0 & C \\ C^T & 0 \end{bmatrix}$.

4 NUMERICAL EXPERIMENTS

In this section we present numerical results of Algorithm 1 on a range of matrix normalization problems. We provide a `Julia` implementation of the Algorithm, along with code that generates all the results of this section at:
`github.com/oxfordcontrol/DoublyStochastic.jl`

4.1 Normalizing Hi-C Contact Matrices of the Human Genome in the 3D Space

We first present results on the application of our method to real-world contact matrices describing the 3D structure of the human genome, starting with a description of the nature of these matrices. The human genome has an end-to-end length on the order of meters when unfolded, but fits inside the cell nucleus with dimensions on the order of micrometers, implying that the genome is heavily folded in the 3D space. The 3D structure of the genome can be examined by breaking the genome into a number of pieces and measuring how many contacts exist between each piece in the 3D space (Rao et al., 2014). This produces *Hi-C contact matrices*, where the term *Hi-C* describes the particular experimental procedure used.

The process is, however, subject to errors and experimental constraints. To alleviate these issues, the contact matrix is normalized so that all its rows and columns sum to the same value. The matrix balancing approach is often the method of choice for this task (Rao et al., 2014, Supplemental Material II.b), but other methods have also been suggested in the literature (Yaffe and Tanay, 2011).

We will show that our approach can also be used for this task even for contact matrices containing hundreds of millions of nonzero entries as in Rao et al. (2014). In particular, we consider normalization of the contact matrix of the 7$^{\text{th}}$ chromosome of the GM12878 cell[3], thus replicating (Rao et al., 2014, Figure 1.C). Since the genome consists of sequences of the bases *adenine, guanine, cytosine* and *thymine*, it is typical to measure the length of each genome piece by the average number of bases it contains. The total range of the contact matrices is 0 to 160 mega-bases (Mb). We consider two discretization lengths, 1 kilobase (Kb), and 5Kb, which result in contact matrices of 151 and 82 million nonzeros respectively. Figure 1 provides detailed views of the contact matrices, spanning the range [137.2–137.8Mb] for the contact map at 5Kb resolution and the [137.55–137.75Mb] for the one at 1Kb resolution. These regions were chosen to highlight interesting regions of the contact matrix as they appear in (Rao et al., 2014, Figure 1.C, rightmost column, two bottom subfigures).

Our approach produces considerably different normalized contact matrices than the matrix balancing approach. In particular, our approach results in contact matrices that have increased sparsity and higher contrast. This is unlike the matrix balancing approach which results in a normalized matrix that has exactly the same nonzeros as the original matrix. Although further investigation is necessary for the evaluation of

[3]The data corresponding to the thresholding criterion MAPQ $\geq$ 30 were used (Rao et al., 2014, Supplemental Material IIa.4). Obtained from the GM12878 "combined" intrachromosomal tarball at `ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525`

the suitability of the approach in Hi-C data, the results indicate that our method can be used to normalize very large Hi-C datasets leading to promising visual results.

4.2 Spectral clustering problems

Next, we present results of running our Algorithm on correlation matrices arising from spectral clustering

