STRUCTURALLY-CONSTRAINED GRADIENT DESCENT FOR MATRIX FACTORIZATION IN HAPLOTYPE ASSEMBLY PROBLEMS

Changxiao Cai

Tsinghua University
Department of Electronic Engineering
Beijing, China

Sujay Sanghavi, Haris Vikalo

The University of Texas at Austin
Department of Electrical and Computer Engineering
Austin, TX, USA

ABSTRACT

In matrix decomposition problems, one often seeks to represent a data matrix by the product of two matrices – one capturing meaningful information contained in the data and the other specifying how this information is combined to generate the data matrix. We consider matrix decomposition that arises in haplotype assembly, an important problem in genomics. The observed matrix contains noisy samples of the product of an informative matrix with rows having entries from a finite alphabet and a matrix with rows that are standard unit basis. Structurally-constrained gradient descent algorithm for finding the two aforementioned matrices is proposed and its convergence is analyzed. Simulation results demonstrate superior accuracy and speed of the proposed method compared to state-of-the-art haplotype assembly techniques.

Index Terms— Matrix factorization, low-rank, gradient descent, haplotype assembly

1. INTRODUCTION

Finding a low-rank approximation to a partially observed matrix has gained a lot of attention in recent years (e.g., see [1, 2, 3, 4] and the references therein). This line of research has been motivated by a large number of applications, including by now classical collaborative filtering problem where the goal is to infer preference of users for unrated items based on a limited number of rankings (as in, e.g., Netflix problem [5]). In many scenarios, it is of interest to represent the rank-k matrix $\mathbf{M} \in \mathcal{R}^{n\times m}$ as $\mathbf{M} = \mathbf{U}\mathbf{V}^T$ where $\mathbf{U} \in \mathcal{R}^{n\times k}$ and $\mathbf{V} \in \mathcal{R}^{m\times k}$. Examples include applications to clustering [6] and sparse PCA [7]. This bi-linear parametrization of the unknown matrix $\mathbf{M}$ leads to the problem of finding $\mathbf{U}$ and $\mathbf{V}$ such that a chosen performance metric (e.g., the Frobenius norm of the difference between the partial, noisy observations of $\mathbf{M}$ and $\mathbf{U}\mathbf{V}^T$) is optimized. The bi-linearity of the representation renders the problem non-convex and, therefore, challenging. Among the often used heuristics, alternating minimization where one keeps either $\mathbf{U}$ or $\mathbf{V}$ fixed and optimizes over the other, has gained popularity [8, 4]. This is due to the fact that each of the two subproblems is convex and hence can be solved in a computationally efficient manner.

At the same time, there has been a surge of interest in DNA sequencing and studies of genetic variations that it enables. High-throughput sequencing platforms often employ so-called shotgun sequencing strategy and oversample the target genome with a library of relatively short overlapping reads. Each read in this library provides information about a subsequence of the chromosome from which the read is sampled. In the haplotype assembly applications, the reference genome is typically known and therefore a read can be mapped to the reference (i.e., relative positions of the reads with respect to the reference can be established). A read that stretches across multiple SNPs of a chromosome may be used to assemble the haplotype associated with that chromosome. Recent sequencing technologies are capable of generating pairs of reads that are separated by inserts of unknown content but known length. These so-called paired-end reads help connect information across large distances of a chromosome.

Sequencing is erroneous, which leads to ambiguities regarding the origin of a read and therefore renders the haplotype assembly challenging. Recent haplotype assembly methods focus on the minimum error correction (MEC) criterion where the goal is to find the smallest number of nucleotides in reads that need to be changed so that any read partitioning ambiguities would be resolved. It has been shown that finding optimal solution to the MEC formulation of the haplotype assembly problem is NP-hard [9, 10]. In [11], the authors used a computationally intensive branch-and-bound scheme to minimize the MEC objective over the space of reads. High complexity of the exact solution has motivated several heuristics including the one in [12], where a greedy algorithm was used to assemble haplotypes of the first complete diploid individual genome. A max-cut formulation of the haplotype assembly problem was proposed in [13], and an efficient algorithm (HapCUT) that solves it and significantly outperforms the method in [12] was developed. The Bayesian methods relying on MCMC and Gibbs sampling schemes were proposed in [14] and [15], respectively. A greedy cut approach was pro-

This work was supported in part by the National Science Foundation CCF-1320273.

posed in [16], and a flow-graph based approach in [17], and, most recently, maximum-likelihood scheme in [18].

