

# Chapter 7

## A Global Optimization Algorithm for Sparse Mixed Membership Matrix Factorization



Fan Zhang, Chuangqi Wang, Andrew C. Trapp, and Patrick Flaherty

### 7.1 Introduction

Mixed membership matrix factorization (MMMF) has been used in document topic modeling (Blei et al. 2003), collaborative filtering (Mackey et al. 2010), population genetics (Pritchard et al. 2000), and social network analysis (Airoldi et al. 2008). The underlying assumption is that an observed feature for a given sample is a mixture of shared, underlying groups. These groups are called topics in document modeling, subpopulations in population genetics, and communities in social network analysis; in bioinformatics applications the groups are called subtypes and we adopt that terminology here. MMMF simultaneously identifies

---

F. Zhang

Center for Data Sciences at Brigham and Women's Hospital, Boston, MA, USA

Broad Institute of Massachusetts Institute of Technology and Harvard University, Boston, MA, USA

Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA

e-mail: [fanzhang@broadinstitute.org](mailto:fanzhang@broadinstitute.org)

C. Wang

Department of Biomedical Engineering, Worcester Polytechnic Institute, Worcester, MA, USA

e-mail: [cwang7@wpi.edu](mailto:cwang7@wpi.edu)

A. C. Trapp

Robert A. Foisie Business School, Worcester Polytechnic Institute, Worcester, MA, USA

e-mail: [atrapp@wpi.edu](mailto:atrapp@wpi.edu)

P. Flaherty (✉)

Department of Mathematics & Statistics, University of Massachusetts Amherst, Amherst, MA, USA

e-mail: [flaherty@math.umass.edu](mailto:flaherty@math.umass.edu)

© Springer Nature Switzerland AG 2019

L. Zhang et al. (eds.), *Contemporary Biostatistics with Biopharmaceutical*

*Applications*, ICSA Book Series in Statistics,

[https://doi.org/10.1007/978-3-030-15310-6\\_7](https://doi.org/10.1007/978-3-030-15310-6_7)

both the underlying subtypes and the distribution over those subtypes for each individual sample.

### ***7.1.1 Mixed Membership Models***

The MMMF problem can be viewed as inference in a particular statistical model (Singh and Gordon 2008). The model typically has a latent Dirichlet random variable that allows each sample to have its own distribution over subtypes and a latent variable for the feature weights that describe each subtype. The inferential goal is to estimate the joint posterior distribution over these latent variables and thus obtain the distribution over subtypes for each sample and the feature vector for each subtype. Non-negative matrix factorization techniques have been used in image analysis and collaborative filtering applications (Lee and Seung 1999; Mackey et al. 2010). Topic models for document clustering have also been cast as a matrix factorization problem (Xu et al. 2003).

The basic mixed membership model structure has been extended in various interesting ways. A hierarchical Dirichlet prior allows one to obtain a posterior distribution over the number of subtypes (Teh et al. 2005). A prior on the subtype variables allows one to impose specific sparsity constraints on the subtypes (Kabán 2007; MacKay 1992; Taddy 2013). Correlated information may be incorporated to improve the coherence of the subtypes (Blei and Lafferty 2006). Gaussian-Laplace-Dirichlet Model (GLAD) is hierarchical model that performs mixed membership matrix factorization with sparsity inducing Laplace prior on feature weights (Sad-diki et al. 2015).

Sampling or variational inference methods are commonly used to estimate the posterior distribution of interest for mixed membership models, but these only provide local or approximate estimates. A mean-field variational algorithm (Blei et al. 2003) and a collapsed Gibbs sampling algorithm have been developed for Latent Dirichlet Allocation (Xiao and Stibor 2010). However, Gibbs sampling is approximate for finite chain lengths and variational inference is only guaranteed to converge to a local optimum (Blei et al. 2017).

### ***7.1.2 Benders' Decomposition and Global Optimization (GOP)***

In many applications it is important to obtain a globally optimal solution rather than a local or approximate solution. Recently, there have been significant advances in deterministic optimization methods for general biconvex optimization problems (Floudas and Gounaris 2008; Horst and Tuy 2013). Here, we show that mixed membership matrix factorization can be cast as a biconvex optimization problem and the  $\epsilon$ -global optimum can be obtained by these deterministic optimization methods.

Benders' decomposition exploits the idea that in a given optimization problem there are often *complicating variables*—variables that when held fixed yield a much simpler problem over the remaining variables (Benders 1962). Benders developed a cutting plane method for solving mixed integer optimization problems that can be so decomposed. Geoffrion later extended Benders' decomposition to situations where the primal problem (parametrized by fixed complicating variable values) no longer needs to be a linear program (Geoffrion 1972). The Global OPTimization (GOP) approach is an adaptation of the original Benders' decomposition that can handle a more general class of problems that includes mixed-integer biconvex optimization problems (Floudas 2013). Here, we exploit the GOP approach for solving a particular mixed membership matrix factorization problem.

### 7.1.3 Contributions

Our contribution is bringing the Global OPTimization (GOP) algorithm into contact with the mixed membership matrix factorization problem, computational improvements to the branch-and-bound GOP algorithm, and experimental results. Our discussion of the GOP algorithm here is necessarily brief. The details of problem conditions, convergence properties, and a full outline of the algorithm steps for the branch-and-bound version of the algorithm are found elsewhere (Floudas 2013).

We outline the general sparse mixed membership matrix factorization problem in Sect. 7.2. In Sect. 7.3, we use GOP to obtain an  $\epsilon$ -global optimum solution for the mixed membership matrix factorization problem. In Sect. 7.4, we develop an A-star search algorithm that significantly improves the computational efficiency of our method. In Sect. 7.5, we show empirical accuracy and convergence time results on a synthetic data set. We also explore the performance of our algorithm on a small gene expression data set. Finally, we discuss further computational and statistical issues in Sect. 7.6.

## 7.2 Problem Formulation

The problem data is a matrix  $y \in \mathbb{R}^{M \times N}$ , where an element  $y_{ji}$  is an observation of feature  $j$  in sample  $i$ . We would like to represent each sample as a convex combination of  $K$  subtype vectors,  $y_i = x\theta_i$ , where  $x \in \mathbb{R}^{M \times K}$  is a matrix of  $K$  subtype vectors and  $\theta_i$  is the mixing proportion of each subtype. We would like  $x$  to be sparse because doing so makes interpreting the subtypes easier and often  $x$  is believed to be sparse *a priori* for many interesting problems. In the specific case of cancer subtyping,  $y_{ji}$  may be a normalized gene expression measurement for gene  $j$  in sample  $i$ . We write this matrix factorization problem as

$$\begin{aligned}
& \underset{\theta, x}{\text{minimize}} && \|y_i - x\theta_i\|_2^2 \\
& \text{subject to} && \|x\|_1 \leq P \\
& && \theta_i \in \Delta^{K-1} \quad \forall i,
\end{aligned} \tag{7.1}$$

where  $\Delta^{K-1}$  is a  $K$ -dimensional simplex.

Optimization problem (7.1) can be recast with a biconvex objective and a convex domain as

$$\begin{aligned}
& \underset{\theta, x, z}{\text{minimize}} && \|y - x\theta\|_2^2 \\
& \text{subject to} && \sum_{j=1}^M \sum_{k=1}^K z_{jk} \leq P, \\
& && -z_{jk} \leq x_{jk} \leq z_{jk} \quad \forall (j, k), \\
& && \theta_i \in \Delta^{K-1} \quad \forall i, \\
& && z_{jk} \geq 0 \quad \forall (j, k)
\end{aligned} \tag{7.2}$$

If either  $x$  or  $\theta$  is fixed then (7.2) reduces to a convex optimization problem. Indeed, if  $x$  is fixed, the optimization problem is a form of constrained linear regression. If  $\theta$  is fixed, we have a form of LASSO regression. We prove that (7.1) is a biconvex problem in Appendix 2. Since both problems are computationally simple, we could take either  $x$  or  $\theta$  to be the complicating variables in Benders' decomposition and we choose  $\theta$ .

A common approach for solving an optimization problem with a nonconvex objective function is to alternate between fixing one variable and optimizing over the other. However, this approach only provides a local optimum (Gorski et al. 2007). A key to the GOP algorithm is the Benders'-based idea that feasibility and optimality information is shared between the primal problems in the form of constraints.

### 7.3 Algorithm

The Global OPTimization (GOP) algorithm, which we describe here, solves for  $\epsilon$ -global optimum values of  $x$  and  $\theta$  (Floudas and Visweswaran 1990; Floudas 2000, 2013). The algorithm proceeds by first partitioning the optimization problem decision variables into complicating and non-complicating variables. Then, the GOP algorithm alternates between solving a *primal problem* over  $\theta$  for fixed  $x$ , and solving a *relaxed dual problem* over  $x$  for fixed  $\theta$ . The primal problem provides an upper bound on the original optimization problem because it contains more constraints than the original problem ( $x$  is fixed). The relaxed dual problem contains

fewer constraints and forms a valid global lower bound. The algorithm iteratively tightens the upper and lower bounds on the global optimum by alternating between the primal and relaxed dual problem.

