6-15-2017

Jointly Learning Features and Temporal Contingency for Prediction in Large Scale Datasets

Tingyang Xu
University of Connecticut - Storrs, xuty_007@hotmail.com

Jointly Learning Features and Temporal Contingency for Prediction in Large Scale Datasets

Tingyang Xu, Ph.D.
University of Connecticut, 2017

ABSTRACT

Temporal data such as time series data and longitudinal data are pervasive across almost all human endeavors, including medicine, finance, climate, and genetics. As such, it is hardly surprising that temporal data mining has attracted significant attention and research effort. Only very recently, feature selection has drawn adequate attention in the context of longitudinal modeling. Standard statistical techniques, such as generalized estimating equations (GEE), have been modified to identify important features by imposing sparsity-inducing regularizers. However, they do not explicitly model how a dependent variable relies on features measured at proximal time points. Recent machine learning models can select features at lagged time points but ignore the temporal correlations within an individual's repeated measurements. With advances in data acquisition technologies and availability of big data, ultra-high dimensions with complex structure are present in many subjects recorded in a continuous time period, which imposes another challenge on temporal data analysis. In order to effectively model the complex data structure, huge data size, and lagged effects along time of temporal data, we propose in this thesis study several novel machine learning methods.

First, we propose an approach called Longitudinal LASSO (i.e., Least Absolute Shrinkage and Selection Operator), to automatically and simultaneously determine both the relevant features and the time points that impact the current observation of a dependent variable. Meanwhile, the proposed approach models the fact that data are not independently and identically distributed ($i.i.d$) due to the temporal correlations within an individual. This approach decomposes model parameters into a summation of two components and imposes separate block-wise LASSO penalties on each component when building a linear model in terms of τ repeated measurements of a set of features. One component is used to select features whereas the other is used to select temporal contingent points.

Second, we extend the first method to a new tensor-based quadratic inference function, (Tensor-QIF), which aims to select structured features along each dimension of the tensor data. Assume that the data example is a k-way tensor and we build a linear model with respect to the tensor, the parameters in the model naturally form another k-way tensor. Mathematically, we decompose the k-way parameter tensor into a summation of k sparse k-way tensors. These tensors each present sparsity along

one direction of the parameter tensor. In order to correct for the non-$i.i.d$ nature of the data, we employ QIF to estimate within-individual correlations, which brings advantages over the classic GEE methods because presumed covariance structures in GEE always mis-specify complex correlation structures.

Due to the immense growth of data, it is necessary to take advantage of modern high performance computing (HPC) systems. In other words, parallelized optimization solvers are helpful to solve the above two models with the issues of huge data size and longtime recordings for large-scaled time-related datasets. Hence, third, we propose a hybrid stochastic dual coordinate ascent (hybrid-SDCA) solver for a multi-core cluster, the most common high performance computing environment that consists of multiple computing nodes with each having multiple cores and its own shared memory. We distribute data across nodes where each node solves a local problem in an asynchronous parallel fashion on its cores, and then the local updates are aggregated via an asynchronous across-node update scheme. The proposed double asynchronous method converges to a global solution for L-Lipschitz continuous loss functions, and at a linear convergence rate if a smooth convex loss function is used.

Jointly Learning Features and Temporal Contingency for Prediction in Large Scale Datasets

Tingyang Xu

Bachelor of Sciense, Shanghai Jiao Tong University, China, 2010
Master, University of Connecticut, USA, 2017

A Dissertation
Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
at the
University of Connecticut

2017

APPROVAL PAGE

Doctor of Philosophy Dissertation

Jointly Learning Features and Temporal Contingency for Prediction in Large Scale Datasets

Presented by
Tingyang Xu, M.S.

Major Advisor _____
Jinbo Bi

Associate Advisor _____
Sanguthevar Rajasekaran

Associate Advisor _____
Jun Yan

Associate Advisor _____
Jason K. Johannesen

University of Connecticut
2017

ACKNOWLEDGMENTS

My dissertation research was supported by multiple grants, including NSF grants IIS-1320586, DBI-1356655, and CCF-1514357, NIH grant R01DA037349, and VA grant I21 RX001731 (PI: Jinbo Bi).

Over the past six years I have received a lot of support and encouragement from my advisors, collaborators, family members and my friends. I would like to express my sincere gratitude to all of these individuals.

Firstly, I offer my deepest appreciation and acknowledgment to my major advisor Dr. Jinbo Bi, who has supported me throughout my research work with her patience, motivation and immense knowledge. She has also been a mentor, colleague and friend of mine. Without her guidance, I would achieve nothing. I could not imagine having a better advisor for my Ph.D. study.

Besides, I would appreciate my associate advisors, Dr. Sanguthevar Rajasekaran, Dr. Jun Yan and Dr. Jason K. Johannesen, for their insightful comments and encouragement. Their advices and questions have inspired me to broaden my research in various perspectives.

My sincere thanks also go to Dr. Jiangwen Sun, Dr. Soumitra Pal, Dr. Guoqing Chao, Dr. Ko-shin Chen, Dr. Xin Wang, Jin Lu, Aaron Palmer, Guannan Liang, Chao Shang, Qianqian Tong, and all other collaborators. We have worked together for countless hours of discussions, derivations, programming and proofreading I will remember those many sleepless nights that we were working hard in the lab and all the fun we have had in the past six years.

At last but not least, I would like to thank my family for supporting me throughout my Ph.D. study and my life.

Contents

Chapter 1

Introduction

1 Motivation and Challenges

Temporal data such as time series or longitudinal measurements are pervasive across almost all human endeavors, including finance [58, 4], science [65, 16, 7, 3], climate [37, 4], and genetics[75]. As such, it is hardly surprising that temporal data mining has attracted significant attention and research efforts. Several statistic methods have been developed to estimate temporal data, including generalized estimating equations (GEE) [35], Granger causality [19, 4, 37], quadratic inference functions (QIF) [54, 53, 5, 77], generalized mixture effect models [45, 79], etc. in order to explore the input-output relationships throughout time. However, as data acquisition technologies advances and the amount of data increases, it is difficult to employ these methods directly to analyze big datasets that contain ultra-high dimensions with complex structures and many subjects recorded in a long time period. The ultra-high dimensions make these methods inapplicable and the repeated measurements

in longitudinal studies break the assumptions required by these statistic methods. In contrast, many machine learning methods are able to discover the potential complex proximity structures of ultra-high dimensions in a big dataset. However, most of these methods are ineffective to model the correlations among an individual's repeated measurements. Therefore, it is currently a challenge to analyze temporal data taking into account the high dimensions, lagged time effects and sample correlations simultaneously.

Typically, longitudinal data are analyzed by extending generalized linear models (GLM) with different assumptions [13]. For example, the marginal modeling is one of the extensions of GLM which marginalizes the joint distribution of the temporal measures within subject and across subjects into a univariate normal distribution. Generalized estimating equations (GEE) [35] and Quadratic Inference Function (QIF) [54, 77] are the most widely used methods for marginal modeling, and estimate a predictive model to predict the current outcome based on longitudinal correlations and temporal feature effects. The resultant predictive models by GEE are generally more accurate than those of classic regression analysis that assumes independently and identically distributed (*i.i.d.*) observations [35]. However, GEE requires prespecification of working correlation. When the working correlation is assumed in a wrong form, the correlation structure presumed by GEE no longer results in optimal estimation of coefficients. Instead, QIF explores a linear combination of many possible correlation structures, so the estimator always exists. Researches on feature selection in longitudinal data lead to a new family of methods based on the penalized GEE (PGEE) [17] and penalized QIF (PQIF) [5].

However, the marginal models only estimate the correlations within individual's repeated measurements but not detect causal relationships from temporal changes

of features to the current outcome. In many studies, it is however necessary and the most important goal to model lagged causal effects of features while coping with the non-iid nature. For example, researchers record electroencephalogram (EEG) or functional magnetic resonance imaging (fMRI) to understand brain disorders. The changes in brain activities in an early stage usually predict a later symptom of a brain disorder.

Moreover, with advances in data acquisition technologies, ultra-high dimensional data with complex feature structure are collected in many disciplines and industrial areas. Often times, a single data example forms a high dimensional tensor itself. In a neuroscience study, as an example, researchers examine different EEG recordings to distinguish patient trials (recordings) with successful working memory from those without. A single EEG feature, such as the α signal amplitude, can be extracted at different brain information processing stages and from various scalp locations (or EEG electrodes). This single feature naturally forms a matrix with one dimension along the temporal line and the other along the spatial line. When multiple EEG features are extracted from an EEG recording, a trial can be represented by multiple such matrices altogether, forming a tensor, or more specifically a 3D array [8]. Moreover, repeated measurements of fMRI can create tensors in very high dimensions as well because the fMRI images themselves are a 3D volume [21, 47, 72, 22, 73, 63]. In order to use classic statistical tools, one has to flatten an example represented by a tensor into a long vector before building a regression or classification model, thus losing complex proximity structure.

Therefore, researchers have begun to leverage tensor techniques, such as low-rank tensor regression [25, 71], Schatten 1-norm [15, 56, 69], or latent tensor norm based regularization approaches [68], to directly build a regression or classification model

as a function of the tensor, preserving the multilinear data structure in the model [11, 67, 29, 87, 74]. However, these methods are usually formulated into non-convex optimization problems that are hard to solve. Moreover, they usually do not model the lagged influences over time, thus producing poor models such as those built for the repeated fMRI data.

To efficiently analyze big datasets, many efforts have been undertaken to create distributed or parallel machine learning algorithms. For example, stochastic gradient descent (SGD) [83, 70, 48, 2, 76, 34], alternating direction method of multipliers (ADMM) [9, 51, 26, 64], and (stochastic) dual coordinate ascent (DCA) algorithm [60, 80, 30, 39] are all trying to 'divide' a big dataset into smaller parts that can be solved independently. Then the final solution is reached by 'accumulating' the partial solutions using a single round of communications. Using distributed and parallel computing mechanisms, we can design schemes to solve prediction problems with large-scaled time-related datasets.

2 Overview

We propose in this thesis study to perform a longitudinal lasso and a sparse tensor-based QIF to select features from ultra-high dimensions and design efficient parallel and distributed algorithms for these methods. There are three major components in my dissertation research.

First, we propose to include lagged effects (i.e., features observed at multiple time points) in a matrix-based regressive model. The proposed method (Longitudinal LASSO, or long-LASSO) makes predictions based on lagged data from current and

previous time points. It decomposes the model coefficients into a summation of two components and imposes different block-wise LASSO to the two components. One regularizer is used to detect the contingency of specific time points whereas the other is used to select features. The proposed method learns simultaneously a structured correlation matrix from the temporal data which is similar to GEE, and a predictive model. The correlations among the outcomes observed at proximal time points may imply changing trends of the outcomes within each subject. We have also developed a family of methods where the outcome variable is assumed to follow a distribution from the exponential family, including Bernoulli, Gaussian and Poisson distributions.

The second approach is based on a quadratic inference function to model tensor data directly, which we call Tensor-QIF, and aims to construct a linear predictive model that selectively use features along each dimension of the tensor. If a data example is represented by a k-way tensor, the parameters in the linear model form another k-way tensor. In our approach, we decompose the k-way parameter tensor into a summation of k sparse k-way tensors. Each of these tensors is enforced sparsity along one direction of the parameter tensor. In other words, we impose a regularizer on a component tensor so it zeros out some of the $(k-1)$-way tensors along a direction of the k-way tensor. In order to take into account correlation structures in non-$i.i.d$ samples, such as the repeated measures within an individual, the proposed method employs QIF to estimate within-individual correlations while constructing a predictive model. Figure 2.1 shows the main idea of our approach using an example of a 3-way tensor. The original 3-way tensor is decomposed into a summation of three 3-way tensors. By sparsity-inducing regularization, each component tensor may shrink an entire plain along a specific direction of the 3-way tensor to zero.

The above two methods are novel and effectively model the correlated temporal

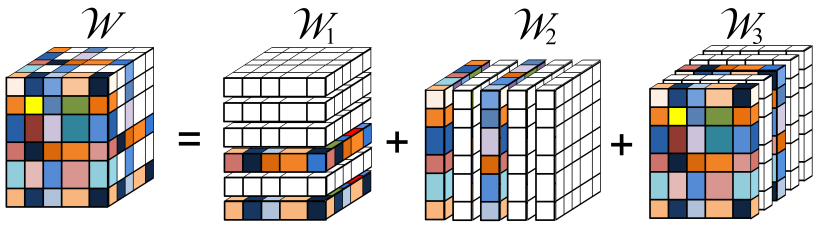


Figure 2.1: If we associate with each entry of the data tensor a weight in our additive predictive model, then the model coefficients form a tensor $\mathcal{W}$. If the coefficient tensor is sparse respectively in each mode, then the resultant model will be selective in terms of three different directions of $\mathcal{W}$: $\mathcal{W}_1, \mathcal{W}_2$, and $\mathcal{W}_3$.

data as a regularized risk minimization (RRM) formulation. The methods are sophisticated and advanced given they take into account complex data structures. Hence, it is necessary to explore efficient algorithms that implement these methods so they can be scalable and deployable to modern high performance computing (HPC) systems. Third, we design parallelized optimization solvers to solve the proposed RRM model and test on massive public datasets. Specifically, we propose a new hybrid stochastic dual coordinate ascent (hybrid-SDCA) algorithm, for a multi-core cluster, the most common high performance computing environment that consists of multiple nodes each having multiple cores and its own shared memory. We distribute data across nodes where each node solves a local problem in an asynchronous parallel fashion on its cores, and then the local updates are aggregated via another asynchronous across-node update. The proposed double asynchronous algorithm converges to a global solution for L-Lipschitz continuous loss functions at a linear convergence rate if a smooth convex loss function is used.

Chapter 2

Longitudinal LASSO: Jointly Learning Features and Temporal Contingency for Outcome Prediction

1 Introduction

A longitudinal study collects and analyzes repeated measurements of a set of features for a group of subjects through time. Longitudinal analyses are important in many areas, such as in social and behavioral science [65, 16, 7], in economics [58, 4], in climate [37, 4], and in genetics [75]. For example, to predict binge drinking of college students, a longitudinal study may be designed to monitor them weekly or even daily in terms of multiple covariates, such as, the level of stress, status of negative effects and social behaviors [7, 3]. The fluctuation of these covariates is used to analyze and predict binge drinking (the dependent or outcome variable) of a student at the

current observation time point. Changes of the covariates in the proximal time points are anticipated to alter the likelihood that a student binge drinks at the current observation point. To precisely understand how covariates affect the outcome, the analysis has to model not only the current values of the covariates but also their proximal values as well as take into account the correlation structure in the repeated measurements.

Typically, longitudinal data are analyzed by extending generalized linear models (GLM) with different assumptions, such as marginal models, random effects models, and transition models [13]. For example, a marginal model regresses the outcome on the current observation of features but factors in a within-subject correlation matrix that is estimated for a few proximal time points. In contrast, a random effects model reflects the variability among individuals rather than the population average comparing with marginal models. For marginal modeling, generalized estimating equations (GEE) are the most widely used methods which estimate a predictive model to predict the current outcome together with correlations among different outcomes observed temporally. The resultant predictive models are generally more accurate than those of classic regression analysis that assumes independently and identically distributed (*i.i.d.*) observations [35]. Research on feature selection in longitudinal data leads to a new family of methods based on the penalized GEE (PGEE) [17]. For random effects models, generalized linear mixture model (GLMM) [32, 41] is the major method. It explores natural heterogeneity across individuals in the regression coefficients and represents this heterogeneity by a probability distribution.

None of those extensions of GLM aim to detect causal relationships from temporal changes of covariates to the outcomes of the current effect. In many studies, it is however necessary and insightful to model simultaneously the correlation among out-

come records and the lagged causal effects of covariates [3]. For example, psychologists have identified that there is lagged effect in the alcohol use behavior. An individual's drinking today may be a response to an elevated level of stress two days back rather than the current day. It is actually an important question for psychologists to find out both which temporal points and which covariates influence the current outcome the most. This lagged effect is not used by temporal marginal modeling to make predictions.

On the other hand, researchers have developed machine learning approaches for longitudinal analysis that predict an outcome using feature values at multiple time points [4, 37]. For example, graphical Granger modeling [4], and grouped graphical Granger modeling [37] are insightful to explore the influences from past temporal information present in time series data in the modeling and understanding of the causal relationships. These methods assume that past values of certain time series features causally affect an outcome variable, and hence construct a model based on these values to predict future outcomes. Often, they estimate causality relationship (causal graph) among all features. However, these methods assume *i.i.d.* samples which are clearly violated in longitudinal data, and moreover they are incapable of selecting the most influential time points.

All existing methods either assume *i.i.d.* samples in Granger causality modeling or assume correlated samples but do not model *temporal* causal effects. Therefore, we propose a new learning formulation that constructs predictive models as functions of covariants not only from the current observation but also from multiple previous consecutive observations, and simultaneously determine the temporal contingency and the most influential features. The proposed method has the following advantages:

1. The proposed method makes predictions based on lagged data from current and previous time points. It decomposes the model coefficients into a summation of two components and impose different block-wise *least absolute shrinkage and selection operators* (LASSO) to the two components. One regularizer is used to detect the contingency of specific time points whereas the other is used to select covariates.

2. The proposed method also learns simultaneously a structured correlation matrix from the data. The correlations among the outcomes themselves imply the changing trend of the outcomes in the proximal time points within each subject.

3. We develop a family of methods where the outcome variable is assumed to follow a distribution from the exponential family, including Bernoulli, Gaussian and Poisson distributions. The formulations for these distributions are discussed in Section 3.3.

4. We provide the convergence analysis in Section 3.1 and asymptotic analysis in Section 3.2 to show that the proposed algorithm can find the optimal solution for the predictive models.

We have empirically compared the proposed method against the state of the art on both synthetic and real world datasets. The computational results demonstrate the effectiveness and the capability of our approach.

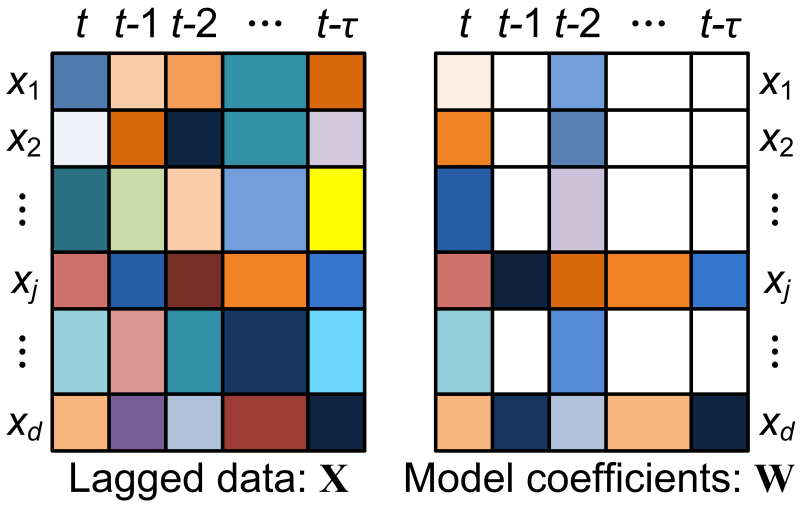


Figure 1.1: The outcome y_t at time t can be relevant to multiple covariates $x_1, x_2, \cdots, x_d$ observed at current and several previous time points $t-1, t-2, \cdots, t-\tau$, which forms a data matrix $\mathbf{X}$ (left). If we associate with each entry of this matrix a weight in our additive prediction model, then our model coefficients form a matrix $\mathbf{W}$ (right). If the coefficient matrix is sparse, then the resultant model will be selective in terms of covariates and time points.

2 Method

In our approach, the predictive model takes the form of the *trace* of the product of the lagged data $\mathbf{X}$ and the model coefficient matrix $\mathbf{W}$ as shown in Figure 1.1. The model coefficients are organized into a matrix rather than a vector used in traditional analysis because this way reflects the structure in the lagged data. Note that the lagged observations of y can also be included in the data matrix $\mathbf{X}$ to be used in the predictive model. For notational convenience, we just use $\mathbf{X}$ to represent the data that are used to form the model.

We first briefly review two most relevant sets of longitudinal analytics in Section 2.1 which will help elucidate the advantages of our proposed formulation.

2.1 Preliminaries

We introduce the notation that is used throughout the paper. A bold lower case letter denotes a vector, such as $\mathbf{v}$. The $\|\mathbf{v}\|_p$ refers to the ℓ_p norm of a vector $\mathbf{v}$, which is formed as $\|\mathbf{v}\|_p = (\sum_{i=1}^{d} |v_i|^p)^{1/p}$, where v_i is the i-th component of $\mathbf{v}$ and d is the length of $\mathbf{v}$. A bold upper case letter denotes a matrix such as $\mathbf{M}$. Similarly, $\mathbf{m}_{(i,)}$, $\mathbf{m}_{(,j)}$ and m_{ij} represent the i-th row, j-th column and (i,j)-th component of $\mathbf{M}$, respectively. The Frobenius norm and $\ell_{p,q}$ norm of a matrix $\mathbf{M}$ refer, respectively, to $\|\mathbf{M}\|_F$, which is equal to $(tr(\mathbf{M}^\top \mathbf{M}))^{1/2}$, and $\|\mathbf{M}\|_{p,q}$, defined by $\left(\sum_{i=1}^{n} \left(\|\mathbf{m}_{(i,)}\|_q\right)^p\right)^{1/p}$, where n is the number of rows in $\mathbf{M}$, and $tr(\mathbf{M})$ indicates the trace of $\mathbf{M}$. We assume that vect($\mathbf{M}$) is the column-major vectorization of $\mathbf{M}$, which is defined as vect($\mathbf{M}$) $= (\mathbf{m}_{(,1)}^\top, \cdots, \mathbf{m}_{(,k)}^\top)^\top$ assuming k columns are in $\mathbf{M}$. Then, $\langle \mathbf{M}_1, \mathbf{M}_2 \rangle$ is the inner product of two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ that is computed as the inner product of vect($\mathbf{M}_1$) and vect($\mathbf{M}_2$). The operator reshape($\mathbf{v}$) re-shapes $\mathbf{v}$ into a matrix of a proper size determined by the specific context.

Assume that we are given data of m number of individuals on d number of features (independent variables) that are repeatedly measured at n_i time points for each individual i. The data of each individual i is represented by a matrix $\mathbf{X}^{(i)}$ of size $d \times n_i$, and $\mathbf{x}_t^{(i)}$ refers to the d-entry data vector of individual i at time point t. Without loss of generality, we assume that all individuals have data at the same consecutive time points ($n_i = n$) to simplify the notation and the subsequent analysis. Data on the dependent variable (outcome) is also given in $\mathbf{y}^{(i)}$ of length n that contains the observations at the n time points for individual i. Typically, a longitudinal study aims to estimate the effect of covariates on the dependent variable.

Granger Causality

The notion of *Granger Causality* was introduced by the Nobel prize winning economist, Clive Granger, and has proven useful in time series analysis [19]. It is based on the intuition that if a time series variable causally affects another, the past observations of the former should be useful in predicting the future outcome of the latter.

Specifically, a time series observation x is said to *Granger cause* another time series outcome, y, if the regressing for y in terms of past y and x is significantly better than the regressing just with past values of y. The so-called Granger test first performs two regressions:

$$y_t^{(i)} = \sum_{j=1}^{\tau} \left(a_j y_{t-j}^{(i)} + w_j^\top x_{t-j}^{(i)} \right), \tag{2.1}$$

and $y_t^{(i)} = \sum_{j=1}^{\tau} a_j y_{t-j}^{(i)}$, where τ is the maximum "lag" in the past observations, and then uses a hypothesis test such as an F-test to determine if the outcome y_t can be predicted significantly better from the past covariate x. Recent graphical Granger models [4, 37] extend it from a single time series covariate $\mathbf{x}$ to multiple covariates $\mathbf{X}$. They learn the coefficients $\mathbf{a}$ and $\mathbf{w}$'s with LASSO type of regularizers and evaluate if coefficients are non-zero for Granger causality.

Generalized Estimating Equations (GEE)

GEE estimates the parameters of a GLM while taking into account the correlations in the training examples. Similar to GLM, it assumes that the dependent variable comes from a class of distributions known as the exponential family. For each member in this family, there exists a link function that can be used to translate the nonlinear model into a linear model. The expectation of the outcome $y_t^{(i)}$ for subject i at time

t is computed as:

$$E(y_t^{(i)}) = \mu_t^{(i)} = g^{-1}(\eta_t^{(i)}), \tag{2.2}$$

where $\mu_t^{(i)}$ represents the mean model, g^{-1} is the inverse of a link function g in a GLM [40], and $\eta_t^{(i)} = \left(\mathbf{x}_t^{(i)}\right)^\top \mathbf{w}$. The variance of $y_t^{(i)}$ is computed as $\text{var}(y_t^{(i)}) = \text{var}(\mu_t^{(i)})/\phi$ where ϕ is a scaling parameter that may be known or estimated.

GEE presumes a so-called working correlation structure, typically denoted by $\mathbf{R}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ is a parameter to be determined from data. The common choices of $\mathbf{R}(\boldsymbol{\alpha})$ include exchangeable, tri-diagonal and the first-order autoregressive (AR(1)) formula [35]. The exchangeable correlation structure, also called *equi-correlation*, assumes that $corr(y_{it}, y_{it'}) = \alpha$ for all $t \neq t'$. The tri-diagonal structure uses a tridiagonal matrix as $\mathbf{R}(\boldsymbol{\alpha})$ where $corr(y_{it}, y_{it'}) = \alpha$ if $t' = t \pm 1$ or 0 otherwise. The AR(1) formula assumes a correlation structure along continuous time, and uses $corr(y_{it}, y_{it'}) = \alpha^{|t-t'|}$.

To estimate the regression coefficients $\mathbf{w}$, GEE uses the the estimating equations that are formulated, in general, by setting the derivative of an appropriate loss function to 0. Although a loss function may not be explicitly written out, the estimating equations always can be computed by

$$EE(\mathbf{w}, \boldsymbol{\alpha}) = \sum_{i=1}^{m} \left(\mathbf{D}^{(i)}\right)^\top \left(\boldsymbol{\Sigma}^{(i)}\right)^{-1} \mathbf{s}^{(i)} = 0. \tag{2.3}$$

where the $n \times d$ matrix $\mathbf{D}^{(i)} = \partial \boldsymbol{\mu}^{(i)}/\partial \mathbf{w}$ where $\boldsymbol{\mu}^{(i)}$ combines all $\mu_t^{(i)}, \forall t = 1, \cdots, n$ into a vector, $\mathbf{s}^{(i)} = \mathbf{y}^{(i)} - \boldsymbol{\mu}^{(i)}(\mathbf{w})$. The $n \times n$ matrix $\boldsymbol{\Sigma}^{(i)}$ is the estimated covariance structure as:

$$\boldsymbol{\Sigma}^{(i)}(\boldsymbol{\alpha}) = \left(\mathbf{A}^{(i)}\right)^{1/2} \mathbf{R}(\boldsymbol{\alpha}) \left(\mathbf{A}^{(i)}\right)^{1/2} /\phi \tag{2.4}$$

where $\mathbf{A}^{(i)}$ is an $n \times n$ diagonal matrix with $\text{var}(\mu_t^{(i)})$ as the t-th diagonal element. Algorithms are given in [35] columns, to compute $\mathbf{w}$ and $\boldsymbol{\alpha}$ for the different choices of $\mathbf{R}(\boldsymbol{\alpha})$.

2.2 The Proposed Formulation

In our approach, each training example consists of the current and τ previous records of the repeated measurements. Let

$$\mathbf{X}_{(i;t)} = [\mathbf{x}_t^{(i)}, \mathbf{x}_{t-1}^{(i)}, \cdots, \mathbf{x}_{t-\tau}^{(i)}]$$

be a $d \times (\tau + 1)$ data matrix for subject i. Given T total measurements for each subject, the index t of $\mathbf{X}_{(i;t)}$ starts from $\tau + 1$ in order to have enough previous observations in the first training example. Hence, there are totally $n = T - \tau$ training examples for each subject. If $\mathbf{X}_{(i;t)}$ includes previous $\tau + 1$ values of $y^{(i)}$ as a feature, then the model $y_t^{(i)} = tr\left(\mathbf{X}_{(i;t)}^{\top}\mathbf{W}\right)$ where $\mathbf{W} = [\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_\tau]$ essentially gives the same model like Eq.(2.1) in the graphical Granger models.

The Granger models would assume that the training examples are *i.i.d.*. However, the consecutive examples are not mutually independent because they contain overlapping records (e.g., $\mathbf{X}_{(i;t)}$ and $\mathbf{X}_{(i;t+1)}$ share $\tau - 1$ records $\mathbf{x}_t^{(i)}$, $\cdots$, $\mathbf{x}_{t-\tau+1}^{(i)}$). GEE provides a mechanism to estimate the sample correlation simultaneously while constructing predictive models, and to extend the linear models to generalized linear models. To apply GEE to our model, we replace $\eta_t^{(i)}$ used in GEE by the following formula

$$\eta_t^{(i)} = tr\left(\mathbf{X}_{(i;t)}^{\top}\mathbf{W}\right). \tag{2.5}$$

Substituting Eq.(2.5) for η in Eq.(2.2) yields a formulation similar to GEE. The regression coefficients $\mathbf{W}$ can be estimated through the well-developed GEE estimators. In particular, the quasi-likelihood methods of GEE estimate $\mathbf{W}$ by minimizing a loss function that is defined via the model deviance. The model deviance measures the difference between the log-likelihood of the estimated mean model $\boldsymbol{\mu}^{(i)}$ and that of the observed values $\mathbf{y}^{(i)}$. For instance, the model deviance for a linearly regressive response is written by $Dev^{(i)}(\mathbf{W}, \boldsymbol{\alpha}) = (\mathbf{y}^{(i)} - \boldsymbol{\mu}^{(i)})^\top \mathbf{R}(\boldsymbol{\alpha})(\mathbf{y}^{(i)} - \boldsymbol{\mu}^{(i)})$ where $\mathbf{y}^{(i)}$ contains the observed responses for subject i, and $\boldsymbol{\mu}^{(i)}$ is the estimated expectations of y for subject i. If the response follows an arbitrary distribution, the model deviance may not correspond to an explicit function. For the exponential family, it takes a special form as discussed in Theorem 1 below, which is still complicated. We denote by $Dev^{(i)}(\mathbf{W}, \boldsymbol{\alpha})$ the deviance occurred on subject i. GEE minimizes a loss function of $\sum_{i=1}^{m} Dev^{(i)}(\mathbf{W}, \boldsymbol{\alpha})$ for the optimal $\mathbf{W}$ by solving the *estimating equations*, i.e., taking the derivatives of the loss function and setting them to 0.

Now, to select among features and discover the most influential time points in predicting y over time, (and also to control the model capacity,) we apply regularizers to the model parameters. We first decompose $\mathbf{W}$ into a summation of two components as $\mathbf{W} = \mathbf{U} + \mathbf{V}$ and apply different regularizers to $\mathbf{U}$ and $\mathbf{V}$. The block-wise LASSO, such as the $\ell_{1,2}$ matrix norm, is widely-used in multi-task learning or feature selection with group structures, but has not been explored within the GEE setting. To the best of our knowledge, it has not been studied in longitudinal analytics how to produce shrinkage effects simultaneously on both features and contingent temporal records through proper regularization. The general $\ell_{1,p}$ matrix norm [85] calculates the sum of the ℓ_p norms of the rows in a matrix. Regularizers based on the $\ell_{1,p}$ norms encourage row sparsity by shrinking the entire rows to have zero entries.

In our parameter matrix $\mathbf{W}$, rows correspond to features and columns correspond to the observation time points. If we apply the $\ell_{1,2}$ norm to $\mathbf{U}$ (row-wisely), the optimal solution of $\mathbf{U}$ will contain rows with all zero entries. Thus, a selected subset of features in the $\tau+1$ observations will be used in the predictive model to predict the current outcome. The $\ell_{1,2}$ norm of $\mathbf{V}^\top$ (column-wisely) encourages to select among columns of $\mathbf{V}$. If the k-th column of $\mathbf{V}$ contains the largest values in the selected columns, the current outcome is most contingent on the previous $(k-1)$-th record, thus having the $(k-1)$ "lagged" effect. Overall, we solve the following optimization problem for the best model parameters $\mathbf{W}$ which is computed as $\mathbf{U}+\mathbf{V}$:

$$\min_{\mathbf{U},\mathbf{V}} \quad \sum_{i=1}^{m} Dev^{(i)}(\mathbf{U}+\mathbf{V},\boldsymbol{\alpha}) + \lambda_1\|\mathbf{U}\|_{1,2} + \lambda_2\|\mathbf{V}^\top\|_{1,2} \tag{2.6}$$

where $\mathbf{W}$ in the deviance is simply replaced by $\mathbf{U}+\mathbf{V}$.

The optimization of Eq.(2.6) is challenging. In general, even solving the GEE formulation is not easy as it estimates not only the model expectation but also the variance term $\boldsymbol{\Sigma}^{(i)}$. The algorithm that solves the GEE (i.e., the estimating equations) applies the Newton-Raphson method in the iterative reweighted least squares (IRLS) procedure [17] to estimate $\mathbf{w}$ and $\boldsymbol{\Sigma}^{(i)}$. However, this method does not solve any formula that uses regularizers. By modifying the Newton-Raphson method or shooting algorithm [17], it can be extended only to the regularizers that are decomposable into individual parameters w_j. For instance, the ℓ_1 vector norm of $\mathbf{w}$ can be decomposed into the summation of individual $|w_j|$, $j=1,\cdots,d$. The $\ell_{1,2}$ matrix norm, unfortunately, can not be decomposed in such a way. Therefore, we have developed an accelerated gradient descent method based on the fast iterative shrinkage-thresholding algorithm (FISTA) [6]. Further, the following theorem shows

that Eq.(2.6) is a convex optimization problem in terms of $\mathbf{W}$. Our algorithm can be proved to find the global optimal solution $\mathbf{W}$ of Eq.(2.6) when $\boldsymbol{\alpha}$ is fixed (to a consistent estimate given by GEE).

Theorem 1. *The first term of Eq.(2.6) is convex and continuously differentiable with respect to* $\mathbf{U}$ *and* $\mathbf{V}$ *if the distribution of* $\mathbf{y}^{(i)}$ *is in a natural exponential family and the link function is continuous.*

Proof. First, let us recall that the probability density function of a distribution in the exponential family takes the following form:

$$f(y_t^{(i)}) = \exp\left\{\frac{y_t^{(i)}\eta_t^{(i)} - b(\eta_t^{(i)})}{a_t^{(i)}(\phi)} + c(y_t^{(i)}, \phi)\right\},$$

where $a_t^{(i)}(\phi)$, $b(\eta_t^{(i)})$, and $c(y_t^{(i)}, \phi)$ are known functions and specified for each member of the exponential family, and $\eta_t^{(i)}$ is a parameter in the mean as defined in Eq.(2.2). Typically, $a_t^{(i)}(\phi) = \phi$. Then, the deviance of the exponential family can be computed as

$$Dev = 2\frac{\sum_{i=1}^{m}\left(y_t^{(i)}(\tilde{\eta}_t^{(i)} - \hat{\eta}_t^{(i)}) - b(\tilde{\eta}_t^{(i)}) + b(\hat{\eta}_t^{(i)})\right)}{\phi},$$

where $\tilde{\eta}_t^{(i)}$ denotes the true value under a saturated model, $\hat{\eta}_t^{(i)}$ denotes the fitted values of the model. Thus, $\tilde{\eta}_t^{(i)}$ and $b(\tilde{\eta}_t^{(i)})$ are constant in model fitting. The derivative of b always satisfies $b'(\eta_t^{(i)}) = \mu_t^{(i)}$. Moreover, it has been proved that $b(\hat{\eta}_t^{(i)})$ is a convex function on the natural parameter space $\mathbf{H} = \{\hat{\boldsymbol{\eta}} | b(\hat{\boldsymbol{\eta}}) < \infty\}$ [59]. Thus, the deviance contains either linear terms or a convex term with respect to $\hat{\eta}$. In our model (2.5), $\hat{\eta}$ is linear with respect to $\mathbf{W}$. Hence, the deviance term in Eq.(2.6) is convex with respect to $\mathbf{U}$ and $\mathbf{V}$.