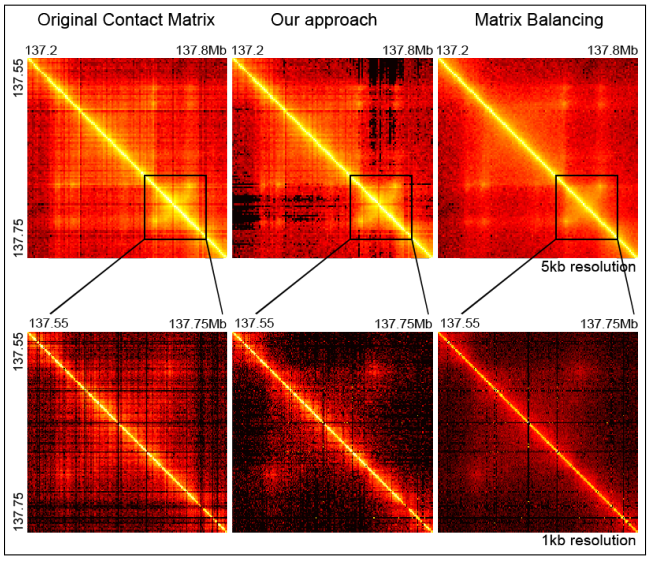

Figure 1: Details of Hi-C Contact Matrices for the 7$^{\text{th}}$ chromosome of the GM12878 cell, corresponding to (Rao et al., 2014, Figure 1.C, rightmost column, two bottom subfigures). Top row shows the area $[137.2 - 137.8\text{Mb}]^2$ for the contact matrix of 5Kb resolution. Bottom row shows the area $[137.55 - 137.75\text{Mb}]^2$ for the contact matrix of 1Kb resolution. Areas representing zero contacts are depicted in black, while areas with a high number of contacts are shown in yellow. The total area of the contact matrices is $[0 - 160\text{Mb}]^2$, thus the areas depicted are zoomed 71 and 640 thousand times in the top and bottom figures respectively. The leftmost column shows the original contact matrices. Rao et al. (2014) normalize the $n \times n$ contact matrix C via the matrix balancing method of (Knight and Ruiz, 2013) so that its columns and rows sum to $\sum_{i,j} C_{i,j}/n$. The resulting normalized matrices are depicted in the rightmost column. The middle column depicts the results of Algorithm 1 for normalizing the matrices so that its columns and rows sum to $\sum_{i,j} C_{i,j}/n$. The Conjugate Gradient method is used to solve the linear system (16) since using a Cholesky factorization resulted in memory issues. A tolerance of 10^{-3} is used for the termination of our Algorithm. The normalization of the 5Kb and 1Kb contact matrix take 701 and 3136 seconds respectively on a single-threaded implementation on Intel Gold 5120, 192GB memory.

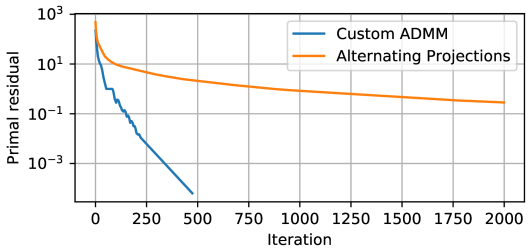

Figure 2: Comparison of the primal convergence (i.e. feasibility) of Algorithm 1 vs. the approach of (Zass and Shashua, 2007) (21) on the Spambase dataset with (22) as affinity criterion with $\sigma = 100$.

(Zass and Shashua, 2007). In spectral clustering one is given a set of points $\{x_i \in \mathbb{R}^d\}$ to be arranged into l clusters. To this end, an *affinity matrix* C is generated with each entry $C_{i,j}$ representing a measure of the pairwise similarity between points i and j. This matrix is then normalized and used by later stages of the clustering procedure. Zass and Shashua (2007) suggested that the normalization

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|X - C\|^2 \\ \text{subject to} & X \text{ nonnegative} \\ & X\mathbf{1}_n = \mathbf{1}_n,\ X^T\mathbf{1}_n = \mathbf{1}_n, \end{array} \tag{20}$$

leads to superior clustering performance in various different test cases. Note that solving (20) is equivalent to solving $\mathcal{P}$ for $C + \epsilon_{\text{p}}\mathbf{1}_{n\times n}$. Note that the formulation (20) of Zass and Shashua (2007) does not exploit sparsity in C. Nevertheless, Zass and Shashua (2007) suggested that the following iterative scheme can be used to solve 20

$$\tilde{X}^k = X^k + n^{-2}(\mathbf{1}_n^T X^k \mathbf{1}_n + n)\mathbf{1}_{n\times n} - n^{-1}(X^k\mathbf{1}_{n\times n} + \mathbf{1}_{n\times n}X^k) \tag{21a}$$

