Degree Project in Optimization and Systems Theory
Second Cycle, 30 credits

Minimum Cost Distributed Computing using Sparse Matrix Factorization

SEIF HUSSEIN

Stockholm, Sweden 2023
Abstract

Distributed computing is an approach where computationally heavy problems are broken down into more manageable sub-tasks, which can then be distributed across a number of different computers or servers, allowing for increased efficiency through parallelization. This thesis explores an established distributed computing setting, in which the computationally heavy task involves a number of users requesting a linearly separable function to be computed across several servers. This setting results in a condition for feasible computation and communication that can be described by a matrix factorization problem. Moreover, the associated costs with computation and communication are directly related to the number of nonzero elements of the matrix factors, making sparse factors desirable for minimal costs. The Alternating Direction Method of Multipliers (ADMM) is explored as a possible method of solving the sparse matrix factorization problem. To obtain convergence results, extensive convex analysis is conducted on the ADMM iterates, resulting in a theorem that characterizes the limiting points of the iterates as KKT points for the sparse matrix factorization problem. Using the results of the analysis, an algorithm is devised from the ADMM iterates, which can be applied to the sparse matrix factorization problem. Furthermore, an additional implementation is considered for a noisy scenario, in which existing theoretical results are used to justify convergence. Finally, numerical implementations of the devised algorithms are used to perform sparse matrix factorization.
Sammanfattning

Acknowledgements

I would like to thank my supervisor, Saeed Razavi, for his continuous support throughout this master thesis, and particularly the interesting and insightful discussions we have had. Thank you for the possibility to research and work on this interesting topic.

I would also like to thank prof. Carlo Fischione for introducing me to Saeed, and allowing me to do my master thesis at the Division of Network and Systems Engineering at KTH. Thank you for this amazing opportunity.

Lastly, I would like to thank Tova Stroeven for all the support throughout this thesis, and in particular for helping me with the TikZ-environment in LaTeX. Thank you for all the joy that you have brought me, and for making this journey not only bearable, but enjoyable.
Contents

1 Preliminaries 1
  1.1 Optimization Problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
  1.2 Lagrangian and Dual Problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
  1.3 Convexity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
  1.4 Convex Envelope . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  1.5 Subgradients . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  1.6 KKT-Conditions for Linearly-Equality Constrained Optimization Problems . . . . . 3
  1.7 Matrix Notation, Inner Products, and Norms . . . . . . . . . . . . . . . . . . . . . . 4
    1.7.1 The $\ell_0$ Pseudonorm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
    1.7.2 The $\ell_1$ Norm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
    1.7.3 The Frobenius Inner Product and Norm . . . . . . . . . . . . . . . . . . . . . 4

2 Background and Related Work 5
  2.1 Distributed Computing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
  2.2 Matrix Factorization and Algorithms . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
    2.2.1 Alternating Direction Method of Multipliers . . . . . . . . . . . . . . . . . . . 7

3 Problem Description 10
  3.1 Computation Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
  3.2 Connection to Matrix Factorization . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
  3.3 Cost Metrics and Sparseness . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
    3.3.1 Computational Cost . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
    3.3.2 Communication Cost . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
    3.3.3 Sparseness and Matrix Factorization . . . . . . . . . . . . . . . . . . . . . . . 13

4 Nonconvexity of the Matrix Factorization Problem 14
  4.1 Nonconvexity of the Unconstrained Matrix Factorization Problem . . . . . . . . . . 14
  4.2 Convexity in One Variable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
  4.3 Handling the Regularization Terms . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
  4.4 Final Problem Formulation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

5 Solving the Matrix Factorization Problem 19
  5.1 ADMM for the Noise-Free Case . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
    5.1.1 Convergence of subproblems . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
    5.1.2 Convergence of Main Problem . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.3 Implementation of ADMM for Noise-Free Matrix Factorization</td>
<td>28</td>
</tr>
<tr>
<td>5.2 ADMM for the Noisy Case</td>
<td>29</td>
</tr>
<tr>
<td>5.2.1 Convergence of Main Problem</td>
<td>30</td>
</tr>
<tr>
<td>5.2.2 Implementation of ADMM for Noisy Matrix Factorization</td>
<td>31</td>
</tr>
<tr>
<td>6 Results and Simulations</td>
<td>33</td>
</tr>
<tr>
<td>6.1 Phase-Transition and Convergence Plots</td>
<td>33</td>
</tr>
<tr>
<td>6.1.1 Noise-Free Environment</td>
<td>34</td>
</tr>
<tr>
<td>6.2 Noisy Environment</td>
<td>39</td>
</tr>
<tr>
<td>7 Conclusions and Future Work</td>
<td>44</td>
</tr>
<tr>
<td>A Derivation of Closed-Form Solutions to ADMM-Subproblems</td>
<td>48</td>
</tr>
</tbody>
</table>
Chapter 1

Preliminaries

In this section, we introduce some fundamental concepts and notation used throughout the text. Concepts such as norms and convexity will be briefly mentioned, as well as matrix norms and notation. The material used here is mainly from [1, 2].

1.1 Optimization Problem

An optimization problem involves minimizing or maximizing an objective function, subject to certain constraints on the variables. In its general form, an optimization problem can be written as:

$$\min_{x \in S} f(x), \quad (1.1)$$

where $f(x)$ is the objective function and $S$ is the set of feasible solutions, defined by the constraints on the variables. A solution $x^* \in S$ to Problem $\text{(1.1)}$ has the smallest objective value of all $x \in S$.

In this thesis, the focus will be on sets $S$ that can be described as an affine function of $x$, meaning problems will be on the form

$$\min_{x} f(x)$$

subject to $Ax - b = 0$,

for some matrix $A$ and vector $b$.

1.2 Lagrangian and Dual Problem

Consider the optimization problem again

$$\min_{x} f(x), \quad (1.2)$$

subject to $Ax = b$, 

1
where \( x \in \mathbb{R}^n \), and \( A \in \mathbb{R}^{m \times n} \). Problem (1.2) is the so-called **primal problem**, where \( x \) is called the primal variable. The primal problem can be reformulated by partitioning the rows of \( A = [A_1 \ldots A_m]^T \), where \( A_i \in \mathbb{R}^{1 \times n} \). The affine constraint becomes \( Ax = [A_1 x \ldots A_m x]^T = [b_1 \ldots b_m]^T \), and the optimization problem becomes

\[
\min_x f(x) \\
\text{subject to } A_1 x = b_1 \\
\vdots \\
A_m x = b_m.
\]

The idea of the Lagrangian dual is to incorporate the constraints into the objective function as a weighted sum. The Lagrangian function \( L(x, y) \) associated with (1.3) is

\[
L(x, y) = f(x) + \sum_{i=1}^{m} y_i (A_i x - b_i) = f(x) + y^\top (Ax - b),
\]

where \( y \in \mathbb{R}^m \) is the vector of weights or the so-called Lagrange multiplier, associated with the equality constraints in (1.3).

The **dual function**, denoted by \( g(y) \), is defined as the minimum of the Lagrangian function with respect to the primal variable \( x \)

\[
g(y) = \min_x L(x, y) = \min_x (f(x) + y^\top (Ax - b)).
\]

Furthermore, if \( \hat{x} \in \mathbb{R}^n \) is feasible to (1.3) then for \( y \in \mathbb{R}^m \),

\[
g(y) = \min_x L(x, y) \leq L(\hat{x}, y) = f(\hat{x}) + y^\top (A\hat{x} - b) = f(\hat{x}).
\]

In particular, the minimizer \( x^* \) of (1.3) is a feasible point, meaning the dual function serves as a lower bound to the minimum objective value \( f(x^*) \).

To obtain the best lower bound for \( f(x^*) \), one considers the **dual optimization problem** defined as

\[
\max_y g(y) = \max_y \min_x L(x, y).
\]

The maximizer \( y^* \) of (1.4) is called the optimal Lagrange multiplier.

### 1.3 Convexity

A set \( S \subseteq \mathbb{R}^n \) is convex if, for any two points \( x, y \in S \), the line segment connecting them, given by \( tx + (1 - t)y \) for \( 0 \leq t \leq 1 \), also belongs to \( S \). A function \( f : S \to \mathbb{R} \) is convex on \( S \) if its domain \( S \) is a convex set, and for any two points \( x, y \) in its domain, we have:

\[
f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y),
\]

2
for all $0 \leq t \leq 1$. The inequality (1.5) is referred to as Jensen’s inequality. Note that when Jensen’s inequality holds strictly, the function is called strictly convex.

The importance of convexity relates to the optimization problem in (1.1). In particular, if the set $S$ constitutes a convex set, and the function $f$ is convex on $S$, then any local optimum for the corresponding optimization problem is a global optimum. This property is particularly useful since it means that first-order methods can yield conditions for global optimality.

### 1.4 Convex Envelope

The convex envelope of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the largest convex function that is pointwise below $f$. Formally, the convex envelope $f_{\text{env}}$ of $f$ is defined as:

$$f_{\text{env}}(x) = \sup \{ g(x) \mid g \text{ is convex, } g(x) \leq f(x) \forall x \in \mathbb{R}^n \}.$$

### 1.5 Subgradients

Let the function $f : \mathbb{R}^n \to \mathbb{R}$. A vector $g \in \mathbb{R}^n$ is called a subgradient of $f$ at a point $x \in \mathbb{R}^n$ if, for all $y \in \mathbb{R}^n$, we have:

$$f(y) \geq f(x) + g^\top(y - x).$$

The set of all subgradients of $f$ at $x$ is called the subdifferential of $f$ at $x$, denoted by $\partial f(x)$. If $f$ is convex on $S$ and differentiable at $x \in S$, then $\partial f(x) = \{ \nabla f(x) \}$, where $\nabla f(x)$ is the gradient of $f$ at $x$. A function is called subdifferentiable at the point $x$ if the set $\partial f(x)$ is non-empty.

Moreover, a point $x^*$ minimizes $f$, if and only if $f$ is subdifferentiable at $x^*$, and

$$0 \in \partial f(x^*).$$

The inclusion of $0$ in $\partial f(x^*)$ follows from the definition, as

$$f(x) \geq f(x^*) + 0^\top(x - x^*) = f(x^*),$$

for every $x$ in the domain of $f$.

### 1.6 KKT-Conditions for Linearly-Equality Constrained Optimization Problems

Consider a linearly-equality constrained optimization problem

$$\min_x f(x) \quad \text{subject to } Ax = b$$

with the corresponding Lagrangian

$$L(x, y) = f(x) + y^\top(Ax - b).$$
For the case of \( f(x) \) being differentiable, the KKT-points are defined as the set of points \((x^*, y^*)\) satisfying
\[
\frac{\partial L(x^*, y^*)}{\partial x} = f'(x^*) + A^T y^* = 0 \\
\frac{\partial L(x^*, y^*)}{\partial y} = Ax^* - b = 0.
\]

In the case of \( f(x) \) being sub-differentiable but not differentiable, the definition instead becomes
\[
\frac{\partial L(x^*, y^*)}{\partial x} = \partial f(x^*) + A^T y \ni 0 \\
\frac{\partial L(x^*, y^*)}{\partial y} = Ax^* - b = 0.
\]

Here \( 0 \in \partial f(x^*) + A^T y \) means that there exists some \( g \in \partial f(x^*) \) such that \( g + A^T y = 0 \).

### 1.7 Matrix Notation, Inner Products, and Norms

Let \( E \) be an \( m \times n \) matrix. We use \( E_{\cdot j} \) to denote the \( j \)-th column of \( E \), and \( E_{i \cdot} \) to denote the \( i \)-th row of \( E \).

#### 1.7.1 The \( \ell_0 \) Pseudonorm

The \( \ell_0 \) pseudonorm for matrices is defined as the number of nonzero elements in the matrix. Formally, for a matrix \( E \in \mathbb{R}^{m \times n} \), it is given by:
\[
\|E\|_0 = \sum_{i=1}^{m} \sum_{j=1}^{n} I(e_{ij} \neq 0),
\]
where \( I \) is the indicator function, which takes the value 1 if the argument is true, and 0 otherwise.

#### 1.7.2 The \( \ell_1 \) Norm

The \( \ell_1 \) norm can be defined elementwise as the sum of the absolute values of all its elements,
\[
\|E\|_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} |e_{ij}|.
\]

#### 1.7.3 The Frobenius Inner Product and Norm

The Frobenius inner product of two matrices \( A, B \) is defined as
\[
\langle A, B \rangle_F = \operatorname{Tr}(A^T B)
\]

The Frobenius norm is then defined as
\[
\|E\|_F = \sqrt{\langle E, E \rangle_F} = \sqrt{\operatorname{Tr}(E^T E)} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} e_{ij}^2}.
\]
Chapter 2

Background and Related Work

Having established the necessary preliminaries, we can now proceed to discuss the context and background of the work in this thesis.

2.1 Distributed Computing

Distributed computing offers a means to tackle large-scale, computationally taxing tasks by fragmenting the computations into manageable subtasks, which are then allocated to a network of computer nodes. Specifically, MapReduce has emerged as a prevalent framework for distributed computing over the past few years [3]. This framework operates in three stages: Map, Shuffle, and Reduce.

In the Map stage, each node in the computing network undertakes a portion of the larger computation task, executing an intermediate computation based on a predetermined Map function. The outcomes of these computations are then transmitted between nodes in the Shuffle stage, leading up to the final result to be calculated during the Reduce stage via the designated Reduce functions.

While distributed computing frameworks like MapReduce or Spark offer scalability benefits alongside impressive performance and precision, there are significant bottlenecks [4]. Firstly, there is the issue of communication cost; the Shuffle stage can involve extensive internode communication, which, if the communication bandwidth is restricted, can significantly impair performance [5]. Secondly, there is the issue of computation cost; the complexity of tasks and the variance in hardware across nodes can lead to slower-performing nodes or stragglers [6]. This disparity can result in substantial latency in the total computation time. Recent innovations have seen the adoption of coding-based strategies as a means to mitigate the costs associated with communication and computation [7][8]. Other works have explored the potential trade-off between these costs [9][10].