Moreover, it is true that $b'(\hat{\eta}_t^{(i)}) = \hat{\mu}_t^{(i)} = g^{-1}(\hat{\eta}_t^{(i)})$ which is the inverse of a

continuous link function [59]. The first term of Eq.(2.6) is continuously differentiable with respect to $\mathbf{U}$ and $\mathbf{V}$. Thus, theorem 1 holds. □

2.3 Optimization Algorithm

To solve Eq.(2.6), we design an alternating optimization algorithm that alternates between optimizing two working sets of variables: one set consisting of $\mathbf{U}$ and $\mathbf{V}$ and the other consisting of $\boldsymbol{\alpha}$.

(a) Find* $\mathbf{U}$ *and* $\mathbf{V}$ *when* $\boldsymbol{\alpha}$ *is fixed

When $\boldsymbol{\alpha}$ is fixed, the objective function of Eq.(2.6), denoted by $f(\mathbf{U}, \mathbf{V})$, is convex with a continuously differentiable part $\ell(\mathbf{U}, \mathbf{V})$ that is the deviance and a nonsmooth part $R(\mathbf{U}, \mathbf{V})$ that constitutes the two regularizers. We hence have

$$f(\mathbf{U}, \mathbf{V}) = \ell(\mathbf{U}, \mathbf{V}) + R(\mathbf{U}, \mathbf{V}).$$

We develop a FISTA algorithm in the following iterative procedure to find optimal $\mathbf{U}$ and $\mathbf{V}$.

Denote the iterates at the k-th iteration by $\mathbf{U}_k$ and $\mathbf{V}_k$. Let $\nabla_{\mathbf{U}}\ell(\mathbf{U}, \mathbf{V})$, $\nabla_{\mathbf{V}}\ell(\mathbf{U}, \mathbf{V})$ be the partial derivative of $\ell(\mathbf{U}, \mathbf{V})$ with respect to $\mathbf{U}$ and $\mathbf{V}$, respectively, For any given point $(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$, the following $Q_{L,\tilde{\mathbf{U}},\tilde{\mathbf{V}}}(\mathbf{U}, \mathbf{V})$ is a *well-defined* proximal map for the non-smooth R

$$\begin{aligned} Q_{L,\tilde{\mathbf{U}},\tilde{\mathbf{V}}}(\mathbf{U}, \mathbf{V}) = \ell(\tilde{\mathbf{U}}, \tilde{\mathbf{V}}) + R(\mathbf{U}, \mathbf{V}) + \langle \nabla_{\mathbf{U}}\ell(\tilde{\mathbf{U}}, \tilde{\mathbf{V}}), \mathbf{U} - \tilde{\mathbf{U}} \rangle + \frac{L}{2}\|\mathbf{U} - \tilde{\mathbf{U}}\|_F^2 \\ + \langle \nabla_{\mathbf{V}}\ell(\tilde{\mathbf{U}}, \tilde{\mathbf{V}}), \mathbf{V} - \tilde{\mathbf{V}} \rangle + \frac{L}{2}\|\mathbf{V} - \tilde{\mathbf{V}}\|_F^2. \end{aligned}$$

If $\ell(\mathbf{U}, \mathbf{V})$ has Lipschitz continuous gradient with Lipschitz modulus L. Then, ac-

cording to the Lemma 2.1 in [6], the inequality

$$f(\mathbf{U}, \mathbf{V}) \leq Q_{L,\tilde{\mathbf{U}},\tilde{\mathbf{V}}}(\mathbf{U}, \mathbf{V}).$$

holds indicating that $Q_{L,\tilde{\mathbf{U}},\tilde{\mathbf{V}}}(\mathbf{U}, \mathbf{V})$ is the upper bound of $f(\mathbf{U}, \mathbf{V})$.

Starting from an initial point $(\mathbf{U}_0, \mathbf{V}_0)$, we iteratively search for the optimal solution. At each iteration k, we first use the iterates $(\mathbf{U}_{k-1}, \mathbf{V}_{k-1})$ and $(\mathbf{U}_{k-2}, \mathbf{V}_{k-2})$ to compute (at the first iteration, $(\tilde{\mathbf{U}}_1, \tilde{\mathbf{V}}_1) = (\mathbf{U}_0, \mathbf{V}_0)$)

$$\begin{aligned} \tilde{\mathbf{U}}_k &= \mathbf{U}_{k-1} + \left(\frac{t_{k-1} - 1}{t_k}\right)(\mathbf{U}_{k-1} - \mathbf{U}_{k-2}), \\ \tilde{\mathbf{V}}_k &= \mathbf{V}_{k-1} + \left(\frac{t_{k-1} - 1}{t_k}\right)(\mathbf{V}_{k-1} - \mathbf{V}_{k-2}), \end{aligned} \tag{2.7}$$

where t_k is a scalar and updated at each iteration as:

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}. \tag{2.8}$$

Then, we solve the following problem

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \quad & \langle \nabla_{\mathbf{U}} \ell_k, \mathbf{U} - \tilde{\mathbf{U}}_k \rangle + \frac{L}{2}\|\mathbf{U} - \tilde{\mathbf{U}}_k\|_F^2 + \langle \nabla_{\mathbf{V}} \ell_k, \mathbf{V} - \tilde{\mathbf{V}}_k \rangle + \frac{L}{2}\|\mathbf{V} - \tilde{\mathbf{V}}_k\|_F^2 \\ & + R(\mathbf{U}, \mathbf{V}) \end{aligned} \tag{2.9}$$

for a solution $(\mathbf{U}_k, \mathbf{V}_k)$, where $\nabla_{\mathbf{U}} \ell_k$ and $\nabla_{\mathbf{V}} \ell_k$ are respectively the partial derivatives of ℓ computed at $(\tilde{\mathbf{U}}_k, \tilde{\mathbf{V}}_k)$, and L acts as a learning step size.

Since there is no interacting term between $\mathbf{U}$ and $\mathbf{V}$ in Eq.(2.9), the problem can

be decomposed into two separate subproblems as follows:

$$\min_{\mathbf{U}} \langle \nabla_{\mathbf{U}} \ell_k, \mathbf{U} - \tilde{\mathbf{U}}_k \rangle + \frac{L}{2} \|\mathbf{U} - \tilde{\mathbf{U}}_k\|_F^2 + \lambda_1 \|\mathbf{U}\|_{1,2}, \tag{2.10}$$

$$\min_{\mathbf{V}} \langle \nabla_{\mathbf{V}} \ell_k, \mathbf{V} - \tilde{\mathbf{V}}_k \rangle + \frac{L}{2} \|\mathbf{V} - \tilde{\mathbf{V}}_k\|_F^2 + \lambda_2 \|\mathbf{V}^\top\|_{1,2}. \tag{2.11}$$

The two subproblems share the same structure and thus can be solved following the same procedure. Hence, we only show how to solve (2.10) for the best $\mathbf{U}$.

Eq.(2.10) is equivalent to the following problem

$$\min_{\mathbf{U}} \frac{1}{2} \left\| \mathbf{U} - \left(\tilde{\mathbf{U}}_k - \frac{1}{L} \nabla_{\mathbf{U}} \ell_k \right) \right\|_F^2 + \frac{\lambda_1}{L} \|\mathbf{U}\|_{1,2}$$

after omitting constants, and this problem has a closed-form solution where each row of $\mathbf{U}_k$, $\mathbf{U}^k_{(i,)}$ is:

$$\mathbf{U}^k_{(i,)} = \max \left(0, 1 - \frac{\lambda_1}{L \|\mathbf{P}^{(k)}_{(i,)}\|_2} \right) \mathbf{P}^{(k)}_{(i,)},$$

and $\mathbf{P}^{(k)} = \tilde{\mathbf{U}}_k - \frac{1}{L} \nabla_{\mathbf{U}} \ell_k$. The gradient vector $\nabla_{\mathbf{U}} \ell_k$ (i.e., the gradient of the deviance) can be computed by Eq.(2.3) with the fixed $\boldsymbol{\alpha}$, i.e.

$$\nabla_{\mathbf{U}} \ell_k = \text{reshape} \left(\sum_{i=1}^{m} \left(\mathbf{D}^{(i)} \right)^\top \left(\boldsymbol{\Sigma}^{(i)} \right)^{-1} \mathbf{s}^{(i)}_k \right) \tag{2.12}$$

where $\mathbf{s}^{(i)}_k = \mathbf{y}^{(i)} - \boldsymbol{\mu}^{(i)}$, and $\mu^{(i)}_t = g^{-1}(tr(\mathbf{X}^\top_{(i;t)}(\tilde{\mathbf{U}}_k + \tilde{\mathbf{V}}_k)))$.

In the above discussion, the Lipschitz modulus L is computed and given. However, the calculation of L can be computational expensive. We therefore follow the similar argument in [18] to find a proper approximation L_k at each iteration k starting from

$L_0 > 0$. Recall that the Lipschitz constant L is defined:

$$L = \max_{\mathbf{W}} \lambda_{\max} \left(\nabla \nabla \ell_{\mathbf{W}} \right)$$

where $\lambda_{\max}(\cdot)$ indicates the maximum singular value of the Hessian of ℓ. Decompose the Hessian matrix $\nabla \nabla \ell_{\mathbf{W}}|_{\mathbf{W} \to 0}$ into $\mathbf{M}^\top \mathbf{M}$ where $\mathbf{M} \in \mathbb{R}^{d(\tau+1) \times q}$ and q is the rank of the Hessian matrix. We have an upper bound of L as follows:

$$L \leq ||\mathbf{M}||_{\infty,1} ||\mathbf{M}^\top||_{\infty,1}. \tag{2.13}$$

We use the upper bound $\tilde{L}$ in Eq.(2.13) as L in our iterations. Using this upper bound may increase the number of iterative steps for convergence. Algorithm 1 summarizes the steps for finding optimal $\mathbf{U}$ and $\mathbf{V}$ with fixed $\boldsymbol{\alpha}$.

Algorithm 1: Search for optimal $\mathbf{U}$ and $\mathbf{V}$ with fixed $\boldsymbol{\alpha}$

Input: $\mathbf{X}$, $\mathbf{y}$, $\boldsymbol{\Sigma}$, λ_1, λ_2
Output: $\mathbf{U}$, $\mathbf{V}$
1. $k = 1$, compute $\tilde{L}$ and initialize $t_1 = 1$, $\mathbf{U}_0 = \tilde{\mathbf{U}}_1 = \mathbf{0}$ and $\mathbf{V}_0 = \tilde{\mathbf{V}}_1 = \mathbf{0}$;
2. Solve Eq.(2.9) to obtain $\mathbf{U}_k$ and $\mathbf{V}_k$.
3. Compute t_{k+1} by Eq.(2.8).
4. Compute $\tilde{\mathbf{U}}_{k+1}$ and $\tilde{\mathbf{V}}_{k+1}$ by Eq.(2.7).
5. $k = k + 1$.
Repeat $2 \sim 5$ until convergence.

(b) Find $\boldsymbol{\alpha}$ when $\mathbf{U}$ and $\mathbf{V}$ are fixed

When $\mathbf{U}$ and $\mathbf{V}$ are fixed, the regularizers no longer appear in the objective of Eq.(2.6). Eq.(2.6) is degenerated into just the GEE formula with $\boldsymbol{\alpha}$ as the variables. Hence, $\boldsymbol{\alpha}$ can be estimated via the standard GEE procedure, i.e., from the current

Pearson residuals defined by:

$$\gamma_t^{(i)} = \frac{y_t^{(i)} - tr\left(\left(\mathbf{X}_{(i;t)}\right)^\top (\mathbf{U}+\mathbf{V})\right)}{(\sigma_{t,t}^{(i)})^{(1/2)}}.$$

where $\sigma_{t,t}^{(i)}$ is the t-th diagonal entry in the matrix $\mathbf{\Sigma}^{(i)}$ [35]. The specific estimator of $\boldsymbol{\alpha}$ depends on the choices of $\mathbf{R}(\boldsymbol{\alpha})$. This GEE-based procedure has been shown to find a *consistent* estimate of $\boldsymbol{\alpha}$ [35].

Let $N = mn$ be the total number of training examples, and $p = d(\tau+1)$ be the practical number of parameters in $\mathbf{W}$. A general approach to estimating $\mathbf{R}$ is given by:

$$r_{j,k} = \sum_{i=1}^{m} \frac{\gamma_j^{(i)}\gamma_k^{(i)}}{N-p}, \tag{2.14}$$

for $j = 1, \cdots, n$, and $k = 1, \cdots, n$. In addition, the scaler parameter ϕ in Eq.(2.4) can be estimated as follows:

$$\phi = (N-p)/\sum_{i=1}^{m}\sum_{t=1}^{n}\left(\gamma_t^{(i)}\right)^2. \tag{2.15}$$

Algorithm 2 depicts the overall procedure for solving Eq.(2.6).

Algorithm 2: Main algorithm - Jointly select features and temporal points

Input: $\mathbf{X}$, $\mathbf{y}$, λ_1, λ_2
Output: $\mathbf{U}$, $\mathbf{V}$
1. Set $\mathbf{R}(\alpha) = \mathbf{I}$;
2. Solve for $\mathbf{U}$ and $\mathbf{V}$ using Algorithm 1.
3. Estimate α using a proper estimator in [35] and compute $\mathbf{R}(\alpha)$ by Eq.(2.14) and ϕ by Eq.(2.15).
Repeat $2 \sim 3$ until convergence.

3 Theoretical Analysis

We provide a convergence analysis for Algorithm 1 and an asymptotic analysis for the proposed formulation.

3.1 Convergence Analysis

We show that Algorithm 1 converges to the optimal solution with a convergence rate of $O(1/k^2)$. The proof follows largely the arguments in [6]. We only provide a sketch here.

Theorem 2. *Let* $\mathbf{U}_k$ *and* $\mathbf{V}_k$ *be the pair of the matrix generated by Algorithm 1. Then for any* $k \geq 1$

$$f(\mathbf{U}_k, \mathbf{V}_k) - f(\hat{\mathbf{U}}, \hat{\mathbf{V}}) \leq \frac{2\tilde{L}\left(||\mathbf{U}_0 - \hat{\mathbf{U}}||_F^2 + ||\mathbf{V}_0 - \hat{\mathbf{V}}||_F^2\right)}{(k+1)^2}$$

where $(\hat{\mathbf{U}}, \hat{\mathbf{V}})$ *is a globally optimal solution of Eq.(2.6).*

Proof. We start with defining the following quantities

$$v_k = f(\mathbf{U}_k, \mathbf{V}_k) - f(\hat{\mathbf{U}}, \hat{\mathbf{V}}), \quad a_k = \frac{2}{L_k} t_k^2 v_k,$$
$$b_k = ||t_k \mathbf{U}_k - (t_k - 1)\mathbf{U}_{k-1} - \hat{\mathbf{U}}||_F^2 + ||t_k \mathbf{V}_k - (t_k - 1)\mathbf{V}_{k-1} - \hat{\mathbf{V}}||_F^2,$$
$$c = ||\tilde{\mathbf{U}}_1 - \hat{\mathbf{U}}||_F^2 + ||\tilde{\mathbf{V}}_1 - \hat{\mathbf{V}}||_F^2 = ||\mathbf{U}_0 - \hat{\mathbf{U}}||_F^2 + ||\mathbf{V}_0 - \hat{\mathbf{V}}||_F^2,$$

where $\tilde{\mathbf{U}}_1 = \mathbf{U}_0$, $\tilde{\mathbf{V}}_1 = \mathbf{V}_0$, and subsequent $\tilde{\mathbf{U}}_k$ and $\tilde{\mathbf{V}}_k$ are defined by Eq.(2.7). Following the proof of Theorem 4.4 in [6], in the first iteration, given $t_1 = 1$, we have $a_1 = \frac{2}{L_1} v_1$, and $b_1 = ||\mathbf{U}_1 - \hat{\mathbf{U}}||_F^2 - ||\mathbf{V}_1 - \hat{\mathbf{V}}||_F^2$. We show that $a_1 + b_1 \leq c$ by applying

Lemma 2.3 in [6], which yields

$$
\begin{aligned}
&f(\hat{\mathbf{U}}, \hat{\mathbf{V}}) - f(\mathbf{U}_1, \mathbf{V}_1) = -v_1 \\
\geq &\frac{L_1}{2}||\mathbf{U}_1 - \tilde{\mathbf{U}}_1||_F^2 + L_1\langle \tilde{\mathbf{U}}_1 - \hat{\mathbf{U}}, \mathbf{U}_1 - \tilde{\mathbf{U}}_1\rangle + \frac{L_1}{2}||\mathbf{V}_1 - \tilde{\mathbf{V}}_1||_F^2 + L_1\langle \tilde{\mathbf{V}}_1 - \hat{\mathbf{V}}, \mathbf{V}_1 - \tilde{\mathbf{V}}_1\rangle \\
= &\frac{L_1}{2}(||\mathbf{U}_1 - \hat{\mathbf{U}}||_F^2 - ||\tilde{\mathbf{U}}_1 - \hat{\mathbf{U}}||_F^2) + \frac{L_1}{2}(||\mathbf{V}_1 - \hat{\mathbf{V}}||_F^2 - ||\tilde{\mathbf{V}}_1 - \hat{\mathbf{V}}||_F^2).
\end{aligned}
$$

Reorganizing the above inequality yields

$$
\frac{2}{L_1}t_1^2 v_1 + ||\mathbf{U}_1 - \hat{\mathbf{U}}||_F^2 + ||\mathbf{V}_1 - \hat{\mathbf{V}}||_F^2 \leq ||\tilde{\mathbf{U}}_1 - \hat{\mathbf{U}}||_F^2 + ||\tilde{\mathbf{V}}_1 - \hat{\mathbf{V}}||_F^2
$$

Thus, $a_1 + b_1 \leq c$ holds.

Then, according to Lemma 4.1 in [6], we have for every $k \geq 1$, $a_k - a_{k+1} \geq b_{k+1} - b_k$, together with $a_1 + b_1 \leq c$, which derives into the following inequality,

$$
c \geq a_1 + b_1 \geq a_2 + b_2 \geq \cdots \geq a_k + b_k \geq a_k.
$$

Therefore, we obtain that

$$
\frac{2}{L_k}t_k^2 v_k \leq ||\mathbf{U}_0 - \hat{\mathbf{U}}||_F^2 + ||\mathbf{V}_0 - \hat{\mathbf{V}}||_F^2, \tag{3.1}
$$

Given t_k is updated according to Eq.(2.8), it is easy to show that $t_k \geq \dfrac{(k+1)}{2}$. Substituting this inequality into Eq.(3.1) yields

$$
v_k \leq \frac{2L_k\left(||\mathbf{U}_0 - \hat{\mathbf{U}}||_F^2 + ||\mathbf{V}_0 - \hat{\mathbf{V}}||_F^2\right)}{(k+1)^2}
$$

By the Remark 3.2 in [6] and the inequality (2.13), we also know that an upper bound of L_k is $\tilde{L}$. Hence,

$$f(\mathbf{U}_k, \mathbf{V}_k) - f(\hat{\mathbf{U}}, \hat{\mathbf{V}}) \leq \frac{2\tilde{L}\left(||\mathbf{U}_0 - \hat{\mathbf{U}}||_F^2 + ||\mathbf{V}_0 - \hat{\mathbf{V}}||_F^2\right)}{(k+1)^2}$$

In our algorithm, we set $L_k = \tilde{L}, \forall k$. □

Remark 1. *The loss function, $\ell(\mathbf{U}, \mathbf{V})$, of an exponential distribution has Lipschitz continuous gradient within the range $\{||\mathbf{U}||_{1,2} \leq \delta_1, ||\mathbf{V}^\top||_{1,2} \leq \delta_2\}$ where δ_1, δ_2 are constant values in terms of λ_1, λ_2, respectively to guarantee the non-trivial step size $\frac{\lambda}{L}$. Otherwise, it may lead to a sub-optimal solution.*

3.2 Asymptotic Analysis

To facilitate the asymptotic analysis, we re-write the notation as follows: let

$$\boldsymbol{\beta} = [\text{vect}(\mathbf{U})^\top, \text{vect}(\mathbf{V})^\top]^\top, \quad \mathbf{H}^{(i)} = [\mathbf{h}_{\tau+1}^{(i)}, \cdots, \mathbf{h}_n^{(i)}]$$

and

$$\mathbf{h}_t^{(i)} = [\text{vect}(\mathbf{X}_{i;t})^\top, \text{vect}(\mathbf{X}_{i;t})^\top]^\top$$

where one block $\mathbf{X}_{i;t}$ corresponds to $\mathbf{U}$ and the other to $\mathbf{V}$. Then, correspondingly, we have $\eta_t^{(i)} = (\mathbf{h}_t^{(i)})^\top \boldsymbol{\beta}$, and $f(\mathbf{U}, \mathbf{V})$ can be re-written as $f(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + R(\boldsymbol{\beta}; \lambda_1, \lambda_2)$.

Solve Eq.(2.6) yields a solution to the penalized estimating equations:

$$\sum_i (\mathbf{D}^{(i)})^\top (\boldsymbol{\Sigma}^{(i)})^{-1} \mathbf{s}^{(i)} + \lambda \frac{\partial R(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0 \tag{3.2}$$

assuming $\lambda_1 = \lambda_2 = \lambda$ for notational convenience which will not change the property. Given our model definition (2.5), $\mathbf{D}^{(i)} = \mathbf{A}^{(i)}(\mathbf{H}^{(i)})^\top$. The first term in (3.2) is the estimating functions in GEE [35] whereas the second term corresponds to the regularizers. The asymptotic property of Eq.(2.6) can be naturally derived from the results in [35] which have proved that the estimating equations $L(\boldsymbol{\beta}) = \sum_i (\mathbf{D}^{(i)})^\top (\boldsymbol{\Sigma}^{(i)})^{-1}\mathbf{s}^{(i)}$ of GEE gives a consistent estimator of $\boldsymbol{\beta}$. We extend the same argument to our formulation Eq.(2.6) in Theorem 3 under the following regularity conditions: $\mathbf{H}^{(i)}$ is bounded, and $\lim_{m\to\infty}(\sum_i \mathbf{H}^{(i)})/m = \mathbf{H}^{(0)}$, and $(\mathbf{H}^{(i)})^\top\mathbf{H}^{(i)}$ are not singular, and the following limit is also not singular

$$\lim_{m\to\infty}(\sum_i (\mathbf{H}^{(i)})^\top \mathbf{H}^{(i)})/m;$$

Moreover, $L(\boldsymbol{\beta})$ is twice continuously differentiable with respect to $\boldsymbol{\beta}$, and $\partial L/\partial\boldsymbol{\beta}$ is positive definite.

Theorem 3. *Assume that: (1) $\hat{\boldsymbol{\alpha}}$ is a consistent estimator given $\boldsymbol{\beta}$; (2) $\hat{\phi}$ is a consistent estimator given $\boldsymbol{\beta}$; and (3) the tuning parameter $\lambda_m = o(\sqrt{m})$. Under the regularity conditions listed above, optimizing Eq.(2.6) yields an asymptotically consistent and normally distributed estimator $\hat{\boldsymbol{\beta}}$, that is:*

$$\sqrt{m}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \to_d N(0, \boldsymbol{\Sigma}) \quad as \quad m \to \infty$$

where $\boldsymbol{\beta}^*$ is the true model coefficients in a model of $E(y_t^{(i)}) = g^{-1}((\mathbf{h}_t^{(i)})^\top\boldsymbol{\beta})$ and Σ is a positive definite variance-covariance matrix (see [35] for details of Σ).

Proof. Multiplying $1/m$ to both sides of Eq.(3.2) yields

$$\frac{1}{m}\sum_i (\mathbf{D}^{(i)})^\top (\boldsymbol{\Sigma}^{(i)})^{-1}\mathbf{s}^{(i)} + \frac{\lambda_m}{m}\frac{\partial R(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0. \tag{3.3}$$

It is known that solving $\frac{1}{m}\sum_i (\mathbf{D}^{(i)})^\top (\boldsymbol{\Sigma}^{(i)})^{-1}\mathbf{s}^{(i)} = 0$ yields an estimate of $\hat{\boldsymbol{\beta}}$ that is asymptotically consistent with $\boldsymbol{\beta}^*$:

$$\sqrt{m}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \to_d N(0, \boldsymbol{\Sigma}) \;\text{ as }\; m \to \infty \;\; [35].$$

Since our regularizer R (based on the $\ell_{1,2}$ matrix norm) is Lipschitz continuous, its partial derivative $\partial R(\boldsymbol{\beta})/\partial \boldsymbol{\beta}$ is bounded. The second term of Eq.(3.3) vanishes when $m \to \infty$, and thus the conclusion holds. □

Recall how $\hat{\boldsymbol{\alpha}}$ and $\hat{\phi}$ are estimated in the proposed method. Those estimates from the Pearson residuals are consistent. Thus, the estimate $\hat{\boldsymbol{\beta}}$ in the proposed method is asymptotically consistent and normally distributed according to Theorem 3.

3.3 Exemplar Exponential Families with Lipschitz Condition

The purposed algorithm is suitable to optimize any loss function that has Lipschitz continuous gradient. In this section, we discuss that three exemplar exponential families: Gaussian, Bernoulli, and Poisson, satisfy the Lipschitz condition. We specify how to compute the gradient of the loss function for these distributions. The gradients will instantiate (and replace) Eq.(2.12) used in our algorithm.

Gaussian Distribution

If the outcome follows a Gaussian distribution, then the outcome y is linearly regressive in terms of the covariates in the observations. The mean and the conditional covariance of y with a working correlation structure $\mathbf{R}(\boldsymbol{\alpha})$ are calculated as:

$$E(y_t^{(i)}) = \mu_t^{(i)} = tr\left(\mathbf{X}_{(i;t)}^{\top}\mathbf{W}\right),$$
$$cov(\mathbf{y}^{(i)}) = \boldsymbol{\Sigma}^{(i)} = \mathbf{R}(\boldsymbol{\alpha}),$$

so the gradient $\nabla_{\mathbf{U}}\ell_k$ in Eq.(2.12) at the k-th iteration can be computed as

$$\nabla_{\mathbf{U}}\ell_k = \text{reshape}\left(\sum_{i=1}^{m}\left(\mathbf{D}^{(i)}\right)^{\top}\left(\mathbf{R}(\boldsymbol{\alpha})\right)^{-1}\mathbf{s}_k^{(i)}\right),$$

where

$$\mathbf{D}^{(i)} = \frac{\partial\boldsymbol{\mu}^{(i)}}{\partial\text{vect}\left(\tilde{\mathbf{U}}_k\right)} = \left[\text{vect}\left(\mathbf{X}_{(i;1)}\right),\ldots,\text{vect}\left(\mathbf{X}_{(i;n)}\right)\right]^{\top},$$

and

$$\mathbf{s}_k^{(i)} = \mathbf{y}^{(i)} - \left(\mathbf{D}^{(i)}\right)^{\top}\text{vect}(\tilde{\mathbf{U}}_k).$$

The gradient $\nabla_{\mathbf{V}}\ell_k$ can be similarly computed. Hence, the gradient is linear in terms of $\boldsymbol{\beta}$, and thus Lipschitz continuous.

Bernoulli Distribution

If the generalized variables μ follow a Bernoulli distribution and the outcomes are binary variables. The relationship between the outcome and covariates can be learned by a logistic regression which is a special case of the GLM with the Bernoulli as-

sumption. Hence, the mean and the conditional covariance of y with the working correlation structure $\mathbf{R}(\boldsymbol{\alpha})$ are formulated as

$$E(y_t^{(i)}) = \mu_t^{(i)} = \frac{\exp(\eta_t^{(i)})}{1+\exp(\eta_t^{(i)})} \tag{3.4}$$
$$cov(\mathbf{y}^{(i)}) = \mathbf{\Sigma}^{(i)} = \frac{\left(\mathbf{A}^{(i)}\right)^{1/2}\mathbf{R}(\boldsymbol{\alpha})\left(\mathbf{A}^{(i)}\right)^{1/2}}{\phi}$$

where $\mathbf{A}^{(i)} = \text{diag}\left(\langle \boldsymbol{\mu}^{(i)}, 1-\boldsymbol{\mu}^{(i)}\rangle\right) = \text{diag}\left(\frac{\exp(\eta_t^{(i)})}{\left(1+\exp(\eta_t^{(i)})\right)^2}\right)$ and $\eta_t^{(i)} = tr(\mathbf{X}_{(i;t)}^\top \mathbf{W})$.

The gradient $\nabla_{\mathbf{U}}\ell_k$ in Eq.(2.12) can be written as:

$$\text{reshape}\left(\left(\mathbf{D}^{(i)}\right)^\top (\mathbf{A}^{(i)})^{-1/2}\mathbf{R}(\boldsymbol{\alpha})^{-1}(\mathbf{A}^{(i)})^{-1/2}\mathbf{s}_k^{(i)}\right)$$

where $\mathbf{D}^{(i)} = \frac{\partial \boldsymbol{\mu}^{(i)}}{\partial \boldsymbol{\eta}^{(i)}} \times \frac{\partial \boldsymbol{\eta}^{(i)}}{\partial \text{vect}\left(\tilde{\mathbf{U}}_k\right)} = \mathbf{A}^{(i)}\left[\text{vect}\left(\mathbf{X}_{(i;1)}\right), \ldots, \text{vect}\left(\mathbf{X}_{(i;n)}\right)\right]^\top$, and $\mathbf{s}_k^{(i)} = \mathbf{y}^{(i)} - \boldsymbol{\mu}^{(i)}(\tilde{\mathbf{U}}_k)$. The gradient $\nabla_{\mathbf{V}}\ell_k$ can be similarly computed.

Poisson Distribution

If the generalized variables μ follow a Poisson distribution and the outcomes contain count values. The relationship of the outcome and covariates is learned by a Poisson regression. The mean and the conditional covariance of y with the working correlation structure $\mathbf{R}(\boldsymbol{\alpha})$ are formulated as

$$E(y_t^{(i)}) = \mu_t^{(i)} = \exp(\eta_t^{(i)})$$
$$cov(\mathbf{y}^{(i)}) = \mathbf{\Sigma}^{(i)} = \frac{\left(\mathbf{A}^{(i)}\right)^{1/2}\mathbf{R}(\boldsymbol{\alpha})\left(\mathbf{A}^{(i)}\right)^{1/2}}{\phi}$$

where $\mathbf{A}^{(i)} = \text{diag}\left((\boldsymbol{\mu}^{(i)})'\right) = \text{diag}\left(\exp(\eta_t^{(i)})\right)$. The gradient $\nabla_{\mathbf{U}}\ell_k$ can be computed using the general formula Eq.(2.12). The loss function of Poisson regression does not have globally Lipschitz continuous gradient. But the regularized loss function is equivalent to requiring the constraints, $||\mathbf{U}||_{1,2} \leq \delta_1$ and $||\mathbf{V}^\top||_{1,2} \leq \delta_2$ [50] for appropriate values of δ_1 and δ_2 that are determined according to λ_1 and λ_2. The loss function of Poisson regression does have Lipschitz continuous gradient within the confined region.

4 Empirical Evaluation

We validated the proposed approach by comparing it to several most relevant and recent methods. Three GLM-based [49] methods: GEE [35], GLMM [32, 41], and RE-EM tree[1] [58] were compared. The recent graphical Granger modeling[2] [37] and a support vector machine based method called CSVM were also used. RE-EM tree and graphical Granger modeling could only be applied to regression problems (linearly regressive data from Gaussian distributions), and CSVM was only suitable to classification tasks (logistically regressive data from Bernoulli distributions). We named our approach by LGL (longitudinal group lasso). The normalized mean squared error (nMSE), which is the MSE divided by the variance of y [84, 18], was used to measure regression performance. The area under the ROC curve (AUC) [10] was used to measure classification performance.

[1] An R package is available in the Comprehensive R Archive Network (CRAN)

[2] downloaded from the author's website http://www-bcf.usc.edu/∼liu32/code.html

4.1 Synthetic Data

We generated a data matrix $\mathbf{X} \in \mathbb{R}^{d \times Tm}$ from the normal distribution $N(0, 16)$, where $d = 200$, $T = 30$, and $m = 400$. All training examples $\mathbf{X}_{(i;t)}(i = 1, \cdots, m,$ $\forall t = \tau + 1, \cdots, T)$ and $\tau = 4$ were formed from the matrix $\mathbf{X}$. Then, $\mathbf{U}$ and $\mathbf{V}$ were generated from the normal distribution $N(0, 49)$. We set the rows corresponding to features from 1 to 150 in $\mathbf{U}$ to zero and the columns 2 and 5 of $\mathbf{V}$ to zero, and computed $\mathbf{W} = \mathbf{U} + \mathbf{V}$. The residuals $\mathbf{s}^{(i)}$ of every subject were generated from a multivariate normal distribution of different variances, $N(0, 1^2), N(0, 2^2), N(0, 3^2)$. The covariance matrix of the residual followed different working correlation structures $\mathbf{R}(\alpha)$ with the parameter $\alpha = 0.64$. We generated 9 sets of regression residuals by choosing different combinations of the variances and the working correlation structures. Finally, the outcome variables $\mathbf{y}^{(i)}$ were computed as

$$\mathbf{y}^{(i)} = \left[\text{vect}\left(\mathbf{X}_{(i;\tau+1)}\right), \ldots, \text{vect}\left(\mathbf{X}_{(i;n)}\right)\right]^\top \text{vect}(\mathbf{U} + \mathbf{V}) + \mathbf{s}^{(i)}.$$

The above procedure produced regression data. Using the same data $\mathbf{X}$, the outcome $y_t^{(i)}$ of a classification problem was generated from the Bernoulli Distribution with $\text{B}(1, \mu_t^{(i)})$ where we used Eq.(3.4) with the regression $\mathbf{y}^{(i)}$ to obtain $\boldsymbol{\mu}^{(i)}$. We hence obtained totally 18 synthesized data with 9 datasets for each distribution. We used the 25 early records of each subject to compose the training data and the rest 5 records to form test data.

Table 4.1 shows the results where we can see that LGL outperformed all other methods on all the simulated datasets. The proposed method with correct correlation assumptions always performed the best. The graphical Granger modeling performed reasonably well but lacked of consideration of temporal correlation in the consecutive

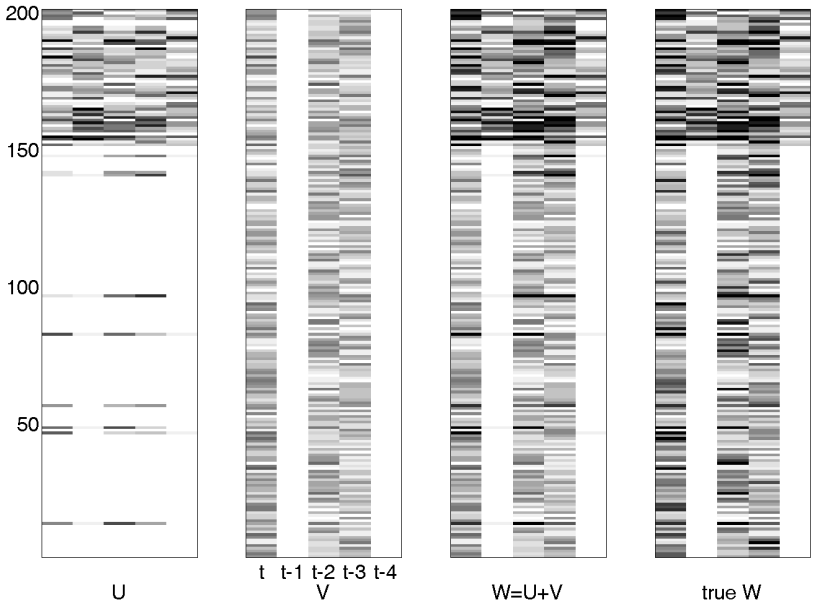


Figure 4.1: The model constructed by our approach LGL on a synthetic dataset.