In this paper, we formulate haplotype assembly as the partially observed low rank matrix factorization problem and propose a variant of the gradient descent algorithm to solve it at low computational cost. The algorithm explicitly imposes constraints on the special structure of the matrix $\mathbf{U}$ in factorization $\mathbf{M} = \mathbf{U}\mathbf{V}^T$ that is inherent to the problem.

2. MATHEMATICAL MODEL

Prior to haplotype assembly, one needs to infer the order of nucleotides in reads (so-called base calling [19, 20]), align reads to a reference, and perform SNP and genotype calling. SNPs occur at a relatively low frequency, e.g., 1 polymorphism in 1000 nucleotides. Segments of the reads which do not cover any SNP locations are discarded. Furthermore, a read covering only a single SNP position does not help in the process of inferring a haplotype and is hence discarded as well. The remaining n reads (more precisely, the segments of n reads bearing information relevant for haplotype assembly) are organized into an $n \times m$ SNP fragment matrix $\mathbf{R}$, where m denotes the haplotype length. The i^{th} row of $\mathbf{R}$, $\mathbf{r}_i$, consists of the haplotype-relevant information provided by the i^{th} read. In humans and other diploid organisms, SNP sites are bi-allelic (they are often so in polyploid ones as well) – this means only two out of four possible nucleotides A, C, G or T are possible to find in any SNP position. Therefore, we can label the nucleotides in SNP positions using binary symbols $\{1, -1\}$ where the mapping between letters and binary symbols at any position follows arbitrary convention. For convenience, entries in $\mathbf{r}_i$ that do not provide any SNP information are labeled by 0. After such a labeling, the resulting matrix $\mathbf{R}$ consists of ternary $\{-1, 0, 1\}$ entries. Specifically, the (i, j) entry of $\mathbf{R}$ is the information about the j^{th} SNP site provided by the i^{th} read; if the i^{th} read does not cover the j^{th} SNP site, the (i, j) entry of $\mathbf{R}$ is $R_{ij} = 0$.

Let $\mathcal{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_k\}$ denote the set of haplotype sequences of a k-ploid organism. It is convenient to introduce a projector operator $P_\Omega(\cdot)$ defined as

$$P_\Omega(\mathbf{M}) = \begin{cases} M_{ij}, & \text{if } (i,j) \in \Omega, \\ 0, & \text{otherwise,} \end{cases}$$

where Ω denotes the set of indices (i, j) such that $R_{ij} \neq 0$ (i.e., R_{ij} is an informative entry of $\mathbf{R}$). Therefore, $P_\Omega(\mathbf{M})$ is an operator that describes how sequencing reads, each read corresponding to a row of $\mathbf{R}$, sample the haplotypes. For instance, $R_{ij} = -1$, $(i, j) \in \Omega$ implies that the i^{th} read covers the j^{th} SNP positions and provides information encoded as "-1"; it is unknown, however, which of the k haplotypes is sampled by the i^{th} read. In general, matrix $\mathbf{R}$ can be thought of as being obtained by sampling, with errors, a low-rank $n \times m$ matrix $\mathbf{M}$,

$$\mathbf{M} = \mathbf{U}\mathbf{V}^T,$$

where $\mathbf{U}$ and $\mathbf{V}$ are $n \times k$ and $m \times k$ matrices, respectively, and where k denotes the ploidy (the number of haplotypes) of an organism.[1] The j^{th} column of $\mathbf{V}$, $\mathbf{v}_j$, is the sequence of the i^{th} haplotype, i.e., $\mathbf{v}_j = \mathbf{h}_j \in \mathcal{H}$. The i^{th} row of $\mathbf{U}$, $\mathbf{u}_i$, is the indicator of the origin of the i^{th} read. More specifically, the rows of $\mathbf{U}$ are the k-dimensional standard unit vectors consisting of all 0's except for one entry which is equal to 1. For instance, $\mathbf{u}_i = \mathbf{e}_l$ indicates that the i^{th} read is obtained by sampling the l^{th} chromosome/haplotype. Note that each row of the (unobservable) matrix $\mathbf{M}$, $\mathbf{m}_i$, is a full haplotype sequence (i.e., $\mathbf{m}_i \in \mathcal{H}$).