In this work, we consider a special case of distributed computing where the function to be computed is linearly separable. The model is based on the work done in [11]. Specifically, we have that for an input dataset \((D_1, \ldots, D_N)\), the function to be computed is

\[
F(D_1, \ldots, D_L) = f(f_1(D_1), \ldots, f_L(D_L)),
\]

(2.1)
for some linear map $f$. Note that no assumptions on $f_i(D_l)$ are made here; they may be nonlinear functions. Moreover, we consider $K \in \mathbb{N}$ users requesting the computations of functions $F_k$, of the same form as in Equation 2.1, from a master node.

One common application of linearly separable distributed computing is gradient coding [6], where $f_i(D_l)$ is a partial gradient of some function and $f(f_1(D_1), \ldots, f_L(D_L))$ is given by the matrix multiplication

$$
\begin{bmatrix}
1 & \ldots & 1
\end{bmatrix} \begin{bmatrix}
f_1(D_1) \\
\vdots \\
f_L(D_L)
\end{bmatrix} = \sum_{l=1}^L f_i(D_l).
$$

Another application is for a matrix multiplication $A^\top B$ [12], where some matrix $A$ is split into $N$ submatrices $A = [A_1, \ldots, A_N]$. Then each worker $k \in \{1, \ldots, K\}$ is assigned some linear combination of matrices, denoted, $A_k$, and the final result is obtained by appropriately combining the computations $A_k^\top B$ for every worker $k$. In other words, the functions $f_i(D_l)$ are the matrix multiplications $A_k^\top B$, and the function $f$ is the appropriate linear combination done by each worker after computation.

### 2.2 Matrix Factorization and Algorithms

A consequence of the linearly separable setting is the presence of matrix multiplications. In particular, the previously mentioned linear operator $f$ can be described as some matrix $F$ that acts on a vector containing the elements $f_i(D_l)$. Additionally, if we consider a MapReduce framework where the Shuffle step contains internode transmissions in the form of a linear map, then in the Reduce step, these transmissions are again linearly combined to recover the final computation. If we denote $D$ as the matrix representation of the Reduce map and $E$ as the matrix representation of the Shuffle map, the Shuffle and Reduce steps are essentially the composition of the two linear maps as $DE$.

Consequently, a fault-free computation implies that $F$ and $DE$ are the same linear map acting on the elements $f_i(D_l)$, meaning $F = DE$. In other words, given a linear operator $F$, the problem of finding a feasible transmission and computation pair $(D, E)$ is equivalent to solving the matrix factorization problem $F = DE$. More formally, given $F \in \mathbb{R}^{K \times L}$, we seek to find $D \in \mathbb{R}^{K \times N}$ and $E \in \mathbb{R}^{N \times L}$ such that the function

$$
f(D, E) = \|F - DE\|_F^2.
$$

is minimized. This function obtains its minimum at $F = DE$, whenever factorization is feasible. It is also possible to pose the matrix factorization problem as a feasibility problem, meaning we seek to solve the following optimization

$$
\min_{D, E} 0
$$

subject to $F - DE = 0$,

where the cost function is simply the zero value and the factorization is set to be the only constraint of the optimization problem. By introducing an auxiliary variable $Z \in \mathbb{R}^{K \times L}$, we can reformulate
the problem as

$$\min_{D,E,Z} \|Z\|_F^2$$

subject to $F - DE = Z$.

Matrix factorization is widely used in areas such as machine learning or signal processing. There are several methods that yield exact factorization, such as SVD or Cholesky factorization, but these methods might not be suitable if the dimensions become too large or if certain properties, such as sparsity or non-negativity [13, 14], are desired alongside the factorization. In these cases, one might consider the additional constraint that $D, E \in S$, where $S$ is some set with the desired properties. Such problems might be of the form

$$\min_{D,E \in S} g(D, E)$$

where $g$ is some function that represents a desired quantity to minimize. For example, $S$ might be the set of non-negative matrices, and $g(D, E)$ be defined as (2.2). Such a problem can be formulated as

$$\min_{D,E} \|F - DE\|_F^2$$

subject to $D \geq 0$

$E \geq 0$.

These types of matrix factorization problems are nonconvex and, in general, NP-hard and intractable [15]. Despite its intractability, unconstrained matrix factorization, as considered here, is a biconvex problem, meaning it is convex in both $D$ and $E$, but separately. The property of biconvexity makes alternating methods desirable, as minimizing one variable at a time might lead to tractable subproblems.

Some common algorithms for constrained matrix factorization are Proximal Alternating Linearized Minimization (PALM), Alternating Least Squares (ALS), Majorization-Minimization (MM), and Alternating Direction Method of Multipliers (ADMM) [16, 17, 18, 19]. Note that MM is not an algorithm but a recipe to construct one. We consider ADMM as the primary method to approach matrix factorization in this work.

### 2.2.1 Alternating Direction Method of Multipliers

In this section, the Alternating Direction Method of Multipliers (ADMM for short) is presented, based on the formulation in [13].

Consider the following optimization problem,

$$\min_{x,z} f(x) + g(z)$$

subject to $Ax + Bz = c$,

where $f : \mathbb{R}^n \to \mathbb{R}$, and $g : \mathbb{R}^m \to \mathbb{R}$ are convex functions. Note that the objective function of the optimization problem is separable in $x$ and $z$. 
Define the augmented Lagrangian as
\[ L_\rho(x, z, y) = f(x) + g(z) + y^\top (Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2, \]  
(2.4)
where \( \rho > 0 \). The idea of ADMM is to perform dual ascent but staggering the dual update as
\[ G(z, y) = \inf_x L_\rho(x, z, y), \]
and (order may be swapped)
\[ G(y) = \inf_z L_\rho(x^*, z, y). \]

The primal updates are simply the minimizing primal variable for the dual function, meaning the gradient ascent algorithm at iteration \( k \) has the form
\[
\begin{align*}
x^{k+1} &= \arg\min_x L(x, z^k, y^k) \quad \text{(2.5)} \\
z^{k+1} &= \arg\min_z L(x^{k+1}, z, y^k) \quad \text{(2.6)} \\
y^{k+1} &= y^k + \alpha^k (Ax^{k+1} + Bz^{k+1} - c),
\end{align*}
\]

since \( \nabla G(y) = Ax^* + Bz^* - c \).

The natural choice of parameter \( \alpha^k \) is \( \rho \), since this choice results in dual feasibility with respect to \( z^{k+1} \), in every iteration. It can be shown as follows,
\[
z^{k+1} = \arg\min_z L(x^{k+1}, z, y^k) \implies \nabla_z L(x^{k+1}, z^{k+1}, y^k) = 0,
\]
where
\[
\nabla_z L(x^{k+1}, z^{k+1}, y^k) = \nabla g(z^{k+1}) + B^\top (y^k + \rho (Ax^{k+1} + Bz^{k+1} - c)) = 0.
\]
The condition for (partial) dual feasibility in step \( k + 1 \) is
\[
\nabla g(z^{k+1}) + B^\top y^{k+1} = 0,
\]
meaning \( y^{k+1} = y^k + \rho (Ax^{k+1} + Bz^{k+1} - c) \) satisfies the feasibility criteria exactly.

**Convergence of ADMM**

Two assumptions are made for the convergence of ADMM. The first one is on the functions \( f \) and \( g \), and it states that they must be closed, proper, and convex. The assumption is equivalent to the condition that the set
\[
\{(x, z, t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \mid f(x) \leq t, g(z) \leq t\}
\]
is closed, convex, and nonempty. The second assumption is that the Lagrangian of (2.3) has a saddle point, meaning there exists a triplet \((x^*, z^*, y^*)\) such that
\[
L(x^*, z^*, y) \leq L(x^*, z^*, y^*) \leq L(x, z, y^*),
\]
for every $x, y, z$.

With these assumptions in place, the iterates converge in feasibility, meaning
\[
\lim_{k \to \infty} Ax^k + Bz^k - c = 0,
\]
and to the minimizing value of (2.3). Finally, $y^k$ converges to a dual optimal point as $k \to \infty$.

**Scaled Form of ADMM**

Here we briefly go through the derivation of the scaled form of ADMM which makes the iterations more convenient to express and calculate.

**Proposition 2.2.1.** The scaled form of ADMM has a Lagrangian of the form
\[
L_\rho(x, z, u) = f(x) + g(z) + \frac{(\rho/2)}{2} \|Ax + Bz - c + u\|^2_2 + r(u),
\]
where $u = (1/\rho)y$, and $r(u)$ is some function only dependent on $u$.

**Proof.** Rewrite the Lagrangian as
\[
L_\rho(x, z, y) = f(x) + g(z) + y^\top (Ax + Bz - c) + (\rho/2)\|Ax + Bz - c + (1/\rho)y - (1/\rho)y\|^2_2
\]
\[
= f(x) + g(z) + y^\top (Ax + Bz - c) + (\rho/2)\|Ax + Bz - c + (1/\rho)y\|^2_2
\]
\[
- (\rho/2)(1/\rho)y^\top (Ax + Bz - c) - (\rho/2)(Ax + Bz - c)^\top (1/\rho)y - (\rho/2)(1/\rho)^2 y^\top y
\]
\[
= f(x) + g(z) + (\rho/2)\|Ax + Bz - c + (1/\rho)y\|^2_2 - \frac{y^\top y}{2\rho},
\]
where we can identify $r(u) = -\frac{y^\top y}{2\rho} = -(\rho/2)\|u\|^2_2$.

With the scaled form introduced, we can compactly express the ADMM updates as
\[
x^{k+1} = \arg\min_x f(x) + (\rho/2)\|Ax + Bz^k - c + u^k\|^2_2
\]
\[
z^{k+1} = \arg\min_z g(z) + (\rho/2)\|Ax^{k+1} + Bz - c + u^k\|^2_2
\]
\[
u^{k+1} = u^k + Ax^{k+1} + Bz^{k+1} - c
\]

**ADMM for Biconvex Problems**

Due to the biconvex property of the matrix factorization problem, as discussed in Section 2.2, it is of interest to formulate the ADMM updates for these types of problems, specifically. Following the model in [18], we have some biconvex objective function $F(x, z)$ which is convex when considering $x$ and $z$ separately, and a bi-affine constraint $G(x, z) = 0$, which is affine when considering $x$ and $z$ separately. Then the ADMM updates become
\[
x^{k+1} = \arg\min_x F(x, z^k) + (\rho/2)\|G(x, z^k) + u^k\|^2_2
\]
\[
z^{k+1} = \arg\min_z F(x^{k+1}, z) + (\rho/2)\|G(x^{k+1}, z) + u^k\|^2_2
\]
\[
u^{k+1} = u^k + G(x^{k+1}, z^{k+1}).
\]
The biconvexity of $F(x, z)$, and biaffinity of $G(x, z)$ result in convex subproblems, which may prove to be useful in deriving convergence results related to the nonconvex overall problem.
Chapter 3
Problem Description

We consider a distributed computing setting from [11]. The setup consists of \( N \) computing nodes, where a certain output function is to be calculated from a set of input datasets \( D_1, \ldots, D_L \) for each user \( k \in \{1, \ldots, K\} \). Specifically, the output function \( F_k(D_1, \ldots, D_L) \) is linearly separable. The computation is broken down into three phases, similar to the MapReduce framework. However, instead of considering a Shuffle step that broadcasts intermediate computations between the nodes, the setting consists of a set of users requesting functions \( F_k \) to be computed for each user \( k \), and a server \( n \in \{1, \ldots, N\} \) broadcasting to some specific set of users instead. These phases result in quantifiable computation costs tied to the number of computations carried out by node \( n \). Similarly, the communication costs are defined in relation to the number of users \( k \) broadcasted to by node \( n \).

Furthermore, the authors of [11] present a criterion for feasible computation communication that, given the context of linear separability, can be framed as a matrix factorization problem. The computation and communication costs can be directly related to the sparsity of the solutions to the matrix factorization problem.

3.1 Computation Procedure

Let \( \{D_1, D_2, \ldots, D_L\} \) be the set of input datasets. Then each user \( k \in \{1, \ldots, K\} \) requests the linearly separable function

\[
F_k(D_1, D_2, \ldots, D_L) = \sum_{l=1}^{L} f_{k,l}f_l(D_l),
\]

where \( f_{k,l} \) are scalar coefficients, and \( f_l(D_l) \) are subfunctions evaluated at points corresponding to some single dataset. Note that these subfunctions are in general not linear.

In the described setting, three phases are considered. In the demand phase, each user \( k \) requests \( F_k \) from a master node. The master node proceeds to deduce the separable form of this function according to (3.1). In the following assignment and computation phase, the master
node assigns server sets \( W_l \subset \{1, \ldots, N\} \) to calculate the corresponding subfunction \( f_l(D_l) \). In the transmission and decoding phase, each server \( n \) broadcasts the linear combination

\[
z_n = \sum_{l=1}^{L} e_{n,l} f_l(D_l),
\]

where \( n \in \{1, \ldots, N\} \), and \( e_{n,l} \) are the coefficients corresponding to the specific server. Note that \( e_{n,l} \) can be zero, which simply means that a particular server \( n \) is not assigned to calculate \( f_l(D_l) \). This linear combination is then transmitted to a subset of users denoted \( T_n \subset \{1, \ldots, K\} \), where the index specifies that server \( n \) broadcasts to \( T_n \) which is a subset of users specific to this particular server. In the decoding part of the phase, the user \( k \) forms the linear combination

\[
F'_k = \sum_{n=1}^{N} d_{k,n} z_n,
\]

where \( d_{k,n} \) are decoding coefficients decided by the master node. Successful decoding for user \( k \) means that \( F'_k = F_k \). The computation procedure is shown in Figure 3.1.

\[
\text{Figure 3.1: Visualization of a general network consisting of } K \text{ users, } N \text{ servers and } L \text{ datasets.}
\]

### 3.2 Connection to Matrix Factorization

In this section, it is shown that the system model described previously can be expressed as a matrix factorization problem. The result is as follows

**Proposition 3.2.1.** Assume that the user requested function \( F_k \) is linearly separable. Then the condition of successful decoding \( F'_k = F_k \), for every \( k \in \{1, \ldots, K\} \), implies the matrix factorization
\( F = DE \), where the entries of \( F \) are the scalar coefficients \( f_{k,l} \), and the entries of the matrices \( E \) and \( D \) are the encoding coefficients \( e_{n,l} \) and decoding coefficients \( d_{k,n} \), respectively.

