Artificial Intelligence and Protein Engineering:  
Information Theoretical Approaches to  
Modeling Enzymatic Catalysis  
by  
James W. Weis  

Submitted to the Department of Electrical Engineering and Computer Science  
on January 5, 2017, in partial fulfillment of the  
requirements for the degree of  
Master of Science in Electrical Engineering and Computer Science  

Abstract  
The importance of dynamics in enzymatic function has been extensively recognized in the literature. Despite this, the vast majority of enzyme engineering approaches are based on the energetic optimization of static molecular structures. Using computational sampling techniques, especially rare-event sampling methods like Transition Path Sampling, we can now generate large datasets of reactive and non-reactive enzyme trajectories. However, due to the high dimensionality of phase space, as well as the highly correlated nature of proteins, using these simulations to understand the drivers of reactivity, or as a starting point for enzyme engineering, is difficult.  

In this work, we apply information theoretic techniques to both formulate and propose potential solutions to this problem. We also present a novel feature selection algorithm that leverages entropy approximations to find the subset of features $X_1,...,X_d$ such that $H(Y|X_1,...,X_d)$, the entropy of the output variable conditioned on the identified features, is minimized. We validate the advantages of this algorithm through a comparative analysis on synthetic datasets—demonstrating both improved predictive performance, as well as reduced conditional entropy—and finally apply these techniques in the analysis of a large dataset of simulated ketol-acid reductoisomerase trajectories. We find not only that we are are capable of predicting enzyme reactivity with a small number of well-chosen molecular features, but also that the GCMI algorithm constitutes a powerful framework for feature selection, and is well-suited for the analysis of such datasets.  

Thesis Supervisor: Professor Bruce Tidor  
Title: Professor of Biological Engineering and Computer Science  

3
Acknowledgments

I'm deeply appreciative of Professor Bruce Tidor’s guidance, without which this thesis would not have been possible. I'm certain that his advice will serve as a guiding voice throughout the rest of my doctoral studies—and into life beyond. I also thank him particularly for his help in the compilation and honing of this document.

I thank the entire Tidor Group for making the lab an enjoyable place to pursue my graduate studies. I learned much from lab discussions, and my thesis—and myself—have benefited from the environment.

I thank Brian Bonk for his patient chemical tutorship. The KARI application in this project would not have been possible without his support—and the past years would have been far less enjoyable without his friendship.

I thank Raja Srinivas for his advice. His insightful input was essential to navigating my initial years—and his support and encouragement were truly valued.

Finally, I am grateful to my family.
THIS PAGE INTENTIONALLY LEFT BLANK
Contents

List of Figures .................................................. 11

List of Tables .................................................. 13

1 The Problem .................................................. 15
   1.1 Introduction ............................................. 15
   1.2 Problem Statement ...................................... 17
       1.2.1 General Problem .................................... 17
       1.2.2 Additional Considerations ......................... 18

2 Theoretical Fundamentals .................................... 19
   2.1 Information Theory ...................................... 19
       2.1.1 Entropy ............................................. 19
       2.1.2 Mutual Information ................................ 21
       2.1.3 Conditional Mutual Information ................. 22
       2.1.4 Differential Entropy ............................... 23
       2.1.5 Maximum Information Spanning Trees ............ 25

3 Approaches & Algorithms .................................... 27
   3.1 Convex Relaxation Approach ............................. 27
   3.2 Greedy, Information Theoretic Approach ............... 28
       3.2.1 CMIM ................................................ 29
       3.2.2 GCMI ................................................ 30
   3.3 Other Benchmark Approaches ............................. 31
3.3.1 Random Sampling ............................................. 31
3.3.2 Mutual Information ......................................... 32
3.3.3 Kolmogorov–Smirnov Test ................................. 32

4 Comparative Analysis of Algorithms on Synthetic Data 33
4.1 Synthetic Dataset Γ ............................................... 33
  4.1.1 Data Generation Methodology .............................. 34
  4.1.2 Features Selected ........................................ 34
  4.1.3 Classification Performance ................................. 34
4.2 Synthetic Dataset Δ ............................................... 36
  4.2.1 Data Generation Methodology .............................. 37
  4.2.2 Features Selected ........................................ 37
  4.2.3 Comparison with Analytical $H(Y|X_1...X_d)$ ............ 39
  4.2.4 Calculation of $H(Y|X_1...X_d)$ .......................... 39
  4.2.5 Evaluation of Algorithms .................................. 40
  4.2.6 Classification Performance ................................. 41