### 7.3.1 Initialization

The algorithm starts by partitioning the problem into a relaxed dual problem and a primal problem. The solution of the relaxed dual problem is an optimal  $x$  for fixed values of the complicating variables  $\theta$  and the solution of the primal problem is an optimal  $\theta$ . An iteration counter  $T = 1$  is initialized.

For each iteration, the relaxed dual problem is solved by forming a partition of the domain of  $x$  and solving a relaxed dual subproblem for each subset. A branch-and-bound tree data structure is used to store the solution of each of these relaxed dual subproblems and we initialize the root node  $n(0)$  where  $T = 0$ . The parents of  $n(T)$  is denoted  $\text{par}(n(T))$ , the set of ancestors of  $n(T)$  is denoted  $\text{anc}(n(T))$ , and the set of children of  $n(T)$  is denoted  $\text{ch}(n(T))$ . The root node is formed by initializing  $x$  at a random feasible point,  $x^{n(0)}$ , and storing it in  $n(0)$ .

### 7.3.2 Solve Primal Problem and Update Upper Bound

The primal problem (7.2) is constrained to a fixed value of  $x$  at  $n(T)$ ,  $x^{(n(T))}$ ,

#### Primal problem

( $x$  fixed)

$$\begin{aligned} & \underset{\theta}{\text{minimize}} && \|y - x\theta\|_2^2 \\ & \text{subject to} && \theta_i^T \mathbf{1}_K = 1 \quad \text{for all } i, \\ & && \theta_{ki} \geq 0 \quad \text{for all } k, i \end{aligned} \tag{7.3}$$

Since the primal problem is more constrained than (7.2), the solution,  $S^{(n(T))}$ , is a global upper bound. The value of the upper bound is  $\text{PUBD} \leftarrow \min\{\text{PUBD}, S^{(n(T))}\}$ , so PUBD holds the tightest upper bound across iterations.

### 7.3.3 Solve the Relaxed Dual Problem and Update Lower Bound

The relaxed dual problem is a relaxed version of (7.2) in that it contains fewer constraints than the original problem. Initially, at the root node,  $n(0)$ , the domain

of the relaxed dual problem is the entire domain of  $x$ ,  $\mathcal{X}$ . Each node stores a set of linear constraints (cuts) such that when all of the constraints are satisfied, they define a region in  $\mathcal{X}$ . Sibling nodes form a partition of parent's region and a node deeper in the tree defines a smaller region than shallower nodes when incorporating the constraints of the node and all of its ancestors. These partitioning constraints are called *qualifying constraints*. Since the objective function is convex in  $\theta$  for a fixed value of  $x$ , a Taylor series approximation of the Lagrangian with respect to  $\theta$  provides a valid lower bound on the objective function. Since the objective function is convex in  $\theta$ , the Taylor approximation is linear and the optimal objective is at a bound of  $\theta$ . The GOP algorithm as outlined in (Floudas and Gounaris 2008) makes these ideas rigorous.

The relaxed dual problem for the mixed membership matrix factorization problem (7.2) for a node  $n(T)$  is below.

**Relaxed Dual Problem**  
( $\theta$  fixed)

$$\begin{aligned}
 & \text{minimize} && Q \\
 & && Q, x, z \\
 & \text{subject to} && \sum_{j=1}^M \sum_{k=1}^K z_{jk} \leq P, \\
 & && -z_{jk} \leq x_{jk} \leq z_{jk}, \quad z_{jk} \geq 0, \\
 & && L(x, \theta^B(t), y, \lambda^t, \mu^t) \Big|_{x^t, \theta^t}^{\text{lin}} \leq Q \quad \text{for } t \in \{\text{anc}(n(T)), n(T)\}, \\
 & && g_{ki}^t \Big|_{x^t}^{\text{lin}}(x) \leq 0 \quad \text{if } \theta^B(t)_{ki} = 1, \\
 & && g_{ki}^t \Big|_{x^t}^{\text{lin}}(x) \geq 0 \quad \text{if } \theta^B(t)_{ki} = 0
 \end{aligned} \tag{7.4}$$

The function  $L(x, \theta^B(t), y, \lambda^t, \mu^t) \Big|_{x^t, \theta^t}^{\text{lin}}$  is the linearized Lagrangian of (7.2),  $g_{ki}^t \Big|_{x^t}^{\text{lin}}(x)$  is the  $ki$ -th qualifying constraint, and  $\theta^B(t)$  is the value of  $\theta$  at the bound such that the linearized Lagrangian is a valid lower bound in the region defined by the qualifying constraints at node  $t$ . We have taken a second Taylor approximation with respect to  $x$  to ensure the qualifying constraints are linear in  $x$  and thus valid cuts as recommended in (Floudas and Gounaris 2008).

The algorithm for solving the relaxed dual problem comprises five steps:

1. Construct a child node in the branch-and-bound tree
2. Populate the child node with the linearized Lagrange function and qualifying constraints

3. Solve the relaxed dual subproblem at the child nodes
4. Update the lower bound
5. Check convergence

### 7.3.3.1 Construct a Child Node in the Branch-and-Bound Tree

Recall, a unique region in  $\mathcal{X}$  for the leaf node  $\text{ch}(n(T))$  is defined by the  $t$ -th row of  $\theta^B$  derived from the primal problem at node  $n(T)$ . This region can be expressed as the qualifying constraint set,

$$g_{ki}^{\text{ch}(n(T))} \Big|_{x^{n(T)}}^{\text{lin}}(x) \leq 0 \text{ if } \theta_{ki}^B(t) = 1,$$

$$g_{ki}^{\text{ch}(n(T))} \Big|_{x^{n(T)}}^{\text{lin}}(x) \geq 0 \text{ if } \theta_{ki}^B(t) = 0.$$

To generate the  $t$ th child node of  $n(T)$  and populate it with this constraint set and  $\theta^B(t)$  which will be used in the construction of the Lagrange function lower bound in the relaxed dual problem.

### 7.3.3.2 Populate the Child Node with the Linearized Lagrange Function and Qualifying Constraints

The qualifying constraint sets contained in each node along the path in the branch-and-bound tree from  $\text{ch}(n(T))$  to the root, inclusively, are added to the relaxed dual subproblem at the newly constructed child node. For example, the qualifying constraint set for a node  $n'$  along the path is

$$g_{ki}^{n'} \Big|_{x^{n'}}^{\text{lin}}(x) \leq 0 \text{ if } \theta^B(n')_{ki} = 1$$

$$g_{ki}^{n'} \Big|_{x^{n'}}^{\text{lin}}(x) \geq 0 \text{ if } \theta^B(n')_{ki} = 0,$$

where  $g_{ki}^{n'}$  is the node's  $ki$ th qualifying constraint,  $x^{n'}$  is the node's relaxed dual problem optimizer, and  $\theta^B(n')$  is a 0-1 vector defining the unique region for node  $n'$  since  $\theta_{ki} \in [0, 1]$ .

Then, the Lagrangian function lower bound constraints from each node along the path in the branch-and-bound tree from  $\text{ch}(n(T))$  to the root, inclusively, are added to the relaxed dual subproblem. For example the linearized Lagrange function for node  $n'$ ,

$$L(x, \theta^B(n'), y, \lambda^{(n')}, \mu^{(n')}) \Big|_{x^{(n')}, \theta^{(n')}}^{\text{lin}}.$$

The Lagrangian function for the primal problem is

$$\begin{aligned}
 L(x, \theta, \lambda, \mu) &= \sum_{i=1}^N L(x, \theta_i, \lambda_i, \mu_i) \\
 &= \sum_{i=1}^N (y_i - x\theta_i)^\top (y_i - x\theta_i) \\
 &\quad - \lambda_i (\theta_i^\top \mathbf{1}_K - 1) - \mu_i^\top \theta_i \\
 &= \sum_{i=1}^N y_i^\top y_i - 2y_i^\top x\theta_i + \theta_i^\top x^\top x\theta_i \\
 &\quad - \lambda_i (\theta_i^\top \mathbf{1}_K - 1) - \mu_i^\top \theta_i
 \end{aligned} \tag{7.5}$$

with Lagrange multipliers  $\mu \in \mathbb{R}_+^{K \times N}$  and  $\lambda \in \mathbb{R}^N$ .