**Proof.** By definition, 
\[
F'_k = \sum_{n=1}^{N} \sum_{l=1}^{L} d_{k,n} e_{n,l} f_l(D_l).
\]
The sum can be compactly expressed as
\[
F'_k = d_k^\top E w,
\]
where \( d_k = [d_{k,1}, d_{k,2}, \ldots, d_{k,N}]^\top \), \( w = [f_1(D_1), f_2(D_2), \ldots, f_L(D_L)]^\top \), and \( E \in \mathbb{R}^{N \times L} \) is the matrix with row \( n \in \{1, \ldots, N\} \) containing the encoding coefficients \( e_{n,l} \) for \( l \in \{1, \ldots, L\} \).

Similarly, we can express
\[
F_k = f_k^\top w,
\]
where \( f_k = [f_{k,1}, f_{k,2}, \ldots, f_{k,L}]^\top \).

The condition for successful decoding, \( d_k^\top E w = f_k^\top w \), for every \( k \in \{1, \ldots, K\} \), can then be expressed as
\[
Fw = DEw,
\]
where \( F \in \mathbb{R}^{K \times L} \) is a matrix with row \( k \in \{1, \ldots, K\} \) containing \( f_k^\top \), and \( D \in \mathbb{R}^{K \times N} \) with row \( k \in \{1, \ldots, K\} \) containing \( d_k^\top \).

This equality holds for an arbitrary linear combination of requested functions by the users, meaning an arbitrary \( w \). We obtain the result
\[
F = DE.
\]

### 3.3 Cost Metrics and Sparseness

Here, the cost metrics associated with the distributed computing problem are defined, and related to the notion of sparseness. This connection allows the formulation of an optimization problem that describes successful decoding with minimal communication and computational cost.

#### 3.3.1 Computational Cost

By definition, \( \mathcal{W}_l \) is the subset of servers that calculate the subfunction \( f_l(D_l) \). This set corresponds to the nonzero encoding coefficients \( e_{n,l} \) for a particular \( l \). Recall that \( E \) is the \( N \times L \) matrix with entries being the encoding coefficients \( e_{n,l} \). Consequently, the number of nonzero elements in column \( l \) of \( E \) indicates how many servers are assigned to calculate a function \( f_l(D_l) \). In other words,
\[
|\mathcal{W}_l| = \|E_{:,l}\|_0.
\]

Summing this quantity for all columns \( l \) in \( E \) gives the total number of servers assigned to perform a calculation of a subfunction.

**Definition 3.3.1.** The normalized computational cost \( \gamma \) is defined as the total number of servers assigned to calculate some subfunction divided by \( NL \), meaning
\[
\gamma = \frac{\sum_{l=1}^{L} |\mathcal{W}_l|}{NL}.
\]
This definition of $\gamma$ allows the computational cost to be expressed in terms of the $\ell_0$ norm, since combining equations $[3.4]$ and $[3.5]$ yields
\[
\gamma = \frac{\sum_{l=1}^L |W_l|}{NL} = \frac{\sum_{l=1}^L \|E_{nl}\|_0}{NL} = \frac{\|E\|_0}{NL}.
\]

### 3.3.2 Communication Cost

In this case, we consider the set $T_n$, which is the subset of users that a particular server $n$ broadcasts the linear combination described in Equation $[3.2]$. This set corresponds to the nonzero decoding coefficients for a particular server $n$. Now recall that $D$ is the $K \times N$ matrix with entries being the decoding coefficients $d_{k,n}$, so the number of nonzero elements in a column $n$ indicates how many users that a server $n$ broadcasts to. We obtain
\[
|T_n| = \|D_{*n}\|_0, \tag{3.6}
\]
and proceeding in the same manner as previously, the normalized communication cost can be defined.

**Definition 3.3.2.** The normalized communication cost $\delta$ is defined as the total number of users that a server $n$ broadcasts to, divided by $KN$.
\[
\delta = \frac{\sum_{n=1}^N |T_n|}{KN} \quad \tag{3.7}
\]
Again, one can see that
\[
\delta = \frac{\sum_{n=1}^N |T_n|}{KN} = \frac{\sum_{n=1}^N \|D_{*n}\|_0}{KN} = \frac{\|D\|_0}{KN}. \tag{3.8}
\]

### 3.3.3 Sparseness and Matrix Factorization

With the cost metrics from Definition $[3.3.1]$ and Definition $[3.3.2]$, we now recall Proposition $[3.2.1]$ that tells us that successful, or feasible, decoding gives us a matrix factorization problem. In particular, the $\ell_0$ norm of these factors $D, E$ are of interest for a successful recovery with minimal cost. There are several ways to formulate the condition of sparse feasible decoding as an optimization problem, and here we consider two formulations, namely
\[
\min_{D,E} \lambda_D \|D\|_0 + \lambda_E \|E\|_0 \quad \tag{3.9}
\]
subject to $F = DE,$
and
\[
\min_{D,E} \frac{1}{2} \|F - DE\|_F^2 + \lambda_D \|D\|_1 + \lambda_E \|E\|_1. \quad \tag{3.10}
\]
The weights $\lambda_D, \lambda_E > 0$ penalize dense matrices $D, E$ and introduce a trade-off between accuracy and sparseness, as the nonconvex nature of the problem might lead to difficulties finding local minima with an accurate enough approximation of $F$ for a given weighting $(\lambda_D, \lambda_E)$. 13
Chapter 4

Nonconvexity of the Matrix Factorization Problem

In this chapter, we examine the matrix factorization problem more closely. The inherent nonconvexity of the problem of factorizing a matrix $F$ into factors $D, E$ simultaneously is established. Furthermore, the nonconvexity of the pseudo $\ell_0$-norms are also shown. However, the matrix factorization problem is shown to be convex in considering exactly one of the variables $D$ or $E$. Additionally, for the task of obtaining a convex relaxation of the $\ell_0$-pseudonorms, we find that the convex $\ell_1$-norm is a natural candidate for a tight bound.

The property of convexity in exactly one of the variables $D$ or $E$ suggests that algorithms with inherent separability and alternation, such as ADMM, are promising candidates for the matrix factorization problem.

4.1 Nonconvexity of the Unconstrained Matrix Factorization Problem

Consider the unconstrained problem,

$$\min_{D,E} f(D, E) = \frac{1}{2} \| F - DE \|_F^2. \tag{4.1}$$

It is straightforward to see that the optimal solution to this problem is any pair of matrices $D, E$ whose product $DE = F$, if such matrices exist. If this problem is convex, then the set of local minima necessarily constitute a convex set. The following proposition shows how this condition fails for (4.1).

**Proposition 4.1.1.** Problem $(4.1)$ is nonconvex in the variables $D$ and $E$.

**Proof.** Consider a solution of the form $XY = F$. Clearly, there exists another solution $\bar{X}\bar{Y}$, where $\bar{X} =XA$, $\bar{Y} =A^TY$, and $A \in \mathbb{R}^{N \times N}$ satisfies that $AA^T = I_{N \times N}$. If these solutions lie in a convex set, then the convex combination of the solutions must yield globally optimal solutions to $(4.1)$ as
well. In other words, the function \( f(D, E) \) must be constant on the line segment \( t(\bar{X}, \bar{Y}) + (1 - t)(X, Y) \) for \( t \in (0, 1) \). For a given \( t \in (0, 1) \), we obtain a new point \( (t\bar{X} + (1 - t)X, t\bar{Y} + (1 - t)Y) \), which implies that

\[
F = (tXA + (1 - t)X)(tA^\top Y + (1 - t)Y) = t^2 XY + (1 - t)^2 XY + t(1 - t)(XAY + XA^\top Y).
\]

This equality holds for every \( t \in (0, 1) \) if and only if \( XAY = XA^\top Y = XY \), which is not true in general if \( A \) is orthogonal. An example is \( A \) being a rotation matrix. Thus our solutions do not lie in a convex set, which disproves the convexity of (4.1). Note that this result also implies that the set of \( \{D, E\} \) such that \( F = DE \) is a nonconvex set.

\[ \blacksquare \]

### 4.2 Convexity in One Variable

Now, we show that formulating Problem (4.1) as an optimization problem in one variable yields a convex problem.

**Proposition 4.2.1.** The function given by \( f(E) = \|F - DE\|^2_F \) is convex.

**Proof.** Using the definition of convexity, we want to show that

\[
f(tE_1 + (1 - t)E_2) \leq tf(E_1) + (1 - t)f(E_2),
\]

or equivalently,

\[
f(tE_1 + (1 - t)E_2) - tf(E_1) - (1 - t)f(E_2) \leq 0.
\]

Insertion gives

\[
\|t(F - DE_1) + (1 - t)(F - DE_2)\|^2_F - t\|F - DE_1\|^2_F - (1 - t)\|F - DE_2\|^2_F \geq 0.
\]

We use the definition of the Frobenius norm to expand the positive term

\[
\|t(F - DE_1) + (1 - t)(F - DE_2)\|^2_F = \sum_{i,j} (t(F - DE_1)_{ij} + (1 - t)(F - DE_2)_{ij})^2
\]

\[
= t^2 \sum_{i,j} (F - DE_1)_{ij}^2 + (1 - t)^2 \sum_{i,j} (F - DE_2)_{ij}^2
\]

\[
+ 2t(1 - t) \sum_{i,j} (F - DE_1)_{ij} (F - DE_2)_{ij}
\]

\[
= t^2 \|F - DE_1\|^2_F + (1 - t)^2 \|F - DE_2\|^2_F
\]

\[
+ 2t(1 - t)\langle F - DE_1, F - DE_2 \rangle_F.
\]

By the Cauchy-Schwartz inequality, we have

\[
2t(1 - t)|\langle F - DE_1, F - DE_2 \rangle| \leq 2t(1 - t)\|F - DE_1\|_F \|F - DE_2\|_F.
\]
So we get that (4.2) is bounded from above by
\[ t^2 \|F - DE_1\|_F^2 + (1 - t)^2 \|F - DE_2\|_F^2 + 2t(1 - t)\|F - DE_1\|_F\|F - DE_2\|_F \\
- t\|F - DE_1\|_F^2 - (1 - t)\|F - DE_2\|_F^2. \]

Simplifying this expression gives
\[ -t(1 - t)(\|F - DE_1\|_F^2 + \|F - DE_2\|_F^2 - 2\|F - DE_1\|_F\|F - DE_2\|_F), \]
or equivalently,
\[ -t(1 - t)(\|F - DE_1\|_F - \|F - DE_2\|_F)^2, \]
which is clearly less than or equal to 0. In other words, (4.2) is bounded from above by 0 which proves the claim.

To show that \( f(D) = \|F - DE\|_F^2 \) is convex, the proof is identical.

Note that these results imply that the set of \{D\} and the set of \{E\}, such that \( F = DE \), constitute convex sets when considered separately. Indeed, in one variable this equation is simply an affine constraint on the variables \( D \) or \( E \), and affine constraints constitute convex sets.

### 4.3 Handling the Regularization Terms

We showed that the quadratic term in (3.10) is biconvex, meaning it is convex in exactly one of the variables \( D, E \). However, the pseudo \( l_0\)-norm is not a convex function. To see that it is nonconvex, one can use Jensen’s inequality. Set \( f(X) = \|X\|_0 \), then assume that \( f(X_1) = \|X_1\|_0 = \alpha \), and \( f(X_2) = \|X_2\|_0 = \beta \), which gives \( f(tX_1 + (1 - t)X_2) \) as
\[
\|tX_1 + (1 - t)X_2\|_0 = \begin{cases} 
\alpha + \beta, & \text{if } X_1 \text{ and } X_2 \text{ share no overlap in nonzero elements}, \\
Q & \text{otherwise},
\end{cases}
\]

where \( 0 \leq \max(\alpha, \beta) \leq Q \leq \alpha + \beta \). Now \( tf(X_1) + (1 - t)f(X_2) = t\|X_1\|_0 + (1 - t)\|X_2\|_0 = t\alpha + (1 - t)\beta \), resulting in the difference \( f(tX_1 + (1 - t)X_2) - tf(X_1) - (1 - t)f(X_2) \) being
\[
\|tX_1 + (1 - t)X_2\|_0 - t\|X_1\|_0 - (1 - t)\|X_2\|_0 = \begin{cases} 
(1 - t)\alpha + \beta, & \text{if } X_1 \text{ and } X_2 \text{ share no overlap in nonzero elements}, \\
Q - ta - (1 - t)\beta & \text{otherwise}.
\end{cases}
\]

Note that for the case of overlap, considering \( Q = \max(\alpha, \beta) \) yields
\[
\max(\alpha, \beta) - ta - (1 - t)\beta = \begin{cases} 
(1 - t)(\alpha - \beta) & \text{if } \alpha > \beta, \\
t(\beta - \alpha) & \text{if } \alpha < \beta, \\
0 & \text{if } \alpha = \beta.
\end{cases}
\]

Consequently, for \( \{X_1, X_2\} \) such that \( Q \geq \max(\alpha, \beta) \), a lower bound is obtained for \( f(tX_1 + (1 - t)X_2) - tf(X_1) - (1 - t)f(X_2) \) as
\[
\|tX_1 + (1 - t)X_2\|_0 - t\|X_1\|_0 - (1 - t)\|X_2\|_0 \geq 0.
\]

16
In other words, the pseudo $\ell_0$-norm is not convex, as Jensen’s inequality fails to hold for the set $\{X_1, X_2\}$ such that $Q \geq \max(\alpha, \beta)$.

