

Wavelet Regression using MapReduce and Analysis of Multiple Sclerosis Clinical Data

by

Hanyu Song

Department of Department of Statistical Science
Duke University

Date: _____

Approved:

Li Ma, Supervisor

Meng Li

Cliburn Chan

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Department of Department of Statistical Science
in the Graduate School of Duke University
2017

ABSTRACT

Wavelet Regression using MapReduce and Analysis of
Multiple Sclerosis Clinical Data

by

Hanyu Song

Department of Department of Statistical Science
Duke University

Date: _____

Approved:

Li Ma, Supervisor

Meng Li

Cliburn Chan

An abstract of a thesis submitted in partial fulfillment of the requirements for
the degree of Master of Science in the Department of Department of Statistical
Science
in the Graduate School of Duke University
2017

Copyright © 2017 by Hanyu Song
All rights reserved except the rights granted by the
Creative Commons Attribution-Noncommercial Licence

Abstract

Two topics, one related to scalable methods and the other on applications of statistical methods to clinical data are studied in this thesis. Motivated by the increasingly common large datasets, we propose a novel parallel MapReduce framework for implementation of wavelet regression and shrinkage in Chapter 2. The new method consists of two stages of MapReduce, with discrete wavelet transform performed in the first stage and regression and shrinkage in the second. In comparison to the conventional one (implemented without parallelization), computational complexity analysis and numerical experiments show the proposed algorithm can be computationally superior when the response dimension (or number of time measurements) is big.

In Chapter 3, we provide a preliminary analysis of a Multiple Sclerosis (MS) clinical dataset provided by Biogen, which comprises 579 actively managed MS patients enrolled at a single center for up to 5 years. MS is a chronic heterogenous disease with unpredictable trajectory for many patients. Some present clinical symptoms very early on but progress slowly, while others do not have symptoms till later in life but thereafter progress rapidly. Therefore, we are interested in predicting the trajectory of a patient based on current marker profiles. This can serve as a powerful guide for a proper early treatment to delay the build up of irreversible damages and thus improve long-term health of patients. An explanatory data analysis and logistic regression of disease courses and predictive modeling of Expanded Disability

Status Scale (EDSS) scores are provided in this thesis. We notice that, using logistic regression, prediction of disease courses reaches 90% in terms of out-of-sample accuracy. While conducting predictive modeling of EDSS scores, we find the linear mixed-effects model yields the highest accuracy among all the models considered, and its model assumptions are roughly satisfied as evidenced by the diagnostic plots.

To my parents, husband Azeem Zaman and Mr. Lion

Contents

Abstract	iv
List of Tables	ix
List of Figures	x
List of Abbreviations and Symbols	xi
Acknowledgements	xiii
1 Introduction	1
2 Wavelet regression using MapReduce	3
2.1 Introduction	3
2.2 Wavelet transform	5
2.3 Wavelet regression	9
2.4 Wavelet regression using MapReduce	11
2.5 Time Complexity Analysis	14
2.5.1 Complexity of the conventional algorithm	14
2.5.2 Complexity of wavelet regression using MapReduce	17
2.6 Illustrative simulation	21
2.6.1 Comparison: computational performance	22
2.7 Concluding remarks	23
3 Analysis of Multiple Sclerosis Clinical Data	24
3.1 Introduction	24

3.2	Dataset	26
3.2.1	Demographic Characteristics	26
3.2.2	Disease-Modifying Treatments	27
3.2.3	Clinical variables	27
3.2.4	MRI assessments	31
3.2.5	SIENA assessments	31
3.2.6	SIENAX assessments	31
3.3	EDA and Data Preprocessing	31
3.3.1	Data Preprocessing	31
3.3.2	EDA	33
3.4	Model fitting	35
3.4.1	Prediction of disease courses: logistic regression	35
3.4.2	Prediction of EDSS: five models	35
3.5	Concluding remarks	37
	Bibliography	39
	Biography	41

List of Tables

2.1	Computational complexity of OLS estimates ($n > p$)	16
2.2	Computational complexity of the conventional wavelet regression (Note that the complexity of step 3 depends on the shrinkage rules applied. <i>SureShrinkage</i> is used for illustration.)	17
2.3	Time in seconds for MapReduce and conventional algorithm to run with T time measurements. Here $n = 1000$ and $p = 20$ for all trials. Reported results are trimmed mean using five out of a total of ten replications.	22
3.1	Age distribution of 579 MS patients at enrollment	26
3.2	Distribution of disease courses at enrollment	29
3.3	Distribution of age of onset of the 579 MS patients	31
3.4	Grouping of disease course	32
3.5	Variable <code>year</code> created from <code>Visit</code>	32
3.6	In-sample accuracies of EDSS prediction	37

List of Figures

2.1	Time and frequency resolutions associated with STFT (Source: Gao and Yan (2011))	6
2.2	Time and frequency resolutions of the wavelet transform ($s_2 = 2s_1$) (Source: Gao and Yan (2011))	6
2.3	How the wavelet coefficient vector is interpreted at various scales in the original time domain (Source: Alsberg et al. (1998))	8
2.4	Wavelet-shrinkage paradigm	10
2.5	First-stage MapReduce: Row-wise DWT of response \mathbf{Y}	15
2.6	Second-stage MapReduce: Wavelet regression, shrinkage and inverse DWT	16
3.1	Development pattern of disease activities (Source: Lublin et al. (2014))	28
3.2	Relationships between demographic and a few clinical variables. Green represents the disease course PMS, and red represents the disease course PMS	33
3.3	Relationships between clinical variables. Green represents the disease course PMS, and red represents RMS.	34
3.4	Diagnostic plots of multiple linear regression	38

List of Abbreviations and Symbols

Symbols

Y	The response matrix
$\mathbf{y}_{(j)}$	The j^{th} column of matrix Y
\mathbf{y}_i	The i^{th} row of matrix Y
X	The design matrix
n	Sample size
p	Number of features in a design matrix
t	Number of columns in a response matrix, if applicable
s	$s \in \mathbb{R}^+$, scaling parameter
τ	translation parameter
W	Wavelet transform represented by a matrix
$\tilde{\mathbf{Y}}$	Result of row-wise discrete wavelet transform of Y
M	Number of mappers in a MapReduce framework
R	Number of reducers in a MapReduce framework

Abbreviations

DWT	Discrete Wavelet Transform
FWT	Fast Wavelet Transform
flops	Floating-point Operations Per Second
LASSO	Least Absolute Shrinkage and Selection Operator

MS	Multiple Sclerosis
CNS	Central Nervous System
MRI	Magnetic Resonance Imaging
DMT	Disease-modifying Treatment
EPIC	Expression/genomics, Proteomic, Imaging, and Clinical
EDSS	Expanded Disability Status Scale
MSFC	Multiple Sclerosis Functional Composite
T25W	Timed 25-Foot Walk
9-HPT	9-Hole PEG Test
EDA	Explanatory Data Analysis
CIS	Clinically Isolated Syndrome
PPMS	Primary Progressive MS
PRMS	Progressive-relapsing MS
RRMS	Relapsing-remitting MS
SPMS	Secondary progressive MS
MSFC	MS Functional Composite
SIENA	Structural Image Evaluation, using Normalization, of Atrophy
PMS	PPMS, SPMS or PRMS
RMS	RRMS or CIS

Acknowledgements

Joining Duke University for the Master's program in Statistics is so far one of the best decisions in my life. I fortunately met many incredible faculty members, guiding me and helping me achieve my dream. Beyond invaluable academic guidance, their true care for me as a student has also strongly motivated me. First of all, I would like to express my deep gratefulness to my thesis advisor Prof. Li Ma. At our first meeting, he warmly invited me to his regular group meeting and encouraged me to give a presentation one day, which made me feel sense of belonging. His way of approaching problems influences my statistical thinking, and his work spirits strongly motivate me. But for his insightful comments, this thesis could be far worse. I would also like to thank Prof. David Dunson, my academic advisor, who led me to the fun world of statistics by teaching me the first course in Bayesian statistics. His intuitive explanations bring statistics to life. I always feel growing enthusiasm after a conversation with him about my research. I thank Dr. Meng Li for introducing me to collaboration with Biogen, which exposes me to a real application of statistics. He is both an approachable mentor and friend. I have also benefited tremendously from the talks with prof. Surya Tokdar, director of graduate study, for his understanding and genuine advice for Azeem and me.

Finally, I want to thank Dr. Xiangyu Wang and Azeem Zaman. Dr. Xiangyu Wang was my collaborator, and indeed my mentor, for my first research experience. Busy preparing for his dissertation defense, he was patient with all my questions. I

could not have been admitted to the PhD program at Duke but for his help. Azeem Zaman, whom I met on my first day at Duke, is my best friend and husband. His company and mental support dissolve my pain on the way of exploration. Discussion with him also contributed tremendously to this thesis. We will continue to work hard together in pursuit of a PhD in statistics.

1

Introduction

In modern science and technology applications, it has become routine to collect complex datasets with large sample sizes and data dimensions. In such cases, the classical feature selection methods may be impractical due to high computational costs. Consider wavelet regression and shrinkage. *Wavelet regression* can be regarded as a regression with wavelet transform performed beforehand as a pre-processing step. It has wide applications such as image processing, and when the number of pixels becomes huge, it will experience computational bottlenecks. In response to this, we focus on datasets with large data dimensions and study the analogue of wavelet regression and shrinkage implemented in a parallel framework. Such procedures often incur communication cost across computers at each iteration or whenever data must be transmitted between computers, driving the inefficiency of distributed algorithms. In this thesis, we propose a MapReduce algorithm as a solution to implementing wavelet regression and shrinkage for datasets with large dimensions in the response matrix. The complexity analysis shows when the number of covariates p or/and the dimension of response t are huge, the proposed new algorithm can be computationally superior to the conventional one, especially when the communication cost in

MapReduce is small for the large dataset given in Chapter 2 as an example.

Motivated by the practical applications of statistics in clinical trials, we study a Multiple Sclerosis (MS) clinical dataset provided by Biogen, which comprises 579 actively managed MS patients enrolled at a single center for up to 5 years. MS is a chronic heterogenous disease with unpredictable trajectory for many patients. Some present clinical symptoms very early on but progress slowly, while others do not have symptoms till later in life but thereafter progress rapidly. Therefore, we are interested in predicting the trajectory of a patient based on current marker profiles. This can serve as a powerful guide for a proper early treatment to delay the build up of irreversible damages and thus improve long-term health of patients. In Chapter 3, an explanatory data analysis, logistic regression of disease courses and predictive modeling of Expanded Disability Status Scale (EDSS) scores, are provided. While conducting predictive modeling of EDSS scores, we find the linear mixed-effects model yields the highest accuracy among all the models considered, and its model assumptions are roughly satisfied as evidenced by the diagnostic plots.