DNA sequencing is erroneous and thus $P_\Omega(\mathbf{R}) \neq P_\Omega(\mathbf{M})$. We assume the model where the entries in $\mathbf{R}$ are perturbed versions of the corresponding entries in $\mathbf{M}$, i.e., the $(i, j) \in \Omega$ entry in $\mathbf{R}$, R_{ij}, is obtained as

$$R_{ij} = \begin{cases} M_{ij}, & \text{w.p.} \quad 1-p, \\ -M_{ij}, & \text{w.p.} \quad p, \end{cases}$$

where p denotes the sequencing/genotyping error rate.

3. STRUCTURALLY-CONSTRAINED GRADIENT DESCENT

Given the SNP fragment matrix $\mathbf{R}$, the haplotype assembly task can be solved by performing the low-rank matrix factorization $\mathbf{M} = \mathbf{U}\mathbf{V}^T$ of the unobservable matrix $\mathbf{M}$ from its noisy sample with missing entries, $\mathbf{R}$. This can be done in a computationally efficient manner by relying on, e.g., gradient descent. Define the objective function

$$f(\mathbf{U}, \mathbf{V}) = \|P_\Omega(\mathbf{R} - \mathbf{U}\mathbf{V}^T)\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of its argument. We would like to find $\mathbf{U}$ and $\mathbf{V}$ that minimize $f(\mathbf{U}, \mathbf{V})$ in (1). Let $\mathbf{U}_0$ and $\mathbf{V}_0$ denote the initial guesses of $\mathbf{U}$ and $\mathbf{V}$, respectively. Gradient descent search iteratively updates estimates of $\mathbf{U}_0$ and $\mathbf{V}_0$ in the direction of the respective derivatives. However, the conventional gradient descent algorithm does not exploit the special structure of matrix $\mathbf{U}$ – in particular, it ignores the fact that the rows of $\mathbf{U}$ are standard unit vectors which may have detrimental effects on the accuracy of the method. To enforce the structure of $\mathbf{U}$, we perform iterations

$$\mathbf{V}_{t+1} = \mathbf{V}_t - \alpha \nabla f(\mathbf{V}_t) \tag{2}$$

and

$$\mathbf{U}_{t+1} = \arg\min_{\mathbf{u}_i \in \Phi} f(\mathbf{U}, \mathbf{V}_{t+1}), \tag{3}$$

where the optimization in (3) is done by exhaustively searching over k vectors in $\Phi = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k\}$ to find the most likely $\mathbf{U}_{t+1}$. Since the number of haplotypes k is relatively small (typically, $k \leq 6$), the complexity of the exhaustive search (3) is very low. This modified gradient descent is formalized below as Algorithm 1.

[1] In the diploid ($k = 2$) case, $\mathbf{V}$ drops the rank since $\mathbf{v}_1 = -\mathbf{v}_2$ and thus $\mathbf{M}$ is rank-1. In the k-ploid case ($k > 2$), the rank of $\mathbf{V}$ (and $\mathbf{M}$) is k.

Algorithm 1 Structurally-Constrained Gradient Descent

Input: The SNP matrix $\mathbf{R}$

Initialization: Use power iteration method to generate k left-singular vectors $\mathbf{U}_0$ and right-singular vectors $\mathbf{V}_0$.
$\Phi = \{(1, 0, \cdots, 0), (0, 1, 0, \cdots, 0), \cdots, (0, 0, \cdots, 1)\}$.

repeat

$\nabla f(\mathbf{V}_t) = -2(P_\Omega(\mathbf{R} - \mathbf{U}_t\mathbf{V}_t^T))^T\mathbf{U}_t$

$\mathbf{V}_{t+1} = \mathbf{V}_t - \alpha\nabla f(\mathbf{V}_t)$

$\mathbf{U}_{t+1} = \arg\min_{u_i \in \Phi} \sum_{(i,j)\in\Omega} \|P_\Omega(\mathbf{R} - \mathbf{U}_t\mathbf{V}_{t+1}^T)\|_F^2$

until termination criterion is met.