One way to enforce convexity in one variable without losing the property of sparsity is to relax the pseudo $\ell_0$-pseudonorm into the convex $\ell_1$-norm. Using the scalability property, as well as the triangle inequality for the $\ell_1$-norm, it is straightforward to see that Jensen’s inequality

$$
\|tX_1 + (1 - t)X_2\|_1 \leq \|tX_1\|_1 + \|(1 - t)X_2\|_1 = t\|X_1\|_1 + (1 - t)\|X_2\|_1
$$

holds for the $\ell_1$-norm. One can also show that for a matrix $X$, with $\|X\|_\infty \leq 1$, the $\ell_1$-norm $\|X\|_1$ is the convex envelope of the $\ell_0$-norm $\|X\|_0$ [20]. In other words, formulating the problems (3.9) (3.10) in terms of the $\ell_1$-norm is a natural biconvex relaxation. See Figure 4.1 for an intuitive illustration for the claim of $\ell_1$ being the convex envelope of $\ell_0$.

Figure 4.1: Illustrative plot of the norm unit balls in $\mathbb{R}^2$, reinforcing the intuition behind the $\ell_1$-norm being the convex envelope of the $\ell_0$-pseudo norm.
### 4.4 Final Problem Formulation

Following the convexity analysis in this chapter and the results obtained, we suggest the following two steps approach problems (3.9), (3.10). The first being to simply replace $\ell_0$ with $\ell_1$ in the objective functions, meaning we consider the problems

$$\min_{D, E} \lambda_D \|D\|_1 + \lambda_E \|E\|_1$$  \hspace{1cm} (4.3)

subject to $F = DE$,

and

$$\min_{D, E} \frac{1}{2} \|F - DE\|_F^2 + \lambda_D \|D\|_1 + \lambda_E \|E\|_1.$$  \hspace{1cm} (4.4)

The main difference between the problems is the feasible region of the solution obtained. In Problem (4.3), the feasible region is the set of matrices $D, E$ such that $F = DE$. However, Problem (4.4) is unconstrained, meaning the set of feasible solutions is less restrictive. Both formulations are useful in their own way, since if one expects a noise-free setting, and desires an exact recovery of the matrix $F$ with sparse factors, Problem (4.3) may be a more useful way to attempt solving the matrix factorization problem. If, however, noise is present and one seeks to approximate $F$ with sparse factors, Problem (4.4) is a more robust approach.

An important note is that these approaches do not deal with the inherent nonconvexity of the two-variable matrix factorization problem. However, in the context of alternating methods, such as ADMM, one obtains convex subproblems (2.5), (2.6) in the primal variable updates. Since the variables are updated independently, and as we have shown the convexity of the problems in one variable, it follows that the subproblems are convex. So the second step is simply to use an alternating method to solve the optimization problems (4.3), (4.4). In the context of this work, ADMM will be the method considered.
Chapter 5

Solving the Matrix Factorization Problem

We are now ready to apply the results of previous discussions to the ADMM framework. Two cases will be considered, namely Problem 4.3 for the noise-free case and Problem 4.4 for the noisy case.

5.1 ADMM for the Noise-Free Case

We consider the equality-constrained problem

\[
\min_{D,E} \lambda_D \|D\|_1 + \lambda_E \|E\|_1 + \frac{\rho}{2} \|F - DE + \Lambda^{(k)}/\rho\|^2_F - \frac{\|\Lambda\|^2_F}{2\rho},
\]

subject to \(F = DE\).

As previously shown, the objective function is convex in \(D\) for every \(E\) and analogously for \(E\). The equality constraint is affine in \(D\) for every \(E\), and analogously for \(E\). These observations motivate the previously presented biconvex formulation of the ADMM-iterates from section 2.2.1. The augmented Lagrangian for this problem becomes

\[
L_{\rho}(D, E, \Lambda) = \lambda_D \|D\|_1 + \lambda_E \|E\|_1 + \frac{\rho}{2} \|F - DE + \Lambda^{(k)}/\rho\|^2_F - \frac{\|\Lambda\|^2_F}{2\rho},
\]

where \(\rho > 0\). The ADMM-updates become

\[
D^{(k+1)} = \arg\min_{D} \lambda_D \|D\|_1 + \frac{\rho}{2} \|F - DE^{(k)} + \Lambda^{(k)}/\rho\|^2_F
\]

\[
E^{(k+1)} = \arg\min_{E} \lambda_E \|E\|_1 + \frac{\rho}{2} \|F - D^{(k+1)}E + \Lambda^{(k)}/\rho\|^2_F
\]

\[
\Lambda^{(k+1)} = \Lambda^{(k)} + \rho(F - D^{(k+1)}E^{(k+1)})
\]

5.1.1 Convergence of subproblems

For the objective functions associated with Problems (5.3), (5.4) it can be noted that they are convex. Additionally, the functions are closed due to their continuity and the fact that they approach...
∞ for any sequence of $D, E$ approaching $\infty$. Furthermore, the functions are considered proper as they have non-empty domains and do not attain the value of $-\infty$. Consequently, the first assumption for ADMM to converge is satisfied for the subproblems. Note, however, that the subproblems are unconstrained. In order to apply ADMM to these problems, we introduce auxiliary variables $P_D, P_E$, and re-formulate them as

$$D^{(k+1)} = \arg\min_{D, P_D} \lambda_D\|P_D\|_1 + (\rho/2)\|F - DE + \Lambda^{(k)}/\rho\|_F^2 \tag{5.6}$$

subject to $D = P_D$.

and

$$E^{(k+1)} = \arg\min_{E, P_E} \lambda_E\|P_E\|_1 + (\rho/2)\|F - D^{(k+1)}E + \Lambda^{(k)}/\rho\|_F^2 \tag{5.7}$$

subject to $E = P_E$.

Note that the introduction of auxiliary variables does not affect the first assumption of ADMM convergence. We are now ready to analyze the respective Lagrangians. We consider the case of $D^{(k+1)}$, since the case of $E^{(k+1)}$ follows. We momentarily drop the dependence on $k$, and define the Lagrangian of $(5.6)$ as

$$L(D, P_D, \Lambda_D) = \lambda_D\|P_D\|_1 + (\rho/2)\|F - DE + \Lambda/\rho\|_F^2 + (\Lambda_D^T, D - P_D)_F. \tag{5.8}$$

For the second assumption, we have to show that there exists points $(D^*, P_D^*, \Lambda_D^*)$ such that

$$L(D^*, P_D^*, \Lambda_D^*) \leq L(D^*, P_D^*, \Lambda_D^*), \tag{5.9}$$

for every $(D, E, \Lambda_D)$. Note that points $(D^*, E^*, \Lambda_D^*)$ that satisfy the KKT-system for $(5.6)$

$$0 \in \partial\|P_D\|_1 - \Lambda_D^*$$

$$0 = \Lambda_D - \rho(F - D^*E + \Lambda/\rho)E^T$$

$$0 = D^* - P_D^*,$$

also satisfy the saddle point condition $(5.9)$. To see this, note that $D^* = P_D^* \implies L(D^*, P_D^*, \Lambda_D^*) = L(D^*, P_D^*, \Lambda_D^*) \leq L(D^*, P_D^*, \Lambda_D^*)$ for any $\Lambda_D^*$. The remaining KKT-conditions simply minimize $D, P_D$ for a given $\Lambda_D$ which gives $L(D^*, P_D^*, \Lambda_D^*) \leq L(D, P_D, \Lambda_D)$. What is left is to show that the set of solutions to the KKT-system is non-empty. To see this, note that the objective function of $(5.3)$ is lower-bounded, and specifically it is coercive in $D$, meaning the objective function diverges to $\infty$ as $\|D\|_F \rightarrow \infty$ which implies that the set of solutions to the corresponding optimization problem is non-empty. In particular, there exists at least one global minimizer $[21]$ page 25]. By extension, the KKT system for $(5.6)$ is non-empty, as it is necessary and sufficient for a global minimizer to solve the KKT-system in a convex optimization problem with affine equality constraints (and no inequality constraints). Obtaining a $D^*$ that solves $(5.3)$ and setting $P_D = D^*$, as well as $\Lambda_D = \rho(F - D^*E(k) + \Lambda(k)/\rho)E(k)^T$ yields a KKT-point. Consequently, the Lagrangian given by $(5.8)$ has a saddle point. Identical arguments can be made for the case of considering $E$ instead of $D$.

Thus, we have shown that the convergence analysis for ADMM discussed in $2.2.1$ applies to subproblems $(5.6), (5.7)$. In the following proposition, we make the connection between Problems $(5.3), (5.4)$ and $(5.6), (5.7)$.
Proposition 5.1.1. For \((D^*, P^D, A^D_D)\) that is a KKT-point of (5.6) the corresponding \(D^*\) also minimizes (5.3).

Proof. Define the Lagrangian for (5.6) as

\[
L(D, P, A_D) = \lambda_D \|P_D\|_1 + (\rho/2)\|F - DE^{(k)} + (\Lambda^{(k)}/\rho)\|^2_F + \langle A_D D - P_D \rangle_F
\]

Then the KKT-conditions become

\[
\begin{align*}
0 \in \partial \|P_D^*\|_1 - A_D^* \\
0 = \Lambda_D^* - \rho(F - D^*E^{(k)} + (\Lambda^{(k)}/\rho)E^{(k)})^T \\
0 = D^* - P^D^*
\end{align*}
\]

and straightforward substitutions give

\[
0 \in \partial \|D^*\|_1 - \rho(F - D^*E^{(k)} + (\Lambda^{(k)}/\rho)E^{(k)})^T,
\]

which is precisely the optimality condition for (5.3).

The result implies that we may assume that \(D\) obtained by the ADMM-scheme for (5.6) satisfies the optimality condition for (5.3). For the case of \(E\) and (5.4), the analysis is identical and therefore omitted.

5.1.2 Convergence of Main Problem

We turn our attention to the original problem of

\[
\min_{D, E} \lambda_D \|D\|_1 + \lambda_E \|E\|_1
\]

subject to \(F = DE\).

The augmented Lagrangian is defined as

\[
L_\rho(D, E, A) = \lambda_D \|D\|_1 + \lambda_E \|E\|_1 + (\rho/2)\|F - DE + (\Lambda/\rho)\|^2_F - \|A\|^2/(2\rho).
\]

For the analysis in this section, we consider ADMM iterates on an augmented Lagrangian that is strongly convex with respect to \(D\) and \(E\), separately. However, the current formulation of the Lagrangian is not strongly convex in \(D\) or \(E\). To see this, assume that \(E\) is rank-deficient, so that there exists \(D_1, D_2\) such that \(F = D_1E = D_2E\), where \(D_1 \neq D_2\). Then we have (omitting dependence on \(E, A\)) that \(L_\rho(tD_1 + (1-t)D_2) - tL_\rho(D_1) - (1-t)L_\rho(D_2)\) is

\[
\lambda_D \|tD_1 + (1-t)D_2\|_1 + (\rho/2)\|0 + (\Lambda/\rho)\|^2_F - t\lambda_D \|D_1\|_1 - (1-t)\lambda_D \|D_2\|_1 - (\rho/2)\|0 + (\Lambda/\rho)\|^2_F,
\]

which can be simplified to

\[
\lambda_D \|tD_1 + (1-t)D_2\|_1 - t\lambda_D \|D_1\|_1 - (1-t)\lambda_D \|D_2\|_1.
\]

A necessary condition for strong convexity is strict convexity, which means that Jensen’s inequality holds strictly, meaning

\[
\lambda_D \|tD_1 + (1-t)D_2\|_1 - t\lambda_D \|D_1\|_1 - (1-t)\lambda_D \|D_2\|_1 < 0.
\]
It is straightforward to see that the inequality does not hold in a strict sense, as strictly positive elements in \(D_1, D_2\), for example, would yield equality since \(|tx + (1-t)y| = tx + (1-t)y = t|x| + (1-t)|y|\) if \(x, y \geq 0, t \in (0,1)\). An identical argument can be made for the case of considering \(E\).

To introduce strong convexity with respect to \(D\) and \(E\), respectively, in the augmented Lagrangian, one can add proximal terms \((\alpha/2)\|D - D^{(k)}\|^2_F\) and \((\beta/2)\|E - E^{(k)}\|^2_F\) to the objective function, where \(\alpha, \beta > 0\). As every term in [5.2] which depends on \(D\) or \(E\) has been shown to be convex in exactly one of the respective variables, the addition of strongly convex proximal terms yields strong convexity in each of these variables. To see this, one only needs to consider a definition of strong convexity, which states that a function \(f\) is strongly convex, with convex constant \(\mu > 0\), if the function \(g(x) = f(x) - (\mu/2)\|x\|^2\) is convex for every \(x\). The result trivially follows, as the proximal terms added to the augmented Lagrangian correspond to the term \(\mu/2\|x\|^2\), which after subtraction leaves us with the original augmented Lagrangian which was shown to be convex in \(D\) and \(E\), respectively. The corresponding constants associated with the strong convexity in \(D\) and \(E\) are \(\mu_D \geq \alpha\) and \(\mu_E \geq \beta\), respectively. The bounds come from the fact that \(L_\rho - (\mu_D/2)\|D\|^2_F\) is convex in \(D\), for \(\mu_D = \alpha\). However, depending on the specifics of \(D, E\), there might exist \(\mu_D > \alpha\), such that the property of convexity is not violated. Therefore, we may bound \(\mu_D \geq \alpha\), and similarly for the case of \(\mu_E \geq \beta\).

