This report contains a summary of the Industrial Mathematical and Statistical Modeling Workshop for Graduate Students, held in the Department of Mathematics at North Carolina State University (NCSU) in Raleigh, North Carolina, 15 – 25 July 2018.

This was the twenty-fourth such workshop held at NCSU. It brought together 39 graduate students from mainly Mathematics and Statistics Departments at over 30 different universities.

The goal of the IMSM workshop is to expose mathematics and statistics students from around the country to real-world problems from industry and government laboratories; interdisciplinary research involving mathematical, statistical and modeling components; as well as experience in a team approach to problem solving.

Following the past couple of years’ success, the IMSM workshop again included a Bootcamp on Building Software in Teams on the first day (16 July 2018), where the students learned about shell commands, version control and Git.

On the morning of the second day (17 July 2018), industrial and government scientists presented six research problems. The presenters, together
with specially selected faculty mentors, then guided teams of 6–7 students and helped them to discover a solution. In contrast to neat, well-posed academic exercises that are typically found in coursework or textbooks, the IMSM problems are challenging real world problems that require the varied expertise and fresh insights of the group for their formulation, solution and interpretation. Each group spent the remaining days of the workshop investigating their project, and reported their findings in 20-minute public presentations on the afternoon of the final day (25 July 2018).

The IMSM workshops have been highly successful for the students as well as the presenters and faculty mentors. Often projects lead to new research results and publications. The projects can also serve as a catalyst for future collaborations between project presenter and faculty mentor. More information can be found at http://www.samsi.info/IMSM-19

Sponsors

Statistical and Applied Mathematical Sciences Institute (SAMSI)
Center for Research in Scientific Computation (CRSC)
Department of Mathematics, North Carolina State University

Instructor for Bootcamp on Building Software in Teams

John Pearson (Duke University)

Problem presenters

Jordan Massad (Sandia National Laboratories)
Agustin Calatroni and Petra LeBeau (Rho Inc.)
Ty Hesser and Matthew Farthing (US Army Corps of Engineers)
Elizabeth Mannshardt, Barron Henderson and Brett Gantt (EPA)
Grant Weller, Victoria Mansfield and Yinglong Guo (United Health Group)
John Peach (MIT Lincoln Laboratory)

Faculty mentors

Alen Alexanderian (North Carolina State University)
Emily Kang (University of Cincinnati)
Daniel Luckett (UNC-Chapel Hill)
Paul Miles (North Carolina State University)
Brian Reich (North Carolina State University)
Arvind Saibaba (North Carolina State University)
Ralph Smith (North Carolina State University)

Projects

Computational Techniques to Enhance Laser-based Characterization of Thin Film Thermomechanical Properties
Problem presenter: Jordan Massad
Faculty mentors: Ralph Smith and Paul Miles
Students: Esther Amfo, Jiahui Chen, Rasika Rajapakshage, Klajdi Sinani, John Wakefield, Meng Zhang

Visualizing and Interpreting Machine Learning Models for Liver Disease Detection
Problem Presenters: Agustin Calatroni and Petra LeBeau
Faculty mentor: Emily Kang
Students: Adams Kusi Appiah, Sharang Chaudhry, Chi Chen, Simona Nallon, Upeksha Perera, Manisha Singh, Ruyu Tan

Estimation of Coastal Hydrodynamics with Machine Learning
Problem Presenter: Ty Hesser and Matthew Farthing
Faculty mentor: Arvind Saibaba
Students: Taylor Baumgard, Cassidy Krause, Evan North, Ramchandra Rimal, Jacob Roth, Arsenios Tsokas

Splicing of Multi-Scale Downscaler Air Quality Surfaces
Problem Presenters: Elizabeth Mannshardt, Barron Henderson and Brett Gantt
Faculty mentor: Brian Reich
Students: Elizabeth Herman, Jeonghwa Lee, Kartik Lovekar, Fatemeh Norouzi, Dorcas Ofori-Boateng, Benazir Rowe, Jianhui Sun

Semi-analytical BRDF-Based Quantification of Light Reflection
Problem presenter: John Peach
Faculty mentors Alen Alexanderian
Students: Michael Byrne, Fatoumata Sanogo, Pai Song, Kevin Tsai, Hang Yang, Li Zhu
Identifying Precision Treatment for Rheumatoid Arthritis with Reinforcement Learning

**Problem Presenters:** Grant Weller, Victoria Mansfield and Yinglong Guo

**Faculty mentor:** Daniel Luckett

**Students:** Chixiang Chen, Ashley Gannon, Duwani Katumullage, Miaoqi Li, Mengfei Liu, Rebecca North, Jialu Wang
Computational Techniques to Enhance Laser-based Characterization of Thin Film Thermomechanical Properties

Esther Amfo1, Jiahui Chen2, Rasika Rajapakshage3, Klajdi Sinani4, John Wakefield5, Meng Zhang6

Problem Presenter: Jordan Massad7, Faculty Mentors: Paul R. Miles, Ralph C. Smith8

Abstract

Thin films are widely used in various devices including microelectronics and microelectromechanical systems. For this reason, it is important to accurately identify their properties. A Laser Deflectometer (LD) measures the warpage of thin films as temperature varies. To control the temperature, a thermal enclosure is needed. The top quartz plate of the thermal enclosure affects the path of the beam; hence, it distorts the warpage measurement. In this report, we provide two physical models: one relies on geometrical analysis, whereas the other is developed using operator-based modeling to correct the wafer warpage measurement. Also, an efficient statistical model based on experimental data predicts the true warpage given the LD measurement. We then analyze the statistical properties and validation using new datasets. In conclusion, we summarize our analysis and provide recommendations.

1 Problem Description

Microfabricated thin films are key constituents of coatings, microelectronics, and microelectromechanical systems (MEMS). Thermomechanical behavior of thin films is critical to the functionality and reliability in these applications. Thin films are merely nanometers to micrometers thick and their thermomechanical material properties differ from those observed in bulk dimensions, which is sensitive to fabrication processes. Thin films are fabricated on thicker, functional substrates; removing the films from their substrates can destroy the films. This presents a challenge in characterizing the thin film thermomechanical properties, as isolating the thin films for testing may be impractical and adding fabrication steps to produce film samples free of the substrate may alter the film properties. Therefore, a method to identify thin film properties in situ is needed.

The integrated circuit (IC) industry has deployed a non-contact, laser-based profilometer that is tailored to measuring warpage of thin films on standard substrates (wafers) at varying temperatures. The profilometer determines warpage by measuring how a laser beam impinging on the film surface reflects onto a position-sensitive photosensor as it scans across the film. These measurements are readily realized at ambient temperatures; however, to measure warpage at controlled, non-ambient temperatures, the wafer must be enclosed to maintain thermal stability. The laser and photosensor lie outside the thermal enclosure, so the laser light must enter the enclosure before reaching the film and exit the enclosure after reflecting off the film during the temperature-controlled program.

The alteration of the beam path through a quartz window in the enclosure distorts film warpage data. Traditionally, the IC industry has disregarded distortion of warpage data by the enclosure, since it assumes that film stress, a primary quantity of interest derived from differential warpage data, is unaltered. Yet, highly desired thin film mechanical properties can be identified using the absolute warpage data, provided effects of the enclosure are accommodated. Modifying hardware or employing an alternate technology that could be adopted by the IC industry may be a long-term solution, but such changes are impractical. Instead, there is an immediate need to use existing profilometer functionality to ascertain non-distorted wafer warpage. This project aims to develop computational techniques that can effectively...
remove warpage distortion induced by the thermal enclosure, thereby enhancing the capability of characterizing thin film properties with existing hardware. Various approaches were implemented in order to construct a model that predicts the true warpage given an LD warpage measurement. We developed both, physical and statistical models, in order to solve this problem. First, we establish a physical model following two different approaches. Then, we construct a statistical model using measurement data obtained from the LD. Finally, we include conclusions and recommendations on potential actions that could improve the warpage predictions for thin films. Quantities and terms of interest in this report are defined and explained in the Glossary at the end of this report.

2 Mathematical Models

Various approaches can be taken to modeling this problem. In this report we outline two different methods, which are summarized as

1. Model Approach 1: Geometrical Analysis
2. Model Approach 2: Algebraic Operations

Before describing these methods in detail, we will first clarify several mathematical concepts that are common to both methods. First, we explain a key step in conversion of angle measurements $(x_i, \theta_i)$ into computed radii $R$.

2.1 Finding Radii

The following derivations reference information reported in Figure (1). Since we control the placement of the substrate with respect to the coordinate system, we place the wafer so that $G$ lies on the $y$-axis and $|GH|$ is the diameter ($D$) of the wafer, which is known. $G$ and $H$ denote the edges of the locator ring. Using this placement, the $x$-coordinate of the curvature center $C \equiv (a, b)$ is the same as the $x$-coordinate of center of the wafer i.e. $a = \frac{D}{2}$. Therefore, the center of the substrate is $C \equiv (\frac{D}{2}, R)$ where $R$ is the radius of the curvature of the substrate. The laser beam hits the wafer at point $F_i$. Let $L_i$ be a random position for the source of a laser beam. Then, the beam hits the substrate at $F_i \equiv (x_i, y_i)$. The angle of the laser beam with the vertical direction is given by $\angle L_iF_iE_i = \alpha$. The angle the reflected beam forms with the vertical direction is $\angle E_iX_iM = \alpha_i$. Additionally, denote $E_iX_iC = \theta_i$. Next, we express $x_i$ and $y_i$ in terms of $R$, $a,b$ and the given angles. From basic trigonometric identities we have,

$$a - x_i = R \sin \theta_i$$
and
\[ b - y_i = R \cos \theta_i \]
Since the radius of curvature \( R \) corresponds to the surface normal of the curvature, and \( \alpha + \theta_i \) is the angle of incidence, while \( \alpha_i - \theta_i \) is the angle of reflection, by the Law of Reflection, we have
\[ \alpha + \theta_i = \alpha_i - \theta_i. \]
Hence,
\[ \theta_i = \frac{\alpha_i - \alpha}{2}. \]
As a result,
\[ R \sin \frac{\alpha_i - \alpha}{2} = a - x_i \]  \hspace{1cm} (1)
Writing (1) in terms of \( \theta \) we get
\[ R \sin \theta = a - x_i \]  \hspace{1cm} (2)
Differentiating (2) with respect to theta we obtain
\[ \frac{d\theta}{dx_i} = -\frac{1}{R \cos \theta} \]
\[ \frac{1}{R} \]  \hspace{1cm} (4)
Consider
\[ 0 < \theta \ll 1 \Rightarrow \cos \theta \approx 1. \]
Because we are interested in making physical measurements, it is necessary to evaluate the error that this approximation introduces. This is explored through computational experiment in Section 2.3.1 and in terms of maximum possible angles (within the design constraints of the deflectometer) here.

We can establish a maximum error by using the design constraints. We know that the diameter of the wafer is given \( D = 150 \) mm and the deflectometer cannot measure radii larger than 7 m. Solving Equation (2) for \( \theta \) we get
\[ \theta = \arcsin(1 - \frac{x_i}{R}). \]  \hspace{1cm} (5)
Hence,
\[ \cos \theta = \sqrt{1 - \left(\frac{x_i}{R}\right)^2}. \]  \hspace{1cm} (6)
Thus,
\[ \cos \theta \geq \sqrt{1 - \left(\frac{D}{R}\right)^2} \approx 0.9998 \]  \hspace{1cm} (7)
It follows that
\[ 1 \leq \frac{1}{\cos \theta} \leq 1.000230 \]  \hspace{1cm} (8)
even in the worst case.

### 2.2 Model Approach 1: Geometrical Analysis

This approach utilizes geometrical and trigonometrical analysis to develop a quasi-analytical solution to the problem of interest. We pick up our geometric analysis that we started in the previous section. For our analysis we use additional identities which we obtain next. Solving for \( \alpha_i \) in Equation (1) we obtain
\[ \alpha_i = 2 \arcsin \left(\frac{a}{R} - \frac{x_i}{R}\right) + \alpha \]  \hspace{1cm} (9)
Let \( t_i \) be the \( x \)-coordinate of the beam source at any given time. Then, \( x_i \) is a linear function of \( t_i \) with slope equal to one. So we write Equation (9) as
\[ \alpha_i = 2 \arcsin \left(\frac{a}{R} - \frac{x_i(t_i)}{R}\right) + \alpha \]  \hspace{1cm} (10)
Differentiating Equation (10) with respect to \( t_i \) and solving for \( R \) we get
\[ R = -2 \left(\frac{dx_i}{dt_i} \cos \left(\frac{\alpha_i - \alpha}{2}\right)\right)^{-1} \]  \hspace{1cm} (11)
In the following subsection we describe how we use Equation (11) to construct the desired model.
2.2.1 Methodology

Our goal is to construct a relation between the actual radius as measured when there is no quartz ($R_N$) and the radius as measured by the LD with quartz ($R_Q$) using Equation (11). We assume the substrate radius as measured with no quartz ($R_N$) is known.

Denote

\[ G(\alpha_i, t_i) = \frac{d\alpha_i}{dt_i} \]  

Then, assuming $R_N$ is known, we rewrite Equations 10 and 11 as

\[ \alpha_i = 2 \sin^{-1} \left( \frac{a - x_i(t_i)}{R_N} \right) + \alpha \]  

\[ R_Q = -2 \left( G(\alpha_i, t_i) \cos \left( \frac{\alpha_i - \alpha}{2} \right) \right)^{-1} \]  

Combining Equations 13 and 14, we obtain a model that corrects the radius as follows

\[ R_Q = -2 \left( G(\alpha_i, t_i) \cdot \sqrt{1 - \left( \frac{a - x_i(t_i)}{R_N} \right)} \right)^{-1} \]  

Equation (15) is the crux of this approach. We have a model that takes $R_n$ as an input, and outputs $R_Q$. In order to implement Equation (15), we need $\frac{d\alpha_i}{dt_i}$. Using trigonometric identities and following Figure 2 we obtain a relation between $\alpha_i$ and $t_i$.

Note that

\[ |OB| = t_i + w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 \]  

where $t_i$ is the horizontal distance of the laser (beam source) from the $y$-axis, and $w_1, w_2, w_3, w_4, w_5, w_6$ and $w_7$ are distances referenced in Figure 2. Using the given dimensions for the LD, the specifications for the quartz, and Snell’s Law, we compute each term in Equation (16) separately.
The horizontal distance from the laser L to the point A where the beam first hits the quartz is given by

\[ w_1 = l_1 \tan \alpha. \]  \hspace{1cm} (17)

As the beam is entering the quartz, it gets refracted. Denote the angle of refraction \( \angle KAG = \beta \). By Snell’s Law we have

\[ \beta = \arcsin \left( \frac{n_A}{n_Q} \sin \alpha \right). \]  \hspace{1cm} (18)

For, simplicity we use \( n = \frac{n_A}{n_Q} \). So, the horizontal distance from the point A where the beam enters the quartz to the point G where it leaves the quartz is

\[ w_2 = l_2 \tan \beta \] \hspace{1cm} (19)

As the beam is leaving the quartz it gets refracted again, so, the horizontal distance from point G, where the beam leaves the quartz, to point \( F_i \) where it hits the wafer is

\[ w_3 = l_3 \tan \alpha \] \hspace{1cm} (20)

Since the surface of the wafer is curved, the surface normal is distinct from the vertical direction, thus, \( \alpha_i \neq \alpha \). As a result

\[ w_4 = l_3 \tan \alpha_i \] \hspace{1cm} (21)

After reflection, the beam hits the quartz again, and it gets refracted. Denote \( \angle Z_iQ_iS_i = \beta_i \). Applying Snell’s Law again, we have

\[ \beta_i = \arcsin \left( \frac{n_A}{n_Q} \sin \alpha_i \right). \]  \hspace{1cm} (22)

Hence,

\[ w_5 = l_3 \tan \beta \] \hspace{1cm} (23)

Note that the beam \( S_iM_i \), which illustrates the direction of the beam after it leaves the quartz, and the line \( CM_i \) which denotes the direction of the mirror intersect at the point \( M_i \). For simplicity, let

\[ c_i = w_1 + w_2 + w_3. \]

The coordinates \( S_i \) are \( x_{S_i} = t_i + c_i + w_4 + w_5 \) and \( y_{S_i} = l_2 + l_3 \), and the angle \( \angle Y_iS_iM_i = \beta_i \). Therefore,

\[ y_{S_i} = \cot \alpha_i x_{S_i} + b, \]

where \( b \) is the \( y \)-intercept for the line \( S_iM_i \). Thus,

\[ b = y_{S_i} - \cot \alpha_i x_{S_i}. \]

So the equation for line \( S_iM_i \) is

\[ y = \cot \alpha_i x + y_{S_i} - \cot \alpha_i x_{S_i}. \]

The point \( C \) is the center of the mirror. Its horizontal distance from the laser is \( h_M \), and its vertical distance from the floor of the machine is \( v_M \). Thus, the coordinates of point \( C \) are \( x_C = h_M + t_i \) and \( y_C = v_M \). The mirror is placed at an angle \( \theta \) with the vertical direction. Consequently,

\[ y_C = \cot \theta x_C + p, \]

where \( p \) is the \( y \)-intercept of the mirror direction. Hence

\[ p = y_C - \cot \theta x_C. \]

As a result, the equation for the line where the mirror is located is given by

\[ y = \cot \theta x + y_C - \cot \theta x_C. \]

We compute the coordinates of \( M_i \) by finding the intersection of the mirror with the beam \( S_iM_i \). We get

\[ x_{M_i} = \frac{y_C - \cot \theta x_C - y_{S_i} + \cot \alpha_i x_{S_i}}{\cot \alpha_i - \cot \theta}. \] \hspace{1cm} (24)

We obtain \( y_{M_i} \) by plugging \( x_{M_i} \) into the equation of the mirror line. Hence,

\[ y_{M_i} = \cot \theta x_{M_i} + y_C - \cot \theta x_C. \] \hspace{1cm} (25)
Then, we have,

\[ w_6 = (y_{M_i} - l_2 - l_3) \tan \alpha_i \] (26)

Finally, note \( w_7 = x_{H_i} - x_{M_i} \). Thus,

\[ w_7 = w + t_i - x_{M_i}. \] (27)

Recall that

\[ t_i + w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 = w + t_i. \]

Thus,

\[ t_i + w_1 + w_2 + w_3 + w_4 + w_5 + w_6 - x_{M_i} = 0. \]

Substituting and simplifying appropriately, we can write \( w_6 - x_{M_i} \) in terms of \( t_i \) and \( \alpha_i \) as follows:

\[
w_6 - x_{M_i} = \frac{[v_M - \cot \theta (h_M + t_i) - l_2 + \cot \alpha_i (t_i + c_1 + l_2 \tan (\sin^{-1} (n \sin \alpha_i))) (\cot \theta - 1)]}{\cot \alpha_i - \cot \theta} + v_M - \cot \theta (h_M + t_i). \] (28)

We also have

\[ t_i + w_1 + w_2 + w_3 + w_4 + w_5 = t_i + c_1 + l_2 \tan \alpha_i + l_2 \tan (\sin^{-1} (n \sin \alpha_i)) \] (29)

Note that these identities describe a relationship between \( \alpha_i \) and \( t_i \) where all the other quantities are known. So using Equations 28 and 29 we define a function \( F \) such that

\[ F(\alpha_i, t_i) = 0 \] (30)

For the sake of aesthetics, we write

\[ F(\alpha_i, t_i) = F_1(\alpha_i, t_i) + F_2(\alpha_i, t_i) = 0, \] (31)

where

\[ F_1(\alpha_i, t_i) = t_i + c_1 + l_2 \tan \alpha_i + l_2 \tan (\sin^{-1} (n \sin \alpha_i)), \] (32)

and,

\[ F_2(\alpha_i, t_i) = \frac{[v_M - (h_M + t_i) \cot \theta - l_2 + (t_i + c_1 + l_2 \tan (\sin^{-1} (n \sin \alpha_i))) \cot \alpha_i] (\cot \theta - 1)}{\cot \alpha_i - \cot \theta} + v_M - (h_M + t_i) \cot \theta \] (33)

Using the Chain Rule and implicit differentiation we obtain \( \frac{d\alpha_i}{dt_i} \). Finally,

\[ \frac{d\alpha_i}{dt_i} = - \frac{\frac{\partial F_1}{\partial \alpha_i} + \frac{\partial F_2}{\partial \alpha_i}}{\frac{\partial F_1}{\partial t_i} + \frac{\partial F_2}{\partial t_i}} \] (34)

\[ \frac{\partial F_1}{\partial t_i} = 1; \] (35)

\[ \frac{\partial F_1}{\partial \alpha_i} = l_3 \sec^2 (\alpha_i) + \frac{l_2 n \cos (\alpha_i)}{(1 - n^2 \sin^2 (\alpha_i))^{3/2}} \] (36)

\[ \frac{\partial F_2}{\partial t_i} = \cot (\theta) \frac{(\cot (\alpha_i) - \cot \theta + 1)}{(\cot (\alpha_i) - 2 \cot \theta)} \] (37)

\[ \frac{\partial F_2}{\partial \alpha_i} = [K \csc^2 (\alpha_i) + l_2 (\cot (\theta) - 1)] [(\cot (\alpha_i) - \cot \theta) K - K + \tilde{K} \cot (\alpha_i)]; \] (38)

where

\[ K = \frac{\cot (\alpha_i) \cos (\alpha_i) \sec (\arcsin (n \sin (\alpha_i)))}{\sqrt{1 - n^2 \sin^2 (\alpha_i)}}; \quad \tilde{K} = \left[ v_m - (h_m + t_i) \cot \theta - l_2 (t_i + c_1) \right]. \]

This approach is work in progress since we were not able to validate it computationally, however, we believe the theoretical and analytical basis for this model are strong.
(a) ‘Down’ vector at horizontal position \( x \) and angle \( \phi \) from vertical.

(b) ‘Up’ vector at horizontal position \( x \) and angle \( \phi \) from vertical. \( \phi \) is measured as the supplement of in the ‘down’ case.

Figure 3: Illustration of ‘vectors’ as used in the operator method. Note that the difference between an ‘up’ vector and a ‘down’ vector is equivalent to viewing upward motion as down motion in the reflection of the domain across the wafer.

2.3 Modeling Approach 2: Composition of Simple Operations

A different approach that is more straightforward but also more computationally intensive could also be taken. In short, we

1. Define simple mathematical objects,
2. Compose them in complex ways,
3. Evaluate these compositions with a symbolic computer algebra system, and
4. Evaluate expressions numerically where necessary.

Consider ‘vectors’ \((\phi, x)\) where \(\phi\) is the angle the beam travels from vertical and \(x\) is the horizontal position of the beam as shown in Figure 3. We can then define operations on these ‘vectors’ that correspond to motion forward along the beam’s path. For example, advancement along the path corresponding to a vertical distance \(l\) can be defined

\[
T_l(\phi, x) = (\phi, x + l \tan(\phi)).
\]

Table 1: Parameters used in simulation. The mirror height at a position \(x\) is given by

\[ h_M = b_M + (x - x_L) \tan(\beta_M) \]

where \(x_L\) is the horizontal position of the laser.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>0.06981317007977318 radians (4 degrees)</td>
<td>Angle laser makes with vertical</td>
</tr>
<tr>
<td>(n_A)</td>
<td>1.0003</td>
<td>Index of refraction of air</td>
</tr>
<tr>
<td>(n_N)</td>
<td>1.0003</td>
<td>Index of refraction of nitrogen gas</td>
</tr>
<tr>
<td>(n_Q)</td>
<td>1.456</td>
<td>Index of refraction of fused silica (quartz)</td>
</tr>
<tr>
<td>(l_0)</td>
<td>0.2514 m</td>
<td>Distance from laser to top of quartz plate</td>
</tr>
<tr>
<td>(l_1)</td>
<td>0.0032 m</td>
<td>Thickness of quartz plate</td>
</tr>
<tr>
<td>(l_2)</td>
<td>0.007 m</td>
<td>Distance from quartz plate to wafer</td>
</tr>
<tr>
<td>(W)</td>
<td>0.101 m</td>
<td>Distance from laser to sensor (width)</td>
</tr>
<tr>
<td>(x_{\text{start}})</td>
<td>0.015 m</td>
<td>Start position of scan on wafer</td>
</tr>
<tr>
<td>(x_{\text{stop}})</td>
<td>0.135 m</td>
<td>End position of scan on wafer</td>
</tr>
<tr>
<td>(N_{\text{step}})</td>
<td>50</td>
<td>Number of measurements taken across wafer</td>
</tr>
<tr>
<td>(b_M)</td>
<td>0.276 m</td>
<td>Height of mirror (see caption)</td>
</tr>
<tr>
<td>(\beta_M)</td>
<td>0.698 radians (40 degrees)</td>
<td>Angle of mirror (see caption)</td>
</tr>
</tbody>
</table>
Similarly, the change of angle upon hitting an interface and reflection off a surface can be defined as

\[ I_{n_1,n_2}(\phi, x) = \left( \sin \left( \frac{n_1}{n_2} \sin(\phi) \right), x \right) \]  

\[ R_{\beta_S}(\phi, x) = (\phi + 2\beta_S, x) \]

respectively where \( n_1 \) and \( n_2 \) are the indices of refraction of the two materials and \( \beta_S \) is the angle of the surface normal vector. We can simplify computation by considering the path away from the surface with the same notation and use the complement of the previous angle as in Figure 3. Finally we need a functional mapping from an upward facing ‘vector’ above the quartz plate (if present) to a vertical sensor position. This is given by

\[ S_h(\phi, x) = x_M \cos(\beta_M) + b_M + \frac{(W - x_M) \sin(\frac{\pi}{2} - 2\beta_M + \phi)}{\sin(\phi) \cot(\beta_M) - \cos(\phi)} \]

where the x-coordinate of contact with the mirror, \( x_M \), is given by

\[ x_M = \frac{(y - b_M) \sin(\phi) - x \cos(\phi)}{\sin(\phi) \cot(\beta_M) - \cos(\phi)} \].

With these tools we can easily obtain expressions for quantities like the position a beam hits the sensor given an initial condition. For example, if the laser is at height \( L = l_0 + l_1 + l_2 \) and position \( x \) at angle \( \alpha \), the position the beam hits the sensor is

\[ d^{NQ} = (S_0 \circ R_{\beta_S} \circ T_L)(\alpha, x). \]  

With a quartz plate, the sensor reading is

\[ d^Q = (S_{l_1+l_2} \circ I_{n_{A},n_A} \circ T_{l_1} \circ I_{n_N,n_N} \circ T_{l_2} \circ R_{\beta_S} \circ T_{l_2} \circ I_{n_{Q},n_{Q}} \circ T_{l_1} \circ I_{n_A,n_A} \circ T_{l_0})(\alpha, x). \]

These formulas are impractical to compute by hand, but computer algebra systems were used, turning these into explicit formulas. For example, for a semicircle centered at \( x = c \) with radius \( R \) and no quartz plate,

\[ d^{NQ} = b_M + \left( W + \frac{b_M \sin(\alpha + A) + L \cos(\alpha + A) \tan(\alpha)}{\sin(\alpha + A) \cot(\beta_M) - \cos(\alpha + A)} \right) \cos(\alpha - 2\beta_M + A) - \frac{b_M \sin(\alpha + A) + L \cos(\alpha + A) \tan(\alpha)}{\sin(\alpha + A) \cot(\beta_M) - \cos(\alpha + A)} \cot(\beta_M) \]

\[ A = 2 \tan \left( \frac{\text{sign}(R)(x - c + L \tan(\alpha))}{\sqrt{R^2 - (x - c + L \tan(\alpha))^2}} \right). \]

These formulas can be generated as needed without regard for complexity as the symbolic manipulations are done by computer.

These relationships define an implicit relationship between the computed value of \( R_Q \) and the actual value of \( R_{NQ} \). This is equivalent to finding a relationship between \( \theta \), the angle the beam is reflected from the wafer and \( d \), the position (height) the beam hits the sensor. It is useful to adopt a paradigm of considering two independent interpretations of one sensor reading.