The relaxed dual problem makes use of this Lagrangian function linearized about  $\theta^{(t)}$  which we obtain through a Taylor series approximation,

$$\begin{aligned}
 L(x, \theta_i, \lambda_i, \mu_i) \Big|_{\theta^{(t)}}^{\text{lin}} &\triangleq L \left( x, \theta_i^{(t)}, \lambda_i^{(t)}, \mu_i^{(t)} \right) \\
 &\quad + \sum_{k=1}^K g_{ki}^{(t)}(x) \cdot \left( \theta_{ki} - \theta_{ki}^{(t)} \right),
 \end{aligned} \tag{7.6}$$

where the qualifying constraint function is

$$\begin{aligned}
 g_i^{(t)}(x) &\triangleq \nabla_{\theta_i} L \left( \theta_i, x, \lambda_i^{(t)}, \mu_i^{(t)} \right) \Big|_{\theta_i^{(t)}} \\
 &= -2y_i^\top x + 2\theta_i^{(t)\top} x^\top x \\
 &\quad - \mathbf{1}_K^\top \lambda_i^{(k)} - \mu_i^{(k)\top}.
 \end{aligned} \tag{7.7}$$

The qualifying constraint  $g_i^{(t)}(x)$  is quadratic in  $x$ . However, the qualifying constraints must be linear in  $x$  to yield a convex domain whether  $g_i^{(t)}(x) \geq 0$  or  $g_i^{(t)}(x) \leq 0$ . So, the Lagrangian is linearized first with respect to  $x$  about  $x^{(t)}$  then about  $\theta_i$  at  $\theta_i^{(t)}$ . While the linearized Lagrangian is not a lower bound everywhere in  $x$ , it is a valid lower bound in the region bound by the qualifying constraints with  $\theta_i$  set at the corresponding bounds in the Lagrangian function.



The Lagrangian function linearized about  $x^{(t)}$  is

$$\begin{aligned} L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x^{(t)}}^{\text{lin}} &\triangleq y_i^T y_i - \theta_i^T x^{(t)\top} x^{(t)} \theta_i \\ &\quad - 2y_i^T x \theta_i + 2\theta_i^T x^{(t)\top} x \theta_i \\ &\quad - \lambda_i (\theta_i^T 1_K - 1) - \mu_i^T \theta_i. \end{aligned} \quad (7.8)$$

Subsequently, the Lagrangian function linearized about  $(x^{(t)}, \theta_i^{(t)})$  is

$$\begin{aligned} L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x^{(t)}, \theta_i^{(t)}}^{\text{lin}} &\triangleq y_i^T y_i + \theta_i^{(t)\top} x^{(t)\top} x^{(t)} \theta_i^{(t)} \\ &\quad - 2\theta_i^{(t)\top} x^{(t)\top} x^{(t)} \theta_i \\ &\quad - \lambda_i \left( 1_K^T \theta_i - 1 \right) - \mu_i^T \theta_i, \\ &\quad - 2\theta_i^{(t)\top} x^T x^{(t)} \theta_i^{(t)\top} - 2y_i^T x \theta_i \\ &\quad + 2\theta_i^{(t)\top} \left( x^{(t)\top} x + x^T x^{(t)} \right) \theta_i \end{aligned} \quad (7.9)$$

and the gradient used in the qualifying constraint is

$$\begin{aligned} g_i^{(t)} \Big|_{x^{(t)}}^{\text{lin}}(x) &\triangleq \nabla_{\theta_i} \left[ L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x_0}^{\text{lin}} \right] \Big|_{\theta_i^{(t)}} \\ &= -2x^{(t)\top} x^{(t)} \theta_i^{(t)} - 2x^T y_i \\ &\quad + 2(x^{(t)\top} x + x^T x^{(t)}) \theta_i^{(t)} - \lambda_i 1_K - \mu_i. \end{aligned} \quad (7.10)$$

The qualifying constraints, Lagrange function constraints, and Lagrangian comprise the relaxed dual subproblem at child node  $\text{ch}(n(T))$ .

### 7.3.3.3 Solve the Relaxed Dual Subproblem at the Child Node

Once the valid constraints from the previous  $t = 1, \dots, T - 1$  iterations have been identified and incorporated, the constraint for the current  $T$ th iteration is

$$\begin{aligned} Q &\geq L(x, \theta^{B_T}, y, \lambda^{(t)}, \mu^{(t)}) \Big|_{x^{(t)}, \theta^{(t)}}^{\text{lin}} \\ g_{ki}^{(T)} \Big|_{x^{(t)}}^{\text{lin}}(x) &\leq 0 \text{ if } \theta_{ki}^{B_T} = 1 \\ g_{ki}^{(T)} \Big|_{x^{(t)}}^{\text{lin}}(x) &\geq 0 \text{ if } \theta_{ki}^{B_T} = 0. \end{aligned}$$

The resulting relaxed dual problem is a linear program and can be solved efficiently using the off-the-shelf LP solver Gurobi (Gurobi Optimization, Inc. 2018). We store the optimal objective function value and the optimizing decision variables in the node.

### 7.3.3.4 Update the Lower Bound

The global lower bound, RLBD, is provided by the lowest lower bound across all the leaf nodes in the branch-and-bound tree. Operationally, a hash table maintains a value that is a pointer to a branch-and-bound tree node whose key is the optimal value of the relaxed dual problem at that leaf node. Using this dictionary, branch-and-bound selects the smallest key and bounds to the node of the tree indicated by the value. This element is eliminated from the dictionary since at the end of the next iteration, it will be an interior node and not available for consideration. The iteration count is incremented,  $T \leftarrow T + 1$ , and the global lower bound is updated with the optimal value of the relaxed dual problem at the new node.

### 7.3.3.5 Check Convergence

Since RLBD maintains the lowest lower bound provided by the relaxed dual problem, the lower bound is non-decreasing. If the convergence criteria  $\text{PUBD} - \text{RLBD} \leq \epsilon$  has been met, then the algorithm is exited and the optimal  $\theta$  from the node's primal problem and the optimal  $x$  from the node's relaxed dual problem is reported. Finite  $\epsilon$ -convergence and  $\epsilon$ -global optimality proofs can be found elsewhere (Floudas 2000).

## 7.4 Computational Improvements

In the relaxed dual problem branch-and-bound tree, a leaf node below the current node  $n(T)$  is constructed for each unique region defined by the hyperplane arrangement. In the GOP framework, there are  $KN$  hyperplanes, one for each connected variable and all of the  $KN$  elements of  $\theta$  are connected variables. So, an upper bound on the number of regions defined by  $KN$  cuts is  $2^{KN}$  because each region may be found by selecting a side of each cut. Thus we have the computationally complex situation of needing to solve a relaxed dual problem for each of the  $2^{KN}$  possible regions.

Let an arrangement  $\mathcal{A}$  denote a set of hyperplanes and  $r(\mathcal{A})$  denote the set of unique regions defined by  $\mathcal{A}$ . In our particular situation, all of the hyperplanes pass through the unique point  $x^{(n(T))}$ , so all of the regions are unbounded except by the constraints provided in  $\mathcal{R}$ . A recursive algorithm for counting the number of regions  $|r(\mathcal{A})|$  known as Zaslavsky's Theorem, is outlined in (Zaslavsky 1975).

Indeed,  $|r(\mathcal{A})|$  is often much less than  $2^{|\mathcal{A}|}$ . Due to its recursive nature, computing the number of hyperplanes using Zaslavsky's theorem can be computationally slow, though it can also be much better than the original  $2^{KN}$  number of subproblems.

### 7.4.1 Cell Enumeration Algorithm

To address the computational complexity we have developed an A-star search algorithm for cell enumeration to simultaneously identify and count the set of unique regions defined by arrangement  $\mathcal{A}$  with sign vectors. The algorithm proceeds as follows. First, preprocess the arrangement  $\mathcal{A}$  to eliminate trivial and redundant hyperplanes. Next, eliminate a hyperplane from  $\mathcal{A}$  if the coefficients are all zero and eliminate duplicate hyperplanes in  $\mathcal{A}$  (see Appendix 3). What is left is a reduced arrangement,  $\mathcal{A}'$ .

Here, we define two concepts, *strict hyperplane* and *adjacent region*. A strict hyperplane is defined as non-redundant bounding hyperplane in a single region. If two regions exist that have sign vectors differing in only one hyperplane, then this hyperplane is a strict hyperplane. We define an adjacent region of region  $r$  as a neighbor region of  $r$  if they are separated by exactly one strict hyperplane. The general idea of the A-star algorithm uses ideas from partial order sets. We first initialize a root region using an interior point method and then determine all of its adjacent regions by identifying the set of strict hyperplanes. This process guarantees that we can enumerate all unique regions.