5 Learning Sparse Models of KARI Reactivity 43
5.1 Experimental Set-Up ............................................ 43
  5.1.1 Ketol-Acid Reductoisomerase .............................. 43
  5.1.2 Transition Path Sampling ................................ 44
  5.1.3 Molecular Features & Hypotheses .......................... 46
5.2 Comparing Approaches .......................................... 47
  5.2.1 Features Selected ........................................ 47
  5.2.2 Predictive Power ........................................ 47

6 Summary, Conclusion, & Future Work 53
6.1 Summary .......................................................... 53
6.2 Conclusion ....................................................... 54
6.3 Future Work ..................................................... 54

Bibliography 57
Appendices 63

A Figures 65

B Tables 69
List of Figures

4-1 The normalized frequency of selection of each feature in $\Gamma$ by each algorithm. The first 4 features constitute the true feature set, and all other features are randomly sampled. ........................................... 35

4-2 Mean area under the ROC curve for logistic regression (LR), classification tree (CT), linear kernel support vector machine (LSVM) and radial basis kernel SVM (RSVM) on $\Gamma$ using the features identified by the indicated feature selection algorithms. ............................... 36

4-3 The normalized frequency of selection of each feature in $\Delta$ by each algorithm. The first $\delta_v = 6$ features constitute the true feature set, the final $\delta_v = 6$ are the corresponding noise corrupted features, and all other features are randomly sampled. ........................................... 38

4-4 Calculation of $H(Y|X_1...X_d)$ for $d = 1...6$ (left) and $d = 3...6$ (right) for each feature selection algorithm. The light blue line indicates $H(Y|X_\emptyset)$ and represents the minimum possible uncertainty of $Y$. ................................. 41

4-5 Mean area under the ROC curve for logistic regression (LR), classification tree (CT), linear kernel support vector machine (LSVM) and radial basis kernel SVM (RSVM) on $\Delta$ using the features identified by the indicated feature selection algorithms. ............................... 42

5-1 PyMol cartoon visualization of KARI structure [16]. The active site of the enzyme (which was modeled quantum mechanically) is shown as a blue blob. Details of the alkyl migration reaction are shown in Figure 5-2. 44
5-2 The alkyl migration catalyzed by the KARI enzyme which was simulated using TPS and studied in this work.

5-3 Schematic of tensor with extracted features and response variables for each simulated trajectory and order parameter.

5-4 Classification ability on the simulated KARI dataset using the top $z = 3$ features identified by each algorithm. The top row in each heatmap indicates the AUC achieved with just the top feature, while the subsequent rows are calculated using the indicated feature and all the features above (that is, the AUC shown in the second row, for example, is calculated using the top two features).

5-5 Classification ability for $\theta^{GCM}$ and $\theta^{LASSO}$ on the simulated KARI dataset for the $z = 5$ top features. Each row indicates the AUC achieved using the corresponding feature and all features above.

A-1 The normalized frequency of selection of each feature by each feature selection algorithm on $\Delta$. The first $\delta_v = 6$ features constitute the true feature set, and the final $\delta_c = 6$ are the corresponding noise corrupted features.

A-2 Mean area under the ROC curve for logistic regression (LR), classification tree (CT), linear kernel support vector machine (LSVM) and radial basis kernel SVM (RSVM) on $\Gamma$ using the features identified by each indicated feature selection algorithm.

A-3 Mean area under the ROC curve for logistic regression (LR), classification tree (CT), linear kernel support vector machine (LSVM) and radial basis kernel SVM (RSVM) on $\Delta$ using the features identified by each indicated feature selection algorithm.