1. One in the universe of machine computation (without a quartz plate) and
2. One in the the universe of better modeling (with a quartz plate).\(^9\)

In both cases we assume that we can approximate the wafer as an arc. We then have that for any laser position \( x \),

\[ d^Q(x, \theta) = d^{NQ}(x, \theta). \]  

Because there exists a one-to-one correspondence between \( \theta \) and \( R_{NQ} \) for each fixed \( x \), we will abuse notation to write

\[ d^Q(x, R_{NQ}) = d^{NQ}(x, \theta). \]

It is then possible to obtain a function \( R_{NQ} \mapsto R_Q \) by doing the following:

\(^9\)What inversion the LD does to compute \((x_j, \theta_j)\) is irrelevant as long as we assume it is done correctly (i.e., this is the angle consistent with sensor position \( d^{NQ} \)) and it ignores the quartz plate.
Figure 4: A plot of \((x_j, \theta_j)\) as inferred from modeled sensor positions for a wafer of radius \(R = 50\) m.

Figure 5: A plot of \((x_j, \theta_j)\) reported by the LD for a calibration wafer with a radius of \(R = 50\) m.

1. Choose a collection of positions \(\{x_j\}\) consistent with the scanning range and number of steps taken by the LD.
2. For each use the relationship in Equation (49) to obtain pairs \(\{(x_j, \theta_j)\}\).
3. Numerically compute the derivative of \(\theta(x)\).
4. Utilize the relationship in Equation (4) to find \(R_Q\).

Steps 3 and 4 require further clarification. First, as suggested by the device manufacturers and our own analysis (Section 2.1), the cosine term in Equation (3) is negligible; \(\theta'\) is nearly constant. Additionally, though this is not a constraint in modeling, the measured data is noisy, requiring redundant data. Thus we use linear regression to attempt to model the LD’s approach to computing \(R^{-1}\), as would be done for machine readings. However, for computations with noiseless, simulated data, it is sufficient to compute reflected angles at only a few simulated laser positions.

2.3.1 Model Validation

There are two ways that we can verify that this model is correct and implemented correctly. The first is that we are modeling the entirety of the system at hand and can verify intermediate computations are reasonable and reversible. The second, and a key advantage of this method, is that we can use some of the LD data in \((x_j, \theta_j)\) to verify that our model replicates the desired computation (Figures 4 and 5).

We can draw a few conclusions from this experiment. First, our model matches the sensor data up to a difference in chosen reference position. Second, our data does not have the sensor noise present in the solution, but does show evidence of the cosine term neglected in Equation (8). The scale of this
Figure 6: A plot of \((x_j, \theta_j)\) as inferred from modeled sensor positions with simulated sensor noise for a wafer of radius \(R = 50\) m. This is equivalent to a Figure 4 in all respects except for the addition of noise.

simulated error provides additional supporting evidence that this approximation is reasonable. Third, the computation is stable under slight addition of noise in the detector position, a necessary condition for the model and practical computations (Figure 6). The validation done here is limited, in part by the a dearth of specifications of the deflectometer and data available to directly validate against. Additional work in this regard would be required to increase trust in the values the model predicts, but the authors have sufficient confidence in the model to allow it to provide insight to a statistical approach toward compensating for additional factors.

2.3.2 Correction Curves
The most important relationship for this model to predict is between \(R_{NQ}\) and \(R_Q\) (or their reciprocals) to explore possible correction methods (Figure 7). We can make some observations from the modeled results. We predict the miscomputed value to be slightly smaller than the actual value. Further, we predict the relationship between the two to be linear with intercept zero.\(^{10}\) As a result, for the model, linearity holds in curvature as well.

2.3.3 Sensitivity
One insight that is very difficult to obtain experimentally is variation upon the parameter set. To do so we consider the effect on the slope in Figure 7 upon scaling each parameter by a factor near one; these are shown in Figure 8. The vertical offsets of the mirror and the laser appear to be more significant than the other parameters. However, this may not be representative of absolute effect as these two lengths are far larger than the others in the model. Additionally we observe the angle of the mirror and the height of the quartz plate above the sample are not insignificant in determination of a correction curve, but our model does not suggest they are of primary concern.

2.4 Ability to Correct Data and Radius Values
The goal of modeling the laser deflectometer is three-fold. First, modeling can verify understanding of the mechanism and corresponding computations. Second, modeling can guide a statistical approach towards developing a method to correct for a quartz plate. Third, if well tuned, a model could give explicit corrections to radii based on either \((x_j, \theta_j)\) pairs or \(R_Q\) values. Tools to accomplish these tasks are written for experimental purposes, but are not ready for practical use. With further development this may be a feasible application of these models.

\(^{10}\)For a flat surface, it is clear \(\theta'(x) = 0\) for any \(x\). When considering small curvatures, this prediction is based upon results the model produces.
3 Data Analysis

3.1 Data Description

Experiments were conducted on 34 different wafer specimens, and each experiment returns 5 types of experimental data,

- **Quartz**: when a “quartz”, “glass”, or “window” is referred to throughout this document this means a transparent and thermally fused silica insulating shield is used to enable temperature regulation of the sample.
- **Radius**: when the laser scans from one side of the wafer to the other, it is possible to estimate the radius of the (approximate) arc that matches the deformation of the wafer. In addition, we also use **curvature** for analysis which is the reciprocal of radius.
- **Bow**: a bow is defined as the difference from the zero surface to the horizontal line of the edge of warpage.
- **Intensity**: the power transferred per unit area, where the area is measured on the plane perpendicular to the direction of propagation of the energy. In our project we consider it to be the amount of light reaching the detector in terms of volts. The intensity for the 670nm laser is nominally 3.08 volts, and for the 780nm laser is nominally 2.75 volts. The higher the intensity is, the more accurate the result will be.
- **Wavelength**: the wavelength contains two types of lasers, the 670nm is the Red laser and the 780nm is the Infrared laser.

The experiments fall into 4 different categories:

- Measurements with quartz and without nitrogen denoted by $Q$;
- Measurements without quartz and without nitrogen denoted by $NQ$;
- Measurements with quartz and with nitrogen denoted by $QN_2$;
- Measurements without quartz and with nitrogen denoted by $NQN_2$.

We will use $R$ to refer to radii and denote curvatures by $K = R^{-1}$.

Figure 7: The relationship between $R_{NQ}$, the ‘correct’ radius of curvature, and $R_Q$, the incorrectly computed radius of curvature. The relationship between the inverse is also considered.
3.2 Analysis on the Data

First, we plot the whole data set of radius, curvature, bow and intensity to investigate the potential pattern in the four types experiments. Moreover, the whole data-set consists of 35 different thin films and each specimen has different data when measured at room temperature.

In Figure 9 we show the radius of curvature for all the wafer specimens. It is clearly seen that most of the large values of $R$ occur when looking at a case with quartz ($Q$) or quartz with nitrogen ($Q N_2$). By taking the reciprocal of the radius, we can also plot the curvature of each specimen, as seen in Figure 10. For the curvature data, we observe similar patterns between neighboring groups. This leads us to assess different testing setups individually in order to determine what causes this pattern. Moreover, Figure 11 shows that the bow data has a relation with the curvature data which is an inverse version of curvature data with some factor.

3.3 Individual Set Analysis

The whole data-set contains 35 different thin film specimens, and each specimen has different data. We isolate each specimen data set individually, and make plots for the thin film with the quartz window and without quartz window to observe the pattern.

In the case which contains two different types of wavelength, we can observe the influence of the quartz from the 35 data-sets as follows (if we consider the thin film concave up as positive):

- The data of bow and radius for the same thin film are always in opposite sign. Since if the thin film is concave up, then the bottom line to measure the bow is lower than the flat position (if we treat it as 0-surface).
- If the radius measured for the non-quartz case and quartz case are both positive (concave up), then if we put quartz on, the thin film looks more concave up.
- If the radius measured for non-quartz case and quartz case are both negative (concave down), then with the quartz on, the thin film looks flatter (opposite to the previous one), the radius may change to a different sign (if the radius is large enough, then the sign does not matter).
- If the measured radius of the non quartz case showed that the thin film had a very large positive/negative radius (flatter), then when the quartz window is on, you likely obtain a smaller and positive radius as the machine result (mildly concave up).
The smaller the radius is, the more accurate the result for the with-quartz case will be. If the magnitude of radius is less than 20, then the test may have a negligible error. If the magnitude of radius is less than 100, then the error caused by reflection of the quartz window of radius will be within ±20.

For an individual data-set, if the radius of the quartz and non-quartz cases has the same sign in Figure 12a, then the radius of NQ is larger than the radius of Q for same time section. In addition, if in the data-set, the radius of the quartz and non-quartz has opposite sign, like the shape in Figure 12b then radius of NQ is always negative. Figure 12c shows a perfect data-set to analyze the trend of impacts from quartz window. Data-set 8-12 in Figure 12d has a huge jump in non-quartz case from quartz case since the magnitude of radius is too small. In Figure 12d data-set 1-3, where the radius for quartz and non-quartz cases is negative, the magnitude of $R_Q$ is larger.

In summary, the quartz window will always make the thin film specimens happier.

3.4 Data Cleaning Methodology

After analyzing the whole data set and then breaking it into individual components, we find that for each data set, the fluctuations depend on record number (which is a serial number). In a real situation, the experimental results are affected by a variety of factors, with quartz or without quartz, and also with nitrogen or without nitrogen. We only consider data collected at room temperature using quartz, non-quartz, nitrogen and without nitrogen data and we remove the non-room temperature data. Therefore, if two record numbers have a large gap between their record number, which means two tests are done in different time periods, then the thin film might have deformed or their properties might have changed. To clean the data and to get corresponding data of quartz vs non-quartz, and quartz non-nitrogen vs quartz nitrogen, we use the following methodology to clean the data-set. The data can be treated as corresponding data if their record numbers are continuous or close, and the average is taken for same type and same time period data.

Figure 13 shows that for different data indices, there is always a difference between quartz data and non-quartz data. For data indices in the range from 15 to 20, the differences are larger than the rest value, while for data indices in the range from 48 to 52, the differences are smaller than the rest. Figure 14 shows the differences between cases with and without nitrogen. It has the same data index as Figure 13, where the greatest difference is observed.
3.5 Cleaning Set Analysis

Since we observe the pattern of the radius from the individual analysis for $R_{NQ}$ and $R_Q$, it is reasonable to further explore effects of the radius. The overall radius plot for all 34 data-sets can be seen in Figure 15.

In Figure 16, we hypothesize a function $R_{NQ} = -\frac{1}{R_Q} - f$, for some $e, f > 0$. However, there are many types of models that could be used to fit this graph, and it can take a lot of time finding the coefficients to fit each model to the data. Therefore, for the next step, we consider comparing $K_{NQ}$ with $K_Q$, where $K_Q = 1/R_Q$, and plot them as $K_{NQ}$ versus $K_Q$ in Figure 16.

By observing Figure 13 a straight line pattern suggests fitting a linear model. The correlation between two variables is 0.9997 which supports the conclusion of a strong linear relationship (closer to 1 implies linear). Even though we have some missing data (like the experiment data of radius in the range of [10, 12] and [16, 50]), from the data we have, it is still more reasonable to fit with a linear model, such as $K_{NQ} = aK_Q - \epsilon$. Since we use curvature, the dependent variable ($K_{NQ}$) only lies in a small range, i.e. [-0.02, 0.1]. However, if you flip it over, then there will be a significant difference indicating using $\epsilon$ to represent a small number for the shift from the original point. We observe from the graph that the slope of the linear model is almost one. Thus, the linear model has more similarities with $K_{NQ} = K_Q - \epsilon$, $\epsilon > 0$.

We consider the need for the term $\epsilon$ as being due to the following reasons:

- When we add a quartz window, the machine looks at the thin film though quartz differently. From previous analysis, we observed that the quartz window always makes the machine think that the thin film is smiling more. The changes in radius induced by the quartz become larger if the radius with quartz on is flatter and concave down.

- Many factors in the test environment can affect the result of the experiment on the specimen. When people in industry perform different tests (like with quartz or without), they cannot be carried out at the same time. If the experimenter changes the environment frequently (like putting on the quartz or letting the nitrogen flow), then it may cause the thin film to change properties in a short time. There can also be error due to imperfections in the quartz window. The measurements obtained via the tracking laser may also be an additional source of error.
3.6 Modeling

We use linear regression to fit the curvature data and have the quartz model:

\[ K_{NQ} = 1.001 K_Q - 0.0045 \]  

(50)

Table 2 shows how well the statistical model fits the data set. It is observed that 99.94% of the total variation in curvature without quartz can be explained by the linear relationship between curvature with quartz and without quartz (as described by the regression in Equation (50)). In addition, the estimate of error standard deviation is small. We used hypothesis tests with P-values both less than 0.0001, indicating that the model seems more reliable. Then we also plot other features such as a residual (the predicted curvature minus the real curvature), see Figure 18.

Table 2: The comparison of real radius and predicted radius of the nitrogen model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Standard Deviation</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>-0.004462</td>
<td>0.00011</td>
</tr>
<tr>
<td>Slope</td>
<td>1.00407</td>
<td>0.00267</td>
</tr>
</tbody>
</table>

In Figure 19 we zoom in on a small area \( R_Q \in (-500 < R_Q < 2000) \) and \( R_{NQ} \in (-2000 < R_{NQ} < 2000) \). There are 2 vertical and horizontal lines representing singularities at \( x = 224.9804 \approx 225 \) and \( y = -224.9804 \) (meters), which are also present in Equation (51), where we flip the curvature to get the relationship between radii.

\[ R_{NQ} = \frac{1}{1.0040659/R_Q - 0.004462903} \]  

(51)

In Figure 20, when the radius with quartz lies in the range \([-500, 0]\) meters, we note that the radius without quartz is also negative. The slope of the region gets close to 0 when the magnitude of the radius with quartz becomes larger. The plot shows when we have the measured data for radius with quartz it is concave down a bit, and more concave (smaller radius in magnitude) than the measured result.

We can also observe in Figure 20 that when the radius with quartz lies in the range \([0, 225]\) meters, then the radius without quartz is all positive. The slope in this region becomes steeper as the radius
without quartz becomes larger. If we have measured data of a thin film with quartz mostly concave up, we observe that measurement data having a radius close to 225 meters is actually a very flat surface.

Moreover, when the radius with quartz lies between the singularity and positive infinity, i.e. in the interval \([225, 2000]\) meters, the radius without quartz becomes negative again. The curve approaches a horizontal asymptote as the radius with quartz becomes larger. This means that, if the machine looks though the thin film with a quartz window, then it may observe the radius to have an opposite sign change due to the large radius.

If we plot relative error with and without the quartz, i.e. \(R_Q - R_{NQ}\) with respect to \(R_{NQ}\), then we have the plots and model corresponding to Equation (51) and Figure 21, respectively, where the red line is a fitted curve. From the graph, we can conclude that the error of quartz is getting larger when \(R_{NQ} > 0.45\) and \(R_{NQ} < -448.59\) meters. However, when \(R_{NQ}\) is close to the singularity point on both left and right sides, the graph shows an opposite trend.

Next, we observe how nitrogen affects the radius. In Figure 22, it is hard to observe a clear relation between the two variables, and the correlation is low. However, if the radius is large enough, we can treat the warpage as flat. Therefore, we decided to consider only \(R_{QN}^2 \in [-1000, 1000]\), which turns out be a very linear region. The correlation between the two variables is \(0.99860678\).

We fit the linear model for \(R_Q\) and \(R_{QN}^2\) as we did for \(K_Q\) and \(K_{NQ}\) previously, and we plot the radius relationship with nitrogen flow and the 95% confidence interval in Figure 23, and the associated residual plot in Figure 24. The equation of the nitrogen model is

\[
R_Q = 1.000443R_{QN}^2 + 0.000067. \tag{52}
\]

From Equation (52), we can conclude that the nitrogen only slightly affects the radius since the slope is almost one and the intercept is close to 0. If the measured radius with nitrogen is small, then the difference is negligible. However, when the measured radius gets larger (above 500 meters), then the predicted nitrogen model will become less accurate as shown in Figure 24b. From equation (52), we can also conclude that \(|R_{QN}^2| < |R_Q|\).
3.7 Model Analysis

The test did not provide enough paired data for the relationship between $R_{NQ}$ and $R_{QN_2}$. However, since we have the prediction for two models, it is reasonable to combine them. We get the mapping from $R_{QN_2}$ to $R_{NQ}$, which is the relationship:

$$R_{NQ} = \frac{1}{\bar{a} + eR_{QN_2}} - \bar{\epsilon} = \frac{b + cR_{QN_2}}{a - eb - ceR_{QN_2}} = \frac{b + cR_{QN_2}}{\bar{a} - \bar{\epsilon}R_{QN_2}},$$

(53)

where $\bar{a} = a - eb$, $\bar{\epsilon} = ec$ and the coefficients are $a = 1.00406$, $b = 0.000067$, $c = 1.000443$, $\epsilon = 0.00446$, $\bar{a} = 1.04060$ and $\bar{\epsilon} = 0.0044649$.

Figure 25 shows model mapping from $R_{QN_2}$ to $R_{NQ}$. The dashed lines represent coordinates of $R_{QN_2} = 0$ and $R_{NQ} = 0$. Moreover, we still have $R_{QN_2} = 224.8808 \approx 225$ and $R_{NQ} = -224.8808$ in meters for the singularity point in the denominator of Equation (53). It looks very similar to Figure 20, since nitrogen only affects the radius a little. The analysis follows the same methodology as that used for the quartz model.

Next, we can also apply the nitrogen model to map the radius without quartz window and with nitrogen flow to the radius without quartz and nitrogen, which is mapping $R_{NQN_2}$ to $R_{NQ}$. Since we are short of data for the radius without quartz window and with nitrogen flow, we predict based on the nitrogen model, and the resulting equation will look like

$$R_{NQ} = 1.000443R_{NQN_2} + 0.0000673,$$

(54)

allowing us to link all 4 cases together.

3.8 Error Analysis

In error analysis, it is important to understand how to express data, and how to analyze and draw meaningful conclusions from it. Thus descriptive statistics are used to describe the thin film specimen data.

3.8.1 Nitrogen Model Error Analysis

From the real and the predicted data, we observed that the error is very small, possibly occurring as a result of variations in the measurements. The experimentalists have little or no control over variations in
the measurements, or the error may be caused by the manner in which the experiment was conducted. In Figure 26, all the values seem to move closer to zero and clusters are between the 20th to 40th and between the 60th and 75th observations on the horizontal line. The number of outliers is small, thus the model seems reliable. In addition, the standard deviation of the absolute error in Table 3 is an indication of the dispersion of the residuals. The relative error scales the absolute error with respect to $R_Q$. The standard deviation of the relative error indicates that the dispersion of the error values are only a little further away from zero. Since the standard deviations are small, the predicted values appear to be reliable. The 95% confidence levels for the quartz model predictions were found to lie within the range $[R_Q - 2.124, R_Q + 2.124]$.

3.8.2 The Quartz Model Error Analysis for the Thin Film

The error of the real and predicted data is very small, possibly because the model produced outputs that are close to the real values.

In Figure 27, most of the errors are approaching zero with few outliers, meaning that the errors are quite small. Since the standard deviation is small in both absolute and relative sense, the dispersion of the error is small enough and thus the model is reliable. 95% confidence level means that all the values for the curvature falls within the range of $K_{NQ} \pm 0.000168$.

In Figure 28 as some of the curvature values become very small most of the error values become closer causing a cluster between -0.02 and 0. Also as the value of the curvature increases the relative error is smaller and some values lies on the horizontal line where relative error is 0, suggesting an accurate result. Considering the standard deviation of the relative error, which is quite small, we say that the error is dispersed very small.

3.9 Validation

For data validation, we use two data-sets to test our nitrogen model and quartz model. Table 4 shows the prediction of the nitrogen model from $R_{QN^2}$ to $R_Q$ and from $R_{NQN^2}$ to $R_{NQ}$ and compares the predicted radius to the real radius. The relative errors are calculated and indicate that the nitrogen model has an accurate prediction when the magnitude of the radius is small.

In Table 5, we test our combined model to predict the range of the radius. Both of the models predict well when the radius is in the range $[-500, 1000]$, and exhibit large relative error for a flat thin film. Therefore, the conclusion can be drawn that the nitrogen model works well when the radius is in...
Figure 15: The plots of $R_{NQ}$ and $R_Q$ for all data-set.

Figure 16: The plots of $K_{NQ}$ and $K_Q$ for all data-set.

[−500, 1000] which supports our modeling analysis. We also acquired new samples from Sandia for data testing. From the output of Sample 1 and Sample 2, the predicted $R_{NQ}$ range lies inside the actual $R_{NQ}$ range to within one standard deviation. These results indicate that our model is appropriate, noting that the large error bound in Sample 1, is due to the machine error.

4 Conclusions

In summary, we analyzed the experimental data, and used a linear model to fit the reciprocal of the quartz and non-quartz radius, which is the quartz model, and also for the nitrogen and non-nitrogen
radius, which is the nitrogen model. The correlation coefficients of the quartz model and the nitrogen model are 0.9997 and 0.9986, respectively, indicating a strong linear relationship in the data. For the quartz model, when the radius with quartz is larger than the singularity ($R_Q \approx 225m$), the radius without quartz is negative and approaches a horizontal asymptote. When the quartz radius is smaller than the singularity and goes to negative infinity, the radius without quartz starts from positive infinity and approaches a negative asymptote. We also provide the testing results that demonstrate that the model works when the magnitude of the radius is small. The geometry-based approach has potential to be efficient since we only need to evaluate an analytical expression, however, it is error prone due to the high number of terms needed. On the other hand, the operator-based approach is more flexible though more computationally intensive.

5 Recommendations

• Acquire more data at room temperature using the quartz plate, especially with samples of radii.

Figure 18: Residual plots with different x-axis for quartz model.
Figure 19: The plots of fitting simple linear regression for all radius.

Table 3: The Error Analysis for the Quartz Model and Nitrogen model.

<table>
<thead>
<tr>
<th></th>
<th>Quartz model</th>
<th>Nitrogen model</th>
</tr>
</thead>
<tbody>
<tr>
<td>std</td>
<td>7.72e-4</td>
<td>3.30e-2</td>
</tr>
<tr>
<td>mean</td>
<td>4.70e-3</td>
<td>3.30e-2</td>
</tr>
<tr>
<td>var.</td>
<td>5.95e-9</td>
<td>6.42e-4</td>
</tr>
<tr>
<td>conf.</td>
<td>1.69e-4</td>
<td>1.73e-2</td>
</tr>
</tbody>
</table>

- Use the correction formulae in terms of curvature.
- When we have large enough data-sets, then the prediction from $R_{QN2}$ to the $R_{NQ}$ can be separated into different magnitudes of the radius. For example separating $|K_{QN2}| < 100$ and $100 < |K_{QN2}| < 1000$ and $|K_{QN2}| > 1000$, can improve predictive accuracy of the model.
- The two statistical models are recommended to be used only in range [-500,1000].
- To further improve the model for every time frame, perform the quartz, non-quartz, quartz with nitrogen and quartz without nitrogen measurements before and after the experiment.
- To further develop the physical models, refine estimates for key parameters.
Figure 20: Zoomed in the plots of fitting simple linear regression for all radius.

Table 4: The comparison of real radius and predicted radius of the nitrogen model.

<table>
<thead>
<tr>
<th></th>
<th>Set 1.1</th>
<th>Set 1.2</th>
<th>Set 2.1</th>
<th>Set 2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>real $R$</td>
<td>40.69</td>
<td>49.08</td>
<td>261.28</td>
<td>-1829.00</td>
</tr>
<tr>
<td>pred. $R$</td>
<td>40.51</td>
<td>48.93</td>
<td>255.72</td>
<td>-2258.86</td>
</tr>
<tr>
<td>rel. err.</td>
<td>0.43%</td>
<td>0.30%</td>
<td>2.13%</td>
<td>23.50%</td>
</tr>
</tbody>
</table>

Figure 21: The plots for difference in the radius vs $R_{NQ}$. 
Figure 22: The plots and 95% CI for the radius with nitrogen environment.

Figure 23: The simple linear regression of the nitrogen model for $R_{QN_2} \in [-500, 1000]$.

Table 5: The comparison of real radius and predicted radius of the quartz model.

<table>
<thead>
<tr>
<th>Testing Data</th>
<th>Input range of $R_{QN_2}$ (m)</th>
<th>Predict Range of $R_{NQ}$ (m)</th>
<th>Real $R_{QN}$ (m) range within 1 standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample 1</td>
<td>[254.50, 265.03]</td>
<td>[-1925.24, -1479.14]</td>
<td>[-1925.34, -1479.09]</td>
</tr>
<tr>
<td>Sample 2</td>
<td>[40.57, 40.77]</td>
<td>[49.33, 49.62]</td>
<td>[49.31, 49.63]</td>
</tr>
</tbody>
</table>
Figure 24: Residual plots with different x-axis for nitrogen model.

Figure 25: The predict model for mapping from $R_{QN_2}$ to $R_{NQ}$. 
Figure 26: Absolute error for the nitrogen model.

Figure 27: Absolute error of the quartz model.
Figure 28: relative error against the curvature
References


Visualizing and Interpreting Machine Learning Models for Liver Disease Detection

Adams Kusi Appiah\textsuperscript{1}, Sharang Chaudhry\textsuperscript{2}, Chi Chen\textsuperscript{3}, Simona Nallon\textsuperscript{4}, Upeksha Perera\textsuperscript{5}, Manisha Singh\textsuperscript{6}, Ruyu Tan\textsuperscript{7}

Problem Presenters:
Agustin Calatroni and Petra LeBeau
Rho, Inc.

Faculty Mentor:
Emily Kang
University of Cincinnati

Abstract

The use of machine learning algorithms has been embraced in several areas of biomedical research. These algorithms are able to take into account high degree of interactions and create nonlinear response functions to make predictions with high accuracy. This success comes at the cost of diminished interpretability. This makes evaluation of machine learning methods’ trustworthiness difficult. In this work, several ad-hoc tools that leverage the predictive power of machine learning methods are explored to provide insights into the method’s decision-making rationale. For context, a generalized boosting machine is used to predict liver disease. The interpretability and visualization have been explored from several vantage points and are presented with insightful commentary.

Keywords
Gradient boosting machine, locally interpretable model-agnostic explanations, variable importance measures, partial dependence plots, individual conditional expectation plots, surrogate model

1 Introduction

The success of machine learning (ML) algorithms in medicine and multi-omic studies over the last decade have come as no surprise to ML researchers. This can be largely attributed to their superior predictive accuracy and their ability to work on both large volume and high-dimensional datasets. The key notion behind their performance is self-improvement. That is, these algorithms make predictions and improve by analyzing mistakes made in those predictions. The difficulty with this “predict and learn” paradigm is that these algorithms suffer from diminished interpretability. This is often referred to as the “black-box” nature of ML methods, and, typically, the loss in interpretability is due to the nonlinear and high number of interactions embedded within the resulting models.