The rest of the thesis is organized as follows. In Chapter 2, we present the MapReduce framework for implementation of wavelet regression and shrinkage and provide a comprehensive time complexity analysis to compare the performance of existing and the proposed methods. Chapter 3 introduces our study of MS clinical data.

Wavelet regression using MapReduce

2.1 Introduction

In this chapter, we develop a MapReduce algorithm for implementing wavelet regression. *Wavelet regression* can be considered a regression with wavelet transform performed beforehand as a pre-processing step. Wavelet transform is effective in preserving part of both time and frequency-like information from a time-varying input vector (Alsberg et al. (1998)). Through variations of the scale and time translation of a mother wavelet function, the *discrete wavelet transform (DWT)* often compacts the energy of a signal into a few wavelet coefficients with large amplitudes and spreads the energy of noise over a large number of wavelet coefficients with small amplitudes. Such a representation can separate signals and noise, or in other words, helps identify variables as being part of short- or long-scale features in the wavelet domain, and is thus useful for feature selection (Alsberg et al. (1998)).

Suppose we have an independent multivariate sample $\mathbf{y}_1, \dots, \mathbf{y}_n$. For each observation \mathbf{y}_i , we have p covariates (including the intercept term) \mathbf{x}_i . Let $\mathbf{Y} = (\mathbf{y}_1^\top, \dots, \mathbf{y}_n^\top)^\top$ and $\mathbf{X} = (\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top)^\top$. For example, \mathbf{y}_i can be blood pressure

measurements taken from patient i at t equally-spaced time points. Data collected from n patients form an n by t matrix \mathbf{Y} . If we also observe the patients' demographic and clinical characteristics, e.g. age of onset, smoking habits and family history, then data of p covariates span an n by p design matrix \mathbf{X} . Applying a DWT to each row of matrix \mathbf{Y} allows us to separate short- or long- scale patterns in blood pressure changes. After DWT, we can perform a multivariate linear regression of the wavelet coefficients on the patients' covariates and shrinkage of fitted coefficients in the wavelet domain. An inverse DWT can map shrunk estimates into the original scale, i.e. time domain. Based on the estimates in the original scale, we can infer the effects of a covariate, e.g. age of onset on the mean function of blood pressure over time.

Unfortunately, wavelet regression and shrinkage experiences its computational bottleneck amid the arrival of “big” datasets era. In some real world applications, the datasets can be very large in the sample size n , the response dimension t or/and number of features p . Thus, a single workstation might be incapable of handling such big data. One potential solution is to down scale the problem size by first partitioning the dataset into subsets and fitting models using distributed algorithms. But this procedure often incurs communication cost across computers at each iteration or whenever data must be transmitted between computers, driving the inefficiency of distributed algorithms.

In this chapter, we propose a two-stage MapReduce framework to implement wavelet regression. The first-stage MapReduce involves DWT performed on each observation through an embarrassingly parallel algorithm with almost no communication cost. The second-stage MapReduce fits a set of regression models and performs shrinkage and inverse DWT, which incurs communication cost during the shuffling phase. However, as we will see, transmitting a dataset with $t = 10^6$ and $p = 10^4$ only takes less than 2 minutes over gigabit Ethernet. This suggests promising applications

of this new paradigm.

The rest of the chapter is organized as follows. In Section 2.2, we introduce wavelet transform. Section 2.3 briefly reviews wavelet regression to be analyzed in this chapter. In Section 2.4, we depict the details of the proposed two-stage MapReduce algorithm. Section 2.5 provides a theoretical complexity analysis and shows the superior performance of the new parallelized framework. Section 2.6 evaluates the performance of the new algorithm via extensive numerical experiments.

2.2 Wavelet transform

The development of wavelet transform was motivated by the limitations of the short-time Fourier transform (STFT). STFT was actually a solution to the problem in Fourier transform, which turns a signal in the time domain to another function in the frequency domain (Hubbard (1948)). In other words, Fourier transform captures absolute precision of frequency but zero resolution on temporal spread. To address this limitation, STFT was introduced as a time-localized Fourier transform, which employs a sliding window function of fixed length and performs Fourier transform within each window consecutively (Gao and Yan (2011)). The fixed window leads to uniform resolution in the frequency domain, an undesirable feature for many applications. For example, low frequencies can be imprecisely depicted with short windows, whereas short pulses are poorly localized in time with long windows. Also, selection of a suitable window size is generally challenging given an unknown frequency content of some signal (Gao and Yan (2011)). Alfred Haar shed light on this problem in his dissertation in 1909 and proposed the wavelet transform as a solution, which enables variable window sizes in analyzing different frequency components within a signal. Figure 2.1 and Figure 2.2 illustrate the difference between STFT and wavelet transform.

Figure 2.2 demonstrates that time and frequency resolution are inversely related

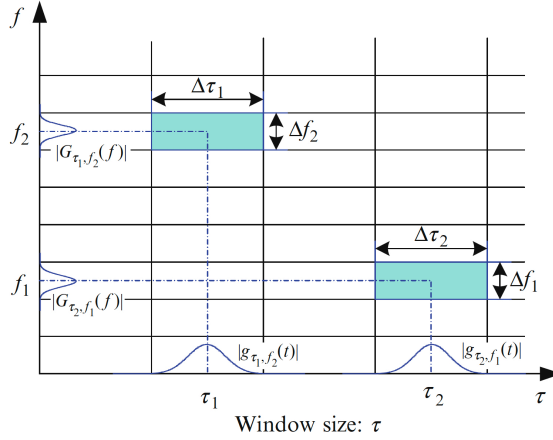


FIGURE 2.1: Time and frequency resolutions associated with STFT (Source: Gao and Yan (2011))

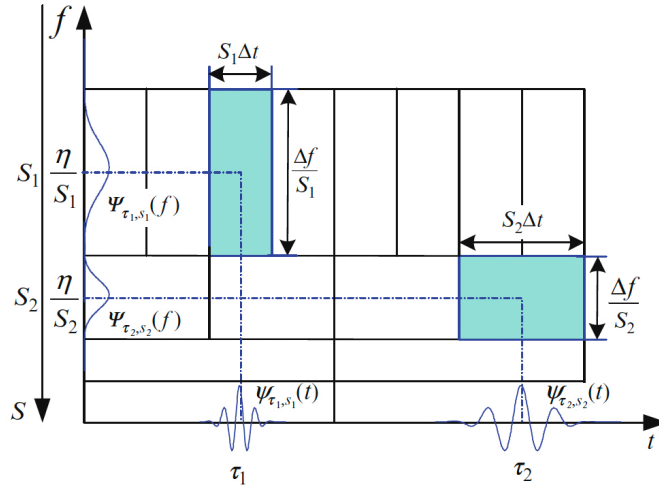


FIGURE 2.2: Time and frequency resolutions of the wavelet transform ($s_2 = 2s_1$) (Source: Gao and Yan (2011))

while varying the scale. An increase of the scale from s_1 to s_2 results in halving the time resolution and doubling the frequency resolution, as is signaled by the change in the width of time and frequency window respectively (Gao and Yan (2011)). Specifically, at a lower scale s_1 , the wavelet transform is better at capturing rapid changing details or high frequency components with a compressed basis function; whereas at a higher scale s_2 , it performs better in decomposing slowly changing coarse features or low frequency components. Therefore, the wavelet coefficients reflect the

patterns of variation e.g. a sharp peak or discontinuity in the original time domain. As such, we can smooth the data by thresholding the wavelet coefficients and then reconstructing the shrunk coefficients to the time domain.

More formally, in continuous wavelet transform (CWT), a given signal of finite energy is projected on a continuous family of frequency bands or subspaces of various scales in $\mathcal{L}_2(\mathbb{R})$. For instance the signal may be represented on each subspace of scale s for all $s > 0$. For each s , the subspace is generated by the translations and rescales of one single function $\psi(x) \in \mathcal{L}_2(\mathbb{R})$, called *the mother wavelet*:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right), \quad (2.1)$$

where $s \in \mathbb{R}^+$ defines the scaling parameter and $\tau \in \mathbb{R}$ represents the translation parameter. s reflects the time and frequency resolutions of the scaled mother wavelet $\psi\left(\frac{t-\tau}{s}\right)$ and τ represents its translation along the time axis.

The projection of a function $f \in \mathcal{L}_2(\mathbb{R})$ onto the subspace of s has the form

$$f_s(t) = \int \mathcal{WT}_f(s, \tau) \psi_{s,\tau}(t) d\tau \quad (2.2)$$

with the *wavelet transform* defined by

$$\mathcal{WT}_f(s, \tau) = \langle f, \psi_{s,\tau} \rangle = \int f(t) \overline{\psi_{s,\tau}(t)} dt, \quad (2.3)$$

The realization of $\mathcal{WT}_x(s, \tau)$ is the *wavelet coefficient* of a signal x at scale s and time shift τ .

Using a continuous scale is computationally intractable and thus *discrete wavelet transform* (DWT) comes into play, which minimizes the transformation using discrete values of s and τ and still guarantees its invertibility. The parameters defined by

$$s = 2^j \text{ and } \tau = k2^j, \text{ for all } k, j \in \mathbb{N}.$$

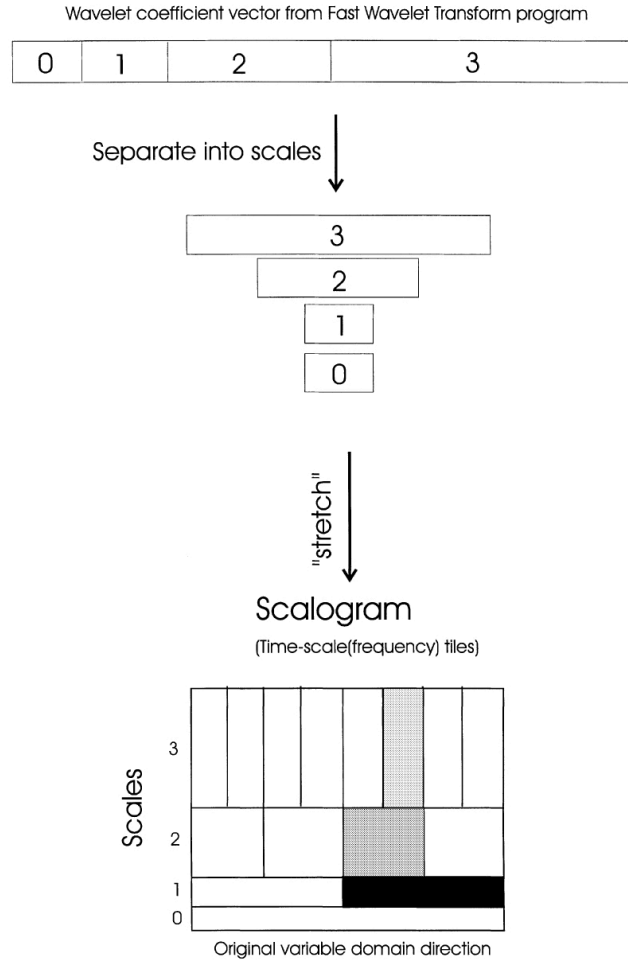


FIGURE 2.3: How the wavelet coefficient vector is interpreted at various scales in the original time domain (Source: Alsberg et al. (1998))

produces the minimal basis (Vidakovic (1955)). Essentially, DWT maps an input data vector of length 2^{J+1} ($J \in \mathbb{Z}^+$) from the time domain to another vector of the same size in the wavelet domain. Such transform can be represented by a matrix \mathbf{W} with

$$\tilde{\mathbf{z}} = \mathbf{W}\mathbf{z},$$

where \mathbf{z} is the input signal vector and $\tilde{\mathbf{z}}$ is the wavelet coefficient vector. Figure 2.3 shows the output wavelet coefficients are ordered according to the scales s .