We define  $\theta^B \in \{0, 1\}^{r(\mathcal{A}') \times KN}$ . The rows are regions and there are  $KN$  columns. Each element of this matrix is either 0 or 1. The  $b$ th region in  $r(\mathcal{A}')$  is uniquely identified by the zero-one vector in the  $b$ th row of  $\theta^B$ . If the  $b$ th element of the  $k$ th row of  $\theta^B$  is +1, then  $g_{ki} \leq 0$ . Similarly, if the  $b$ th element of the  $k$ th row of  $\theta^B$  is 0, then  $g_{ki} \geq 0$ . The A-star search algorithm completes the  $\theta^B$  matrix for the current node  $n(T)$  and a leaf node is generated for each row of  $\theta^B$ . Thus each unique region defined by the qualifying constraint cuts provided by the Lagrange dual of the primal problem at the current node. The details of the A-star search algorithm are covered in Appendix 3.

### 7.4.2 Theoretical Time Complexity

The GOP algorithm has four main components: primal problem, preprocessing, unique region identification, and relaxed dual problems. We analyze the computational complexity of each in turn.

### 7.4.2.1 Primal Problem

The primal problem is a convex quadratic program with  $KN$  decision variables. The time complexity for the primal problem solving is then  $O(K^3N^3)$  (Boyd and Vandenberghe 2004).

### 7.4.2.2 Preprocessing

We address the cases of overlapping qualifying constraint cuts by sorting the rows of the  $KN \cdot M$  qualifying constraint coefficient matrix and comparing the coefficients of adjacent rows. We first sort the  $KN$  rows of the qualifying constraint coefficient matrix using heapsort which takes  $O(KN \cdot \log(KN))$  time on average. The algorithm subsequently passes through the rows of the matrix to identify all-zero coefficients and duplicate cuts; each pass takes  $O(KN)$  time. We define  $|\mathcal{A}'|$  as the number of unique qualifying constraints.

### 7.4.2.3 Unique Region Identification

The interior point method that we used in the A-star search algorithm is a linear program of size  $|\mathcal{A}'| \cdot MK$  with the time complexity of  $O(|\mathcal{A}'| \cdot MK)$ . The time complexity for enumerating the set of unique regions is  $O(|\mathcal{A}'| \cdot (|\mathcal{A}'| \cdot MK))$ , which exhibits polynomial behavior. The time complexity of the partial order A-star algorithm is polynomial in the best case and exponential in the worst case, depending on the heuristic. We define  $|r(\mathcal{A}')|$  as the number of identified unique regions.

### 7.4.2.4 Relaxed Dual Problems

There are  $2MK + 1$  decision variables for each relaxed dual problem, so the time complexity for each is  $O(M^3K^3)$ . The total time for solving the relaxed dual problems is  $O(|r(\mathcal{A}')| \cdot M^3K^3)$ , which depends on the number of relaxed dual problems.

## 7.5 Experiments

In this section, we present our experiments on synthetic data sets and show accuracy and convergence speed. Computational complexity is evaluated by both the theoretical and empirical time complexity.

### 7.5.1 Illustrative Example

We use a simple data set to show the operation of the algorithm in detail and facilitate visualization of the cut sets. The data set,  $y$ , and true decision variable values,  $(x^*, \theta^*)$ , are

$$x^* = [0, -1], \theta^* = \begin{bmatrix} 1, 0, 0.5 \\ 0, 1, 0.5 \end{bmatrix},$$

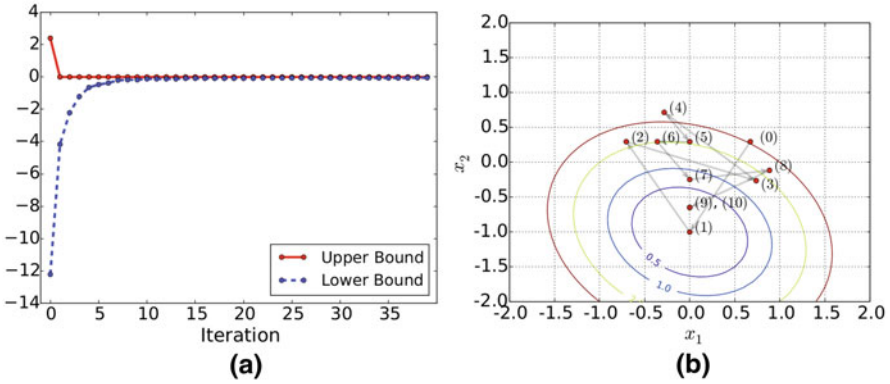
$$y = [0, -1, -0.5].$$

We ran the GOP algorithm with sparsity constraint variable  $P = 1$  and convergence tolerance  $\epsilon = 0.01$ . There are  $KN = 6$  connected variables, so we solve at most  $2^{KN} = 64$  relaxed dual problems at each iteration. These relaxed dual problems are independent and can be distributed to different computational threads or cores. The primal problem is a single optimization problem and will not be distributed. The optimal decision variables after 72 iterations are

$$\hat{x} = x^{(72)} = [0.080, -0.920], \hat{\theta} = \theta^{(72)} = \begin{bmatrix} 1.00, 0.080, 0.580 \\ 0.00, 0.920, 0.420 \end{bmatrix}, \quad (7.11)$$

and the Lagrange multipliers are  $\hat{\lambda} = [-0.147, 0, 0]$  and  $\hat{\mu} = [0, 0, 0; 0.160, 0, 0]$ .

Figure 7.1a shows the convergence of the upper and lower bounds by iteration. The upper bound converges quickly and the majority of the time in the algorithm is spent proving optimality. With each iteration regions of the solution space are tested until the lower bound is tightened sufficiently to meet the stopping criterion. Figure 7.1b shows the first ten  $x$  values considered by the algorithm with isoclines of the objective function with  $\theta^*$  fixed. It is evident that the algorithm is not performing hill-climbing or any other gradient ascent algorithm during its search for the global optimum. Instead, the algorithm explores a region bound by the qualifying constraints to construct a lower bound on the objective function. We run it using 20 random initial values and the optimal objective functions for all random initializations are all 0, which shows that the GOP algorithm found the globally optimal solutions of this small instance. Furthermore, the algorithm does not search nested regions, but considers previously explored cut sets (Fig. 7.1b).