The modified augmented Lagrangian instead becomes

\[
C_{\rho, \alpha, \beta}(D, E, A, D^{(k)}, E^{(k)}) = \lambda_D \|D\|_1 + \lambda_E \|E\|_1 + (\rho/2)\|\mathbf{F} - DE + A/\rho\|^2_F - \|A\|^2_F/(2\rho) + (\alpha/2)\|D - D^{(k)}\|^2_F + (\beta/2)\|E - E^{(k)}\|^2_F.
\]

The ADMM iterates with respect to the modified Lagrangian become

\[
\begin{align*}
D^{(k+1)} &= \arg\min_D \lambda_D \|D\|_1 + (\rho/2)\|\mathbf{F} - DE^{(k)} + A^{(k)}/\rho\|^2_F + (\alpha/2)\|D^{(k)} - D\|^2_F, \\
E^{(k+1)} &= \arg\min_E \lambda_E \|E\|_1 + (\rho/2)\|\mathbf{F} - D^{(k+1)}E + A^{(k)}/\rho\|^2_F + (\beta/2)\|E^{(k)} - E\|^2_F, \\
A^{(k+1)} &= A^{(k)} + \rho(\mathbf{F} - D^{(k+1)}E^{(k+1)}),
\end{align*}
\] (5.10)  (5.11)  (5.12)

An important note is that the addition of the proximal terms does not change anything in regard to the analysis in the previous section. The subproblems are still convex with respect to each primal variable, which was the underlying assumption. In particular, the result obtained from Proposition [5.1.1] still holds.

Another note is that the modified augmented Lagrangian \(C_{\rho, \alpha, \beta}\) is not the corresponding augmented Lagrangian for solving (5.1). It is, however, possible to relate the ADMM iterates on \(C_{\rho, \alpha, \beta}\) to \(L_\rho\). In particular, the following Lemma allows us to upper bound the successive differences of the augmented Lagrangian \(L_\rho\) given that \((\mathbf{D}^{(k+1)}, \mathbf{E}^{(k+1)}, A^{(k+1)})\) are generated from \((\mathbf{D}^{(k)}, \mathbf{E}^{(k)}, A^{(k)})\) are generated from \([5.10] [5.11] [5.12]

**Lemma 5.1.1.** If \((\mathbf{D}^{(k+1)}, \mathbf{E}^{(k+1)}, A^{(k+1)})\) are generated from \([5.10] [5.11] [5.12]\), then the following lower bound holds for the successive differences of the augmented Lagrangian associated to Problem [5.1].

\[
L_\rho(D^{(k)}, E^{(k)}, A^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, A^{(k+1)}) \\
\geq (\alpha/2)\|D^{(k)} - D^{(k+1)}\|^2_F + (\beta/2)\|E^{(k)} - E^{(k+1)}\|^2_F - (1/\rho)\|A^{(k)} - A^{(k+1)}\|^2_F.
\]
Proof. We begin by stating a property of strongly convex functions, namely that if $f$ is strongly convex with convex constant $\mu$, then

$$f(x) \geq f(y) + \nabla f(y)^{\top} (x - y) + (\mu/2)\|x - y\|^2,$$

(5.13)

for all pairs $(x, y)$. In our case, the modified augmented Lagrangian $C_{\rho,\alpha,\beta}(D, E, A, D^{(k)}, E^{(k)})$ is not differentiable with respect to $D, E$, but a result in [22] states that the inequality (5.13) holds for $\nabla f(x)$ replaced with any subgradient $g \in \partial f(x)$, which we shall use here. Recall that the modified augmented Lagrangian is not strongly convex in the primal pair $(D, E)$, but we have strong convexity in $D$ and in $E$. Consequently, we may write

$$C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k)}, A^{(k)}) \geq C_{\rho,\alpha,\beta}(D^{(k+1)}, E^{(k)}, A^{(k)}) + (\partial C_{\rho,\alpha,\beta}(D^{(k+1)}, E^{(k)}, A^{(k)}))^{\top} (D^{(k)} - D^{(k+1)}) + (\mu_D/2)\|D^{(k)} - D^{(k+1)}\|_F^2,$$

for any pair $(D^{(k+1)}, D^{(k)})$. Similarly, for $E$, we may write

$$C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k)}, A^{(k)}) \geq C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k+1)}, A^{(k)}) + (\partial C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k+1)}, A^{(k)}))^{\top} (E^{(k)} - E^{(k+1)}) + (\mu_E/2)\|E^{(k)} - E^{(k+1)}\|_F^2,$$

for any pair $(E^{(k+1)}, E^{(k)})$.

Note that we have suppressed the last two arguments of $C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k)}, A^{(k)}, D^{(k)}, E^{(k)})$ as $C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k)}, A^{(k)})$. Since any time strong convexity is invoked with respect to some argument, we must assume all other arguments are constant. Therefore, the last two arguments are implicitly set by the ones written out.

As previously shown, the constants $\mu_D, \mu_E$, are lower bounded by $\alpha$ and $\beta$, respectively. Therefore, we may use $\alpha, \beta$ in place of $\mu_D, \mu_E$ henceforth.

Another observation is that $D^{(k+1)}$ is the minimizer of $C_{\rho,\alpha,\beta}(D, E^{(k)}, A^{(k)})$ with respect to $D$, and similarly $E^{(k+1)}$ minimizes $C_{\rho,\alpha,\beta}(D^{(k)}, E, A^{(k)})$ with respect to $E$. This observation follows directly from (5.10) and (5.11). Consequently, for any subgradients $g_D \in \partial C_{\rho,\alpha,\beta}(D^{(k+1)}, E^{(k)}, A^{(k)})$ and $g_E \in \partial C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k+1)}, A^{(k)})$, we have

$$g_D^{\top}(D^{(k)} - D^{(k+1)}) \geq 0$$

and

$$g_E^{\top}(E^{(k)} - E^{(k+1)}) \geq 0.$$

This is not a trivial result, and a derivation of it can be found in [2].

Using these results, we obtain the bounds

$$C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k)}, A^{(k)}) \geq C_{\rho,\alpha,\beta}(D^{(k+1)}, E^{(k)}, A^{(k)}) + (\alpha/2)\|D^{(k)} - D^{(k+1)}\|_F^2,$$

and

$$C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k)}, A^{(k)}) \geq C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k+1)}, A^{(k)}) + (\beta/2)\|E^{(k)} - E^{(k+1)}\|_F^2.$$

For the case of pairs $(A^{(k+1)}, A^{(k)})$, we can write $L_p(D^{(k+1)}, E^{(k+1)}, A^{(k)}) - L_p(D^{(k+1)}, E^{(k+1)}, A^{(k+1)})$ as

$$\langle A^{(k)}, F - D^{(k+1)}E^{(k+1)} \rangle_F - \langle A^{(k+1)}, F - D^{(k+1)}E^{(k+1)} \rangle_F.$$
which after simplification becomes

\[
\langle \Lambda^{(k)} - \Lambda^{(k+1)}, F - D^{(k+1)} E^{(k+1)} \rangle_F = \langle \Lambda^{(k)} - \Lambda^{(k+1)}, -(1/\rho)(\Lambda^{(k)} - \Lambda^{(k+1)}) \rangle_F = -(1/\rho)\|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2,
\]

where we used the unscaled augmented Lagrangian, and that \( \Lambda^{(k+1)} = \Lambda^{(k)} + \rho(F - D^{(k+1)} E^{(k+1)}) \).

We are now ready to show the Lemma. Consider the difference of successive points in all three variables, namely

\[
L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}) = L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)})
\]  

(5.14)

\[
+ L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}).
\]  

(5.15)

Now express (5.14) as

\[
L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k)})
\]

\[
= L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k)})
\]

\[
+ L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}),
\]

where

\[
L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k)})
\]

\[
= C_{\rho,\alpha,\beta}(D^{(k)}, E^{(k)}, \Lambda^{(k)}) - C_{\rho,\alpha,\beta}(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k)}) + (\alpha/2)\|D^{(k)} - D^{(k+1)}\|_F^2
\]

\[
\geq \alpha\|D^{(k)} - D^{(k+1)}\|_F^2
\]

(5.14)

\[
L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)})
\]

\[
= C_{\rho,\alpha,\beta}(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}) - C_{\rho,\alpha,\beta}(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}) + (\beta/2)\|D^{(k)} - D^{(k+1)}\|_F^2
\]

\[
\geq \beta\|E^{(k)} - E^{(k+1)}\|_F^2.
\]

We obtain the following lower bound for (5.14)

\[
L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k)})
\]

\[
\geq \alpha\|D^{(k)} - D^{(k+1)}\|_F^2 + \beta\|E^{(k)} - E^{(k+1)}\|_F^2.
\]

Finally, for (5.15) we have

\[
L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}) = -(1/\rho)\|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2.
\]

Combining (5.14) and (5.15) we get the bound

\[
L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)})
\]

\[
\geq \alpha\|D^{(k)} - D^{(k+1)}\|_F^2 + \beta\|E^{(k)} - E^{(k+1)}\|_F^2 - (1/\rho)\|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2,
\]

which is the claim. \(\square\)

24
An observation regarding Lemma 5.1.1 is that the result can be extended to the case of $\rho^{(k)}$ being variable per iteration. To see this, consider the difference

$$L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}, \rho^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}, \rho^{(k+1)})$$

$$= L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}, \rho^{(k)}) - L_\rho(D^{(k+1)}, E^{(k)}, \Lambda^{(k)}, \rho^{(k)})$$

$$+ L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k)}, \rho^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}, \rho^{(k+1)}). \tag{5.16}$$

Here (5.17) can be written as

$$(\rho^{(k)}/2)\|F - D^{(k+1)}E^{(k+1)}\|_F^2 + \langle A^{(k)}, F - D^{(k+1)}E^{(k+1)} \rangle_F - (\rho^{(k+1)}/2\|F - D^{(k+1)}E^{(k+1)}\|_F^2$$

$$- \langle A^{(k)} + \rho^{(k)}(F - D^{(k+1)}E^{(k+1)}), F - D^{(k+1)}E^{(k+1)} \rangle_F,$$

which after simplification becomes

$$- \frac{\rho^{(k)} + \rho^{(k+1)}}{2} \|F - D^{(k+1)}E^{(k+1)}\|_F^2 = - \frac{\rho^{(k)} + \rho^{(k+1)}}{2(\rho^{(k)})^2} \|A^{(k)} - A^{(k+1)}\|_F^2. \tag{5.18}$$

This is the only difference in the bound. With the Lemma shown, we are ready to present the main theorem regarding convergence for the noise-free ADMM iterations.

**Theorem 5.1.2.** For a sequence $\{D^{(k)}, E^{(k)}, \Lambda^{(k)}\}$ generated by the iterates

$$D^{(k+1)} = \arg\min_D \|D\|_1 + (\rho^{(k)}/2)\|F - DE^{(k)} + \Lambda^{(k)}\rho^{(k)}\|_F^2 + (\alpha/2)\|D^{(k)} - D\|_F^2 \tag{5.19}$$

$$E^{(k+1)} = \arg\min_E \|E\|_1 + (\rho^{(k)}/2)\|F - D^{(k+1)}E + \Lambda^{(k)}\rho^{(k)}\|_F^2 + (\beta/2)\|E^{(k)} - E\|_F^2 \tag{5.20}$$

$$\Lambda^{(k+1)} = \Lambda^{(k)} + \rho^{(k)}(F - D^{(k+1)}E^{(k+1)}), \tag{5.21}$$

with the following assumptions,

**Assumption 1.** The sequence $\{\Lambda^{(k)}\}$ is bounded

**Assumption 2.** The step size $\rho^{(k)}$ satisfies $\rho^{(k)} = \sigma^k \rho^{(0)}$ for $\rho^{(0)} > 0$ and $\sigma > 1$,

then $\{D^{(k)}, E^{(k)}, \Lambda^{(k)}\}$ approaches a KKT-point $(D^*, E^*, \Lambda^*)$ of (5.1) as $k \to \infty$.

**Proof.** By Lemma 5.1.1 and (5.18) we have the following bound on successive differences of the augmented Lagrangian,

$$L_\rho(D^{(k)}, E^{(k)}, \Lambda^{(k)}, \rho^{(k)}) - L_\rho(D^{(k+1)}, E^{(k+1)}, \Lambda^{(k+1)}, \rho^{(k+1)})$$

$$\geq \alpha\|D^{(k)} - D^{(k+1)}\|_F^2 + \beta\|E^{(k)} - E^{(k+1)}\|_F^2 - \frac{\rho^{(k)} + \rho^{(k+1)}}{2(\rho^{(k)})^2} \|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2.$$

Compactly expressing the left-hand side as $L^{(k)} - L^{(k+1)}$, and taking a summation from 0 to $K$ on both sides yields

$$\sum_{k=0}^{K} (L^{(k)} - L^{(k+1)}) \geq \sum_{k=0}^{K} \left( \alpha\|D^{(k)} - D^{(k+1)}\|_F^2 + \beta\|E^{(k)} - E^{(k+1)}\|_F^2 - \frac{\rho^{(k)} + \rho^{(k+1)}}{2(\rho^{(k)})^2} \|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2 \right).$$
Note that the left-hand side is a telescoping sum, and in particular we have
\[ \sum_{k=0}^{K} (L^{(k)} - L^{(k+1)}) = L^{(0)} - L^{(K)}. \]

Writing out the negative term yields
\[-L^{(K)} = -\lambda_D \|D^{(K)}\|_1 - \lambda_E \|E^{(K)}\|_1 - (\rho^{(K)}/2)\|F - D^{(K)}E^{(K)} + \Lambda/\rho^{(K)}\|_F^2 + \|\Lambda^{(K)}\|_F^2/(2\rho^{(K)})\]

Note that the only positive term is \(\|\Lambda^{(K)}\|_F^2/(2\rho^{(K)})\). By Assumption 1, \(\{\Lambda^{(k)}\}\) is a bounded sequence, meaning this positive term is upper bounded by some real constant \(L\) for every \(k\). As the rest of the terms in the Lagrangian are negative, this bound extends to all terms of \(-L^{(K)}\). Clearly, \(L^{(0)}\) is also bounded by some constant, let us denote it \(M\). Let \(N = \max(L, M)\), then taking the limit as \(K \to \infty\), we obtain
\[ \lim_{K \to \infty} L^{(0)} - L^{(K)} \leq N, \]

for some \(N \in \mathbb{R}\). As a result, we obtain a bound on the right-hand side of the inequality as \(K \to \infty\),
\[ \sum_{k=0}^{\infty} \left(\alpha \|D^{(k)} - D^{(k+1)}\|_F^2 + \beta \|E^{(k)} - E^{(k+1)}\|_F^2 - \rho^{(k)} + \rho^{(k+1)} - \rho^{(k+1)}/2\rho^{(k)}\|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2 \right) \leq N. \quad (5.22) \]

Consider now the series \(\sum_{k=0}^{\infty} \rho^{(k)} \|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2\). By Assumption 2, we can write each term
\[-\frac{\rho^{(k)} + \rho^{(k+1)}}{2\rho^{(k)}\rho^{(k+1)}} \|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2 = -\frac{1 + \rho}{2\rho \sigma^k} \|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2, \]

which inserted into the series gives
\[ \sum_{k=0}^{\infty} -\frac{1 + \rho}{2\rho \sigma^k} \|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2 \leq -\frac{L(1 + \sigma)}{\rho^{(0)}} \sum_{k=0}^{\infty} \frac{1}{\sigma^k} = -\frac{L(1 + \sigma)}{\rho^{(0)}} \frac{\sigma}{\sigma - 1}, \]

where we have used the fact that \(\Lambda^{(k)}\) is bounded, and that \(\sigma > 1 \implies |\frac{1}{\sigma}| < 1\).