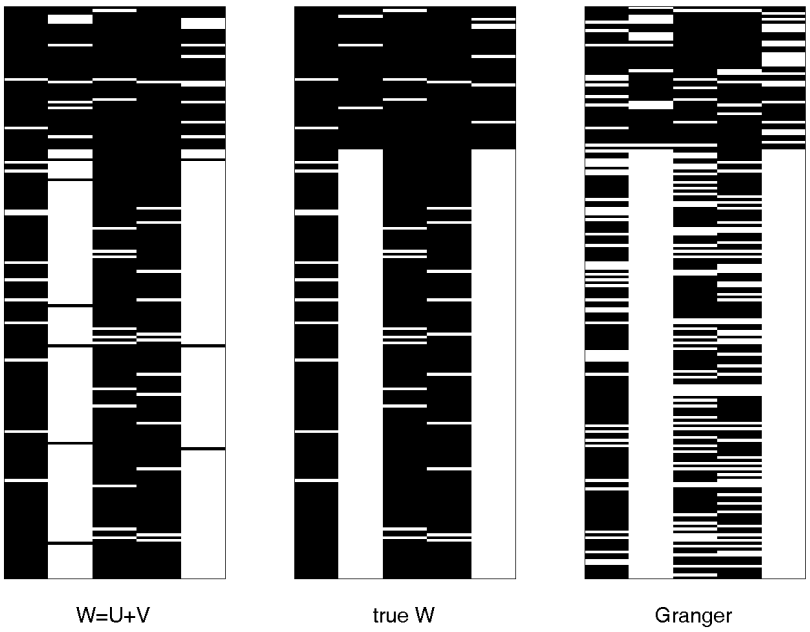


Figure 4.2: Comparison between the constructed models by LGL and Granger.

records. When the simulated noise increased, the performance of all methods had dropped as expected. We further demonstrate the selected features and temporal contingency. Figure 4.1 shows the constructed $\mathbf{U}, \mathbf{V}$, and $\mathbf{W}$ by the LGL on the regression data with the AR(1) covariance structure and $N(0, 3^2)$ residual where darker colors indicate larger values (and white means 0). Most of the features from 150 to 200 were selected in $\mathbf{U}$ and the correct columns (i.e., $1, 3, 4$) were selected in $\mathbf{V}$. We compared our approach with the Granger model that also learned $\mathbf{W}$ in Figure 4.2. Obviously, the Granger model excluded too many variables in the model. These results demonstrate the capability of LGL in terms of simultaneously capturing the important features and lagged effects.

4.2 Real-world Data of National Longitudinal Survey of Youth (NLSY)

We tested our approach on two real-world datasets: the college alcohol use dataset; and the NLSY dataset[3]. All comparison methods were used except GLMM due to its prohibitive computational costs. The college alcohol use dataset consisted of data from 504 college students on 52 variables in a period of continuous 30 days. The 52 variables measured each subject on daily stress, moods, emotion and substance use behavior. One of the variables measured the number of night-time drinks, which was our outcome variable, forming a regression problem. We also predicted the binge drinking behavior which is defined as having 5 or more night-time drinks, which formed a classification problem. The NLSY dataset consisted of 11 yearly data for 3,376 subjects on 27 variables. The outcome variable measured the number of days

[3]http://www.bls.gov/nls/nlsy97.htm

that a subject had binge drinking in past 30 days, forming a regression problem. The other 26 variables measured features, such as smoking, drug use, family support and education.

For the college alcohol use data, we experimented with using the last $t = 3, 5, 8, 10$ days of records as test data, and the rest for training. We found $\tau = 3$ was feasible. Larger τ would not change the results because the extra time points would be excluded by our model. However, it practically would cut down the sample size of each subject. The parameters λ_1 and λ_2 in our approach and any tuning parameters in other methods were tuned in a three-fold cross validation within the training data. Table 4.2 shows the results where our approach LGL outperformed other methods in most settings. Among the four different correlation assumptions, LGL with AR(1) obtained the best performance on three of the four settings. The results also confirmed that modeling the correlation among repeated observations improved prediction performance [35]. We also observed that for instance, 16 out of 51 variables were selected when we used the last 5 days to test binge drinking prediction. Features related to exited mood, under stress and interacting with friends during night time were the risk factors for binge drinking. The past 3 days were all included in the model, showing there was "lagged" effects in alcohol use. The effect of past days was reduced with prolonged time lag.

For the NLSY dataset, we experimented respectively with using the last one, two and three years from each subject for test and the rest in training. We also considered $\tau = 3$, which means we used 3 year lagged data to predict the current year's behavior. All tuning parameters were tuned using a within-training two-fold cross validation. The results are reported in Table 4.3. For any assumption of the working correlation structure, LGL had comparative performance with RE-EM tree and consistently out-

Table 4.1: Comparison of different algorithms on synthetic data: (top) regression; (bottom) classification.

			LGL				GEE						
	Structures	e	AR(1)	exchangeable	Tri-diag	ind	AR(1)	exchangeable	Tri-diag	ind	GLMM	RE-EM tree	Granger
Regression	AR(1)	$N(0,1^2)$	**0.0018**	0.0020	0.0019	0.0020	0.6613	0.6615	0.6614	0.6617	0.6657	0.9873	0.0664
		$N(0,2^2)$	**0.0025**	0.0026	0.0028	0.0039	0.7223	0.7236	0.7224	0.7242	0.7323	0.9998	0.0667
		$N(0,3^2)$	**0.0032**	0.0034	0.0036	0.0038	0.7191	0.7185	0.7182	0.7192	0.7179	0.9924	0.0676
	exchangeable	$N(0,1^2)$	0.0018	0.0016	**0.0015**	0.0022	0.6872	0.6875	0.6872	0.6873	0.6914	0.9977	0.0656
		$N(0,2^2)$	0.0024	**0.0023**	0.0024	0.0025	0.6927	0.6930	0.6927	0.6930	0.6931	0.9982	0.0691
		$N(0,3^2)$	0.0027	**0.0026**	0.0028	0.0032	0.7204	0.7204	0.7204	0.7205	0.7204	0.9797	0.0635
	Tri-diag	$N(0,1^2)$	0.0021	0.0021	**0.0021**	0.0022	0.7514	0.7514	0.7514	0.7514	0.7515	0.9925	0.0665
		$N(0,2^2)$	0.0018	0.0023	**0.0013**	0.0026	0.6790	0.6792	0.6791	0.6793	0.6840	0.9991	0.0680
		$N(0,3^2)$	0.0033	0.0035	**0.0031**	0.0041	0.7226	0.7235	0.7226	0.7226	0.7222	0.9998	0.0660

			LGL				GEE				
	Structures	e	AR(1)	exchangeable	Tri-diag	ind	AR(1)	exchangeable	Tri-diag	ind	CSVM
Classification	AR(1)	$N(0,1^2)$	**96.490%**	96.485%	96.485%	96.417%	77.691%	77.700%	77.699%	77.715%	76.644%
		$N(0,2^2)$	**96.442%**	96.431%	96.432%	96.653%	74.682%	74.727%	74.682%	74.731%	75.249%
		$N(0,3^2)$	**95.921%**	95.917%	95.917%	95.805%	77.704%	77.746%	77.708%	77.754%	77.547%
	exchangeable	$N(0,1^2)$	95.913%	**95.937%**	95.912%	95.883%	76.115%	75.812%	76.114%	75.923%	75.232%
		$N(0,2^2)$	95.139%	**95.161%**	95.147%	95.150%	70.290%	70.231%	70.275%	70.206%	71.687%
		$N(0,3^2)$	94.127%	94.091%	**94.135%**	93.470%	73.839%	73.782%	73.831%	73.776%	73.894%
	Tri-diag	$N(0,1^2)$	95.976%	95.941%	**95.978%**	95.889%	77.628%	77.634%	77.625%	77.617%	76.778%
		$N(0,2^2)$	95.231%	95.231%	**95.245%**	94.395%	72.132%	72.060%	72.126%	72.054%	71.615%
		$N(0,3^2)$	95.092%	95.087%	**95.094%**	94.231%	77.755%	77.533%	77.748%	77.637%	77.572%

Table 4.2: Comparison of different algorithms on the college alcohol use dataset: (top) predicting the number of night-time drinks (regression); (bottom) predicting the occurrence of binge drinking (classification).

		LGL				GEE					
Regression	# observations	AR(1)	exchangeable	tri-diag	ind	AR	exchangeable	tri-diag	ind	RE-EM tree	Granger
	3	**0.933513**	0.933863	0.935120	0.961841	1.064792	1.073358	1.063948	1.065760	1.115627	1.369948
	5	0.951999	0.954740	**0.951953**	0.976299	1.051219	1.067303	1.049305	1.072745	1.005753	1.420547
	8	**0.759935**	0.760450	0.760136	0.762205	0.787731	0.793329	0.787497	0.794089	0.759968	0.909706
	10	**0.769303**	0.769492	0.769428	0.774937	0.812622	0.818834	0.812011	0.806301	0.774797	0.940940

		LGL				GEE				
Classification	# observations	AR(1)	exchangeable	tri-diag	ind	AR	exchangeable	tri-diag	ind	CSVM
	3	79.737%	75.677%	79.772%	78.579%	78.401%	74.145%	78.650%	77.831%	**80.698%**
	5	**83.290%**	77.237%	83.070%	82.323%	80.371%	78.363%	80.646%	80.438%	83.187%
	8	**88.570%**	87.331%	87.936%	87.787%	85.999%	86.330%	85.714%	86.014%	88.017%
	10	**89.484%**	87.574%	88.853%	88.578%	85.979%	86.622%	85.721%	85.783%	89.041%

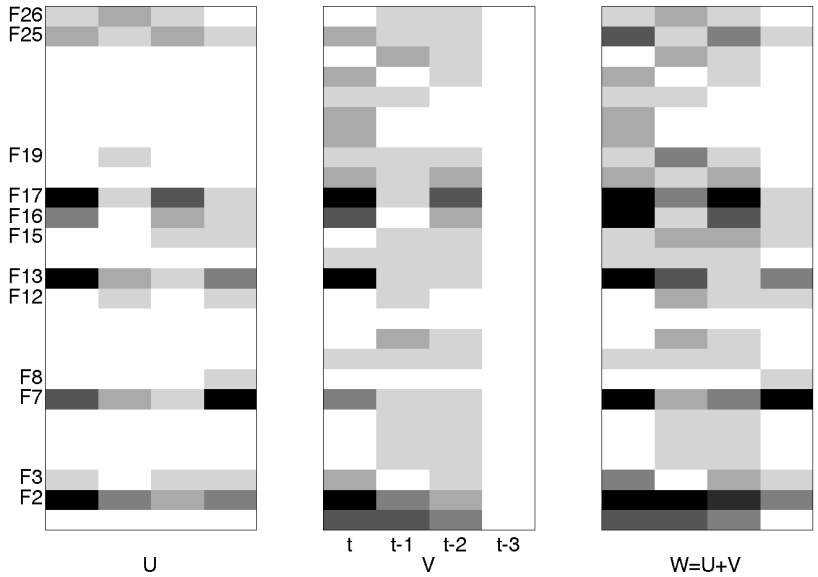


Figure 4.3: The model constructed by our approach on the NLSY dataset.

performed GEE in all of the three experiments. LGL with tri-diagonal correlation performed the best on this dataset. The results here again show that taking care of the correlation among repeated observations improves the performance (given we see that LGL with the independent correlation assumption had the worst performance among all LGL variants).

The gray map of **U**, **V** and **W** constructed by LGL is shown in Figure 4.3 to illustrate an example for the tri-diagonal working correlation assumption. Out of the 26 features, 12 were selected by LGL and we list them below.

F2: # days of smoking a cigarette in the past 30 days

F3: Received a training certificate or vocational license

F7: The grade began during the academic year

F8: # months that respondent did not attend school during the academic year

F12: The college degree working toward or attained

F13: The highest grade completed as of the survey year

F15: The highest grade attended as of the survey day

F16: The highest grade completed as of the survey day

F17: # days of using marijuana in the past 30 days

F19: # times of using some drug or other substance right before school or during school or work hours

F25: As the victim of a violent crime in the survey year

F26: Divorced parents.

Table 4.3: Comparison of different algorithms on the NLSY dataset in terms of test nMSE values.

	LGL				GEE					
# obs	AR(1)	exchangeable	tri-diag	ind	AR	exchangeable	tri-diag	ind	RE-EM tree	Granger
1	0.906552	0.908932	0.904760	0.909446	0.911543	0.918691	0.911885	0.914043	**0.904260**	1.370135
2	0.888608	0.891761	**0.887294**	0.891051	0.898132	0.904225	0.897920	0.898320	0.888822	1.363714
3	0.885448	0.885814	**0.883617**	0.887579	0.892963	0.895863	0.892633	0.890937	0.883958	1.360430

This list shows that a subject's smoking, drug use, education background and family support influenced his or her drinking behavior. Figure 4.3 demonstrates that the data in the third prior year might be obsolete to predict this year's behavior as LGL only selected the past two years for use in the model as seen in the plot of $\mathbf{V}$.

4.3 Real-world Data of EEG Dataset

The EEG recording provides powerful methodology for studying neural dynamics of human cognition. It enables the evaluation of real-time changes in neural activity at distinct information processing stages to behavioral performance [31]. In this experiment, we analyzed schizophrenia (SZ) using the EEG data collected while participants performed a visual Sternberg working memory task [55]. During the trials of the Sternberg working memory task, the participants had to recall whether or not the examined letter used to appear in the early stage. the Sternberg trial responses,

correct vs. incorrect, was recorded.

In this study, an EEG record consisted of 60 features extracted from five frequency bands (δ: 0.5 - 4 Hz, θ: 4 - 8 Hz, α: 8 - 12 Hz, β: 14 - 28 Hz, and γ: 30 - 58 Hz), three brain regions (Fz, Cz and Oz), and four memory stages (baseline, encode, retain and retrieve). Thus, the frequency bands, brain regions, and memory stages described the brain activities in 15 features and 4 different time stages so as to form a data matrix $\mathbf{X}^{(i)} \in \mathbb{R}^{15\times 4}$ for each trial. A binary label associated with each record indicated whether the individual answered correctly (0) or incorrectly (+1) in the trial.

Our study samples consisted of 37 individuals meeting the diagnostic criteria for SZ and 6 healthy normal (HN) adults enrolled in clinical trial NCT00923078[4]. SZ patients went through three sessions of the Sternberg trials, and HN members were only included in the first session. There were 90 trials in each session for each individual. However, very few patients participated all sessions and many trial records had missing values or significant level of noise or outliers, for which we had to clean the data carefully. After data cleaning, there were 1131 trials for 14 SZ in session 1,761 trials for 9 SZ in session 2, and 1191 trials for 14 SZ in session 3. Each patient had 74 to 94 trials, and 83 on average. The rate of incorrect responses for the SZ patients was 27.2%. There were 519 trials for 6 HN participants. Each participant had 82 to 90 trials, and 87 on average. The rate of incorrect responses for HN participants was 14.7%. Note that the current study data contained a limited sample of subjects from the parent study. Additional efforts will be needed to clean and process the full dataset and repeat the analyses reported here.

For each of the SZ and HN datasets, 1/3 of the records were randomly chosen from every subject to form the test data and the rest of the records were used in

[4]https://clinicaltrials.gov

training. The hyper-parameters λ_1 and λ_2 in our approach and GEE (one parameter) were tuned in a two-fold cross validation within the training data. In other words, the training records were further split in half: one used to build a classifier with a chosen parameter value from a range of 1 to 10 with a stepsize 0.1; and the other used to test the resultant classifier. We chose the parameter values that gave the best two-fold cross validation performance, which were $\lambda_1 = 5.9$ and $\lambda_2 = 10$ for SZ and $\lambda_1 = 2$ and $\lambda_2 = 3.1$ for HN.

Table 4.4 provided the AUC comparison results (shown in percentages) between the two methods and for different datasets and sample correlation assumptions. The results in Table 4.4 showed that our approach outperformed the traditional GEE in almost all comparison scenarios in terms of classification accuracy. Most importantly, our approach was able to select along two dimensions: among the features and among the memory information processing stages. Traditional GEE did not have any shrinkage effect to select features. The advanced version of GEE used in our experiments implemented a ℓ_1 regularizer, so it could select among all 60 features. Because it did not use the spatial-temporal structure of the 60 features, it was unable to model along the different dimensions (locations versus temporal stages).

Table 4.4: Comparison of AUC values (in percentage) between our approach and the GEE method on both healthy normal and schizophrenia data and for all different assumptions of correlation structures. (ind - independent sample-correlation structure.)

	GEE				Our Approach			
	AR(1)	Exchangeable	Tri-diagonal	ind	AR(1)	exchangeable	Tri-diagonal	ind
HN	54.1	52.2	55.5	57.3	55.1	54.9	55.0	68.0
SZ	60.3	55.5	43.6	65.0	62.6	60.0	48.2	66.3

We noticed that both GEE and our approach performed the best when using in-

dependent sample-correlation assumption, which was naturally against our intuition because there were multiple trials from a single individual and these trials were expected to correlate. The other assumptions were slightly worse than the independent correlation assumption. However, we also noticed that the trials were not labeled in sequence in our data so the algorithms would not be able to model and distinguish the correlations between consecutive trials from those of far-apart trials. (The trials that an individual performed in a short continuous time-frame may correlate more strongly than trials far apart.)

We included two figures to demonstrate the selected features and stages in the classifiers constructed by our approach. The selected features for SZ patients were shown in Figure 4.4. The selected features for HN participants were shown in Figure 4.5. An obvious observation is that the two populations selected quite different features but the most important information processing stages were the same.

Based on our models, the two groups showed remarkably different patterns, with EEG activity in higher frequency bands during the encoding stage associated with incorrect trial responses in SZ (Figure 4.4). However, these features were positive predictors of trial accuracy in healthy participants (Figure 4.5), for whom engagement of low frequency activity was associated with incorrect responses. It appears that the SZ patients used more brain areas in the memory tasks than the HN participants. Frontal γ was previously identified as important for both SZ and HN subjects, but was not selected for HN participants in our new model, which may warrant further investigation. On the other hand, among the selected three stages of both groups, the features during the retention stage tended to receive the largest weights in magnitude on average. All these results will require careful examination in new studies to confirm the validity and replicate on independent samples.

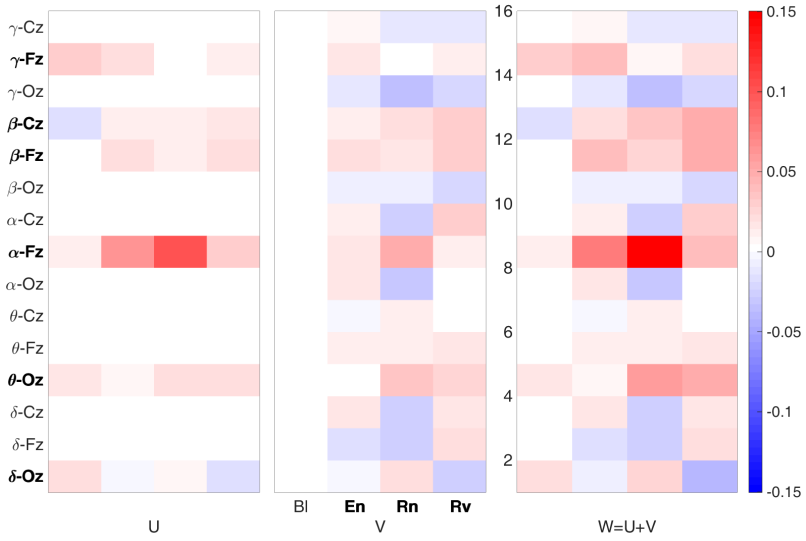


Figure 4.4: Columns and rows selected by the classifier for separating correct versus incorrect Sternberg trials of SZ patients. Red (blue) color indicates that the corresponding features were positive (negative) predictors of the incorrect response. Features with white color were not used in the classifier.

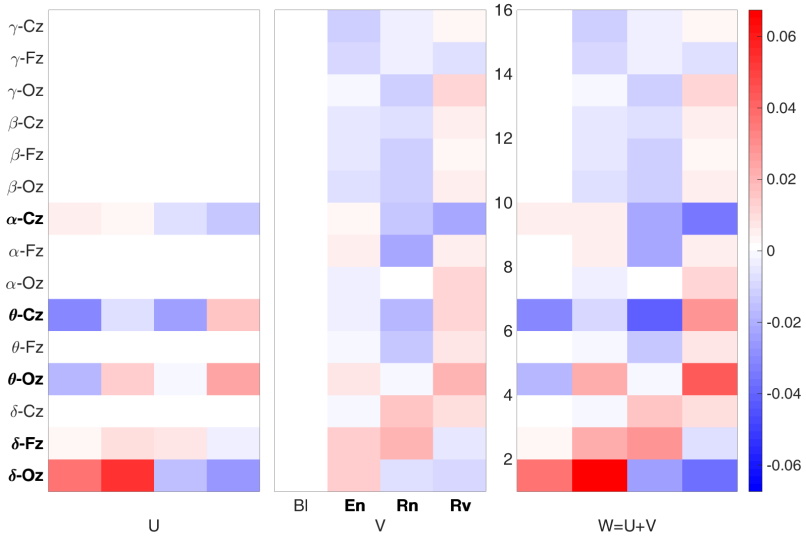


Figure 4.5: Columns and rows selected by the classifier for separating correct versus incorrect Sternberg trials of HN participants. Red (blue) color indicates that the corresponding features were positive (negative) predictors of the incorrect response. Features with white color were not used in the classifier.

5 Discussion

We have proposed a new learning formulation for longitudinal analytics. Unlike existing methods, the proposed approach can simultaneously determine the temporal contingency and the influential features in predicting an outcome over time. The model parameter matrix is computed by the summation of two component matrices: one matrix reflects the selection among covariates; and the other characterizes the dependency along the temporal line. Moreover, our approach simultaneously models the sample correlations in the longitudinal data while constructing a predictive model. The related optimization problem can be efficiently solved by a new accelerated gradient descent algorithm. Convergence analysis shows that the algorithm can find the global optimal solution for the model with a quadratic convergence rate. An asymptotic analysis shows that the solution of our formulation is a consistent estimate of the model parameters. Hence, the proposed approach solves an underdeveloped problem - jointly learning the relevant features and determining how current outcome relies on past observations. Empirical studies on both synthetic and real-world problems demonstrate the superior performance of the proposed approach over the state of the art.

Chapter 3

Jointly Learning Multi-dimensional Features and Temporal Contingency in Longitudinal Data

1 Introduction

Temporal data such as time series data and longitudinal data are pervasive across almost all human endeavors, including finance [58, 4], science [65, 16, 7], climate [37, 4], and genetics [75]. As such, it is hardly surprising that temporal data mining has attracted significant attention and research effort. Meanwhile, with the advances in data acquisition technologies, ultra-high dimensional data with complex structure are collected in many disciplines and industrial societies. Such datasets contain tensor data entries where each observed example is a high dimensional tensor. For example, in a neuroscience study [22, 73, 22], repeated measurement along time of functional magnetic resonance imaging (fMRI) can create tensors in very high dimen-

sions because the fMRI image itself of each measurement is a 3 dimensional volume. Moreover, follow-up diagnoses require an analysis on both baseline fMRI data and data collected at follow-up time points.

Typically, longitudinal data are analyzed by extending generalized linear models (GLM) with different assumptions, such as marginal models, random effects models, and transition models [13]. For marginal modeling, generalized estimating equations (GEE) [35] and Quadratic Inference Function (QIF) [54, 77] are the most widely used methods which estimate a predictive model to predict the current outcome together with longitudinal correlations among different outcomes observed temporally. The resultant predictive models are generally more accurate than those of classic regression analysis that assumes independently and identically distributed (*i.i.d.*) observations [35]. However, when the working correlation is misspecified, the correlation structure presumed by GEE no longer results in the optimal estimation of coefficients while QIF estimates a linear combination of correlation structures so that the estimator always exists, and, even if the correlation is misspecified. Research on feature selection in longitudinal data leads to a new family of methods based on the penalized GEE (PGEE) [17] and penalized QIF (PQIF) [5].

None of those extensions of GLM aims to detect causal relationships from temporal changes of covariates to the outcomes of the current effect. In many studies, it is however necessary and insightful to model simultaneously the correlation among outcome records and the lagged causal effects of covariates, which is so-called Granger causality [19]. For example, some evidences of brain diseases may appear in the fMRI of an early diagnosis before clear symptoms are identified [63]. This lagged effect is not used by temporal marginal modeling to make predictions. Recent graphical Granger models [4, 37] learn the coefficients on both the current covariates and the

covariates in the past time with LASSO type of regularizers and evaluate if coefficients are non-zero for Granger causality. However, they ignore the temporal correlations.

On the other hand, researchers have begun to leverage tensor techniques, such as low-rank tensors [25, 71], Schatten 1-norm tensors [15, 56, 69], latent approach of low-rank tensors [68], to develop new techniques that build a regression or classification model as a function of the tensor, preserving the multilinear data structure in the model. Support vector machines have been extended to multidimensional data and produced better results in document classification [11]. A matrix variate logistic regression model has been developed recently and tested in the analysis of EEG data [29]. Another method directly extended logistic regression to take tensor data as inputs [67]. A nice tensor regression model has been developed by decomposing the coefficient tensor into a summation of several rank-one tensors [87] and its asymptotic properties have also been studied. Another method, *MulSLR*, uses a similar idea that builds a logistic regression model with a rank-one coefficient tensor [74] to better recognize the latent structure in big data. However, these methods usually formulate non-convex optimization problems that are hard to solve. Moreover, the lack of the considerations of influences along time made them perform poorly in applying to temporal data such as fMRI data.

Existing methods either assume *i.i.d.* samples in tensor regression but ignore temporal correlations in longitudinal data or assume correlated samples but are not able to model *temporal* causal effects and *complex* feature structures. Therefore, we propose a new learning formulation that constructs tensor-based predictive models as functions of covariates not only from the current observation but also from multiple previous consecutive observations, and simultaneously determine the temporal contingency and the most influential features along each dimension of the tensor data.

The proposed method makes predictions based on lagged data from current and previous time points. It decompose the K-way parameter tensor into a summation of K sparse K-way tensors as shown in Figure 1.1. These tensors each present sparsity along one direction of the parameter K-way tensor and impose different block-wise *least absolute shrinkage and selection operators* (LASSO) to the components. Hence, our approach formulates a convex optimization problem. The proposed method also learns simultaneously correlation information from the data via quadratic inference function. The correlations among the outcomes themselves imply the changing trend of the outcomes in the proximal time points within each subject. We propose a fast iterative shrinkage-thresholding (FISTA) [6] algorithm to efficiently solve the optimization problem. We validate the effectiveness of the proposed approach in simulations and in the analysis of a real-life fMRI dataset.

The rest of this paper is organized as follows. Section 2 describes the proposed formulation. An optimization algorithm that we develop to solve the formulation is depicted in Section 3. Section 4 provides theoretical analysis of the proposed formulation. Experimental results are included and discussed in Section 5, followed by the conclusions in Section 6.

2 Method

In our approach, the predictive model takes the form of the inner product of the data tensor $\mathcal{X}$ and the model coefficient tensor $\mathcal{W}$. The model coefficients are organized into a tensor rather than a vector or matrix, used in traditional analysis, because this way reflects the structure in different mode directions as shown in Figure 1.1. For

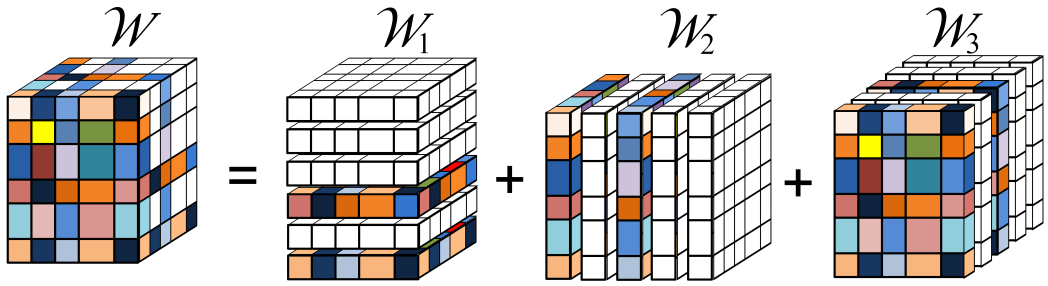


Figure 1.1: If we associate with each entry of the data tensor a weight in our additive prediction model to the outcome, then our model coefficients form a tensor $\mathcal{W}$ (upper left). If the coefficient tensor is sparse, then the resultant model will be selective in terms of vertical direction as $\mathcal{W}_1$, horizontal direction as $\mathcal{W}_2$, and outgoing direction as $\mathcal{W}_3$.

notational convenience, we just use $\mathcal{X}$ to represent the data that are used to form the model.

2.1 Preliminaries

We introduce the notation that is used throughout the paper. A bold lower case letter denotes a vector, such as $\mathbf{v}$. A bold upper case letter denotes a matrix, such as $\mathbf{M}$. A calligraphic upper case letter denotes a tensor such as $\mathcal{A}$. Let $\mathcal{A} \in \mathbb{R}^{d_1 \times \cdots d_K}$ be a K-way tensor. We denote the total number of entries in $\mathcal{A}$ by $N = \prod_{k=1}^{K} d_k$. Similarly, vect$(\mathcal{A})$ is the column-major vectorization of $\mathcal{A}$. The inner product between two tensors $\mathcal{A}$ and $\mathcal{B}$ is defined as $\langle \mathcal{A}, \mathcal{B} \rangle = \text{vect}(\mathcal{A})^\top \text{vect}(\mathcal{B})$. The Frobenius norm of a tensor is defined as $|||\mathcal{A}|||_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. Each dimensionality of a tensor is called a *mode*. The j-th sub-tensor along mode k is denoted as $\mathcal{A}_{(k)}^{(j)} = \mathcal{A}(i_1, \cdots, i_k = j, \cdots, i_K)$ where $i_{\{l|l \in \{1 \cdots K\}/\{k\}\}} = 1 : d_l$ (i.e. a $(K-1)$-way tensor that pulls out from $\mathcal{A}$ at j-th position of mode-k direction.). The mode k *unfolding* $\mathbf{A}_{(k)} \in \mathbb{R}^{d_k \times N/d_k}$ is a matrix that is obtained by concatenating the mode-k fibers along columns, where a mode-k fiber refers to an d_k dimensional vector obtained by fixing all the indices

but the kth index of $\mathcal{A}$. The operator, $[\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_m]_k$, concatenates the tensors along kth direction. It can create a $(K+1)$-way tensor by concatenating the K-way tensors as $[\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_m]_{K+1}$.

2.2 The Proposed Formulation

In our approach, assume that we are given data of m individuals (subjects) containing d_1 number of features (independent variables) that are repeatedly measured at T different time points and from $(K-2)$ different conditions (e.g. positions) for each individual i. The data of each individual i at time point t is represented by a $(K-1)$-way tensor $\mathcal{X}_t^{(i)}$ of size $\{d_1 \times \cdots \times d_{K-1}\}$. Each training example consists of the current and τ previous records of the repeated measurements. Let

$$\mathcal{X}_{(i;t)} = \left[\mathcal{X}_t^{(i)}, \mathcal{X}_{t-1}^{(i)}, \cdots, \mathcal{X}_{t-\tau}^{(i)}\right]_K$$

be a K-way data tensor for subject i of size $\{d_1 \times \cdots \times d_{K-1} \times (\tau+1)\}$ at time point t. Thus, there are totally $N = (\tau+1)\prod_{k=1}^{K-1} d_k$ elements in $\mathcal{X}_{(i;t)}$. Given T total measurements for each subject, the index t of $\mathcal{X}_{(i;t)}$ starts from $\tau+1$ in order to have enough previous observations in the first training example. Hence, there are totally $n = T - \tau$ training examples for each subject. If $\mathcal{X}_{(i;t)}$ includes previous $\tau+1$ values of $y^{(i)}$ as a feature, then the model $y_t^{(i)} = \langle \mathcal{X}_{(i;t)}, \mathcal{W} \rangle$ essentially gives the same model as in the graphical Granger models. The Granger models would assume that the training examples are *i.i.d.*. However, the consecutive examples are not mutually independent because they contain overlapping records (e.g., $\mathcal{X}_{(i;t)}$ and $\mathcal{X}_{(i;t+1)}$ share $\tau - 1$ records $\mathcal{X}_t^{(i)}$, $\cdots$, $\mathcal{X}_{t-\tau+1}^{(i)}$).

QIF provides a mechanism to estimate the sample correlation simultaneously while constructing predictive models, and to extend the linear models to generalized linear models. It assumes that the dependent variable comes from a class of distributions known as the exponential family. For each member in this family, there exists a link function that can be used to translate the nonlinear model into a linear model. The expectation of the outcome $y_t^{(i)}$ for subject i at time t is computed as:

$$E(y_t^{(i)}) = \mu_t^{(i)} = l^{-1}(\eta_t^{(i)}),$$

where $\mu_t^{(i)}$ represents the mean model, l^{-1} is the inverse of a link function g in a GLM [40], and $\eta_t^{(i)}$ is defined as $\eta_t^{(i)} = \langle \mathcal{X}_{(i;t)}, \mathcal{W} \rangle$.

We estimates the parameters $\mathcal{W}$ by satisfying the moment assumption of the quasilikelihood equation of the problem for m subjects as

$$EE(\mathcal{W}) = \sum_{i=1}^{m} \left(\mathbf{D}^{(i)}\right)^{\top} \left(\mathbf{\Sigma}^{(i)}\right)^{-1} \mathbf{s}^{(i)} = 0. \tag{2.1}$$

where the $\{n \times N\}$ matrix $\mathbf{D}^{(i)} = \partial \boldsymbol{\mu}^{(i)} / \partial \text{vect}(\mathcal{W})$, $\boldsymbol{\mu}^{(i)}$ combines all $\mu_t^{(i)}, \forall t = 1, \cdots, n$ into a vector, and $\mathbf{s}^{(i)} = \mathbf{y}^{(i)} - \boldsymbol{\mu}^{(i)}(\mathcal{W})$. The $n \times n$ matrix $\mathbf{\Sigma}^{(i)}$ is the estimated covariance structure as:

$$\mathbf{\Sigma}^{(i)}(\boldsymbol{\alpha}) = \left(\mathbf{A}^{(i)}\right)^{1/2} \mathbf{R}(\boldsymbol{\alpha}) \left(\mathbf{A}^{(i)}\right)^{1/2} \tag{2.2}$$

where $\mathbf{A}^{(i)}$ is an $n \times n$ diagonal matrix with $\text{var}(\mu_t^{(i)})$ as the t-th diagonal element.

Similar to QIF, we models $\mathbf{R}^{-1}$ by the class of matrices

$$\sum_{i=1}^{d} a_i \mathbf{M}_i, \tag{2.3}$$

where $\mathbf{M}_1, \cdots, \mathbf{M}_d$ are known matrices and $a_1, \cdots, a_d$ are unknown constants. The choices of the set of $\mathbf{M}$s have been well studied in [54]. We set $\mathbf{M}_1$ as an identity matrix, $\mathbf{M}_2$ to be all 1s on off-diagonal entries, $\mathbf{M}_3$ to be all 1s on 1st off-diagonal entries, $\mathbf{M}_4$ to be all 1s on 2nd off-diagonal entries and so on.

Substituting Eq.(2.2) and Eq.(2.3) into Eq.(2.1), we can obtain an extended score vector as

$$\mathbf{g}(\mathcal{W}) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{g}_i(\mathcal{W}) = \frac{1}{m} \begin{pmatrix} \sum_{i=1}^{m} \left(\mathbf{D}^{(i)}\right)^{\top} \left(\mathbf{A}^{(i)}\right)^{-1/2} \mathbf{M}_1 \left(\mathbf{A}^{(i)}\right)^{-1/2} \mathbf{s}^{(i)} \\ \vdots \\ \sum_{i=1}^{m} \left(\mathbf{D}^{(i)}\right)^{\top} \left(\mathbf{A}^{(i)}\right)^{-1/2} \mathbf{M}_d \left(\mathbf{A}^{(i)}\right)^{-1/2} \mathbf{s}^{(i)} \end{pmatrix},$$

so that the estimating function becomes a linear combination of the elements of the extended score vector. We want to minimize the errors of the linear combination as following

$$\min_{\mathcal{W}} \qquad \mathcal{Q}(\mathcal{W}) = m\mathbf{g}(\mathcal{W})^{\top}\mathbf{C}^{-1}\mathbf{g}(\mathcal{W}). \tag{2.4}$$

where $\mathbf{C}^{-1} = \mathbf{a}\mathbf{a}^{\top}$. [20] has shown that $\mathcal{W}$ is efficiently estimated if $\mathbf{C}$ is the variance matrix of $\mathbf{g}$ as $\mathbf{C} = (1/m)\sum_{i=1}^{m} \mathbf{g}_i(\mathcal{W})\mathbf{g}_i(\mathcal{W})^{\top}$.