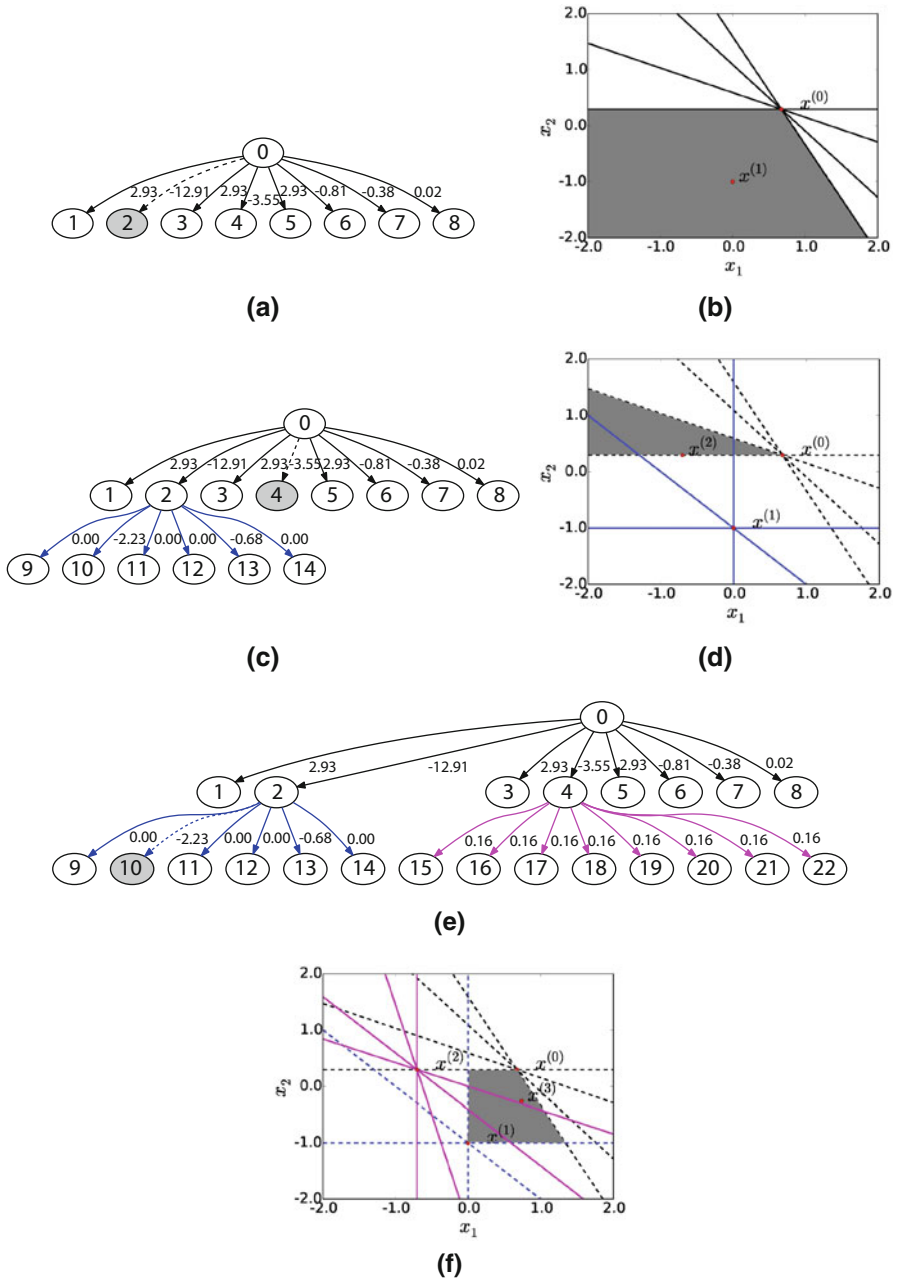


**Fig. 7.1** (a) GOP optimal upper and lower bounds, (b) GOP optimal relaxed dual problem decision variables

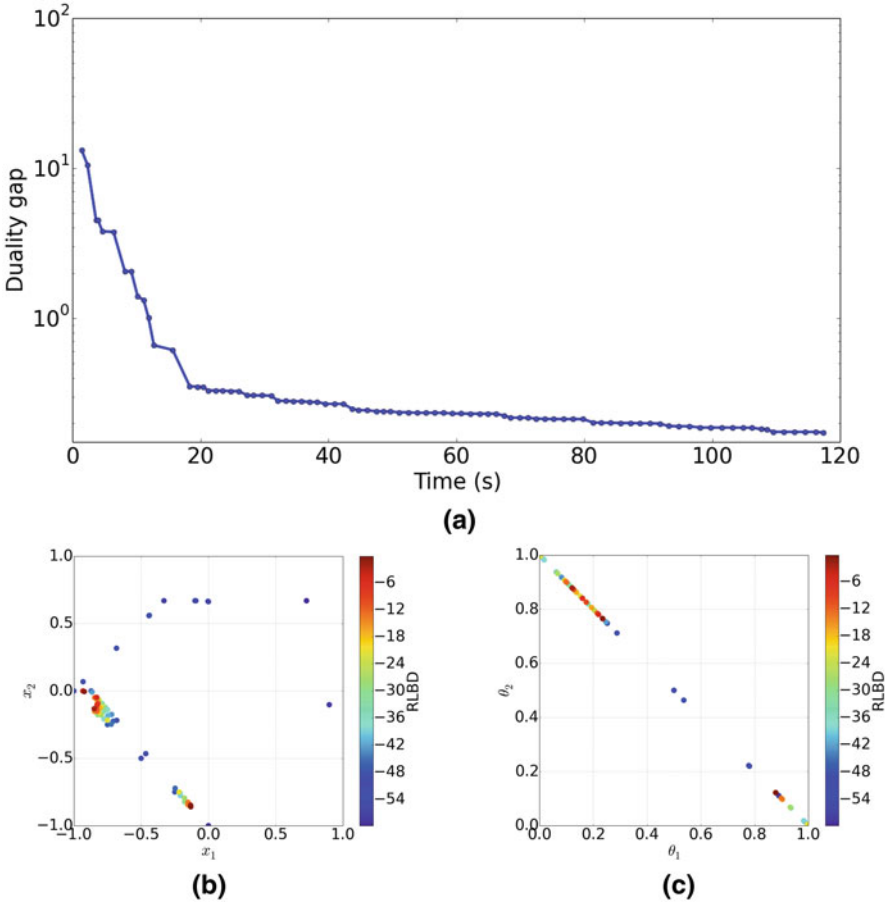
Figure 7.2a and b shows the branch-and-bound tree and corresponding  $x$ -space region with the sequence of cut sets for the first three iterations of the algorithm. One cut in Fig. 7.2c–f is obtained for each of the  $KN$  qualifying constraints. We initialize the algorithm at  $x^{(0)}$ .

### 7.5.2 Accuracy and Convergence Speed

We ran our GOP algorithm using 64 processors on a synthetic data set which is randomly generated on the scale of one feature ( $M = 1$ ), two subtypes ( $K = 2$ ) and ten samples ( $N = 10$ ). Figure 7.3a shows that our GOP algorithm converges very quickly to  $-0.17$  duality gap (PUBD  $-$  RLBD) in the first 89 iterations in 120 s. The optimal  $x$  ( $x_1, x_2$ ) and  $\theta$  ( $\theta_1, \theta_2$ ) of each iteration are shown with a range of colors to represent corresponding RLBD in Fig. 7.3b,c. The dark blue represents low RLBD and the dark red represents high RLBD. The RLBD of the initial  $x$ ,  $x^{(0)}$ , is  $-59.87$ ; The RLBD of iteration 89,  $x^{(89)}$ , is  $-0.17$ . It demonstrates that the GOP algorithm can change modes very easily without getting stuck in local optima.



**Fig. 7.2** (a) Branch-and-bound tree at iteration 1, (b)  $x$ -space region at iteration 1, (c) Branch-and-bound tree at iteration 2, (d)  $x$ -space region at iteration 2, (e) Branch-and-bound tree at iteration 3, (f)  $x$ -space region at iteration 3



**Fig. 7.3** (a) Duality gap through the first 120 s, (b) Optimal  $x$  of each iteration. The true  $x$  is  $(0, -1)$ , (c) Optimal  $\theta$  of each iteration. The true  $\theta$  is  $(0.22, 0.78)$

### 7.5.3 Computational Complexity

We compare our theoretical complexity analysis with empirical measurements of the time complexity on simulated data sets.

We constructed 12 synthetic data sets in a full-factorial arrangement with  $M \in \{20, 40, 60, 80\}$ ,  $K \in \{2\}$ , and  $N \in \{4, 5, 6\}$  and measured CPU time for each



component of one iteration. For each arrangement, each element of the true  $x^*$  is:

$$x_{mk}^* = \begin{cases} 1 & \text{if } 0 \leq m < M/4, k = 0 \\ -1 & \text{if } M/4 \leq m < M/2, k = 1 \\ \mathcal{N}(0, 0.5^2) & \text{if } M/2 \leq m < M, \forall k \\ 0 & \text{otherwise} \end{cases}$$

Here  $\mathcal{N}(0, 0.5^2)$  is the sample from a Normal distribution by its mean 0 and standard deviation 0.5. For the true  $\theta^*$ ,  $\theta_{kn}^*$  for  $k = 0$  are  $n$  evenly spaced samples over the interval of  $[0, 1]$ ;  $\theta_{kn}^*$  for  $k = 1$  are  $n$  evenly spaced samples over the interval of  $[1, 0]$ .

**Table 7.1** Timing profile (in seconds) of each component of the GOP algorithm for one iteration varying problem size

N	M	Primal problem	Preprocessing	Unique region ID	Relaxed dual problems	Total
4	20	0.10	1.69	1.29	1.54 (33%)	4.62
	40	0.12	1.91	1.72	1.69 (31%)	5.44
	60	0.12	2.03	1.11	1.77 (35%)	5.03
	80	0.13	2.39	2.05	3.70 (45%)	8.27
5	20	0.11	1.99	1.31	11.26 (77%)	14.67
	40	0.11	2.07	1.37	11.45 (76%)	15.00
	60	0.11	1.86	1.41	12.33 (78%)	15.71
	80	0.12	2.23	1.26	17.96 (83%)	21.57
6	20	0.14	2.21	2.50	65.71 (93%)	70.56
	40	0.13	2.83	2.49	67.08 (92%)	72.53
	60	0.12	3.45	2.80	69.00 (92%)	75.37
	80	0.12	3.15	2.80	77.62 (93%)	83.69

Table 7.1 shows that the time per iteration increases linearly with  $M$  when  $K$  and  $N$  are fixed. The time for solving all the relaxed dual problems increases as the number of samples increases. Even though the step of solving all the relaxed dual problems takes more than 90% of the total time per iteration when the number of samples is 6, our algorithm is easily parallelized to solve the relaxed dual problems, allowing the algorithm to scale nearly linearly with the size of the data set.

### 7.5.4 Real Data Analysis

To explore the performance of our algorithm on real data, we performed experiments on the TCGA pancancer high throughput DNA sequencing data set (Weinstein et al. 2013; Dheeru and Karra Taniskidou 2017). The original data was subsetted to the top two most variable genes and the top ten most variable samples by standard

deviation. Then it was log transformed and centered across genes. The number of clusters was set to  $K = 2$ , the sparsity constraint was set to  $P = 1$ .

At early iterations, the optimal  $\theta$  is a nearly 0–1 matrix, so we report the samples associated with each of the  $K = 2$  subtypes. Samples 1, 4, 5, and 7 were assigned to subtype A and samples 2, 3, 6, 8, 9 and 10 were assigned to subtype B; subtypes are labeled arbitrarily with letters. The optimal  $x$  values were  $x_A = [-0.204, 0]$  and  $x_B = [0.561, 0.234]$ . The inference algorithm enforced the  $L_1$  penalty—the sum of the absolute values of  $x$  are at  $P = 1$ . And, the  $L_1$  penalty clearly enforced sparsity in that one of the elements is exactly equal to zero.