In cases where interpretability is crucial, for instance in studies of disease pathologies, ad-hoc methods leveraging the strong predictive nature of these methods have to be implemented. These methods are used as aides for ML users to answer questions of the following nature: ‘why did the algorithm make certain decisions?’, ‘what variables were the most important in predictions?’, and/or ‘is the model trustworthy?’. In this work, interpretability of a particular class of ML methods called gradient boosting machines (GBM) has been studied on the prediction of liver disease status of patients. It should be pointed out that the scope of this work

\textsuperscript{1}Department of Biostatistics, University of Nebraska Medical Center
\textsuperscript{2}Department of Mathematical Sciences, University of Nevada, Las Vegas
\textsuperscript{3}Department of Biostatistics, State University of New York at Buffalo
\textsuperscript{4}Department of Statistics, California State University, East Bay
\textsuperscript{5}Department of Mathematics and Statistics, Sam Houston State University
\textsuperscript{6}Department of Information Science, University of Massachusetts
\textsuperscript{7}Department of Applied Mathematics, University of Colorado at Boulder
is not to improve or optimize performance of the GBM but to understand the method’s intrinsic latent behavior. Many methods have been developed because of the need for interpretation of the models. Typically, these methods tend to use relevant information required for answering specific questions that are of interest. Given that interpretation is a wide concept with a large scope, often a variety of them need to be used in conjunction. Some of the most commonly used methods are Variable Importance Plots (Hjerpe, 2016), Partial Dependence Plots (Lemmens and Croux, 2006), Individual and Conditional Expectations (Goldstein et al., 2015) among others. All three of these techniques compute statistics that account for the change in predictions if the value(s) of the input feature(s) were changed. Besides attempting to derive metrics from a change in predictive nature of the methods, the use of surrogate models has been suggested previously in literature (Murali and Carey, 2014). Surrogates are defined as interpretable models that typically produce similar predictions as the non-interpretable model of interest. The rationale is that models with similar predictive patterns may have similar decision-making ideology.

Broadly, the scope of interpretation is divided into areas. The first is global, where the entire dataset is used for interpretation and the second is local, where a subset of the data is used for deriving an interpretive analysis of the model. Variable Importance Plots and Partial Dependence Plots fall in the global category. As far as the use of surrogate models is concerned, it is possible to obtain them both at global and local levels. The intricacies of these methods are further discussed in the analysis section of the paper.

The remainder of this paper is organized as follows: Section 2 presents details on the implementation of the method, Section 3 presents several different visualization and interpretation tools, Section 4 provides a discussion and context for future work.

2 Data and method implementation

This section presents information about the data along with details about GBM and its implementation.

2.1 Data

To demonstrate and interpret GBM, the Indian Liver Patient Dataset is used. This dataset is openly available for public use (Dheeru and Karra Taniskidou, 2017). The data contains records of 583 liver patients from North East India. The data is comprised of eleven variables: ten independent variables and a response variable indicating disease status of the patient. Out of the 583 patients, 167 don’t have liver disease and 416 have liver disease.

In clinical practice, detection of liver disease involves monitoring levels of direct and indirect bilirubin (pigments formed during the breakdown of hemoglobin), alanine aminotransferase & aspartate aminotransferase (enzymes), and albumin (a protein made in the liver). Often, tests for additional enzymes like alkaline phosphatase and gamma-glutamyl transpeptidase can also be conducted. It has previously been explored in literature that gender and age can also be potential factors for liver disease (Luk et al., 2007). A dreadful fact is that liver disease is not easily discovered. The liver is capable of maintaining normal functions even when partially damaged. Thus, early diagnosis is one of the most important steps in liver disease treatment.

The following features were used to predict liver disease status:

- Age of the Patient: The patients range from ages 4 to 90 with a median of 45 years old.
- Gender of the Patient: There are 142 women and 441 men in the study.
- Total Bilirubin: Bilirubin is a yellow pigment that is formed in the liver to break down hemoglobin and found in blood and stool. Total bilirubin consists of both conjugated and unconjugated bilirubin. The normal levels of total bilirubin are 0.1 to 1.2 mg/dL.
• Direct Bilirubin (Conjugated Bilirubin): Direct bilirubin flows directly into the blood. The normal level for direct bilirubin is less than 0.3 mg/dL.

• Alkaline Phosphatase (ALP): Alkaline Phosphatase is an enzyme that is found in the blood and helps with breaking down proteins. Abnormal levels of ALP can indicate that the liver and gallbladder are not functioning properly. The normal range of alkaline phosphatase is from 44 to 147 IU/L.

• Alanine Aminotransferase (ALT): This enzyme is found in the blood and is a good indicator to verify whether a liver is damaged especially due to cirrhosis and hepatitis. ALT can be measured using a test called Serum glutamic pyruvic transaminase (SGPT). Normal levels of ALT are 20-60 IU/L.

• Aspartate Aminotransferase (AST): AST is an enzyme and can be measured using a test called Serum glutamic oxaloacetic transaminase (SGOT). Normal levels of AST range from 10 to 40 units per liter. High levels of AST indicate damage in an organ such as heart or liver.

• Total Proteins: Total proteins consist of the proteins albumin and globulin. The test for total proteins measures the amount of these proteins in your body. The normal range of this is between 6-8.3 g/dL.

• Albumin: Albumin is the protein that prevents the fluid in blood from leaking out into the tissues. The normal range for albumin is 35-55 g/liter.

• Albumin to Globulin Ratio: This is a good indicator of the disease status of the liver.

2.2 Gradient Boosting Machines

GBM is a class of machine-learning techniques that can be used to solve many real-world problems. The main idea behind GBM is to start with a weak learner and develop a learner with higher predictive power (Natekin and Knoll, 2013). A weak learner is a model that gives slightly better classification than the random classification. GBM consecutively fits new models to provide a more accurate estimate of the response variable at each step. The final model is composed of all models trained along the way. The optimization algorithm behind the GBM is gradient descent.

As presented in (Natekin and Knoll, 2013), the goal is to construct a functional dependence between the predictors and the response. Let \( f \) represent the functional dependency between the response and the covariates. Next, let the loss function be \( \Psi(y, f) \) and \( z(x, \theta) \) be a custom base-learner. Sometimes the solution to the parameter estimates is difficult to obtain. Therefore, it was proposed to choose a new function \( z(x, \theta_t) \) to be parallel to the negative gradient \( \{g_t(x_i)\}_{i=1}^N \).

\[
g_t(x) = \nabla \Psi(y, f(x)) \bigg|_{f(x) = f^{t-1}(x)}
\]

The new function increment is chosen to be correlated with \(-g_t(x)\). Proceeding, it is possible to replace the optimization task with the general least squares minimization:

\[
(\rho_t, \theta_t) = \arg \min_{\rho, \theta} \sum_{i=1}^{N} \left[ -g_t(x_i) + \rho z(x_i, \theta) \right]^2
\]

An algorithmic overview of the gradient boosting algorithm, as originally proposed by Friedman (2001), is shown below.
Algorithm 1 Algorithm for obtaining learners through regression and clustering

Require: Input data \((x, y)^N_i=1\)

Require: Number of iterations \(M\)

Require: Choice of the loss-function \(\Psi(y, z(x))\)

Require: Choice of the base-learner model \(z(x, \theta)\)

1: Initialize \(\hat{f}_0\)
2: For \(t = 1\) to \(M\) do
3: Compute the negative gradient \(g_t(x)\)
4: Fit a new base learner function \(z(x, \theta_t)\)
5: Find the best gradient descent step-size \(\rho_t; \rho_t = \arg \min_\rho \sum_{i=1}^N \Psi[y_i, \hat{f}_{t-1}(x_i) + \rho z(x_i, \theta_t)]\)
6: Update the function estimate \(\hat{f}_t = \hat{f}_{t-1} + \rho_t z(x, \theta_t)\)
7: End for.

2.3 Software

The GBM was fit using H2O using a large parameter search grid. H2O is an open source platform that many companies and researchers use to interpret and address complex problems (Kochura et al., 2017). H2O is very versatile because it can be used as a stand alone package or through R and Python. H2O contains many popular machine learning algorithms including generalized linear models. Random forest and gradient boosting are also popular algorithms. Once the models have been created, H2O can be used to make predictions on these models and compare their results.

3 Analysis

The accuracy of prediction of the GBM on the liver disease status is computed on both training (95.6%) and testing (67.2%) datasets. The point of this was to grossly check whether or not the model is worth interpreting. Clearly, from the extremely high predictive accuracy on the training dataset, it can be believed that the GBM has found an effective rationale. Furthermore, even though there is a dip in the accuracy from training to testing, the GBM maintains an appreciable predictive level that shouldn’t be disregarded. The rest of this section presents applications of the visual and interpretability tools of the GBM on the Indian Liver Patient Dataset. Additionally, Table 1 & 2 present the confusion matrix detailing the result of the accuracy. As expected, on the main diagonal of these matrices, larger numbers corresponding to the correctly classified cases are observed.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>NO</th>
<th>YES</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>128</td>
<td>6</td>
<td>134</td>
</tr>
<tr>
<td>YES</td>
<td>11</td>
<td>322</td>
<td>333</td>
</tr>
<tr>
<td>Total</td>
<td>139</td>
<td>328</td>
<td>467</td>
</tr>
</tbody>
</table>

Table 2: Confusion matrix for testing data

<table>
<thead>
<tr>
<th>Predicted</th>
<th>NO</th>
<th>YES</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>14</td>
<td>19</td>
<td>33</td>
</tr>
<tr>
<td>YES</td>
<td>19</td>
<td>64</td>
<td>83</td>
</tr>
<tr>
<td>Total</td>
<td>33</td>
<td>83</td>
<td>116</td>
</tr>
</tbody>
</table>
3.1 Variable Importance Plot (VIP)

The VIP provides a list of the most significant variables in descending order by a mean decrease in the error. The top variables contribute more to the model than the bottom ones and have high predictive power in classifying liver disease subjects. This approach allows data scientists to weed out certain features that contribute less to the model.

To formalize the approach to plot variable importance in terms of the algorithm, we assume \( \hat{f} \) is the trained model, \( X \) is a \( N \times p \) feature matrix, \( Y \) represents \( N \) responses observed. The error is measured through a loss function \( L(Y, \hat{f}(X)) \). The details of algorithm can be found in Algorithm 2.

**Algorithm 2** Algorithm for obtaining permutation feature importance

1: Estimate the original model error \( \hat{\epsilon}_{\text{orig}} = L(Y, \hat{f}(X)) \)
2: For \( j = 1, ..., p \) do
3:     For \( m = 1, ..., M \) do
4:         Update feature matrix as \( X^{(j)} \) by permuting the \( j \)th feature \( X_j \) and replace it back into \( X \).
5:         Estimate error of model \( \hat{\epsilon}_{j,m} = L(Y, \hat{f}(X^{(j)})) \) based on the predictions of the permuted data.
6:     End for
7:     Obtain the average permutation feature importance for the \( j \)th subject \( FI_j = \frac{1}{M} \sum_{m=1}^{M} \{\hat{\epsilon}_{j,m} - \hat{\epsilon}_{\text{orig}}\} \).
8: End for
9: Obtain scaled importance score \( S_j = FI_j / \max\{FI_1, ..., FI_p\} \)
10: Sort variables by ranking its \( S_j \) in descending order.

A variable importance plot was created for the liver patient data. This plot helps identify which variables contribute the most in predicting liver disease of the GBM. From the VIP in Figure 1, alkaline phosphatase, age, and albumin are the three variables that have most contributions to the GBM. These variables also have high predictive power in classifying individuals as cases (liver disease) and controls (non-liver disease). However, aspartate aminotransferase, total bilirubin, and gender are the three least important features for GBM.

![Figure 1: Variable Importance Plot](image-url)

Although variable importance plots can be treated as a simple and explicit way to describe the importance of
each covariate in the model, they have limitations when the detail of marginal effect of a specific variable on the prediction needs to be studied. Additionally, VIP does not present an explanation if a variable’s importance has anything to do with its interactions with other variables in the model.

### 3.2 Partial Dependence Plots

The theoretical prediction function is fixed at a few values of the chosen features and averaged over the other features, which can be formed as

$$\hat{f}_{X_S}(x_s) = \int \hat{f}(x_s, x_C)d\bar{P}(x_C)$$

where $x_s$ is the set of features for which the partial dependence function should be plotted and $x_C$ are the other features that were used in the model. In practice, we are able to estimate $\hat{f}_{X_S}(x_s)$ via sample means from observations,

$$\hat{f}_{X_S}(x_s) = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(x_s, x_{C_i})$$

PDP is a visual aid that attempts to isolate the effect of a particular variable of interest on the response variable. All other variables in the system are global and graphical representations of the marginal effect that a set of variables have on the target field ignoring the rest of variables. Partial dependence plots can be used for classifying and regressing ensembles. PDPs represent the ensemble results not the data values. Partial dependence plots help with understanding how each predictor or variable affects the model predictions and helps answer the questions. PDPs marginalize over the values of all other features and show how each predictor affects the model’s prediction. Overall, partial dependence plots are a great way to gain trust and understanding from these complex models.

Three PDPs are presented in this report constructed on the training data; alkaline phosphatase (Fig. 2), alanine aminotransferase (Fig. 3), and total bilirubin (Fig. 4). It is possible to see that the predicted probability of classifying a liver disease patient decreases when alkaline phosphatase level increases mildly. Then, the predicted probability of liver disease increases as alkaline phosphatase level becomes greater. As the alanine aminotransferase and total bilirubin levels increase, the model shows an increase in the predicted probability of the disease.

![Figure 2: PDP for alkaline phosphatase](image-url)
Figure 3: PDP for alanine aminotransferase

Figure 4: PDP for total bilirubin
3.3 Individual Conditional Expectation (ICE)

It is known that the PDP is used for visualizing the average effect of a feature because it does not focus on specific instances, but on an overall average. A more specific version of the PDP that is at the subject level is called the individual conditional expectation (ICE) plot. Individual conditional expectation plots are newer and not as well-known. They are very useful when the explanatory variables have strong relationships.

Assume we have observations \( \mathbf{x} = (x_{Si}, x_{Ci}) \) for \( i = 1, ..., N \) and estimated response \( \hat{f} \). For a specific subject \( i \), \( x_{Ci} \) is set as the value observed, a curve of \( \hat{f}^{(i)}_S \) against a grid of values of \( x_{Si} \) can be plotted, where the x-coordinate represents the value of \( x_{Si} \) and the y-coordinate represents the value of \( \hat{f}^{(i)}_S \). This curve conveys the information of conditional expectation of \( f^{(i)}_S \). Finally, a cluster of \( N \) curves can be aggregated on the plot. The detailed algorithm for plotting an individual ICE curve can be summarized as:

We assume \( \mathbf{X} \) is a \( n \times p \) predictor matrix, \( S \in \{1, ..., p\} \), \( \mathbf{x}_C \) are the covariates to be fixed as constant values that are observed. Algorithm 3 describes how to obtain ICE plots.

**Algorithm 3 Algorithm for ICE**

1: Function ICE(\( \mathbf{X}, \hat{f} \))
2: For \( i = 1, ..., N \) do
3: \( \hat{f}^{(i)}_S = \mathbf{0}_{N \times p} \)
4: Set \( \mathbf{x}_C = \mathbf{X}[i, C] \)
5: For \( l = 1, ..., N \) do
6: Set \( \mathbf{x}_S = \mathbf{X}[l, S] \)
7: \( \hat{f}^{(i)}_S = \hat{f}(\| \mathbf{x}_S, \mathbf{x}_C \|) \)
8: End for
9: End for
10: Return \( [\hat{f}^{(1)}_S, ..., \hat{f}^{(N)}_S] \)

Figs. 5-7 represent the ICE plots for the three features selected in Section 3.2. In these figures, several lines, each corresponding to a different individual appear. This signifies how the change of a particular variable may affect different individuals in different manners.

Figure 5: ICE for alkaline phosphotase
Figure 6: ICE for alanine aminotransferase

Figure 7: ICE for total bilirubin
3.4 Surrogate Models

The tools presented in the previous sections do not present a system level interpretation of the model. To mitigate this challenge, the use of a surrogate models is proposed. Surrogates are models that have similar predictive patterns as our GBM but are inherently interpretable. The rationale is that similar predictive patterns could be a consequence of similar decision making processes. In essence, surrogate models are used to overcome the lack of transparency and interpretability in ML algorithms. The surrogate models train on the prediction made by the ML method of interest. Often, surrogate models are developed by training linear regression models or decision trees to capitalize on their inherent interpretability. Surrogate models can be produced at both the global and local levels.

3.4.1 Global

Two models were used in this work by training on the predictions of the GBM. These include a linear regression model and a decision tree. Fig. 8 shows the fitted decision tree.

![Figure 8: Decision Tree for liver disease prediction](image)

If successful, these methods can potentially unravel the complex structure of our GBM. This naturally leads to the question of how to measure success. To choose the best model for the visualization, the $R^2$ (Eq. 1) measure from linear regression models is borrowed. In this context, the surrogate models attempt to explain the variability in the predicted values of the GBM. Table 3 shows how well the four selected models explained the GBM.

$$R^2 = \frac{\sum_{i=1}^{n} (\hat{y}_i - \hat{y})^2}{\sum_{i=1}^{n} (y_i - \hat{y})^2}$$  \hspace{1cm} (1)

The decision tree with $R^2 = 89\%$ can be considered to be a potential surrogate that provides insight into the GBM, where as the linear regression model may not be a good fit since $R^2 = 56\%$. Interestingly, the decision tree classified total bilirubin as the most important variable, and alkaline phosphotase was put in the second line of decision making. This may lead to the belief that alkaline phosphotase’s rank in the VIP could have been caused by its interactive effect with total bilirubin (classified as low on the VIP). Note that the
two models used during this work do not form an exhaustive list of candidates and it is further possible to optimize the candidates selected in this work as well. This can potentially be explored in the future.

Table 3: R squared values for models

<table>
<thead>
<tr>
<th>Model</th>
<th>$R^2$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision tree</td>
<td>89</td>
</tr>
<tr>
<td>Linear regression without interactions</td>
<td>56</td>
</tr>
</tbody>
</table>

Given that a suitable global surrogate was not obtained but an existing desire to interpret the GBM, it can be thought that surrogates can be found locally since the local complexity is easier to approximate.

### 3.4.2 Local

It can be thought that the behavior of complex models can be approximated by simpler models when the data is considered at an individual level. This is akin to the thought that nonlinear curves can be approximated locally by lines. The idea behind the local interpretability is to focus on a specific data point and identify the neighborhood in the feature space around that point and then identify the model decisions based on this region. Local interpretability does not consider the fundamental assumptions and the structure of the model (Phillips et al., 2004). Now, since the complexity of the model is less of an issue, the local interpretation can be thought of as model-agnostic. This leads to the paradigm of Local Interpretable Model-agnostic Explanations (LIME). In this work, a R-package named LIME has been used for local interpretation.

In order to explain a particular prediction, the observation is permuted $n$ times around its neighborhood. Then the complex model is used to get predictions for all permuted observations. The LIME algorithm then uses a distance calculation method to calculate the distance between each permuted observation and the original observation. The data points around the neighborhood are weighted by their proximity to the original instance. Then a set of features that describe the prediction are selected. A simple model is fitted to the permuted observations to analyze the prediction given by the black box model (Crawley, 2012).

Proceeding, feature importance plots are produced for each observation. In addition to the importance, LIME plots also show the directionality of the effect. This allows users to fairly understand which features contributed most strongly towards the prediction and classification. As an illustration, six individuals’ LIME plots are presented in Fig. 9. The header information on the plot consists of the case number (corresponding to the observation number), class (predicted result of classification), probability of belonging to the class, and explanation of fit ($R^2$ as explained in Section 3.4.1). Furthermore, on the y-axis the five most important features are listed. The length of each bar (x-axis) indicates the weight of the feature. The green (positive) weights contribute to the prediction class and red (negative) weights contradict the classification of the individual.

For instance, consider case 74. We note that the patient has elevated levels of enzyme, protein and bilirubin, and are naturally predicted to have liver disease. Juxtaposed with case 92, also predicted to have liver disease, lower bilirubin levels are noticed. This indicated that even though the two subjects are predicted to have liver disease, they may not be suffering from the same ailment and may require different disease treatment. This validates the need for local surrogate models especially when global surrogate models are not easily obtained.

### 3.5 Model Stability

For any model to be considered trustworthy, it is imperative to explore its predictive performance on data that has not been used in its training. In the proceeding, two comparative strategies exploring the GBM’s ability to translate its learning from training dataset to testing dataset are presented. In both sections, the GBM’s predictive behavior is explored on enzyme, protein, and bilirubin levels. These levels correspond to eight features in the dataset. Using the normal levels presented in Section 2.1, these features are discretized in
low, normal, and high groups. Note that in certain cases not all groups had patients. In that event, naturally, no prediction was made.

3.5.1 Predictive probability plots

Figs. 10 and 11, depict the average probability for all groups created by the discretization. The average probabilities were obtained by simply taking the mean of the predicted probabilities of all individuals in that corresponding group. This idea is the same as the one introduced in the calculation in the creation of PDPs. The interest here is not necessarily in the levels of the probabilities but to compare whether the pattern discovered in the training set remain consistent within the training set.

For the training dataset, as noted before, high levels in most features of these biomarkers result in higher predicted probability of liver disease. Across the board, the patterns in the training dataset were repeated with the testing dataset. This includes the fact that low albumin levels indicate higher risk of liver disease. This analysis shows that the model is extremely faithful to its learned rules.
3.5.2 Classification accuracy plots

To further assess model stability, it was decided to see if the accuracy patterns in classification remain stable across training and testing. In Figs. 12 and 13, it can be seen that the patterns remain fairly consistent. The only exception being that there were perfect misclassifications in the testing dataset when high levels of proteins were observed. This is attributed to the fact that only 1 case fell into that category. Another clear observation is that classification accuracy rates fell across the board. This might bring into contention that the learned GBM may suffer from overfitting. It can be left as future work to further perform cross validation experiments to resolve this issue.

4 Conclusion and discussion

In the previous sections, a GBM was fit on the liver dataset. The GBM showed promising classification accuracy on both training and testing data. The challenge at this point was to discover the reasoning behind this success. Through the VIP, an importance ranking of features was obtained. Alkaline Phosphotase, an enzyme found in the liver, was determined to be the most important. Gender, on the other hand, was judged to be
Figure 13: Classification accuracy vs. effect plot for testing data

the least important. Even though this information provided some insights into the GBM’s decision making process, the inference was not extremely informative in terms of how the variables affect the response or if the importance was due to the marginal or interactive effects of the features.

To further understand the effects of features on the response, PDP and ICE plots were created for the alkaline phosphotase (highest on VIP), alanine aminotransferase (middle on VIP), and total bilirubin (second lowest on VIP). The general trend among all three was found to be that high levels increase the chances of liver disease. This observation matches intuition and the clinical understanding of liver functioning. It should be noted that the concept of PDP and ICE can be extended to multiple dimensions at the same time. Specifically, contour and surface plots can be made to describe the interactive effects of two variables. It is also possible to obtain projection plots for higher degree interactions, but these plots carry interpretability challenges of their own.

Next, the concept of surrogate models was introduced. For the particular GBM, a decision tree was found to be a good global surrogate. Further desire to understand the behavior of the model, prompted the use of attempted inference from local surrogate models. Using LIME plots, two patients predicted to have liver disease were presented with very different profiles. The key take away was that local interpretation can help ML users find helpful latent patterns. In the case of liver data, these local interpretations can show how different liver diseases have distinct patient profiles or provide insight into the acute or chronic nature of the disease.

It is worth mentioning that local feature ranking can also be done without the use of local surrogate models. Specifically, the concept of producing Shapley values (Shapley et al., 1988), obtained in cooperative game theory, can be applied. In this paradigm, the features are thought of as players and prediction process is imagined to be the game. The Shapley values rank the features’ by estimating their contributions in deviating the predicted value from the mean predicted value. Finally, in the context of ML interpretation, it is an extremely challenging task to determine the best interactions to study. Some typical methods include the use of VIP to pick features pairwise or running pairwise regressions. This problem is left as impetus for future work.

References


Hjerpe, A. (2016). Computing random forests variable importance measures (vim) on mixed numerical and categorical data.


Splicing of Multi-Scale Downscaler Air Quality Surfaces

Elizabeth Herman\textsuperscript{1}, Jeonghwa Lee\textsuperscript{2}, Kartik Lovekar\textsuperscript{3}, Fatemeh Norouzi\textsuperscript{4}, Dorcas Ofori-Boateng\textsuperscript{5}, Benazir Rowe\textsuperscript{6}, Jianhui Sun\textsuperscript{7}

Problem presenters: Elizabeth Mannshardt\textsuperscript{8}, Barron Henderson\textsuperscript{8} and Brett Gantt\textsuperscript{8}
Faculty Mentor: Brian Reich\textsuperscript{9}

Abstract

The United States Environmental Protection Agency (EPA) makes use of a suite of statistical data fusion techniques that combine ambient monitoring data with air quality model results to characterize pollutant concentrations for use in various policy assessments. Data fusion can overcome some of the spatial limitations of monitoring networks and benefit from the spatial and temporal coverage of air quality modeling. The current EPA air pollution prediction model uses a downscaler (DS) model to estimate pollutant concentrations on a national surface. Of interest are ways to improve the performance of the DS in certain areas of the continental US, particularly those with sparse monitor representation. The current methodology utilizes the same spatial range parameter across the continental United States. In order to capitalize on the strengths of spatial modeling capabilities, we consider predictions run on a regional scale. We do this by allowing for a flexible spatial range parameter. By allowing the spatial range parameter to be more localized, we may achieve more precise predictions. We use the output from the DS model run separately on the nine (9) National Oceanic and Atmospheric Administration (NOAA) climatic regions of the continental US with overlap. Findings show that regional and national DS air quality predictions differ significantly for regions where Air Quality Systems (AQS) density is low. We consider an assessment of the regional DS runs to the national runs, as well as a comparison to two (2) national monitoring station networks. Significant discrepancies are seen in the areas where the density of AQS stations is low, such as the Northwest, Northern Rockies, West and Southwest regions. A critical step is splicing the regions back together along the regional boundaries to create a smooth national air quality surface, that is applicable for regions with both low and high discrepancies. The method demonstrates visual smoothing of the two (2) overlapping surfaces and can be extended to handle higher dimensional overlap. Our smoothing parameter is based on the longitudinal distance to the edge of the DS regions. In our paper, we focus on providing smoothed estimates based on predictions from two regional DS runs and provide a comparison of three approaches to generate smoothed estimates for overlap regions.

1 Introduction

In 2016, the US EPA estimated that approximately 123 million people lived in counties where air quality concentrations were above the primary US EPA’s National Ambient Air Quality Standards (NAAQS) \cite{7}. Additionally, in 2018, the World Health Organization also reported an estimate of about 7 million premature deaths caused by ambient air pollution for the same timeline \cite{8}. One type of air pollution is called Particulate matter (PM). It is a complex mixture of extremely small particles and liquid droplets in the air. Research, over the years, has established linkage between exposure to Particulate Matter (PM) and health-related risks such as aggravated asthma, difficulty breathing, chronic bronchitis, decreased lung function, and premature death \cite{3, 4, 6}. PM is also associated with environmental impacts such as visibility impairment and disruption to the natural nutrient and chemical balance of the soil. Specifically, PM\textsubscript{2.5} describes fine inhalable particles, with 2.5 micrometers and smaller diameters. We will focus on PM\textsubscript{2.5} for our assessment.

\textsuperscript{1}Department of Mathematics, North Carolina State University
\textsuperscript{2}Department of Statistics and Probability, Michigan State University
\textsuperscript{3}Department of Statistics, The Ohio State University
\textsuperscript{4}Department of Mathematics, Morgan State University
\textsuperscript{5}Department of Mathematical Sciences, The University of Texas at Dallas
\textsuperscript{6}Department of Mathematical Sciences, University of Nevada Las Vegas
\textsuperscript{7}Department of Statistics, University of Virginia
\textsuperscript{8}US Environmental Protection Agency
\textsuperscript{9}Department of Statistics, North Carolina State University
EPA’s AQS provides measurements of air pollutant concentrations from a monitoring network throughout the United States. The EPA’s Community Multi-scale Air Quality (CMAQ) model includes simulations that combine current knowledge in atmospheric science and air quality modeling and multi-processor computing techniques to produce pollutant estimates on a grid surface. Each source of air quality information has different advantages and limitations including the spatial coverage of monitoring networks and the possible modeling bias and uncertainty associated with estimation of air quality. The US EPA currently utilizes a spatial prediction framework called “downscaling” [1, 2] to fuse these two data sources. The DS model fuses CMAQ output with AQS readings using a spatially weighted model that regresses monitoring data on a CMAQ derived regressor. The output of downscaling is estimates of air quality across the United States.