On each scale, they are ordered according to the number of shifts τ along the

time axis. The wavelet coefficients can be represented by

$$\tilde{\mathbf{z}} = (c, \mathbf{d}_0^\top, \mathbf{d}_1^\top, \dots, \mathbf{d}_J^\top)^\top,$$

where c is the coarsest *father wavelet coefficient* or *scaling function coefficient* necessary for recovering the input vector given wavelet coefficients, and \mathbf{d}_j consists of the wavelet coefficients $\mathbf{d}_{j,k}$ at scale $s = 2^j$, for $k = 0, \dots, 2^j - 1$ and $j = 0, \dots, J - 1$.

Same as the order of wavelet coefficients, the corresponding transform matrix \mathbf{W} can be represented by

$$\mathbf{W} = (\mathbf{C}, \mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_J)^\top,$$

where \mathbf{C} is a scaling function to produce c , and \mathbf{W}_j are wavelet basis at scale $s = 2^j$ ($j = 0, 1, \dots, J$) for finding \mathbf{d}_j . \mathbf{W} is orthogonal or “close to orthogonal” depending on the boundary condition (Vidakovic (1955)). We restrict \mathbf{W} to orthogonal matrices in this chapter. Wavelet coefficients computed via a matrix multiplication operation $\tilde{\mathbf{z}} = \mathbf{W}\mathbf{z}$ requires $O(n^2)$ operations. This was computationally challenging until Mallat (1989) gave an efficient recipe, called fast wavelet transform (FWT) or pyramidal algorithm, which takes only $O(n)$ to produce results.

Due to the orthogonality of \mathbf{W} , the original vector can be easily reconstructed by:

$$\mathbf{z} = \mathbf{W}^\top \tilde{\mathbf{z}}.$$

2.3 Wavelet regression

Wavelet regression can be considered a regression with wavelet transform performed beforehand as a pre-processing step. As introduced in Section 2.3, wavelet transform is effective in preserving part of both time and frequency-like information from a time-varying input vector. Such a transform helps identify variables as being part of short- or long-scale features in the wavelet domain, and is thus useful for feature selection (Alsberg et al. (1998)). In particular, variables in short wavelet scale regions tend

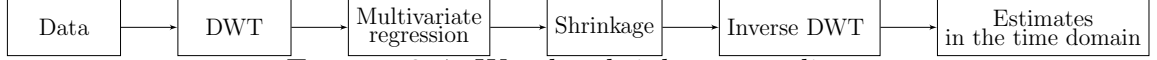


FIGURE 2.4: Wavelet-shrinkage paradigm

to indicate high-frequency components, e.g. sharp peaks; variables in long wavelet scale regions tend to manifest the opposite pattern.

A simple wavelet shrinkage recipe is illustrated in Figure 2.4.

Recall our data structure: response matrix $\mathbf{Y} = (\mathbf{y}_1^\top, \dots, \mathbf{y}_n^\top)^\top$ and design matrix $\mathbf{X} = (\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top)^\top$. Denote the row vectors of response \mathbf{Y} by \mathbf{y}_i for $i = 1, \dots, n$. The wavelet regression analyzed in this chapter can be formulated in two steps:

Step 1 Row-wise wavelet transform

1. Apply (row-wise) DWT to the response matrix \mathbf{Y} on each observation.
2. Save the output of wavelet coefficients $\tilde{\mathbf{y}}_i$ in the original row order as $\tilde{\mathbf{Y}}$.

Using the notations in Section 2.2, such a transform can be represented by

$$\tilde{\mathbf{Y}} = \mathbf{Y}\mathbf{W}^\top$$

Note that $\tilde{\mathbf{Y}}$ has the same dimension n by t as \mathbf{Y} . Denote the row vectors and column vectors of wavelet coefficients matrix by $\tilde{\mathbf{y}}_i$ and $\tilde{\mathbf{y}}_{(j)}$ respectively, for $i = 1, \dots, n$ and $j = 1, \dots, t$.

Step 2 Multivariate linear model

Consider a multivariate linear model when $n > p$.

$$\tilde{\mathbf{Y}} = \mathbf{X}\boldsymbol{\beta} + \mathbf{E},$$

where $\tilde{\mathbf{Y}} \in \mathbb{R}^{n \times t}$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, $\boldsymbol{\beta} \in \mathbb{R}^{p \times t}$ and $\mathbf{E} \in \mathcal{S}_n^+ = (\mathbf{e}_{(1)}^\top, \dots, \mathbf{e}_{(n)}^\top)^\top$ with $\mathbf{e}_{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i)$. This is essentially a multiple linear regression of each $\tilde{\mathbf{y}}_i$ on \mathbf{X} with:

$$\tilde{\mathbf{y}}_{(i)} = \mathbf{X}\boldsymbol{\beta}_{(i)} + \mathbf{e}_{(i)},$$

When $n \leq p$, we can fit a high-dimensional regression model, such as LASSO or ridge regression. One can easily show that when $n > p$, the least squares estimate of $\tilde{\beta}$ is:

$$\hat{\tilde{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1}(\mathbf{X}^\top \tilde{\mathbf{Y}}).$$

When $n \leq p$, the ridge regression estimate of $\tilde{\beta}$ is:

$$\hat{\tilde{\beta}} = (\mathbf{X}_c^\top \mathbf{X}_c + \lambda \mathbf{I}_p)^{-1}(\mathbf{X}_c^\top \tilde{\mathbf{Y}}_c),$$

where \mathbf{X}_c and $\tilde{\mathbf{Y}}_c$ are column-centered \mathbf{X} and $\tilde{\mathbf{Y}}$. In particular, $\tilde{\mathbf{Y}}_c = (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n})\tilde{\mathbf{Y}}$.

After pruning estimates $\hat{\tilde{\beta}}$ in the wavelet domain, we can recover estimates in the original domain using the orthogonality of \mathbf{W} via an inverse DWT

$$\hat{\beta}_0 = \hat{\tilde{\beta}}_0 \mathbf{W},$$

where $\hat{\tilde{\beta}}_0$ are the shrunk estimates of $\hat{\tilde{\beta}}$ in the wavelet domain.

The regression steps can be summarized below:

1. Regress each $\tilde{\mathbf{y}}_i$ on \mathbf{X} ,
2. Save the set of fitted coefficients $\{\hat{\beta}_i\}$ and summary statistics,

Step 3 Wavelet shrinkage

Shrink estimates $\hat{\tilde{\beta}}$ in the wavelet domain using some thresholding rules.

Step 4 Inverse DWT of fitted coefficients

Apply inverse DWT to obtain shrunk estimates $\hat{\beta}_0$ in the original (or time) domain.

2.4 Wavelet regression using MapReduce

In view of the parallel nature of the above algorithm, we propose its analogue implemented in the framework of MapReduce. The new algorithm consists of two stages

of MapReduce, with DWT performed in the first stage and regression and shrinkage in the second. Throughout this section, denote any *row index* and *column index* of a matrix by i and j .

During the first stage, the input *key-value* pairs to the Map tasks are (row index i of response \mathbf{Y} , row vector \mathbf{y}_i). The outputs from the mappers are (row index i of response \mathbf{Y} , wavelet coefficients of \mathbf{y}_i). What remains after DWT is rearrangement of the wavelet coefficients according to the row index of the (original) response \mathbf{Y} . Such a procedure can be conducted on the master computer, after it collects the outputs from all the Map tasks, by putting the wavelet coefficients vector in the desired place, as indicated by the row index of response \mathbf{Y} . Therefore, the tasks in the first MapReduce can be completed in parallel and reducers are *not* necessary. This design avoids the *shuffling*, the process of transferring data from mappers to reducers and thus saves communication cost which could otherwise become the dominant component of the computation time.

In the second stage, the input *key-value* pairs to the Map tasks are [column index j of wavelet coefficient matrix $\tilde{\mathbf{Y}}$, (column vector $\tilde{\mathbf{y}}_{(j)}$, design matrix \mathbf{X})]. After regressing each $\tilde{\mathbf{y}}_{(j)}$ on \mathbf{X} , we obtain fitted coefficients $\hat{\beta}_{(j)}$. The outputs from the mappers are key-value pairs:

$$[i, \{j, \text{column vector } \hat{\beta}_{ij}\}],$$

where i and j represent a row index and column index of the fitted coefficient matrix $\hat{\beta}$ respectively, for $i = 1, \dots, n$ and $j = 1, \dots, t$. However, the specific design of reducers depends on the shrinkage and other statistical procedures. For example, if we use hard or soft thresholding, we can shrink the estimates $\hat{\beta}$ at each wavelet node independently, and this procedure can be conducted within the mappers. In this case, the reducers are only responsible for inverse DWT of each row vector $\hat{\beta}_i$. However, if we employ *SureShrink* (Donoho and Johnstone (1995)), a shrinkage

scheme that fits a (decomposition) level-dependent threshold, then for each row of $\hat{\beta}$, we need information from all columns to decide on the threshold. In this case, the reducers will conduct both shrinkage and inverse DWT. As we will see later, the shuffling phases of these two designs are indeed similar.

Based on the above discussions, the new framework can be fully specified as follows:

First-stage MapReduce (See Figure 2.5)

Map

Input: (Keys, Values) = (row index i of \mathbf{Y} , row vectors \mathbf{y}_i)

1. Partition \mathbf{Y} horizontally (in the sample space) into $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_m$ and distribute to m mappers,
2. On each subset \mathbf{Y}_j within a mapper, perform (row-wise) DWT by key using predefined wavelet basis.

Output: (Keys, Values) = (row index i of \mathbf{Y} , wavelet coefficient vectors $\tilde{\mathbf{y}}_i$)

Second-stage MapReduce (See Figure 2.6)

Map

Input: [keys, values] = [column index j of wavelet coefficient $\tilde{\mathbf{Y}}$, (column vectors $\tilde{\mathbf{y}}_{(j)}$, design matrix \mathbf{X})]

1. Partition $\tilde{\mathbf{Y}}$ vertically (in the wavelet domain) into $\tilde{\mathbf{Y}}_1, \tilde{\mathbf{Y}}_2, \dots, \tilde{\mathbf{Y}}_k$ and distribute to k mappers.
2. On each subset $\tilde{\mathbf{Y}}_j$ within a mapper, regress each column of $\tilde{\mathbf{Y}}_j$ on \mathbf{X} and save the fitted coefficient estimates as $\hat{\beta}_{(j)}$.