The data set provides the anatomical regions associated with each of the cancer samples, and we explored those assignments to see if there is an association between the subtypes and the anatomical site of the cancer. Subtype A contains three colon adenocarcinomas and one prostate adenocarcinomas; subtype B contains four breast invasive carcinomas, one lung adenocarcinoma, and one kidney adenocarcinoma. Clearly, the algorithm is effectively clustering colon adenocarcinomas and cancers that are genomically more like that type from breast adenocarcinomas and cancers that are genomically more like that type.

At later iterations, when the duality gap had narrowed to 3.65, the optimal  $\theta$  is more mixed. Still, the majority of the colorectal adenocarcinomas had subtype A as their largest component, and the majority of breast invasive carcinomas had subtype B as their largest component. These results indicate that this globally optimal inference algorithm performs well on a real data set. Since the algorithm provides both upper and lower bounds, a proof of  $\epsilon$ -optimality is provided. Within this tolerance, the algorithm provides confidence that the provided estimates are globally optimal and not merely an artifact of local convergence.

## 7.6 Discussion

We have presented a global optimization algorithm for a mixed membership matrix factorization problem. Our algorithm brings ideas from the global optimization community (Benders' decomposition and the GOP method) into contact with statistical inference problems for the first time. The naive computational cost of the global optimal solution is the need to solve a number of linear programs that grows exponentially in the number of connected variables in the worst case—in this case the  $KN$  elements of  $\theta$ . Many of these linear programs are redundant or yield optimal solutions that are greater than the current upper bound and thus not useful. A branch-and-bound framework (Floudas 2000) reduces the need to solve all possible relaxed dual problems by fathoming parts of the solution space. We further mitigate this cost by developing a search algorithm for identifying and enumerating the true number of unique linear programs.

Finally, we have derived an algorithm for particular loss functions for the sparsity constraint and objective function. The GOP framework can handle integer variables and thus may be used with an  $\ell_0$  counting “norm” rather than the  $\ell_1$  norm to induce

sparsity. This would give us a mixed-integer biconvex program, but the conditions for the framework. Structured sparsity constraints can also be defined as is done for elastic-net extensions of LASSO regression. It may be useful to consider other loss functions for the objective function depending on the application.

We are exploring the connections between GOP and the other alternating optimization algorithms such as the expectation maximization (EM) and variational EM algorithm. Since the complexity of GOP only depends on the connected variables, the graphical model structure connecting the complicating and non-complicating variables may be used to identify the worst-case complexity of the algorithm prior to running the algorithm. A factorized graph structure may provide an approximate, but computationally efficient algorithm based on GOP. Additionally, because the Lagrangian function factorizes into the sum of Lagrangian functions for each sample in the data set, we may be able to update the parameters based on GOP for a selected subset of the data in an iterative or sequential algorithm. We are exploring the statistical consistency properties of such an update procedure.

**Acknowledgements** We acknowledge Hachem Saddiki for valuable discussions and comments on the manuscript.

## Appendix 1: Derivation of Relaxed Dual Problem Constraints

The Lagrange function is the sum of the Lagrange functions for each sample,

$$L(y, \theta, x, \lambda) = \sum_{i=1}^n L(y_i, \theta_i, x, \lambda_i, \mu_i), \quad (7.12)$$

and the Lagrange function for a single sample is

$$L(y_i, \theta_i, x, \lambda_i, \mu_i) = y_i^T y_i - 2y_i^T x \theta_i + \theta_i^T x^T x \theta_i - \lambda_i (\theta_i^T 1_K - 1) - \mu_i^T \theta_i. \quad (7.13)$$

We see that the Lagrange function is biconvex in  $x$  and  $\theta_i$ . We develop the constraints for a single sample for the remainder.

### *Linearized Lagrange Function with Respect to $x$*

Casting  $x$  as a vector and rewriting the Lagrange function gives

$$L(y_i, \theta_i, \bar{x}, \lambda_i, \mu_i) = a_i - 2b_i^T \bar{x} + \bar{x}^T C_i \bar{x} - \lambda_i (\theta_i^T 1_K - 1) - \mu_i^T \theta_i, \quad (7.14)$$

where  $\bar{x}$  is formed by stacking the columns of  $x$  in order. The coefficients are formed such that

$$\begin{aligned} a &= y_i^T y_i, \\ b_i^T \bar{x} &= y_i^T x \theta_i, \\ \bar{x}^T C_i \bar{x} &= \theta_i^T x^T x \theta_i. \end{aligned}$$

The linear coefficient matrix is the  $KM \times 1$  vector,

$$b_i = [y_i \theta_{1i}, \dots, y_i \theta_{Ki}]$$

The quadratic coefficient is the  $KM \times KM$  and block matrix

$$C_i = \begin{bmatrix} \theta_{1i}^2 I_M & \cdots & \theta_{1i} \theta_{Ki} I_M \\ \vdots & \ddots & \vdots \\ \theta_{Ki} \theta_{1i} I_M & \cdots & \theta_{Ki}^2 I_M \end{bmatrix}$$

The Taylor series approximation about  $x_0$  is

$$L(y_i, \theta_i, \bar{x}, \lambda_i, \mu_i) \Big|_{\bar{x}_0}^{\text{lin}} = L(y_i, x_0, \theta_i, \lambda_i, \mu_i) + (\nabla_x L|_{x_0})^T (x - x_0). \quad (7.15)$$

The gradient with respect to  $x$  is

$$\nabla_x L(y_i, \theta_i, \bar{x}, \lambda_i, \mu_i) = -2b_i + 2C_i \bar{x}. \quad (7.16)$$

Plugging the gradient into the Taylor series approximation gives

$$L(y_i, \theta_i, \bar{x}, \lambda_i) \Big|_{\bar{x}_0}^{\text{lin}} = a_i - 2b_i^T \bar{x}_0 + \bar{x}_0^T C_i \bar{x}_0 - \lambda_i (\theta_i^T \mathbf{1}_K - 1) - \mu_i^T \theta_i + (-2b_i + 2C_i \bar{x}_0)^T (\bar{x} - \bar{x}_0). \quad (7.17)$$

Simplifying the linearized Lagrange function gives

$$L(y_i, \theta_i, \bar{x}, \lambda_i, \mu_i) \Big|_{\bar{x}_0}^{\text{lin}} = (y_i^T y_i - \bar{x}_0^T C_i \bar{x}_0 - \lambda_i (\theta_i^T \mathbf{1}_K - 1) - \mu_i^T \theta_i) - 2b_i^T \bar{x} + 2\bar{x}_0^T C_i \bar{x} \quad (7.18)$$

Finally, we write the linearized Lagrangian using the matrix form of  $x_0$ ,

$$L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x_0}^{\text{lin}} = y_i^T y_i^T - \theta_i^T x_0^T x_0 \theta_i - 2y_i^T x \theta_i + 2\theta_i^T x_0^T x \theta_i - \lambda_i (\theta_i^T \mathbf{1}_K - 1) - \mu_i^T \theta_i \quad (7.19)$$

While the original Lagrange function is convex in  $\theta_i$  for a fixed  $x$ , the linearized Lagrange function is not necessarily convex in  $\theta_i$ . This can be seen by collecting the quadratic, linear and constant terms with respect to  $\theta_i$ ,

$$L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x_0}^{\text{lin}} = (y_i^T y_i^T + \lambda_i) + (-2y_i^T x - \lambda_i 1_K^T - \mu_i^T) \theta_i + \theta_i^T (2x_0^T x - x_0^T x_0) \theta_i. \quad (7.20)$$

Now, if and only if  $2x_0^T x - x_0^T x_0 \succeq 0$  is positive semidefinite, then  $L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x_0}^{\text{lin}}$  is convex. The condition is satisfied at  $x = x_0$  but may be violated at some other value of  $x$ .