Now, to select among features along different directions in predicting y, (and also to control the model capacity,) we apply regularizers to the model parameters. We first decompose $\mathcal{W}$ into a summation of K components as $\mathcal{W} = \sum_{k=1}^{K} \mathcal{W}_k$. We propose a novel subtensor-wise LASSO on those $\mathcal{W}_k$s which is similar as the $\ell_{1,2}$ norm for a

matrix. If we denote a set $\Phi_K = \{\mathcal{W}_1, \cdots, \mathcal{W}_K\}$ as a set of function input arguments, the regularization terms are adopted as following,

$$R(\Phi_K) = \sum_{k=1}^{K} \left(\lambda_k \sum_{j=1}^{d_k} \|\!| (\mathcal{W}_k)_{(k)}^{(j)} |\!\| _F \right), \tag{2.5}$$

where $d_K := \tau + 1$ and $\lambda_k, k = 1, \cdots, K$ are the hyper-parameters for the regularization terms and need to be tuned according to the data. To the best of our knowledge, it has not been studied how to produce sparse effects simultaneously on all the features from all different directions through proper regularization.

In our tensor components $\mathcal{W}_k$, we apply the Frobenius norm to the sub-tensors of $\mathcal{W}_k$ along the mode k. Then, a LASSO-like ℓ_1 norm operation shrinks the vector that consists of the values of the Frobenius norm of those sub-tensors so that the optimal solution of $\mathcal{W}_k$ will contain some sub-tensors along mode k with all zero entries. Therefore, a subset of features along mode k is used in the predictive model to predict the current outcome. Overall, by adding Eq.(2.5) into problem (2.4), we solve the following optimization problem for the best model parameters $\mathcal{W}$ which is computed as $\sum_{k=1}^{K} \mathcal{W}_k$:

$$\begin{aligned} &\min_{\Phi_K} f(\Phi_K) = \mathcal{Q}(\mathcal{W}) + R(\Phi_K) \\ &= m\mathbf{g}(\mathcal{W})^\top \mathbf{C}^{-1} \mathbf{g}(\mathcal{W}) + \sum_{k=1}^{K} \left(\lambda_k \sum_{j=1}^{d_k} \|\!| (\mathcal{W}_k)_{(k)}^{(j)} |\!\| _F \right) \\ &s.t. \qquad \mathcal{W} = \sum_{k=1}^{K} \mathcal{W}_k. \end{aligned} \tag{2.6}$$

We adopt an accelerated gradient descent method based on the fast iterative shrinkage-thresholding algorithm (FISTA) [6] to solve the problem since Eq.(2.6) is

a convex optimization problem in terms of $\mathcal{W}$. Our algorithm can be proved to find the global optimal solution $\mathcal{W}$ of Eq.(2.6).

3 Algorithm

In this section, we provide an algorithm to solve the problem described in Eq.(2.6). We then give out the exemplar exponential families that are suitable to the algorithm.

3.1 Optimization Algorithm

To solve Eq.(2.6), we develop a FISTA algorithm in the following iterative procedure to find optimal $\mathcal{W}_k$s.

Denote the iterates at the r-th iteration by $\mathcal{W}_k^r$s. Let $\nabla_{\mathcal{W}_k}\mathcal{Q}(\mathcal{W})$ be the partial derivative of $\mathcal{Q}(\mathcal{W})$ with respect to every $\mathcal{W}_k$ for given $k \in [1, \cdots, K]$, respectively, For any given point $\tilde{\Phi}_K = (\tilde{\mathcal{W}}_1, \cdots, \tilde{\mathcal{W}}_K)$, the following $Q_{L,\tilde{\Phi}_K}(\Phi_K)$ is a *well-defined* proximal map for the non-smooth R,

$$Q_{L,\tilde{\Phi}_K}(\Phi_K) = \mathcal{Q}(\tilde{\mathcal{W}}) + R(\Phi_K) + \sum_{k=1}^{K}\frac{L}{2}\|\mathcal{W}_k - \tilde{\mathcal{W}}_k\|_F^2 + \sum_{k=1}^{K}\langle\nabla_{\mathcal{W}_k}\mathcal{Q}(\tilde{\mathcal{W}}), \mathcal{W}_k - \tilde{\mathcal{W}}_k\rangle$$

if $\mathcal{Q}(\mathcal{W})$ has Lipschitz continuous gradient with Lipschitz modulus L. Then, according to the Lemma 2.1 in [6], the inequality $f(\Phi_K) \leq Q_{L,\tilde{\Phi}_K}(\Phi_K)$ holds indicating that $Q_{L,\tilde{\Phi}_K}(\Phi_K)$ is the upper bound of $f(\Phi_K)$.

Then, the update of the extrapolated point, $\tilde{\mathcal{W}}_k^r$, is defined as

$$\tilde{\mathcal{W}}_k^r = \mathcal{W}_k^{r-1} + \left(\frac{s^{r-1}-1}{s^r}\right)(\mathcal{W}_k^{r-1} - \mathcal{W}_k^{r-2}), \tag{3.1}$$

where s^r is a scalar and updated at each iteration as:

$$s^{r+1} = \frac{1 + \sqrt{1 + 4(s^r)^2}}{2}. \tag{3.2}$$

The optimal solution to problem Eq.(2.6) is equivalent to solving the following problem

$$\min_{\Phi_K} \sum_{k=1}^{K} \langle \nabla_{\mathcal{W}_k} \mathcal{Q}^r, \mathcal{W}_k - \tilde{\mathcal{W}}_k^r \rangle + \sum_{k=1}^{K} \frac{L}{2} |||\mathcal{W} - \tilde{\mathcal{W}}_k^r|||_F^2 + R(\Phi_K) \tag{3.3}$$

for a solution set Φ_K, where all $\nabla_{\mathcal{W}_1} \mathcal{Q}^r, \cdots, \nabla_{\mathcal{W}_K} \mathcal{Q}^r$ are the partial derivatives of $\mathcal{Q}$ computed at $\tilde{\Phi}_K^r$, and L actives as a learning step size.

Since there is no interacting term among $\mathcal{W}_k$s in Eq.(3.3), the problem can be decomposed into K separate subproblems as follows:

$$\min_{\mathcal{W}_k} \langle \nabla_{\mathcal{W}_k} \mathcal{Q}^r, \mathcal{W}_k - \tilde{\mathcal{W}}_k^r \rangle + \frac{L}{2} |||\mathcal{W}_k - \tilde{\mathcal{W}}_k^r|||_F^2 + \lambda_k \sum_{i=1}^{n_k} |||(\mathcal{W}_k)_{(k)}^{(i)}|||_F \tag{3.4}$$

where $k \in \{1, \ldots, K\}$. Those K subproblems share the same structure and thus can be solved following the same procedure. Hence, we only show how to solve (3.4) for the best $\mathcal{W}_k$.

Eq.(3.4) w.r.t $\mathcal{W}_k$ is equivalent to the following problem

$$\min_{\mathcal{W}_k} \frac{1}{2} \left| \left| \left| \mathcal{W}_k - \left(\tilde{\mathcal{W}}_k^r - \frac{1}{L} \nabla_{\mathcal{W}_k} \mathcal{Q}^r \right) \right| \right| \right|_F^2 + \frac{\lambda_1}{L} \sum_{i=1}^{n_k} |||(\mathcal{W}_k)_{(k)}^{(i)}|||_F$$

after omitting constants, this problem has a closed-form solution where each subtensor of $\mathcal{W}_k^r$ is:

$$(\mathcal{W}_k^r)_{(k)}^{(i)} = \max \left(0, 1 - \frac{\lambda_1}{L |||(\mathcal{P}^r)_{(k)}^{(i)}|||_F} \right) (\mathcal{P}^r)_{(k)}^{(i)},$$

and $\mathcal{P}^r = \tilde{\mathcal{W}}_k^r - \frac{1}{L}\nabla_{\mathcal{W}_k}\mathcal{Q}^r$.

[54] has given the gradient of QIF as:

$$\nabla_{\mathcal{W}_k}\mathcal{Q} = 2\nabla_{\mathcal{W}_k}\mathbf{g}^\top\mathbf{C}^{-1}\mathbf{g} - \mathbf{g}^\top\mathbf{C}^{-1}\nabla_{\mathcal{W}_k}\mathbf{C}\mathbf{C}^{-1}\mathbf{g}, \tag{3.5}$$

where $\nabla_{\mathcal{W}_k}\mathbf{g}$ is the $\{dN \times N\}$ matrix and $\nabla_{\mathcal{W}_k}\mathbf{C}$ is the 3D array with the size $\{dN \times dN \times N\}$. Note that the product of a matrix to a tensor means to multiply the matrix to each slice of the tensor and result a new tensor. If we concatenate $\mathbf{g}_i$ as $G = [\mathbf{g}]_{i=1}^m$, then $\nabla_{\mathcal{W}_k}\mathbf{C}$ can be represented as

$$\nabla_{\mathcal{W}_k}\mathbf{C} = \left(G\nabla_{\mathcal{W}_k}(\mathbf{G}^\top) + \nabla_{\mathcal{W}_k}\mathbf{G}\mathbf{G}^\top\right), \tag{3.6}$$

where $\nabla_{\mathcal{W}_k}\mathbf{G}$ is a $\{dN \times m \times N\}$ tensor that concatenate matrices $\nabla_{\mathcal{W}_k}\mathbf{g}_i$s as its columns.

In the above discussion, the Lipschitz modulus L needs to be computed. However, the calculation of L can be computationally expensive. We therefore follow the similar argument in [78] to find a proper approximation $\tilde{L}$. Recall that the Lipschitz constant L is defined as $L = \max_{\mathcal{W}} \lambda_{\max}\left(\nabla\nabla\mathcal{Q}(\mathcal{W})\right)$ where $\lambda_{\max}(\cdot)$ indicates the maximum singular value of the Hessian of $\mathcal{Q}$. Decomposing the Hessian matrix $\nabla\nabla\mathcal{Q}_{\mathbf{W}}|_{\mathbf{W}\to 0}$ into $\mathbf{M}^\top\mathbf{M}$ where $\mathbf{M} \in \mathbb{R}^{d(\tau+1)\times q}$ and q is the rank of the Hessian matrix yields an upper bound of L as $L \leq \tilde{L} = ||\mathbf{M}||_{\infty,1}||\mathbf{M}^\top||_{\infty,1}$. We use the upper bound $\tilde{L}$ as L in our iterations. However, using this upper bound may increase the number of iterative steps for convergence.

Algorithm 3 summarizes the steps for finding optimal $\mathcal{W}_k$s with $k = 1, \cdots, K$.

Algorithm 3: Search for optimal $\mathcal{W}_k$s

Input: $\mathcal{X}$, $\mathbf{y}$, $\lambda_1, \cdots, \lambda_K$
Output: solution set Φ_K
1. $r = 1$, compute $\tilde{L}$ and initialize $s^1 = 1$, $\mathcal{W}_k^0 = \tilde{\mathcal{W}}_k^1 = \mathbf{0}$ for given $k = 1, \cdots, K$;
2. Solve Eq.(3.3) to obtain Φ_K^r.
3. Compute s^{r+1} by Eq.(3.2).
4. Compute $\tilde{\Phi}_K^{r+1}$ by Eq.(3.1).
5. $r = r + 1$.
Repeat $2 \sim 5$ until convergence.

3.2 Exemplar Exponential Families

The proposed algorithm is suitable for optimizing any loss function that has a Lipschitz continuous gradient. In this section we discuss two exemplar exponential families: Gaussian and Bernoulli here. We specify how to compute the gradient of the extended score vectors, $\mathbf{g}_i(\mathcal{W})$ for these distributions. The gradients will be substituted in Eq.(3.5) and Eq.(3.6) used in our algorithm.

Exemplar of Gaussian Distribution

If the outcome follows a Gaussian distribution, then y is linearly regressive in terms of the covariates in the observations. The mean and variance of μ are calculated as:

$$\mu_t^{(i)} = E(y^{(i)}) = \langle \mathcal{X}^{(i;t)}, \mathcal{W} \rangle,$$
$$\mathbf{A}^{(i)} = \text{var}\left(\boldsymbol{\mu}^{(i)}\right) = \mathbf{I},$$

so the gradient $\nabla_{\mathcal{W}_k}\mathbf{g}$ at the r-th iteration can be computed as

$$\nabla_{\mathcal{W}_k}\mathbf{g} = -\frac{1}{m}\sum_{i=1}^{m}\begin{pmatrix} \left(\mathbf{D}^{(i)}\right)^{\top}\mathbf{M}_1\mathbf{D}^{(i)} \\ \vdots \\ \left(\mathbf{D}^{(i)}\right)^{\top}\mathbf{M}_d\mathbf{D}^{(i)} \end{pmatrix},$$

where $\mathbf{D}^{(i)} = [\text{vect}(\mathcal{X}_{(i;1)}), \cdots, \text{vect}(\mathcal{X}_{(i;n)})]^{\top}$ is derived from $\partial\boldsymbol{\mu}^{(i)}/\partial\text{vect}(\mathcal{W})$.

Exemplar of Bernoulli Distribution

If the generalized variables μ follow a Bernoulli distribution and the outcomes are binary variables. The relationship between the outcome and covariates can be learned by a logistic regression which is a special case of the GLM with the Bernoulli assumption. Hence, the mean of y and loss function are calculated as:

$$\begin{aligned} E(y^{(i)}) = \mu^{(i)} &= \frac{\exp(\langle\mathcal{X}^{(i)}, \mathcal{W}\rangle)}{1+\exp(\langle\mathcal{X}^{(i)}, \mathcal{W}\rangle)}, \\ \mathbf{A}^{(i)} &= \text{var}\left(\boldsymbol{\mu}^{(i)}\right) = \text{diag}\left(\langle\boldsymbol{\mu}^{(i)}, 1-\boldsymbol{\mu}^{(i)}\rangle\right), \\ \mathbf{X} &= [\text{vect}(\mathcal{X}_{(i;1)}), \cdots, \text{vect}(\mathcal{X}_{(i;n)})], \\ \mathcal{A}^{(i)} = \nabla_{\mathcal{W}_k}\mathbf{A}^{(i)} &= \text{tendiag}\left(\left(-\frac{1}{2}\exp(\frac{1}{2}\langle\mathcal{X}^{(i)}, \mathcal{W}\rangle) + \frac{1}{2}\exp(\frac{1}{2}\langle\mathcal{X}^{(i)}, \mathcal{W}\rangle)\right)\mathbf{X}^{\top}\right) \end{aligned}$$

where tendiag place the $\{n \times N\}$ matrix into the diagonal plant of the $\{n \times n \times N\}$ tensor. And

$$\begin{aligned} \mathbf{D}^{(i)} = \nabla_{\mathcal{W}_k}\boldsymbol{\mu}^{(i)} &= \frac{\exp(\langle\mathcal{X}^{(i)}, \mathcal{W}\rangle)}{\left(1+\exp(\langle\mathcal{X}^{(i)}, \mathcal{W}\rangle)\right)^2}\mathbf{X}^{\top}, \\ \mathcal{D} = \nabla_{\mathcal{W}_k}\nabla_{\mathcal{W}_k}\boldsymbol{\mu}^{(i)} &= \frac{\exp(3\langle\mathcal{X}^{(i)}, \mathcal{W}\rangle) - \exp(\langle\mathcal{X}^{(i)}, \mathcal{W}\rangle)}{\left(1+\exp(\langle\mathcal{X}^{(i)}, \mathcal{W}\rangle)\right)^4} \otimes \left(\mathbf{X}\mathbf{X}^{\top}\right), \end{aligned}$$

where $\otimes$ is outer product of vector/matrix and $\mathcal{D}$ is an $\{n \times N \times N\}$ tensor. Then, we define that

$$\begin{aligned}\nabla_{\mathcal{W}_k}\mathbf{g}_{\mathbf{M}_i} = &-\left(\mathbf{D}^{(i)}\right)^{\top}\left(\mathbf{A}^{(i)}\right)^{-\frac{1}{2}}\mathbf{M}_i\left(\mathbf{A}^{(i)}\right)^{-\frac{1}{2}}\mathbf{D}^{(i)} + \left(\mathcal{D}^{(i)}\right)^{\top}\left(\mathbf{A}^{(i)}\right)^{-\frac{1}{2}}\mathbf{M}_i\left(\mathbf{A}^{(i)}\right)^{-\frac{1}{2}}\mathbf{s}^{(i)} \\ &+\left(\mathbf{D}^{(i)}\right)^{\top}\left(\mathcal{A}^{(i)}\right)^{-\frac{1}{2}}\mathbf{M}_i\left(\mathbf{A}^{(i)}\right)^{-\frac{1}{2}}\mathbf{s}^{(i)} + \left(\mathbf{D}^{(i)}\right)^{\top}\left(\mathbf{A}^{(i)}\right)^{-\frac{1}{2}}\mathbf{M}_i\left(\mathcal{A}^{(i)}\right)^{-\frac{1}{2}}\mathbf{s}^{(i)}.\end{aligned}$$

Therefore, the gradient $\nabla_{\mathcal{W}_k}\mathbf{g}$ at the r-th iteration can be computed as

$$\nabla_{\mathcal{W}_k}\mathbf{g} = -\frac{1}{m}\sum_{i=1}^{m}\begin{pmatrix}\nabla_{\mathcal{W}_k}\mathbf{g}_{\mathbf{M}_1}\\ \vdots \\ \nabla_{\mathcal{W}_k}\mathbf{g}_{\mathbf{M}_d}\end{pmatrix},$$

4 Theoretical Analysis

We provide a convergence analysis for Algorithm 1 and an asymptotic analysis of the model. We also theoretically discuss the group support in terms of the values of λ_k.

4.1 Convergence Analysis

In this section, we give out the convergence analysis and the convergence rate of the proposed algorithm. We prove that our algorithm will converge to a global optimal solution with a convergence rate of $O(1/r^2)$. The proof follows largely the arguments in [6]. We only provide a sketch proof here.

Theorem 4. *Let $\Phi_K^r = \{\mathcal{W}_1^r, \cdots, \mathcal{W}_K^r\}$ be the set of the matrix generated by Algo-*

rithm 1. Then for any $k \geq 1$

$$f(\Phi_K^r) - f(\hat{\Phi}_K) \leq \frac{2\tilde{L}\sum_{k=1}^K \|\mathcal{W}_k^0 - \hat{\mathcal{W}}_k\|_F^2}{(r+1)^2}$$

where $\hat{\Phi}_K = (\hat{\mathcal{W}}_1, \cdots, \hat{\mathcal{W}}_K)$ *is a globally optimal solution set of*

$$\min_{\mathcal{W}_k, k=1,\cdots,K} f(\Phi_K) = \mathcal{Q}(\mathcal{W}) + R(\Phi_K) = \mathbf{g}(\mathcal{W})^\top \mathbf{C}^{-1}\mathbf{g}(\mathcal{W}) + \sum_{k=1}^K \left(\lambda_k \sum_{i=1}^{n_k} \|(\mathcal{W}_k)_{(k)}^{(i)}\|_F\right) \tag{4.1}$$

$$s.t. \qquad \mathcal{W} = \sum_{k=1}^K \mathcal{W}_k.$$

Proof. We start with defining the following quantities

$$v^r = f(\mathcal{W}_1^r, \cdots, \mathcal{W}_K^r) - f(\hat{\mathcal{W}}_1, \cdots, \hat{\mathcal{W}}_K), \qquad a^r = \frac{2}{L^r}(s^r)^2 v^r,$$
$$b^r = \sum_{k=1}^K \|s^r\mathcal{W}_k^r - (s^r-1)\mathcal{W}_k^{r-1} - \hat{\mathcal{W}}_k\|_F^2, \qquad c = \sum_{k=1}^K \|\tilde{\mathcal{W}}_k^1 - \hat{\mathcal{W}}_k\|_F^2 = \sum_{k=1}^K \|\mathcal{W}_k^0 - \hat{\mathcal{W}}_k\|_F^2,$$

where $\tilde{\mathcal{W}}_k^1 = \mathcal{W}_k^0$ for given $k = 1, \cdots, K$, and subsequent $\tilde{\mathcal{W}}_k^r$ for given $k = 1, \cdots, K$ are defined by Eq.(3.1).

Following the proof of Theorem 4.4 in [6], in the first iteration, given $s^1 = 1$, we have $a^1 = \frac{2}{L^1}v^1$, and $b^1 = \sum_{k=1}^K \|\mathcal{W}_k^1 - \hat{\mathcal{W}}_k\|_F^2$. We show that $a_1 + b_1 \leq c$ by applying

Lemma 2.3 in [6], which yields

$$\begin{aligned} & f(\hat{\mathcal{W}}_1, \cdots, \hat{\mathcal{W}}_K) - f(\mathcal{W}_1^r, \cdots, \mathcal{W}_K^r) = -v^1 \\ \geq & \frac{L^1}{2} \sum_{k=1}^{K} |||\mathcal{W}_k^1 - \tilde{\mathcal{W}}_k^1|||_F^2 + L^1 \sum_{k=1}^{K} \langle \tilde{\mathcal{W}}_k^1 - \hat{\mathcal{W}}_k, \mathcal{W}_k^1 - \tilde{\mathcal{W}}_k^1 \rangle \\ = & \frac{L^1}{2} \sum_{k=1}^{K} (|||\mathcal{W}_k^1 - \hat{\mathcal{W}}_k|||_F^2 - |||\tilde{\mathcal{W}}_k^1 - \hat{\mathcal{W}}_k|||_F^2). \end{aligned}$$

Reorganizing the above inequality yields

$$\frac{2}{L^1}(s^1)^2 v^1 + \sum_{k=1}^{K} |||\mathcal{W}_k^1 - \hat{\mathcal{W}}_k|||_F^2 \leq \sum_{k=1}^{K} |||\tilde{\mathcal{W}}_k^1 - \hat{\mathcal{W}}_k|||_F^2$$

Thus, $a^1 + b^1 \leq c$ holds.

Then, according to Lemma 4.1 in [6], we have for every $h \geq 1$, $a^r - a^{r+1} \geq b^{r+1} - b^r$, together with $a^1 + b^1 \leq c$, which yields the following inequality,

$$c \geq a^1 + b^1 \geq a^2 + b^2 \geq \cdots \geq a^k h + b^h \geq a^r.$$

Therefore, we obtain that

$$\frac{2}{L^h}(s^r)^2 v^h \leq \sum_{k=1}^{K} |||\mathcal{W}_k^0 - \hat{\mathcal{W}}_k|||_F^2. \tag{4.2}$$

Given s^r is updated according to

$$s^{r+1} = \frac{1 + \sqrt{1 + 4(s^r)^2}}{2},$$

it is easy to show that $s^r \geq \frac{(r+1)}{2}$. Substituting this inequality into Eq.(4.2) yields

$$v^h \leq \frac{2L^h \left(\sum_{k=1}^{K} \|\mathcal{W}_k^0 - \hat{\mathcal{W}}_k\|_F^2 \right)}{(r+1)^2}.$$

By the Remark 3.2 in [6] and the inequality

$$L \leq \tilde{L} = ||\mathbf{M}||_{\infty,1}||\mathbf{M}^\top||_{\infty,1},$$

we also know that an upper bound of L^r is $\tilde{L}$. Hence,

$$f(\mathcal{W}_1^r, \cdots, \mathcal{W}_K^r) - f(\hat{\mathcal{W}}_1, \cdots, \hat{\mathcal{W}}_K) \leq \frac{2\tilde{L} \sum_{k=1}^{K} \|\mathcal{W}_k^0 - \hat{\mathcal{W}}_k\|_F^2}{(r+1)^2}$$

In our algorithm, we set $L^r = \tilde{L}, \forall r$. □

4.2 Asymptotic Analysis

Let $\mathbf{w} = \text{vect}(\mathcal{W})$ and $\mathbf{X}_i$ be a $\{N \times n\}$ matrix whose t-th column is $\text{vect}(\mathcal{X}_{(i;t)})$. Then we have $\boldsymbol{\eta}^{(i)} = \mathbf{X}_i^T \mathbf{w}$ and the minimization problem (2.6) is equivalent to

$$\min_{\mathbf{w}} \mathbf{g}(\mathbf{w})^T \mathbf{C}^{-1} \mathbf{g}(\mathbf{w}) + \frac{1}{m} R(\mathbf{w}, \lambda_1, ..., \lambda_K). \tag{4.3}$$

For the asymptotic property of Eq.(4.3), we require the following regularity conditions [53]: (1) N is bounded and $\mathbf{C}$ converges to an invertible constant matrix $\mathbf{C}_0$ as $m \to \infty$; (2) $\mathbf{g}(\mathbf{w})$ converges to $\mathbf{g}_0(\mathbf{w})$ and $E[\mathbf{g}_0(\mathbf{w})]$ is continuous in $\mathbf{w}$; (3) The parameter space is compact and there is a unique interior point w^* satisfying $E[\mathbf{g}_0(\mathbf{w}^*)] = 0$; (4) $\mathbf{g}$ is differentiable and $\frac{\partial \mathbf{g}}{\partial \mathbf{w}}|_{\mathbf{w}=\hat{\mathbf{w}}}$ converges in probability to $\mathbf{J}_0 := E[\frac{\partial \mathbf{g}_0}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w}_0}]$ when $\hat{\mathbf{w}}$

converges in probability to $\mathbf{w}_0$.

Theorem 5. *Assume $\lambda_1 = \lambda_2 = ... = \lambda_K = \lambda$ are fixed in (4.3). Let $\hat{\mathbf{w}}$ be the estimator obtained by solving (4.3) and $\mathbf{w}^*$ be the true model coefficient. Then with the regularity conditions listed above, we have*

$$\sqrt{m}(\hat{\mathbf{w}} - \mathbf{w}^*) \to \mathcal{N}(\mathbf{0}, (\mathbf{J}_0^T \mathbf{C}_0^{-1} \mathbf{J}_0)^{-1}).$$

Proof. The minimizer $\hat{\mathbf{w}}$ of (4.3) satisfies the equation

$$\left(\frac{\partial \mathbf{g}}{\partial \mathbf{w}}\right)^T \mathbf{C}^{-1}\mathbf{g} - \frac{1}{2}\mathbf{g}^T\mathbf{C}^{-1}\left(\frac{\partial \mathbf{C}}{\partial \mathbf{w}}\right)\mathbf{C}^{-1}\mathbf{g} + \frac{\lambda}{2m}\frac{\partial R(\mathbf{w})}{\partial \mathbf{w}} = 0. \tag{4.4}$$

Since the second term in (4.4) is $O_p(m^{-1})$ and $\frac{\partial R}{\partial \mathbf{w}}$ is bounded, solving (4.4) is asymptotically equivalent to solving

$$(\mathbf{J}_0)^\top \mathbf{C}_0^{-1}\mathbf{g} = 0. \tag{4.5}$$

Then the Central Limit Theorem yields the conclusion. □

4.3 Group Support: The Value of λ_k

In this section, we focus on the linear model in which each component of $\boldsymbol{\mu}^{(i)}$ is given by $\mu_t^{(i)} = \langle \mathcal{X}_{(i;t)}, \sum_{k=1}^K \mathcal{W}_k \rangle$ and the components of outcome $\mathbf{y}^{(i)}$ are of the form $y_t^{(i)} = \mu_t^{(i)} + s_t^{(i)}$, where $\tau \le t \le T$. Since the QIF in this setting is symmetric with respect to tensors $\mathcal{W}_1, ..., \mathcal{W}_K$, we have for $1 \le k \le K$,

$$\nabla_{\mathcal{W}_k} Q(\mathcal{W})|_{\mathcal{W}=\mathcal{W}^*} = \nabla_{\mathcal{W}} Q(\mathcal{W})|_{\mathcal{W}=\mathcal{W}^*} := \mathcal{D} \tag{4.6}$$

where $\mathcal{D} = \mathcal{D}(\mathcal{X}, \mathbf{s})$ is a tensor with the same dimension as $\mathcal{W}$.

Motivated by the algorithm, we consider the following optimization problem for a fixed k:

$$\min_{\mathcal{W}_k} \frac{1}{2} \|\mathcal{W}_k - \mathcal{W}_k^* + \mathcal{D}\|_F^2 + \frac{\lambda_k}{L} \sum_{j=1}^{d_k} \|(\mathcal{W}_k)_{(k)}^{(j)}\|_F \tag{4.7}$$

Our goal is to estimate the group support for $\mathcal{W}_k^*$, i.e. obtain the subset $S_k^* \subset \{1, 2, ..., d_k\}$ such that $(\mathcal{W}_k^*)_{(k)}^{(j)} \neq 0$ if and only if $j \in S_k^*$.

Lemma 1. *Assume $\hat{\mathcal{W}}_k$ is a solution of* (4.7)*. Then either*

$$(\hat{\mathcal{W}}_k)_{(k)}^{(j)} \neq 0 \quad \textit{and} \quad (\hat{\mathcal{W}}_k)_{(k)}^{(j)} - (\mathcal{W}_k^*)_{(k)}^{(j)} + \mathcal{D}_{(k)}^{(j)} = -\frac{\lambda_k}{L} \frac{(\hat{\mathcal{W}}_k)_{(k)}^{(j)}}{\|(\hat{\mathcal{W}}_k)_{(k)}^{(j)}\|_F},$$

or

$$(\hat{\mathcal{W}}_k)_{(k)}^{(j)} = 0 \quad \textit{and} \quad \|(\mathcal{W}_k^*)_{(k)}^{(j)} + \mathcal{D}_{(k)}^{(j)}\|_F \leq \frac{\lambda_k}{L}.$$

Proof. These conditions satisfied KKT conditions. □

Lemma 2. *Assume*

$$\max_{1 \leq j \leq n_k} \|\mathcal{D}_{(k)}^{(j)}\|_F \leq \frac{\lambda_k}{2}. \tag{4.8}$$

Then (4.7) *has a solution $\hat{\mathcal{W}}_k$ such that*

$$\{j : (\hat{\mathcal{W}}_k)_{(k)}^{(j)} \neq 0\} := \hat{S}_k \subset S_k. \tag{4.9}$$

Furthermore, $\hat{S}_k = S_k^*$ if $\lambda_k < \frac{L}{2} \min_{j \in S} \|(\mathcal{W}_k^*)_{(k)}^{(j)}\|_F$.

Proof. For any tensor $\mathcal{W}$ and a set of indies S, we define $(\mathcal{W})^S_{(k)}$ by

$$((\mathcal{W})^S_{(k)})^{(j)}_{(k)} = \begin{cases} (\mathcal{W})^{(j)}_{(k)} & \text{if } j \in S \\ 0 & \text{otherwise.} \end{cases}$$

Let $\hat{\mathcal{W}}_k$ be a solution of the restricted version of (4.7):

$$\hat{\mathcal{W}}_k = \operatorname{argmin} \left\{ \frac{1}{2} \left\| (\mathcal{W}_k)^{S_k^*}_{(k)} - (\mathcal{W}_k^*)^{S_k^*}_{(k)} + \mathcal{D}^{S_k^*}_{(k)} \right\|^2_F + \frac{\lambda_k}{L} \sum_{j \in S} \|(\mathcal{W}_k)^{(j)}_{(k)}\|_F \right\}.$$

Then $(\hat{\mathcal{W}}_k)^{(j)}_{(k)} = 0$ for $j \in S_k^{*c}$. From Lemma 1 and (4.8), $\hat{\mathcal{W}}_k$ is a solution of (4.7) and $(\hat{\mathcal{W}}_k)^{(j)}_{(k)}$ satisfies

$$(\hat{\mathcal{W}}_k)^{(j)}_{(k)} - (\mathcal{W}_k^*)^{(j)}_{(k)} + \mathcal{D}^{(j)}_{(k)} = -\frac{\lambda_k}{L} (\mathcal{A})^{(j)}_{(k)}$$

for $j \in S_k^*$. Here $\|(\mathcal{A})^{(j)}_{(k)}\|_F \leq 1$ and

$$(\mathcal{A})^{(j)}_{(k)} = \frac{(\mathcal{W}_k)^{(j)}_{(k)}}{\|(\mathcal{W}_k)^{(j)}_{(k)}\|_F} \quad \text{if } (\mathcal{W}_k)^{(j)}_{(k)} \neq 0.$$

By the triangle inequality we have

$$\|(\hat{\mathcal{W}}_k)^{(j)}_{(k)}\|_F \geq \min_{j \in S_k^*} \|(\mathcal{W}_k^*)^{(j)}_{(k)}\|_F - \max_{j \in S_k^*} \|(\mathcal{U})^{(j)}_{(k)}\|_F$$

where

$$(\mathcal{U})^{(j)}_{(k)} = -\mathcal{D}^{(j)}_{(k)} - \frac{\lambda_k}{L} (\mathcal{A})^{(j)}_{(k)}.$$

Using (4.8) we deduce

$$\max_{j\in S_k^*} |||(\mathcal{U})_{(k)}^{(j)}|||_F \le \max_{j\in S_k^*} |||\mathcal{D}_{(k)}^{(j)}|||_F + \frac{\lambda_k}{L} \le \frac{2\lambda_k}{L}.$$

Thus $|||(\hat{\mathcal{W}}_k)_{(k)}^{(j)}|||_F > 0$ if $\frac{2\lambda_k}{L} < \min_{j\in S_K^*} |||(\mathcal{W}_k^*)_{(k)}^{(j)}|||_F$. □

Now we discuss the assumption (4.8) in Lemma 2. Suppose each entry $e_{k;i_1,i_2}$ of $\mathcal{D}_{(k)}^{(j)}$ independently follows a normal distribution $N(0, \sigma_{k,j;i_1,i_2}^2)$. Then

$$P(|||\mathcal{D}_{(k)}^{(j)}|||_F^2 \ge \frac{\lambda_k^2}{4}) \le P(Y_{k,j} \ge \frac{\lambda_k^2}{4\sigma_{max}^2})$$

where $\sigma_{max}^2 = \max_{j,i_1,i_2} \sigma_{k,j;i_1,i_2}^2$ and $Y_{k,j} = \sum_{i_1,i_2} \frac{e_{k;i_1,i_2}^2}{\sigma_{k,j;i_1,i_2}^2}$ follows $\chi_{D_k}^2$ with the degree of freedom $D_k = \prod_{\tilde{k}\neq k} d_{\tilde{k}}$. Note that D_k is also the size of $(W_k)_{(k)}^{(j)}$. For a random variable $Z \sim \chi_D^2$, we have

$$P(Z \ge z + (\sqrt{z} + \sqrt{D})^2) \le \exp(-z). \tag{4.10}$$

Applying (4.10) for $z = \frac{1}{4}\left(\sqrt{\frac{\lambda_k^2}{2\sigma_{max}^2} - D_k} - \sqrt{D_k}\right)^2$ yields

$$P(|||D_{(k)}^{(j)}|||_F^2 \ge \frac{\lambda_k^2}{4}) \le \exp\left[-\frac{1}{4}\left(\sqrt{\frac{\lambda_k^2}{2\sigma_{max}^2} - D_k} - \sqrt{D_k}\right)^2\right].$$

This indicates that to have (4.8) hold with high probability, we can choose $\lambda_k > \sigma_{max}\sqrt{2D_k}$.

5 Empirical Evaluation

In this section we presented the results of both synthetic examples and real-life fMRI example. We conducted to test the efficiency and effectiveness of the proposed method comparing to the state-of-the-art methods.

For comparison purposes, the data that contained the continuous response were examined by the proposed method, named as *TenQIF*, two baseline methods: **Linear Regression** (LR) and **least absolute shrinkage and selection operator** (LASSO), and four marginal models: QIF [54], PQIF [5], GEE [35], PGEE [17], and Graphical Granger Modeling [37]. For GEE and PGEE, We set the presumed correlation structure as 1st-order autoregressive structure (AR(1)). Namely, $corr(y_t^{(i)}, y_{t'}^{(i)}) = \alpha^{|t-t'|}$ for $0 < \alpha < 1$. The normalized Mean Square Error, nMSE, was employed to evaluate the performance of the predicting models.