3. Shrink $\hat{\beta}_{(j)}$ and compute uncertainty quantifications (e.g. standard errors, t statistics and p -values) within each mapper. (Note that the shrinkage is performed in this step if it can be performed at each wavelet node independently.)

Output: [keys, values]= [i , (j , column vectors $\hat{\beta}_{ij}$)], where i and j represent a row index and column index of the fitted coefficient matrix $\hat{\beta}$ respectively.

Shuffle

Sort the *output* from the mappers by *row index* i of $\hat{\beta}$ and transfer the data to the reducers as inputs.

Reduce

1. Sort the inputs with the same row index i according to *column index* j to form the row vectors $\hat{\beta}_i$.
2. Perform shrinkage or compute statistical quantities. (If shrinkage is *already* done in the Map tasks, omit it in this step.)
3. Apply (row-wise) inverse DWT to the row vectors $\hat{\beta}_i$.

2.5 Time Complexity Analysis

In this section we analyze the time complexity of the conventional wavelet regression and shrinkage algorithm and the proposed MapReduce-based framework.

2.5.1 Complexity of the conventional algorithm

Recall Section 2.3, step 1 applies DWT to each observation in the response matrix \mathbf{Y} . Using FWT (or pyramidal algorithm), the time complexity is $O(t)$, i.e. linear with respect to the size of the input vector. This operation is performed n times to each row and thus incurs cost $O(nt)$.

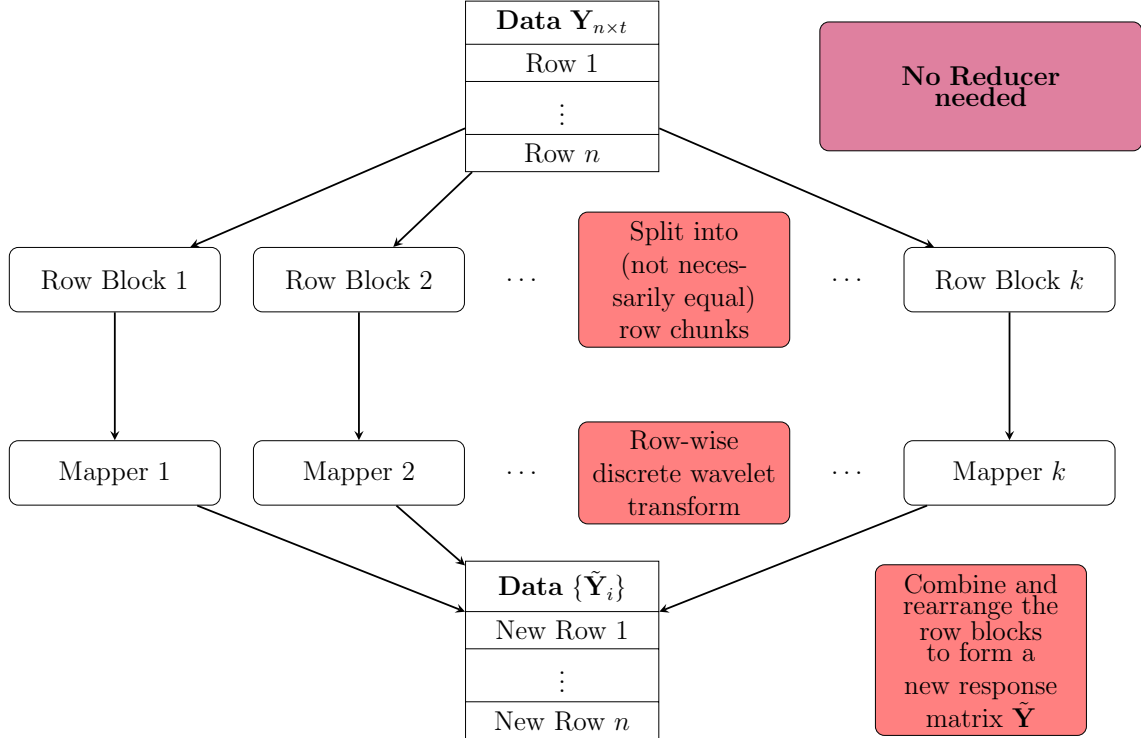


FIGURE 2.5: First-stage MapReduce: Row-wise DWT of response \mathbf{Y}

In step 2, assume $n > p$ and we fit linear models. Then we find OLS estimates by solving *normal equations*

$$\mathbf{X}^\top \mathbf{X} \tilde{\beta}_{(j)} = \mathbf{X}^\top \tilde{\mathbf{y}}_{(j)}, \text{ for } j = 1, 2, \dots, t$$

Estimates $\tilde{\beta}_{(j)}$ can be obtained via efficient matrix decomposition algorithms such as Cholesky decomposition, QR factorization and singular value decomposition (SVD). Cholesky is asymptotically the fastest but numerically unstable for nearly singular or rank deficient matrices ($\mathbf{X}^\top \mathbf{X}$). In comparison, QR factorization improves numerical stability but leads to heavier computation cost and difficulty in exploiting sparsity for large sparse \mathbf{X} . In our analysis, we choose QR factorization given its good performance and prevalence in practice. In particular, the most widely used QR factorization method is Householder algorithm, whose complexity is $2np^2 - (2/3)p^3$ floating-point operations per second (flops). Table 2.1 summarises the computational complexity of OLS estimates.

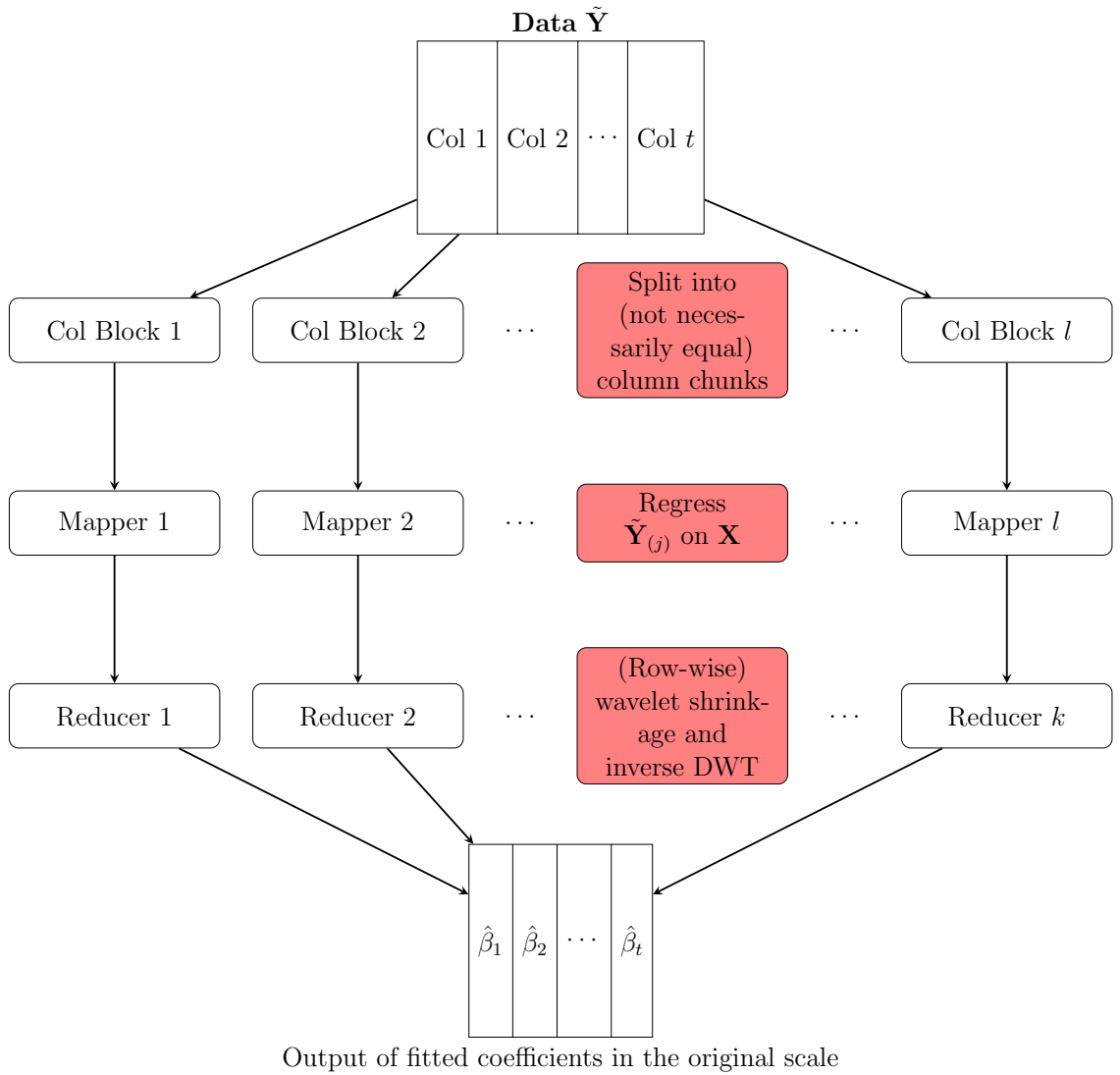


FIGURE 2.6: Second-stage MapReduce: Wavelet regression, shrinkage and inverse DWT

Table 2.1: Computational complexity of OLS estimates ($n > p$)

Operation	Complexity (flops)
a. Multiplying \mathbf{X}^\top by \mathbf{X}	$2np^2 - p^2$
b. Multiplying \mathbf{X}^\top by $\mathbf{y}_{(j)}$	$2np - n$
c. Solving the j th normal equation	$2np^2 - (2/3)p^3$

Table 2.2: Computational complexity of the conventional wavelet regression (Note that the complexity of step 3 depends on the shrinkage rules applied. *SureShrinkage* is used for illustration.)

Procedure	Complexity
Step 1. Row-wise DWT of response \mathbf{Y}	$O(nt)$
Step 2. Multivariate linear regression	$O(2ntp^2 - (2/3)tp^3)$
Step 3. <i>SureShrinkage</i> *	$O(pt \log t)$
Step 4. Inverse DWT of fitted coefficients $\tilde{\beta}$	$O(pt)$

In step 2, fitting t linear models involves t repetitive operations of b and c, which consist of $2ntp - nt + 2ntp^2 - (2/3)tp^3$ flops in total. Combining operation a, the total complexity is $O(2ntp^2 - (2/3)tp^3)$. Step 3 is thresholding fitted coefficients $\tilde{\beta}_{p \times t}$, which requires as much as $O(t \log t)$ calculations for each row vector $\tilde{\beta}_i$ of length t and thus $O(pt \log t)$ calculations in total. Step 4 is row-wise inverse DWT of fitted coefficients $\tilde{\beta}_{p \times t}$, resulting in $O(pt)$ operations in total. We observe from Table 2.2 that step 2, in particular solving normal equations, dominates the computation cost. Combining all the steps the algorithm has complexity $O(2ntp^2 - (2/3)tp^3 + pt \log t)$ for the entire dataset. Note that $O(pt \log t)$, a seemingly smaller cost than $O(2ntp^2 - (2/3)tp^3)$ is included, because t can be enormous in comparison to n and p in some applications, e.g. t is the total number of pixels in a high-definition image.