$$X^{k+1} = \Pi_+\left(\tilde{X}^k\right). \tag{21b}$$

The first step in (21) minimizes the objective of $\mathcal{P}$ subject to the equality constraints, while the second projects the iterate to the nonnegative orthant. However, this approach is not guaranteed to solve $\mathcal{P}$ to optimality. For example, in the simple case of $C = \frac{1}{10}\begin{bmatrix} 1 & 9 & 9 \\ 9 & 1 & 0 \\ 9 & 0 & 9 \end{bmatrix}$, (21) converges to a suboptimal point $\bar{X}$ with $\left\|\bar{X} - X^*\right\|_\infty = 0.0\overline{7}$ where X^* is the optimizer. Nevertheless, it appears that, in general, (21) converges to a feasible point. However, even convergence to a feasible point can be much slower than our approach, as demonstrated in Figure 2.

Besides guaranteed convergence to the optimizer of $\mathcal{P}$, our approach can also handle sparsity in the affinity matrices. To demonstrate how exploiting sparsity can lead to significant speedups we consider the Spambase

Table 1: Normalizing correlation matrices with Algorithm 1 for spectral clustering on Spambase. A tolerance of 10^{-4} is used for the termination of our Algorithm. The Timings are expressed in seconds. and compared against Gurobi with its default settings (on $\mathcal{P}_1$) and against solving $C+\epsilon_{\mathrm{p}}\mathbf{1}_{n\times n}$ with the approach of §3.2 where and is $C(\sigma)$ the original affinity matrix. Hardware used: Intel i7-5557U CPU @ 3.10GHz, 8GB Memory.

σ	1.0	5.0	10.0	20.0
n_{nz}	3.9×10^4	1.8×10^6	4.0×10^6	7.3×10^6
t_{admm}	1.1×10^{-1}	5.7	1.5×10^1	2.6×10^1
t_{gurobi}	3.9×10^{-1}	4.6×10^1	1.1×10^2	2.0×10^2
$t^{\mathrm{dense}}_{\mathrm{admm}}$	7.7×10^2	6.9×10^2	6.4×10^2	7.2×10^2

dataset[4] (considered in Zass and Shashua (2007)) with an RBF kernel as an affinity criterion

$$C_{i,j}(\sigma) = e^{-\|x_i-x_j\|^2/\sigma^2}, \tag{22}$$

where σ is a parameter that is typically tuned to achieve the best clustering performance. Note that due to the exponential form of $C(\sigma)$, some values will be very small. Therefore, we truncate to zero all entries with value less than 10^{-7}. The runtimes of applying our Algorithm to this dataset are listed in Table 1 and compared to timings achieved with Gurobi (note that we use $\mathcal{P}_1$ for Gurobi as we consider $\mathcal{P}_1$ to be a "standard" reformulation of $\mathcal{P}$). We observe that exploiting sparsity can lead to significant speedup as compared to treating the affinity matrix as fully dense, even if we follow the approach of §3.2. At the same time, the optimizers of $\mathcal{P}$ for the affinity matrices $C(\sigma)$ considered in Table 1 appear to coincide with the ones for the fully dense $C(\sigma)+\epsilon_{\mathrm{p}}\mathbf{1}_{n\times n}$.

4.3 Matrices from the SuiteSparse Collection

Finally, we consider all Undirected Weighted Graph Matrices, with less than 50 million nonzeros, contained in the SuiteSparse collection[5]. 69 matrices meet these criteria. We use Algorithm 1 to normalize every matrix C so that all of its columns and rows sum to $\max_{i,j}(C_{i,j})$. All of the problems, except two, have nonnegative entries. For these two exceptions, we change the negative entries to their absolute value, as we are unaware of practical cases where C is expected to have negative entries.

Detailed comparison of our results with Gurobi presented in the Supplementary Material. Figure 3 shows the timings achieved by our method, and the speedups relative to Gurobi (on $\mathcal{P}_1$), as a function of each problem's nonzeros.