5.1 Synthetic Data

First, we constructed synthetic datasets to investigate whether the proposed method can effectively discover the latent patterns along different modes. The dataset contained 150 subjects with 20 time points per subject. The data at each time point was a matrix with various sizes in $\{5 \times 5,\ 10 \times 10,\ 15 \times 15\}$, represented as $\mathbf{X}_t^{(i)}, i = 1, \cdots, 150$ and $t = 1, \cdots, 20$. $\mathbf{X}_t^{(i)}$s were generated from the normal distribution $N(0, 2^2)$. τ was set to 4. The tensor of the coefficients, $\mathcal{W}$, consisted of its decomposed tensors, $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$. Those three components were generated from the uniform distribution $U(0, 3^2)$. Each $\mathcal{W}_k, k = 1, 2, 3$ simulated the latent pattern along mode-k of the coefficient tensor. Respectively, $\mathcal{W}_1$ had patterns (i.e. non-zero values) in the 1st to 3rd features along mode-1, $\mathcal{W}_2$ had patterns in the 2nd and 3rd features

Table 5.1: Comparison of nMSE values among our method and the other methods on the synthetic datasets with different tensor sizes and real-life fMRI dataset.

$d_1 \times d_2 \times (\tau+1)$	LR	LASSO	QIF	PQIF	GEE	PGEE	Granger	TenQIF
$5 \times 5 \times 5$	0.903	0.833	0.766	0.755	0.903	0.851	0.316	**0.014**
$10 \times 10 \times 5$	0.947	0.851	0.812	0.684	0.944	0.863	0.419	**0.017**
$15 \times 15 \times 5$	0.975	0.863	0.842	0.827	0.921	0.883	0.565	**0.024**
fMRI	0.993	0.973	0.968	0.967	-	-	0.925	**0.741**

along mode-2, and $\mathcal{W}_3$ selected lagged patterns at the 1st, 3rd, and 5th lagged time points. We set the other parts of the component tensors without latent patterns to zero and computed $\mathcal{W} = \mathcal{W}_1 + \mathcal{W}_2 + \mathcal{W}_3$. We added residuals, $\mathbf{s}$, and $\sin(t)$ to the model in order to generate the outcome variables $\mathbf{y}$ for the subjects. The residuals $\mathbf{s}^{(i)}$ of every subject were generated from multivariate normal distribution with an AR(1) correlation structure at $\alpha = 0.6$. Thus, with different data sizes,, we have 3 different synthetic datasets. Finally, the outcome $y_t^{(i)}$ was computed as

$$y_t^{(i)} = \langle \mathcal{X}^{(i;t)}, (\mathcal{W}_1 + \mathcal{W}_2 + \mathcal{W}_3) \rangle + s_t^{(i)} + \sin(t).$$

In our implementation, we initialized all component tensors as zeros and iterated all the methods until a certain termination condition reached. Such termination condition could be either a maximum number of iteration steps or a small enough change of the value of the objective function in the iteration. The hyper-parameters were set to $\lambda_1 = \lambda_2 = \lambda_3 = 0.3$. 80% of the subjects were randomly selected to form the training data and the others were formed as the testing data.

Table 5.1 provides the nMSE comparison results between the proposed method and the other 7 methods for 3 synthetic datasets. The proposed method outperformed the traditional regression methods in all comparison scenarios in terms of predicting

accuracy. LR and LASSO had poor results in the synthetic datasets. This might be because the number of the records was relatively less in terms of the increasing number of total features in the data tensor. For the marginal models, QIF-based methods performed better than the baseline methods because they could efficiently estimate the coefficients even though the correlation matrix was misspecified while GEE-based methods performed poorer because they misspecified the correlation matrix. The Graphical Granger Modeling showed relatively high performance because it modeled the effects from lagged time points. However, its performance suffered from the high correlations within the subjects.

As illustrations, we plotted the mode-1 unfolded matrices of the three component tensors in Figure 5.1, which were resolved from *TenQIF* on the dataset of cube size $\{10 \times 10 \times 5\}$. This plot was interesting because in the three dimensional case the weight of the (i, j, k)-th entry was $W_{(i,j,k)}$, and we could clearly observe the selected sub-tensors (showed as vectors or matrices in the figures.) in those three component tensors. As shown in Figure 5.1, all 3 sub-tensors $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3$ captured the correct patterns as expected. This compiled with the designed structure of the synthetic data and explained the reason of achieving around 0.020 nMSE by the proposed method.

Figure 5.2 illustrated the results from the Granger modeling. The coefficient matrix was reshaped to align the matrices in Figure 5.1. It clearly showed the wrong selections on the first lagged time point which was due to the correlations within the subjects.

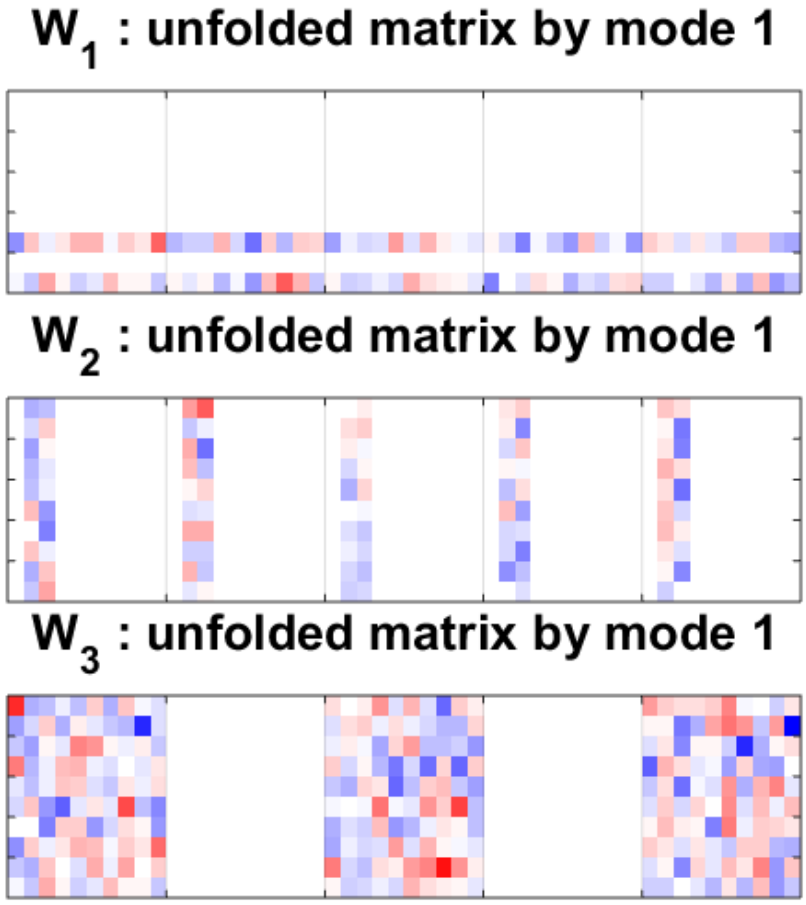


Figure 5.1: The model constructed by our approach TenQIF on the synthetic dataset with cube size $\{10 \times 10 \times 5\}$. Red (blue) color indicates that the corresponding features were positive (negative) predictors of the response variable. Features with white color were not selected by the model.

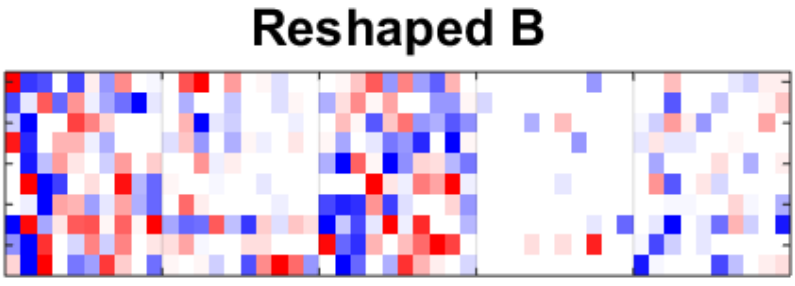


Figure 5.2: The model constructed by Granger modeling on the synthetic dataset with cube size $\{10 \times 10 \times 5\}$. The coefficient matrix is reshaped.

5.2 fMRI Data

Functional magnetic resonance imaging or functional MRI (fMRI) is a functional neuroimaging procedure using MRI technology that measures brain activity by detecting associated changes in blood flow. fMRI is an effective alternative approach to investigate the brain function related to the earliest symptoms of Alzheimer's disease, possibly before development of significant irreversible structural damage.

The fMRI data used in the experiment were collected by the Alzheimer's Disease Neuroimaging Initiative (ADNI)[1]. We cleaned up the fMRI data by filtering out the incomplete or low quality observations. After data cleaning, the data we used included 147 subjects diagnosed with mild cognitive impairment (MCI) from the year of 2009 to 2016. We used the participants' first fMRI scan as baseline and the other fMRI scans in 6th, 12th, 18th, and 24th months of the study. 67 out of 137 brain areas and 4 properties (CV,SA,TA,TS) out of 6 properties of the brain cortex were used in the model.[2]. These properties were **CV**: Cortical Volume; **SA**: Surface Area; **TA**: Thickness Average; **TS**: Thickness Standard Deviation. The outcome used in this experiment was the *mini-mental state examination* (MMSE) score quantified by a 30-point questionnaire, which was used extensively in clinical and research settings to measure cognitive impairment. At each time point, the MMSE score would be evaluated from participants' answers of the questionnaire.

20% of subjects in the dataset were used as the testing data. The lag variable τ was set to 2 when training the model. The hyper-parameters λ_1, λ_2, and λ_3 were tuned in a two-fold cross validation within the training data. In other words, the training records were further split into half: one used to build a model with a chosen

[1] http://adni.loni.usc.edu/

[2] Feature descriptions are available at http://adni.bitbucket.org/ucsffresfr.htmlFeature

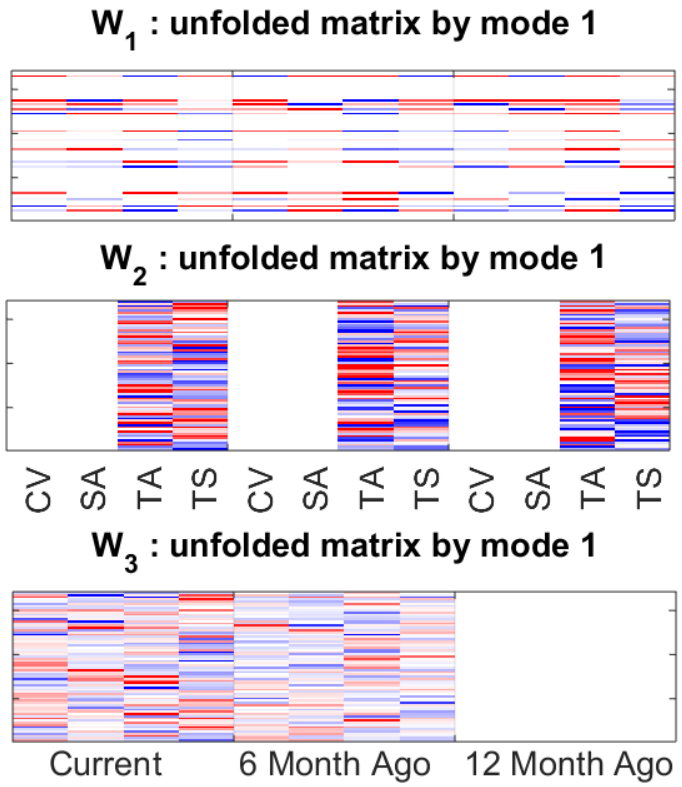


Figure 5.3: Columns, rows, and slices selected by the model for predicting MMSE score from participants' fMRI information.

parameter value from a range of 1 to 20 with a step size of 0.1; and the other used to test the resultant model. We chose the parameter values that gave the best two-fold cross validation performance. As shown in Table 5.1, our method performed the best predictions. Two GEE-based methods failed in the estimation of the presumed correlation matrix.

Moreover, our approach was able to select patterns along three dimensions: among the features, among the brain areas, and among the different lagged months. The hyper-parameters were chosen as $\lambda_1 = 6$, $\lambda_2 = 20$, and $\lambda_3 = 24$ for the training data. As shown in Figure 5.3, the structural damage of AD started from 6 months before played a major role in the development of the AD. Larger means and standard derivations of the thickness implied a higher risk of the AD. The proposed model selected 14 out of 68 brain areas that affected the MMSE score. According to the selections of the brain areas, the data at Cuneus area and Transverse Temporal area in both sides, and the data at right Inferior Parietal area, and so on might be important to predict the cognitive impairment.

5.3 EEG Data

Human memory function can be assayed in real-time by electroencephalographic (EEG) recording. However, the clinical utility of this method is dependent on the reliable determination of functionally and diagnostically relevant features. The proposed method approaches capable of modeling non-stationary signal have been explored as a way to synthesize large arrays of EEG data because the EEG record could be more precisely characterized by a tensor (e.g., a 3D matrix) representing processing stages, spatial locations, and frequency bands as individual dimensions.

Schizophrenia (SZ, $n = 40$) patients and healthy control (HC, $n = 20$) participants completed an EEG Sternberg task. EEG was analyzed to extract 5 frequency components (delta, theta, alpha, beta, gamma) at 4 processing stages (baseline, encoding, retention, retrieval) and 12 scalp sites representing central midline, and bi-lateral frontal and temporal regions. The proposed method and comparing methods were applied to the resulting 240 features (forming a $5 \times 4 \times 12$ tensor) to classify correct (-1) vs. incorrect (+1) responses on a trial-by-trial basis. In this approach, the proposed method guided the respective selection of spectral frequency, temporal (processing stages), and spatial (electrode sites) dimensions most related to trial performance. The correlations among processing stages were also estimated by the proposed method. Separate models were constructed for SZ and HC samples for comparison of common and disparate feature patterns across the dimensions.

For each of the SZ and HN datasets, 1/5 of the records were randomly chosen from every subject to form the test data and the rest of the records were used in training. The hyperparameters λ_1, λ_2, and λ_3 in our approach and GEE/PGEE (one parameter) were tuned in a two-fold cross validation within the training data. We chose the parameter values that gave the best two-fold cross validation performance, which were $\lambda_1 = 7.5, \lambda_2 = 5.5, \lambda_3 = 7.4$ for SZ and $\lambda_1 = 3.3, \lambda_2 = 2.1, \lambda_3 = 3.1$ for HN.

As shown in Figure 5.4, in both groups, task performance was most dependent on encoding and retrieval stage activity, with higher encoding uniformly and lower retrieval activity generally associated with better task performance across electrode sites. This pattern appears most prominently in central alpha activity (Figure 5.4; blue border). This indicated the same findings as in [8]. Groups differed in two main ways: (1) centroparietal theta, beta, and gamma during encoding and retention

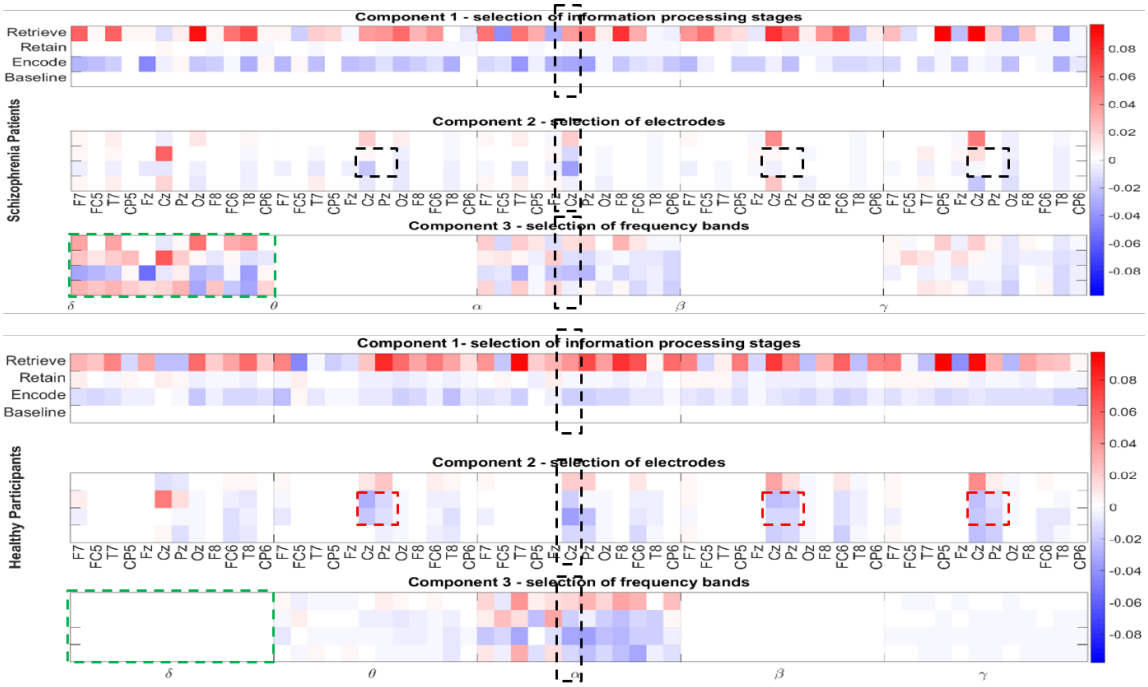


Figure 5.4: Columns, rows, and slices selected by the model to predict the successes of the memory tasks for SZ (top) and HN (bottom), respectively.

predicted higher accuracy in HC (Figure 5.4; red border), and (2) delta activity across stages and electrodes (Figure 5.4; green border) predicted lower accuracy in SZ. Here, the experimental results gave much clearer details of the working electrode sites and spectral frequencies comparing to the results in [8]. The proposed method outperformed GEE and SVM solutions according to AUC values (HC: 55.5%; SZ: 58.8% versus the best AUC 53% from the other methods). This was because the proposed method enabled interpretation and summary across all dimensions, which was not possible for classifiers based on single vectors.

6 Discussion

We have proposed new learning formulations called $TenQIF$ for longitudinal analytics, which can directly take data matrices or tensors as inputs and make predictions on that. The proposed method can simultaneously determine the influential features from the observations of different modes and the temporal contingency without any alternating strategies. The model parameter tensor is computed by the summation of K component tensors; each reflecting the selection among one mode. Moreover, the related optimization problem can be efficiently solved by a new accelerated gradient descent algorithm. Theoretical analysis shows the properties of the proposed method. Empirical studies on both synthetic and real-life fMRI and EEG problems demonstrate the superior performance of the proposed method.

Chapter 4

Hybrid-SDCA: A Double Asynchronous Approach for Stochastic Dual Coordinate Ascent

1 Introduction

The immense growth of data has made it important to efficiently solve large scale machine learning problems. It is necessary to take advantage of modern high performance computing (HPC) environments such as multi-core settings where the cores communicate through shared memory, or multi-processor distributed memory settings where the processors communicate by passing messages. In particular, a large class of supervised learning formulations, including support vector machines (SVMs), logistic regression, ridge regression and many others, solve the following generic regularized risk minimization (RRM) problem: given a set of instance-label pairs of data points

$(\mathbf{x}_i, y_i), i = 1, \ldots, n,$

$$\min_{\mathbf{w}\in\mathbb{R}^d} P(\mathbf{w}) := \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{x}_i^\top \mathbf{w}; y_i) + \frac{\lambda}{2} g(\mathbf{w}), \tag{1.1}$$

where $y_i \in \mathbb{R}$ is the label for the data point $\mathbf{x}_i \in \mathbb{R}^d$, $\mathbf{w} \in \mathbb{R}^d$ is the linear predictor to be optimized, ϕ is a loss function that is convex with respect to its first argument, λ is a regularization parameter that balances between the loss and a regularizer $g(\mathbf{w})$, especially the ℓ_2-norm penalty $\|\mathbf{w}\|_2^2$.

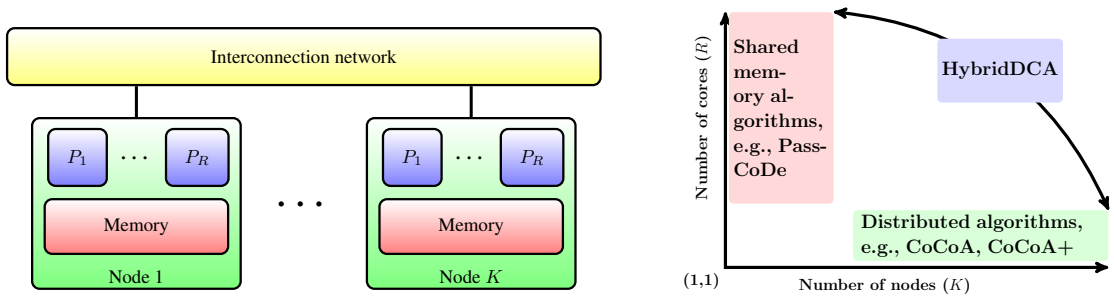
Many efficient sequential algorithms have been developed in the past decades to solve (1.1), *e.g.,* stochastic gradient descent (SGD) [83], or alternating direction method of multipliers (ADMM) [9]. Especially, (stochastic) dual coordinate ascent (DCA) algorithm [60] has been one of the most widely used algorithms for solving (1.1). It efficiently optimizes the following dual formulation (1.2)

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} D(\boldsymbol{\alpha}) := -\frac{1}{n}\sum_{i=1}^{n} \phi^*(-\alpha_i) - \frac{\lambda}{2} g^*\left(\frac{1}{\lambda n}\mathbf{X}\boldsymbol{\alpha}\right), \tag{1.2}$$

$$\text{using} \qquad \mathbf{w}(\boldsymbol{\alpha}) = \nabla g^*\left(\frac{1}{\lambda n}\mathbf{X}\boldsymbol{\alpha}\right), \tag{1.3}$$

where ϕ^*, g^* are the convex conjugates of ϕ, g, respectively, defined as, *e.g.,* $\phi^*(u) = \max_z(zu - \phi(z))$ and it is known that if $\boldsymbol{\alpha}^*$ is an optimal dual solution then $\mathbf{w}^* = \mathbf{w}(\boldsymbol{\alpha}^*)$ is an optimal primal solution and $P(\mathbf{w}^*) = D(\boldsymbol{\alpha}^*)$. The dual objective has a separate dual variable associated with each training data point. The stochastic DCA updates dual variables, one at a time, while maintaining the primal variables by calculating (1.3) from the dual variables.

Recently, many efforts have been undertaken to solve (1.1) in a distributed or parallel framework. It has been shown that distributed DCA algorithms have comparable



(a) (b)

Figure 1.1: (a) A simplified view of the modern HPC system and (b) Algorithms on this architecture.

and sometimes even better convergence than SGD-based or ADMM-based distributed algorithms [80]. The distributed DCA algorithms can be grouped into two sets. The first set contains synchronous algorithms in which a random dual variable is updated by each processor and the primal variables are synchronized across the processors in every iteration [30, 39, 80]. This approach incurs a large communication overhead. The second set of algorithms avoids communication overhead by exploiting the shared memory in a multi-core setting [28] where the primal variables are stored in a primary memory shared across all the processors. Further speedups have been obtained by using (asynchronous) atomic memory operations instead of costly locks for shared memory updates [28, 52]. Nevertheless, this approach is difficult to scale up for big datasets that cannot be fully accommodated in the shared memory. This leads to a challenging question: how do we scale up the asynchronous shared memory approach for big data while maintaining the speed up?

We address this challenge by proposing and implementing a hybrid strategy. The modern HPC platforms can be viewed as a collection of K nodes interconnected through a network as shown in Figure 1.1(a). Each node contains a memory shared

among R processing cores. Our strategy exploits this architecture by equally distributing the data across the local shared memory of the K nodes. Each of the R cores within a node runs a computing thread that asynchronously updates a random dual variable from those associated with the data allocated to the node. Each node also runs a communicating thread. One of the communicating threads is designated as *master* and the rest are *workers*. After every round of H local iterations in each computing thread, each worker thread sends the local update to the master. After accumulating the local updates from S of the K workers, the master broadcasts the global update to the contributing workers. However, to avoid a slower worker falling back too far, the master ensures that in every Γ consecutive global updates there is at least one local update from each worker. Figure 1.1(b) shows how our scheme is a generalization of the existing approaches: for $K = 1$, our setup coincides with the shared memory multi-core setting [28] and for $R = 1, S = K$ our setup coincides with the synchronous algorithms in distributed memory setting [30, 39, 80]. With a proper adjustment of the parameters H, S, Γ our strategy could balance the computation time of the first setting with the communication time of the second one, while ensuring scalability in big data applications.

Thus, our contributions are 1) we propose and analyze a hybrid asynchronous shared memory and asynchronous distributed memory implementation (*Hybrid-SDCA*) of the mostly used SDCA algorithm to solve (1.1); 2) we prove a strong guarantee of convergence for L-Lipschitz continuous loss functions, and further linear convergence when a smooth convex loss function is used; and 3) the experimental results using our light-weight OpenMP+MPI implementation show that our algorithms are much faster than existing distributed memory algorithms [30, 39], and easily scale up with the volume of data in comparison with the shared memory based algorithms [28] as

the shared memory size is limited.

2 Related Work

Sequential Algorithms. SGD is the oldest and simplest method of solving problem (1.1). Though SGD is easy to implement and converges to modest accuracy quickly, it requires a long tail of iterations to reach 'good' solutions and also requires adjusting a step-size parameter. On the other hand, SDCA methods are free of learning-rate parameters and have faster convergence rate around the end [44, 46]. A modified SGD has also been proposed with faster convergence by switching to SDCA after quickly reaching a modest solution [60]. Recently, 'variance reduced' modifications to the original SGD have also caught attention. These modifications estimate gradients with small variance as they approach to an optimal solution. Mini-batch algorithms are also proposed to update several dual variables (data points) in a batch rather than a single data point per iteration[66]. Mini-batch versions of both SGD and SDCA have slower convergence when the batch size increases[57, 61]. All these sequential algorithms become ineffective when the datasets get bigger.

Mini-batch Algorithms. To process big datasets faster researchers have proposed mini-batch algorithms, in which, instead of just using a single data sample per iteration, updates due to several data samples are 'batched' in each iteration. The idea is that the updates for the data samples in a batch can be computed in parallel using multiple processors. However, updates in a batch are tightly coupled and are useful only for smaller batches. Mini-batch versions of both SGD and SDCA have slower convergence when the batch size increases.

One-shot Communication Schemes. Researchers have also tried to 'decompose' the datasets into independent parts. Each of these parts can be solved independently and the final solution can be reached by 'accumulating' the partial solutions using a single round of communications. However, in general, the datasets from real world cannot be easily decomposed in such a manner and hence the one-shot schemes have very limited use.

Distributed Algorithms. In the early single communication scheme [43, 23, 42], a dataset is 'decomposed' into smaller parts that can be solved independently. The final solution is reached by 'accumulating' the partial solutions using a single round of communications. This method has limited utility because most datasets cannot be decomposed in such a way. Using the primal-dual relationship (1.3), fully distributed algorithms of DCA are later developed where each processor updates a separate α_i which is then used to update $\mathbf{w}(\boldsymbol{\alpha})$, and synchronizes $\mathbf{w}$ across all processors (*e.g.,* CoCoA [30]). To trade off communications vs computations, a processor can solve its subproblem with H dual updates before synchronizing the primal variable (*e.g.,* CoCoA+ [39],DisDCA [80]). In [80, 39], a more general framework is proposed in which the subproblem can be solved using not only SDCA but also any other sequential solver that can guarantee a Θ-approximation of the local solution for some $\Theta \in (0, 1]$. Nevertheless, the synchronized update to the primal variables has the inherent drawback that the overall algorithm runs at a speed of the slowest processor even when there are fast processors [1].

Parallel Algorithms. Multi-core shared memory systems have also been exploited, where the primal variables are maintained in a shared memory, removing the communication cost. However, updates to shared memory requires synchronization primitives, such as locks, which again slows down computation. Recent methods

[28, 36] avoid locks by exploiting (asynchronous) atomic memory updates in modern memory systems. There is even a wild version in [28] that takes arbitrarily one of the simultaneous updates. Though the shared memory algorithms are faster than the distributed versions, they have an inherent drawback of being not scalable, as there can be only a few cores in a processor board.

Other Distributed Methods for RRM. Besides distributed DCA methods, there are several recent distributed versions of other algorithms with faster convergence, including distributed Newton-type methods (DISCO [86], DANE [62]) and distributed stochastic variance reduced gradient method (DSVRG [33]). It has been shown that they can achieve the same accurate solution using fewer rounds of communication, however, with additional computational overhead. In particular, DISCO and DANE need to solve a linear system in each round, which could be very expensive for higher dimensions. DSVRG requires each machine to load and store a second subset of the data sampled from the original training data, which also increase its running time.

The ADMM [9] and quasi-Newton methods such as L-BFGS also have distributed solutions. These methods have low communication cost, however, their inherent drawback of computing the full batch gradient does not give computation vs communications trade-off. In the context of consensus optimization, [82] gives an asynchronous distributed ADMM algorithm but that does not directly apply to solving (1.1).

To the best of our knowledge, this paper is the first to propose, implement and analyze a hybrid approach exploiting modern HPC architecture. Our approach is the amalgamation of three different ideas – 1) CoCoA+/DisDCA distributed framework, 2) asynchronous multi-core shared-memory solver [28] and 3) asynchronous distributed approach [82] – taking the best of each of them. In a sense ours is the

first algorithm which asynchronously uses updates which themselves have been computed using asynchronous methods.

3 Algorithm

At the core of our algorithm, the data is distributed across K nodes and each node, called a *worker*, repeatedly solves a perturbed dual formulation on its data partition and sends the local update to one of the nodes designated as the *master* which merges the local updates and sends back the accumulated global update to the workers to solve the subproblem once again, unless a global convergence is reached. Let $\mathcal{I}_k \subseteq \{1, 2, \ldots, n\}, k = 1, \ldots, K$ denote the indices of the data and the dual variables residing on node k and $n_k = |\mathcal{I}_k|$. For any $\mathbf{z} \in \mathbb{R}^n$ let $\mathbf{z}_{[k]}$ denote the vector in $\mathbb{R}^n$ defined in such a way that the ith component $(\mathbf{z}_{[k]})_i = z_i$ if $i \in \mathcal{I}_k$, 0 otherwise. Let $\mathbf{X}_{[k]} \in \mathbb{R}^{d \times n}$ denote the matrix consisting of the columns of the $\mathbf{X} \in \mathbb{R}^{d \times n}$ indexed by $\mathcal{I}_k$ and replaced with zeros in all other columns.

Ideally, the dual problem solved by node k is (1.2) with $\mathbf{X}, \boldsymbol{\alpha}$ replaced by $\mathbf{X}_{[k]}, \boldsymbol{\alpha}_{[k]}$, respectively, and hence is independent of other nodes. However, following the efficient practical implementation idea in [80, 39], we let the workers communicate among them a vector $\mathbf{v} \in \mathbb{R}^d$, an estimate of $\mathbf{w}(\boldsymbol{\alpha}) = \frac{1}{\lambda n}\mathbf{X}\boldsymbol{\alpha}$ that summarizes the last known global solution $\boldsymbol{\alpha}$. Also following [80, 39] for faster convergence, each worker in our algorithm solves the following perturbed local dual problem, which we henceforth call

the *subproblem*:

$$\max_{\boldsymbol{\delta}_{[k]} \in \mathbb{R}^n} Q_k^{\sigma}(\boldsymbol{\delta}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]}) := -\frac{1}{n_k} \sum_{i \in \mathcal{I}_k} \phi^*(-\alpha_i - \delta_i) - \frac{\lambda}{S} g^*(\mathbf{v}) - \langle \frac{1}{n} X_{[k]}^{\top} \nabla g^*(\mathbf{v}), \boldsymbol{\delta}_{[k]} \rangle - \frac{\lambda \sigma}{2} \left\| \frac{1}{\lambda n} \mathbf{X}_{[k]} \boldsymbol{\delta}_{[k]} \right\|^2 \tag{3.1}$$

where $\boldsymbol{\delta}_{[k]}$ denotes the local (incremental) update to the dual variable $\boldsymbol{\alpha}_{[k]}$ and the *scaling parameter* σ measures the difficulty of solving the given data partition (see [80, 39]) and must be chosen such that

$$\sigma \geq \sigma_{min} := \nu \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{\|\mathbf{X}\boldsymbol{\alpha}\|^2}{\sum_{k=1}^{K} \left\|\mathbf{X}\boldsymbol{\alpha}_{[k]}\right\|^2} \tag{3.2}$$

where the *aggregation parameter* $\nu \in [\frac{1}{S}, 1]$ is the weight given by the master to each of local updates from the contributing workers while computing the global update. The second term in the objective of our subproblem has denominator S instead of K. Unlike the synchronous all reduce approach in [39], our asynchronous method merges the local updates from only S out of K nodes in each global update. By Lemma 3.2 in [39], $\sigma := \nu S$ is a safe choice to hold condition (3.2).

3.1 Asynchronous updates by cores in a worker node

In each communication round, each worker k solves its subproblem using a parallel asynchronous DCA method [28] on the R cores. Let the data partition $\mathcal{I}_k$ stored in the shared memory be logically divided into R subparts where subpart $\mathcal{I}_{k,r} \subseteq \mathcal{I}_k, r = 1, \ldots, R$, is exclusively used by core r. In each of the H iterations, core r chooses a random coordinate $i \in \mathcal{I}_{k,r}$ and updates $\boldsymbol{\delta}_{[k]}$ in the ith unit direction by a step size ε

Algorithm 4: Hybrid-SDCA: Worker node k

Input: Initial $\boldsymbol{\alpha}_{[k]} \in \mathbb{R}^n$, data partition I_k,
scaling parameter σ, aggregation parameter ν

$\mathbf{v} \leftarrow \frac{1}{\lambda n} X \boldsymbol{\alpha}_{[k]}$;
for $t \leftarrow 0, 1, \ldots$ **do**
  $\boldsymbol{\delta}_{[k]} \leftarrow \mathbf{0}, \quad \mathbf{v}_{old} \leftarrow \mathbf{v}$;
  for *cores* $r \leftarrow 1, \ldots, R$ ***in parallel*** **do**
    for $h \leftarrow 0, 1, \ldots, H-1$ **do**
      Randomly pick i from $I_{k,r}$;
      $\varepsilon \leftarrow \text{argmax}_{\varepsilon} \, Q_k^{\sigma}(\varepsilon \mathbf{e}_i; \mathbf{v}, \boldsymbol{\alpha}_{[k]} + \boldsymbol{\delta}_{[k]})$;
      $\boldsymbol{\delta}_{[k]} \leftarrow \boldsymbol{\delta}_{[k]} + \varepsilon \mathbf{e}_i$;
      $\mathbf{v} \xleftarrow{atomic} \mathbf{v} + \nabla g^* \left(\frac{1}{\lambda n} X \varepsilon \mathbf{e}_i \right)$;
  send $\Delta \mathbf{v} \leftarrow \mathbf{v} - \mathbf{v}_{old}$ to the master;
  receive $\mathbf{v}$ from the master;
  $\boldsymbol{\alpha}_{[k]} \leftarrow \boldsymbol{\alpha}_{[k]} + \nu \boldsymbol{\delta}_{[k]}$;

computed using a single variable optimization problem:

$$\varepsilon = \underset{\varepsilon \in \mathbb{R}}{\text{argmax}} \, Q_k^{\sigma}(\varepsilon \mathbf{e}_i; \mathbf{v}, \boldsymbol{\alpha}_{[k]} + \boldsymbol{\delta}_{[k]}) \tag{3.3}$$

which has a closed form solution for SVM problems [14], and a solution using an iterative solver for logistic regression problems [81]. The local updates to $\mathbf{v}$ are also maintained appropriately. While the coordinates used by any two cores and hence the corresponding updates to $\boldsymbol{\delta}_{[k]}$ are independent of each other, there might be conflicts in the updates to $\mathbf{v}$ if the corresponding columns in X have nonzero values in some common row. We use lock-free *atomic* memory updates to handle such conflicts. When all cores complete H iterations, worker k sends the accumulated update $\Delta \mathbf{v}$ from the current round to the master; waits until it receives the globally updated $\mathbf{v}$ from the master; and repeats for another round unless the master indicates termination.