Output: An estimate of the haplotype matrix $\mathbf{V}$ generated by quantizing entries of the most recent iteration $\mathbf{V}_{t_{max}}$ to ± 1.

It can be shown (the proof omitted for brevity) that the above algorithm converges if the step size is selected as

$$\alpha = C\frac{\|\nabla f(V_t)^T\|_F^2}{\|P_\Omega(U_t\nabla f(V_t)^T)\|_F^2},$$

where $C \in (0, 1)$ is a constant.

4. RESULTS

To obtain numerical results in this section, we implemented our algorithms in Matlab and ran the codes on a single core processor laptop (2.7 GHz Intel Core i5, 8GB RAM).

When the ground truth is known, as in simulation studies, the ability of an algorithm to reconstruct a haplotype may be measured by the reconstruction rate. In the case of diploids, the reconstruction rate is conveniently defined as [21]

$$R_r = 1 - \frac{\min(D(\mathbf{h}^1, \hat{\mathbf{h}}^1) + D(\mathbf{h}^2, \hat{\mathbf{h}}^2), D(\mathbf{h}^1, \hat{\mathbf{h}}^2) + D(\mathbf{h}^2, \hat{\mathbf{h}}^1))}{2m}$$

where $D(\mathbf{h}^i, \hat{\mathbf{h}}^j) = \sum_{l=1}^{m} d(h_l^i, \hat{h}_l^j)$ denotes the generalized Hamming distance between $\mathbf{h}^i$ and $\hat{\mathbf{h}}^j$, $(\mathbf{h}^1, \mathbf{h}^2)$ is the pair of true haplotypes, and $(\hat{\mathbf{h}}^1, \hat{\mathbf{h}}^2)$ is the pair of reconstructed haplotypes.

Using the data sets from [21], we compare performance of our Algorithm 1 with several existing haplotype assembly methods and report the results in Table 1. Specifically, we show the comparison of the achieved reconstruction rates with those of SpeedHap, FastHare, 2d-med, MLF and SHR-tree, algorithms benchmarked and discussed in [21]. As evident from the table, structurally-constrained gradient descent outperforms all the competing methods. Note that the percentage of observed entries in $\mathbf{R}$ in Table 1 is 1%.

We further tested the performance of our proposed algorithm on the experimental data generated as part of the *1000 Genomes Project*, an international study meant to provide a detailed map of human genetic variation. The MEC score (serving as a proxy for the reconstruction rate) and the runtimes of the structurally constrained gradient search and the recently proposed HapTree [18] are shown in Table 2 for the first 5 human chromosomes. As can be seen there, our proposed algorithm achieves better accuracy and significant improvement in speed as compared to the competing scheme.

Table 2: *A comparison of our Algorithm 1 and HapTree.*

chr	GD (Algorithm 2)		HapTree	
	MEC	Time(s)	MEC	Time(s)
1	1300	3.35	1479	921.76
2	1763	4.84	1793	1908.00
3	1434	4.27	1610	920.13
4	1663	6.74	1840	950.73
5	1330	4.37	1488	829.37

5. CONCLUSION

We studied the problem of reconstructing haplotypes using high-throughput DNA sequencing reads. To this end, we proposed a novel formulation of the problem as the one of factorizing a partially observed low rank matrix. Each row of the matrix corresponds to a sequencing read; the read is aligned to the reference and spans only those columns associated with single nucleotide polymorphisms covered by the read. Since the reads are much shorter than the haplotype blocks, most of the entries in each row of the data matrix are missing – hence the matrix is only partially observed. Moreover, each row can be thought of as being sampled from one among few haplotype sequences – therefore, the matrix is low rank. Finally, since the sequencing is erroneous, the observed matrix contains incorrect entries. We developed a structurally-constrained gradient search algorithm that imposes the special structure of the matrices in the decomposition. Performance of the proposed algorithm was extensively tested, demonstrating its superiority in terms of both accuracy and speed over competing haplotype assembly schemes.

6. REFERENCES

[1] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 2, pp. 713–730, July 2008.