2.5.2 Complexity of wavelet regression using MapReduce

MapReduce allows for parallelism and thus requires less time to execute each task, however its computational performance can be seized by the communication cost, the process of moving data among tasks. Afrati et al. (2013) argued that transporting the outputs of Map tasks to their proper Reduce tasks contributes to the largest portion of the total time spent. Thus, we shall investigate the communication cost in addition to the processing cost, and look for a solution that optimally trade the communication cost against the degree of parallelism.

The *communication cost of a task* is defined by the size of the input to the task, measured in *bytes*, and the *communication cost of an algorithm* is the sum of the communication cost of all the tasks implementing that algorithm. Suppose the entries in all the input data matrices are of double precision; if the matrix has m rows and n columns, the size of the input is $8mn$ bytes. Assume the number of mappers and reducers are M and R respectively.

First-stage MapReduce

1. Processing cost

In the first-stage MapReduce, key-value pair is $(i, \text{row vector } \mathbf{y}_i)$. Recall the complexity for DWT is $O(t)$ using Mallat (1989)'s FWT. n such operations make the total Map cost $O(nt)$ and per Mapper cost $O(nt/M)$. No reducers are necessary as discussed before.

2. Communication cost

The cost from shuffling, which could otherwise be dominant, is zero due to the absence of reducers. Communication only occurs when inputting a file to the Map tasks and outputting results to the master computer. In our first-stage MapReduce, the input to the Map tasks is the size of \mathbf{Y} , $8nt$ bytes. The sum of the outputs of the Map tasks—the size of the transformed response matrix $\tilde{\mathbf{Y}}$ —is no larger than the input \mathbf{Y} since DWTs introduce sparsity to the wavelet coefficients. Therefore, the communication, occurs between the disk of the master computer and the memory of the mappers, is $16nt$ bytes in total.

Second-stage MapReduce

1. Processing cost

Recall our discussion in Section 2.5.1, the total Map cost is $O(2ntp^2 - (2/3)tp^3)$ and per Mapper cost is $O(2ntp^2/M - (2/3)tp^3/M)$. If *SureShrinkage* and inverse

DWT are performed in the reducers, the total Reduce cost is $O(pt \log t + pt)$, as indicated by Table 2.2; per Reducer cost is $O(pt \log t/R + pt/R)$, which is essentially $O(pt \log t/R)$.

2. Communication cost

First consider the communication cost from inputs to mappers and outputs to the master computer. They include wavelet coefficients matrix $\tilde{\mathbf{Y}}$ ($8nt$ bytes), design matrix \mathbf{X} ($8np$ bytes), and shrunk coefficients $\hat{\beta}_0$ in the original scale ($8pt$ bytes) and are of size $8nt + 8np + 8pt$ bytes in total.

Next we consider the communication cost from the shuffling phase. Before discussing this, we introduce two parameters, the reducer size q and the replication rate r , that characterize MapReduce algorithms. According to Leskovec et al. (2014), the reducer size is defined as “the upper bound on the number of values that are allowed to appear in the list associated with a single key”. For example, when counting word frequency in a document, the reducer size can be the total number of words. Replication rate is defined as the number of key-value pairs created by mappers from their inputs, divided by the total number of inputs (Afrati et al. (2013)), which can be interpreted as a measure of communication cost per input.

Choosing a sufficiently small reducer size, if possible, is beneficial in two ways. First, it could ensure that the computation associated with a single reducer is executed entirely in the main memory of the compute node where its Reduce task is located, which avoids moving data between main memory and disk (Leskovec et al. (2014)). Second, it allows us to create so many Reduce tasks that only a few (or even one) are assigned to each reducer, leading to a high degree of parallelism and thereby a low wall-clock time (Leskovec et al. (2014)). However, such a desired reducer size is extremely hard to achieve in light of

the tradeoff between the reducer size and the replication rate.

In the second-stage MapReduce, the key-value pairs produced by the mappers are [row index i of fitted coefficients $\hat{\beta}$, (column index j , $\hat{\beta}_{ij}$)]. Since the dimension of $\hat{\beta}$ is p by t , the replication rate r is 1 given by

$$\frac{\text{number of key-value pairs}}{\text{number of inputs}} = \frac{pt}{pt} = 1$$

For the reducers, the associated value list with a single key has p elements, i.e. number of columns of $\hat{\beta}$. Thus, the reducer size q is t .

Therefore, the total number of bytes communicated from Map tasks to Reduce tasks is pt (for the number of inputs) times 1 (for the replication), times 8 bytes (for the size of each data entry), which equals to $8pt$ bytes. If $t = 10^6$ and $p = 10^4$, communicating 8×10^{10} bytes of data over gigabit Ethernet would take 10^2 seconds, less than 2 minutes. This is desirable for such a huge dataset.

Further optimization may be achieved via grouping inputs, which is not discussed here.

To conclude, per Mapper cost in the two stages sum up to $O(2ntp^2/M - (2/3)tp^3/M)$, again mostly contributed by solving the linear systems; per Reducer cost, incurred in the second stage, is $O(pt \log t/R)$. In addition, the communication cost from inputs to mappers and outputs to the master computer totals $16nt + 8nt + 8np + 8pt$ bytes and is almost always minimal in comparison to the shuffling cost. The shuffling cost, incurred in the second-stage MapReduce, is about 2 minutes if a dataset has 1 million time measurements (or pixels) and 10,000 covariates.

To compare the efficiency of the two implementations, we notice that the processing of the conventional method requires $O(2ntp^2 - (2/3)tp^3 + pt \log t)$ operations, while that of the MapReduce implementation requires $O(2ntp^2/M - (2/3)tp^3/M +$

$pt \log t/R$) operations, assuming $n > p$. A similar analysis can be performed when $n < p$ and the bottleneck of the processing cost probably still lies in model fitting and matrix decomposition. For example, implementing LASSO (when either $n \leq p$ or $n > p$) will require $O(p^3 + np^2)$ operations for each column of $\tilde{\mathbf{Y}}$ as a response and thus $O(tp^3 + ntp^2)$ operations for all columns. In this case, the processing cost will be $O(tp^3 + ntp^2 + pt \log t)$ and $O(tp^3/M + ntp^2/M + pt \log t/R)$ respectively if the conventional and MapReduce framework are employed. This implies when p and t are huge, the proposed new algorithm can be computationally superior to the conventional one, especially when the communication cost in MapReduce, as analyzed above, is not too big for the large dataset illustrated. However, when t is small, the communication cost of MapReduce may well dominate the computation cost, leading to inferior performance. Please see the simulation results in Section 2.6 for further discussion.

2.6 Illustrative simulation

In this section, we use synthetic datasets to illustrate the empirical performance of the conventional and MapReduce paradigm for implementing wavelet regression and shrinkage via extensive numerical experiments. To ensure reliable comparison, we verify without showing here that the results from MapReduce and the conventional algorithm are consistent with each other. With this guarantee, we verify our claim in Section 2.5.2 that the new algorithm achieves computational gain when the response dimension t is large.

The synthetic datasets are simulated with $\mathbf{X} \sim N(0, \Sigma)$ and $\epsilon \sim N(0, \sigma^2)$. The support of β is $S = \{1, 2, 3, 4, 5\}$. We generate row vector \mathbf{x}_i from a standard multivariate normal distribution with independent components. The coefficients are specified as $\beta_{ij} = (-1)^{Ber(0.5)} \left(|N(0, 1)| + 5\sqrt{\log p/n} \right) \mathbb{I}(i \in S)$, where β_{ij}

is the coefficient associated with \mathbf{x}_i and \mathbf{y}_j . The variance σ^2 is chosen such that $\hat{R}^2 = \text{var}(\mathbf{X}\beta_{(j)})/\text{var}(\mathbf{y}_{(j)}) = 0.9$, for $j = 1, 2, \dots, t$. We consider the model structure as follows:

$$\mathbf{y}_{(j)} = \mathbf{X}\beta_{(j)} + \epsilon, \text{ for } j = 1, 2, \dots, t.$$

2.6.1 Comparison: computational performance

As discussed in Section 2.5.2, the proposed MapReduced-based algorithm scales better in the dimension of response matrix t in comparison to the conventional one. We verify this property using the model specified above. The number of features p is 20, the sample size n is 1000 and the number of true signals $s = 5$. For each model, we simulate 10 synthetic datasets and record the trimmed average computational time using the middle 50%.

Table 2.3 shows that when the dimension of response matrix (or number of time measurements) $t < 512 = 2^8$, the MapReduce algorithm has a higher computation cost that hinders performance in small data sets. However, the MapReduce framework scales much better than the conventional algorithm, eventually surpassing it.

Table 2.3: Time in seconds for MapReduce and conventional algorithm to run with T time measurements. Here $n = 1000$ and $p = 20$ for all trials. Reported results are trimmed mean using five out of a total of ten replications.

T	MapReduce	Conventional
8	22.4	1.9
16	28.5	2.8
32	33.9	4.2
64	39.2	6.9
128	48.6	12.0
256	62.0	22.3
512	84.4	48.7
1024	119.4	89.5
2048	136.5	167.6
4096	157.8	324.6

2.7 Concluding remarks

In this chapter, we propose a parallel framework MapReduce for distributed implementation of wavelet regression and shrinkage. The new algorithm consists of two stages of MapReduce, with DWT performed in the first stage and regression and shrinkage in the second. Assuming $n > p$ and using multivariate linear regression and *SureShrinkage*, the processing cost is shown to be $O(2ntp^2 - (2/3)tp^3 + pt \log t)$ for conventional method and $O(2ntp^2/M - (2/3)tp^3/M + pt \log t/R)$. Similar results can be derived assuming $n \leq p$ and LASSO is performed in the regression step. This implies when the number of covariates p and the dimension of response t are huge, the proposed new algorithm can be computationally superior to the conventional one, especially when the communication cost in MapReduce is not too big for the large dataset given as an example. However, when t is small, the communication cost of MapReduce may well dominate the computation cost, leading to inferior performance.

The analysis in this chapter primarily deals with the case where the dimension of response t is a power of two, which is rare in real applications. Fortunately, we can resort to other transforms well defined for any sample size, such as maximal overlap DWT and similarly analyze the computational complexity. We should also extend the current analysis of 1D input vectors to accommodate 2D input vectors.