Going back to (5.22) we have that the accumulated contribution of the term \(-\frac{\rho^{(k)} + \rho^{(k+1)}}{2\rho^{(k)}\rho^{(k+1)}} \|\Lambda^{(k)} - \Lambda^{(k+1)}\|_F^2\) is bounded, meaning we must have that
\[ \sum_{k=0}^{\infty} \left(\alpha \|D^{(k)} - D^{(k+1)}\|_F^2 + \beta \|E^{(k)} - E^{(k+1)}\|_F^2 \right) < \infty. \]

Consequently,
\[ D^{(k)} - D^{(k+1)} \to 0 \]
\[ E^{(k)} - E^{(k+1)} \to 0, \]
as the terms in the series are positive.

Next, we turn our attention to the KKT-conditions of (5.1). They are
We begin by considering
\[ \mathbf{A}^{(k+1)} \mathbf{E}^{(k)\top} = \mathbf{A}^{(k)} \mathbf{E}^{(k)\top} + \rho^{(k)} (\mathbf{F} - \mathbf{D}^{(k+1)} \mathbf{E}^{(k)}) \mathbf{E}^{(k)\top}. \] (5.23)

Note that if \( \mathbf{D}^{(k+1)} \) minimizes (5.19) then necessarily we have
\[ 0 \in \partial \lambda_D \| \mathbf{D}^{(k+1)} \|_1 + \alpha (\mathbf{D}^{(k)} - \mathbf{D}^{(k+1)}) - \rho^{(k)} (\mathbf{F} - \mathbf{D}^{(k+1)} \mathbf{E}^{(k)}) \mathbf{E}^{(k)\top}. \]

Note that we may rearrange this expression, since it simply means that
\[ 0 = \mathbf{g} + \alpha (\mathbf{D}^{(k)} - \mathbf{D}^{(k+1)}) - \rho^{(k)} (\mathbf{F} - \mathbf{D}^{(k+1)} \mathbf{E}^{(k)}) \mathbf{E}^{(k)\top}, \]
for some \( \mathbf{g} \in \partial \lambda_D \| \mathbf{D}^{(k+1)} \|_1 \).

Rearranging yields
\[ \mathbf{A}^{(k)} \mathbf{E}^{(k)\top} \in \partial \lambda_D \| \mathbf{D}^{(k+1)} \|_1 + \alpha (\mathbf{D}^{(k)} - \mathbf{D}^{(k+1)}) - \rho^{(k)} (\mathbf{F} - \mathbf{D}^{(k+1)} \mathbf{E}^{(k)}) \mathbf{E}^{(k)\top}. \] (5.24)

Insertion of (5.24) into (5.23) yields
\[ \mathbf{A}^{(k+1)} \mathbf{E}^{(k)\top} \in \partial \lambda_D \| \mathbf{D}^{(k+1)} \|_1 + \alpha (\mathbf{D}^{(k)} - \mathbf{D}^{(k+1)}) - \rho^{(k)} (\mathbf{F} - \mathbf{D}^{(k+1)} \mathbf{E}^{(k)}) \mathbf{E}^{(k)\top} \]
\[ + \rho^{(k)} (\mathbf{F} - \mathbf{D}^{(k+1)} \mathbf{E}^{(k)}) \mathbf{E}^{(k)\top}, \]
which approaches KKT-1 as \( k \to \infty \), since the limit implies \( \mathbf{E}^{(k)} - \mathbf{E}^{(k+1)} \to 0 \), \( \mathbf{D}^{(k)} - \mathbf{D}^{(k+1)} \to 0 \).

Next, we consider
\[ \mathbf{D}^{(k+1)} \mathbf{A}^{(k+1)} = \mathbf{D}^{(k+1)} \mathbf{A}^{(k)} + \rho^{(k)} \mathbf{D}^{(k+1)} \mathbf{E}^{(k+1)\top}. \] (5.25)

As previously, if \( \mathbf{E}^{(k+1)} \) minimizes (5.20) then by necessity we have
\[ 0 \in \partial \lambda_E \| \mathbf{E}^{(k+1)} \|_1 + \beta (\mathbf{E}^{(k)} - \mathbf{E}^{(k+1)}) - \rho^{(k)} \mathbf{D}^{(k+1)} \mathbf{E}^{(k+1)\top} (\mathbf{F} - \mathbf{D}^{(k+1)} \mathbf{E}^{(k+1)} + \mathbf{A}^{(k)} / \rho^{(k)}), \]
and rearranging gives
\[ \mathbf{D}^{(k+1)} \mathbf{A}^{(k)} \in \partial \lambda_E \| \mathbf{E}^{(k+1)} \|_1 + \beta (\mathbf{E}^{(k)} - \mathbf{E}^{(k+1)}) - \rho^{(k)} \mathbf{D}^{(k+1)} (\mathbf{F} - \mathbf{D}^{(k+1)} \mathbf{E}^{(k+1)}). \] (5.26)

Inserting (5.26) into (5.25) yields
\[ \mathbf{D}^{(k+1)} \mathbf{A}^{(k+1)} \in \partial \lambda_E \| \mathbf{E}^{(k+1)} \|_1 + \beta (\mathbf{E}^{(k)} - \mathbf{E}^{(k+1)}), \]
which means KKT-2 is satisfied in the limit \( k \to \infty \), since the limit implies \( \mathbf{E}^{(k)} - \mathbf{E}^{(k+1)} \to 0 \).
Lastly, for KKT-3, we have
\[
\lim_{k \to \infty} F - D^{(k+1)}E^{(k+1)} = \lim_{k \to \infty} \frac{1}{\rho(k)}(\Lambda^{(k+1)} - \Lambda^{(k)})
\]
\[
= \lim_{k \to \infty} \frac{1}{\rho(0)\sigma} (\Lambda^{(k+1)} - \Lambda^{(k)})
\]
\[
= 0,
\]
where the last equality holds due to \(\Lambda^{(k)}\) being bounded and \(\sigma > 1\), meaning KKT-3 is satisfied as \(k \to \infty\). This concludes the proof. ■

5.1.3 Implementation of ADMM for Noise-Free Matrix Factorization

The implementation consists of three parts, where we have an outer straightforward algorithm as summarized in Algorithm 1. We introduce auxiliary variables \(P_D, P_E\) for the inner problems,

**Algorithm 1** ADMM for the Noise-Free Matrix Factorization Problem

```
Input \(\rho(0), \lambda_E, \lambda_D, \sigma, D^{(0)}, E^{(0)}, \Lambda^{(0)}\)

for \(k = 1\) to max_iter do

\(D^{(k+1)} \leftarrow \text{argmin}_{D} \lambda_D \|D\|_1 + (\rho(k)/2)\|F - DE^{(k)} + \Lambda^{(k)}/\rho\|_F^2\)

\(E^{(k+1)} \leftarrow \text{argmin}_{E} \lambda_E \|E\|_1 + (\rho(k)/2)\|F - D^{(k+1)}E + \Lambda^{(k)}/\rho\|_F^2\)

\(\Lambda^{(k+1)} \leftarrow \Lambda^{(k)} + \rho(k)\left(F - D^{(k+1)}E^{(k+1)}\right)\)

\(\rho^{(k+1)} \leftarrow \sigma \rho^{(k)}\)

end for
```

(5.27)

\[
D^{(k+1)} = \text{argmin}_{D, P_D} \lambda_D \|P_D\|_1 + (\rho/2)\|F - DE^{(k)} + \Lambda^{(k)}/\rho\|_F^2
\]

subject to \(D = P_D\),

and

(5.28)

\[
E^{(k+1)} = \text{argmin}_{E, P_E} \lambda_E \|P_E\|_1 + (\rho/2)\|F - D^{(k+1)}E + \Lambda^{(k)}/\rho\|_F^2
\]

subject to \(E = P_E\).

The problems are then solved using ADMM. Derivations of the closed form ADMM-updates for (5.27) (5.28) are found in Appendix A. The algorithms are
Algorithm 2 ADMM for the D-subproblem

```
Input \( \rho^{(k)}, \lambda_D, \alpha, D^{(k)}, E^{(k)}, A^{(k)} \)
\[ \theta^{(k)} \leftarrow \|E^{(k)}\|_F^2 \]
\[ A \leftarrow (\rho E^{(k)} + (\alpha + \theta^{(k)})I)^{-1} \]
for \( i = 1 \) to max_iter do
\[ D^{(i+1)} \leftarrow (\rho^{(k)}(F + A^{(k)})/\rho^{(k)})E^{(k)} + \alpha D^{(i)} - \theta^{(k)}(A_D^{(i)}/\theta^{(k)} - P_D^{(i)}))A \]
\[ P_D^{(i+1)} \leftarrow S_{\lambda_D/\theta^{(k)}}(A_D^{(i)}/\theta^{(k)} + D^{(i)}) \]
\[ A_D^{(i+1)} \leftarrow A_D^{(i)} + \theta^{(k)}(D^{(i+1)} - P_D^{(i+1)}) \]
end for

Algorithm 3 ADMM for the E-subproblem