[2] E. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[3] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

Table 1: *Reconstruction rates for several haplotype assembly algorithms on diploid data.*

Data error rate	Coverage	Gradient descent (Algorithm 1)	SPH	FAST	2d	MLF	SHR
0.1	3	0.8686	0.7047	0.8295	0.7860	0.6982	0.6679
0.1	5	0.9513	0.9471	0.9408	0.8805	0.8094	0.7158
0.1	8	0.9965	0.9848	0.9859	0.9483	0.8632	0.7429
0.1	10	0.9986	0.9861	0.9955	0.9649	0.8839	0.7260
0.2	3	0.6770	0.1990	0.6518	0.6468	0.6240	0.5913
0.2	5	0.7850	0.6810	0.7118	0.6969	0.6820	0.6171
0.2	8	0.8992	0.8006	0.8078	0.7512	0.7475	0.6529
0.2	10	0.9340	0.8127	0.8719	0.7780	0.7650	0.6748

[4] P. Netrapalli, P. Jain, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Symposium on Theory of Computing (STOC)*, 2013.

[5] "Netflix prize." [Online]. Available: http://www.netflixprize.com/.

[6] J. Kim and H. Park, "Sparse nonnegative matrix factorization for clustering," Technical Report GT-CSE-08-01, Georgia Institute of Technology, Tech. Rep., 2008.

[7] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 262–286, 2006.

[8] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[9] G. Lancia, V. Bafna, S. Istrail, R. Lippert, and R. Schwartz, "Snps problems, complexity, and algorithms," in *Algorithms—ESA 2001*. Springer, 2001, pp. 182–193.

[10] R. Cilibrasi, L. Van Iersel, S. Kelk, and J. Tromp, "On the complexity of several haplotyping problems," in *Algorithms in Bioinformatics*. Springer, 2005, pp. 128–139.

[11] R.-S. Wang, L.-Y. Wu, Z.-P. Li, and X.-S. Zhang, "Haplotype reconstruction from snp fragments by minimum error correction," *Bioinformatics*, vol. 21, no. 10, pp. 2456–2462, 2005.

[12] S. Levy, G. Sutton, P. C. Ng, L. Feuk, A. L. Halpern, B. P. Walenz, N. Axelrod, J. Huang, E. F. Kirkness, G. Denisov *et al.*, "The diploid genome sequence of an individual human," *PLoS biology*, vol. 5, no. 10, p. e254, 2007.

[13] V. Bansal and V. Bafna, "Hapcut: an efficient and accurate algorithm for the haplotype assembly problem," *Bioinformatics*, vol. 24, no. 16, pp. i153–i159, 2008.

[14] V. Bansal, A. L. Halpern, N. Axelrod, and V. Bafna, "An MCMC algorithm for haplotype assembly from whole-genome sequence data," *Genome research*, vol. 18, no. 8, pp. 1336–1346, 2008.

[15] J. H. Kim, M. S. Waterman, and L. M. Li, "Diploid genome reconstruction of ciona intestinalis and comparative analysis with ciona savignyi," *Genome research*, vol. 17, no. 7, pp. 1101–1110, 2007.

[16] J. Duitama, G. K. McEwen, T. Huebsch, S. Palczewski, S. Schulz, K. Verstrepen, E.-K. Suk, and M. R. Hoehe, "Fosmid-based whole genome haplotyping of a hapmap trio child: evaluation of single individual haplotyping techniques," *Nucleic acids research*, p. gkr1042, 2011.

[17] D. Aguiar and S. Istrail, "Hapcompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data," *Journal of Computational Biology*, vol. 19, no. 6, pp. 577–590, 2012.

[18] E. Berger, D. Yorukoglu, J. Peng, and B. Berger, "Haptree: A novel bayesian framework for single individual polyplotyping using ngs data," *PLoS Computational Biology*, vol. 10, no. 3, 2014.

[19] S. Das and H. Vikalo, "Onlinecall: fast online parameter estimation and base calling for illumina's next-generation sequencing," *Bioinformatics*, vol. 28, no. 13, pp. 1677–83, 2012.

[20] ——, "Base calling for high-throughput short-read sequencing: Dynamic programming solutions," *BMC Bioinformatics*, vol. 14, no. 129, 2013.

[21] F. Geraci, "A comparison of several algorithms for the single individual snp haplotyping reconstruction problem," *Bioinformatics*, vol. 26, no. 18, pp. 2217–2225, 2010.