Algorithm 5: Hybrid-SDCA: Master node

Input: Initial $\boldsymbol{\alpha} \in \mathbb{R}^n$, aggregation parameter ν, barrier bound S, delay bound Γ

$\mathbf{v}^{(0)} \leftarrow \frac{1}{\lambda n}\mathbf{X}\boldsymbol{\alpha}$; $\quad \mathcal{P} = \emptyset$;
for $t \leftarrow 0, 1, \ldots$ **do**
  while $|\mathcal{P}| < S$ **or** $\max_k \Gamma_k > \Gamma$ **do**
    receive update $\Delta\mathbf{v}_k$ from some worker k;
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{k\}$; $\quad \Gamma_k \leftarrow 1$;
  $\mathcal{P}_S^{(t)} \leftarrow S$ workers in $\mathcal{P}$ with oldest updates;
  $\mathbf{v}^{(t+1)} \leftarrow \mathbf{v}^{(t)} + \nu \sum_{k \in \mathcal{P}_S^{(t)}} \Delta\mathbf{v}_k$; $\quad \mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{P}_S^{(t)}$;
  foreach $k \notin \mathcal{P}_S^{(t)}$ **do** $\Gamma_k \leftarrow \Gamma_k + 1$;
  broadcast $\mathbf{v}^{(t+1)}$ to all workers in $\mathcal{P}_S^{(t)}$;

3.2 Merging updates from workers by master

If the master had to wait for the updates from all the workers, it could compute the global updates only after the slowest worker finished. To avoid this problem, we use *bounded barrier*: in each round, the master waits for updates from only a subset $\mathcal{P}_S$ of $S \leq K$ workers, and sends them back the global update $\mathbf{v} = \mathbf{v} + \nu \sum_{k \in \mathcal{P}_S} \Delta\mathbf{v}_k$. However, due to this relaxation, there might be some slow workers with out-of-date $\mathbf{v}$. When updates from such workers are merged by the master, it may degrade the quality of the global solution and hence may cause slow convergence or even divergence. We ensure sufficient freshness of the updates using *bounded delay*: the master makes sure that no worker has a stale update older than Γ rounds. This asynchronous approach has two benefits: 1) the overall progress is no more bottlenecked by the slowest processor, and 2) the total number of communications is reduced. On the flip side, convergence may get slowed down for very small S or very large Γ.

Example: Figure 3.1 shows a possible sequence of important events in a run of our algorithm on a dataset having $n = 12$ data points in $d = 3$ dimensions using $K = 3$

nodes each having $R = 2$ cores such that each core works with only $|I_{k,r}| = 2$ data points. The activities in solving the subproblem using $H = 1$ local iterations in a round is shown in a rectangular box. For the first subproblem, core 1 and core 2 in worker 1 randomly selects dual coordinates such that the corresponding data points have nonzero entries in the dimensions $\{1, 3\}, \{1, 2, 3\}$, respectively. Each core first reads the entries of $\mathbf{v}$ corresponding to these nonzero data dimensions, and then computes the updates $[0.1, 0, 0.7], [0.15, 0.5, 0.4]$, respectively, and finally applies these updates to $\mathbf{v}$. The atomic memory updates ensure that all the conflicting writes to $\mathbf{v}$, such as v_1 in the first write-cycle, happen completely. At the end of H local iterations by each core, worker 1 sends $\Delta\mathbf{v} = [0.25, 0.5, 1.1]$ to the master, the responsibility of which is shared by one of the 3 nodes, but shown separately in the figure. By this time, the faster workers 2 and 3 already complete 3 rounds. As $S = 2$, the master takes first 2 updates from $\mathcal{P}_S^{(1)} = \mathcal{P}_S^{(2)} = \{2, 3\}$ and computes the global updates using $\nu = 1$. However, as $\Gamma = 2$, the master holds back the third updates from workers 2, 3 until the first update from worker 1 reaches master. The subsequent events in the run are omitted in the figure.

4 Convergence Analysis

In this section we prove the convergence of the global solution computed by our hybrid algorithm. For ease we prove for $g(w) = \|w\|^2$; the proof can be similarly extended for other regularizers $g(w)$. The analysis is divided into two parts. First we show that the solution of the subproblem computed by each node locally is indeed not far from the optimum. Using this result on the subproblem, we next show the convergence of

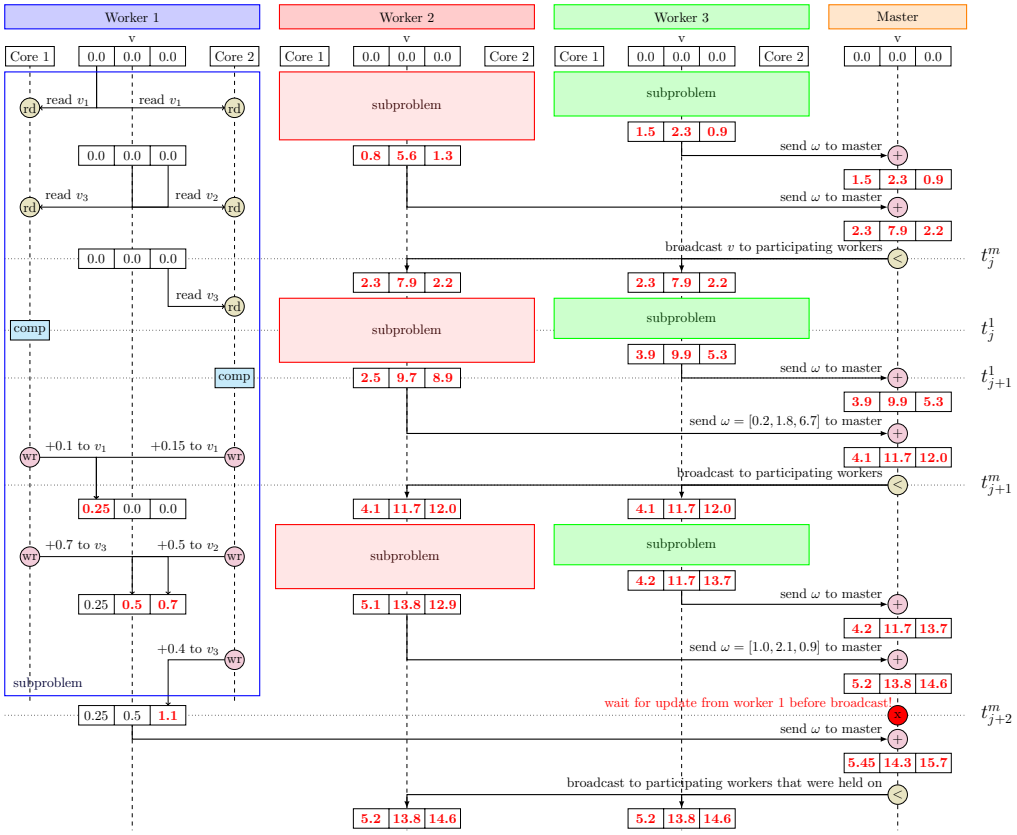


Figure 3.1: Sequence of important events in an example run of Hybrid-SDCA where $n = 12, d = 3, K = 3, R = 2, S = 2, \Gamma = 2, \nu = 1$.

the global solution. Though our proofs for the two parts are based on the works [28] and [38], respectively, we need to make significant adjustments in the proofs due to our modified framework handling two cascaded levels of asynchronous updates.

4.1 Near optimality of the solution to the local subproblem

Definition. For given $\mathbf{v}, \boldsymbol{\alpha}_{[k]}$, a solution $\boldsymbol{\delta}_{[k]}$ to the subproblem (3.1) is said to be Θ-approximate, $\Theta \in [0, 1)$, if

$$\mathbb{E}\left[Q_k^\sigma(\boldsymbol{\delta}_{[k]}^*; \mathbf{v}, \boldsymbol{\alpha}_{[k]}) - Q_k^\sigma(\boldsymbol{\delta}_{[k]}; \mathbf{v}, \boldsymbol{\alpha}_{[k]})\right] \leq \Theta\left[Q_k^\sigma(\boldsymbol{\delta}_{[k]}^*; \mathbf{v}, \boldsymbol{\alpha}_{[k]}) - Q_k^\sigma(\mathbf{0}; \mathbf{v}, \boldsymbol{\alpha}_{[k]})\right] \quad (4.1)$$

where $\boldsymbol{\delta}_{[k]}^*$ is the optimum solution to (3.1).

The main challenge in using the results of [28] to prove (4.1) for the solution returned by the parallel asynchronous stochastic DCA solver used by each worker in Algorithm 4 is to tackle the following two modifications in our approach: 1) the solver here solves only a part of the dual problem and 2) the subproblem is now perturbed (see Section 3). While the first modification is simply handled by considering the updates by the cores in worker k only, the second modification needs changes in each step of the proof in [28], including the definition of the proximal operators T_i defined below.

We consider the updates made in the current round by all the cores in the ascending order of the actual time point (U_j^k in Figure 3.1) when the step size ε of the update is computed (breaking ties arbitrarily) and prove (4.1) by showing sufficient progress in between two successive updates in this order, however, under some assumptions similar to those used in [28].

For all $i \in \mathcal{I}_k$, we have following definitions:

$$
\begin{aligned}
h_i(u) :=& \frac{\phi_i^*(-u)}{n \left\|\mathbf{x}_i\right\|^2} + \frac{\lambda}{2}\left(\frac{1}{S} - \frac{1}{\sigma}\right)\frac{\left\|\mathbf{w}\right\|^2}{\left\|\mathbf{x}_i\right\|^2} \\
\operatorname{prox}_i(s) :=& \operatorname*{argmin}_u \frac{1}{2}(u-s)^2 + h_i(u) \\
T_i(\mathbf{w}, s) :=& \operatorname*{argmax}_u -\frac{1}{\sigma}\frac{\lambda}{2}\left\|\mathbf{w}\right\|^2 - \frac{1}{n}\mathbf{w}^\top \mathbf{x}_i(u-s) - \frac{\lambda}{2}\sigma(\frac{1}{\lambda n}\mathbf{x}_i(u-s))^2 \\
& - \frac{1}{n}\phi_i^*(-u) - \frac{\lambda}{2}\left(\frac{1}{K}\left\|\mathbf{w}\right\|^2 - \frac{1}{\sigma}\left\|\mathbf{w}\right\|^2\right) \\
=& \operatorname*{argmax}_u -\frac{\lambda}{2}\left\|\frac{\mathbf{w}}{\sqrt{\sigma}} + \frac{\sqrt{\sigma}}{\lambda n}(u-s)\mathbf{x}_i\right\|^2 - \left\|\mathbf{x}_i\right\|^2 h_i(u) \\
=& \operatorname*{argmin}_u \frac{1}{2}\left(u - \left(s - \frac{\lambda n \mathbf{w}^\top \mathbf{x}_i}{\sigma\left\|\mathbf{x}_i\right\|^2}\right)\right)^2 + h_i(u),
\end{aligned}
$$

where $\mathbf{w} \in \mathbb{R}^d$ denotes any fixed vector and $s \in \mathbb{R}$. $\operatorname{prox}(s)$ denotes the proximal operator. We can see the connection of above operator and proximal operator: $T_i(\mathbf{w}, s) = \operatorname{prox}_i\left(s - \frac{\mathbf{w}^\top \mathbf{x}_i}{\sigma\|\mathbf{x}_i\|^2}\right)$. Here both $h_i(u)$ and $T_i(\wedge, s)$ were revised from [28] to satisfy the subproblem 3.1.

Definition. First, we define that:

$$
\begin{aligned}
\beta_t^{l+1} =& \begin{cases} T_t\left(\hat{\mathbf{w}}^l, \beta_t^l\right) & \text{if } t = i(l), \\ \beta_t^l & \text{if } t \neq i(l) \end{cases}, & \varepsilon^l =& \beta_{i(l)}^{l+1} - \beta_{i(l)}^l, \\
\tilde{\beta}^{l+1} =& T\left(\hat{\mathbf{w}}^l, \boldsymbol{\beta}^l\right), & \bar{\beta}^{l+1} =& T\left(\bar{\mathbf{w}}^l, \boldsymbol{\beta}^l\right),
\end{aligned}
$$

where $\boldsymbol{\beta}^l$ denotes the sequence generated by the *local atomic solver* and $\hat{\mathbf{w}}^l$ denotes the actual values of w maintained at update l in the *local atomic solver*. Note that, $\tilde{\beta}_{i(l)}^{l+1} = \beta_{i(l)}^{l+1}$ and $\tilde{\beta}^{l+1} = \operatorname{prox}\left(\boldsymbol{\beta}^l - \frac{\lambda n}{\sigma}\bar{\mathbf{X}}\hat{\mathbf{w}}^l\right)$.

The following propositions are similar to [28]. We keep the conclusions of those propositions for the future use in our proof.

Proposition 6.

$$\mathbb{E}_{i(l)}\left(\left\|\boldsymbol{\beta}^{l+1}-\boldsymbol{\beta}^{l}\right\|^{2}\right)=\frac{1}{n}\left\|\tilde{\boldsymbol{\beta}}^{l+1}-\boldsymbol{\beta}^{l}\right\|^{2}, \tag{4.2}$$

Proposition 7.

$$\left\|\bar{\mathbf{X}}\bar{\mathbf{w}}^{j}-\bar{\mathbf{X}}\hat{\mathbf{w}}^{j}\right\|\leq\frac{1}{\lambda n}M\sum_{t=j-\gamma}^{j-1}|\varepsilon^{t}|, \tag{4.3}$$

Proposition 8.

$$\left|T_{i}\left(\mathbf{w}_{1},s_{1}\right)-T_{i}\left(\mathbf{w}_{2},s_{2}\right)\right|\leq\left|s_{1}-s2+\frac{\left(\mathbf{w}_{1}-\mathbf{w}_{2}\right)^{\top}\mathbf{x}_{i}}{\left\|\mathbf{x}_{i}\right\|^{2}}\right|, \tag{4.4}$$

Proposition 9. *Let* $M\geq 1$, $q=\frac{6(\gamma+1)eM}{\sqrt{n}}$, $\rho=(1+q)^{2}$, *and* $\theta=\sum_{t=1}^{\gamma}\rho^{t/2}$. *If* $q(\gamma+1)\leq 1$ *and* $\sigma\geq 1$, *then* $\rho^{(\gamma+1)/2}\leq e$, *and*

$$\rho^{-1}\leq 1-\frac{4}{\sqrt{n}}-\frac{4M+4M\theta}{\sqrt{n}}\leq 1-\frac{4}{\sqrt{n}}-\frac{4M+4M\theta}{\sigma\sqrt{n}}, \tag{4.5}$$

Proposition 10. *For all* $j>0$, *we have*

$$D\left(\boldsymbol{\alpha}^{j}\right)\leq D\left(\bar{\boldsymbol{\alpha}}^{j+1}\right)-\frac{\sigma\left\|\mathbf{x}_{i(j)}\right\|^{2}}{2}\left\|\boldsymbol{\alpha}^{j}-\bar{\boldsymbol{\alpha}}^{j+1}\right\|^{2}, \tag{4.6}$$

$$D\left(\boldsymbol{\alpha}^{j}\right)\geq D\left(\bar{\boldsymbol{\alpha}}^{j+1}\right)-\frac{L_{max}}{2}\left\|\boldsymbol{\alpha}^{j}-\bar{\boldsymbol{\alpha}}^{j+1}\right\|^{2} \tag{4.7}$$

Because of the atomic updates, the step size computation may not include all the latest updates, however, we assume all the updates before the $(j-\gamma)$-th update have already been written into $\mathbf{v}$. Let $\bar{\mathbf{X}}_{[k]}$ denote the normalized data matrix where each row is $\bar{\mathbf{x}}_{i}^{\top}=\mathbf{x}_{i}^{\top}/\|\mathbf{x}_{i}\|, i\in\mathcal{I}_{k}$. Define $M_{[k],i}=\max_{\mathcal{D}\subseteq[d]}\|\sum_{t\in\mathcal{D}}\bar{\mathbf{X}}_{[k](:,t)}X_{[k](i,t)}\|$,

$M = \max_k \max_i M_{[k],i}$, where $[d]$ is the set of all the feature indices, and $\bar{\mathbf{X}}_{[k](:,t)}$ is the t-th column of $\bar{\mathbf{X}}_{[k]}$. Moreover, R_{min} is defined as the minimum value of global data matrix, *i.e.,* $R_{min} = \min_{i=1,\ldots,n} \|\mathbf{x}_i\|^2$.

Assumption 1 (Bounded delay of updates γ)**.**

$$(\gamma + 1)^2 \leq \frac{\sqrt{n_k}}{6eM}, \text{ where } e \text{ is the Euler's number.} \tag{4.8}$$

Lemma 3. *Under Assumption 1, and let* $\boldsymbol{\beta}^l_{[k]} = \boldsymbol{\alpha}_{[k]} + \nu\boldsymbol{\delta}^l_{[k]}$, $\rho = \left(1 + \frac{6(\gamma+1)eM}{\sqrt{n_k}}\right)^2$. *Then, the local subproblem satisfy:*

$$\mathbb{E}\left[\left\|\boldsymbol{\beta}^{l-1}_{[k]} - \tilde{\boldsymbol{\beta}}^{l}_{[k]}\right\|^2\right] \leq \rho\mathbb{E}\left[\left\|\boldsymbol{\beta}^{l}_{[k]} - \tilde{\boldsymbol{\beta}}^{l+1}_{[k]}\right\|^2\right], \tag{4.9}$$

where $l \neq h$, *represents the* l*-th update to* ω *in a* local solver*.*

Proof. We omit the subscript $_{[k]}$ of the notations, which specifies the k-th data partition, in the proof. We prove Eq (4.9) by induction. As shown in [28], we have

$$\left\|\boldsymbol{\beta}^{l-1} - \tilde{\boldsymbol{\beta}}^{l}\right\|^2 - \left\|\boldsymbol{\beta}^{l} - \tilde{\boldsymbol{\beta}}^{l+1}\right\|^2 \leq 2\left\|\boldsymbol{\beta}^{l-1} - \tilde{\boldsymbol{\beta}}^{l}\right\|\left\|\boldsymbol{\beta}^{l} - \tilde{\boldsymbol{\beta}}^{l+1} - \boldsymbol{\beta}^{l-1} + \tilde{\boldsymbol{\beta}}^{l}\right\|. \tag{4.10}$$

The second of factor in the r.h.s of Eq 4.10 is bounded as follows with the revisions:

$$
\begin{aligned}
&\left\|\boldsymbol{\beta}^l-\tilde{\boldsymbol{\beta}}^{l+1}-\boldsymbol{\beta}^{l-1}+\tilde{\boldsymbol{\beta}}^l\right\| \\
\leq&\left\|\boldsymbol{\beta}^l-\boldsymbol{\beta}^{l-1}\right\|+\left\|\operatorname{prox}\left(\boldsymbol{\beta}^l-\frac{\lambda n}{\sigma}\bar{\mathbf{X}}\hat{\mathbf{w}}^l\right)-\operatorname{prox}\left(\boldsymbol{\beta}^{l-1}-\frac{\lambda n}{\sigma}\bar{\mathbf{X}}\hat{\mathbf{w}}^{l-1}\right)\right\| \\
\leq&\left\|\boldsymbol{\beta}^l-\boldsymbol{\beta}^{l-1}\right\|+\left\|\left(\boldsymbol{\beta}^l-\frac{\lambda n}{\sigma}\bar{\mathbf{X}}\hat{\mathbf{w}}^l\right)-\left(\boldsymbol{\beta}^{l-1}-\frac{\lambda n}{\sigma}\bar{\mathbf{X}}\hat{\mathbf{w}}^{l-1}\right)\right\| \\
\leq&2\left\|\boldsymbol{\beta}^l-\boldsymbol{\beta}^{l-1}\right\|+\frac{\lambda n}{\sigma}\left\|\bar{\mathbf{X}}\hat{\mathbf{w}}^l-\bar{\mathbf{X}}\hat{\mathbf{w}}^{l-1}\right\| \\
=&2\left\|\boldsymbol{\beta}^l-\boldsymbol{\beta}^{l-1}\right\|+\frac{\lambda n}{\sigma}\left\|\bar{\mathbf{X}}\hat{\mathbf{w}}^l-\bar{\mathbf{X}}\bar{\mathbf{w}}^l+\bar{\mathbf{X}}\bar{\mathbf{w}}^l-\bar{\mathbf{X}}\bar{\mathbf{w}}^{l-1}+\bar{\mathbf{X}}\bar{\mathbf{w}}^{l-1}-\bar{\mathbf{X}}\hat{\mathbf{w}}^{l-1}\right\| \\
\leq&2\left\|\boldsymbol{\beta}^l-\boldsymbol{\beta}^{l-1}\right\|+\frac{\lambda n}{\sigma}\left(\left\|\bar{\mathbf{X}}\bar{\mathbf{w}}^l-\bar{\mathbf{X}}\bar{\mathbf{w}}^{l-1}\right\|+\left\|\bar{\mathbf{X}}\hat{\mathbf{w}}^l-\bar{\mathbf{X}}\bar{\mathbf{w}}^l\right\|+\left\|\bar{\mathbf{X}}\bar{\mathbf{w}}^{l-1}-\bar{\mathbf{X}}\hat{\mathbf{w}}^{l-1}\right\|\right) \\
\leq&\left(2+2\frac{\lambda n}{\sigma}\frac{M}{\lambda n}\right)\left\|\boldsymbol{\beta}^l-\boldsymbol{\beta}^{l-1}\right\|+2\frac{\lambda n}{\sigma}\frac{M}{\lambda n}\sum_{t=l-\gamma-1}^{l-2}|\varepsilon^t| \quad \text{(Proposition 7)} \qquad (4.11) \\
\leq&\left(2+2\frac{M}{\sigma}\right)\left\|\boldsymbol{\beta}^l-\boldsymbol{\beta}^{l-1}\right\|+2\frac{M}{\sigma}\sum_{t=l-\gamma-1}^{l-2}|\varepsilon^t| \qquad (4.12)
\end{aligned}
$$

No we start the induction. Although some steps may be the same as the steps in [28], we still keep them here to make the proof self-contained.

Induction Hypothesis. We prove the following equivalent statement. For all j,

$$
\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^l\right\|^2\right)\leq\rho\mathbb{E}\left(\left\|\boldsymbol{\beta}^l-\tilde{\boldsymbol{\beta}}^{l+1}\right\|^2\right),
$$

Induction Basis. When $l=1$,

$$
\begin{aligned}
\mathbb{E}\left(\left\|\boldsymbol{\beta}^0-\tilde{\boldsymbol{\beta}}^1\right\|^2\right)-\mathbb{E}\left(\left\|\boldsymbol{\beta}^1-\tilde{\boldsymbol{\beta}}^2\right\|^2\right)\leq&2E\left(\left\|\boldsymbol{\beta}^0-\tilde{\boldsymbol{\beta}}^1\right\|\left\|\boldsymbol{\beta}^1-\tilde{\boldsymbol{\beta}}^2-\boldsymbol{\beta}^0+\tilde{\boldsymbol{\beta}}^1\right\|\right) \\
\leq&\left(4+4\frac{M}{2}\right)\mathbb{E}(\left\|\boldsymbol{\beta}^0-\tilde{\boldsymbol{\beta}}^1\right\|\left\|\boldsymbol{\beta}^0-\boldsymbol{\beta}^1\right\|).
\end{aligned}
$$

By Proposition 6 and AM-GM inequality, which for any $b_1,b_2>0$ and any $c>0$, we

have

$$b_1 b_2 \leq \frac{1}{2}\left(c b_1^2 + c^{-1} b_2^2\right) \tag{4.13}$$

Therefore, we have

$$\begin{aligned}
\mathbb{E}\left(\left\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^1\right\| \left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^1\right\|\right) \leq & \frac{1}{2}\mathbb{E}\left(\sqrt{n}\left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^1\right\|^2 + \frac{1}{\sqrt{n}}\left\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^1\right\|^2\right) \\
= & \frac{1}{2}\mathbb{E}\left(\frac{1}{\sqrt{n}}\left\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^1\right\|^2 + \frac{1}{\sqrt{n}}\left\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^1\right\|^2\right) \quad \text{(Proposition 6)} \\
= & \frac{1}{\sqrt{n}}\mathbb{E}\left(\left\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^1\right\|^2\right)
\end{aligned}$$

Therefore,

$$\mathbb{E}\left(\left\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^1\right\|^2\right) - \mathbb{E}\left(\left\|\boldsymbol{\beta}^1 - \tilde{\boldsymbol{\beta}}^2\right\|^2\right) \leq \left(\frac{4}{\sqrt{n}} + \frac{4M}{\sigma\sqrt{n}}\right)\mathbb{E}\left(\left\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^1\right\|^2\right),$$

which implies

$$\mathbb{E}\left(\left\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^1\right\|^2\right) \leq \left(1 - \frac{4}{\sqrt{n}} - \frac{4M}{\sigma\sqrt{n}}\right)^{-1}\mathbb{E}\left(\left\|\boldsymbol{\beta}^1 - \tilde{\boldsymbol{\beta}}^2\right\|^2\right) \leq \rho\mathbb{E}\left(\left\|\boldsymbol{\beta}^1 - \tilde{\boldsymbol{\beta}}^2\right\|^2\right),$$

where the last inequality is based on Proposition 9 and the fact $\theta M \geq 1$.

Induction Step. By the induction hypothesis, we assume

$$\mathbb{E}\left(\left\|\boldsymbol{\beta}^{t-1} - \tilde{\boldsymbol{\beta}}^t\right\|^2\right) \leq \rho\mathbb{E}\left(\left\|\boldsymbol{\beta}^t - \tilde{\boldsymbol{\beta}}^{t+1}\right\|^2\right) \ \forall t \leq l-1. \tag{4.14}$$

To show

$$\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1} - \tilde{\boldsymbol{\beta}}^l\right\|^2\right) \leq \rho\mathbb{E}\left(\left\|\boldsymbol{\beta}^l - \tilde{\boldsymbol{\beta}}^{l+1}\right\|^2\right),$$

we firstly show that for all $t < j$,

$$
\begin{aligned}
&\mathbb{E}\left(\left\|\boldsymbol{\beta}^{t}-\boldsymbol{\beta}^{t+1}\right\|\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|\right)\\
\leq&\frac{1}{2}\mathbb{E}\left(\sqrt{n}\rho^{(t+1-l)/2}\left\|\boldsymbol{\beta}^{t}-\boldsymbol{\beta}^{t+1}\right\|^{2}+\frac{1}{\sqrt{n}}\rho^{(l-1-t)/2}\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right) \quad \text{(Eq. 4.13)}\\
=&\frac{1}{2}\mathbb{E}\left(\sqrt{n}\rho^{(t+1-l)/2}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{t}-\boldsymbol{\beta}^{t+1}\right\|^{2}\right)+\frac{1}{\sqrt{n}}\rho^{(l-1-t)/2}\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right)\\
=&\frac{1}{2}\mathbb{E}\left(\frac{1}{\sqrt{n}}\rho^{(t+1-l)/2}\left\|\boldsymbol{\beta}^{t}-\tilde{\boldsymbol{\beta}}^{t+1}\right\|^{2}+\frac{1}{\sqrt{n}}\rho^{(l-1-t)/2}\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right) \quad \text{(Proposition 6)}\\
\leq&\frac{1}{2}\mathbb{E}\left(\frac{1}{\sqrt{n}}\rho^{(t+1-l)/2}\rho^{l-t-1}\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}+\frac{1}{\sqrt{n}}\rho^{(l-1-t)/2}\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right) \quad \text{(Eq. 4.14)}\\
\leq&\frac{\rho^{(l-1-t)/2}}{\sqrt{n}}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right). \qquad (4.15)
\end{aligned}
$$

Let $\theta = \sum_{t=1}^{\gamma}\rho^{t/2}$. We have

$$
\begin{aligned}
&\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right)-\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l}-\tilde{\boldsymbol{\beta}}^{l+1}\right\|\right)\\
\leq&\mathbb{E}\left(2\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|\left(\left(2+2\frac{M}{\sigma}\right)\left\|\boldsymbol{\beta}^{l-1}-\boldsymbol{\beta}^{l}\right\|+2\frac{M}{\sigma}\left\|\boldsymbol{\beta}^{t-1}-\boldsymbol{\beta}^{t}\right\|\right)\right) \quad \text{(Eq. 4.10, Eq. 4.11)}\\
=&\left(4+4\frac{M}{\sigma}\right)\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|\left\|\boldsymbol{\beta}^{l-1}-\boldsymbol{\beta}^{l}\right\|\right)+4\frac{M}{\sigma}\sum_{t=l-\gamma-1}^{l-1}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|\left\|\boldsymbol{\beta}^{t-1}-\boldsymbol{\beta}^{t}\right\|\right)\\
\leq&\frac{4\sigma+4M}{\sigma\sqrt{n}}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right)+\frac{4M}{\sigma\sqrt{n}}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|\right)\sum_{t=l-\gamma-1}^{l-2}\rho^{(l-1-t)/2} \quad \text{(Eq. 4.15)}\\
\leq&\frac{4\sigma+4M}{\sigma\sqrt{n}}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right)+\frac{4M}{\sigma\sqrt{n}}\theta\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|\right)\\
\leq&\left(\frac{4}{\sqrt{n}}+\frac{4M+4M\theta}{\sigma\sqrt{n}}\right)\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right)
\end{aligned}
$$

which implies that

$$
\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right)\leq\frac{1}{1-\frac{4}{\sqrt{n}}-\frac{4M+4M\theta}{\sigma\sqrt{n}}}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l}-\tilde{\boldsymbol{\beta}}^{l+1}\right\|^{2}\right)\leq\rho\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l-1}-\tilde{\boldsymbol{\beta}}^{l}\right\|^{2}\right)
$$

by Proposition 9. □

Definition (Global error bound)**.** For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, the optimization problem: $\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$ admits a global error bound if there is a constant κ such that

$$\|\boldsymbol{\beta} - P_S(\boldsymbol{\beta})\| \leq \kappa \|T(\boldsymbol{\beta}) - \boldsymbol{\beta}\|, \tag{4.16}$$

where $P_S(\cdot)$ is the Euclidean projection to the set of optimal solutions, and $T : \mathbb{R}^n \to \mathbb{R}^n$ is the operator defined as

$$T_i(\boldsymbol{\beta}) = \arg\min_u f\left(\boldsymbol{\beta} + (u - \beta_i)\mathbf{e}_t\right) \quad \forall i \in [n].$$

The optimization problem admits a relaxed condition called global error bound from the beginning if (4.16) holds for any $\boldsymbol{\beta}$ satisfying $f(\boldsymbol{\beta}) \leq F$ for some constant F.

Assumption 2. The local subproblem formulation (3.1) admits global error bound from the beginning for $F = Q(\boldsymbol{\delta}_{[k]}^{(j)}; \mathbf{v}^{(j)}, \boldsymbol{\alpha}_{[k]}^{(j)})$ and any update j.

The global error bound helps prove that our subproblem solver achieves significant improvement after each update. It has been shown that when the loss functions are hinge loss or squared hinge loss, the local subproblem formulation (3.1) does indeed satisfy global error bound condition [28].

Assumption 3. The local subproblem objective (3.1) is L_{max}-Lipschitz continuous.

Assumption 4 (Bounded M, L_{max})**.**

$$2L_{max}\left(1 + \frac{e^2\gamma^2M^2}{\sigma^2 n_k}\right)\left(\frac{e^2\gamma^2M^2}{\sigma^2 n_k}\right) \leq 1$$

Lemma 4. *When Assumptions 1-4 hold, the solutions computed in two successive updates by the* local subproblem *solver has a linear convergence rate in expectation,* i.e.,

$$\mathbb{E}\left[Q_k^\sigma(\boldsymbol{\delta}_{[k]}^*) - Q_k^\sigma(\boldsymbol{\delta}_{[k]}^{(j)})\right] \leq \eta \left[Q_k^\sigma(\boldsymbol{\delta}_{[k]}^*) - Q_k^\sigma(\boldsymbol{\delta}_{[k]}^{(j-1)})\right]$$

where $\boldsymbol{\delta}_{[k]}^{(j)}$ *is the* $\boldsymbol{\delta}_{[k]}$ *after the* j*th update,*

$$\eta = 1 - \frac{\kappa R_{min}}{2nL_{max}}\left(1 - \frac{2L_{max}}{R_{min}}\left(1 + \frac{e^2\gamma^2M^2}{\sigma^2\tilde{n}}\right)\left(\frac{e^2\gamma^2M^2}{\sigma^2\tilde{n}}\right)\right),$$

and $\tilde{n} = \max_k n_k$ *is the size of the largest data part. Moreover,* $\boldsymbol{\delta}_{[k]}^{(H)}$ *is a* Θ*-approximate solution for*

$$\Theta = \eta^H. \tag{4.17}$$

Proof. We also omit the subscript $_{[k]}$ of the notations in the proof. We can bound the

expected distance $\mathbb{E}\left(\left\|\bar{\boldsymbol{\beta}}^{j+1}-\tilde{\boldsymbol{\beta}}^{l+1}\right\|^2\right)$ by the following derivation.

$$
\begin{aligned}
\mathbb{E}\left(\left\|\bar{\boldsymbol{\beta}}^{l+1}-\tilde{\boldsymbol{\beta}}^{l+1}\right\|^2\right) =& \mathbb{E}\left(\sum_{t=1}^{n}\left(T_t(\bar{\mathbf{w}}^l,\beta_t^l)-T_t(\hat{\mathbf{w}}^l,\beta_t^l)\right)^2\right)\\
\leq& \mathbb{E}\left(\sum_{t=1}^{n}\left(\frac{\lambda n\left(\bar{\mathbf{w}}^l-\hat{\mathbf{w}}^l\right)^\top \mathbf{x}_t}{\sigma\left\|\mathbf{x}_t\right\|^2}\right)^2\right) \quad \text{(Proposition 8)}\\
=& \frac{\lambda^2 n^2}{\sigma^2}\mathbb{E}\left(\left\|\bar{\mathbf{X}}\left(\bar{\mathbf{w}}^l-\hat{\mathbf{w}}^l\right)\right\|^2\right)\\
\leq& \frac{M^2}{\lambda^2 n^2}\frac{\lambda^2 n^2}{\sigma^2}\mathbb{E}\left(\left(\sum_{t=l-\gamma}^{l-1}\left\|\boldsymbol{\beta}^t-\boldsymbol{\beta}^{t+1}\right\|\right)^2\right) \quad \text{(Proposition 7)}\\
\leq& \frac{M^2}{\sigma^2}\mathbb{E}\left(\gamma\left(\sum_{t=l-\gamma}^{l-1}\left\|\boldsymbol{\beta}^t-\boldsymbol{\beta}^{t+1}\right\|^2\right)\right) \quad \text{(Cauchy Schwarz Inequality)}\\
\leq& \frac{\gamma M^2}{\sigma^2}\mathbb{E}\left(\gamma\left(\sum_{t=1}^{\gamma}\rho^t\left\|\boldsymbol{\beta}^l-\boldsymbol{\beta}^{l+1}\right\|^2\right)\right) \quad \text{(Lemma 3)}\\
\leq& \frac{\gamma M^2}{\sigma^2 n}\left(\sum_{t=1}^{\gamma}\rho^t\right)\mathbb{E}\left(\left\|\boldsymbol{\beta}^l-\tilde{\boldsymbol{\beta}}^{l+1}\right\|^2\right) \quad \text{(Proposition 6)}\\
\leq& \frac{\gamma^2 M^2}{\sigma^2 n}\rho^\gamma\mathbb{E}\left(\left\|\boldsymbol{\beta}^l-\tilde{\boldsymbol{\beta}}^{l+1}\right\|^2\right)\\
\leq& \frac{\gamma^2 M^2 e^2}{\sigma^2 n}\mathbb{E}\left(\left\|\boldsymbol{\beta}^l-\tilde{\boldsymbol{\beta}}^{l+1}\right\|^2\right). \quad \text{(Proposition 9)} \qquad (4.18)
\end{aligned}
$$

(4.19)

Moreover,

$$
\begin{aligned}
\mathbb{E}\left(\left\|\bar{\boldsymbol{\beta}}^{l}-\boldsymbol{\beta}^{l+1}\right\|^2\right) =& \mathbb{E}\left(\left\|\bar{\boldsymbol{\beta}}^{l+1}-\tilde{\boldsymbol{\beta}}^{l+1}+\tilde{\boldsymbol{\beta}}^{l+1}-\boldsymbol{\beta}^{l}\right\|^2\right)\\
\leq& \mathbb{E}\left(2\left(\left\|\bar{\boldsymbol{\beta}}^{l+1}-\tilde{\boldsymbol{\beta}}^{l+1}\right\|^2+\left\|\tilde{\boldsymbol{\beta}}^{l+1}-\boldsymbol{\beta}^{l}\right\|^2\right)\right) \quad \text{(Cauchy}-\text{Schwarz)}\\
\leq& 2\left(1+\frac{\gamma^2 M^2 e^2}{\sigma^2 n}\right)\mathbb{E}\left(\left\|\tilde{\boldsymbol{\beta}}^{l+1}-\boldsymbol{\beta}^{l}\right\|^2\right) \qquad (4.20)
\end{aligned}
$$