```
Input \( \rho^{(k)}, \lambda_E, \beta, D^{(k)}, A^{(k)} \)
\[ \phi^{(k)} \leftarrow \|D^{(k)}\|_F^2 \]
\[ B \leftarrow (\rho^{(k)}D^{(k)\top}D^{(k)} + (\beta + \phi^{(k)})I)^{-1} \]
for \( j = 1 \) to max_iter do
\[ E^{(j+1)} \leftarrow B(\rho^{(k)}D^{(k)\top}(F + A^{(k)})/\rho^{(k)}) + \beta E^{(j)} - \phi^{(k)}(A_E^{(j)}/\phi^{(k)} - P_E^{(j)})) \]
\[ P_E^{(j+1)} \leftarrow S_{\lambda_E/\phi^{(k)}}(A_E^{(j)}/\phi^{(k)} + E^{(j)}) \]
\[ A_E^{(j+1)} \leftarrow A_E^{(j)} + \phi^{(k)}(E^{(j+1)} - P_E^{(j+1)}) \]
end for
```

The choice of step size parameters \( \theta^{(k)}, \phi^{(k)} \) for the subproblems is empirically motivated.

This concludes the ADMM formulation, analysis, and implementation of the noise-free matrix factorization problem.

### 5.2 ADMM for the Noisy Case

For the case of having noise present, we consider the unconstrained problem

\[
\min_{D, E} (1/2)\|F - DE\|_F^2 + \lambda_D\|D\|_1 + \lambda_E\|E\|_1.
\]

In order for (5.29) to fit the ADMM-framework, we introduce an auxiliary variable \( Z = DE \), which gives a constrained problem that is separable in \( Z \) and \( (D, E) \). The problem becomes

\[
\min_{D, E, Z} (1/2)\|F - Z\|_F^2 + \lambda_D\|D\|_1 + \lambda_E\|E\|_1 \tag{5.30}
\]

subject to \( Z - DE = 0 \).

Additionally, for the ADMM-iterates, proximal terms \((\alpha/2)\|D^{(k)} - D\|_F^2, (\beta/2)\|E^{(k)} - E\|_F^2\) with \( \alpha, \beta > 0 \), are introduced into the objective function in order to leverage strong convexity in \( D \) and in \( E \), as in the previous section. We consider the splitting \( E \) and \( (D, Z) \) for the iterations, yielding
the updates
\[ E^{(k+1)} = \arg\min_E \lambda E\|E\|_1 + (\rho/2)\|Z^{(k)} - D^{(k)}E + \Lambda^{(k)}/\rho\|_F^2 \]  
(5.31)

\[ (D^{(k+1)}, Z^{(k+1)}) = \arg\min_{D, Z} \|F - Z\|_F^2 + \lambda_D\|D\|_1 + (\rho/2)\|Z - DE^{(k)} + \Lambda^{(k)}/\rho\|_F^2 \]  
(5.32)

\[ \Lambda^{(k+1)} = \Lambda^{(k)} + \rho(Z^{(k+1)} - D^{(k+1)}E^{(k+1)}). \]  
(5.33)

The convergence analysis of subproblems (5.31) and (5.32) is identical to the subproblems in the previous section, and is therefore omitted. Specifically, the convergence assumptions for ADMM hold with respect to these subproblems, as the main difference is the presence of an additional quadratic term which is positively weighted. This does not affect convexity or coercivity.

5.2.1 Convergence of Main Problem

For the convergence of iterates (5.31), (5.32), (5.33), a proposition is derived from a result in [23]. The proposition states

**Proposition 5.2.1.** For a sequence \(\{D^{(k)}, E^{(k)}, Z^{(k)}, \Lambda^{(k)}\}\) generated by the iterates

\[ E^{(k+1)} = \arg\min_E \lambda E\|E\|_1 + (\rho/2)\|Z^{(k)} - D^{(k)}E + \Lambda^{(k)}/\rho\|_F^2 + (\beta/2)\|E^{(k)} - E\|_F^2 \]  
(5.31)

\[ (D^{(k+1)}, Z^{(k+1)}) = \arg\min_{D, Z} \|F - Z\|_F^2 + \lambda_D\|D\|_1 + (\rho/2)\|Z - DE^{(k)} + \Lambda^{(k)}/\rho\|_F^2 + (\alpha/2)\|D^{(k)} - D\|_F^2 \]  
(5.32)

\[ \Lambda^{(k+1)} = \Lambda^{(k)} + \rho(Z^{(k+1)} - D^{(k+1)}E^{(k+1)}). \]  
(5.33)

where \(\rho > 1\), the following holds

- The duality gap goes to zero, meaning
  \[ \lim_{k \to \infty} \|Z^{(k)} - D^{(k+1)}E^{(k+1)}\|_F \to 0. \]

- \((\Lambda^*, D^*, E^*, Z^*)\) are stationary points for Problem (5.30)

The proof of this proposition follows directly from the proof of Theorem 1 in [23].
5.2.2 Implementation of ADMM for Noisy Matrix Factorization

As previously, the implementation consists of an outer algorithm,

**Algorithm 4** ADMM for the Noisy Matrix Factorization Problem

<table>
<thead>
<tr>
<th>Input</th>
<th>( \rho, \lambda_E, \lambda_D, \alpha, \beta, D^{(0)}, E^{(0)}, Z^{(0)}, \Lambda^{(0)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>for ( k = 1 ) to max_iter do</td>
<td></td>
</tr>
<tr>
<td>( E^{(k+1)} \leftarrow \arg\min_{E} \lambda_E |E|_1 + (\rho/2)|Z^{(k)} - D^{(k)}E + \Lambda^{(k)}/\rho|_F^2 + (\beta/2)|E^{(k)} - E|_F^2 )</td>
<td></td>
</tr>
<tr>
<td>( (D^{(k+1)}, Z^{(k+1)}) \leftarrow \arg\min_{D, Z} (1/2)|F - Z^2_F + \lambda_D |D|_1 + (\rho/2)|Z - DE^{(k+1)} + \Lambda^{(k)}/\rho|_F^2 + (\alpha/2)|D^{(k)} - D|_F^2 )</td>
<td></td>
</tr>
<tr>
<td>( \Lambda^{(k+1)} \leftarrow \Lambda^{(k)} + \rho(Z^{(k+1)} - D^{(k+1)}E^{(k+1)}) )</td>
<td></td>
</tr>
<tr>
<td>end for</td>
<td></td>
</tr>
</tbody>
</table>

and two inner algorithms. For the inner algorithms, we again introduce auxiliary variables \( P_D, Z_E \), yielding problems of the form

\[
E^{(k+1)} = \arg\min_{E \in P} \lambda_E \|E\|_1 + (\rho/2)\|Z^{(k)} - D^{(k+1)}E + \Lambda^{(k)}/\rho\|_F^2 \tag{5.34}
\]

subject to \( E = P_E \),

and

\[
(D^{(k+1)}, Z^{(k+1)}) = \arg\min_{D, Z, P_D} \lambda_D \|D\|_1 + (\rho/2)\|Z - DE^{(k)} + \Lambda^{(k)}/\rho\|_F^2 \tag{5.35}
\]

subject to \( D = P_D \).

The inner algorithms are then

**Algorithm 5** ADMM for the E-subproblem

<table>
<thead>
<tr>
<th>Input</th>
<th>( \rho, \lambda_E, \beta, D^{(k)}, Z^{(k)}, \Lambda^{(k)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varphi^{(k)} \leftarrow |D^{(k)}|_F^2 )</td>
<td></td>
</tr>
<tr>
<td>( B \leftarrow (\rho D^{(k)\top}D^{(k)} + (\beta + \varphi^{(k)})I)^{-1} )</td>
<td></td>
</tr>
<tr>
<td>for ( j = 1 ) to max_iter do</td>
<td></td>
</tr>
<tr>
<td>( E^{(j+1)} \leftarrow B(\rho D^{(k)\top}(Z^{(k)} + \Lambda^{(k)}/\rho) + \beta E^{(j)} - \varphi^{(k)}(\Lambda^{(j)}E^{(j)}/\varphi^{(k)} - P_E^{(j)})) )</td>
<td></td>
</tr>
<tr>
<td>( P_E^{(j+1)} \leftarrow S_{\lambda_E/\varphi^{(k)}}(\Lambda_E^{(j)}/\varphi^{(k)} + E^{(j)}) )</td>
<td></td>
</tr>
<tr>
<td>( \Lambda_E^{(j+1)} \leftarrow \Lambda_E^{(j)} + \varphi^{(k)}(E^{(j+1)} - P_E^{(j+1)}) )</td>
<td></td>
</tr>
<tr>
<td>end for</td>
<td></td>
</tr>
</tbody>
</table>
Algorithm 6 ADMM for the \((D, Z)\)-subproblem

Input \(\rho, \lambda_D, \alpha, D^{(k)}, E^{(k)}, A^{(k)}\)

\[\theta^{(k)} \leftarrow \frac{\|E^{(k)}\|^2_F}{2}\]

\[A \leftarrow (\rho E^{(k)})^T E^{(k+1)} + (\alpha + \theta^{(k)})I^{-1}\]

for \(i = 1\) to \(\text{max.iter}\) do

\[D^{(i+1)} \leftarrow (\rho^{(k)}(Z^{(i)} + A^{(k)}/\rho^{(k)})E^{(k+1)^T} + \alpha D^{(i)} - \theta^{(k)}(A_D^{(i)}/\theta^{(k)} - P^{(i)D}))A\]

\[Z^{(i+1)} \leftarrow \frac{1}{\rho + 1}(F + \rho(D^{(i)E^{(k+1)}} + A^{(k)}/\rho))\]

\[P^{(i+1)D} \leftarrow S_{\lambda_D/\theta^{(k)}}(A_D^{(i)}/\theta^{(k)} + D^{(i)})\]

\[A_D^{(i+1)} \leftarrow A_D^{(i)} + \theta^{(k)}(D^{(i+1)} - P^{(i+1)D})\]

end for

The choice of step size parameters \(\theta^{(k)}, \varphi^{(k)}\) for the subproblems is, again, empirically motivated.

This concludes the ADMM formulation, analysis, and implementation of the noisy matrix factorization problem.
Chapter 6

Results and Simulations

In the following simulations, Algorithm 1 is used for the noiseless case and Algorithm 4 is used for the noisy case. The matrix $F \in \mathbb{R}^{200 \times 200}$ is generated by multiplying $D \in \mathbb{R}^{200 \times 10}$ and $E \in \mathbb{R}^{10 \times 200}$, with both matrices having elements sampled from $\mathcal{N}(0, 1)$. The supports of the matrices $D$ and $E$ were randomly chosen. The algorithms were initiated with $D(0)$ and $E(0)$ generated randomly. All corresponding Lagrange multipliers were initiated as $\Lambda(0) = 0$, and the auxiliary variable $Z$ was initiated as $Z(0) = F$.

Relating to the setting of distributed computing, the dimensions of the matrices here indicate a setup where the number of users and datasets ($L = K = 200$) are significantly larger than the number of servers, or computing nodes ($N = 10$).

6.1 Phase-Transition and Convergence Plots

For the following simulations, the matrix $F$ is generated from the factors $D$ and $E$ having all combinations of density of nonzero elements $(S_D, S_E) \in \{0.1, 0.2, \ldots, 0.9\} \times \{0.1, 0.2, \ldots, 0.9\}$. For a given pair $(S_D, S_E)$, we take the average of 25 simulations with respect to two different metrics, where, $D, E$ and their support are randomly generated in every simulation. The first metric is the relative error in reconstructing $F$ from $D^{(k)}, E^{(k)}$, which is here defined as $\frac{\|F - D^{(k)}E^{(k)}\|_F}{\|F\|_F}$. The second metric consists of the associated costs $\delta$ and $\gamma$, which is measured as the density of nonzero elements in the following manner

$$
\delta = \frac{\|D\|_0}{KN},
$$

$$
\gamma = \frac{\|D\|_0}{NL}.
$$

The idea of the simulation setup is to identify how well the algorithm can reconstruct both the density of nonzero elements $(S_D, S_E)$ of $D, E$ and how well $F$ is reconstructed. It is not interesting whether $D^{(k)} \rightarrow D, E^{(k)} \rightarrow E$ in the limit $k \rightarrow \infty$, since there is ambiguity in both scaling and support, for the matrix factorization problem. What is interesting here is the number of nonzero elements of $D^{(k)}, E^{(k)}$. The term phase-transition means that the goal is to differentiate regions of good and bad performance, depending on the quality of the recovered factors.
6.1.1 Noise-Free Environment

For the noise-free case, the simulations are done with parameters $\rho^{(0)} = 1.1, \sigma = 1.0001, \alpha = \beta = 0.1, \lambda_D = \lambda_E = 0.5$, for all pairs $(S_D, S_E)$.

Figure 6.1: The relative error $\frac{\|F - D^{(k)} E^{(k)}\|_F}{\|F\|_F}$ at the final iteration for each pair $(S_D, S_E)$. The value in each section is the result of 25 Monte Carlo simulations. All simulations ran for 5000 iterations in the outer ADMM loop.
Figure 6.2: The density of nonzero elements in $D^{(k)}$ at the final iteration. The value in each section is the result of 25 Monte Carlo simulations. All simulations ran for 5000 iterations in the outer ADMM loop.
The results indicate that the algorithm indeed converges to a solution for the pairs \((S_D, S_E)\) considered. There is a clear region \(S_D, S_E \geq 0.7\) where recovery of \(D, E\) fails in terms of sparsity, as seen in Figures 6.2, 6.3. Note, however, that it is only the sparsity that is unsuccessfully recovered in these regions, as Figure 6.1 shows that the matrix \(F\) is still recovered to a precision within \(10^{-6}\) for \(S_D, S_E \geq 0.7\). With increasing \((S_D, S_E)\), the generated factors \(D, E\) become less sparse, meaning the resulting \(F\) is eventually a full matrix, which seems to impact sparse factor recovery.

The following plots show how the algorithm converges for a particular Monte Carlo simulation,
Figure 6.4: Convergence of $\|D^{(k)}\|_0$ and $\|E^{(k)}\|_0$ for one of the Monte Carlo simulations with Algorithm 1 and $(S_D, S_E) = (0.5, 0.5)$. 

Density of nonzero elements in $D$ and $E$
Figure 6.5: Convergence of the objective function for one of the Monte Carlo simulations with Algorithm 1 and $(S_D, S_E) = (0.5, 0.5)$.

Indeed, the algorithm converges to a solution after around 1000 iterations in this case.
6.2 Noisy Environment

For the noisy case, the simulations are done with parameters $\rho = 1.1$, $\alpha = \beta = 0.1$, $\lambda_D = \lambda_E = 1$, for all pairs $(S_D, S_E)$. The noise is introduced additively as $F = D\mathbf{E} + N$, where $N \sim \mathcal{N}(0, 0.1^2)$.

Figure 6.6: Noisy case. The relative error $\frac{\|D\mathbf{E} - D^{(k)}\mathbf{E}^{(k)}\|_F}{\|D\mathbf{E}\|_F}$ at the final iteration for each pair $(S_D, S_E)$. The value in each section is the result of 25 Monte Carlo simulations. All simulations ran for 5000 iterations in the outer ADMM loop.
Figure 6.7: Noisy case. The density of nonzero elements in $D^{(k)}$ at the final iteration. The value in each section is the result of 25 Monte Carlo simulations. All simulations ran for 5000 iterations in the outer ADMM loop.
Figure 6.8: Noisy case. The density of nonzero elements in $E^{(k)}$ at the final iteration. The value in each section is the result of 25 Monte Carlo simulations. All simulations ran for 5000 iterations in the outer ADMM loop.