### ***Linearized Lagrange Function with Respect to $\theta_i$***

Now, we linearize (7.18) with respect to  $\theta_i$ . Using the Taylor series approximation with respect to  $\theta_{0i}$  gives

$$L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x_0, \theta_{0i}}^{\text{lin}} = L(y_i, \theta_{0i}, x, \lambda_i, \mu_i) \Big|_{x_0}^{\text{lin}} + \left( \nabla_{\theta_i} L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x_0, \theta_{0i}}^{\text{lin}} \right)^T (\theta_i - \theta_{0i}) \quad (7.21)$$

The gradient for this Taylor series approximation is

$$g_i(x) \triangleq \nabla_{\theta_i} L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x_0, \theta_{0i}}^{\text{lin}} = -2x_0^T x_0 \theta_{0i} - 2x^T y_i + 2(x_0^T x + x^T x_0) \theta_{0i} - \lambda_i 1_K - \mu_i, \quad (7.22)$$

where  $g_i(x)$  is the vector of  $K$  qualifying constraints associated with the Lagrange function. The qualifying constraint is linear in  $x$ . Plugging the gradient into the approximation gives

$$\begin{aligned} L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x_0, \theta_{0i}}^{\text{lin}} &= y_i^T y_i^T - \theta_{0i}^T x_0^T x_0 \theta_{0i} - 2y_i^T x \theta_{0i} + 2\theta_{0i}^T x_0^T x \theta_{0i} - \lambda_i (\theta_{0i}^T 1_K - 1) \\ &\quad - \mu_i^T \theta_{0i} + \left( -2x_0^T x_0 \theta_{0i} - 2x^T y_i + 2(x_0^T x + x^T x_0) \theta_{0i} - \lambda_i 1_K - \mu_i \right)^T (\theta_i - \theta_{0i}) \end{aligned} \quad (7.23)$$

The linearized Lagrange function is bi-linear in  $x$  and  $\theta_i$ . Finally, simplifying the linearized Lagrange function gives

$$L(y_i, \theta_i, x, \lambda_i, \mu_i) \Big|_{x_0, \theta_{0i}}^{\text{lin}} = y_i^T y_i^T + \theta_{0i}^T x_0^T x_0 \theta_{0i} - 2\theta_{0i}^T x_0^T x_0 \theta_i - \lambda_i (1_K^T \theta_i - 1) - \mu_i^T \theta_i - 2\theta_{0i}^T x^T x_0 \theta_{0i} - 2y_i^T x \theta_i + 2\theta_{0i}^T (x_0^T x + x^T x_0) \theta_i \quad (7.24)$$

## Appendix 2: Proof of Biconvexity

To prove the optimization problem is biconvex, first we show the feasible region over which we are optimizing is biconvex. Then, we show the objective function is biconvex by fixing  $\theta$  and showing convexity with respect to  $x$ , and then vice versa.

### *The Constraints Form a Biconvex Feasible Region*

Our constraints can be written as

$$\|x\|_1 \leq P \quad (7.25)$$

$$\sum_{k=1}^K \theta_{ki} = 1 \quad \forall i \quad (7.26)$$

$$0 \leq \theta_{ki} \leq 1 \quad \forall (k, i). \quad (7.27)$$

The inequality constraint (7.25) is convex if either  $x$  or  $\theta$  is fixed, because any norm is convex. The equality constraints (7.26) is an affine combination that is still affine if either  $x$  or  $\theta$  is fixed. Every affine set is convex. The inequality constraint (7.27) is convex if either  $x$  or  $\theta$  is fixed, because  $\theta$  is a linear function.

### *The Objective Is Convex with Respect to $\theta$*

We prove the objective is a biconvex function using the following two theorems.

**Theorem 1** *Let  $A \subseteq \mathbb{R}^n$  be a convex open set and let  $f : A \rightarrow \mathbb{R}$  be twice differentiable. Write  $H(x)$  for the Hessian matrix of  $f$  at  $x \in A$ . If  $H(x)$  is positive semidefinite for all  $x \in A$ , then  $f$  is convex (Boyd and Vandenberghe 2004).*

**Theorem 2** *A symmetric matrix  $A$  is positive semidefinite (PSD) if and only if there exists  $B$  such that  $A = B^T B$  (Lancaster et al. 1985).*

The objective of our problem is,

$$f(y, x, \theta) = \|y - x\theta\|_2^2 = (y - x\theta)^T (y - x\theta) \quad (7.28)$$

$$= (y^T - \theta^T x^T)(y - x\theta) \quad (7.29)$$

$$= y^T y - y^T x\theta - \theta^T x^T y + \theta^T x^T x\theta. \quad (7.30)$$

The objective function is the sum of the objective functions for each sample.

$$f(y, x, \theta) = \sum_{i=1}^N f(y_i, x, \theta_i) \quad (7.31)$$

$$= \sum_{i=1}^N y_i^T y_i - 2y_i^T x\theta_i + \theta_i^T x^T x\theta_i. \quad (7.32)$$

The gradient with respect to  $\theta_i$ ,

$$\nabla_{\theta_i} f(y_i, x, \theta_i) = -2y_i^T x + \left( x^T x + (x^T x)^T \right) \theta_i \quad (7.33)$$

$$= -2x^T y_i + 2x^T x\theta_i. \quad (7.34)$$

Take second derivative with respect to  $\theta_i$  to get Hessian matrix,

$$\nabla_{\theta_i}^2 f(y_i, x, \theta_i) = \nabla_{\theta_i} \left( -2x^T y_i + 2x^T x\theta_i \right) \quad (7.35)$$

$$= 2\nabla_{\theta_i} \left( x^T x\theta_i \right) \quad (7.36)$$

$$= 2 \left( x^T x \right)^T \quad (7.37)$$

$$= 2x^T x. \quad (7.38)$$

The Hessian matrix  $\nabla_{\theta_i}^2 f(y_i, x, \theta_i)$  is positive semidefinite based on Theorem 2. Then, we have  $f(y_i, x, \theta_i)$  is convex in  $\theta_i$  based on Theorem 1. The objective  $f(y, x, \theta)$  is convex with respect to  $\theta$ , because the sum of convex functions,  $\sum_{i=1}^N f(y_i, x, \theta_i)$ , is still a convex function.

### ***The Objective Is Convex with Respect to $x$***

The objective function for sample  $i$  is

$$f(y_i, x, \theta_i) = y_i^T y_i - 2y_i^T x \theta_i + \theta_i^T x^T x \theta_i. \quad (7.39)$$

We cast  $x$  as a vector  $\bar{x}$ , which is formed by stacking the columns of  $x$  in order. We rewrite the objective function as

$$f(y_i, \bar{x}, \theta_i) = a_i - 2b_i^T \bar{x} + \bar{x}^T C_i \bar{x}. \quad (7.40)$$

The coefficients are formed such that

$$a = y_i^T y_i, \quad (7.41)$$

$$b_i^T \bar{x} = y_i^T x \theta_i, \quad (7.42)$$

$$\bar{x}^T C_i \bar{x} = \theta_i^T x^T x \theta_i. \quad (7.43)$$

The linear coefficient matrix is the  $KM \times 1$  vector

$$b_i = [y_i \theta_{1i}, \dots, y_i \theta_{Ki}] \quad (7.44)$$

The quadratic coefficient is the  $KM \times KM$  and block matrix

$$C_i = \begin{bmatrix} \theta_{1i}^2 I_M & \cdots & \theta_{1i} \theta_{Ki} I_M \\ \vdots & \ddots & \vdots \\ \theta_{Ki} \theta_{1i} I_M & \cdots & \theta_{Ki}^2 I_M \end{bmatrix} \quad (7.45)$$

The gradient with respect to  $\bar{x}$

$$\nabla_{\bar{x}} f(y_i, \bar{x}, \theta_i) = -2b_i + 2C_i \bar{x}. \quad (7.46)$$

Take second derivative to get Hessian matrix,

$$\nabla_{\bar{x}^2} f(y_i, \bar{x}, \theta_i) = 2C_i^T \quad (7.47)$$

$$= 2 \left( \theta_i \theta_i^T \right)^T \quad (7.48)$$

$$= 2 \left( \theta_i^T \right)^T \left( \theta_i^T \right). \quad (7.49)$$



The Hessian matrix  $\nabla_{\bar{x}}^2 f(y_i, \bar{x}, \theta_i)$  is positive semidefinite based on Theorem 2. Then, we have  $f(y_i, \bar{x}, \theta_i)$  is convex in  $\bar{x}$  based on Theorem 1. The objective  $f(y, x, \theta)$  is convex with respect to  $x$ , because the sum of convex functions,  $\sum_{i=1}^N f(y_i, x, \theta_i)$ , is still a convex function.

The objective is biconvex with respect to both  $x$  and  $\theta$ . Thus, we have a biconvex optimization problem based on the proof of biconvexity of the constraints and the objective.

### Appendix 3: A-Star Search Algorithm

In this procedure, first we remove all the duplicate and all-zero coefficients hyperplanes to get unique hyperplanes. Then we start from a specific region  $r$  and put it into a open set. Open set is used to maintain a region list which need to be explored. Each time we pick one region from the open set to find adjacent regions. Once finishing the step of finding adjacent regions, region  $r$  will be moved into a closed set. Closed set is used to maintain a region list which already be explored. Also, if the adjacent region is a newly found one, it also need to be put into the open set for exploring. Finally, once the open set is empty, regions in the closed set are all the unique regions, and the number of the unique regions is the length of the closed set. This procedure begins from one region and expands to all the neighbors until no new neighbor is existed.