The bound of the increase of local objective function value by

$$
\begin{aligned}
&\mathbb{E}\left(D\left(\boldsymbol{\beta}^{l+1}\right)\right)-\mathbb{E}\left(D\left(\boldsymbol{\beta}^{l}\right)\right)\\
=&\mathbb{E}\left(-\left(D\left(\boldsymbol{\beta}^{l}\right)-D\left(\bar{\boldsymbol{\beta}}^{l+1}\right)\right)\right)-\mathbb{E}\left(\left(D\left(\bar{\boldsymbol{\beta}}^{l+1}\right)-D\left(\boldsymbol{\beta}^{l+1}\right)\right)\right)\\
\geq&\mathbb{E}\left(\frac{\sigma\left\|\mathbf{x}_{i(l)}\right\|^{2}}{2}\left\|\boldsymbol{\beta}^{l}-\bar{\boldsymbol{\beta}}^{l+1}\right\|^{2}\right)-\mathbb{E}\left(\frac{L_{max}}{2}\left\|\boldsymbol{\beta}^{l+1}-\bar{\boldsymbol{\beta}}^{l+1}\right\|^{2}\right)\quad\text{(Proposition 10)}\\
\geq&\frac{R_{min}}{2n}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l}-\bar{\boldsymbol{\beta}}^{l+1}\right\|^{2}\right)-\frac{L_{max}}{2n}\mathbb{E}\left(\left\|\tilde{\boldsymbol{\beta}}^{l+1}-\bar{\boldsymbol{\beta}}^{l+1}\right\|^{2}\right)\\
\geq&\frac{R_{min}}{2n}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l}-\bar{\boldsymbol{\beta}}^{l+1}\right\|^{2}\right)-\frac{L_{max}}{2n}\frac{\gamma^{2}M^{2}e^{2}}{\sigma^{2}n}\mathbb{E}\left(\left\|\tilde{\boldsymbol{\beta}}^{l+1}-\boldsymbol{\beta}^{l}\right\|^{2}\right)\quad\text{(Eq. 4.18)}\\
\geq&\frac{R_{min}}{2n}\mathbb{E}\left(\left\|\boldsymbol{\beta}^{l}-\bar{\boldsymbol{\beta}}^{l+1}\right\|^{2}\right)-\frac{2L_{max}}{2n}\frac{\gamma^{2}M^{2}e^{2}}{\sigma^{2}n}\left(1+\frac{\gamma^{2}M^{2}e^{2}}{\sigma^{2}n}\right)\mathbb{E}\left(\left\|\bar{\boldsymbol{\beta}}^{l+1}-\boldsymbol{\beta}^{l}\right\|^{2}\right)\quad\text{(Eq. 4.20)}\\
\geq&\frac{R_{min}}{2n}\left(1-\frac{2L_{max}}{2n}\left(1+\frac{\gamma^{2}M^{2}e^{2}}{\sigma^{2}n}\right)\left(\frac{\gamma^{2}M^{2}e^{2}}{\sigma^{2}n}\right)\right)\mathbb{E}\left(\left\|\bar{\boldsymbol{\beta}}^{l+1}-\boldsymbol{\beta}^{l}\right\|^{2}\right)\\
\geq&\frac{\kappa R_{min}}{2n}\left(1-\frac{2L_{max}}{2n}\left(1+\frac{\gamma^{2}M^{2}e^{2}}{\sigma^{2}n}\right)\left(\frac{\gamma^{2}M^{2}e^{2}}{\sigma^{2}n}\right)\right)\mathbb{E}\left(\left\|P_{S}\left(\boldsymbol{\beta}^{l}\right)-\boldsymbol{\beta}^{l}\right\|^{2}\right)\\
\geq&\frac{\kappa R_{min}}{2nL_{max}}\left(1-\frac{2L_{max}}{2n}\left(1+\frac{\gamma^{2}M^{2}e^{2}}{\sigma^{2}n}\right)\left(\frac{\gamma^{2}M^{2}e^{2}}{\sigma^{2}n}\right)\right)\mathbb{E}\left(D^{*}-D\left(\boldsymbol{\beta}^{l}\right)\right)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
D^{*}-\mathbb{E}\left(D\left(\boldsymbol{\beta}^{l+1}\right)\right)=&D^{*}-\mathbb{E}\left(D\left(\boldsymbol{\beta}^{l}\right)\right)-\left(\mathbb{E}\left(D\left(\boldsymbol{\beta}^{l+1}\right)-\mathbb{E}\left(D\left(\boldsymbol{\beta}^{l}\right)\right)\right)\right)\\
\leq&\eta\left(D^{*}-\mathbb{E}\left(D\left(\boldsymbol{\beta}^{l}\right)\right)\right)
\end{aligned}
$$

Let us assume that $\beta_{[k]}^{*}$ is the optimal solution of the subproblem 3.1 denotes as:

$$
\beta_{[k]}^{*}=\arg\max_{\beta_{[k]}\in\mathbb{R}^{n_k}}D(\beta_{[k]};\bar{\mathbf{w}}). \tag{4.21}
$$

According to above proof of Lemma 4, the local atomic solver has a linear convergence

rate in expectation, that is,

$$D(\beta^*_{[k]};\bar{\mathbf{w}}) - \mathbb{E}\left(D(\beta^{j+1}_{[k]};\bar{\mathbf{w}})\right) \leq \eta\left(\mathbb{E}\left(D(\beta^*_{[k]};\bar{\mathbf{w}}) - D(\beta^{j}_{[k]};\bar{\mathbf{w}})\right)\right)$$

It is obvious that $\Theta = \eta^H$. Thus, we can easily get the induction as

$$\begin{aligned} &D(\beta^*_{[k]};\bar{\mathbf{w}}) - \mathbb{E}\left(D(\beta^{H}_{[k]};\bar{\mathbf{w}})\right) \leq \eta\left(D(\beta^*_{[k]};\bar{\mathbf{w}}) - \mathbb{E}\left(D(\beta^{H-1}_{[k]};\bar{\mathbf{w}})\right)\right) \\ \leq &\eta^2\left(\mathbb{E}\left(D(\beta^*_{[k]};\bar{\mathbf{w}}) - D(\beta^{H-2}_{[k]};\bar{\mathbf{w}})\right)\right) \leq \cdots \leq \Theta\left(D(\beta^*_{[k]};\bar{\mathbf{w}}) - \mathbb{E}\left(D(\beta^0_{[k]};\bar{\mathbf{w}})\right)\right). \end{aligned}$$

Notice that $\beta^0_{[k]}$ are the start points of the local atomic solver and $\beta^H_{[k]}$ are the final results of $\beta_{[k]}$ of the local atomic solver. So the following equations hold for the global problem:

$$\begin{aligned} \beta^0_{[k]} &= \alpha_{[k]} \\ \beta^H_{[k]} - \beta^0_{[k]} &= \boldsymbol{\delta}_{[k]} \\ \beta^*_{[k]} - \beta^0_{[k]} &= \boldsymbol{\delta}^*_{[k]} \end{aligned}$$

Therefore, we have:

$$\mathbb{E}\left[Q^\sigma_k(\boldsymbol{\delta}^*_{[k]};\mathbf{v},\boldsymbol{\alpha}_{[k]}) - Q^\sigma_k(\boldsymbol{\delta}_{[k]};\mathbf{v},\boldsymbol{\alpha}_{[k]})\right] \leq \Theta\left[Q^\sigma_k(\boldsymbol{\delta}^*_{[k]};\mathbf{v},\boldsymbol{\alpha}_{[k]}) - Q^\sigma_k(\mathbf{0};\mathbf{v},\boldsymbol{\alpha}_{[k]})\right]$$

with $\Theta = \eta^H$. □

4.2 Convergence of global solution

Although we showed that the local subproblem solver outputs a Θ-approximate solution, we cannot directly apply the results of [38] for the global solution because our algorithm uses updates from only a subset of workers which is unlike the synchronous all-reduce of the updates from all workers used in [38]. We need to handle this asynchronous nature of the global updates, just like we handled asynchronous updates for the local subproblem. Let us consider the global updates in the order the master computed them (at time U_t^m in Figure 3.1). Let $\boldsymbol{\alpha}^{(t)}$ denote the value of $\boldsymbol{\alpha}$ distributed across all the nodes at the time master computed tth global update $\mathbf{v}^{(t)}$. If $k \in \mathcal{P}_S^{(t)}$ then the update $\boldsymbol{\delta}_{[k]}^{(t)}$ has already been included in $\mathbf{v}^{(t)}$. However, if $k \notin \mathcal{P}_S^{(t)}$ then it may not be included. Let ξ be such that for all $l \leq \xi$ and for all k, $\boldsymbol{\delta}_{[k]}^{(l)}$ has been included in $\mathbf{v}^{(t)}$. By the design of our algorithm, $t - \Gamma \leq \xi \leq t-1$. Let $\hat{\boldsymbol{\alpha}}^{(t)}$ be defined as follows: $\hat{\boldsymbol{\alpha}}_{[k]}^{(t)} = \boldsymbol{\alpha}_{[k]}^{(t)}, \forall k \in \mathcal{P}_S^{(t)}$ and $= \boldsymbol{\alpha}_{[k]}^{(\xi)}$ for the latest ξ for which the update is already included in global $\mathbf{v}$, $\forall k \notin \mathcal{P}_S^{(t)}$. Let $\mathbf{w}^{(t)}, \hat{\mathbf{w}}^{(t)}$ be $\mathbf{w}(\boldsymbol{\alpha}^{(t)})$ and $\mathbf{w}(\hat{\boldsymbol{\alpha}}^{(t)})$ respectively. Note that $\mathbf{w}^{(t)} = \hat{\mathbf{w}}^{(t)} + \frac{1}{\lambda n}\sum_{l=\xi}^{t-1} \mathbf{X}\boldsymbol{\delta}^{(l)}$.

Lemma 5. *For any dual $\boldsymbol{\alpha}^{(t)}, \boldsymbol{\delta}^{(t)} \in \mathbb{R}^n$, primal $\hat{\mathbf{w}}^{(t)} = \mathbf{w}(\hat{\boldsymbol{\alpha}}^{(t)})$ and real values ν, σ satisfying (3.2), it holds that*

$$D\left(\boldsymbol{\alpha}^{(t)} + \nu \sum_{k \in \mathcal{P}_S^{(t)}} \boldsymbol{\delta}_{[k]}^{(t)}\right) \geq (1-\nu) D(\hat{\boldsymbol{\alpha}}^{(t)}) - \frac{\lambda}{2}(\left\|\mathbf{w}^{(t)}\right\|^2 - \left\|\hat{\mathbf{w}}^{(t)}\right\|^2)$$
$$+\nu \sum_{k \in \mathcal{P}_S^{(t)}} Q_k^{\sigma}\left(\boldsymbol{\delta}_{[k]}^{(t)}; \hat{\mathbf{w}}^{(t)}, \boldsymbol{\alpha}_{[k]}^{(t)}\right) - \frac{\nu}{n}\sum_{k \in \mathcal{P}_S^{(t)}} (\mathbf{w}^{(t)} - \hat{\mathbf{w}}^{(t)})^{\top} \mathbf{X}\boldsymbol{\delta}_{[k]}^{(t)}. \tag{4.22}$$

Proof. Assume that $\mathcal{I} = \bigcup_{k\in\mathcal{P}_S} \mathcal{I}_k$. Then, we have

$$
\begin{aligned}
D\left(\boldsymbol{\alpha} + \nu \sum_{k\in\mathcal{P}_S} \boldsymbol{\delta}_{[k]}\right) =& -\frac{1}{n}\sum_{i=1}^{n} \phi_i^*\left(-\alpha_i - \nu\left(\sum_{k\in\mathcal{P}_S} \boldsymbol{\delta}_{[k]}\right)_i\right) \\
& - \frac{\lambda}{2}\left\|\frac{1}{\lambda n}\mathbf{X}\left(\boldsymbol{\alpha} + \nu\sum_{k\in\mathcal{P}_S}\boldsymbol{\delta}_{[k]}\right)\right\|^2 \\
=& -\frac{1}{n}\sum_{i\notin\mathcal{I}} \phi_i^*(-\alpha_i) - \frac{1}{n}\sum_{k\in\mathcal{P}_S}\left(\sum_{i\in\mathcal{I}_k}\phi_i^*(-(1-\nu)\alpha_i - \nu(\boldsymbol{\alpha}+\boldsymbol{\delta}_{[k]})_i)\right) \\
& - \frac{\lambda}{2}\left(\|\mathbf{w}(\boldsymbol{\alpha})\|^2 + \frac{2\nu}{\lambda n}\sum_{k\in\mathcal{P}_S}\mathbf{w}(\boldsymbol{\alpha})^\top\mathbf{X}\boldsymbol{\delta}_{[k]} + \left(\frac{\nu}{\lambda n}\right)^2\left\|\sum_{k\in\mathcal{P}_S}\mathbf{X}\boldsymbol{\delta}_{[k]}\right\|^2\right) \\
\geq& -\frac{1}{n}\sum_{k\in\mathcal{P}_S}\left(\sum_{i\in\mathcal{I}_k}((1-\nu)\phi_i^*(-\alpha_i) + \nu\phi_i^*(-(\boldsymbol{\alpha}+\boldsymbol{\delta}_{[k]})_i))\right) \\
& - \frac{\lambda}{2}\left(\|\hat{\mathbf{w}}\|^2 + \frac{2\nu}{\lambda n}\sum_{k\in\mathcal{P}_S}\hat{\mathbf{w}}^\top\mathbf{X}\boldsymbol{\delta}_{[k]} + \left(\frac{\nu}{\lambda n}\right)^2\left\|\sum_{k\in\mathcal{P}_S}\mathbf{X}\boldsymbol{\delta}_{[k]}\right\|^2\right) \\
& - \frac{\nu}{n}\sum_{k\in\mathcal{P}_S}(\mathbf{w}-\hat{\mathbf{w}})^\top\mathbf{X}\boldsymbol{\delta}_{[k]} - \frac{\lambda}{2}(\|\mathbf{w}\|^2 - \|\hat{\mathbf{w}}\|^2) - \frac{1}{n}\sum_{i\notin\mathcal{I}}\phi_i^*(-\alpha_i) \\
=& \underbrace{-\frac{1}{n}\sum_{k\in\mathcal{P}_S}\left(\sum_{i\in\mathcal{I}_k}(1-\nu)\phi_i^*(-\alpha_i)\right) - (1-\nu)\frac{\lambda}{2}\|\mathbf{w}(\hat{\boldsymbol{\alpha}})\|^2}_{(1-\nu)D(\hat{\boldsymbol{\alpha}})} \\
& + \nu\sum_{k\in\mathcal{P}_k}\left(-\frac{1}{n}\sum_{i\in\mathcal{I}_k}\phi_i^*(-(\boldsymbol{\alpha}+\boldsymbol{\delta}_{[k]})_i) - \frac{1}{K}\frac{\lambda}{2}\|\mathbf{w}(\hat{\boldsymbol{\alpha}})\|^2\right. \\
& \left. -\frac{1}{n}\mathbf{w}(\hat{\boldsymbol{\alpha}})^\top\mathbf{X}\boldsymbol{\delta}_{[k]} - \frac{\lambda}{2}\sigma\left\|\frac{1}{\lambda n}\mathbf{X}\boldsymbol{\delta}_{[k]}\right\|^2\right) \\
& - \frac{\nu}{n}\sum_{k\in\mathcal{P}_S}(\mathbf{w}-\hat{\mathbf{w}})^\top\mathbf{X}\boldsymbol{\delta}_{[k]} - \frac{\lambda}{2}(\|\mathbf{w}\|^2 - \|\hat{\mathbf{w}}\|^2) - \frac{1}{n}\sum_{i\notin\mathcal{I}}\phi_i^*(-\alpha_i) \\
=& (1-\nu)D(\hat{\boldsymbol{\alpha}}) + \nu\sum_{k\in\mathcal{P}_S} Q_k^\sigma(\boldsymbol{\delta}_{[k]}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}}) \\
& - \frac{\nu}{n}\sum_{k\in\mathcal{P}_S}(\mathbf{w}-\hat{\mathbf{w}})^\top\mathbf{X}\boldsymbol{\delta}_{[k]} - \frac{\lambda}{2}(\|\mathbf{w}\|^2 - \|\hat{\mathbf{w}}\|^2) - \frac{1}{n}\sum_{i\notin\mathcal{I}}\phi_i^*(-\alpha_i).
\end{aligned}
$$

□

Assumption 5. There exists a $\varrho < e^{\frac{2}{\Gamma+1}}$ such that

$$\left\|\boldsymbol{\delta}^{(t-1)}\right\|^2 \leq \varrho \left\|\boldsymbol{\delta}^{(t)}\right\|^2. \tag{4.23}$$

Lemma 6. *If ϕ_i^* are all $(1/\mu)$-strongly convex and Assumptions 1-5 are satisfied then for any $s \in [0,1]$, any round t of Algorithm 5 satisfies*

$$\mathbb{E}[D(\boldsymbol{\alpha}^{(t+1)}) - D(\boldsymbol{\alpha}^{(t)})] \geq \Psi(1-\Theta)\left(sG(\hat{\boldsymbol{\alpha}}) - \frac{\sigma}{2\lambda}\left(\frac{s}{n}\right)^2 \hat{R}\right) \tag{4.24}$$

where

$$\Psi := \nu\left(1 - \frac{\Gamma^2 eML_{max}}{4\lambda n^2} - \frac{SML_{max}}{4n}\right) \leq 1, \text{ and} \tag{4.25}$$

$$\hat{R} := -\frac{\lambda\mu n(1-s)}{\sigma s}\left\|\hat{\mathbf{u}} - \hat{\boldsymbol{\alpha}}\right\|^2 + \sum_{k\in\mathcal{P}_S}\left\|\mathbf{X}(\hat{\mathbf{u}} - \hat{\boldsymbol{\alpha}})_{[k]}\right\|^2, \tag{4.26}$$

for $\hat{\mathbf{u}} \in \mathbb{R}^n$ with $-\hat{u}_i \in \partial\phi_i(\mathbf{w}(\hat{\boldsymbol{\alpha}})^\top \mathbf{x}_i)$.

Proof. For sake of notation, we will write $\boldsymbol{\alpha}$ instead of $\boldsymbol{\alpha}^t$, $\mathbf{w}$ instead of $\mathbf{w}(\boldsymbol{\alpha}^t)$, $\hat{\mathbf{w}}$ instead of $\mathbf{w}(\hat{\boldsymbol{\alpha}})$, and $\boldsymbol{\delta}$ instead of $\boldsymbol{\delta}^t$.

Now, the expected change of the dual objective is

$$\begin{aligned}\mathbb{E}[D(\boldsymbol{\alpha}^t) - D(\boldsymbol{\alpha}^{(t+1)})] =& \mathbb{E}[D(\boldsymbol{\alpha}^t) - D(\hat{\boldsymbol{\alpha}}) + D(\hat{\boldsymbol{\alpha}}) - D(\boldsymbol{\alpha}^{(t+1)})] \\ =& \mathbb{E}[D(\boldsymbol{\alpha}^t) - D(\hat{\boldsymbol{\alpha}})] + \mathbb{E}[D(\hat{\boldsymbol{\alpha}}) - D(\boldsymbol{\alpha}^{(t+1)})]\end{aligned}$$

Thus, it is a summation of two parts. Let us estimate both parts as following,

$$\begin{aligned}\mathbb{E}[D(\boldsymbol{\alpha}^t) - D(\hat{\boldsymbol{\alpha}})] =& \mathbb{E}\left[-\frac{1}{n}\sum_{i\notin\mathcal{I}}\phi_i^*(-\alpha_i) - \frac{1}{n}\sum_{k\in\mathcal{P}_S}\left(\sum_{i\in\mathcal{I}_k}\phi_i^*(-\alpha_i)\right) - \frac{\lambda}{2}\|\mathbf{w}\|^2\right. \\ &\left. + \frac{1}{n}\sum_{k\in\mathcal{P}_S}\left(\sum_{i\in\mathcal{I}_k}\phi_i^*(-\alpha_i)\right) + \frac{\lambda}{2}\|\hat{\mathbf{w}}\|^2\right] \\ =& \mathbb{E}\left[-\frac{\lambda}{2}\left(\|\mathbf{w}\|^2 - \|\hat{\mathbf{w}}\|^2\right) - \frac{1}{n}\sum_{i\notin\mathcal{I}}\phi_i^*(-\alpha_i)\right]\end{aligned}$$

$$\begin{aligned}&\mathbb{E}[D(\hat{\boldsymbol{\alpha}}) - D(\boldsymbol{\alpha}^{(t+1)})] \\ =& \mathbb{E}\left[D(\hat{\boldsymbol{\alpha}}) - D(\boldsymbol{\alpha} + \nu\sum_{k\in\mathcal{P}_S}\boldsymbol{\delta}_{[k]})\right] \\ \leq& \mathbb{E}\left[D(\hat{\boldsymbol{\alpha}}) - (1-\nu)D(\hat{\boldsymbol{\alpha}}) - \nu\sum_{k\in\mathcal{P}_S}Q_k^\sigma(\boldsymbol{\delta}_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}})\right] \\ &+ \mathbb{E}\left[\frac{\nu}{n}\sum_{k\in\mathcal{P}_S}(\mathbf{w}-\hat{\mathbf{w}})^\top\mathbf{X}\boldsymbol{\delta}_{[k]} + \frac{\lambda}{2}(\|\mathbf{w}\|^2 - \|\hat{\mathbf{w}}\|^2) + \frac{1}{n}\sum_{i\notin\mathcal{I}}\phi_i^*(-\alpha_i)\right] \quad \text{(Lemma 5)}\end{aligned}$$

Therefore,

$$
\begin{aligned}
&\mathbb{E}[D(\boldsymbol{\alpha}^t) - D(\boldsymbol{\alpha}^{(t+1)})] \\
\leq &\mathbb{E}\left[-\frac{\lambda}{2}\left(\|\mathbf{w}\|^2 - \|\hat{\mathbf{w}}\|^2\right) - \frac{1}{n}\sum_{i\notin\mathcal{I}}\phi_i^*(-\alpha_i)\right] \\
&+ \mathbb{E}\left[D(\hat{\boldsymbol{\alpha}}) - (1-\nu)D(\hat{\boldsymbol{\alpha}}) - \nu\sum_{k\in\mathcal{P}_S} Q_k^\sigma(\boldsymbol{\delta}_{[k]}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}})\right] \\
&+ \mathbb{E}\left[\frac{\nu}{n}\sum_{k\in\mathcal{P}_S}(\mathbf{w}-\hat{\mathbf{w}})^\top \mathbf{X}\boldsymbol{\delta}_{[k]} + \frac{\lambda}{2}(\|\mathbf{w}\|^2 - \|\hat{\mathbf{w}}\|^2) + \frac{1}{n}\sum_{i\notin\mathcal{I}}\phi_i^*(-\alpha_i)\right] \\
= &\nu\mathbb{E}\Bigg[D(\hat{\boldsymbol{\alpha}}) - \sum_{k\in\mathcal{P}_S} Q_k^\sigma(\boldsymbol{\delta}^*_{[k]}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}}) \\
&+ \sum_{k\in\mathcal{P}_S} Q_k^\sigma(\boldsymbol{\delta}^*_{[k]}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}}) - \sum_{k\in\mathcal{P}_S} Q_k^\sigma(\boldsymbol{\delta}_{[k]}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}})\Bigg] \\
&+ \mathbb{E}\left[\frac{\nu}{n}\sum_{k\in\mathcal{P}_S}(\mathbf{w}-\hat{\mathbf{w}})^\top \mathbf{X}\boldsymbol{\delta}_{[k]}\right] \\
\leq &\nu\Bigg(D(\hat{\boldsymbol{\alpha}}) - \sum_{k\in\mathcal{P}_S} Q_k^\sigma(\boldsymbol{\delta}^*_{[k]}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}}) \\
&+ \Theta\Big(\sum_{k\in\mathcal{P}_S} Q_k^\sigma(\boldsymbol{\delta}^*_{[k]}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}}) - \underbrace{\sum_{k\in\mathcal{P}_S} Q_k^\sigma(\mathbf{0}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}})}_{D(\hat{\boldsymbol{\alpha}})}\Big)\Bigg) \quad \text{(Lemma 4)} \\
&+ \mathbb{E}\left[\frac{\nu}{n}\sum_{k\in\mathcal{P}_S}(\mathbf{w}-\hat{\mathbf{w}})^\top \mathbf{X}\boldsymbol{\delta}_{[k]}\right] \\
= &\nu(1-\Theta)\left(D(\hat{\boldsymbol{\alpha}}) - \sum_{k\in\mathcal{P}_S} Q_k^\sigma(\boldsymbol{\delta}^*_{[k]}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}})\right) + \mathbb{E}\left[\frac{\nu}{n}\sum_{k\in\mathcal{P}_S}(\mathbf{w}-\hat{\mathbf{w}})^\top \mathbf{X}\boldsymbol{\delta}_{[k]}\right] \\
\leq &\nu(1-\Theta)\left(D(\hat{\boldsymbol{\alpha}}) - \sum_{k\in\mathcal{P}_S} Q_k^\sigma(\boldsymbol{\delta}^*_{[k]}; \boldsymbol{\alpha}_{[k]}, \hat{\mathbf{w}})\right) \\
&+ \frac{\nu}{2n}\underbrace{\left(\mathbb{E}\left[\|\mathbf{w}-\hat{\mathbf{w}}\|^2\right] + \mathbb{E}\left[\left\|\sum_{k\in\mathcal{P}_S}\mathbf{X}\boldsymbol{\delta}_{[k]}\right\|^2\right]\right)}_{A}
\end{aligned}
$$

Note that $\mathbf{w} = \hat{\mathbf{w}} + \frac{1}{\lambda n}\sum_{j=t-\Gamma}^{t-1}\mathbf{X}\boldsymbol{\delta}^j$. Now, let us bound the term A. We have

$$\begin{aligned}
A =& \mathbb{E}\left[\frac{1}{\lambda n}\left\|\sum_{j=t-\Gamma}^{t-1}\mathbf{X}\boldsymbol{\delta}^j\right\|^2\right] + \mathbb{E}\left[\left\|\sum_{k\in\mathcal{P}_S}\mathbf{X}\boldsymbol{\delta}_{[k]}\right\|^2\right] \\
\leq& \mathbb{E}\left[\frac{\Gamma}{\lambda n}\sum_{j=t-\Gamma}^{t-1}\left\|\mathbf{X}\boldsymbol{\delta}^j\right\|^2\right] + \mathbb{E}\left[S\sum_{k\in\mathcal{P}_S}\left\|\mathbf{X}\boldsymbol{\delta}_{[k]}\right\|^2\right] \quad \text{(Cauchy Schwarz Inequality)} \\
\leq& \mathbb{E}\left[\frac{\Gamma M}{\lambda n}\sum_{j=t-\Gamma}^{t-1}\left\|\boldsymbol{\delta}^j\right\|^2\right] + \mathbb{E}\left[SM\sum_{k\in\mathcal{P}_S}\left\|\boldsymbol{\delta}_{[k]}\right\|^2\right] \quad \text{(Proposition 7)} \\
\leq& \mathbb{E}\left[\frac{\Gamma M}{\lambda n}\left(\sum_{j=t-\Gamma}^{t-1}\varrho^j\right)\left\|\boldsymbol{\delta}^{t-1}\right\|^2\right] + \mathbb{E}\left[SM\sum_{k\in\mathcal{P}_S}\left\|\boldsymbol{\delta}_{[k]}\right\|^2\right] \quad \text{(By (4.23))} \\
\leq& \frac{\Gamma M L_{max}}{2\lambda n}\sum_{j=t-\Gamma}^{t-1}\varrho^j\left(D(\hat{\boldsymbol{\alpha}}) - \sum_{k\in\mathcal{P}_S}Q_k^\sigma(\boldsymbol{\delta}_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}})\right) \\
&+ \frac{SML_{max}}{2}\sum_{k\in\mathcal{P}_S}\left(D(\hat{\boldsymbol{\alpha}}) - Q_k^\sigma(\boldsymbol{\delta}_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}})\right) \quad \text{(Proposition 10)}
\end{aligned}$$

Here $D(\hat{\boldsymbol{\alpha}}) = Q_k^\sigma(\mathbf{0};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}})$. Thus, Eq. 4.1 can be rewritten as,

$$\begin{aligned}
\mathbb{E}\left[Q_k^\sigma(\boldsymbol{\delta}^*_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}}) - Q_k^\sigma(\boldsymbol{\delta}_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}})\right] \leq& \Theta\left(Q_k^\sigma(\boldsymbol{\delta}^*_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}}) - D(\hat{\boldsymbol{\alpha}})\right) + D(\hat{\boldsymbol{\alpha}}) - D(\hat{\boldsymbol{\alpha}}) \\
D(\hat{\boldsymbol{\alpha}}) - Q_k^\sigma(\boldsymbol{\delta}_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}}) \leq& (1-\Theta)D(\hat{\boldsymbol{\alpha}}) - (1-\Theta)Q_k^\sigma(\boldsymbol{\delta}^*_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}}) \\
D(\hat{\boldsymbol{\alpha}}) - Q_k^\sigma(\boldsymbol{\delta}_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}}) \leq& -(1-\Theta)\left(Q_k^\sigma(\boldsymbol{\delta}^*_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}}) - D(\hat{\boldsymbol{\alpha}})\right)
\end{aligned} \tag{4.27}$$

Then, A can be bounded as,

$$\begin{aligned}
A \leq & -\frac{\Gamma^2 ML_{max}}{2\lambda n}\varrho^{\Gamma}(1-\Theta)\sum_{k\in\mathcal{P}_S}\left(D(\hat{\boldsymbol{\alpha}}) - Q_k^{\sigma}(\boldsymbol{\delta}^*_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}})\right) \\
& -\frac{SML_{max}}{2}(1-\Theta)\sum_{k\in\mathcal{P}_S}\left(D(\hat{\boldsymbol{\alpha}}) - Q_k^{\sigma}(\boldsymbol{\delta}^*_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}})\right) \quad \text{(By (4.27))} \\
\leq & -\frac{\Gamma^2 eML_{max}}{2\lambda n}(1-\Theta)\sum_{k\in\mathcal{P}_S}\left(D(\hat{\boldsymbol{\alpha}}) - Q_k^{\sigma}(\boldsymbol{\delta}^*_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}})\right) \\
& -\frac{SML_{max}}{2}(1-\Theta)\sum_{k\in\mathcal{P}_S}\left(D(\hat{\boldsymbol{\alpha}}) - Q_k^{\sigma}(\boldsymbol{\delta}^*_{[k]};\boldsymbol{\alpha}_{[k]},\hat{\mathbf{w}})\right) \quad \text{(Assumption (5))}
\end{aligned}$$

By substituting A, we have

$$\mathbb{E}[D(\boldsymbol{\alpha}^t) - D(\boldsymbol{\alpha}^{(t+1)})] \leq \nu\left(1 - \frac{\Gamma^2 eML_{max}}{4\lambda n^2} - \frac{SML_{max}}{4n}\right)(1-\Theta)\sum_{k\in\mathcal{P}_S}\left(D(\hat{\boldsymbol{\alpha}}) - Q_k^{\sigma}(\boldsymbol{\delta}^*_{[k]})\right)$$

$$\mathbb{E}[D(\boldsymbol{\alpha}^t) - D(\boldsymbol{\alpha}^{(t+1)})] \leq \Psi(1-\Theta)\sum_{k\in\mathcal{P}_S}\left(D(\hat{\boldsymbol{\alpha}}) - Q_k^{\sigma}(\boldsymbol{\delta}^*_{[k]})\right)$$

Using the Eq. C in the proof of Lemma 5 in [39], we can show that

$$\begin{aligned}
\mathbb{E}[D(\boldsymbol{\alpha}^t) - D(\boldsymbol{\alpha}^{(t+1)})] \leq & \Psi(1-\Theta)\left(-sG(\hat{\boldsymbol{\alpha}}) - \frac{1}{2\mu}(1-s)s\frac{1}{n}\left\|\hat{\mathbf{u}} - \hat{\boldsymbol{\alpha}}\right\|^2\right. \\
& \left. + \frac{\sigma}{2\lambda}\left(\frac{s}{n}\right)^2\sum_{k\in\mathcal{P}_S}\left\|\mathbf{X}(\hat{\mathbf{u}} - \hat{\boldsymbol{\alpha}})_{[k]}\right\|\right) \\
= & \Psi(1-\Theta)\left(-sG(\hat{\boldsymbol{\alpha}}) + \frac{\sigma}{2\lambda}\left(\frac{s}{n}\right)^2\hat{R}\right)
\end{aligned}$$

□

Using the main results in [39] and combining Lemma 4 with Lemma 6 gives us the following two convergence results, one for smooth loss functions and the other for the Lipschitz continuous loss functions. The theorems use the quantities $\sigma_{max} =$

$\max_k \sigma_k,\ \sigma_{sum} = \sum_k \sigma_k n_k$ where $\forall k, \sigma_k = \max_{\alpha_{[k]} \in \mathbb{R}^n} \left\|\mathbf{X}\boldsymbol{\alpha}_{[k]}\right\|^2 / \left\|\boldsymbol{\alpha}_{[k]}\right\|^2$.

Theorem 11. *If the loss functions ϕ_i are all $(1/\mu)$-smooth, then in T_1 iterations Algorithm 5 finds a solution with objective at most ϵ_D from the optimal,* i.e., $\mathbb{E}[D(\alpha^*) - D(\alpha^{(T_1)})] \leq \epsilon_D$ *whenever* $T_1 \geq C_1 \log \frac{1}{\epsilon_D}$ *where* $C_1 = \frac{1}{\Psi(1-\Theta)}(1 + \frac{\sigma_{max}\sigma}{\nu\lambda n})$ *and* Θ *is given by* (4.17). *Furthermore, in T_2 iterations, it finds a solution with duality gap at most ϵ_{gap},* i.e., $\mathbb{E}[P(\mathbf{w}(\alpha^{(T_2)})) - D(\alpha^{(T_2)})] \leq \epsilon_{gap}$ *whenever* $T_2 \geq C_1 \log \frac{C_1}{\epsilon_D}$.

Theorem 12. *If the loss functions ϕ_i are all L-Lipschitz, then in T_1 iterations Algorithm 5 finds a solution with duality gap atmost ϵ_{gap},* i.e., $\mathbb{E}[P(\mathbf{w}(\bar{\alpha})) - D(\bar{\alpha})] \leq \epsilon_{gap}$ *for the average iterate* $\bar{\alpha} = \frac{1}{T_1 - T_0} \sum_{t=T_0+1}^{T_1 - 1} \alpha^{(t)}$ *whenever* $T_1 \geq T_0 + \max\{\lceil \frac{1}{\Psi(1-\Theta)} \rceil, \frac{4L^2\sigma_{sum}\sigma}{\lambda n^2 \epsilon_{gap} \Psi(1-\Theta)}\}$, *and* $T_0 \geq \max\{0, \lceil \frac{1}{\Psi(1-\Theta)} \log \frac{2\lambda n^2 (D(\alpha^* - D(\alpha^{(0)})}{4L^2\sigma_{sum}\sigma} \rceil\} + \max\{0, \frac{2}{\Psi(1-\Theta)}(\frac{8L^2\sigma_{sum}\sigma}{\lambda n^2 \epsilon_{gap}} - 1)\}$ *and* Θ *is given by* (4.17).

Theorem 12 establishes the convergence for L-Lipschitz continuous loss functions, and Theorem 11 proves a linear convergence rate for smooth convex loss functions.

5 Communication Cost Analysis

In each communication round, the algorithms based on synchronous updates on all K nodes require $2K$ transmissions, each consisting of all values of $\mathbf{v}$ or $\Delta\mathbf{v}$. Half of these transmissions are from the workers to the master and the rest are from the master to the workers. Whereas, our asynchronous update scheme requires $2S$ transmissions in each round.

Table 6.1: Datasets

Dataset	LIBSVM name	n	d	nnz	File size
`rcv1`	rcv1_test	677,399	47,236	49,556,258	1.2 GB
`webspam`	webspam	280,000	16,609,143	1,045,051,224	20 GB
`kddb`	kddb train	19,264,097	29,890,095	566,345,888	5.1 GB
`splicesite`	splice_site.t	4,627,840	11,725,480	15,383,587,858	280 GB

6 Experimental Results

We implemented our algorithm in C++ where each node runs exactly one MPI process which in turn runs one OpenMP thread on each core available within the node and the main thread handles the inter-node communication. We evaluated for hinge loss, though it applies to other loss functions too, on four datasets from LIBSVM website as shown in Table 6.1, using up to 16 nodes available with the Hornet cluster at University of Connecticut where each node has 24 Xeon E5-2690 cores and 128 GB main memory. The last column in Table 6.1 gives the total number of non-zero entries in the matrix X for each of the four dataset we used.