Figure 1: DS regional areas in the continental United States

DS is usually applied to a national scale, incorporating all data from the AQS sites located in the continental United States, yielding a smooth Air Quality surface map [1, 2]. One caveat is that for predictions in the Northwest region (see next section for detailed division of NOAA regions), the model also considers measurements from the AQS monitors in the Southeast region. In other words, the hypothesis is that while regions adjacent to each other are correlated, readings of PM$_{2.5}$ on the other side of the country should not be considered. By considering model runs on a regional scope, we could potentially see a reduction in computational time and increase in prediction accuracy. However, notice in Figure 1, the national surface is no longer smooth. Here we develop methodology to splice adjacent regions together, allowing for a continuous gradient between regions and smooth national surface. The approach considered here is to run the DS on a regional scale to allow for estimation of a regional scale parameter. Ultimately a smooth national surface is desired, so the regions are configured such that there are overlapping areas in-between the regions. That way, a specific grid point only includes information that is relevant to its corresponding region. The overlap allows for consideration of the transition between regions. An example of the overlap region is provided in Figure 2.

Our goal is to compare national and regional DS results and provide a methodology for smooth splicing for the points in overlap region. To splice regions, the methods we call horizontal mixed densities and horizontal mixed variables are introduced. By these methods, two different density functions or two different variables are combined according to the distance which will be defined in section 4.

In this paper we intend to build a methodological splicing framework/algorithm for regional estimates along regional boundaries to create smooth national surface. The rest of the paper is organized as follows: section
2 contains a description of the data, section 3 compares national versus regional DS results, section 4 outlines the methodology developed for splicing regional surfaces across a continuous gradient, and section 5 provides results and computational assessment of splicing. We summarize results and methodological discoveries in the conclusion.

2 Data

We are using regional and national DS predictions based on Community Multi-scale Air Quality (CMAQ) model and Air Quality System (AQS) measurements of air pollutant concentrations from a monitoring network throughout the United States for the first quarter in year 2014 as well as DS Model predictions generated using the two above mentioned data sources in the same time period. The DS model fuses pollution estimates from the CMAQ model with “ground-truth” data from AQS monitor stations to produce estimates of air quality across the United States. The two sources of information are valuable in different ways. The monitoring data are sparsely collected with some missing data, but provide direct measurement of the true pollutant value up to measurement error. AQS monitors are located on the ground and collect concentrations of pollutants (specifically PM$_{2.5}$), the pollutant we are interested in. While we do have actual readings of the concentration of PM$_{2.5}$, this data is usually collected in densely populated areas. CMAQ estimates grid averages with no missing values, but is subject to calibration error.

We use two types of DS output: National and Regional. The DS data contains location labels of the grid cells inside each region, the latitude and longitude of the grid cells center, as well as, mean and standard error of DS prediction for each cell. The regional DS data has the same structure as National DS data, where the main difference between the two is the amount of data being fed into the model as determined by the physical range of the NOAA region. NOAA Climate Zones [3], are a collection of states defined by NOAA in which the states have similar climates. Regional DS estimates were computed for the regions which roughly approximate two degrees outside of the corners of the rectangle encompassing NOAA climate region, more precisely:

<table>
<thead>
<tr>
<th>Region</th>
<th>Latitude Range</th>
<th>Longitude Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ohio Valley (OV)</td>
<td>(33,45)</td>
<td>(-97,-76)</td>
</tr>
<tr>
<td>North East (NE)</td>
<td>(36,50)</td>
<td>(-83,-65)</td>
</tr>
<tr>
<td>South East (SE)</td>
<td>(23,42)</td>
<td>(-88,-73)</td>
</tr>
<tr>
<td>Upper Midwest (UM)</td>
<td>(39,51)</td>
<td>(-99,-81)</td>
</tr>
<tr>
<td>South (S)</td>
<td>(24,42)</td>
<td>(-108,-86)</td>
</tr>
<tr>
<td>North Rockies (NR)</td>
<td>(38,51)</td>
<td>(-118,-93)</td>
</tr>
<tr>
<td>South West (SW)</td>
<td>(29,44)</td>
<td>(-116,-100)</td>
</tr>
<tr>
<td>North West</td>
<td>(40,51)</td>
<td>(-127,-109)</td>
</tr>
<tr>
<td>West</td>
<td>(31,44)</td>
<td>(-127,-112)</td>
</tr>
</tbody>
</table>
The next component of the data is obtained from DS model (DS) ran on regional surfaces. This model uses a grid of continental United States divided into 12km by 12km pieces. It is essentially a linear regression that incorporates data gathered from the AQS sites and other environmental parameters local to the grid cell. The output of DS is a mean estimate of the pollution and its standard error for each grid cell.

Source: NOAA National Climatic Data Center [5]

The DS model used here is a spatially varying weights model that regresses monitoring data on a derived regressor obtained by smoothing the entire CMAQ output with weights that vary both spatially and temporally. This adaptive smoothing of CMAQ was used to achieve stronger association with the monitoring data by taking advantage of useful spatial information in neighboring CMAQ cells surrounding the cell where the monitoring data occurs. Let $Y(s, t)$ be the observed monitoring station data (AQS) at location $s$ on day $t$, and let $X(B_k, t)$ be the observed numerical model output (CMAQ) for grid block $B_k$ on day $t$. The smoothed DS with spatially varying weights model assumes the prevalent model

$$Y(s, t) = \beta_{0,t} + \beta_0(s, t) + \beta_{1,t}\tilde{X}(s, t) + \epsilon(s, t)$$

(1)
where $\beta_0(s,t)$ is a mean zero Gaussian process with covariance function specified in [2], and $\epsilon(s,t)$ is a white noise Gaussian process with variance $\tau^2$. Furthermore, $\tilde{X}(s,t)$ is monitor-specific weighted average of values from grid cells containing $s$ and some neighbors with the weights specified in [1, 2].

Intuitively, the SVW model assumes independence across time while the spatial dependence is governed by a Gaussian process with exponential covariance function. Markov chain Monte Carlo (MCMC) method is used to obtain a sample from the posterior distribution with vague or conjugate priors given to all model parameters. Draws are obtained using a Gibbs sampler where the Metropolis-Hastings accept-reject algorithm is used where necessary. For each model parameter, DS numerical model will output mean and standard error estimates, which denote the mean and standard error of a large number of MCMC samples after burn-in, respectively.

Another piece of data comes from Inter-agency Monitoring of Protected Visual Environments (IMPROVE) stations. Figure 4 shows the location of the IMPROVE monitors located throughout the United States. This data is an independent source, meaning we did not use IMPROVE to inform our DS runs. We will use this data to provide insight as we evaluate the DS on both the regional and national scales. As with our other data sources, we will focus our analysis on the first quarter of 2014. IMPROVE stations measure the concentrations of the pollutant PM$_{2.5}$ in the air.

Unlike the AQS sites, these sites are located in areas where one expects high air quality. For instance, the IMPROVE sites can be located in National Parks or other rural areas. IMPROVE data-set contains variables such as quarterly averages of the PM$_{2.5}$ value, AQS$_{id}$, latitude and longitude.

### 3 Downscaler Comparative Assessment

Considering DS outputs such as mean and standard error, we noticed that the standard errors were high in the Northern Rockies region. Because mean estimates were constructed using MCMC, we decided to take a closer look at trace plots. The convergence is usually guaranteed by a sufficiently long run of the algorithm. We do some basic diagnostics to test whether the MCMC sample mean follows theoretical posterior distribution. For model parameter $\beta_{0,t}$ and $\beta_{1,t}$, vague conjugate prior distributions are assumed. Additionally, $\beta_{0,t}$ and $\beta_{1,t}$ are assumed to be independent in time and their posterior distribution should be Gaussian. We provide the following trace plots and QQ plots Figure 5 to justify the convergence of MCMC runs from the first quarter in Northern Rockies and Plains region.
Figure 5: Trace plots and QQ plots for model parameters $\beta_{0,t}$ and $\beta_{1,t}$ of the MCMC runs, from the first quarter data in Northern Rockies and Plains region.

Figure 6: Difference between Regional and National DS calculated using (2)

Although the air quality DS model shows relatively satisfactory agreement with AQS and CMAQ in general, the AQS monitoring stations which inform the DS model are concentrated in urban areas. This can leave rural areas with less accurate estimates. Additionally, occasional pollution events may be correctly reported by a CMAQ model, but missed by the AQS monitor, leading to an under-estimation of pollution in an area in the DS model. The reverse is also possible - a pollution event may be modeled correctly by CMAQ, but missed by the closest AQS monitor because pollutants are only measured at that AQS site rather than areas between sites. This could result in over-estimation of pollution at an AQS site, but under-estimation in the affected areas. A third scenario is a pollution event that is missed by both the CMAQ model...
and AQS sites, resulting in under-estimation of pollution in the downwind areas. EPA is interested in the identification of areas with discrepancies between the point-source monitoring network (AQS and IMPROVE) and DS model.

In order to judge whether DS adequately describes “ground-truth” monitoring data, we calculated their relative discrepancies. Relative discrepancy is determined by comparing the model estimates with AQS or IMPROVE observations. We define relative discrepancy through a Fractional Bias (FB) metric, specifically in form of 2.

\[ FB(site_s, DS_k) = \frac{site_s - DS_k}{(site_s + DS_k)/2}, \]

where \(site_s\) is the air pollutant readings from AQS or IMPROVE station at location \(s\), and \(DS_k\) is the DS output from the \(k\)-th grid which includes the AQS or IMPROVE monitor \(s\).

EPA is interested in the identification of areas with discrepancies between the point-source monitoring network (AQS and IMPROVE network) and DS model.

In Figure 6, we provide the comparison between mean predictions from National DS model and Regional DS model, respectively, while in Figure 7, the comparison between standard errors of prediction from National DS model and Regional DS model is shown, respectively. The distinctiveness of NW region, NR region and W region is in line with the conclusion from Figs. 8 and 9. The standard error intuitively characterizes the abundance of information we could get at a certain location. Running DS model in National/Regional scale has a different magnitude of model uncertainty, which is a noteworthy pattern that awaits further interpretation.

We present how DS model describes AQS measurements in Figure 8. DS outputs from the middle figure come from running the model on a national scale and the other nine figures describe DS outputs by running the model in nine regions, respectively. From the figure, we can draw the conclusion that the application of the DS to regional domains (e.g., NOAA climate regions) does not in general improve the DS predictions, and even jeopardizes DS performance in certain regions on the West Coast. In the NW region, National DS and Regional DS tend to have opposite conclusions. Specifically, most mis-predictions of Regional DS are mild over-estimations of AQS measurements while, on the other hand, National DS tends to under-estimates NW AQS stations.
In Figure 9, we show the discrepancy between National/Regional DS and IMPROVE station measurements. IMPROVE stations mainly focus on rural areas and are much more sparsely distributed compared to the AQS monitoring system. Compared to 8, the DS model performs worse in predicting rural areas (Note that the color scale in 9 is more extreme than 8). Most mis-predictions occur in regions on the West Coast, and DS model seems to exhibit a significant over-estimation of IMPROVE measurements.

In Figure 10, we compare the ratio of discrepancy between Regional DS and AQS to discrepancy between National DS and AQS. Note that we keep the sign for both discrepancies to reflect possible sign changes. From the figure, we further strengthen the previous conclusion, which indicates consistency between DS and AQS overall, but that there exist several AQS stations with extreme values. The extreme cases do not exhibit distinctive spatial characteristics and are scattered within each region. We provide similar ratio plots with IMPROVE sites in Figure 11, which indicates comparable discrepancy between IMPROVE measurements and Regional/National DS as well. No noticeable spatial pattern could be detected from this ratio plot.
4 Splicing Methodology

In this section, we consider the situation where we have the overlap between two regions, $R_1, R_2$. The generalization of the methods in this section for the case where two regions overlap primarily along latitudes is straightforward. For each monitoring site, we have DS predicted means and standard deviations from two regions. More specifically, for a monitoring site $s$ in the overlap, $\{\hat{\mu}_i(s), \hat{\sigma}_i(s)\}$ are the predicted DS mean and predicted DS standard deviation at $s$ from DS region $i$. We now consider the following methods.

4.1 Horizontal Mixed Density Method (HMD)

Assume that for site $s$ the distribution of DS from region $i$ is normal with mean $\hat{\mu}_i(s)$ and standard deviation $\hat{\sigma}_i(s), i = 1, 2$. To merge the information from two different regions, we use model averaging where the
The probability density function of PM$_{2.5}$ at $s$ is a weighted average of probability density functions (pdf) from regions 1 and 2.

$$f_s = w_1(s)f_{1,s} + w_2(s)f_{2,s}$$

where $f_i$ is a normal density function with $\mu = \hat{\mu}_i(s)$, $\sigma = \hat{\sigma}_i(s)$. We set the weight as a function of a distance between $s$ and the outer boundary in the overlap, because as the site lies closer to the outer line, it is less affected by the information from the region. In other words, $w_i(s)$ is proportional to the distance from the site $s$ to the outer boundary of region $i$ in the overlap, because the larger this distance is, the closer this point is to the inner area of region $i$, and the less it is influenced by the adjacent region.

**Figure 12: Distance from a site to the boundary**

Figure 12 shows how the distance is calculated when two side-by-side regions are overlapping. If the intersection of two regions occurs vertically, then the distances are obtained by vertical distances of $s$ to the boundaries. The parameter $\phi$ is also introduced to adjust the effect of distance on weights. More specifically, we have the following weight for the density function from region $i$ at point $s$, $w_{i,s}$:

$$w_i(s) = \frac{e^{-\phi d(s,i)}}{e^{-\phi d(s,1)} + e^{-\phi d(s,2)}}$$  

where $d(s,i)$ is the distance of point $s$ to the outer boundary of region $i$ as explained above. This implies that the larger $\phi$ is, the bigger influence the distance has on the weight, making the distribution from each region have a stronger effect on nearby areas. The smaller $\phi$ is, the more two distributions are blended in the overlap. In the extreme case when $\phi$ is zero, the weights are simply evenly distributed among the densities, $w_i(s) = .5$ for $i = 1, 2$. If $\phi = \infty$ the probability density function (pdf) $f$ at $s$ simply becomes either $f_{1,s}$ or $f_{2,s}$ depending on which boundary is closer to $s$, and the weight has no effect on averaging. This is shown in Figure 13.
Now we have the following likelihood function for AQS $y_j$ at $s_j \in R1 \cap R2, j = 1,..,n.$

$$L(y_1,..,y_n; \phi) = \prod_{j=1}^{n} f(y_j) = \prod_{j=1}^{n} w_1(s) f_{1,s}(y_j) + w_2(s) f_{2,s}(y_j)$$

(4)

$$= \prod_{j=1}^{n} e^{-\phi d(s_j,1)} f_{1,s_j}(y_j) + e^{-\phi d(s_j,2)} f_{2,s_j}(y_j)$$

(5)

Using our method, $\hat{\phi}$ is obtained, and the new predicted mean at the site $s$ in the overlap becomes a weighted mean:

$$\hat{\mu}(s) = \int y f(y) dy = \int \hat{w}_1(s) f_{1,s}(y) + \hat{w}_2(s) f_{2,s}(y)$$

(6)

$$= \int ye^{-\hat{\phi}d(s,1)} f_{1,s}(y) + ye^{-\hat{\phi}d(s,2)} f_{2,s}(y) dy$$

(7)

$$= e^{-\hat{\phi}d(s,1)} \hat{\mu}_1(s) + e^{-\hat{\phi}d(s,2)} \hat{\mu}_2(s)$$

(8)

$$= \hat{w}_1(s) \hat{\mu}_1(s) + \hat{w}_2(s) \hat{\mu}_2(s)$$

(9)

where $\hat{w}_i(s) = e^{-\hat{\phi}d(s,i)}/(e^{-\hat{\phi}d(s,1)} + e^{-\hat{\phi}d(s,2)})$, $i = 1,2$.

### 4.2 Horizontal Mixed Variable Method (HMV)

Instead of averaging density functions, we now average two random variables. Assuming that we have two independent normal variables at site $s$ in the overlap, each from DS region $i = 1,2$. In other words, for a site $s$ the information from DS region $i$ is assumed to be represented by $X_i(s) \sim N(\hat{\mu}_i(s), \hat{\sigma}_i(s)), i = 1,2$. Our new combined information at site $s$ is then expressed as follows:

$$X_s = w_1(s)X_1(s) + w_2(s)X_2(s)$$

where the weight $w_i(s)$ is defined as in (3). With the assumption that $X_1(s)$ and $X_2(s)$ are independent normal variables, $X$ becomes normal variable with mean $\mu_\phi = w_1(s)\hat{\mu}_1(s) + w_2(s)\hat{\mu}_2(s)$ and variance $\sigma_\phi^2 = w_1(s)^2\hat{\sigma}_1(s)^2 + w_2(s)^2\hat{\sigma}_2(s)^2$. Thus, the resulting likelihood function is as follows:

$$L(y_1,..,y_n; \phi) = \prod_{j=1}^{n} f(y_j; \phi)$$

where $f(y; \phi)$ is Normal density function with mean $\mu_\phi$ and variance $\sigma_\phi^2$. 

The Maximum Likelihood Estimate (MLE) of $\phi$, $\hat{\phi}$, is obtained and used to compute spliced estimates at the site $s$ in the overlap as $\hat{\mu}(s) = \hat{w}_1(s)\hat{\mu}_1(s) + \hat{w}_2(s)\hat{\mu}_2(s)$, where $\hat{w}_i(s) = e^{-\hat{\phi}d(s,i)}/(e^{-\hat{\phi}d(s,1)} + e^{-\hat{\phi}d(s,2)})$.

### 4.3 Adaptive Horizontal Mixed Variable Method (AHMV)

The estimates given by HMV show a sharp divide at the edges of the intersection region. This is expected since the estimates do not take into account any values outside the intersection. To adjust for distance, we vary $\phi$ linearly with distance from the center of the intersection. By doing this, the degree of influence of distance on weight reduces as we approach edges of the intersection, thereby yielding smoother surfaces at intersection edges.

More specifically, we have the following function for $\phi$:

$$\phi(d(s,c)) = \beta_0 + \beta_1 d(s,c) \tag{10}$$

where $d(s,c)$ is the horizontal distance of $s$ to the vertical center line, i.e., $d(s,c) = |s_{lon} - c_{lon}|$ where $s_{lon}, c_{lon}$ are longitude of point $s$ and center line $c$ respectively. This is explained in Figure 14.

![Figure 14: Distance from a site to the center](image)

Figure 15 compares the weight functions when $\phi$ is constant and when $\phi$ is a linear function as in (9). The black and blue graphs are weight functions with constant $\phi$ values of 2 and 4, respectively. The red graph is the weight function when $\phi$ is the linear function as in (9) with $\beta_0 = .3, \beta_1 = .5$. It is seen that in the red graph weights slowly vary in the center and then rapidly increase or decrease moving away from the middle, whereas in the black and blue graphs the weights are either rapidly changing or slowly changing as one moves away from the center. Therefore, by using this function for $\phi$, we can make the surface in the middle of the overlap region blend more, while blending less at the edge of the overlap, and give more weight to the mean that lies deeper inside the regional surface, compared to constant $\phi$ as in HMV.
Figure 15: Weight functions for constant phi and function phi. (a) The black graph is the weight function when phi is the constant 2. (b) The blue graph is the weight function when phi is the constant 4. (c) The red graph is the weight function when phi is the linear function with $\beta_0 = .3, \beta_1 = .5$

5 Results

5.1 Smooth surfaces from three methods

Since the regional DS shows that the most discrepancy in the area is between NW and NR, we restrict ourselves to these regions, applying the three methods in the previous section on the overlap of NW and NR. The following results were obtained from each method:

Table 2 shows MLE of parameters in each method. For HMD and HMV, we have one parameter $\phi$ and for M3 we have $b_0, b_1$ in weight function. The MLE of $\phi$ is 1.450, 0.222 for HMD and HMV respectively implying that HMV smooths the data in the overlap more than HMD since it has larger $\hat{\phi}$. For M3, $\hat{b}_0 = 0.157, \hat{b}_1 = 0.029$. We apply these estimates to our models and obtain figures 16 and 18.

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMD</td>
<td>$\phi$</td>
<td>1.450</td>
</tr>
<tr>
<td>HMV</td>
<td>$\phi$</td>
<td>0.222</td>
</tr>
<tr>
<td>AHMV</td>
<td>$\beta_0, \beta_1$</td>
<td>(0.157, 0.029)</td>
</tr>
</tbody>
</table>

By looking at Figure 16 and Figure 16 b, we can see that HMV shows smoother surface than HMD. The cause for the difference is explained by the respective functions in Figure 17.

The black and red graphs in Figure 17 are drawn with two normal densities, $N(0, 1), N(3, 1)$ with weight 0.3, 0.7 from HMD and HMV, respectively. The blue graph is drawn with the same densities but with weights 0.1, 0.9 respectively with HMD. The means of the black and red density are the same, 0.21. Density function from HMV is normal density, symmetric, and has a peak at the mean. Density function from HMD is skewed to the left which has smaller weight, and has peak at 3. The peak in HMD is 3 when the second weight is larger than 0.5, highest when the weight is 0.5, and getting lowered making a more skewed distribution as the weight is decreased. This can be seen by comparing the blue and black graphs. It is that property that makes
MLE of $\phi$ in HMD larger than that of HMV, making the values in the overlap stay closer to the mean from closer areas, thus producing a less smooth surface.

With HMV and AHMV, we can see that the AHMV attains better smoothing near the outer boundary of NW region, since we saw the sharp edge on the outer boundary of NW on HMV. This sharpness becomes blurred in AHMV’s image, and the model smoothly blends two predictions together. This is expected as we make $\phi$ a linear function of distance from the center line. There is a sharp edge at the bottom of the overlap which can be eliminated using similar methods run by incorporating latitudinal distances, which is left for future work.

**Remark.** There are many other ways to splice two regions to make a smooth surface. If one uses a different form of $\phi$, and/or different distance measures, different smooth surfaces might be obtained. In this study we only use either longitude of sites or latitude of sites for splicing two regions to measure the distance, but one can also use both latitude and longitude to achieve smoothness in both directions, i.e., horizontally and vertically. Also $\phi$ can be a quadratic function of distance, perhaps yielding a smoother surface. We wish to mention that there could be a case where different smooth surfaces produced by different methods are all very smooth and, at the same time, they all fit the data well. In this case, splicing surfaces would be subjective. Nonetheless our method is intuitive and a useful tool when two different estimates on the same points need to be combined.

We choose model 3 to run diagnostics since it takes into account more information than the other two methods and looks smoother upon a visual inspection. To validate the method, we run it for the regions where regional DS predictions differ most from national estimates, the North West and Northern Rockies. The overlap of those 2 regions contains 5,978 grid cells and 45 AQS stations. We use the 45 AQS observations and corresponding DS predictions from regional DS estimates to fit the model using method 3. We use the R functions `mle` and `ggplot`.

We compute Mean Square Error (MSE) for estimates obtained from the 3 models compared to the AQS and IMPROVE sites.
Figure 17: Density functions from different models: (a) The black graph is drawn with two normal densities, $N(0, 1), N(3, 1)$ with weight 0.3, 0.7 from method 1. (b) The red graph is drawn with the same densities and same weights but with method 2. (c) The blue graph is drawn with the same densities but with weights 0.1, 0.9 respectively with method 1.

Figure 18: AHMV applied on the intersection of NR and NW

Compared to AQS data, the MSE is not very different between two models and is around 2.8. Compared to IMPROVE sites, the MSE is around 45. A possible implication is that DS is not accurate at rural sites, where IMPROVE should possibly be favoured. On the other hand, this is expected since we use AQS data to fit our model and this data has also been used by DS. Also, there are only 14 corresponding IMPROVE sites in the North West - Northern Rockies intersection, which may not be comprehensive.

Next, we perform a validation procedure by subsetting 80% of the AQS data points as the training set and the other 20% of the points as the test set. We compare the Mean Squared Error (MSE) of our smoothed
predictions using full and subset of the available AQS data. The MSE for the estimates obtained using 80% of the data is 4.632, while the MSE for the estimates obtained using all 100% of the AQS stations in the overlap of NR and NW is 4.64. The MSE of both estimates are very close, signaling that that our estimates are likely not very sensitive to the differences in data.

### 6 Summary and Future Work

Each of the three methods presented achieves different degrees of smoothness. Model 3 yields a surface based on the two regional DS model predictions and distance parameters estimated from the AQS data. For any two overlapping DS regions, the method produces a smooth surface while adjusting for both high and low discrepancies between them.

While our approach achieves a smooth surface, we only account for longitudinal distances. Taking latitude into account should improve the smoothing process and get rid of the sharp horizontal edges that are currently present in all 3 methods suggested. Since maps are inherently 2-dimensional, the first extension of the existing approach would be addition of the second dimension (latitudinal distance) to influence blending coefficient. Another extension would be considering more than double overlap regions and combining 3 or 4 predictions simultaneously. Since DS regions are rectangular, the corners could yield a group of rectangular areas with single, double, triple and quadruple overlaps. Next, inclusion of IMPROVE sites in the estimation procedure may provide additional precision of the resulting estimates. This is especially important for regions like Southwest, where AQS stations are mostly located on the edges of the region and the density of AQS stations is relatively low.

Our method could handle regions with both high and low density of AQS stations. It is motivated by providing a smooth surface of the map while optimizing for minimal discrepancy between the predictions and AQS readings. This implies that the method provides a way to optimally mask the discrepancies between regional and national DS predictions.

A high discrepancy in the Regional and National DS predictions signal that the DS model could be inherently local in nature and is highly dependent on the perspective, i.e. chosen size and location of the region as well as the density and topology of the AQS stations in that region. It is not clear yet to what extent all those variables affect the accuracy of the DS model and how they should be accounted, if deemed necessary, when choosing the shape and size of the regions on which the DS model is run.
References


SEMI-ANALYTICAL BRDF-BASED QUANTIFICATION OF LIGHT REFLECTION

MICHAEL BYRNE\textsuperscript{1}, FATOUmATA SANOGO\textsuperscript{2}, PAI SONG\textsuperscript{3}, KEVIN TSAI\textsuperscript{4}, HANG YANG\textsuperscript{5}, AND LI ZHU\textsuperscript{6}

Problem Presenter: John Peach (MIT Lincoln Laboratory)
Faculty Mentor: Alen Alexanderian (NCSU)

CONTENTS
1. Abstract 1
2. Introduction 1
3. Theoretical Background and Tools 2
  3.1. Reflectance Distribution Functions and Optical Cross Section 2
  3.2. Rvachev Functions 4
  3.3. OpenSCAD and Chebfun 6
4. Technical Challenges 9
5. Results and Discussions 9
  5.1. Analytical Solutions for Sphere 9
  5.2. Numerical Calculation of Scattering Function for a Sphere: finite element approach 11
  5.3. Numerical Solutions for Sphere & Application to Ellipsoid: analytic approach 12
  5.4. The Multipath Problem 15
6. Conclusion 18
7. Acknowledgment 19
References 19

1. Abstract

We build a mathematical model of the properties of an object that faithfully captures the relevant optical phenomena for satellite tracking and prediction of reflective response. We develop analytical models to replace facetized CAD models with continuous functions by providing a more accurate, faster and less complex calculation of the Optical Cross Section (OCS) function. Also, we propose an extension of the discrete multipath problem on the continuous domain. Our method shows the potential to reproduce the accuracy observed on simple analytic objects to much more complex objects using constructive solid geometry via Rvachev functions.