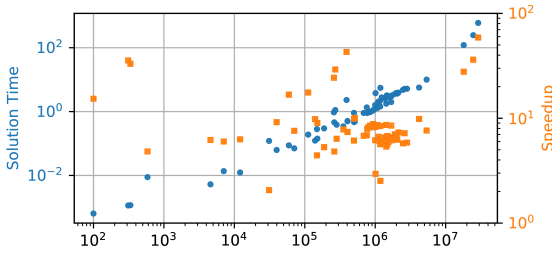

Figure 3: Results of Algorithm 1 for matrices from the SuiteSparse collection. Dots represent the timing of our Algorithm (with 10^{-4} tolerance), while squares represent the speedup achieved over Gurobi with its default settings (on $\mathcal{P}_1$). Hardware used: a single thread running on an Intel Gold 5120 with 192GB of memory.

5 Conclusions

In this paper we have shown that approximating doubly stochastic matrices, under a Frobenius distance metric, can be performed efficiently, even for very large, sparse matrices. We believe that our approach will complement very popular existing methods, such as the matrix balancing or Sinkhorn's algorithm, that solve the same problem under a different "distance" metric, thus giving practitioners more freedom in choosing the most suitable objective for this widely used approximation problem.

Acknowledgments This work was supported by the EPSRC AIMS CDT grant EP/L015987/1 and Schlumberger. We thank Giovanni Fantuzzi for valuable discussions.

References

J. D.. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011. ISSN 1935-8237.

M Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26*, pages 2292–2300. Curran Associates, Inc., 2013.

W. E. Deming and F. F. Stephan. On a least squares adjustment of a sampled frequency table when the

[4] `archive.ics.uci.edu/ml/datasets/spambase`
[5] Available at `sparse.tamu.edu`

expected marginal totals are known. *The Annals of Mathematical Statistics*, 11(4):427–444, 1940. ISSN 00034851.

P. Diggle, P. Heagerty, K. Y. Liang, and S. L. Zeger. *Analysis of longitudinal data.* Oxford statistical science series; 25. 2nd edition, 2002. ISBN 9780198524847.

J. Eckstein and D. P. Bertsekas. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992. ISSN 1436-4646.

M. Garstka, M. Cannon, and P. J. Goulart. COSMO: A conic operator splitting method for convex conic problems. arXiv 1901.10887, 2019.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 4th edition, 2013. ISBN 9781421407944.

Gurobi Optimization LLC. Gurobi Optimizer Reference Manual, 2018. URL `http://www.gurobi.com`.

N. J. Higham. Matrix nearness problems and applications. *Applications of Matrix Theory*, 1989.

M. Idel. A review of matrix scaling and Sinkhorn's normal form for matrices and positive maps. arXiv 1609.06349, 2016.

P. Knight. The Sinkhorn-Knopp Algorithm: Convergence and Applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008.

P. A. Knight and D. Ruiz. A fast algorithm for Matrix Balancing. *IMA Journal of Numerical Analysis*, 33 (3):1029–1047, 2013.

P. Milanfar. A Tour of Modern Image Filtering: New Insights and Methods, Both Practical and Theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, Jan 2013.

B O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.

H. Perfect and L. Mirsky. The Distribution of Positive Elements in Doubly-Stochastic Matrices. *Journal of the London Mathematical Society*, s1-40(1):689–698, 1965.

S. Rao, M. Huntley, N. Durand, E. Stamenova, I. Bochkov, J. Robinson, A. Sanborn, I. Machol, A. Omer, E. Lander, and E. Aiden. A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. *Cell*, 159(7): 1665–1680, 2014. ISSN 0092-8674.

B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: An Operator Splitting Solver for Quadratic Programs. *ArXiv e-prints*, 2017.

E. Yaffe and A. Tanay. Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture. *Nature Genetics*, 43(11):1059, 2011. ISSN 1061-4036.

R. Zass and A. Shashua. Doubly stochastic normalization for spectral clustering. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1569–1576. MIT Press, 2007.