Analysis of Multiple Sclerosis Clinical Data

3.1 Introduction

Multiple Sclerosis (MS) is a chronic, “unpredictable and often disabling disease of the central nervous system (CNS) that disrupts the flow of information within the brain, and between the brain and body” (Society (2017)). Due to damage in the CNS, MS patients develop many symptoms, including blurred vision, loss of balance, poor coordination, numbness, problems with memory and concentration, etc (Society (2017)). These symptoms may remit, persist or aggravate over time. Unfortunately, The cause of MS, despite the current suspicion of genetic risk and environmental factors, remains a puzzle. The diagnosis is also very challenging, because in the early stages symptoms can be transient and common to many neurological diseases. The current diagnosis criteria incorporate medical history, magnetic resonance imaging (MRI), evoked potentials (EP) and spinal fluid analysis. As scientists are still searching for a cure, patients are given disease-modifying treatments (DMTs) to delay disability progression.

Our dataset is a prospective study of MS patients named *EPIC* (expression/ge-

nomics, proteomic, imaging, and clinical) provided by Biogen. The study comprises 579 actively managed MS patients enrolled at the Multiple Sclerosis Center at University of California, San Francisco (UCSF) between July 2004 and September 2005 (Cree et al. (2016)). The patients enrolled were evaluated annually based on clinical and radiologic test results since their baseline visits (i.e. between July 2004 and September 2005) for up to 10 years. We are provided with the first five-year data of EPIC. In addition, we also have access to the longitudinal gene expression profiles of whole-blood RNA from a cohort of 195 MS patients and 66 health controls (Nickles et al. (2013)).

MS is a heterogenous disease with unpredictable trajectory for many patients. Some present clinical symptoms very early on but progress slowly, while others do not have symptoms till later in life but thereafter progress rapidly. We are therefore interested in a tool that helps stage the disease in terms of current severity and predict the trajectory of any patient based on current marker profiles. Both *severity* and *progression* are loosely defined in clinical settings, but we can make proxies from available data. *Severity* may be defined as the ongoing inflammatory activity as measured by current gadolinium (GD) enhancing lesions and new GD/T2 lesion accumulation over a short period. *Progression* can be defined as changes in disability status, which is indicated by changes in expanded disability status scale (EDSS), timed 25-foot walk (T25W), 9-hole PEG test (9-HPT), Paced Auditory Serial Test (PASAT) scores or other test scores. The ability to predict progression can serve as a powerful guide for a proper early treatment to delay the build up of irreversible damages and thus improve long-term health of patients.

We have analyzed the five-year clinical data EPIC through an explanatory data analysis (EDA) and predictive modeling of disease courses and EDSS respectively. Our current ongoing effort involves analysis on summary statistics from MRI scans and longitudinal expression data is still under way. Our goal is to build a statistical

Table 3.1: Age distribution of 579 MS patients at enrollment

Age group	10 - 19	20 - 29	30 - 39	40 - 49	50 - 59	60 - 69
Counts	1	47	172	197	137	25

model to effectively predict disability progression using covariates from a subject’s demographic information, genome and clinical assessments made in the baseline visit.

In this chapter, we provide an introduction to the EPIC dataset and some currently available results. The rest of the chapter is organized as follows. Section 3.2 provides a glimpse of the variables in EPIC. In Section 3.3, we explore the relationships of variables and propose necessary pre-processing procedures via an EDA. Section 3.4 discusses model fitting. In particular, we first demonstrate the logistic regression model for predicting disease courses. We also demonstrate and compare five models used to predict EDSS scores. In Section 3.5 we discuss the limitations of the current models and future research directions.

3.2 Dataset

The dataset contains information of 579 MS patients who were followed up annually for up to 5 years after the baseline visit. In particular, it records their demographic characteristics, DMTs and annual clinical, MRI, SIENA and SIENAX assessments. The important variables in each category are described in the following sections.

3.2.1 Demographic Characteristics

The dataset consists of **gender**, date of birth(DOB) and **smoking history** as demographic variables. There are 401 females and 178 males, which reflects the gender difference that women are about twice as likely as men to develop MS in the population. The ages at enrollment range from 18 to 66, with the pattern given in Table 3.1. Smoking history takes values of *never*, *former* and *now*. It is surprising that 57% of the subjects had never smoked before their enrollment, in light of the fact

that the proportion of never-smokers who experienced early onset is similar to that for subjects who had ever smoked.

3.2.2 Disease-Modifying Treatments

There are 23 different kinds of prescribed drugs. To reduce the analysis complexity, we categorize the therapies into either “platform therapy” or “high-potency therapy”, based on the instructions from Cree et al. (2016) and Biogen. Briefly speaking, only natalizumab, rituximab, mitoxantrone, and cyclo-phosphamid are considered high-potency therapy. Only 15.7% of the patients were on high-potency therapy for at least one year. Among the 8 patients who were on high-potency therapy for at least four years, half experienced EDSS scores no greater than 4.5, a level of disability at which one is “able to work a full day” or “otherwise requires minimal assistance”.

3.2.3 Clinical variables

The major clinical variables include patient ID, family history of MS, age of onset, clinical visit date, disease course, and annual clinical assessments of EDSS, MSSS, walk score, PASAT score, avg peg score, DMT and No of attacks since last visit. A brief introduction to some clinical variables is given below.

Disease course

The dataset comprises five disease courses, clinically isolated syndrome (CIS), primary progressive MS (PPMS), progressive-relapsing MS (PRMS), (relapsing-remitting MS) RRMS and (secondary progressive MS) SPMS. Due to the update of International Advisory Committee on Clinical Trials of MS in 2013 (Lublin et al. (2014)), patients previously diagnosed of PRMS are now considered primary progressive. In addition to a nice graphic explanation (Figure 3.4), I provide an introduction to the disease courses based on the revised categories.

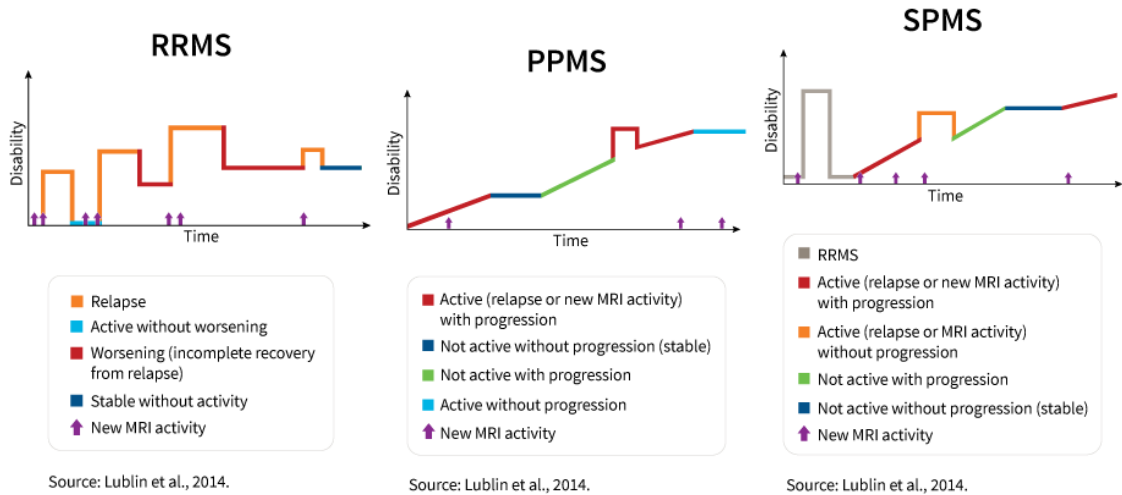


FIGURE 3.1: Development pattern of disease activities (Source: Lublin et al. (2014))

1. CIS (Clinically isolated syndrome)

CIS constitutes “a first episode of neurologic symptoms that lasts at least 24 hours and is caused by inflammation or demyelination in the CNS” (Society (2017)). Depending on MRI findings, e.g. brain lesions, individuals experiencing CIS vary in their risk of developing MS. A prompt and accurate diagnosis is crucial at this moment because early treatment of MS may alleviate future disability.

2. RRMS (Relapsing-remitting MS)

RRMS, the most common disease course, is the initial type that approximately 85% of MS patients are diagnosed with in the population. It is characterized by “clearly defined attacks of new or increasing neurologic symptoms” (Society (2017)). These attacks, or more formally relapses, are followed by periods of temporary recovery or remissions.

3. PPMS (Primary progressive MS)

In contrast to RRMS, PPMS does not have early relapses or remissions and is

Table 3.2: Distribution of disease courses at enrollment

Disease course	CIS	RRMS	PPMS	SPMS
Percentage	16.1%	71.3%	3.3%	8.8%

characterized by deteriorating neurologic function from the onset of symptoms (Society (2017)).

4. SPMS (Secondary progressive MS)

Following an initial pattern of relapses and remission, SPMS is a secondary progressive course characterized by a progressive worsening of neurologic function (accumulation of disability) over time. Most people who initially suffer from relapses and remission, or in other words RRMS patients, will eventually experience a transition to SPMS (Society (2017)).

From Table 3.4 we notice RRMS patients make up 71.3% of the total sample, a proportion less than that in the population.

Disability progression variables

Based on the introduction above, we know disability is one important symptom throughout the disease course. *Disability progression* was defined by clinically significant worsening in the four measurements: Expanded Disability Status Scale (EDSS), the timed 25-foot walk (T25W), the 9-hole peg test (9HPT), and the paced serial auditory addition test (PASAT-3) (Kutzke 1983; Nickles et al. 2013). These measurements are coded as variables `EDSS`, `walk score`, `avg peg score` and `PASAT score` in the dataset (Kutzke 1983; Nickles et al. 2013).

EDSS

EDSS serves as an imprecise but important indicator of disability progression, and as such we are interested in modeling changes in EDSS as a guide for treatment. More formally, the EDSS quantifies disability in eight Functional Systems (FS) and is as-

signed by a neurologist after observing patients performing various tasks (Kurtzke (1983)). It ranges from 0 to 10 in 0.5 unit increments representing higher level of disability and does not take value 0.5 (Kurtzke (1983)). It is not linear, meaning a one-unit increase on the larger side indicates a more significant disability progression (Kurtzke (1983)). The dataset contains EDSS scores ranging from 0 to 8.5 with the distributions roughly constant over the five years.

MSFC: T25W, 9HPT and PASAT-3

MSFC, the abbreviation for MS Functional Composite, was developed in 1999 (Cutter et al. (1999)) to address the limitations of traditional clinical scales (e.g. failure to measure cognitive functions) (Whitaker et al. (1995)) as a multidimensional measure to reflect the varied clinical expression of MS (J.S.Fischer et al. (1999)). It is a composite clinical measure of three tests: T25W, 9HPT and PASAT-3 that indicate leg function/ambulation, arm/hand function and cognitive function respectively (J.S.Fischer et al. (1999)). The formal constructions of T25W, 9HPT and PASAT-3 are not discussed here. In a nutshell, for T25W and 9HPT, a higher score is associated with inferior extremity functions and thereby worse MS disease status, whereas for PASAT-3, a higher score represents superior cognitive functions and thus possibly better MS conditions.