The overview of the A-star search algorithm to identify unique regions is shown in Algorithm 1.

---

#### Algorithm 1 A-star Search Algorithm

---

- 1: Sort the rows of the  $KN \times M$  qualifying constraint coefficient matrix.
  - 2: Compare adjacent rows of the qualifying constraint coefficient matrix and eliminate duplicate rows.
  - 3: Eliminate rows of the qualifying constraint coefficient matrix with all-zero coefficients.
  - 4: Determine the list of unique qualifying constraints by pairwise test.
  - 5: Set  $S$  and  $|\mathcal{A}'|$  to the set of unique, non-trivial qualifying constraints and the number of them.
  - 6: Initialize a region  $root$  using an interior point method (Component 1).
  - 7: Put region  $root$  into the open set.
  - 8: **if** open set is not empty **then**
  - 9:     Get a region  $R$  from the open set.
  - 10:     Calculate the adjacent regions set  $Radj$  (Component 2).
  - 11:     Put region  $R$  into the closed set.
  - 12:     **for** each region  $r$  in  $Radj$  **do**
  - 13:         **if**  $r$  is not in the open set **and** not in the closed set **then**
  - 14:             Put region  $r$  into the open set.
  - 15: Reflect the sign of the regions in the close set.
  - 16: Get all the regions represented by string of 0 and 1.
-

## Hyperplane Filtering

Assuming there are two different hyperplanes  $H_i$  and  $H_j$  represented by  $A_i = \{a_{i,0}, \dots, a_{i,MK}\}$  and  $A_j = \{a_{j,0}, \dots, a_{j,MK}\}$ . We take these two hyperplanes duplicated when

$$\frac{a_{i,0}}{a_{j,0}} = \frac{a_{i,1}}{a_{j,1}} = \dots = \frac{a_{i,MK}}{a_{j,MK}} = \frac{\sum_{l=0}^{MK} a_{i,l}}{\sum_{l=0}^{MK} a_{j,l}}, a_{j,l} \neq 0 \quad (7.50)$$

This can be converted to

$$\left| \sum_{l=0}^{MK} a_{i,l} \cdot a_{j,n} - \sum_{l=0}^{MK} a_{j,l} \cdot a_{i,n} \right| \leq \tau, \forall n \in [0, MK] \quad (7.51)$$

where threshold  $\tau$  is a very small positive value.

We eliminate a hyperplane  $H_i$  represented by  $A_i = \{a_{i,0}, \dots, a_{i,MK}\}$  from hyperplane arrangement  $\mathcal{A}$  if the coefficients of  $A_i$  are all zero,

$$|a_{i,j}| \leq \tau \text{ for all } a_{i,j} \in A_i \text{ and } j \in [0, MK]$$

The arrangement  $\mathcal{A}'$  is the reduced arrangement and  $A'x = b$  are the equations of unique hyperplanes.

## Interior Point Method

An interior point is found by solving the following optimization problem:

$$\begin{aligned} & \text{maximize } z \\ & \text{subject to } -A'_i x + z \leq b_i \quad \text{if } \theta_i^B = 0, \\ & \quad \quad \quad A'_i x + z \leq -b_i \quad \text{if } \theta_i^B = 1, \\ & \quad \quad \quad z > 0 \end{aligned} \quad (7.52)$$

**Algorithm 2** Interior Point Method (Component 1)

- 
- 1: Generate  $2^{|\mathcal{A}'|}$  different strings using 0 and 1.
  - 2: **for** each  $s$  in the strings **do**
  - 3:     Solve an optimization problem to get an interior point.
  - 4:     **if** Get a interior point **then**
  - 5:         Get the *root* region represented by 0 and 1.
- 

**Algorithm 3** Get Adjacent Regions (Component 2)

- 
- 1: Initialize an empty set  $SH$  for strict hyperplanes.
  - 2: Initialize an adjacent region set  $ADJ$ .
  - 3: # Find out all the strict hyperplanes for region  $R$ .
  - 4: **for** each hyperplane  $H$  of  $|\mathcal{A}'|$  hyperplanes **do**
  - 5:     Pick one hyperplane  $H$  from all the hyperplanes defining region  $R$ .
  - 6:     Flip the sign of  $H$  to get  $\neg H$ .
  - 7:     Form a new hyperplane arrangement  $\neg\mathcal{A}'$  with  $\neg H$ .
  - 8:     Solve the problem to get an interior point constrained by  $\neg\mathcal{A}'$ .
  - 9:     **if** the interior point is not Non **then**
  - 10:          $H$  is a strict hyperplane and put into set  $SH$ .
  - 11:     **else**
  - 12:          $H$  is a redundant hyperplane.
  - 13: # Find out all the adjacent regions for region  $R$ .
  - 14: **for** each strict hyperplane  $sh$  in set  $SH$  **do**
  - 15:     Take the opposite sign  $\neg sh$  of  $sh$ .
  - 16:     Form a adjacent region  $adj$  based on  $\neg sh$  and all the else hyperplanes.
  - 17:     Put  $adj$  into set  $ADJ$ .
- 

**References**

- Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.* **9**, 1981–2014 (2008)
- Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* **4**(1), 238–252 (1962)
- Blei, D.M., Lafferty, J.D.: Correlated topic models. In: *Proceedings of the International Conference on Machine Learning*, pp 113–120 (2006)
- Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
- Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* **112**(518), 859–877 (2017)
- Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
- Dheeru, D., Karra T.E.: UCI machine learning repository. URL UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
- Floudas, C.A.: *Deterministic Global Optimization, Nonconvex Optimization and Its Applications*, vol 37. Springer, Boston (2000)
- Floudas, C.A.: *Deterministic Global Optimization: Theory, Methods and Applications*, vol. 37. Springer, Berlin (2013)
- Floudas, C.A., Gounaris, C.E.: A review of recent advances in global optimization. *J. Glob. Optim.* **45**, 3–38 (2008)
- Floudas, C.A., Visweswaran, V.: A global optimization algorithm (GOP) for certain classes of nonconvex NLPs. *Comput. Chem. Eng.* **14**(12), 1–34 (1990)

- Geoffrion, A.M.: Generalized benders decomposition. *J. Optim. Theory Appl.* **10**, 237–260 (1972)
- Gorski, J., Pfeuffer, F., Klamroth, K.: Biconvex sets and optimization with biconvex functions: a survey and extensions. *Math. Methods Oper. Res.* **66**, 373–407 (2007)
- Gurobi Optimization, Inc (2018) Gurobi optimizer version 8.0
- Horst, R., Tuy, H.: *Global Optimization: Deterministic Approaches*. Springer, Berlin (2013)
- Kabán, A.: On Bayesian classification with laplace priors. *Pattern Recognit. Lett.* **28**(10), 1271–1282 (2007)
- Lancaster, P., Tismenetsky, M., et al.: *The theory of matrices: with applications*. Elsevier, San Diego (1985)
- Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* **401**, 788–791 (1999)
- MacKay, D.J.C.: Bayesian interpolation. *Neural Comput.* **4**(3), 415–447 (1992)
- Mackey, L., Weiss, D., Jordan, M.I.: Mixed membership matrix factorization. In: *International Conference on Machine Learning*, pp 1–8 (2010)
- Pritchard, J.K., Stephens, M., Donnelly, P.: Inference of population structure using multilocus genotype data. *Genetics* **155**, 945–959 (2000)
- Saddiki, H., McAuliffe, J., Flaherty, P.: GLAD: a mixed-membership model for heterogeneous tumor subtype classification. *Bioinformatics* **31**(2), 225–232 (2015)
- Singh, A.P., Gordon, G.J.: A unified view of matrix factorization models. In: *Lecture Notes in Computer Science*, vol. 5212, pp. 358–373, Springer, Berlin (2008)
- Taddy, M.: Multinomial inverse regression for text analysis. *J. Am. Stat. Assoc.* **108**(503), 755–770, (2013). <https://doi.org/10.1080/01621459.2012.734168>
- Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Sharing clusters among related groups: hierarchical Dirichlet processes. In: *Advances in Neural Information Processing Systems*, vol. 1, MIT Press, Cambridge (2005)
- Weinstein, J.N., Collisson, E.A., Mills, G.B., Shaw, K.R.M., Ozenberger, B.A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J.M., Network CGAR, et al.: The cancer genome atlas pan-cancer analysis project. *Nat. Genet.* **45**(10), 1113 (2013)
- Xiao, H., Stibor, T.: Efficient collapsed Gibbs sampling for latent Dirichlet allocation. In: Sugiyama, M., Yang, Q. (eds.) *Proceedings of 2nd Asian Conference on Machine Learning*, vol. 13, pp. 63–78 (2010)
- Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval–SIGIR '03*, p. 267 (2003)
- Zaslavsky, T.: *Facing Up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes: Face-Count Formulas for Partitions of Space by Hyperplanes*, vol. 154. American Mathematical Society (1975)