\textsuperscript{1}Arizona State University, mjbyrne2@asu.edu.
\textsuperscript{2}University of Alabama Birmingham, sanogof1@uab.edu.
\textsuperscript{3}Old Dominion University, psong@odu.edu.
\textsuperscript{4}University of California Riverside, ytsai003@ucr.edu.
\textsuperscript{5}Rice University, hang.yang@rice.edu.
\textsuperscript{6}University of Nevada Las Vegas, zhul5@unlv.nevada.edu.
2. Introduction

Through observation of light reflected from an object, various properties of the object may be estimated such as object composition and relative orientation. Light reflection is a well-studied phenomenon; much is known regarding the relationships between wavelength, angles of incidence, and surface material. One way to model the reflectance of an arbitrary object is to form a triangular mesh and from each patch use incident, reflected, and surface normal angles to determine the amount of local reflectance. A distribution function is used to model the reflectance, which is a function of the angles described previously. Since the reflectance of light depends upon the direction of both incident light and the viewing direction, a model of light reflection often comes in the form of a bi-static reflectance distribution functions (BRDF). Distribution functions for light are also specific to the wavelength of light, though in this work we consider a fixed wavelength.

Once reflectance is calculated for each patch, the reflectances can be summed to determine the overall reflectance of the object. The sum of the reflectance is known as the total optical cross section of the object. If the object being considered is simple, such as a sphere, ellipse, or cylinder, optical cross section can be determined analytically. However, more complex objects often require the triangular patch approximation described above. For highly specular surfaces, the triangular patch approximation may result in an undesirable “disco ball” representation of the object. Increasing the number of mesh points (and thus increasing the number of triangular patches) is one possible remedy, though this approach can become expensive. As such, the primary objective of this project is to investigate and develop extensions of analytical modelling of light reflection for more complicated objects.

3. Theoretical Background and Tools

3.1. Reflectance Distribution Functions and Optical Cross Section. When light is projected onto opaque materials, the majority of incident light is transformed into reflected light and absorbed light. As a result, when an observer views an illuminated surface, what is seen is reflected light, i.e. the light that is reflected towards the observer from all visible surface regions. A reflectance distribution function (RDF) describes how much light is reflected when light makes contact with a certain material at a certain point. Informally, RDF determines the detectable “light density” after reflection.

In general, the degree to which light is reflected depends on the viewer and light position relative to the surface normal and tangent. Generally speaking, RDF depends on the wavelength of the light that is projected and is a function of incoming light direction (represented by the incident vector $\mathbf{I}$) and viewing direction (represented by observing vector $\mathbf{V}$) relative to a local orientation at the light interaction point. However, since this is not the concern of our project, we ignore for now the effect of the wave length. When the incidence angle and the viewing angle coincide, i.e. $\mathbf{I} = \mathbf{V}$, the result is called Mono-static Reflectance Distribution Function (MRDF). More generally, when the two angles are different it is called Bi-directional Reflectance Distribution Function (BRDF).

Theoretically, BRDF is material-specific. Therefore, getting an accurate functional relation requires careful measurements through lab experiments. However, there are models that provide substitutes for easy usage. The most famous examples are Cook–Torrance model [3], Ward’s model [6]
Figure 1. The MRDFs from the simplified Blinn–Phong model for varying values of $\alpha$. The aspect angle is the angle between $N$ and $I = V$. Note as the value of $\alpha$ becomes large, the modeled specularity of the surface increases; in other words, there is less diffusion of light as it is being reflected off of the surface.

and the Blinn–Phong model [2]. In our work, we use the Blinn–Phong model,

$$\text{BRDF}(I, V; N) \approx \langle H, N \rangle^\alpha,$$

where $N$ is the surface (outward) normal and

$$H := \frac{I + V}{\|I + V\|}.$$

is the so-called half-way vector. The parameter $\alpha \geq 0$ depends on the light reflectivity of the material used to cover the surface (large reflectivity corresponds to large $\alpha$). Intuitively speaking, the more mirror-like an object is, the sharper its specular peak will be (corresponding to an RDF that decays faster).

A special case of the Blinn–Phong model gives rise to a handy formula for MRDF. Taking $I = V$ and letting the angle between $I$ and $N$ be $\theta$, we get

$$\text{MRDF}(I; N) = (\cos \theta)^\alpha$$

Optical cross section, which we denote by $\sigma$, describes the maximum amount of optical flux that is reflected by an object. It can be computed accordingly by the geometry of the surface of the object and reflectivity of a particular wavelength of an object. Here, we will again ignore wavelength and use the representation

$$\sigma = \int_S \rho \, dA$$

where $\rho$ is the optical density and is given by an RDF, $S$ denotes the surface. In our project, we use the representation

$$\sigma = \int_S \text{BRDF}(I, V; N) \cos \gamma \cos \beta \, dA$$
with $\gamma$ denoting the relative angle between incident light angle and surface normal, $\beta$ denoting the relative angle between viewing angle and the surface normal. The cosine functions are used to account for the projected light along the surface normal. The following are some common Jacobians for projecting surfaces in parametric forms onto $\mathbb{R}^n$ (i.e. $dA = |J|dx_1...dx_n$)

Sphere : $(r \cos \theta \sin \varphi, r \sin \theta \sin \varphi, r \cos \varphi)$, $|J| = r^2 \sin \varphi$

Ellipsoid : $(a \cos \theta \sin \varphi, b \sin \theta \sin \varphi, \cos \varphi)$, $|J| = \sin \varphi \sqrt{\sin \varphi^2 (a^2 \sin \theta^2 + b^2 \cos \theta^2) + b^2 c^2 \cos \varphi^2}$

Cone : $(r \cos \theta, rz \sin \theta, r)$, $|J| = rz \sqrt{1 + r^2}$

Cylinder : $(r \cos \theta, r \sin \theta, z)$, $|J| = r$

Torus : $((c + a \cos u) \cos v, (c + a \cos u) \sin v, a \sin v)$, $|J| = a(c + a \cos v)$

3.2. Rvachev Functions. Rvachev functions (R-functions) describe complex geometric objects with a single inequality or equation. In the 17th century, Decartes suggested the idea of relating geometric objects (e.g. lines, circles and bodies) to analytical objects (e.g. sets, functions and equations). Since then, methods to study geometric properties based on functional descriptions have been developed systematically. This is also known as the direct problem of analytic geometry. As opposed to the direct problem, people also considered the inverse problem: given certain geometric objects equipped with some desired properties, find an analytical representation to such objects. Figure 3 illustrates the union, intersection and difference of a disk and ball.

For simple geometric objects, the inverse problem is not difficult. Yet for more complicated ones, especially when multiple forms and shapes are combined, analytical descriptions become less clean and usually involve logical operations among regions defined by inequalities. R-functions help resolve this issue.

Formally, an R-function is a function whose sign does not depend on the magnitude of its arguments. The following are some simple examples of R-functions:
(a) \( f(x,y) = 1 \)
(b) \( f(x,y,z) = x^2 + y^2 + z^2 + 1 \)
(c) \( f(x,y) = xy \)
(d) \( f(x,y) = \min(x,y) \)
(e) \( f(x,y) = x + y - \sqrt{x^2 + y^2} \)

The definition of an R-function is now introduced.

**Definition 1.** A function \( f(x_1, ..., x_n) : \mathbb{R}^n \to \mathbb{R} \) is an R-function if and only if there exists a Boolean function \( F(x_1, ..., x_n) : \{0,1\}^n \to \{0,1\} \) such that

\[
S_2(f(x_1, ..., x_n)) = F(S_2(x_1), ..., S_2(x_n))
\]

where \( S_2(x) \) is defined as

\[
S_2(x) = \begin{cases} 
1, & x > 0; \\
0, & x < 0 
\end{cases}
\]

Such function \( F \) is called the Boolean companion function of \( f \).

With this definition, we can easily see that (a), (c) and (d) are indeed R-functions with Boolean companions 1, \( \leftrightarrow \) and \( \land \) respectively.

Upon combining elementary “primitives” (sphere, cylinder, etc.) into composite objects with Boolean operations, we can construct R-functions that allows us to operate directly on the formulae/functions of those primitives and obtain a single analytical expression for the composite objects. The set of R-functions used to define these operations has a natural correspondence to the Boolean functions. For the complete system of Boolean functions \( \{0, \neg, \land, \lor\} \), consider the set of R-functions \( \{-1, -x, x_1 \land \alpha x_2, x_1 \lor \alpha x_2\} \) where \( \land \alpha, \lor \alpha \) are defined by

\[
R_\alpha(x_1, x_2) = \frac{1}{1 + \alpha} \left( x_1 + x_2 \pm \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2} \right),
\]
respectively.

In the above definition, \( \alpha \) is a symmetric function in general with range \((-1, 1]\). In practice, we can simply choose \( a \) to be constants and the resulting R-functions are equivalent (in the same branches) in the sense that their companion Boolean functions are exactly the same, i.e. \( \land, \lor \).

There are other systems of R-functions that could also be used to complete the task. For example

\[
R^m_{\alpha}(x_1, x_2) : \frac{1}{1 + \alpha} (x_1 \land \alpha, \lor \alpha x_2)(x_1^2 + x_2^2)^m/2
\]

One of the significant differences between \( R_{\alpha} \) and \( R^m_{\alpha} \) is that while \( R_{\alpha} \) is not differentiable along the diagonal \( x_1 = x_2 \), \( R^m_{\alpha} \) is analytic on the whole plane except at 0 where it is only \( m \)-times differentiable. For the purpose of this work, we shall not dwell on differentiability of the R-functions and will use \( R_{\alpha} \). See [5] for more information. The following example shows the merits of using R-functions. Consider defining the checkerboard in a single equation with the following constituents (primitives)

\[
D_1 = \{ \sin(\pi x_1) \geq 0 \}, \quad D_2 = \{ \sin(\pi x_2) \geq 0 \}, \quad D_3 = \{ 32 - x^2 - y^2 - |x^2 - y^2| \geq 0 \}
\]

Graphically, \( D_1 \) generates vertical stripes, \( D_2 \) generates horizontal stripes and \( D_3 \) defines region enclosed by a rectangular boundary.

The checkerboard will then be represented by the following Boolean equation

\[
(D_1 \lor D_2) \land (\overline{D_1} \lor \overline{D_2}) \land D_3 = (D_1 \leftrightarrow D_2) \land D_3
\]

Recall that “\( \leftrightarrow \)” is the Boolean companion of \( f(x, y) = xy \). Then the formula for the checkerboard can be written as \( (\sin(\pi x_1) \sin(\pi x_2)) \land \alpha (32 - x^2 - y^2 - |x^2 - y^2|) \geq 0 \). This can be further simplified into one equation by noticing that the region defined by \( f(x_1, ..., x_2) \geq 0 \) can be written as \( f - |f| = 0 \).

3.3. OpenSCAD and Chebfun.

3.3.1. OpenSCAD. OpenSCAD is a free software application for creating solid 3D CAD (computer-aided design) objects. It is a script-only based modeller that uses its own description language. The program does constructive solid geometry (CSG) using specified geometric primitives (such as spheres, boxes, cylinders, etc.) and defines how they are modified and combined (for instance by intersection, difference, envelope combination and Minkowski sums) to render a more complex 3D model. Once the model is constructed in OpenSCAD, the finite element mesh may be created using meshing tools such as Gmsh. Figure 4 demonstrates the example of a screwdriver handle constructed with OpenSCAD followed by a triangular mesh generated on the surface of the object.

Numerically, we can handle even more complex objects. However, to accurately represent the surface many thousands to millions of facets are sometimes required and a trade-off must be made.
between accuracy of the surface represented and speed of calculation. For example, Table 1 shows
the numerical calculation of the OCS for a unit sphere. Note that to attain a relative error of about
$10^{-4}$ we need 20,480 elements.

3.3.2. Chebyshev interpolation analysis.
We wish to develop a method that improves the accuracy for much more complex objects. We
consider representing the surface in functional form using Chebyshev polynomials and built up
through Rvachev functions. Using a Chebyshev polynomial representation it is possible to model
the surface of an object to within machine precision given sufficiently many Chebyshev points.

Let us recall some basics regarding Lagrange interpolation using Chebyshev points for univariate
functions. It is common to choose the interpolation nodes to be evenly spaced. In many cases, the
data to be interpolated are available only in some certain forms, for instance, when the data consist
of instrument readings separated by a constant time interval. In other cases, for example, the sine
function, we are free to choose the interpolation nodes. It turns out that the choice of interpolation
points can have a significant effect on the interpolation error. Chebyshev interpolation refers to a
particular optimal way of spacing the points.

Consider the case with a single variable. We start with a function $y = f(x)$ and take data
points from it to build an interpolating polynomial $P(x)$. The interpolation error evaluated at $x^*$
is $f(x^*) - P(x^*)$. The following theorem gives a formula for the interpolation error that is usually
impossible to evaluate exactly, but often can be at least bounded by an error.

**Theorem 1.** Assume that $P(x)$ is the (degree $n-1$ or less) interpolating polynomial fitting the $n$
points $(x_1, y_1), \ldots, (x_n, y_n)$. The interpolation error is

$$f(x) - P(x) = \frac{(x - x_1)(x - x_2) \cdots (x - x_n)}{n!} f^{(n)}(c)$$

where $c$ lies between the smallest and largest of the numbers $x, x_1, \ldots, x_n$.

The motivation for Chebyshev interpolation is to improve control of the maximum value of the
interpolation error given by (1). For a given positive integer $n$, the Chebyshev nodes in the interval
$(-1, 1)$ are

$$x_i = \cos \left( \frac{2i - 1}{2n} \pi \right), \; i = 1, \ldots, n.$$
The interpolation error due to Chebyshev positioning formula (2), is summarized in the following theorem:

**Theorem 2.** The choice of real numbers $-1 \leq x_1, \ldots, x_n \leq 1$ that makes the value of

$$\max_{-1 \leq x \leq 1} |(x-x_1) \cdots (x-x_n)|$$

as small as possible is given by (2), and the minimum value is $\frac{1}{2^{n-1}}$. In fact, the minimum is achieved by

$$(x-x_1) \cdots (x-x_n) = \frac{1}{2^{n-1}} T_n(x)$$

where $T_n(x)$ denotes the degree $n$ Chebyshev polynomial.

**Example 3.1.** In this example, we compare the errors for polynomial interpolation and Chebyshev interpolation with different interpolation points for the function $e^x$ on $[-1, 1]$. Figure 5 shows the comparisons for $N = 5$ and $N = 8$, and there is no significant difference between the two interpolations for $N = 5$. In contrast, we see better approximation for the Chebyshev interpolation near the boundary when $N = 8$.

The interpolation formula (1) gives

$$f(x) - P_4(x) = \frac{(x+1)(x+\frac{1}{2})x(x-\frac{1}{2})(x-1)}{5!} f^{(5)}(c)$$

where $-1 < c < 1$. For $-1 \leq x \leq 1$, the error is bounded by

$$f(x) - P_4(x) = \frac{(x+1)(x+\frac{1}{2})x(x-\frac{1}{2})(x-1)}{5!} f^{(5)}(c) \leq \frac{0.11348}{5!} e \approx 0.002570$$

The interpolation formula (4) gives

$$|e^x - P_4(x)| \leq \frac{e}{2^{15!}} \approx 0.00142$$

**Example 3.2.** In this example, we interpolate function $f(x) = 1/(1 + 12x^2)$ by using both polynomial and Chebyshev method with 15 points. Figure 6 shows the Runge phenomenon from polynomial interpolation.
Figure 7. A model rocket created by combining basic shapes via Rvachev functions.

Figure 6. Interpolation of function $f(x) = \frac{1}{1+12x^2}$

4. Technical Challenges

Creating complex shapes and visualization. Using Rvachev functions, it is possible to create complex objects by combining basic shapes such as elliptic paraboloids, planes, and spheres. In Figure 7, we show a model rocket created using this approach. Finding the correct sequence of shapes and Rvachev functions, is in general non-trivial. Another issue is finding an appropriate visualization tool. For this purpose, we used the MATLAB code `ezimplot3` which can be found in MATLAB Central [4].

Numerical approximation issues. In the pursuit of accurate solutions of optical cross section for more complex objects, one major challenge arises when the object being considered has sharp edges. Even in one dimension, polynomial approximation can be challenging for functions that are not differentiable everywhere, such as $f(x) = |x|$ on the interval $[-1, 1]$. Even with R-functions providing a means of obtaining implicit representations for combinations of objects, these implicit representations often contain terms that make polynomial approximation difficult. For example, the implicit function obtained using R-functions that describes the union of two spheres of radius one with centers at the origin and $(1.5, 0, 0)$ is

$$(x - \frac{3}{2})^2 - \sqrt{(x^2 + y^2 + z^2 - 1)^2 + \left( (x - \frac{3}{2})^2 + y^2 + z^2 - 1 \right)^2 + x^2 + 2y^2 + 2z^2 - 2} = 0$$
Figure 8. An example of approximating an object with sharp edges. Here a cube is approximated by the function $x^n + y^n + z^n = 1$ for $n = 2, 4,$ and $8$ (from left to right).

One solution could be to manually increase the tolerance (and thus decrease the desired accuracy) of the polynomial approximation, though doing so for complex objects might yield approximations that are far less accurate than using facets to discretize the surface. Another approach might be to use smoothed versions of objects, such as the equation $x^n + y^n + z^n = 1$ for large even values of $n$ to approximate a cube centered at the origin (see Figure 8). While there will certainly be a trade-off in the accuracy, investigation will be needed to fully evaluate the viability of this approach and to compare it with the approach of simply increasing the approximation tolerance.

5. Results and Discussions

5.1. Analytical Solutions for Sphere.
Recall that by using the Blinn–Phong model for a sphere, we can compute the OCS through

$$\sigma = \int \int \int_A \text{BRDF}(\mathbf{I}, \mathbf{V}; \mathbf{N}) \cos \gamma \cos \beta \mathbb{1}_{(\mathbf{N}, \mathbf{I}) \geq 0} \mathbb{1}_{(\mathbf{V}, \mathbf{N}) \geq 0} dA$$

where $\gamma$ is the angle between $\mathbf{N} = (\cos \vartheta \sin \varphi, \sin \vartheta \sin \varphi, \cos \varphi)$ and $\mathbf{V} = (V_1, V_2, V_3)$, $\beta$ is the angle between $\mathbf{N}$ and $\mathbf{I} = (I_1, I_2, I_3)$. The geometric relations are shown in the picture below.
The appearances of two indicator functions

\[
1_{(N,I)\geq 0} = \{(\vartheta, \varphi) : \cos \vartheta \sin \varphi I_1 + \sin \vartheta \sin \varphi I_2 + \cos \varphi I_3 \geq 0\}
\]

\[
1_{(V,N)\geq 0} = \{(\vartheta, \varphi) : \cos \vartheta \sin \varphi V_1 + \sin \vartheta \sin \varphi V_2 + \cos \varphi V_3 \geq 0\}
\]

in the formula are to ensure only the common part of the illuminated area and the detectable area are included in the region of integration. The spherical coordinates \((r, \vartheta, \varphi)\) are related to the Cartesian coordinates \((x, y, z)\) via

\[
x = r \cos \vartheta \sin \varphi, \quad y = r \sin \vartheta \sin \varphi, \quad z = r \cos \varphi
\]

with Jacobian

\[
\left| \frac{\partial (x, y, z)}{\partial (r, \vartheta, \varphi)} \right| = r^2 \sin \varphi
\]

Thus by noticing

\[
\cos \gamma = \frac{(N, V)}{\|N\| \|V\|}, \quad \cos \beta = \frac{(N, I)}{\|N\| \|I\|}
\]

we arrive at

\[
\sigma = \int_{0}^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \left( \frac{(I + V, N)}{\|I + V\|} \right)^{\alpha} \frac{(N, V)}{\|N\| \|V\|} \frac{(N, I)}{\|N\| \|I\|} r^2 \sin \varphi 1_{(N,V)\geq 0} 1_{(N,I)\geq 0} \, d\vartheta \, d\varphi
\]

5.2. Numerical Calculation of Scattering Function for a Sphere: finite element approach. For each facet in the model, find the angle \(\theta\) between the incident ray and facet normal, in Figure 9. Target reflectivity, \(\rho\), is given by \(MRDF(\theta)\), and the scattering function, \(\sigma\), is estimated by

\[
\sigma = \frac{\rho A}{\Omega}
\]

where \(A\) is target area, \(\Omega\) is the solid angle of scattering.

For a sphere, we generate an approximation mesh by using an icosahedron. An icosahedron is a polyhedron composed of twenty identical equilateral triangles, each triangle has the same area and each vertex is at the same distance from all of its neighbors. To get a higher number of triangles we need to subdivide each triangle by creating a new vertex at the mid point of each edge which is then normalized, to make it lie on the sphere’s surface. This breaks the initial properties of the icosahedron, the triangles are not equilateral anymore and neither the area nor the distance between adjacent vertices is the same across the mesh. But it is still a better approximation by almost any measure excluding its number of triangles.

![Figure 9. Incident angle \(\theta\) on each triangle facet](image-url)
The calculation of the optical cross section is summarized as follows:

- Generate a mesh for a given model
- For each facet, calculate the reflectance from the incident angle, material properties, etc.
- Sum over all visible facets to obtain the OCS

Table 1 shows the numerical results for different mesh sizes of a sphere with constant $\rho = 1$ and $\alpha = 0$. The analytical OCS for a sphere is $2\pi/3 \approx 2.094395102389723$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>Numerical Results</th>
<th>Relative error</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>320</td>
<td>1.99294</td>
<td>4.84419 $\times$ 02</td>
<td></td>
</tr>
<tr>
<td>1280</td>
<td>2.06845</td>
<td>1.23853 $\times$ 02</td>
<td>0.98381</td>
</tr>
<tr>
<td>5120</td>
<td>2.08787</td>
<td>3.11387 $\times$ 03</td>
<td>0.99593</td>
</tr>
<tr>
<td>20480</td>
<td>2.09276</td>
<td>7.79570 $\times$ 04</td>
<td>0.99898</td>
</tr>
</tbody>
</table>

5.3. **Numerical Solutions for Sphere & Application to Ellipsoid: analytic approach.**

To capture the target area for numerical integration using Chebfun, the Boolean relationship is incorporated to avoid defining explicit boundary angles. This is achieved by multiplying the integrand derived from the Blinn–Phong model by $\max\{0, \langle I, N \rangle\}$ and $\max\{0, \langle V, N \rangle\}$ which rules out area not in the intersection of the lighted area and viewer’s range. Here $\langle \cdot, \cdot \rangle$ is the inner product, $I$ is the light source direction, $V$ is the viewer direction, and $N$ is the outward unit normal of the surface.

**Case 1: Method verification.** Verification of the accuracy of Chebfun approximation is performed by simulating the case where the light source and the viewer are in the same direction.
to capture the optical cross section value. The reflectivity and the light diffusivity of the surface are fixed with $\rho = 1$ and $\alpha = 0$ respectively. The optical cross section (OCS) approximated using Chebfun, with interpolating error tolerance of $10^{-8}$, yields $\sigma_{\text{cheb}} = 2.094395102389723$. The known analytical solution is $\sigma_{\text{analytical}} = 2\pi/3 \approx 2.094395102393195$. The absolute error of the Chebfun approach is in the range of $10^{-12}$ due to its semi-analytical nature. This is superior to the triangular discretization approach which performs at $\approx 10^{-4}$ accuracy. To ensure the Chebfun approach does not produce non-physical results, additional tests were run for different $\alpha$ values and different viewer locations.

![OCS value for selected $\alpha$](image)

**Figure 11.** Simulation of orbiting viewer on the plane with different $\alpha$ values.

The $\alpha$ value is chosen to be in the interval $[0, 20]$. The results show that as $\alpha$ increases the OCS value decreases (Figure 11). Since higher $\alpha$ values represent smaller specular highlight, the viewer would receive less total reflected light across the whole surface. This is in agreement with physical observation.

Next we tested the influence of the viewer position. Two scenarios were tested: parallel orbiting viewer and perpendicular orbiting viewer, with respect to the light source. In the parallel orbiting scenario (Figure 11), the OCS value decreases as the viewer moves away from the light source, followed by restoration to the initial value. The rate of decay is clearly influenced by the $\alpha$ value. In the perpendicular orbiting scenario, the viewer moves in the xy-plane while the light source is fixed at $(\theta, \phi) = (0, 0)$. The simulation result shows that the OCS value remains constant.

Both orbiting simulation results show agreement with the physical phenomenon. This indicates that the Chebfun approach can produce accurate and physically meaningful results.

**Case 2: Ellipsoid I** Following the verification we applied the Chebfun approach to a non-uniform shape. A triaxial ellipsoid was selected due to its simple geometry and non-uniform symmetry. We started with the ellipsoid $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$ and the constants $\rho$ and $\alpha$ are as in the verification section. However, due to the artificial singularities generated from the parameterization
of the surface, we intentionally excluded a small neighborhood near the “north pole” and the “south pole” to avoid instability. The light source is fixed at \((\theta, \phi) = (\pi/2, \pi/2)\) and the viewer orbits in the \(xy\)-plane. The Chebfun approach yields an OCS value \(\approx 0.9821\) when the viewer and the light source are at the same direction. This lower observed value is likely due to the difference in curvatures of the surface facing \((\theta, \phi) = (\pi/2, \pi/2)\) compared to a hemisphere. As in the sphere case, the angle between the unit normal and the viewer direction increases at a constant rate, in the ellipsoid case the the angle approaches 90 degrees at a faster rate. Similar to the sphere case, the OCS value drops as the viewer moves further away from the light source. The decay rates of
Figure 14. Simulation of orbiting viewer on the same plane as the light source. 
\( \alpha = 0 \). The shape of interest here is an ellipsoid.

OCS value between the sphere and ellipsoid case are expected and can be observed in Figure 11 and Figure 14. We also performed the same test simulating viewer orbiting on a plane perpendicular to the light source, the resulting periodic oscillation obtained is determined to be physical (Fig. 15).

**Case 3: Ellipsoid II** We next chose the ellipsoid described by

\[ \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \]

To compare the result with the ellipsoid I case. The remaining setup is identical to the previous case. Using the Chebfun approach we found that the OCS value when the light source and the viewer are at the same direction is about 1.6786. Since the viewer would be facing a larger area, the OCS value is expected to be higher. The result turns out to be as expected such that the OCS value produced is higher than the previous case. Finally we looked at the decay rate of the OCS value based on the increase of specular highlight. The initially faster decay rate of the OCS value for the ellipsoid I case is observed. This is due to change of curvature appearing earlier than the ellipsoid II case as the viewer moves away from the light source (Fig. 13 and Fig. 16).

5.4. **The Multipath Problem.** Let’s now consider the case where light is projected on a concave surface. Unlike the previous cases, the light detected by the sensor can be reflected multiple times from points on the surface due to concavity. Theoretically, the bouncing of light can go on infinitely if there is no energy dissipation. The surface we consider here is the half cylindrical shell facing downwards and the radius of the cylinder is \( r \). Given \( \psi_I \), we denote the incident light by its vector direction \( \mathbf{I}_1 = (\cos \psi_I, \sin \psi_I) \) and the light detector has viewing angle \( \mathbf{V} \). For light that is reflected by the surface at \( D_1, D_2, ..., D_n \) before it is collected by the light detector, denote the (inward) normal vector at \( D_n \) by \( \mathbf{N}_n = (-\cos \psi_n, -\sin \psi_n) \) and unit vector for \( \overrightarrow{D_nD_{n-1}} \) by \( \mathbf{I}_n = (-\cos \psi_n, -\sin \psi_n) \). Define inductively
Figure 15. Simulation of orbiting viewer on the plane perpendicular to the light source. $\alpha = 0$. The shape of interest here is an ellipsoid.