Again, there is convergence for the case of Algorithm 4. It can be noted that the relative errors in Figure 6.6 are significantly larger than in Figure 6.1, which is due to the introduced noise $\mathbf{N} \sim \mathcal{N}(0, 0.1^2)$. The recovery of sparse factors in Figures 6.7, 6.7 looks to be successful in regions where $S_D, S_E \leq 0.7$. Note, however, that the relative error is significantly smaller in the regions $S_D, S_E \geq 0.7$, where it is in the order of the variance of the noise. The larger values of relative errors are likely due to two main reasons. The first one is that the parameters were not carefully tuned for every pair $(S_D, S_E)$ and therefore the high values of $\lambda_D, \lambda_E$ penalized deviations from sparsity too harshly for the lower values of $(S_D, S_E)$ when noise was introduced. The second reason is that the noise $\mathbf{N}$ has a greater relative impact on $\mathbf{F}$ generated from sparser factors $\mathbf{D}, \mathbf{E}$, which could affect recovery.
Figure 6.9: Noisy case. Convergence of $\|D^{(k)}\|_0$ and $\|E^{(k)}\|_0$ for Algorithm 4 with $(S_D, S_E) = (0.5, 0.5)$. 
Figure 6.10: Noisy case. Convergence of the objective function for Algorithm 4 with $(S_D, S_E) = (0.5, 0.5)$.

The convergence plots are similar to the noise-free case, the main difference seems to be that Algorithm 4 converges slightly faster than Algorithm 3.
Chapter 7
Conclusions and Future Work

In this thesis, we have considered the setting of linearly separable distributed computing with \( K \) users, \( N \) servers and \( L \) datasets. Using an established connection between the condition of feasible computation and matrix factorization, where the costs associated with the computation and communication could be described by the sparsity of the factors, two corresponding sparse matrix factorization problems were presented as an approach to minimize costs associated with the feasibility condition. Furthermore, convex analysis was conducted on the suggested sparse matrix factorization problems, resulting in alternative formulations for which an alternating method such as ADMM would yield convex subproblems. Two algorithms for sparse matrix factorization were developed using ADMM, for the cases of considering a noise-free and noisy environment, respectively. Theoretical results for the convergence of the noise-free environment were derived, whereas for the noisy environment, convergence results were presented as a consequence of existing results. Finally, simulations using numerical implementations of the proposed algorithms were conducted, and the results of these simulations were presented. The conclusion from the simulations was that the algorithms generally recover a pair of sparse factors and recover the matrix accurately, for matrices generated with sufficiently sparse factors.

A natural continuation of this work is to consider alternative methods to ADMM for handling the sparse matrix factorization problems. For example, Majorization-Minimization (MM) is a common approach used in compressed sensing to recover sparse signals \( x \) such that \( Ax = y \) [24]. With the MM approach, one seeks to minimize an objective function \( f(x) \) by using a so-called surrogate function \( u(x, y) \), which always lies above or on the function of interest and is significantly easier to minimize. One way to apply MM to the sparse matrix factorization problem, inspired by the work in [25], is to consider alternative measures of sparsity as opposed to the \( \ell_0 \)-pseudonorm. See Figure 7.1 for an illustration of these sparsity measures. As shown in [25], it may be possible to derive quadratic majorizers that approximate these functions, which would lead to tractable optimization problems. There has been recent work on the application of MM to the sparse matrix factorization problem [19, 26], so it does indeed seem like a promising alternative to consider.
Figure 7.1: Different approximations of $\ell_0$ on the interval $[-1,1]$. As $\sigma \to 0$, the approximations gets better.
Bibliography


Appendix A

Derivation of Closed-Form Solutions to ADMM-Subproblems

We begin by defining the auxiliary variable $Z = DE$, turning the problem into

$$\begin{align*}
\min_{D, E, Z} & \|F - Z\|_F^2 + \lambda_D \|D\|_1 + \lambda_E \|E\|_1 \\
\text{s.t.} & \quad Z = DE.
\end{align*} \tag{A.1}$$

The augmented Lagrangian becomes

$$L_\rho = \|F - Z\|_F^2 + \lambda_D \|D\|_1 + \lambda_E \|E\|_1 + (\rho/2)\|Z - DE + U\|_F^2 - (\rho/2)\|U\|_F^2. \tag{A.2}$$

We will consider the splitting $E$ and $(D, Z)$ for the iterations. The ADMM updates become

$$E^{k+1} = \arg\min_E \lambda_E \|P_E\|_1 + (\rho/2)\|Z^k - D^kE + U^k\|_F^2 \tag{A.3}$$

$$(D^{k+1}, Z^{k+1}) = \arg\min_{D, Z} \|F - Z\|_F^2 + \lambda_D \|D\|_1 + (\rho/2)\|Z - DE^{k+1} + U^k\|_F^2 \tag{A.4}$$

$$U^{k+1} = U^k + Z^{k+1} - D^{k+1}E^{k+1}$$

Note that subproblems (A.3) and (A.4) are unconstrained convex optimization problems. Moreover, we can apply ADMM on these subproblems by introducing an appropriate auxiliary variable that gives us separability in the objective function.

For Problem (A.3), we introduce the auxiliary variable $P_E = E$, and formulate the optimization problem on ADMM form as

$$\begin{align*}
\min_{E, P_E} & \lambda_E \|P_E\|_1 + (\rho/2)\|Z^k - D^kE + U^k\|_F^2 \\
\text{s.t.} & \quad E = P_E = 0.
\end{align*} \tag{A.5}$$

The augmented Lagrangian becomes

$$L_\varphi = \lambda_E \|P_E\|_1 + (\rho/2)\|Z^k - D^kE + U^k\|_F^2$$

$$+ (\varphi/2)\|E - P_E + U_E\|_F^2,$$

48
and we get the ADMM iterates as

\[ \mathbf{E}^{j+1} = \arg\min_{\mathbf{E}} (\rho/2) \| \mathbf{Z}^k - \mathbf{D}^k \mathbf{E} + \mathbf{U}^k \|_F^2 + (\varphi/2) \| \mathbf{E} - \mathbf{P}_{\mathbf{E}^j} + \mathbf{U}_{\mathbf{E}^j} \|_F^2 \]  \hspace{1cm} (A.6) \\
\[ \mathbf{P}_{\mathbf{E}}^{j+1} = \arg\min_{\mathbf{P}_{\mathbf{E}}} \lambda_{\mathbf{E}} \| \mathbf{P}_{\mathbf{E}} \|_1 + (\varphi/2) \| \mathbf{E}^{j+1} - \mathbf{P}_{\mathbf{E}} + \mathbf{U}_{\mathbf{E}} \|_F^2 \]  \hspace{1cm} (A.7) \\
\[ \mathbf{U}_{\mathbf{E}}^{j+1} = \mathbf{U}_{\mathbf{E}}^j + \mathbf{E}^{j+1} - \mathbf{P}_{\mathbf{E}}^{j+1} \]

We first consider Problem (A.6). Using the definition of the Frobenius norm, we have

\[ \| \mathbf{Z}^k - \mathbf{D}^k \mathbf{E} + \mathbf{U}^k \|_F^2 = \sum_{i,j} ((z^k + u^k)_{ij} - \sum_{k'} d^k_{k'i} e_{k'j})^2 \] \\
\[ \| \mathbf{E} - \mathbf{P}_{\mathbf{E}} + \mathbf{U}_{\mathbf{E}} \|_F^2 = \sum_{i,j} (- (p_{\mathbf{E}} + u_{\mathbf{E}})_{ij} + e_{ij})^2. \]

Element wise differentiation then gives

\[ \frac{\partial \| \mathbf{Z}^k - \mathbf{D}^k \mathbf{E} + \mathbf{U}^k \|_F^2}{\partial e_{ij}} = -2 \sum_{k'} d^k_{k'i} ((z^k + u^k)_{k'j} - \sum_i d^k_{k'i} e_{ij}) = -2 \sum_{k'} d^k_{k'i} (\mathbf{Z}^k - \mathbf{D}^k \mathbf{E} + \mathbf{U}^k)_{k'j} = [-2 \mathbf{D}^k \mathbf{T} (\mathbf{Z}^k - \mathbf{D}^k \mathbf{E} + \mathbf{U}^k)]_{i,j}. \]

\[ \frac{\partial \| \mathbf{E} - \mathbf{P}_{\mathbf{E}} + \mathbf{U}_{\mathbf{E}} \|_F^2}{\partial e_{ij}} = 2(- (p_{\mathbf{E}} + u_{\mathbf{E}})_{ij} + e_{ij}) = [2(\mathbf{E} - \mathbf{P}_{\mathbf{E}} + \mathbf{U}_{\mathbf{E}})]_{i,j}. \]

Considering all elements simultaneously means that we can express the matrix derivative as

\[ \frac{\partial \| \mathbf{Z}^k - \mathbf{D}^k \mathbf{E} + \mathbf{U}^k \|_F^2}{\partial \mathbf{E}} = -2 \mathbf{D}^k \mathbf{T} (\mathbf{Z}^k - \mathbf{D}^k \mathbf{E} + \mathbf{U}^k), \]
\[ \frac{\partial \| \mathbf{E} - \mathbf{P}_{\mathbf{E}} + \mathbf{U}_{\mathbf{E}} \|_F^2}{\partial \mathbf{E}} = 2(\mathbf{E} - \mathbf{P}_{\mathbf{E}} + \mathbf{U}_{\mathbf{E}}). \]

Since the problem is convex and unconstrained, a necessary and sufficient condition for a solution is

\[ -2(\rho/2) \mathbf{D}^k \mathbf{T} (\mathbf{Z}^k - \mathbf{D}^k \mathbf{E} + \mathbf{U}^k) + 2(\varphi/2)(\mathbf{E} - \mathbf{P}_{\mathbf{E}} + \mathbf{U}_{\mathbf{E}}) = 0, \]

which after rearranging gives

\[ (\rho \mathbf{D}^k \mathbf{T} + \varphi \mathbf{I}) \mathbf{E} - \rho \mathbf{D}^k \mathbf{T} (\mathbf{Z}^k + \mathbf{U}^k) + \varphi (\mathbf{U}_{\mathbf{E}} - \mathbf{P}_{\mathbf{E}}) = 0. \]

Inversion now gives us the optimal \( \mathbf{E} \) as,

\[ \mathbf{E} = (\rho \mathbf{D}^k \mathbf{T} + \varphi \mathbf{I})^{-1} (\rho \mathbf{D}^k \mathbf{T} (\mathbf{Z}^k + \mathbf{U}^k) - \varphi (\mathbf{U}_{\mathbf{E}} - \mathbf{P}_{\mathbf{E}})). \]  \hspace{1cm} (A.8)
Note that \((ρD_k^TD_k + φI)\) is always invertible, since \(ρ, φ > 0\), \(D_k^TD_k \succeq 0\) and the identity matrix is positive definite.

We now turn our attention to Problem [A.7] and consider again the term wise expansion,

\[
λ_E\|P_E\|_1 + (φ/2)\|E - P_E + U_E\|_F^2 = λ_E \sum_{i,j} |(p_e)_{ij}| + (φ/2) \sum_{i,j}(- (p_e)_{ij} + (u_e)_{ij} + e_{ij})^2
\]

\[= \sum_{i,j} (λ_E |(p_e)_{ij}| + (φ/2)(-(p_e)_{ij} + (u_e)_{ij} + e_{ij})^2)
\]

We consider three cases for \((p_e)_{ij}\) here. First, we consider \((p_e)_{ij} > 0\), which yields

\[
\frac{∂(∑_{i,j} (λ_E(p_e)_{ij} + (φ/2)(−(p_e)_{ij} + (u_e)_{ij} + e_{ij})^2))}{∂(p_e)_{ij}} = λ_E - φ(- (p_e)_{ij} + (u_e)_{ij} + e_{ij}),
\]

which is zero for \((p_e)_{ij} = - \frac{λ_E}{φ} + ((u_e)_{ij} + e_{ij})\). The condition \((p_e)_{ij} > 0\) means that \(((u_e)_{ij} + e_{ij}) > - \frac{λ_E}{φ}\).

The second case is \((p_e)_{ij} < 0\), yielding

\[
\frac{∂(∑_{i,j} (-λ_E(p_e)_{ij} + (φ/2)(−(p_e)_{ij} + (u_e)_{ij} + e_{ij})^2))}{∂(p_e)_{ij}} = -λ_E - φ(- (p_e)_{ij} + (u_e)_{ij} + e_{ij}),
\]

which is zero for \((p_e)_{ij} = \frac{λ_E}{φ} + ((u_e)_{ij} + e_{ij})\). The condition \((p_e)_{ij} < 0\) means that \(((u_e)_{ij} + e_{ij}) < - \frac{λ_E}{φ}\).

The last case is \((p_e)_{ij} = 0\). Note that the subgradient of \(|(p_e)_{ij}|\) is not uniquely defined at this point. It can be any value between \([-1, 1]\). Denote this value \(α \in [-1, 1]\), and we get that

\[
0 = αλ_E - φ(−(p_e)_{ij} + (u_e)_{ij} + e_{ij})|(p_e)_{ij}=0 ⇐⇒ αλ_E = ((u_e)_{ij} + e_{ij}),
\]

so we have \((p_e)_{ij} = 0\) for \(((u_e)_{ij} + e_{ij}) \in [- \frac{λ_E}{φ}, \frac{λ_E}{φ}]\).

We can represent each \((p_e)_{ij}\) of these outcomes jointly, as

\[
(p_e)_{ij} = \begin{cases} 
0 & \text{if } |(u_e)_{ij} + e_{ij}| ≤ \frac{λ_E}{φ} \\
- \frac{λ_E}{φ} \text{sgn}((u_e)_{ij} + e_{ij}) + ((u_e)_{ij} + e_{ij}) & \text{if } |(u_e)_{ij} + e_{ij}| > \frac{λ_E}{φ},
\end{cases}
\]

or more compactly, and commonly referred to as element wise soft thresholding,

\[
(p_e)_{ij} = \text{sgn}((u_e)_{ij} + e_{ij}) \max(|(u_e)_{ij} + e_{ij}| - \frac{λ_E}{φ}, 0).
\] \hspace{1cm} (A.9)

Define the operator \(S_λ : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}\) as the matrix representation of the element wise procedure in \([A.9]\), for a given parameter \(λ\), meaning \((S_λ(X))_{ij} = \text{sgn}(x_{ij}) \max(|x_{ij}| - λ, 0)\). Then from \([A.9]\), we get that the solution to Problem \([A.7]\) is

\[
P_E = S_{\frac{λ_E}{φ}} (U_E + E).
\] \hspace{1cm} (A.10)
We can now express the ADMM updates that correspond to Problem (A.6) and Problem (A.7) as

\[ E^{j+1} = (\rho D^{k^T}D^k + \varphi I)^{-1}(\rho D^{k^T}(Z^k + U^k) - \varphi(U_E - P_E)) \]

\[ P_E^{j+1} = S_{\lambda\varphi}(U_E + E). \]

Finally, we consider Problem (A.4). We again introduce an auxiliary variable \( P_D = D \), and formulate the problem on ADMM form as

\[
\begin{align*}
\min_{D, P_D, Z} & \quad \|F - Z\|_F^2 + \lambda_D \|P_D\|_1 + (\rho/2)\|Z - DE^{k+1} + U^k\|_F^2 \\
\text{s.t} & \quad D - P_D = 0.
\end{align*}
\]

The augmented Lagrangian becomes, for \( \theta > 0 \),

\[
L_\theta = \|F - Z\|_F^2 + \lambda_D \|P_D\|_1 + (\rho/2)\|Z - DE^{k+1} + U^k\|_F^2 + (\theta/2)\|D - P_D + U_D\|_F^2.
\]

The derivation of the closed form ADMM iterates are identical to the previously derived formulas, and are therefore omitted. We obtain

\[
\begin{align*}
D^{i+1} &= (\rho(Z^k + U^k)E^{k+1} - \theta(U_{ZD}^i - P_D^i))(\rho E^{k+1} + \varphi I)^{-1} \\
Z^{i+1} &= (\rho + 1)^{-1}I(F + \rho(D^iE^{k+1} + U^k)) \\
P_D^{i+1} &= S_{\lambda\varphi}(D^{i+1} + U_D^i) \\
U_D^{i+1} &= U_D^i + D^{i+1} - P_D^{i+1}
\end{align*}
\]