We experimented with the following algorithms: 1) *Baseline*: an implementation of DCA [27], 2) *CoCoA+* [38], 3) *PassCoDe* [28] and 4) our *Hybrid-SDCA*. The algorithm parameters were varied as follows: 1) regularization parameter $\lambda \in \{10^{-3}, 10^{-4}, 10^{-5}\}$, 2) local iterations $H = \{10000, 20000, 40000\}$, 3) aggregation parameters $\nu = 1$, and 4) scaling parameter $\sigma = K, S$ for *CoCoA+*, *Hybrid-SDCA*, respective, as recommended in [38]. For different combinations of λ, H, we observed similar patterns of results and report for $\lambda = 10^{-4}, H = 40000$ only. The details of other parameter values are given later.

6.1 Comparison with existing algorithms

Figure 6.1 shows the progress of duality gap achieved by the four algorithms on three smaller datasets. We chose the number of nodes ($p \leq K$) and the number of cores ($t \leq R$) per node such that the total number of worker cores ($p \times t$) is the same (16) for all algorithms except *Baseline*. The duality gap is measured as $P(\mathbf{v}) - D(\boldsymbol{\alpha})$ where $\mathbf{v}$ is the estimate of $\mathbf{w}(\boldsymbol{\alpha})$ shared across the nodes. When $S < K$, it is not possible for the master in *Hybrid-SDCA* to gather the parts of $P(\mathbf{v})$ from all workers at the end of each round. We let the master temporarily store $\mathbf{v}$ in disk after each round and at the end of all stipulated rounds, the workers compute the respective parts of $P(\mathbf{v})$ from the stored $\mathbf{v}$ and the master computes the duality gap using a series of synchronous all-reduce computations from all the workers. The bottom row shows the progress of the duality gap across time, while the top row shows progress across each round that consists of a communication round in *CoCoA+* and *Hybrid-SDCA* whereas consists of H local updates in *Baseline* and *PassCoDe*. In this experiment, *Hybrid-SDCA* uses $S = p$ and $\Gamma = 1$ making the global updates synchronous. The progress of baseline is slow as it applies only H updates compared to $H \times p \times t$ updates by the other algorithms. In terms of time, *Hybrid-SDCA* clearly outperforms both *CoCoA+*, as expected, and *PassCoDe* which incurs a larger number of cache-misses when many cores are used. In terms of the number of rounds, *PassCoDe* outperforms both *CoCoA+* and *Hybrid-SDCA*, as expected. However, *PassCoDe* is not scalable beyond the number of cores in a single node. As the number of nodes increases, the convergence of *Hybrid-SDCA* becomes slower due to the costly merging process of many distributed updates.

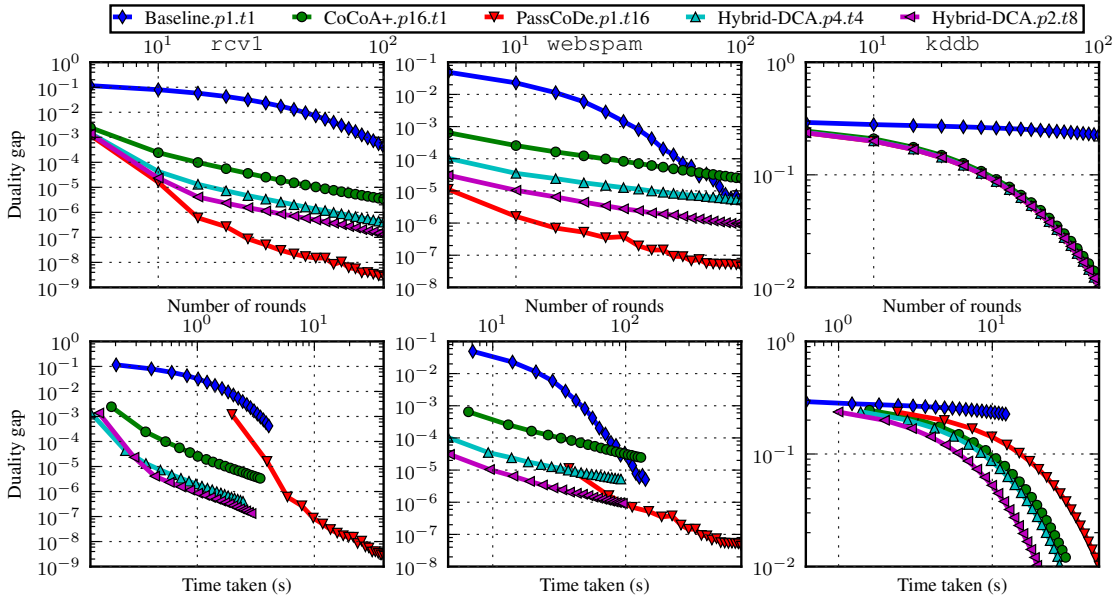


Figure 6.1: Performance of different solvers on three datasets, `rcv1` (left column), `webspam` (middle column), and `kddb` (right column), in terms of the progress of the duality gap across the number of rounds (top row) and across the wall time taken (bottom row).

6.2 Speedup

We ran sufficient rounds of each of the four algorithms such that the duality gap falls below a threshold and noted the time taken by the algorithms to achieve this threshold. Figure 6.2 shows the speedup(p, t) of all the algorithms except, *Baseline*, computed as the ratio of the time taken by *Baseline* to the time taken by the target algorithm on p nodes each with t cores. The thresholds used were $10^{-4}, 10^{-5}, 10^{-1}$ for `rcv1, webspam, kddb`, respectively. *PassCoDe* can be run only on a single node; so we vary only the number of cores. *CoCoA+* uses only 1 core per node. We ran *CoCoA+* and *Hybrid-SDCA* on $p \in \{2, 4, 8, 16\}$ nodes and plotted them separately. For each p, *Hybrid-SDCA* uses $t \in \{2, 4, 8, 16, 24\}$ cores per node, however, under the restriction that the total number of worker cores $(p \times t)$ does not exceed 144, a limit

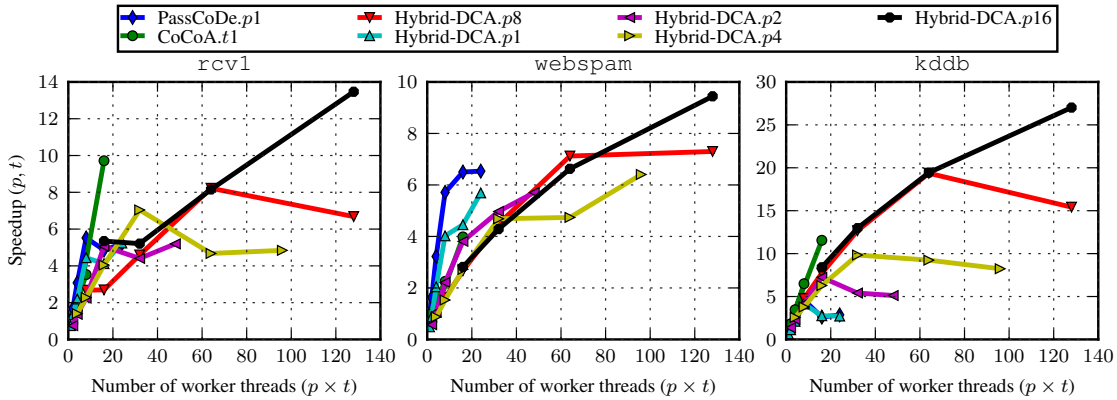


Figure 6.2: Speedup of different parallel or distributed solvers with respect to the sequential implementation *Baseline*.

set by our HPC usage policy. When $t > 8$, the number of cache-misses increases due to thread switching on the physical cores and reduces speedup for both *PassCoDe* and *Hybrid-SDCA*. This could be improved by carefully scheduling the OpenMP threads to the same physical cores. Nevertheless, *Hybrid-SDCA* has good speedup for $t \leq 8$, as evident for $p = 16$.

6.3 Effects of the parameter S

Figure 6.3 shows the results of varying $S \in \{2, 3, 4, 6, 8\}$ with fixed $\Gamma = 10$ on $p = 8$ nodes each with $t = 8$ cores. When $S < p/2$, only a minority of the workers contribute in a round and the duality gap does not progress below some certain level. On the other hand, when at least half of the workers contribute in each round, it is possible to achieve the same duality level obtained using all the workers. However, the reduction in time per round is eventually eaten by the larger number of rounds required to achieve the same duality gap. Nevertheless, the approach is useful for HPC platforms with heterogeneous nodes, unlike ours, where the waiting for updates from all workers

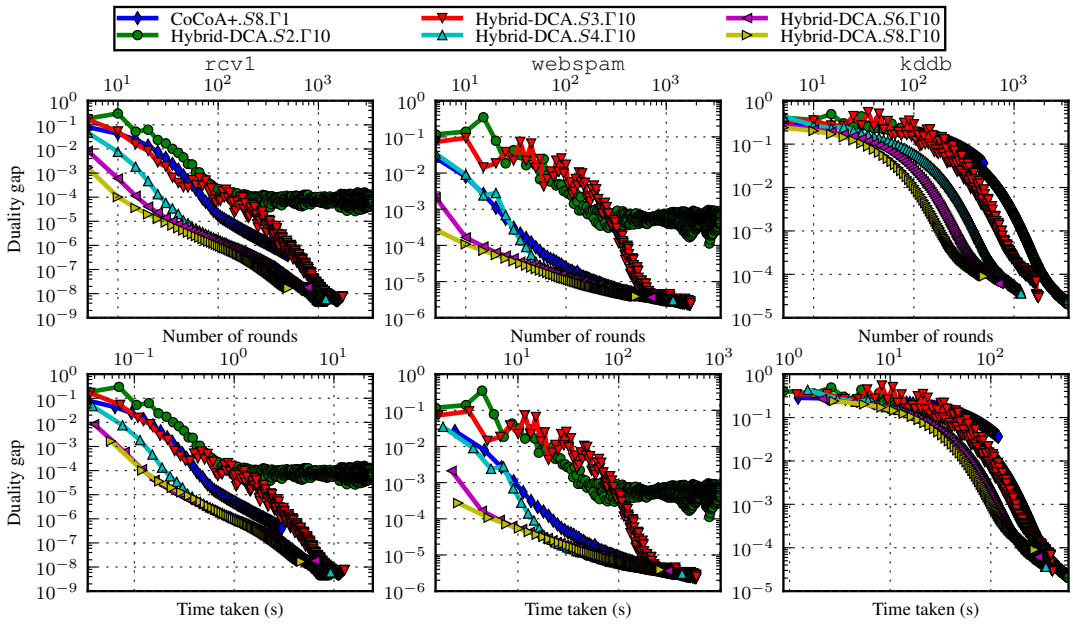


Figure 6.3: Effect of varying S on $p = 8$ worker nodes, with Γ fixed at 10.

has larger penalty per round, or for the case, where the need is to run for a specified number of rounds and quickly achieve a reasonably good duality gap.

6.4 Effects of the parameter Γ

Figure 6.4 shows the results of varying $\Gamma \in \{1, 2, 3, 4, 10\}$ with fixed $S = 6$ on $p = 8$ nodes each with $t = 8$ cores. We do not see much effect of Γ as the HPC platform used for our experiments has homogeneous nodes. Our experimentation showed that even if we use $\Gamma = 10$, the stale value at any worker was for at most 4 rounds. We expect to see a larger variance of staleness in case of heterogeneous nodes.

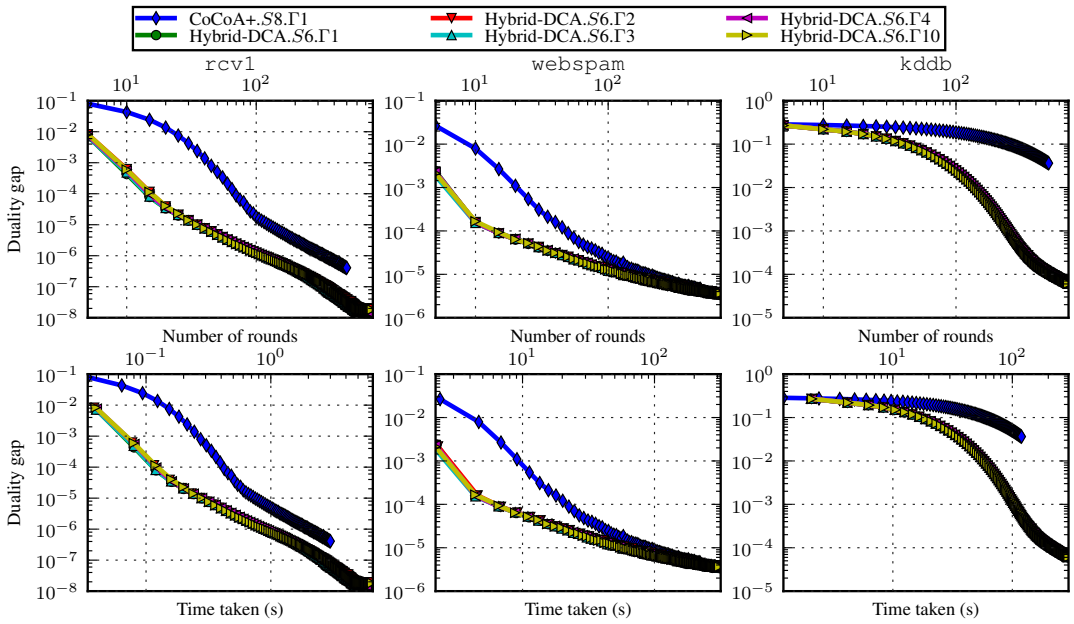


Figure 6.4: Effect of varying Γ on $p = 8$ worker nodes, with S fixed at 8.

6.5 Performance on a big dataset

We experimented our hybrid algorithm on the big dataset `splicesite` of size about 280 GB and compared with the previous best algorithm *CoCoA+*. Because of the enormous size, the dataset cannot be accommodated on a single node and hence *PassCoDe* cannot be run on this dataset. In this experiment, we used the number of local iterations $H = 10,000$. The results are shown in Figure 6.5 where the progress of duality gap across the rounds of communication is shown on the left and across the wall time on the right. To achieve a duality gap of at least 10^{-6} on 16 nodes, *CoCoA+* took more than 300 seconds which somewhat matches the 1200 seconds (20 minutes) time on 4 nodes reported in [38]. *Hybrid-SDCA* on 16 nodes each using 8 cores took about 30 seconds to achieve the same duality gap, showing enough evidence about the scalability of our algorithm. One could also argue that *CoCoA+* can be run on all

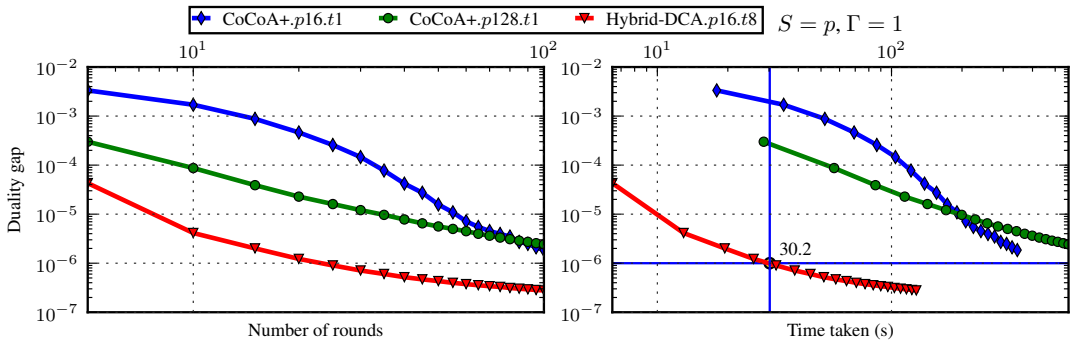


Figure 6.5: Performance of *Hybrid-SDCA* on big dataset `splicesite`.

these 16x8=128 cores, treating each core as a distributed node. We also experimented with this configuration which achieved better progress on duality gap than *CoCoA+* on 16 nodes, however, still performed far worse than *Hybrid-SDCA* in terms of both the number of rounds and the time taken.

7 Conclusions

In this paper, we present a hybrid parallel and distributed asynchronous stochastic dual coordinate ascent algorithm utilizing modern HPC platforms with many nodes of multi-core shared-memory systems. We analyze the convergence properties of this novel algorithm which uses asynchronous updates at two cascading levels: inter-cores and inter-nodes. Experimental results show that our algorithm is faster than the state-of-the-art distributed algorithms and scales better than the state-of-the-art parallel algorithms.

Chapter 5

Concluding Remarks

In this dissertation, we have presented our studies toward jointly learning features and temporal contingency for prediction in large scale datasets, where we construct two learning models using marginal modeling and feature selection technologies as well as a distributed asynchronous solver for solving RRM problems in a hybrid memory system. In the first direction of the study, we propose new models to efficiently learn relevant features and relevant lagged effects from longitudinal data with the consideration of non-i.i.d nature of longitudinal data via GEE. Adding another layer of difficulty, we develop a new approach to learn more complex feature structures with the spectral and temporal information and use advanced technique, QIF, to estimate complex correlations within the temporal data as well. Along the second direction, we propose a novel algorithm on a hybrid memory system which consists of multiple distributed nodes each having multiple cores and its own shared memory. We take SDCA as an example method that can fit into our double asynchronous framework. Our contributions in this dissertation are summarized as the following:

- **Longitudinal LASSO: Jointly Learning Features and Temporal Contingency for Outcome Prediction.** The proposed method makes predictions based on lagged data from current and previous time points. It decomposes the model coefficients into a summation of two components and impose different block-wise *least absolute shrinkage and selection operators* (LASSO) to the two components. One regularizer is used to detect the contingency of specific time points whereas the other is used to select covariates. The proposed method also learns simultaneously a structured correlation matrix from the data. The correlations among the outcomes themselves imply the changing trend of the outcomes in the proximal time points within each subject. Moreover, we develop a family of methods where the outcome variable is assumed to follow a distribution from the exponential family, including Bernoulli, Gaussian and Poisson distributions. We provide the convergence analysis and asymptotic analysis to show that the proposed algorithm can find the optimal solution for the predictive models.

- **Jointly Learning Multi-dimensional Features and Temporal Contingency in Longitudinal Data.** We propose a novel learning formulation that constructs tensor-based predictive models as functions of covariates not only from the current observation but also from multiple previous consecutive observations, and simultaneously determine the temporal contingency and the most influential features along each dimension of the tensor data. The proposed method makes predictions based on lagged data from current and previous time points. It decompose the K-way parameter tensor into a summation of K sparse K-way tensors. Hence, our approach formulates a convex optimization problem.

The proposed method also learns simultaneously correlation information from the data via QIF. The correlations among the outcomes themselves imply the changing trend of the outcomes in the proximal time points within each subject. We provide several theoretical results to show the properties of the proposed model. We empirically illustrate the scenarios where the new formulation are more suitable for temporal data.

- **Hybrid-SDCA: A Double Asynchronous Approach for Stochastic Dual Coordinate Ascent.** We propose and analyze a hybrid asynchronous shared memory and asynchronous distributed memory implementation (*Hybrid-SDCA*) of the mostly used SDCA algorithm to solve RRM problems. We also prove a strong guarantee of convergence for L-Lipschitz continuous loss functions, and further linear convergence when a smooth convex loss function is used. Moreover, the experimental results using our light-weight OpenMP+MPI implementation show that our algorithms are much faster than existing distributed memory algorithms, and easily scale up with the volume of data in comparison with the shared memory based algorithms as the shared memory size is limited.

Bibliography

[1] Alekh Agarwal, Olivier Chapelle, Miroslav Dudík, and John Langford, *A Reliable Effective Terascale Linear Learning System*, Journal of Machine Learning Research **15** (2014), no. 1, 1111–1133.

[2] Sungjin Ahn, Max Welling, and M Welling U V a Nl, *Distributed Stochastic Gradient MCMC*, Proceedings of the 31st International Conference on Machine Learning **32** (2014).

[3] Stephen Armeli, Tamlin S Conner, Jerry Cullum, and Howard Tennen, *A longitudinal analysis of drinking motives moderating the negative affect-drinking association among college students.*, Psychology of Addictive Behaviors **24** (2010), no. 1, 38–47.

[4] Andrew Arnold, Yan Liu, and Naoki Abe, *Temporal causal modeling with graphical granger methods*, Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), ACM, 2007, pp. 66–75.

[5] Yang Bai, Wing K. Fung, and Zhong Yi Zhu, *Penalized quadratic inference functions for single-index models with longitudinal data*, Journal of Multivariate Analysis **100** (2009), no. 1, 152–161.

[6] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences **2** (2009), no. 1, 183–202.

[7] Jinbo Bi, Jiangwen Sun, Yu Wu, Howard Tennen, and Stephen Armeli, *A machine learning approach to college drinking prediction and risk factor identification*, ACM Trans. Intell. Syst. Technol. **4** (2013), no. 4, 72:1–72:24.

[8] Jinbo Bi, Tingyang Xu, Chi-Ming Chen, and Jason Johannesen, *Spatio-temporal modeling of eeg data for understanding working memory*, ICML Workshop on Statistics, Machine Learning and Neuroscience (Stamlins 2015), 2015.

[9] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends® in Machine Learning **3** (2011), no. 1, 1–122.

[10] Christopher D Brown and Herbert T Davis, *Receiver operating characteristics curves and related decision measures: A tutorial*, Chemometrics and Intelligent Laboratory Systems **80** (2006), no. 1, 24–38.

[11] Deng Cai, Xiaofei He, Ji-Rong Wen, Jiawei Han, and Wei-Ying Ma, *Support tensor machines for text categorization*, Technical Report (2006).

[12] Chi-Ming A. Chen, Arielle D. Stanford, Xiangling Mao, Anissa Abi-Dargham, Dikoma C. Shungu, Sarah H. Lisanby, Charles E. Schroeder, and Lawrence S. Kegeles, *Gaba level, gamma oscillation, and working memory performance in schizophrenia*, NeuroImage.Clinical **4** (2014), 531–539 (English).

[13] Peter Diggle, Patrick Heagerty, Kung-Yee Liang, and Scott Zeger, *Analysis of longitudinal data*, Oxford University Press, 2002.

[14] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, *LIBLINEAR: A Library for Large Linear Classification*, Journal of Machine Learning Research **9** (2008), 1871–1874.

[15] Maryam Fazel, Haitham Hindi, and Stephen P Boyd, *A rank minimization heuristic with application to minimum order system approximation*, American Control Conference, 2001. Proceedings of the 2001, vol. 6, IEEE, 2001, pp. 4734–4739.

[16] J. H. Fowler and N. A. Christakis, *Dynamic spread of happiness in a large social network: Longitudinal analysis over 20 years in the framingham heart study editorial comment*, Journal of Urology **181** (2009), no. 5, 2258–2259.

[17] Wenjiang J. Fu, *Penalized estimating equations*, Biometrics **59** (2003), no. 1, pp. 126–132 (English).

[18] Pinghua Gong, Jieping Ye, and Changshui Zhang, *Robust multi-task feature learning*, Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), ACM, 2012, pp. 895–903.

[19] Clive WJ Granger, *Testing for causality: a personal viewpoint*, Journal of Economic Dynamics and Control **2** (1980), 329–352.

[20] Lars Peter Hansen, *Large sample properties of generalized method of moments estimators*, Econometrica: Journal of the Econometric Society (1982), 1029–1054.

[21] David R Hardoon, John Shawe-Taylor, and Ola Friman, *KCCA Feature Selection for fMRI Analysis*, Technical Report CSD-TR-03-02, Royal Holloway University of London, 2003.

[22] Biyu J. He, *Scale-free brain activity: past, present, and future*, Trends in Cognitive Sciences **18** (2014), no. 9, 480–487.

[23] Christina Heinze, Brian McWilliams, and Nicolai Meinshausen, *Dual-loco: Distributing statistical estimation using random projections*, arXiv preprint arXiv:1506.02554 (2015).

[24] Christoph S. Herrmann, Daniel Senkowski, and Stefan Rottger, *Phase-locking and amplitude modulations of eeg alpha: Two measures reflect different cognitive processes in a working memory task*, Experimental Psychology **51** (2004), no. 4, 311 (English).

[25] Frank L Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, Journal of Mathematics and Physics **6** (1927), no. 1, 164–189.

[26] Mingyi Hong, Zhi-quan Luo, and Meisam Razaviyayn, *Convergence Analysis of Alternating Direction Method of Multipliers for a Family of Nonconvex Problems*, SIAM Journal on Optimization **26** (2016), no. 1, 337—-364.

[27] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan, *A dual coordinate descent method for large-scale linear SVM*, Proceedings of the 25th international conference on Machine Learning (ICML), 2008, pp. 408–415.

[28] Cho-Jui Hsieh, Hsiang-Fu Yu, and Inderjit S Dhillon, *PASSCoDe: Parallel ASynchronous Stochastic dual Co-ordinate Descent*, Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015.

[29] Hung Hung and Chen-Chien Wang, *Matrix variate logistic regression model with application to eeg data*, Biostatistics **14** (2013), no. 1, 189–202.

[30] Martin Jaggi, Virginia Smith, Martin Takác, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan, *Communication-Efficient Distributed Dual Coordinate Ascent*, Advances in Neural Information Processing Systems (NIPS), 2014, pp. 3068–3076.

[31] Wolfgang Klimesch, *Eeg alpha and theta oscillations reflect cognitive and memory performance: a review and analysis*, Brain research reviews **29** (1999), no. 2, 169–195.

[32] N. M. Laird and J. H. Ware, *Random-effects models for longitudinal data*, Biometrics **38** (1982), no. 4, 963–974.

[33] Jason Lee, Qihang Lin, Tengyu Ma, and Tianbao Yang, *Distributed stochastic variance reduced gradient methods and a lower bound for communication complexity*, CoRR **arXiv:1507.07595** (2015).

[34] Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu, *Asynchronous Parallel Stochastic Gradient for Nonconvex Optimization*, Advances in Neural Information Processing Systems 28, no. 1, 2015, pp. 2737–2745.

[35] K. Y. Liang and S. L. Zeger, *Longitudinal data-analysis using generalized linear-models*, Biometrika **73** (1986), no. 1, 13–22.

[36] Ji Liu, Stephen J Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar, *An Asynchronous Parallel Stochastic Coordinate Descent Algorithm*, Journal of Machine Learning Research **16** (2015), 285–322.

[37] AC Lozano, Naoki Abe, Yan Liu, and Saharon Rosset, *Grouped graphical granger modeling methods for temporal causal modeling*, Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (2009), 577–585.

[38] Chenxin Ma, Jakub Konečnỳ, Martin Jaggi, Virginia Smith, Michael I Jordan, Peter Richtárik, and Martin Takáč, *Distributed optimization with arbitrary local solvers*, arXiv preprint arXiv:1512.04039 (2015).

[39] Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I Jordan, Peter Richtárik, and Martin Takáč, *Adding vs. Averaging in Distributed Primal-Dual Optimization*, Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015.

[40] P. McCullagh and J. A. Nelder, *Generalized linear models (Second edition)*, London: Chapman & Hall, 1989.

[41] C.E. McCulloch and S.R. Searle, *Generalized, linear, and mixed models*, Wiley, New York, NY, USA, 2001.

[42] Ryan Mcdonald, Mehryar Mohri, Nathan Silberman, Dan Walker, and Gideon S Mann, *Efficient large-scale distributed training of conditional maximum entropy models*, Advances in Neural Information Processing Systems, 2009, pp. 1231–1239.

[43] Brian McWilliams, Christina Heinze, Nicolai Meinshausen, Gabriel Krummenacher, and Hastagiri P Vanchinathan, *Loco: Distributing ridge regression with random projections*, arXiv preprint arXiv:1406.3469 (2014).

[44] Eric Moulines and Francis R Bach, *Non-asymptotic analysis of stochastic approximation algorithms for machine learning*, Advances in Neural Information Processing Systems, 2011, pp. 451–459.

[45] Bengt Muthén and Tihomir Asparouhov, *Growth mixture modeling: Analysis with non-gaussian random effects*, Longitudinal data analysis (2008), 143–165.

[46] Deanna Needell, Rachel Ward, and Nati Srebro, *Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm*, Advances in Neural Information Processing Systems, 2014, pp. 1017–1025.

[47] Radu Stefan Niculescu, Tom M Mitchell, and R Bharat Rao, *Modeling the fMRI signal via Hierarchical Clustered Hidden Process Models.*, AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium (2007), 558–562.

[48] Feng Niu, Benjamin Recht, Christopher Re, and Stephen J Wright, *HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent*, Advances in Neural Information Processing Systems (2011), no. 1, 21.

[49] Ulf Olsson, *Generalized linear models*, vol. 18, 2002.

[50] Michael R Osborne, Brett Presnell, and Berwin A Turlach, *On the lasso and its dual*, Journal of Computational and Graphical statistics **9** (2000), no. 2, 319–337.

[51] Hua Ouyang, N He, and Alexander Gray, *Stochastic ADMM for nonsmooth optimization*, arXiv preprint arXiv:1211.0632 (2012), 1–11.

[52] Zhimin Peng, Yangyang Xu, Ming Yan, and Wotao Yin, *Arock: an algorithmic framework for asynchronous parallel coordinate updates*, arXiv preprint arXiv:1506.02396 (2015).

[53] Annie Qu and Runze Li, *Quadratic inference functions for varying-coefficient models with longitudinal data*, Biometrics **62** (2006), no. June, 379–391.

[54] Annie Qu, Bruce G Lindsay, and Bing Li, *Improving generalised estimating equations using quadratic inference functions*, Biometrika **87** (2000), no. 4, 823–836.

[55] Sridhar Raghavachari, Michael J Kahana, Daniel S Rizzuto, Jeremy B Caplan, Matthew P Kirschen, Blaise Bourgeois, Joseph R Madsen, and John E Lisman, *Gating of human theta oscillations by a working memory task*, The journal of Neuroscience **21** (2001), no. 9, 3175–3183.

[56] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM review **52** (2010), no. 3, 471–501.

[57] Peter Richtárik and Martin Takáč, *Distributed coordinate descent method for learning with big data*, arXiv preprint arXiv:1310.2059 (2013).

[58] Rebecca J. Sela and Jeffrey S. Simonoff, *Re-em trees: a data mining approach for longitudinal and clustered data*, Machine Learning **86** (2012), no. 2, 169–207 (English).

[59] Thomas A Severini, *Elements of distribution theory*, vol. 17, Cambridge University Press, 2005.

[60] Shai Shalev-Shwartz and Tong Zhang, *Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization*, Journal of Machine Learning Research **14** (2013), no. 1, 567–599.

[61] ———, *Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization*, Mathematical Programming **155** (2016), no. 1-2, 105–145.

[62] Ohad Shamir, Nathan Srebro, and Tong Zhang, *Communication-Efficient Distributed Optimization using an Approximate Newton-type Method*, Proceedings of the 31th International Conference on Machine Learning, (ICML), Beijing, China, 21-26 June 2014, 2014, pp. 1000–1008.

[63] Li Shen, Paul M. Thompson, Steven G. Potkin, Lars Bertram, Lindsay A. Farrer, Tatiana M. Foroud, Robert C. Green, Xiaolan Hu, Matthew J. Huentelman, Sungeun Kim, John S K Kauwe, Qingqin Li, Enchi Liu, Fabio Macciardi, Jason H. Moore, Leanne Munsie, Kwangsik Nho, Vijay K. Ramanan, Shannon L. Risacher, David J. Stone, Shanker Swaminathan, Arthur W. Toga, Michael W. Weiner, and Andrew J. Saykin, *Genetic analysis of quantitative phenotypes in AD and MCI: Imaging, cognition and biomarkers*, Brain Imaging and Behavior **8** (2014), no. 2, 183–207.

[64] Changkyu Song and Vladimir Pavlovic, *Fast ADMM Algorithm for Distributed Optimization with Adaptive Penalty*, Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016), 2016, pp. 1–11.

[65] C. A. Stappenbeck and K. Fromme, *A longitudinal investigation of heavy drinking and physical dating violence in men and women*, Addict Behav **35** (2010), no. 5, 479–85.

[66] Martin Takac, Avleen Bijral, Peter Richtarik, and Nati Srebro, *Mini-Batch Primal and Dual Methods for SVMs*, Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 1022–1030.

[67] Xu Tan, Yin Zhang, Siliang Tang, Jian Shao, Fei Wu, and Yueting Zhuang, *Logistic tensor regression for classification*, Intelligent Science and Intelligent Data Engineering, Springer, 2012, pp. 573–581.

[68] Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima, *Estimation of low-rank tensors via convex optimization*, arXiv preprint arXiv:1010.0789 (2010).

[69] Ryota Tomioka and Taiji Suzuki, *Convex tensor decomposition via structured schatten norm regularization*, Advances in neural information processing systems, 2013, pp. 1331–1339.

[70] Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou, *Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty*, Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP **1** (2009), 477.

[71] Ledyard R Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika **31** (1966), no. 3, 279–311.

[72] Dimitri Van de Ville, Juliane Britz, and Christoph M Michel, *EEG microstate sequences in healthy humans at rest reveal scale-free dynamics.*, Proceedings of the National Academy of Sciences of the United States of America **107** (2010), no. 42, 18179–18184.

[73] Gael Varoquaux, Alexandre Gramfort, Fabian Pedregosa, Vincent Michel, and Bertrand Thirion, *Multi-subject dictionary learning to segment an atlas of brain spontaneous activity*, Information Processing in Medical Imaging **6801 LNCS** (2011), 562–573.

[74] Fei Wang, Ping Zhang, Buyue Qian, Xiang Wang, and Ian Davidson, *Clinical risk prediction with multilinear sparse logistic regression*, Proceedings of the 20th

ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14 (2014), 145–154.

[75] L. Wang, J. H. Zhou, and A. N. Qu, *Penalized generalized estimating equations for high-dimensional longitudinal data analysis*, Biometrics **68** (2012), no. 2, 353–360.

[76] Z Wang and G Iyengar, *An Asynchronous Distributed Proximal Gradient Method for Composite Convex Optimization*, Proceedings of the 32nd International Conference on Machine Learning **37** (2015).

[77] Philip M. Westgate and Thomas M. Braun, *An improved quadratic inference function for parameter estimation in the analysis of correlated data*, Statistics in Medicine **32** (2013), no. 19, 3260–3273.

[78] Tingyang Xu, Jiangwen Sun, and Jinbo Bi, *Longitudinal lasso: Jointly learning features and temporal contingency for outcome prediction*, Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1345–1354.

[79] Mingan Yang and David B. Dunson, *Bayesian semiparametric structural equation models with latent variables*, Psychometrika **75** (2010), 675–693.

[80] Tianbao Yang, *Trading Computation for Communication: Distributed Stochastic Dual Coordinate Ascent*, Advances in Neural Information Processing Systems, 2013, pp. 629–637.

[81] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin, *Dual coordinate descent methods for logistic regression and maximum entropy models*, Machine Learning **85** (2011), no. 1-2, 41–75.

[82] Ruiliang Zhang and James Kwok, *Asynchronous distributed ADMM for consensus optimization*, Proceedings of the 31st International Conference on Machine Learning (ICML), 2014, pp. 1701–1709.

[83] Tong Zhang, *Solving large scale linear prediction problems using stochastic gradient descent algorithms*, Proceedings of the twenty-first international conference on Machine learning, ACM, 2004, p. 116.

[84] Yu Zhang and Dit-Yan Yeung, *Multi-task learning using generalized t process*, Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 2010.

[85] Yu Zhang, Dit-Yan Yeung, and Qian Xu, *Probabilistic multi-task feature selection*, Advances in Neural Information Processing Systems, 2010, pp. 2559–2567.

[86] Yuchen Zhang and Lin Xiao, *Communication-efficient distributed optimization of self-concordant empirical loss*, CoRR **abs/1501.00263** (2015).

[87] Hua Zhou, Lexin Li, and Hongtu Zhu, *Tensor regression with applications in neuroimaging data analysis*, Journal of the American Statistical Association **108** (2013), no. 502, 540–552.