Figure 16. OCS value of the ellipsoid simulations with increasing $\alpha$ values. It can be observed that ellipsoid I experiences a faster decay due to the high curvature decay.

\[
\begin{align*}
I_1 &= I = (\cos \psi_I, \sin \psi_I), \\
\delta_n &= \vartheta_{n+1} - \vartheta_n, \\
\psi_n &= \psi_{n-1} - (\pi - \delta_n)
\end{align*}
\]
**Proposition 1.** The Optical Cross Section, up to $N$ bounces, is computed as

$$\sigma = \sum_{n=1}^{N} \sigma_n$$

with

$$\sigma_n = \int_{0}^{\pi} \ldots \int_{0}^{\pi} \left( \frac{\langle I_n + V, N_n \rangle}{\|I_n + V\|} \right) \frac{\alpha}{\|N_n\||V\|} \frac{\langle N_n, I_n \rangle}{\|N_n\||I_n\|}$$

$$\times \prod_{i=1}^{n-1} \left( \frac{\langle I_i + I_{i+1}, N_i \rangle}{\|I_i + I_{i+1}\|} \right) \frac{\alpha}{\|N_i\||I_{i+1}\|} \frac{\langle N_i, I_i \rangle}{\|N_i\||I_i\|} \frac{1}{\|\tau N_{i+1} - \tau N_i\|^2} \right)^{r_n} d\theta_1 \ldots d\theta_n$$

assuming that the integral converges.

**Proof.** The contribution from a single bounce to the OCS is easily computed by using Blinn-Phong's model with the projected BRDF

$$\sigma_1 = \int_{0}^{\pi} \left( \frac{\langle I_1 + V, N_1 \rangle}{\|I_1 + V\|} \right) \frac{\alpha}{\|N_1\||V\|} \frac{\langle N_1, I_1 \rangle}{\|N_1\||I_1\|} r d\theta_1$$

To compute the contribution from light that reflects multiple times before it is received, we need to write the relations between the angles correctly. For the case with two reflections, we can deduce these relations with the help from the picture below. Observe that now $\angle ID_1O = \angle OD_1D_2 = \angle D_1D_2O$ and $\vartheta_1 = \vartheta_2 + \angle D_2D_1O + \angle OD_1I$. Hence by defining $\delta_2 = \vartheta_2 - \vartheta_1$, $\angle D_2D_1O + \angle OD_1I = \pi - \delta_2$. So the polar angle $\psi_2$ for $I_2$ can be written as $\psi_1 - (\pi - \delta_2)$. More generally, the relations among the angles for even more reflections are given by (*).

Now let’s compute the contribution for light that reflects exactly $n$ times (happening at $D_1, \ldots, D_n$ respectively) before they are received. For each reflection at $D_i$, a differential length around $D_1$ becomes the new emitter that shines towards another differential length around $D_{i+1}$. The viewing
angle from $D_{i+1}$ towards $D_i$ now is just the direction of the reflected light $I_i$. So we can invoke the Blinn-Phong model to compute the detectable light at $D_{i+1}$ from $D_i$ and get the projected BRDF

$$\text{BRDF}^*_{D_i} = \left( \frac{\langle I_i + I_{i+1}, N_i \rangle}{\| I_i + I_{i+1} \|} \right)^\alpha \frac{\langle N_i, I_{i+1} \rangle}{\| N_i \| \| I_{i+1} \|} \frac{1}{\| N_i \| \| I_i \|} \frac{1}{\| \overrightarrow{rN_i} \|}$$

Therefore, for light that reflects exactly $n$-times before reaching the detector, the total contributions to $OCS$ would be

$$\sigma_n = \int_0^\pi \ldots \int_0^\pi \left( \frac{\langle I_n + V, N_n \rangle}{\| I_n + V \|} \right)^\alpha \frac{\langle N_n, V \rangle}{\| N_n \| \| V \|} \frac{\langle N_n, I_n \rangle}{\| N_n \| \| I_n \|} \left( \prod_{i=1}^{n-1} \text{BRDF}^*_{D_i} \right) r^n \, d\vartheta_1 \ldots d\vartheta_n$$

The appearance of the factor $r^n$ is due to the Jacobians from rewriting line integrals into usual integrals. Total contributions for up to $n$ reflections is then given by

$$\sigma = \sigma_1 + \sigma_2 + \ldots + \sigma_n$$

We can use the same method to derive the multi-path solution for a spherical bowl shape. On the figure, observe that now the reflected light rays will always stay in the same plane generated by original incident vector and the normal at the first reflection point. The surface boundary for reflections is also half (tilted) circle.

6. Conclusion

In this work we have investigated several aspects of light reflection. Our main goals were to understand the diffusivity of the reflection associated with surface materials. To do so first we construct and de-construct complex geometric shapes using the Rvachev function which is based
on Boolean operations. Rvachev functions provide functional representations of the surface that do not require storage of information such as vertices, unit normal vectors and triangle adjacency. Second, we improve the current discrete model to a continuous model having less data storage, and we learned to use the combination of Chebfun and indicator functions, a computational tool that takes the advantage of Chebyshev polynomial interpolation to perform numerical integration over the surface with less need of explicit expression. Finally, we extend the continuous model to incorporate the multi-path phenomenon starting from the simple case of two-dimensional one reflection model and extended the result to build the theory for the three-dimensional n reflections model.

7. ACKNOWLEDGMENT

This research was conducted as part of the IMSM workshop organized by SAMSI supported by NSF. The authors thank Prof. Mansoor Haider and Thomas Gehrmann for organizing and recruiting. The authors also thank John Peach for introducing and guiding the project, as well as Prof. Alen Alexanderian for instructions and suggestions throughout.

REFERENCES


References

    https://www.mathworks.com/matlabcentral/fileexchange/23623-ezimplot3-implicit-3d-functions-plotter?
Identifying Precision Treatment for Rheumatoid Arthritis with Reinforcement Learning

Chixiang Chen¹, Ashley Gannon², Duwani Katumullage³, Miaoqi Li⁴, Mengfei Liu⁵, Rebecca North⁶, Jialu Wang⁷

Industry Mentors: Grant Weller⁸, Victoria Mansfield⁹, Yinglong Guo¹⁰
Faculty Mentor: Daniel Luckett¹¹

Abstract

Rheumatoid Arthritis (RA) is an autoimmune disease that causes chronic inflammation in the lining of joints leading to multiple complications including painful swelling, long-term damage from bone erosion, and joint deformity. The diagnosis and treatment of RA with precision medicine is challenging, as there is a high rate of co-morbid conditions that effect various organs and there are currently no clinically validated methods for measuring disease progression. In this work, we analyze a set of longitudinal administrative health data from more than 6,000 RA patients, and propose a framework to identify an optimal individualized dynamic treatment regime (DTR) by applying Q-learning which is a popular data-driven methodology for multistage decision problems in both randomized trials and observed data.

1 Introduction

Rheumatoid arthritis (RA) is a heterogeneous, chronic inflammatory disease that affects 1% of people worldwide [1, 2]. RA predominately affects the lining of the joints, causing pain and inflammatory flare ups. It also effects organs such as the heart, skin, eyes and lungs, causing growths, vasculitis, scleritis, Sjogren’s syndrome, and a laundry list of other ailments. In addition to the discomfort and co-morbid conditions, patients diagnosed with RA have a 60% increase in the risk of heart attack one year after diagnosis and are twice as likely as the average person to develop depression. The variety of symptoms that can manifest in this patient population makes RA a difficult disease to diagnose and manage. Additionally, there is currently no single, autoantibody specific diagnostic test available to diagnose patients [3], and the American College of Rheumatology (ACR) does not provide an optimal treatment regime to follow once the patient has been diagnosed. Thus, determining which of the many treatment regimes provided by the ACR is the optimal treatment regime for RA is the focus of this work.

Because RA is a chronic illness, it is imperative to develop a long-term treatment strategy. Treatments vary from person to person and can depend on factors such as treatment history, disease state, age, personal preferences, etc.. Dynamic treatment regimes (DTRs) have been used to generate long-term individualized treatment plans for patients with a variety of chronic illnesses (e.g. as in [4]). In theory, this framework maps an individual’s current characteristics to an optimized set of possible treatments at each decision point. In reality, it is an onerous task to identify optimal DTRs from large data sets. One possible way to approximate DTRs is to use a machine learning approach, specifically a reinforcement learning approach. Reinforcement learning can be used to optimize the expected outcome and produce a data-driven policy. In this work, we will implement one such method, an approximate dynamic programming method known as Q-learning, with the intent to identify an optimal treatment regime for RA patients based on certain input variables. While there are several studies aiming to identify the comparative effectiveness of DTR for RA patients, to our knowledge, no data-driven effort to identify an optimal DTR for RA has been published.
Q-learning has become a popular method used to identify optimal DTRs from observational studies (e.g. as in [5], [6], and [7]), and is an appropriate method to apply here. The data in this work, provided by UnitedHealth Group, are composed of observational longitudinal data, derived from health insurance claims from more than 6,500 RA patients. Most studies that implement Q-learning to identify DTRs assume that the data are collected at small, finite numbers of treatment intervals. In this data set, there exists at minimum 1 year of claims following diagnosis of RA, and a minimum of 6 months of claims prior to diagnosis for each patient. All patient attributes in this data, which include comorbidities, number of visits to medical facilities and prescribed medications, and treatment information, are updated on a weekly basis. Applying a Q-learning algorithm to this data set can be challenging due to an imbalance of observation in treatments, potential latent confounds, and unclear implementation instructions for treatment intervals. Our team will address how we handled these difficulties in Section 3.

The remainder of this paper is organized as follows. Section 1 offers a plethora of descriptive statistics of the full data set and describes the data cleaning process. Section 3 details the basic methodology of Q-learning, followed by fitting a generalized linear model to the first two months of data, then implementing the Q-learning algorithm on the first three months of data. The results of these analyses are presented in Section 4, with a concluding discussion of our findings and suggested future work in Section 5.

2 Data Processing

2.1 Descriptive Statistics

To better understand the full data set, we first perform extensive descriptive statistical analyses on the attributes recorded for each patient. These attributes include age, gender, medical appointments, specific treatments, and preexisting medical conditions. This key step will aid in justifying the models that we develop later on.

The first attributes we consider are gender, age, and comorbidities. From Table 1, we note that 74% of patients are female, and half of the population is between the ages of 41 and 56. Since RA is known to affect nearly three times as many women as it does men, and RA most commonly develops between the ages of 30 and 60, this sample appears to be quite representative of the general population of individuals diagnosed with RA. We also note that there are several comorbidities listed that are not common within the given population (<5%), namely AIDS/HIV, acute myocardial infarction (AMI), dementia, paralysis, and renal failure. On the other hand, the three most common comorbidities are hypertension, chronic obstructive pulmonary disease (COPD), and depression.
Table 1: Descriptive table of age, gender and comorbidities. Continuous data are summarized by the minimum, maximum, mean (standard deviation), and median (interquartile range). The remaining binary data is summarized by their frequency and the percent of the population with the comorbidity (values in ()).

<table>
<thead>
<tr>
<th>Age at First Diagnosis</th>
<th>N = 6846</th>
<th>Dementia</th>
<th>N = 6846</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>12</td>
<td>Prior</td>
<td>7 (0)</td>
</tr>
<tr>
<td>max</td>
<td>64</td>
<td>Post</td>
<td>11 (0)</td>
</tr>
<tr>
<td>mean (sd)</td>
<td>47.26 ± 10.99</td>
<td>Diabetes</td>
<td></td>
</tr>
<tr>
<td>median (iqr)</td>
<td>50.00 (41.00, 56.00)</td>
<td>Prior</td>
<td>782 (11)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gender</th>
<th>Post</th>
<th>N = 866 (13)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>1,753 (26)</td>
<td>Hypertension</td>
</tr>
<tr>
<td>Female</td>
<td>5,093 (74)</td>
<td>Prior</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AIDS/HIV</th>
<th>Post</th>
<th>N = 2315 (34)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>9 (0)</td>
<td>Liver Disease</td>
</tr>
<tr>
<td>Post</td>
<td>13 (0)</td>
<td>Prior</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Acute Myocardial Infarction</th>
<th>Post</th>
<th>N = 557 (8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>67 (1)</td>
<td>Paralysis</td>
</tr>
<tr>
<td>Post</td>
<td>81 (1)</td>
<td>Prior</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Angina</th>
<th>Post</th>
<th>N = 32 (0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>409 (6)</td>
<td>Peripheral Vascular Disease</td>
</tr>
<tr>
<td>Post</td>
<td>417 (6)</td>
<td>Prior</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cancer</th>
<th>Post</th>
<th>N = 275 (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>294 (4)</td>
<td>Renal Failure</td>
</tr>
<tr>
<td>Post</td>
<td>357 (5)</td>
<td>Prior</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cerebrovascular Disease</th>
<th>Post</th>
<th>N = 192 (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>303 (4)</td>
<td>Ulcers</td>
</tr>
<tr>
<td>Post</td>
<td>296 (4)</td>
<td>Prior</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Congestive Heart Failure</th>
<th>Post</th>
<th>N = 97 (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>131 (2)</td>
<td>Depression</td>
</tr>
<tr>
<td>Post</td>
<td>192 (3)</td>
<td>Prior</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COPD</th>
<th>Post</th>
<th>N = 1,129 (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior</td>
<td>1,423 (21)</td>
<td>Skin Ulcers</td>
</tr>
<tr>
<td>Post</td>
<td>1,296 (19)</td>
<td>Prior</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Percent of Observations:</th>
<th>DMARD</th>
<th>NSAID</th>
<th>Glucocorticoid</th>
<th>Opioid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>628,694</td>
<td>1,036,441</td>
<td>1,002,517</td>
<td>1,117,019</td>
</tr>
<tr>
<td>1</td>
<td>499,013</td>
<td>91,266</td>
<td>125,190</td>
<td>10,688</td>
</tr>
</tbody>
</table>

Table 2: Number of times a medication is prescribed after patient diagnosis across all weeks.

For each week, each patient has a treatment ID that denotes his/her treatment for that week. The treatment IDs are a series of five indicators, one for each of the five types of disease-modifying antirheumatic drug (DMARD) that can be prescribed: traditional synthetic DMARDs, namely hydroxychloroquine, leflunomide, and sulfasalazine; methotrexate (MTX), a synthetic DMARD that is also used to treat cancer patients; non anti-tumor necrosis factor (nTNF), biologic DMARD that is composed of monoclonal anti-tumor necrosis factor antibodies and inhibits TNF in the body; anti-tumor necrosis factor (TNF), another type of biologic DMARD that inhibits TNF, but instead contains soluble TNF receptors; and tofacitinib (tofa), a biologic DMARD that inhibits the JAK1 enzyme and disrupts cell signaling pathways. Figure 1 gives an example of a possible treatment ID, and Table 3 gives the frequency with which each combination of DMARDs was...
followed as a treatment. The first item to note from Table 3 is that the treatment ID “00000”, which indicates no DMARD use, has the highest frequency of all treatments. This seemingly counter-intuitive event is likely due to the skewed distribution of treatment initiation as depicted in Figure 2. Second, there are quite a few treatment groups that have significantly smaller frequencies. Upon investigation of this phenomenon, we discovered that these treatment IDs correspond to a transition between two more common treatment groups rather than to a legitimate treatment option.

Table 3: Frequency of treatment occurrence across all weeks following patient diagnosis.

<table>
<thead>
<tr>
<th>Treatment Group</th>
<th>Frequency</th>
<th>Treatment Group</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>413204</td>
<td>10000</td>
<td>158739</td>
</tr>
<tr>
<td>00001</td>
<td>2470</td>
<td>10001</td>
<td>993</td>
</tr>
<tr>
<td>00010</td>
<td>70072</td>
<td>10010</td>
<td>14935</td>
</tr>
<tr>
<td>00011</td>
<td>21</td>
<td>10020</td>
<td>54</td>
</tr>
<tr>
<td>00020</td>
<td>179</td>
<td>10100</td>
<td>1626</td>
</tr>
<tr>
<td>00100</td>
<td>5068</td>
<td>10101</td>
<td>5</td>
</tr>
<tr>
<td>00101</td>
<td>12</td>
<td>10110</td>
<td>16</td>
</tr>
<tr>
<td>00110</td>
<td>34</td>
<td>11000</td>
<td>38013</td>
</tr>
<tr>
<td>00200</td>
<td>3</td>
<td>11001</td>
<td>220</td>
</tr>
<tr>
<td>01000</td>
<td>145060</td>
<td>11010</td>
<td>5436</td>
</tr>
<tr>
<td>01001</td>
<td>753</td>
<td>11100</td>
<td>574</td>
</tr>
<tr>
<td>01010</td>
<td>31991</td>
<td>20000</td>
<td>14221</td>
</tr>
<tr>
<td>01011</td>
<td>7</td>
<td>20001</td>
<td>11</td>
</tr>
<tr>
<td>01020</td>
<td>66</td>
<td>20010</td>
<td>2113</td>
</tr>
<tr>
<td>01100</td>
<td>1635</td>
<td>20100</td>
<td>397</td>
</tr>
<tr>
<td>01101</td>
<td>2</td>
<td>21000</td>
<td>3694</td>
</tr>
<tr>
<td>01110</td>
<td>32</td>
<td>30000</td>
<td>916</td>
</tr>
<tr>
<td>01200</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Even though DMARDs are the primary drug therapy for RA patients to slow down the progression of the disease, RA is a chronic, heterogeneous disease and will thus manifest differently and at varying intensities across patients, and across time. This often leads RA patients to supplementing their primary drug therapy with a glucocorticoid, a non-steroidal anti-inflammatory drug (NSAID), or an opioid. Table 2 gives the frequency across all post-diagnosis weeks with which DMARDs, NSAIDs, glucocorticoids, and opioids were used. We see that there were more weeks where one or more types of drug were not used after diagnosis, at least initially. While this is an encouraging result for the categories of NSAID, glucocorticoid, and opioid, it is something of a surprise for DMARDs. However, examination of the empirical distribution of treatment initiation in Figure 2 allows us to infer that the inflated number of non-DMARD weeks is a result of the lag between diagnosis and treatment commencement for a many of the patients. We can also note that, of the prescribed medications, the most common prescriptions are for disease-modifying antirheumatic drugs (DMARDs), with non-steroidal anti-inflammatory drugs (NSAIDs) and glucocorticoids being prescribed at similar frequencies and opioids prescribed the least.

Next, we consider the frequency with which medications were prescribed for each classification of drug among DMARDs, NSAIDs, glucocorticoids, and opioids. From Figure 3, we find that, by category, the most frequently prescribed medications are methotrexate sodium (36%) among DMARDs, Celecoxib (30%) among NSAIDs, Prednisone (97%) among glucocorticoids, and oxycodone HCL (47%) among opioids.
Figure 2: Empirical distribution of the number of weeks between patient diagnosis and the start of treatment. The minimum lag between diagnosis and treatment is recorded as -4 weeks, and the maximum lag is 336 weeks, or nearly 6 years. The former may be a gap in the claims data or the time it took for a full diagnosis. The median, at least, is 4 weeks, with the third quartile at 25 weeks. The ACR recommends that individuals diagnosed with RA begin DMARD treatment within the first 3 months after diagnosis [8]. This guideline appears to be followed by more than half of the RA patients in our data set, and nearly 75% began treatment in the first 6 months.

Figure 3: Descriptive summary of the frequencies of medications prescribed to RA patients.

Next, we explore the frequency of transitions between the use of glucocorticoids and painkillers as well as the overall use of glucocorticoids and painkillers across all weeks, where painkillers include NSAIDs and
opioids. From Table 4, we notice that most patients do not change their supplemental treatments over time. When a change does occur, it is most often to stop taking a glucocorticoid or painkiller. From Table 5, we see that many RA patients were already using a painkiller or glucocorticoid before diagnosis, which suggests that they were either experiencing many RA symptoms before diagnosis or there were other preexisting conditions that called for the use of these drugs. This fact induces limitations in our analysis in Section 3 when using painkiller or glucocorticoid use as a surrogate response for treatment effectiveness.

<table>
<thead>
<tr>
<th>Change In Treatment</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Glucocorticoid → Glucocorticoid</td>
<td>4.17%</td>
</tr>
<tr>
<td>Glucocorticoid → No Glucocorticoid</td>
<td>17.84%</td>
</tr>
<tr>
<td>No Painkiller → Painkiller</td>
<td>3.54%</td>
</tr>
<tr>
<td>Painkiller → No Painkiller</td>
<td>9.08%</td>
</tr>
<tr>
<td>Glucocorticoid → Painkiller</td>
<td>1.18%</td>
</tr>
<tr>
<td>Painkiller → Glucocorticoid</td>
<td>1.00%</td>
</tr>
<tr>
<td>No Change</td>
<td>63.18%</td>
</tr>
</tbody>
</table>

Table 4: Average percent of supplemental treatment changes that occur across all weeks.

<table>
<thead>
<tr>
<th></th>
<th>Painkillers</th>
<th>Glucocorticoids</th>
<th>Both</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before Diagnosis</td>
<td>37.2%</td>
<td>25.6%</td>
<td>18.0%</td>
</tr>
<tr>
<td>After Diagnosis</td>
<td>47.1%</td>
<td>65.6%</td>
<td>15.7%</td>
</tr>
</tbody>
</table>

Table 5: Percentage of supplemental medications prescribed to patients before and after RA diagnosis.

Finally, we consider various types of medical visits, including inpatient (IP) visits, length of IP stay, outpatient (OP) visits, emergency room (ER) visits, and doctor’s (DR) visits, as well as filled prescriptions (RX scripts). Table 6 shows that the average RA patient will visit a doctor about once a month, go for an outpatient visit once every other month, and take one or two prescription medications each week.

<table>
<thead>
<tr>
<th></th>
<th>IP Visits</th>
<th>IP Days</th>
<th>OP Visits</th>
<th>ER Visits</th>
<th>DR Visits</th>
<th>RX scripts</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.0009</td>
<td>0.0037</td>
<td>0.1484</td>
<td>0.0039</td>
<td>0.3127</td>
<td>1.340</td>
</tr>
<tr>
<td>sd</td>
<td>0.0222</td>
<td>0.0836</td>
<td>0.3528</td>
<td>0.0492</td>
<td>0.5609</td>
<td>1.543</td>
</tr>
</tbody>
</table>

Table 6: This table summarizes the average and standard deviation over all treatment IDs of medical visits and prescription fills.

2.2 Data Cleaning

To simplify the data, we first remove all weeks prior to the start of treatment for each patient, under the assumption that they do not contribute significantly to the choice of treatment. Next, since treatment decisions are typically made on a monthly basis rather than a weekly basis, we collapse the data from weekly records to monthly records. This change makes the data more informative and less computationally expensive. To convert the medical visits (IP, OP, ER, DR) and prescriptions into monthly records, we sum the number of events over the 4 weeks in each month. New treatment IDs are assigned by taking the maximum number that occurs in each column of the four original treatment IDs. We also create new variables to indicate the number of treatment changes that occurred in a given month, if the patient took a painkiller that month (0 if no, 1 if yes), and if the patient took a glucocorticoid that month (0 if no, 1 if yes).

After collapsing the data into monthly intervals, we remove 38 patients who have only one month of treatment data, as our methods require a second month of data to provide an outcome for the analysis. Further data cleaning is completed and noted as needed in the analysis sections.
3 Methodology

Let’s start at the general notation. Our goal is to minimize the following the reward function to get the corresponding optimal decision rules:

\[
\begin{align*}
( \, d_1(X_1), \, d_2(X_2), \ldots, \, d_n(X_n) \,) &= \underset{a_1, a_2, \ldots, a_n}{\text{argmin}} \, \mathbb{E} \{ Y | A_1 = a_1, A_2 = a_2, \ldots, A_n = a_n, X \} \\
\end{align*}
\]  

(1)

Where

\[
\begin{align*}
&i = 1, 2, \ldots, n \\
&X_i \equiv \text{covariate matrix at time point } i \\
&d_i(X_i) \equiv \text{optimal decision at time point } i \\
&Y \equiv \text{last stage outcome} \\
&A_i \equiv \text{treatment/decision at time point } i.
\end{align*}
\]

In this paper, let \( Y \) denote the outcome that indicates treatment efficacy. The true outcome of interest is the frequency of flareups. However, since claims data does not provide an exact measurement of these events, we chose the use of painkillers or glucocorticoids to be a surrogate response. The covariates we use to predict this outcome could be any subset of the patient attributes that are provided in the claims. We allow some overlap among these covariates \( X_i \) at different stages. The decision space for \( a_i, \, i = 1, \ldots, n \), can also vary across time. The key technique we apply to solve (1) is Q-learning. In the next subsection, we illustrate our method by first starting at the simplest situation that only involves a one-stage decision rule, then extend it to a two-stage decision-making process.

3.1 A One-Decision Model

This is the fundamental stage of the DTR, when we only consider the baseline cross sectional data for each patient at the initial stage of their disease. Hence, the dynamic aspect of RA need not be taken into account yet.

In (2), we conform to the general notation given at the beginning of this section to discover the treatment groups that minimize the expected value of the outcome, which we call the optimal decision rule.

\[
\hat{d}(x) = \underset{a}{\text{argmin}} \, \mathbb{E}(\hat{Y}|X = x, A = a).
\]

(2)

Since only one decision is considered here, we can apply logistic regression to model \( \mathbb{E}(Y|X, A) \) with the training data, then directly obtain the optimal decision rule by minimizing the estimated model from the test data.

To evaluate the performance of the optimal decision rule, we need to calculate and compare the predicted \( \mathbb{E}(Y|A) \) based on the optimized decision rule and observed treatment combination in the test data, respectively. A naive prediction formula is given by (3):

\[
\hat{Y} = \frac{\sum^n_i Y_i 1 \{ A_i = \hat{d}(x_i) \}}{\sum^n_i 1 \{ A_i = \hat{d}(x_i) \}} \quad \text{and} \quad \hat{Y}_{\text{obs}} = \frac{\sum^n_i Y_i 1 \{ A_i = a_i \}}{\sum^n_i 1 \{ A_i = a_i \}}.
\]

(3)

However, (3) is not typically acceptable due to the fact that treatment assignment is not randomized in an observational study, and thus cannot guarantee an unbiased estimator for \( \mathbb{E}(Y|A) \). To account for the bias, we can apply the method of inverse probability treatment weights (IPTW), as in [9]:

\[
\hat{Y}_{\text{IPTW}} = \frac{\sum^n_i Y_i 1 \{ A_i = \hat{d}(x_i) \} P(A_i|x_i)}{\sum^n_i 1 \{ A_i = \hat{d}(x_i) \} P(A_i|x_i)} \quad \text{and} \quad \hat{Y}_{\text{IPTW-obs}} = \frac{\sum^n_i Y_i 1 \{ A_i = a_i \} P(A_i=a_i|x_i)}{\sum^n_i 1 \{ A_i = a_i \} P(A_i=a_i|x_i)}.
\]

(4)
Here, \( \tilde{Y}_{\text{IPTW}} \) is the IPTW estimator based on the optimized decision rule, and \( \tilde{Y}_{\text{IPTW-obs}} \) is the IPTW estimator based on the observed treatment combination. The quantity \( P(A|X) \) is the propensity score utilized to adjust for bias, which can be modeled by multinomial logistic regression.

However, in cases where \( P(A|X) \) is misspecified, the estimation will still carry some bias. Thus, with the aim of making the estimation more resilient, we introduce a more robust formula that incorporates both the IPTW and the conditional expectation of the outcome, \( E(Y|X,A) \), known as the double robust estimator \([10]\).

The estimator in (5) allows for the misspecification of either \( P(A|X) \) or \( E(Y|X,A) \), but not both.

\[
\hat{Y}_{\text{IPTW-rob}} = \frac{1}{n} \sum_{i} \left\{ Y_i \mathbb{I}(A_i = \hat{d}(x_i)) - \frac{\hat{p}(A_i = \hat{d}(x_i)|x_i)}{\hat{p}(A_i = d(x_i)|x_i)} \hat{E}\{Y_i|x_i, A_i = \hat{d}(x_i)\} \right\}.
\] (5)

### 3.1.1 Data Set

To proceed with the aforementioned analysis, we extract the first month of data for all patients. We create a new outcome variable, a glucocorticoid-or-painkiller flag, by combining the individual indicators for glucocorticoid and painkiller use. We clean the data by eliminating uncommon treatment IDs. We define uncommon treatment IDs as an ID corresponding to fewer than 100 patients. The remaining treatment IDs for this section can be observed in Table 7. We remove these patients from this data set for a couple of reasons. The first is we want to avoid creating singular covariate matrices. The second is that we do not have enough replicates of these patients to make significant statistical conclusions regarding their treatment IDs. The final data set generated is then randomly divided into training data (70%) and testing data (30%).