Family history of MS

To our surprise, 81% of the 579 patients do not have family history of MS, although one with family history is believed to be at higher risk of developing the disease (Clinic (2015)). Indeed, our model selection results show family history of MS is not selected by any predictive models for either EDSS or disease courses, indicating the disease progression of an MS patient may not depend on the family history when one is diagnosed with MS. However, family history of MS may still remain an important indicator in the population for the potential of developing MS.

Table 3.3: Distribution of age of onset of the 579 MS patients

Age group	10 - 19	20 - 29	30 - 39	40 - 49	50 - 59	60 - 69
Percentage	5.9%	29.5%	38.7%	20.0%	5.7%	0.1%

Age of onset

From Table 3.3, we observe 68.2% of the enrolled patients witnessed their first symptoms between the ages of 20 and 40, consistent with the general findings. Only 1 patient has age of onset later than 60.

3.2.4 MRI assessments

MRI assessments include CE lesion, new T2 lesions, T2 volumes (mm^3) and T1 volumes (mm^3).

3.2.5 SIENA assessments

SIENA (Structural Image Evaluation, using Normalization, of Atrophy) is a two-time-point method for finding percentage brain volume change (PBVC) between two input images of the same subject taken at different time points (Jenkinson (2016)).

3.2.6 SIENAX assessments

Extended from SIENA, SIENAX is an accurate single-time-point method for estimating atrophy state as opposed to atrophy rate (Smith (2005)).

3.3 EDA and Data Preprocessing

3.3.1 Data Preprocessing

1. One subject has avg peg score over 500 during his/her clinical visits in the third year. Since avg peg score can only take values up to 300, it is confirmed to be a bad processing and thus removed.

Table 3.4: Grouping of disease course

Before grouping	After grouping
RRMS, CIS	RMS
SPMS, PPMS, PRMS	PMS

Table 3.5: Variable `year` created from `Visit`

<code>Visit</code>	<code>year</code>
Baseline	0
F/U Yr i , $1 \leq i \leq 6$	i

2. Variable `grouped disease course` is created by grouping disease courses based on the Cree et al. (2016)’s practice. They argued due to the evolution of diagnostic criteria, many patients that were classified as CIS during the course of the study would now be designated as MS (Cree et al. (2016)). Similarly, PRMS is merged with PPMS due to aforementioned updates in diagnosis in 2013. In addition, subjects with PPMS and SPMS are grouped given their “potential common histopathologic and genetic basis” (Cree et al. (2016)). The grouping is summarized in Table 3.4.
3. Variable `EDSS base`, which is the EDSS score measured at the baseline visit, is created and used as an explanatory variable for predicting current EDSS in all relevant models.
4. Variable `year`, which is the clinical visit year index, is extracted from variable `Visit` (Table 3.5) and used as an explanatory variable in predictive models.
5. MSFC measurements `avg peg score` and `walk score` are taken logarithm in the predictive models as explanatory variables for predicting EDSS, due to their strong right skewness indicated by Figure 3.3 in the EDA section.

Relationship between demographic variables

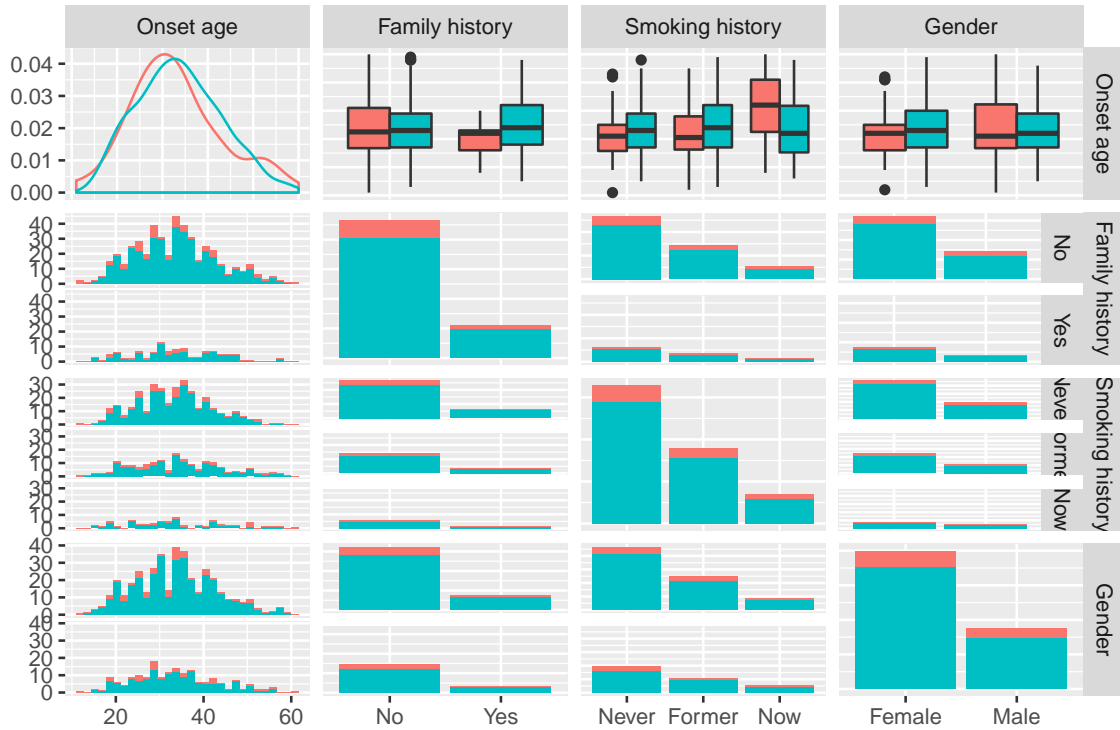


FIGURE 3.2: Relationships between demographic and a few clinical variables. Green represents the disease course PMS, and red represents the disease course PMS

3.3.2 EDA

The distributions of gender, age of onset, disease course and family history of MS are discussed in the previous section Datasets. The remaining EDA consists of scatterplot matrices of demographic, clinical and MRI variables. From Figure 3.2, we observe that

1. The distributions of age of onset appear independent of the grouped disease course.
2. Among subjects that have family history of MS, those who were diagnosed with a more serious disease type, i.e. PMS, at baseline visit tend to experience early onset. However, among subjects that do not have family history of MS, such pattern is not observed.

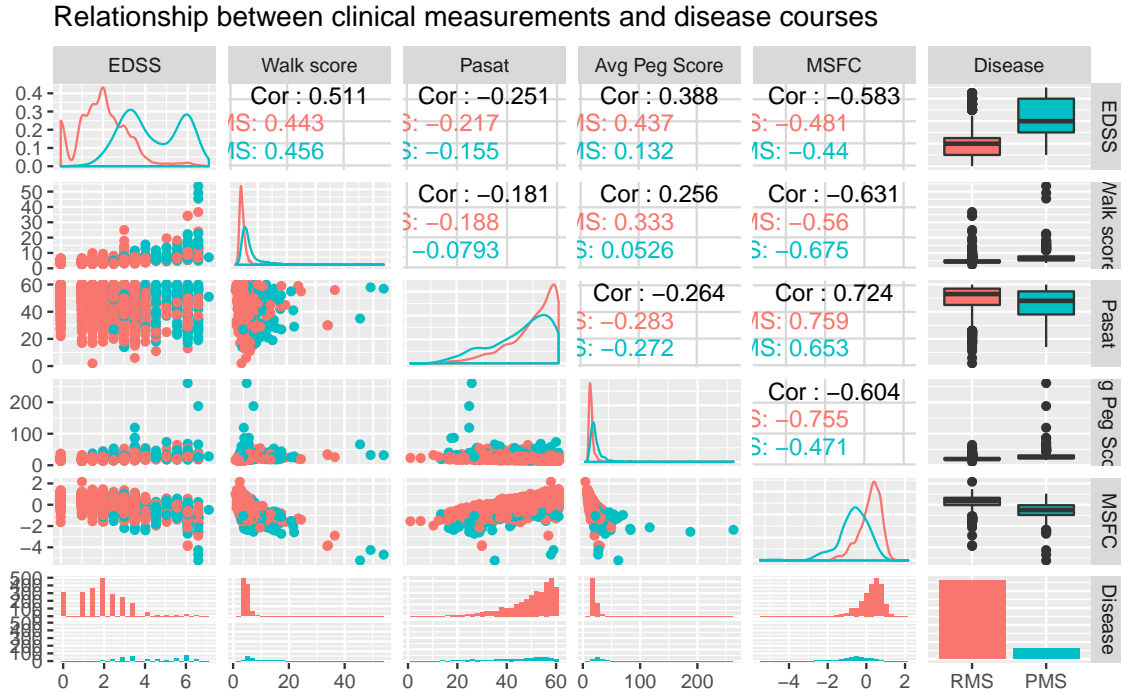


FIGURE 3.3: Relationships between clinical variables. Green represents the disease course PMS, and red represents RMS.

3. Current smokers undergoing PMS, a more serious disease course, tend to witness late onset.

Patterns observed from Figure 3.3 are summarized below:

1. Both distributions of EDSS from RMS and PMS patients are bimodal, and there is not much overlap between them.
2. The distribution of walk score and avg peg score are strongly right skewed, while that of PASAT score shows left skewness. This is reasonable since PASAT score is the number of questions patients answer correctly in a 60-question cognitive function test, implying its maximum value is capped at 60.
3. In addition to the high correlation between MSFC and its three composite measures, there is indeed moderate negative correlation between MSFC and EDSS.

Together with the evidence from the box plot, we conclude a lower MSFC reflects a worse disease status.

3.4 Model fitting

3.4.1 Prediction of disease courses: logistic regression

To assess both in-sample and out-of-sample performance, we randomly partition the dataset into a training (75%) and testing (25%) set. We fit a logistic regression to predict the grouped disease courses RMS vs PMS with all other variables as explanatory variables after appropriate transformations, following the preprocessing discussed in Section 3.3.1. In particular, let Y be a binary response variable with

$$Y_i = 1 \text{ if subject}_i \text{ is diagnosed with PMS,}$$

$$Y_i = 0 \text{ if subject}_i \text{ is diagnosed with RMS.}$$

Define $\pi_i = \mathbb{P}(Y_i = 1|\mathbf{X})$, where \mathbf{X} is the design matrix. Models are selected using backward stepwise selection based on AIC criterion. The selected model is:

$$\begin{aligned} \text{logit}(\hat{\pi}_i) = & -15.35 + 1.52\text{EDSS}_i - 0.48\text{MSSS}_i + 0.84\log(\text{walk score}_i) \\ & + 2.46\log(\text{avg peg score}_i) + 0.53\mathbb{I}(\text{gender}_i = \text{Male}) \\ & - 0.08\mathbb{I}(\text{smoking history}_i = \text{Former}) + 0.68\mathbb{I}(\text{smoking history} = \text{Now}) \\ & + 0.03\text{age of onset}_i - 0.48\text{MSFC}_i \end{aligned}$$

The selected model achieves out-of-sample accuracy of 91.3%.