<table>
<thead>
<tr>
<th>Treatment ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>00010</td>
</tr>
<tr>
<td>01000</td>
</tr>
<tr>
<td>10000</td>
</tr>
<tr>
<td>11000</td>
</tr>
</tbody>
</table>

Table 7: Remaining treatment IDs after data cleaning.

### 3.1.2 Model Selection

To narrow down the covariates that we will use, we implemented two variable selection methods: Random Forest and backward selection of a logistic regression model. The random forest algorithm is implemented using the Random Forest package in R, and the logistic regression model is fitted using glm() in R. From these two methods and our descriptive statistical analyses, we examine the top covariates and determine which covariates, \( X \), to be used in the approach.

<table>
<thead>
<tr>
<th>Random Forest</th>
<th>Backward selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age at diagnosis</td>
<td>Gender</td>
</tr>
<tr>
<td>Treatment group for month 1</td>
<td>Acute myocardial infarction</td>
</tr>
<tr>
<td>COPD</td>
<td>Hypertension</td>
</tr>
<tr>
<td>Subscriber</td>
<td>Angina</td>
</tr>
<tr>
<td>Gender</td>
<td>Paralysis</td>
</tr>
<tr>
<td>Hypertension</td>
<td>Skin</td>
</tr>
<tr>
<td></td>
<td>Renal failure</td>
</tr>
<tr>
<td></td>
<td>Peripheral vascular disease</td>
</tr>
<tr>
<td></td>
<td>Treatment group for month 1</td>
</tr>
</tbody>
</table>

Table 8: Covariates selected by Random Forest and backward selection.

In the logistic regression model, we notice the presence of covariates that rarely occur in the observed population (< 5%), as given in Table 9. Since such a small proportion of patients suffer from such ailments, we do not consider them in our approach. We also remove the subscriber index, to avoid over-fitting the model and since this covariate is less informative than the others.
### Table 9: Occurrence of covariates in two month data set.

<table>
<thead>
<tr>
<th>Covariate</th>
<th>Percent of Occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acute myocardial infarction</td>
<td>1%</td>
</tr>
<tr>
<td>Hypertension</td>
<td>34%</td>
</tr>
<tr>
<td>Angina</td>
<td>6%</td>
</tr>
<tr>
<td>Paralysis</td>
<td>1%</td>
</tr>
<tr>
<td>Skin ulcers</td>
<td>14%</td>
</tr>
<tr>
<td>Renal failure</td>
<td>1%</td>
</tr>
<tr>
<td>Peripheral vascular disease</td>
<td>1%</td>
</tr>
</tbody>
</table>

The final covariates we use in our model can be seen in Table 10. The logistic regression model for the expected conditional outcome will consist of the main effects of each of these covariates, as well as the interaction effects between these covariates and the treatment group for month 1. Moreover, for the propensity scores, the treatment group for month 1 will be modelled against the rest of the covariates mentioned in Table 10. The detailed result of this analysis will be discussed in Section 4.

### Final List of Covariates for the One Decision Model

<table>
<thead>
<tr>
<th>Covariate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age at diagnosis</td>
</tr>
<tr>
<td>Treatment group for month 1</td>
</tr>
<tr>
<td>COPD</td>
</tr>
<tr>
<td>Hypertension</td>
</tr>
<tr>
<td>Angina</td>
</tr>
<tr>
<td>Skin ulcers</td>
</tr>
<tr>
<td>Gender</td>
</tr>
</tbody>
</table>

Table 10: Final list of covariates for this model.

#### 3.2 A Two-Decision Model

The two-stage analysis is an extension of the one-stage analysis, and can easily be generalized to more than two stages based on the developed methodology and algorithms. Q-learning, a type of reinforcement learning, can be considered as one of the primary tools used in developing dynamic treatment regimes [5]. Let us define the Q-functions for a two stage analysis as follows [9]:

\[
f_2(X_2, A_2, X_1, A_1) = E[Y|X_2, A_2, X_1, A_1], \tag{6}
\]

\[
f_1(X_1, A_1) = E[min_{a_2} f_2(X_2, a_2, X_1, A_1)|X_1, A_1]. \tag{7}
\]

This algorithm implements backwards recursion to determine the optimal regime. We start with the final stage, stage 2 here, to optimize the treatment in that stage. Data are randomly divided into two parts, training and testing sets, composed of 70% and 30% of the patient population, respectively. Logistic regression is applied to the training set, and the individualized optimal treatments for stage 2 are determined based on (8):

\[
d_2(X_2, X_1, A_1) = \arg\min_{a_2} f_2(X_2, a_2, X_1, A_1). \tag{8}
\]

Next, for each patient in the training set, the predicted outcome, \( Y_i^* \in (0,1) \), is ascertained under the individualized optimal treatment regime in stage 2, and is used as the “pseudo-outcome” for stage 1. These pseudo-outcomes are transformed using the logit function, then fit to the covariates in the training set via linear regression. Finally, the individualized optimal regime for stage 1 is determined by (9):

\[
d_1(X_1) = \arg\min_{a_1} f_1(a_1, X_1). \tag{9}
\]
To evaluate the performance of the two-stage Q-learning model, similar criteria to that in Section 3.1 are utilized on the testing set, with exact expressions for these criteria given below:

$$\tilde{Y}_{\text{IPTW-obs}} = \frac{\sum_i Y_i \mathbb{I}(A_{1i} = a_{1i}) \mathbb{I}(A_{2i} = a_{2i})}{P(A_{1i} = a_{1i} | x_{1i}) P(A_{2i} = a_{2i} | x_{2i})} \left(\sum_i \mathbb{I}(A_{1i} = a_{1i}) \mathbb{I}(A_{2i} = a_{2i})\right)^{-1} \left(\sum_i Y_i \mathbb{I}(A_{1i} = d_{1i} | x_{1i}) \mathbb{I}(A_{2i} = d_{2i} | x_{2i})\right)$$

$$\tilde{Y}_{\text{IPTW}} = \frac{\sum_i Y_i \mathbb{I}(A_{1i} = a_{1i}) \mathbb{I}(A_{2i} = a_{2i})}{P(A_{1i} = a_{1i} | x_{1i}) P(A_{2i} = a_{2i} | x_{2i})} \left(\sum_i \mathbb{I}(A_{1i} = a_{1i}) \mathbb{I}(A_{2i} = a_{2i})\right)^{-1} \left(\sum_i Y_i \mathbb{I}(A_{1i} = d_{1i} | x_{1i}) \mathbb{I}(A_{2i} = d_{2i} | x_{2i})\right)$$

$$\tilde{Y}_{\text{IPTW-rob}} = \frac{1}{n} \sum_i \frac{Y_i \mathbb{I}(A_{1i} = \hat{d}_{1i} | x_{1i}) \mathbb{I}(A_{2i} = \hat{d}_{2i} | x_{2i})}{P(A_{1i} = \hat{d}_{1i} | x_{1i}) P(A_{2i} = \hat{d}_{2i} | x_{2i})} \left(\sum_i Y_i \mathbb{I}(A_{1i} = d_{1i} | x_{1i}) \mathbb{I}(A_{2i} = d_{2i} | x_{2i})\right) \frac{1}{P(A_{1i} = d_{1i} | x_{1i}) P(A_{2i} = d_{2i} | x_{2i})} \left(\sum_i Y_i \mathbb{I}(A_{1i} = d_{1i} | x_{1i}) \mathbb{I}(A_{2i} = d_{2i} | x_{2i})\right)^{-1}$$

Here, we assume that $A_1 | X_1$ is independent of $A_2 | X_2$ so that each propensity score can be separately fitted by a multinomial logistic model. This assumption could be dropped by fitting multivariate multinomial model, but for convenience, we keep this assumption in the following analysis. To the author’s knowledge, this may be the first proposal of the double robust estimation of $\mathbb{E}(Y | A_1, A_2)$ in a dynamic treatment regime among current literature.

### 3.2.1 Data Set

The data is again cleaned by eliminating uncommon treatment IDs, using similar criteria to that in Section 3.1.1. The remaining treatment IDs for this section are given in Table 11.

<table>
<thead>
<tr>
<th>Treatment ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
</tr>
<tr>
<td>00010</td>
</tr>
<tr>
<td>01000</td>
</tr>
<tr>
<td>10000</td>
</tr>
<tr>
<td>11000</td>
</tr>
</tbody>
</table>

Table 11: Remaining treatment IDs after data cleaning.

### 3.2.2 Model Selection

Random Forest and backward selection of a logistic regression model are again utilized to detect the most important covariates for the analyses in stage 2. Table 12 summarizes the top variables selected using these two methods, respectively.
Table 12: Covariates selected by Random Forest and backward selection.

We notice that all the covariates selected by the logistic regression method are a subset of covariates selected by the random forest method. We keep all of the covariates selected by both methods in addition to the age at diagnosis and the Dr Visits in month 1. The final covariates we use in our model can be seen in Table 13. It is worthwhile to mention that when applying those important covariates in stage 2 and stage 1, we also include the interaction terms between the Treatment group in month 2 and month 1 respectively with all these other covariates. Moreover, the propensity scores of treatments are obtained by fitting a multinomial logistic model in stages 1 and 2, using covariates listed on Table 10 and Table 13, respectively.

Table 13: Final list of covariates for this model.

<table>
<thead>
<tr>
<th>Final List of Covariates for the Two Decision Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment group in month 2</td>
</tr>
<tr>
<td>Glucocorticoid or painkiller in month 1</td>
</tr>
<tr>
<td>Age at diagnosis</td>
</tr>
<tr>
<td>Prescriptions in month 1</td>
</tr>
<tr>
<td>Dr Visits in month 1</td>
</tr>
<tr>
<td>Treatment group in month 2</td>
</tr>
<tr>
<td>OP visits in month 1</td>
</tr>
</tbody>
</table>

4 Computational Results

4.1 A One-Decision Model

As discussed in Section 3.1.2., we use the logistic regression model to estimate the expected outcome, and use the corresponding test data to predict the outcome values for each patient under each treatment group observed. We then obtain the optimal rule (the optimal treatment group) for each patient based on the minimum predicted outcome.

Ultimately the set of optimal treatment groups we observed for this stage are referenced in table 7. The optimal treatments for patients belonging to different categories based on gender and comorbidity status, can be seen in Figure 4.
Figure 4: This figure displays the optimal treatments for patients who do not have COPD and Angina. For females, if the patients do not have skin ulcers, the treatment method consists of 1 TNF for all ages, regardless of whether they have hypertension or not. If the female patients have Skin Ulcers, the treatment changes to 1 MTX at 46 years old for patients without hypertension, and at 53 years old for patients with hypertension. For male patients, if they do not have skin ulcers, the treatment changes from 1 TNF to 1 DMARD for hypertension patients at around 21 years old. For those patients who do not have hypertension, the treatments change twice. At 42 years old, the treatment changes from one TNF to one DMARD, and at 50 years old, the treatment changes from one DMARD to a combination of one DMARD and one MTX. If the male patients have Skin Ulcers, the treatment is one DMARD for every patient who has hypertension. Whereas treatment changes from one TNF to one MTX at 19 years old for patients without hypertension.

Figure 5: This figure displays treatments for patients who have COPD and Angina. For all female patients without skin ulcers, the treatment is 1 TNF. For the female patients who have Skin Ulcers, the treatment changes from 1 TNF to 1 MTX at 53 years old with hypertension, and the treatment changes from 1 TNF to 1 MTX at 46 years old without hypertension. For male patients, if they do not have Skin Ulcers, the treatment changes from 1 TNF to 1 DMARD at 40 years old with hypertension, and the treatment changes to 1 MTX at 43 years old without hypertension. If they have Skin Ulcers, the treatment changes from 1 TNF to 1 DMARD at 15 years old, and then changes to 1 MTX at 39 years old with hypertension. For those male patients who do not have hypertension, the treatment changes from 1 TNF to 1 MTX at 17 years old.
From Figures 4 and 5, we see that there are two optimal treatment groups in female patients: one TNF and one DMARD. For the female patients who do not have Skin Ulcers, the treatment with one TNF is suitable for all of them. And for those female patients who have Skin Ulcers, the treatment changes to one DMARD at 45 years old. The male patients treatments are more diverse. For the male patients who have Skin Ulcers, the treatments change at very young ages, approximately 15 to 20 years old. The treatments change to one DMARD or one MTX based on whether they have hypertension or not. The most interesting group for male patients is the patients who are having hypertension and Skin Ulcers; the treatment for this group stays the same no matter what is the age of patient, which is 1 DMARD.

We also compare the different estimators for the expected mean outcome of our two-month model that we discussed at Section 3.1 in Table 14. The estimated expected outcome under the estimator that does not account for bias is inaccurate. The more reliable estimators are the double robust estimator and the adjusted estimator that incorporates the IPTW. In either case, it is evident that the estimated expected value of outcome under the optimal treatment groups is smaller than that under the observed treatment groups.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Estimated Expected Value of Outcome</th>
<th>Observed Mean Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without IPTW</td>
<td>0.141</td>
<td>0.186</td>
</tr>
<tr>
<td>Adjusted (With IPTW)</td>
<td>0.151</td>
<td>0.190</td>
</tr>
<tr>
<td>Double robust</td>
<td>0.150</td>
<td>0.184</td>
</tr>
</tbody>
</table>

Table 14: Comparison of estimates of the Expected Mean Outcome of Two-Month Model and the Observed Mean Outcome

### 4.2 A Two-Decision Model

As discussed in Section 3.2, we now obtain the individualized optimal treatment groups for two stages (stage 1 and stage 2) incorporating the dynamic aspect of the disease. The set of optimal decision rules we obtain for stage 1 can be referred to in table 7 and the set of optimal decision rules for stage 2 can be referred to in table 11. As in section 4, we visualize the optimal treatments for patients, based on the covariates of the respective models.
Figure 6: The figure represents patients who are not diagnosed with COPD and Angina. We can see for all female patients, whether they have hypertension or Skin Ulcers, they can be prescribed 1 TNF. However, the male patients treatment groups are more diverse. For the male patients who do not have Skin Ulcers, their treatment groups change at age 40 when they are first diagnosed with RA. If they do not have hypertension, the treatment changes from 1 TNF to 1 MTX at age 40. But for the patients who have hypertension, the treatments need to be changed twice. At age 40, the treatment changes from 1 TNF to 1 DMARD. At 60, the treatment changes from 1 DMARD to 1 DMARD and 1 MTX. And for male patients who have Skin Ulcers, the treatment changes from 1 TNF to 1 DMARD if he is diagnosed with RA at around 64 years old with hypertension. Without hypertension, the treatment changes from 1 TNF to 1 MTX at 56 years old.

Figure 7: This figure represents the patients who have COPD and Angina. For female patients, if they have Skin Ulcers, regardless of hypertension and age at diagnosis, they are treated with 1 TNF. If they do not have Skin Ulcers, the treatment will change to 1 DMARD at 60 years old without hypertension, and at 55 years old with hypertension. For male patient, 1 DMARD and 1 MTX is the most common. If the patients do not have Skin Ulcers, the treatment changes from 1 TNF to 1 MTX at 26 years old without hypertension, and the treatments change from 1 TNF to 1 DMARD at 20 years old with hypertension. If they have Skin Ulcers, the treatments change from 1 TNF to 1 DMARD at 43 years old with hypertension, and from 1 TNF to 1 MTX at around 43 years old without hypertension.

From Figures 6 and 7, the treatment group appears straightforward for women. That is, 1 TNF will work
for most of the female patients. Only a small group of female patients who have COPD and Angina but no Skin Ulcers, use a different treatment: 1 DMARD. However, the treatments for male patients are slightly more complicated. For these two situations, there are four treatments being used: 1 TNF, 1 MTX, 1 DMARD and 1 MTX with 1 DMARD. On average, 1 DMARD treatment and 1 MTX treatment are more suitable for older patients. Except for the male patients who have COPD and angina, the treatment change begins earlier, at approximately 20 years old, for the patients who have hypertension, and at 26 years old for the patients who do not have hypertension. The treatment consisting of 1 MTX and 1 DMARD is only recommend for a small group of male patients who only have hypertension.

Similar to the Section 4.1, we compare estimates of the expected value of outcome using the set of optimal decision rules with the observed mean outcome in Table 15. As expected, we can see that the former is smaller than the latter. Note how we have once again incorporated two different estimators; the adjusted and the double robust estimators.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Estimated Expected Value of Outcome</th>
<th>Observed Mean Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjusted (With IPTW)</td>
<td>0.040</td>
<td>0.122</td>
</tr>
<tr>
<td>Double robust</td>
<td>0.059</td>
<td>0.112</td>
</tr>
</tbody>
</table>

Table 15: Comparison of estimates of the Expected Mean Outcome of Three-Month Model and the Observed Mean Outcome

5 Summary and Future Work

We studied a framework for one and two decision problems in which the rewards are the expected probability of implementing treatment regimes. We proposed applicable Q-learning algorithm for precision medicine of RA patients and novel measurements for expected outcome to evaluate the effect of therapy. The work as presented can be extended to real-world multi-stage decision problems on observational data.

Interesting results are derived from our analysis, which might be instructive for future research work on RA. For example, in our one decision-two month model, we find that there are only two optimized treatment groups in female patients: for the female patients without skin ulcers, the treatment 1 TNF is suitable for all of them, and for the others with skin ulcers, the treatment changes to 1 DMARD at 45 years old. Similarly, we can see in the two decision model that all of the female patients, regardless of if they have hypertension or Skin Ulcers, are optimally prescribed 1 TNF. In contrast, the treatments of male patients are more diverse in both models. Finally, older patients appear to be more suitable for traditional DMARD treatments.

Despite the advantages of the our proposed method, there are theoretical limitations. First, the model we fit relies on the specified likelihood, which is not robust if parametric assumption is violated in practice. It will become more serious when there are more stages to deal with. However, this issue could be properly solved by applying a semi-parametric approach, such as generalized estimating equation method, or non-parametric one, such as random forest, to fit data. Second, the model selection procedure in multiple stages of the decision-making is tricky, as the outcomes in different stages are not observed until the last stage. Selecting the important covariates in each stage remains an open research question both in theory and application. Third, we discard the patients who have only two months of observational data when we implement our 2 stage Q-learning algorithm, due to very low censoring rate. If we were to expand our model beyond 2 stages, we might lose some information if the censoring is not completely random. Thus, some techniques should be applied to deal with this issue, especially when the censoring rate is high. The standard error for predicted expectation $\hat{Y}$ is also of interest, which might be obtained by asymptotic derivation or bootstrap methods. Last but not the least, the global response, i.e. the overall effect of treatment combinations during the study, is of greater interest to researchers than the outcome from the last stage alone. Thus, finding a more efficient algorithm is also an open question.

There are also practical limitations for our model that should be dealt with in future work. For example, the outcome we defined in this study might not be the best indicator for the treatment. More specific information, such as clinical information, is needed to correctly identify the response and covariates in the analysis. For example, levels of C-reactive protein (CRP) from a blood test, might be a better outcome for predicting the effectiveness of the therapy.
References


Estimation of Coastal Hydrodynamics with Machine Learning

Taylor Baumgard, Cassidy Krause, Evan North, Ramchandra Rimal, Jacob Roth, Arsenios Tsokas

Mentors: Ty Hesser, Matthew Farthing, Arvind K. Saibaba

Abstract

Near-shore features like wave breaking height, wave breaking period, wave breaking direction, beach slope, and wave breaking location are of prime importance for any coastal community or institution that deals with beach landing crafts. Estimation of these quantities often utilizes bathymetry, tide, and sensor data, and traditional methods model near-shore dynamics with physical wave-transformation models. This paper explores the possibility of replacing a physical model with a model trained by machine learning methods on a mix of traditional and synthetic datasets. Off-shore and bathymetry data were collected from the United States Army Corps of Engineers Field Research Facility at Duck, North Carolina which were used to model near-shore wave dynamics. Various supervised machine learning methods, including support vector regression (SVR) and neural network architectures (NNs) were tested. Finally, a range of experiments were performed to explore the validity of using machine-learning models as a potential surrogate for a physical model.

1 Introduction

Understanding near-shore wave conditions is a vital aspect of bringing ocean craft ashore safely. Whether planning logistics for delivery of goods or supporting near-shore military operations, accurate and timely predictions are important parts of understanding the risks associated with a particular task. In order to plan for efficient and safe operations, defense agencies, transportation companies, and similar agencies want to know the future wave characteristics. For example, a ship coming to port or leaving a port relies on information of wave height, wave direction, and wave length to ensure the safety of the supplies and people that may be on board.

It is convenient and intuitive to characterize a wave as a vector whose direction is expressed in terms of degrees relative to shore and whose magnitude is represented by wave height. Accordingly, the traditional modeling approach has been to use hi-fidelity, physics-based models. These models use boundary conditions collected from off-shore sensors to deterministically model near-shore conditions. Currently, such approaches utilize high-performance computers, yet still may require hours to perform multiple day forecasts. This limits their effectiveness in the field or in time-sensitive situations and motivates the search for methods whose primary computational load may be performed prior to use in a time-constrained environment.

1 Mathematics, University of Texas, Arlington
2 Mathematics, University of Kansas
3 Mathematics, North Carolina State University
4 Mathematics, University of Central Florida
5 Statistics, University of Chicago
6 Center of Applied Optimization, Industrial and Systems Engineering, University of Florida
7 USACE
8 USACE
9 Mathematics, North Carolina State University
In the remainder of the paper, we explore an alternative modeling methodology and study the viability of applying machine learning techniques to "learn" low-fidelity physics of near-shore hydrodynamics. It is organized as follows: Section 2 describes the available data and formulates the problem explicitly; Section 3 describes the model; Section 4 reviews the machine learning framework; Section 5 discusses the results; Section 6 summarizes the findings of this project and highlights some avenues of future work.

2 The Data

All data used in this study were collected at the U.S. Army Corps of Engineers (USACE) Field Research Facility (FRF) in Duck, NC. This project focuses on five specific kinds of data: significant wave height ($H_s$), wave period ($T$), incoming wave angle ($\theta$), bathymetry, and tide. The three wave measurements are measured by an AWAC-11M sensor (see Figure 1), which is a bottom mounted acoustic sensor in 11 meters depth or approximately 1150 m offshore. Incoming wave angle is measured in degrees from true North, and averaged over the course of the hour, and wave period is reported in seconds after being averaged over approximately half and hour. Significant wave height is the average height of the top $\frac{1}{3}$ of waves across a half-hour.

Bathymetry data is gathered with a Coastal Research Amphibious Buggy (CRAB), a 10 m tall vehicle that can survey the bathymetry by driving along the (coastal) ocean floor, recording the depth along the way. These surveys are taken in profiles that extend perpendicularly from the coast, as can be seen in Figure 2. Based on the sitting location of AWAC-11, the bathymetry surveys being used are from the $y = 956$ transect ($y$ refers to the local coordinate system at FRF), and serve as the underwater topography for the one-dimensional model discussed in Section 3. For a more complete description of how bathymetry data is gathered and reported, see [9].

To use the data in our models, we had to make some modifications which we now describe. The bathymetry data was only reported every 12 meters in the surveys, and not every survey started and ended in the same location. In order to standardize our data, every bathymetry survey we sent to the model started with an artificially placed (when necessary) 3m elevation reading at $x = 50$. The model needed data every meter, so a linear interpolant was used to fill in to the first recorded data value. Cubic splines were used to interpolate all values within the recorded dataset, and another linear interpolant was used between the last measured value and the AWAC-11m sensor ($x = 1150$, $depth = 11$).

Figure 1: AWAC-11m sensor (left) and CRAB (right), the primary instruments for gathering data used in these models.
Figure 2: Two examples of depth conditions off the shore of FRF. The solid red contour represents the approximate coastline, while the red dotted line represents the transect along which our sensor is located.

3 Physical Model

Since the phenomenon of shallow water waves is nonlinear and ever-changing, they are complicated to model. Although the existing methods are accurate, they are fiscally and computationally expensive. Rather than use the existing methods, we implement a low-dimensional, linear model to predict the wave height.

There are several forces to consider in wave modeling. One of the most important factors is the effect of the sea floor, or the bathymetry \(b\), on the waves themselves. As the waves encounter shallower water, the sea floor starts to interact with the waves: the shallower the water, the higher the waves become. This effect is called shoaling.

As the waves near shore and get higher, the potential energy increases. This is expressed in the equation

\[
E = \frac{1}{8} \rho g H^2,
\]

where \(E\) is the energy of the wave, \(H\) is the wave height, \(\rho\) is the density of the water, and \(g\) is a gravitational constant. In our model, we assume \(g\) and \(\rho\) to be 9.8 m/s\(^2\) and 1000 kg/m\(^3\), respectively.

The water depth \(h\) (determined by \(h = b + t\) for bathymetry \(b\) and tide \(t\)) is a main factor in determining wave height. Shallow water also has a direct effect on the wave number \(k\), as seen in the dispersion relation:

\[
\sigma^2 = \left(\frac{2\pi}{T}\right)^2 = gk \tanh(kh).
\]

The wave number \(k\) is inversely proportional to the wavelength \(L\). Knowing the wavelength gives us the wave celerity \(C = L/T\). Putting all of this together, we see that shoaling not only increases the height of the waves, but it also slows the wave down.

As the waves travel, they interact with each other and form wave groups. We can calculate not only individual wave celerity, but also the wave group celerity, \(C_g\). Like individual wave celerity, wave group celerity is also a function of wavelength, and is given by

\[
C_g = \frac{1}{2} \frac{L}{T} \left[ 1 + \frac{4\pi h}{L} / \sinh\left(\frac{4\pi h}{L}\right) \right].
\]

Another factor to consider is the angle at which the wave approaches the shore, which is measured in degrees from perpendicular to the shore. This angle changes as the wave approaches the shore,
and this phenomenon is known as wave refraction. As discussed above, when a wave heads toward the shore and encounters a higher bathymetry, shoaling will begin to occur. However, assuming a bathymetry parallel to the shore and a wave that is traveling at a non-zero angle, one portion of the wave will encounter this shallower bathymetry before the rest of the wave. This will cause that first portion to slow down before the other part, which will turn the angle of the wave. Snell’s law gives us this relationship explicitly. If \( C_i \) and \( C_{i+1} \) are different speeds of the wave as it is slowing down, and \( \theta_i \) and \( \theta_{i+1} \) are the corresponding angles, then we have

\[
\frac{\sin(\theta_{i+1})}{\sin(\theta_i)} = \frac{C_{i+1}}{C_i}.
\]

Putting all of this together, we obtain the one-dimensional energy flux conservation equation

\[ \frac{d}{dx} (EC_g \cos(\theta)) = \delta(x), \]  

where \( \delta \) is a dissipative function that represents the breaking point of a wave.

There are several different choices of the \( \delta \) function; we chose to use a model specified in [2]:

\[
\delta = \frac{1}{4h} B \rho g f H_{rms}^3 \left[ \left( R^3 + \frac{3}{2} R \right) \exp(-R^2) + \frac{3}{4} \sqrt{\pi} (1 - \text{erf}(R)) \right],
\]

where \( H_b = \gamma h, H_{rms} = \sqrt{2} H \) is the root-mean-square wave height, \( \text{erf} \) is the error function, and \( R = H_b/H_{rms} \). This \( \delta \) function also uses the constants \( f \) for the representative temporal frequency; a function of the fraction of foam on the face of the breaker, \( B \), which controls the level of energy of dissipation and is of the order of one; and the tuning parameter \( \gamma \).