3.4.2 Prediction of EDSS: five models

EDSS is a traditional disability measurement focusing mainly on the ability to walk. As introduced in Section 3.2, EDSS takes discrete values with 0.5-unit increments on a scale of 0 to 10 except for the value 0.5. Here we use it as a proxy for current disease status and fit five models to predict EDSS scores. The five models are multiple linear regression, proportional odds model, proportional hazards model,

linear mixed effects model and proportional odds mixed-effects model. In particular, EDSS is treated as continuous in multiple linear regression and linear mixed effects model, while in proportional odds, proportional hazards and proportional mixed-effects model, EDSS is treated as a categorical variable. The random effects in linear mixed effects model and proportional odds mixed-effects model are patient IDs or subjects.

We use all other variables as explanatory variables except for MSSS, because MSSS is actually derived from EDSS to measure the MS severity. Model selections are performed via either backward stepwise selection (AIC criterion) or likelihood ratio tests. After a single model is selected for each model type, we compare the selected five models in terms of predictive accuracies. I use both *exact accuracy* and *approximate accuracy* as the criteria. *Exact accuracy* means predicting the exact EDSS, while *approximate accuracy* means predicting the correct or an adjacent EDSS score. For example, if the observed EDSS is 2 and the prediction is either 1.5, 2 or 2.5, then it fulfills the requirement of *approximate accuracy*. Such concepts are tricky for linear models, since EDSS is treated as continuous and predicting the exact discrete values can rarely happen. In this case, I round the fitted value to the nearest 0.5. For example, if EDSS is predicted to be 3.84, then the rounding will yield a score of 4, which is considered accurate if the observed EDSS score is 4 and approximately accurate if it is 4.5 or 3.5.

The best model in terms of predictive performance is *linear mixed-effects model*. The selected model is

$$\begin{aligned} \widehat{\text{EDSS}}_{ij} = & -2.69 + 0.51\text{EDSS base}_{ij} + 1.16\log(\text{walk score}_{ij}) + 0.004\text{PASAT score}_{ij} \\ & + 0.69\log(\text{avg peg score}_{ij}) - 0.03\text{No of attacks since last visit}_{ij} \\ & + 0.70\mathbb{I}(\text{grouped disease course}_{ij} = \text{PMS}) + 0.12\text{year}_{ij} \\ & - 0.11\mathbb{I}(\text{grouped disease course}_{ij} = \text{PMS}) \times \text{year}_{ij} + a_i + \epsilon_{ij}, \end{aligned}$$

Table 3.6: In-sample accuracies of EDSS prediction

Model	Exact accuracy	Approximate accuracy
Multiple linear regression	0.26	0.70
Proportional odds model	0.33	0.75
Proportional hazards model	0.30	0.70
Linear mixed-effects model	0.32	0.81
Proportional odds mixed-effects model	0.33	0.75

where $a_i \sim N(0, 0.43^2)$ and $\epsilon_{ij} \sim N(0, 0.75^2)$. The in-sample predictive accuracies of all the selected models are summarized in Table 3.6. In addition to model comparison, we observe a few interesting patterns:

1. Variable `smoking history` is never selected based on either backward stepwise selection or likelihood ratio tests. Indeed, rank deficiency occurs when we use `smoking history` in proportional odds/hazards models. This indicates that smoking history can possibly be perfectly predicted by other explanatory variables.
2. The interaction effect between grouped disease courses and year is statistically significant in all the predictive models for EDSS scores.
3. The diagnostic plots of both multiple linear regression and linear mixed-effects model suggest linearity assumption is satisfied. The normality assumption is also roughly satisfied, except for a slightly heavy right tail (See Figure 3.4).

3.5 Concluding remarks

In this chapter, we provide an analysis of the EPIC dataset. We first introduce the important variables and provide their distributions in this dataset to facilitate model building and results interpretation. We also shed light on the relationship of variables via EDA to detect patterns and outliers. In the last section, we fit sta-

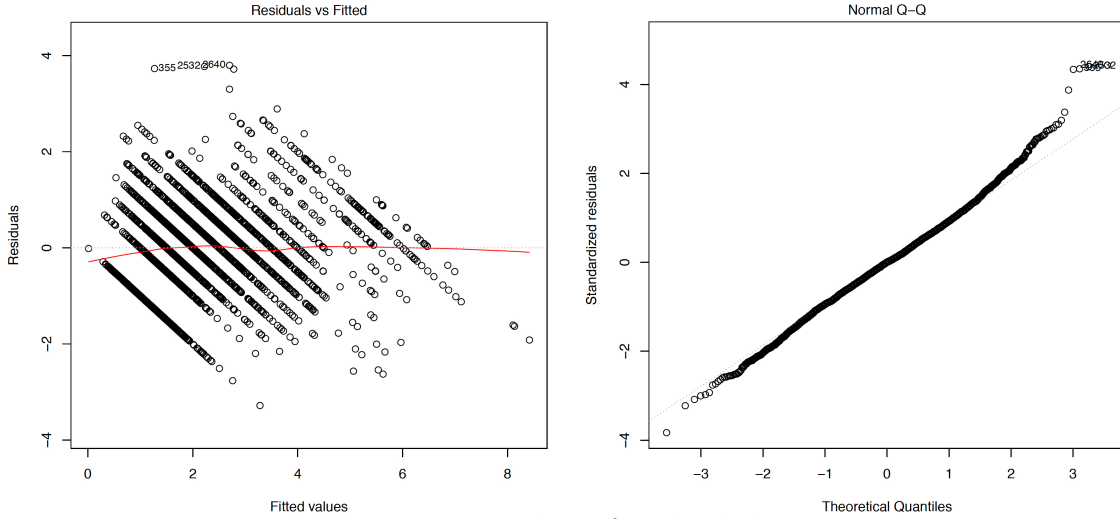


FIGURE 3.4: Diagnostic plots of multiple linear regression

tistical models to predict disease courses and EDSS scores respectively and perform model comparison. We have several interesting observations, e.g. the (linear) model assumptions are roughly satisfied, as evidenced by the residual plot and Q-Q plot.

However, the current analysis primarily focuses on predicting/estimating EDSS scores based on current marker profiles, which is less of a guide for early diagnosis and treatments. Therefore, our next step mainly involves two aspects. First, we hope to predict two-year EDSS progression using baseline marker profiles. Our tentative models, a linear mixed-effects model and a proportional hazards model with random effects, show poor predictive performance with an out-of-sample accuracy at about 30%. Second, we would like to understand the longitudinal gene expression data in combination with the EPIC dataset. In particular, we are curious to identify some gene expression signature that clearly differentiate treated from untreated patients and untreated patients from healthy individuals, and furthermore to infer the potential common genetic basis for each disease course.

Bibliography

- Afrati, F. N., Sarma, A. D., Salihoglu, S., and Ullman, J. D. (2013), “Upper and Lower Bounds on the Cost of a Map-Reduce Computation,” *Proceedings of the VLDB Endowment*, 6.
- Alsberg, B. K., Woodward, A. M., Winson, M. K., Rowland, J. J., and Kell, D. B. (1998), “Variable selection in wavelet regression models,” *Analytica Chimica Acta*, 368, 29–44.
- Clinic, M. (2015), “Symptoms and causes,” <http://www.mayoclinic.org/diseases-conditions/multiple-sclerosis/symptoms-causes/dxc-20131884>.
- Cree, B. A., Gourraud, P.-A., Oksenberg, J. R., Bevan, C., Crabtree-Hartman, E., and Gelfand, J. M. (2016), “Long-Term Evolution of Multiple Sclerosis Disability in the Treatment Era,” *Annals of Neurology*, 80, 499–510.
- Cutter, G., Baier, M., M.S.Rudick, R.A.Cookfair, D.L.Fischer, and J.S.Petkau (1999), “Development of a Multiple Sclerosis Functional Composite as a clinical trial outcome measure,” *Brain*, 122, 101–112.
- Donoho, D. L. and Johnstone, I. M. (1995), “Adapting to Unknown Smoothness via Adapting to Unknown Smoothness via Wavelet Shrinkage,” *Journal of the American Statistical Association*, 90, 1200–1224.
- Gao, R. X. and Yan, R. (2011), *Wavelets: Theory and Applications for Manufacturing*, Springer US.
- Hubbard, B. B. (1948), *The world according to wavelets*, A K Peters.
- Jenkinson, M. (2016), “SIENA,” <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/SIENA>.
- J.S.Fischer, Rudick, R., Cutter, G., and S.C.Reingold (1999), “The multiple sclerosis functional composite measure (MSFC): An integrated approach to MS clinical outcome assessment,” *Multiple Sclerosis*, 5, 244–250.
- Kurtzke, J. F. (1983), “Rating neurologic impairment in multiple sclerosis: An expanded disability status scale (EDSS),” *NEUROLOGY*, 33, 1444–52.

- Leskovec, J., Rajaraman, A., and Ullman, J. D. (2014), *Mining of Massive Datasets*, Cambridge University Press.
- Lublin, F. D., Reingold, S. C., Cohen, J. A., and Cutter, G. R. (2014), “Defining the clinical course of multiple sclerosis,” *American Academy of Neurology*, 83.
- Mallat, S. (1989), “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE Transactions on patterns analysis and machine intelligence*, 11.
- Nickles, D., Chen, H. P., Li, M. M., Khankhanian, P., Madireddy, L., Caillier, S. J., Santaniello, A., Cree, B. A., Pelletier, D., Hauser, S. L., Oksenberg, J. R., and Baranzini, S. E. (2013), “Blood RNA profiling in a large cohort of multiple sclerosis patients and healthy controls,” *Human Molecular Genetics*, 22, 4194–4205.
- Smith, S. (2005), “SIENA - Brain Change Analysis,” <https://www.fmrib.ox.ac.uk/datasets/techrep/tr04ss2/tr04ss2/node14.html>.
- Society, N. M. S. (2017), “National Multiple Sclerosis Society,” <http://www.nationalmssociety.org/>.
- Vidakovic, B. (1955), *Statistical modelling by wavelets*, John Wiley & Sons, Inc.
- Whitaker, J., H.F.McFarland, P.Rudge, and S.C.Reingold (1995), “Outcomes assessment in multiple sclerosis clinical trials: a critical analysis.” *Multiple Sclerosis*, 1, 37–47.

Biography

Hanyu Song was born on Dec 11, 1991 in Shenzhen, Guangdong, China. She received a B.S. in Actuarial Science from University of Hong Kong in June 2014, an M.S. in Statistical Science from Duke University in 2017.

She was part of the team that received the first place for the “Best Recommendation” category from 2016 ASA DataFest.