With these equations as our basic model, we gather boundary condition data that includes the initial angle, wave height, and period; as well as the depth \( h \) throughout the spatial area of interest. Using these boundary conditions, we implement a Forward Euler method to solve for a steady-state solution of the wave height.

With this choice of \( \delta \) and the standard Forward Euler implementation, we would have to use a nonlinear solver, since \( H \) is the quantity of interest on the LHS, yet it is also involved in the computation on the RHS (specifically, in calculating \( H_{rms} \)). Since our primary constraint is efficiency, we avoid this complication by first assuming that \( \delta = 0 \), and doing a Forward Euler step with this modified equation. Each step yields \( \hat{E} \), from which we get a pseudo-\( H \) value. This value is then used to compute \( \delta \). This correction is applied to \( \hat{E} \), and we obtain the wave height \( H \), as desired.

Below, we present numerical evidence that verifies and validates the performance of the numerical method.

### 3.1 Verification

To ensure that the Forward Euler scheme accurately solves the differential equation of interest, we perform model verification. Model verification is a mathematical and software issue whereby known solutions are tested against the model’s output. Our strategy is the following: We specify the energy flux \( F(x) := E(x) C_g(x) \cos \theta(x) \) as a quadratic model. More specifically, we take \( F_{\text{model}}(x) := \alpha + \beta x + \gamma x^2 \) for appropriate parameters \( \alpha, \beta, \gamma \), which are chosen to fit a quadratic model based on real data. By plugging this true solution into [1], the right hand side of the energy flux conservation equation is \( \delta_{\text{model}}(x) = 2\gamma x + \beta \). This delta function is then used as the input to [1], and the exact solution is used to track the accuracy of the numerical solution. The error is measured in the \( \ell_2 \)-norm, observing that the Forward Euler scheme recovers the intended trajectory as seen below. Sample verification results are provided in Figure 3.
We observe first order accuracy in the first case when wave direction is aligned with the shore and the 1m discretization is compared with a coarser 2m discretization. However, when wave direction is not directly aligned with the shore, we observe that the convergence becomes sublinear (order $\approx 0.5$) as shown below:

Figure 4: Physical model verification with $\theta = \pi/3$. As can be seen, the Forward Euler method captures the model solution accurately to sublinear order.

### 3.2 Validation

In addition to making sure that the Forward Euler method is implemented correctly, it is also important to check that our simple, linear model sufficiently matches our dataset. As explained in Section 2, the sensor data consists of wave height, wave angle, and wave period. These sensors are placed at depths of 11 meters, 8 meters, and 6 meters. We use the 11 meter sensor data as boundary conditions for the model, which then computes several values, including wave height, wave angle, and wave period, at one meter intervals towards the shore. Since these three attributes are a part of both the model output and the dataset, we are able to compare the model output values with the sensor data.

To determine how well the model tracks true conditions, we compared the modeled and measured values of wave height at 6m depth during July 2016 and October 2015, as shown in Figures 5 and 6, respectively. July 2016 was a calm month; under these conditions, the physical model predicted wave height within $\approx 30cm$. On the other hand, October 2015 was a stormy period with waves nearly
twice the size as in July 2016. In this case, the model performed surprisingly well and predicted the wave height within a few cm. Surprisingly, our model did relatively well in both of these conditions. In particular, the absolute errors were around 27cm and 4cm for July and October respectively, and the relative errors were 40cm and 3cm.

Figure 5: Physical model validation; Left: predictions, Right: relative error.

Figure 6: Physical model validation; Left: predictions, Right: relative error.

4 Machine Learning

In our research, we chose to use supervised machine learning techniques including support vector machines and neural networks. We chose to use these two particular machine learning techniques because of some previous research that explores forecasting wave conditions with machine learning [5].

In supervised machine learning, one is presented with data of the form $\{(X_i, y_i)\}_{i=1}^S$, where $X_i \in \mathbb{R}^N$ are the given inputs and $y_i \in \mathbb{R}^K$ are the given outputs and $S$ is the number of training samples. Using this data, the goal of the machine learning algorithm is to estimate the function which maps the inputs to the outputs. That is, the goal is to recover the function $f(\cdot)$ that satisfies $y = f(X)$. Different machine learning techniques have different ways of estimating $f$ and lead to different algorithms. The end goal of supervised machine learning is to train the algorithm to approximate the mapping function in a way that when given a new set of input data, it can predict the output variables for that particular data.
Here we used training data $X = (H_0, T_0, \theta_0, h)$ and corresponding output $y = (H_b, T_b, \theta_b, x_b)^T$ where $H_0, T_0, \theta_0$ are scalar inputs and $h \in \mathbb{R}^N$ where $N$ is the length of the discretized bathymetry data. Table 1 summarizes the meaning of each variable in the input and the outputs. Note that $h$ here refers to the bathymetry depth which is available to us in 1m intervals of size 1100. Therefore, the dimension of the input vector is $n = 1103$, whereas the dimension of the output vector is 4.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_0$</td>
<td>Initial Wave height</td>
<td>m</td>
</tr>
<tr>
<td>$T_0$</td>
<td>Initial wave period</td>
<td>s</td>
</tr>
<tr>
<td>$\theta_0$</td>
<td>Initial angle</td>
<td>radians</td>
</tr>
<tr>
<td>$h$</td>
<td>Bathymetry depth</td>
<td>m</td>
</tr>
<tr>
<td>$H_b$</td>
<td>Wave breaking height</td>
<td>m</td>
</tr>
<tr>
<td>$T_b$</td>
<td>Wave breaking period</td>
<td>s</td>
</tr>
<tr>
<td>$\theta_b$</td>
<td>Wave breaking angle</td>
<td>radians</td>
</tr>
<tr>
<td>$x_b$</td>
<td>Wave breaking location</td>
<td>m</td>
</tr>
</tbody>
</table>

Table 1: Description of the various physical quantities that form the inputs and outputs of the machine learning models.

We chose to implement two different supervised machine learning methods: support vector regression and neural networks. To implement these methods, we used two different machine learning packages: sklearn [11] for SVR, whereas for NN, we used TensorFlow [1]. In addition, we made extensive use of other Python packages such as NumPy [10] and SciPy [6].

In this section, we will first explore how support vector machines work and the associated parameters. After that we explain how neural networks work and how they were used in our research. Following these sections are cross validation sections for the respective machine learning techniques. Details on the machine learning tools are given in [3].

4.1 Support Vector Regression

The support vector machine (SVM) is a supervised learning tool used for classification and regression analysis. Classification is about predicting a label, whereas regression involves predicting a quantity. The goal of the SVM is to develop classifiers based on training data that will correctly classify the test observations by using its feature measurements. SVM is allowed to enlarge the feature space by using kernels, a computationally efficient approach, in order to accommodate a non-linear boundary between classes.

Support vector regression (SVR) is an extension of SVM for regression. The use of kernels is important in the implementation of SVR where a kernel is a function that measures the similarity of two observations. There are several different kernels that can be implemented, such as linear, polynomial, and radial kernels. For further details, we address the reader to [12]. Support Vector Regression has been used efficiently in predicting wave height as in [8]. For this particular research, we used a radial kernel which takes the form

$$K(X_i, X_{i'}) = \exp \left( -\gamma \| (X_i - X_{i'}) \|^2 \right).$$

A radial kernel has very local behavior, meaning if a test observation is far in Euclidean distance from a training observation, then the training observation will basically play no role in the predicted class label for the test observation. In a radial kernel, there are two parameter choices: $C$ and $\gamma$. First, $\gamma$ is a positive constant. A small $\gamma$ means the class of the support vector $X_{i'}$ will have an
influence on deciding the class of the vector $X_i$. A large $\gamma$ implies the variance is small and the support vector $X_\nu$ does not have widespread influence. A large $\gamma$ leads to high bias and low variance models. Second, $C$ is a nonnegative tuning parameter that determines the number of violations to the margin that we tolerate. In the SVM, a hyperplane is chosen to separate most of the training observations into two different classes, and it may misclassify a few observations. Then, margins are created on either side of this hyperplane. Observations may fall outside the margins, inside the margins, or on the margins. In the SVR, for example, if $C = 0$, then we would not allow any violations to the margin. As $C$ becomes large, we tolerate more violations to the margin, and, thus, our margin widens. It follows that when $C$ decreases, the margin narrows. A small $C$ means the resulting classifier will have low bias and high variance. In order to find the best parameters to minimize the mean squared error, refer to Section 4.3.1.

4.2 Neural Networks

The main idea of neural networks (NN) is to extract the linear combinations of inputs as the derived features and then model the target as a non linear function of features. Neural networks were first developed as models of the human brain where each unit represents a human neuron and the connection represent synapses. An artificial neural network is based on a collection of connected units, or nodes, known as artificial neurons which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. It is a powerful machine learning technique that has wide applications in many area such as character recognition, image compression, and stock market prediction [3].

Neural network techniques apply to both classification as well as regression problems. Neural networks have been used successfully in predicting ocean wave heights in [7]. In this project, we are implementing neural networks in the regression context for the purpose of predicting near shore hydrodynamics. A neural network is a two stage regression (or classification model) typically represented by a network diagram. An example of this diagram is given in Figure 7.

![Figure 7: A schematic representation of the neural network architecture with one hidden layer.](image)

In Figure 7, $X_i, 1 \leq i \leq n$ are the original inputs, $Z_j, 1 \leq j \leq M$ are hidden units and $Y_k, 1 \leq k \leq K$ are the outputs. As was mentioned in the start of this section, the number of inputs $n = 1103$ and the number of outputs is $K = 4$. The first layer consists of observed values known as input layer, values $Z_j, 1 \leq j \leq M$ not observed directly are known as hidden values and the whole second layer
is known as hidden layer. The last layer, consisting of output values, is known as the output layer. There can be several hidden layers in the network depending upon the problem. Then, the target $Y_k$ can be modeled as follows:

$$Z_j = \psi(\alpha_{0j} + \alpha_{0j}^T X), \quad j = 1, \ldots, M$$

$$\Phi_k = \beta_{0k} + \beta_{0k}^T Z, \quad k = 1, \ldots, K$$

$$f_k(X) = g_k(\Phi), \quad k = 1, \ldots, K$$

where $Z = (Z_1, Z_2, \ldots, Z_M)$, $\Phi = (\Phi_1, \Phi_2, \ldots, \Phi_K)$, $f_k$ is the function to be learned in the data-input space and $g_k$ is the function learned in the feature-space. The function $\psi$ is called the activation function in the literature and the popular choices are a sigmoid function, rectified linear units (ReLU), hyperbolic tangent function, and a radial basis function. In our experiments, we have used a ReLU function.

Neural networks have unknown parameters known as weights. In the above model, the parameters $\alpha_{0j}, \alpha_{0j}, 1 \leq j \leq M$ and $\beta_{0k}, \beta_{0k}, 1 \leq k \leq K$ are the weights, which are to be learned to fit the model. The function $g_k$ does the final transformation of the vector $\Phi$. In the regression context, the function $g_k$ is usually chosen as $g_k(\Phi) = \Phi_k$. The suggestion in [3] was to start the model with a few layers and parameters and increase the size until diminishing returns on the validation loss are observed.

### 4.3 The Need for Cross Validation

As discussed above, both of these machine learning methods work by splitting data into a training set, $X$, which is accompanied with a target $y$, and a testing set $\tilde{X}$ accompanied by its corresponding target $\tilde{y}$. The machine learning method creates a regression rule that relies on the training data used, and one must be aware of the risk of overfitting the machine learning algorithm to the training data. To verify this, one must incorporate cross validation (CV).

When validating a machine learning algorithm, we split the training set $\{X_i, y_i\}_{i=1}^N$ into a subset for training, and a subset for testing. Since we know the true value of $\tilde{y}$, we can compare it with the predicted output of the machine learning algorithm, and calculate the mean squared error. However, a low mean squared error does not validate a machine learning method by itself. Since the regression rule depends on the training data, using a different training and testing set could result in a completely different mean squared error. To test for this sort of robustness, we apply a stratified CV analysis.

In CV, the training set with its known target is split into $k$ equal groups. (A common choice is $k = 10$, and that is the value we use here.) The first of these groups is set aside to be the test set, and the remaining $k-1$ groups are used as the training set. Note that the test data is not simultaneously used as the training data. The predictions of the test set are compared with the known target values, and the mean squared error is calculated. This process is repeated $k$ times, each time choosing a different group to be the test set, and using the remaining $k-1$ groups as the training set. The mean of each of these mean squared errors is the metric we use for evaluating the accuracy of the machine learning algorithms.

#### 4.3.1 Cross Validation for SVR

One way we can use cross validation is to tune the parameters of our machine learning method. SVR has three parameter choices: $C$, $\gamma$, and the choice of the kernel, which are discussed in detail in Section 4.1. To tune the method, we create a grid space of reasonable parameter values, and calculate the mean squared error for each of them. The combination of parameter values that gives the lowest mean squared error will be the parameters we use in our model.
Table 2 shows the mean squared errors for the wave breaking height, using different values of $C$ and $\gamma$ with a radial basis kernel SVR algorithm.

<table>
<thead>
<tr>
<th>$C$ \ $\gamma$</th>
<th>$1e-8$</th>
<th>$1e-7$</th>
<th>$1e-6$</th>
<th>$1e-5$</th>
<th>$1e-4$</th>
<th>$1e-3$</th>
<th>$1e-2$</th>
<th>$1e-1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1e-2$</td>
<td>.902</td>
<td>.902</td>
<td>.902</td>
<td>.902</td>
<td>.901</td>
<td>.890</td>
<td>.806</td>
<td>.588</td>
</tr>
<tr>
<td>$1e-1$</td>
<td>.902</td>
<td>.902</td>
<td>.901</td>
<td>.890</td>
<td>.788</td>
<td>.788</td>
<td>.203</td>
<td>.127</td>
</tr>
<tr>
<td>$1e0$</td>
<td>.902</td>
<td>.902</td>
<td>.901</td>
<td>.890</td>
<td>.785</td>
<td>.136</td>
<td>.019</td>
<td>.049</td>
</tr>
<tr>
<td>$1e1$</td>
<td>.902</td>
<td>.901</td>
<td>.890</td>
<td>.785</td>
<td>.130</td>
<td>.008</td>
<td>.008</td>
<td>.031</td>
</tr>
<tr>
<td>$1e2$</td>
<td>.901</td>
<td>.890</td>
<td>.785</td>
<td>.128</td>
<td>.007</td>
<td>.004</td>
<td>.007</td>
<td>.031</td>
</tr>
<tr>
<td>$1e3$</td>
<td>.890</td>
<td>.785</td>
<td>.128</td>
<td>.007</td>
<td>.004</td>
<td>.004</td>
<td>.007</td>
<td>.031</td>
</tr>
<tr>
<td>$1e4$</td>
<td>.784</td>
<td>.128</td>
<td>.007</td>
<td>.004</td>
<td>.004</td>
<td>.005</td>
<td>.007</td>
<td>.031</td>
</tr>
<tr>
<td>$1e5$</td>
<td>.146</td>
<td>.006</td>
<td>.004</td>
<td>.004</td>
<td>.004</td>
<td>.007</td>
<td>.031</td>
<td>.186</td>
</tr>
<tr>
<td>$1e6$</td>
<td>.059</td>
<td>.002</td>
<td>.004</td>
<td>.004</td>
<td>.004</td>
<td>.007</td>
<td>.031</td>
<td>.186</td>
</tr>
<tr>
<td>$1e7$</td>
<td>5.12</td>
<td>.714</td>
<td>.004</td>
<td>.004</td>
<td>.004</td>
<td>.005</td>
<td>.007</td>
<td>.031</td>
</tr>
<tr>
<td>$1e8$</td>
<td>461</td>
<td>123</td>
<td>.004</td>
<td>.004</td>
<td>.004</td>
<td>.007</td>
<td>.031</td>
<td>.186</td>
</tr>
</tbody>
</table>

Table 2: Mean squared errors of wave breaking height for varying parameters. The best combination of parameters in this case is $\gamma = 10^{-6}$ and $C = 10^7$.

The parameters that minimize the mean squared error for predicting wave height in July, using the radial basis function kernel are $\gamma = 10^{-6}$ and $C = 10^7$. Doing a similar analysis finds the best parameters for the other features as well. The tuned parameters and their respective relative mean errors for each of the target features is shown in Table 3.

<table>
<thead>
<tr>
<th></th>
<th>$H_b$ Error</th>
<th>$\theta$ Error</th>
<th>$X_b$ Error</th>
<th>$T_b$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Tuning</td>
<td>0.133</td>
<td>0.280</td>
<td>5.254</td>
<td>0.004</td>
</tr>
<tr>
<td>Tuning for $H_b$</td>
<td>0.103</td>
<td>2.756</td>
<td>9.266</td>
<td>0.004</td>
</tr>
<tr>
<td>Tuning for $T_b$</td>
<td>0.147</td>
<td>0.771</td>
<td>13.543</td>
<td>0.002</td>
</tr>
<tr>
<td>Tuning for $\theta$</td>
<td>0.126</td>
<td>0.262</td>
<td>10.15</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Table 3: Mean relative error of target output with different optimal tuning parameters

Each row in Table 3 is the result of the optimal parameters for a specific output. In the second row, for example, the mean relative error for the breaking wave height is minimized, but at the cost of a significant increase in $\theta$ and $X_b$. Since we don’t want to sacrifice the accuracy of any of the outputs ($H_b$, $\theta$, $X_b$, and $T_b$), we decided to continue to use the default settings from sklearn of $\gamma = 0.1$ and $C = 1e3$. It is important to note that these parameters are tuned for SVR on the month of July, 2016, with a static bathymetry. Using different training and testing sets would require tuning a different set of parameters. Future work would include finding these other optimal tunings.

### 4.3.2 Cross Validation for Neural Networks

A neural network with one hidden layer and finite number of neurons can approximate any complex function, as shown by works like [4]. Furthermore, the particular choice of activation function does not affect a network’s learning ability. However, the number of neurons required to approximate a function might be prohibitively large. Also, a single layer network might not be able to capture
all patterns in the dataset unless the sample captures all possible realizations of the underlying generative process. On the downside, networks with many hidden layers would require much more time for training; the same argument holds for recurrent neural networks.

![Box plots of Mean Absolute Errors for the July dataset.](image)

Figure 8: Box plots of Mean Absolute Errors for the July dataset.

In the framework of this study, we decided to fix the number of hidden layers to two and experiment with the number of neurons in every layer. We performed repeated 10-fold cross validation for all the proposed neural architectures, examining the mean absolute error on the (normalized) test set. Choosing five neurons in both hidden layers proved to be the optimal choice, balancing accuracy in the predictions and simplicity in architecture. We report the distribution of the mean average error in Figures 8 and 9.
5 Experiment and Results

After successfully implementing and tuning the machine learning methods, we tested them on several different scenarios to see how they behaved. We used sensor and bathymetry data from October 2015, a time period with many storms, as well as July 2016, a relatively calm month. This sensor and bathymetry data was put into our physical model, and we used the model to predict the location, height, period, and angle of the breaking waves. This dataset, together with its output, was then split into two parts for training and testing. This process is detailed in Figure 10.
The schematic for the strategy used in the paper. Wave Rider is the buoy named “awac 11m”.

The tuning parameters chosen for SVR were \(\gamma = 0.1\) and \(C = 1e3\). A neural network with two hidden layers and five neurons in each layer was chosen, after an exhaustive process described above. The rectified linear function was chosen as the activation function. The networks were trained by minimizing the mean squared error of the neurons over 500 epochs and using the RMSprop Optimizer (proposed by Geoff Hinton), which divides the learning rate by an exponentially decaying average of squared gradients. The hypotheses tested are detailed below:

1. Train on 80% of the July data, and test on the remaining (disjoint) 20% of the July data. The conditions in July are calm and relatively predictable, so the machine learning methods should do well.

2. Train on the entire July dataset, and then test on the October dataset. These two datasets
are from different seasons and have vastly different weather conditions. It is expected that the machine learning methods will not do well in this scenario.

3. Train on 80% of the July and October datasets, and test on the remaining 20%. Combining data from October in our training set should make our machine learning methods more robust, and we should see an improvement in predictions.

4. Add in tides and dynamic bathymetry to Experiment 3. The goal of this project is to be able to take any boundary conditions and bathymetry in a reasonable parameter space and be able to predict the near-shore conditions. This is one step closer to achieving that goal.

For the plots in this section, the physical model prediction is plotted along the X-axis and the predicted values of the Machine Learning Models are plotted along the Y-axis. Since we are comparing ML models with the Physical model, we say one ML method performs better than other (is preferable) if the predicted values from that method are closer to the blue line (i.e., the predicted value from physical model); this can be quantified by the Mean Absolute Error of the predicted values from the specific ML model compared to the physical model.

To compare and contrast the various machine learning algorithms, we perform four different kinds of experiments.

5.1 Experiment 1

In Figures 11 and 12 the predicted values of near-shore dynamics from the machine learning techniques Support Vector Regression (SVR) and Neural Networks (NN) are plotted against the predicted values from the physical model. The machine learning model was trained and tested on the month of July 2016. More specifically, a randomly chosen 80% subset of the data was used for training, whereas the other 20% was used for testing.

![Figure 11](image.png)

Figure 11: Figure showing the plot of result of predicting wave breaking angle compared to physical model (left) and wave breaking height compared to physical model (right). Here the model is trained using data in the month July 2016 and the predictions are also made for July 2016.
The rationale for choosing July is that it is a relatively calm month for the near-shore dynamics, so it appears to be a good test case for the machine learning model to learn the data.

It can be seen that both ML models do well overall in predicting all the parameters. In Figure 11 (left), both methods did a very good job in predicting the wave breaking height when compared to the physical model. The wave breaking angle predicted by using NN is very close to the predicted value form the physical model in comparison to SVR model.

In Figure 11 (right), both SVR and NN methods did a very good job in predicting the wave breaking height as compared to the physical model. The wave breaking height predicted by NN is almost indistinguishable from the predicted value from the physical modal. For this case NN outperforms SVR in replicating the result of physical model. In some sense, NN could be a substitute for the physical model in predicting wave breaking period.

In Figure 12 (left), SVR and NN methods both give similar types of results on predicting the wave breaking location. The outputs from NN are closer to that of the physical model as compared to SVR. So, in this case, NN is preferable to SVR. Generally speaking, both ML methods do a poor job in predicting wave breaking location. In Figure 12 (right), SVR and NN methods both give similar results on predicting the wave breaking period and both methods are able to successfully replicate the predicted outcomes of the physical model.

5.2 Experiment 2

In Figure 13 and Figure 14 we train the ML models on the data from July, 2016 and predict on October, 2015. July is the calm month for sea waves, tides and depth but October is full of storms, so there are a lot of irregularities in the predictors. This makes the task of predictions complicated, since the training conditions are not representative of the prediction conditions.
Figure 13: Results of predicted wave breaking angle compared to physical model (left) and wave breaking height compared to physical model (right). Here the model is trained using data from July 2016 and predicts for October 2015.

Figure 14: Figure showing the plot of result of predicting wave breaking location compared to physical model (left) and wave breaking period compared to physical model (right). Here, the model is trained using data from July 2016 and predicts for October 2015.

The ML methods SVR and NN performed relatively well in predicting wave breaking angle (Figure 13, left) and wave breaking period (Figure 14, right). For wave breaking height (Figure 13, right) NN performed much better than SVR. The variance of the SVR prediction compared to the physical model keeps increasing with the wave height. Here, NN clearly outperformed SVR to predict wave breaking height and it may be a good substitute for the physical model.

In the case of wave breaking location (Figure 14, left), as we move from seashore towards the sea, the prediction of ML models get worse. For predicting wave breaking period (Figure 14, right), NN did better than SVR. The tendency of the prediction from NN looks interesting. When the wave period is around 2 to 7 sec, NN prediction is slightly greater than that of the physical model and for wave periods longer than 7 sec, predicted wave breaking period is slightly lower than the prediction from the physical model. The absolute error between NN and the physical model looks
approximately the same throughout the period. On the other hand, the prediction of the SVR gets worse after a wave period of 8 sec.

5.3 Experiment 3

In Figures 15 and 16, both SVM and NN did a good job predicting the wave breaking height (Figure 15, right) and wave breaking period (Figure 16, right). However, SVR is performing better than NN as NN is slightly underestimating wave breaking period and doing poor for larger wave height. In the case of Figure 15 (left), SVR is predicting better than NN when the output of both methods were compared to the physical model. NN is underestimating wave breaking angle after the angle is larger than 20 degrees.

Figure 15: Figure showing the plot of result of predicting wave breaking angle compared to physical model (Left) and wave breaking height compared to physical model (Right). Here the model is trained using data from July and October and predicts for July or October.

Figure 16: Figure showing the plot of result of predicting wave breaking location compared to physical model (left) and wave breaking period compared to physical model (right). Here the model is trained using data from July and October and predicts for July or October.
On the other hand, in Figure [15](right), both SVR and NN are predicting wave breaking location poorly compared to the physical model. When we compare the results from training on July and predicting on October, both methods have shown improvement on their predictions; this implies that if we train on a larger data set, then we would most likely have improved predictive ability.

5.4 Experiment 4

Figures [17] and [18] represent the plot of predictions of ML models versus prediction of the physical model. Experiments 1-3 are generated under the assumption that the boundary conditions are paired with their corresponding bathymetry (i.e., “static h”). We now train on a synthetic dataset whose inputs combine asynchronous boundary conditions and bathymetry/tide data. The depth of water is a variable and it was measured every 1m away from the sensor towards the seashore. We trained the ML model on the data of July 2016 and October 2015 combined, and predicted on the July 2016 or October 2015 data.

![Figure 17](image)

Figure 17: Figure showing the plot of result of predicting wave breaking angle compared to physical model (left) and wave breaking height compared to physical model (right). Here the model is trained using data of July 2016 and October 2015 for variable h (dynamic) and predicts for July 2016 / October 2015.
In the case of predicting wave breaking angle (Figure 17, left), wave breaking height (Figure 17, right), and wave breaking period (Figure 18, right), both ML models perform almost equally well. They replicate the predictions of the physical models quite nicely. For the prediction of wave breaking locations, the predicted values from the ML methods are concentrated towards the values predicted by the physical model; however, it has a high variation. In this situation, we can say that ML models were unable to produce satisfactory results for predicting wave breaking location.

**Conclusions of Machine Learning models** These results show that ML models are comparable to the predictions of the physical model in some cases. If we train the ML models on historical data (the data available for the last 30 years) there is hope that ML model can replicate the outcome of the physical model. That means we are optimistic that ML models can be used as an alternative to the physical model for predicting near-shore dynamics.

6 Summary and Future Work

The goal of this project was to make predictions about near-shore hydrodynamics using boundary and bathymetry data and a machine learning model. We trained the machine learning model using data generated by a simplified linear wave using the one-dimensional energy flux conservation equation. We were then able to successfully predict the breaking height, breaking period, and breaking angle, while giving reasonable estimates for the breaking location. One possible explanation for this error is the difficulty in identifying the true breaking location with the physical model. This outlier can potentially influence the machine learning method’s predictions.

Our approach may be limited by the amount of data that we used to train our machine learning models. We trained our models on relatively small amounts of data compared to the amount of data that is available. One way to improve the predictions of near-shore hydrodynamics would be to use a more robust data set to train on. Ideally, this data set would contain a large variety of tides and different bathymetries. We tuned our machine learning models to the month of July, but, realistically, the models should be tuned for every available training set. Additionally, beach slope is an important
factor that affects near-shore hydrodynamics that we were not able to implement into our research. If all of these concepts were implemented into our machine learning models, then this would cause them to be much more accurate and computationally expensive than they currently are. Thus, comparing our two machine learning methods based on relative speed and accuracy comparisons would help decide which approach is more useful in prediction of near-shore hydrodynamics.

If this project were to continue, one thing to further investigate is how to better identify the breaking wave location. Perhaps a more complicated model, such as those used in practice, would more accurately find the breaking location. With this more accurate breaking location in our training set, we anticipate that the machine learning methods will also achieve more accurate results.

References