A-4 The number of reactive and non-reactive trajectories initially sampled at each order parameter.
List of Tables

5.1 The top three features identified by each algorithm for each order parameter $k = 1...14$. Numbers indicate unique molecular features. . . 48

B.2 The top $z = 10$ features identified by each algorithm for each order parameter $k = 1...14$. Numbers indicate unique molecular features. . . 71

B.1 The top $z = 5$ features identified by each algorithm for each order parameter $k = 1...14$. Numbers indicate unique molecular features. . . 72
Chapter 1

The Problem

1.1 Introduction

Optimized over billions of years, the biological systems underlying life have evolved an enormous and fascinating array of highly tuned capabilities. Breakthroughs in genetic engineering and DNA sequencing, combined with exponential increases in computational and modeling capabilities, have led to a surge of interest in the engineering of biology [12]. This excitement, while at times irrationally directed, is well-justified: the potential held by the ability to rationally redesign biology—from therapeutics and biomaterials to agriculture—is hard to overstate [31].

Enzymes, protein molecules that catalyze the diverse array of reactions required for life, are one of the most interesting of such biological systems. Further, the remarkable speed and specificity with which enzymes accelerate reactions, coupled with their ready availability due to recent advances in molecular biology, make them attractive candidates for the production of many products, from pharmaceuticals to biofuels [26, 37]. Unfortunately, natural enzymes are rarely optimal for the unnatural tasks imagined by scientists and engineers—for example, they may be inappropriately unstable, accept a very small set of substrates, be difficult to produce, or not even exist for the reaction desired [26]. It is this juxtaposition of promise and current limitations that underlies current interest in the design of novel catalysts.

Interestingly, while the importance of enzyme dynamics in reactivity has been
extensively recognized in the literature, current methods for rational enzyme design are largely based on the reduction of activation energy via the energetic optimization of static molecular structures [49, 2, 34, 63]. Transition Path Sampling (TPS) and Transition Interface Sampling (TIS) are relatively new computational methods that allow the generation of a large ensembles of molecular dynamics trajectories containing a rare event, such as an enzymatic reaction, as well as the calculation of the corresponding rate constants [7, 17]. While previous studies have used dynamics-based approaches to evaluate the stability of designed enzymes, TPS or TIS ensembles have not yet been used as a starting point in the rational design of enzymes [26, 49, 2, 34].

One obstacle preventing the full leverage of TPS ensembles is the high-dimensionality and dynamic nature of the mechanistic information produced by the simulations. While human intuition is ill-suited to the analysis of such multivariate data, techniques from artificial intelligence—including machine learning and information theory—provide powerful tools for such tasks [51, 43, 9, 50]. It is the purpose of this work to begin investigating, from both a theoretical and applied perspective, the exciting confluence of these fields.

Specifically, we explore the problem of building sparse models of enzyme reactivity—which are useful not only to help understand the system under study, but also to identify targets for engineering efforts. Ideally, we would like to identify the subset of features that, when taken as a group, tell us the most about the system. Unfortunately, because of the correlated, high-dimensional relationships present in large datasets such as those produced by molecular simulations, simply identifying the features with the highest information about the system, without requiring minimal overlap between those features, can lead to significantly sub-optimal solutions. Enforcing such a constraint, however, is an intractable problem for even mild size datasets—and so approximate solutions are required.

We begin by phrasing our problem generally—as well as outlining specific constraints that stem from our unique application. Then, after introducing the requisite theoretical foundations in Chapter 2, we begin Chapter 3 by outlining two distinct approaches to finding good approximate solutions for the original, typically intractable
problem. We conclude Chapter 3 by introducing the GCMI algorithm, a scalable feature selection method firmly rooted in information theory. We conduct a comparative analysis of GCMI and other modern feature selection algorithms on synthetic datasets in Chapter 4, wherein we quantify both the information theoretic reduction in uncertainty of the system output, as well as the degree to which such solutions transfer to increased performance in classification tasks. We extend this analysis to real TPS data in Chapter 5, and conclude with a discussion and future research directions in Chapter 6.

1.2 Problem Statement

1.2.1 General Problem

We seek a method of identifying the subset of molecular features that, when taken as a group, best describes reactivity. More generally, given some design matrix \( X \) with \( n \) samples \( x \in \mathcal{X} \) and corresponding label vector \( Y \) with \( y \in \mathcal{Y} \), we seek the subset of \( z \) features \( \theta \) that minimizes some function \( \mathcal{L} \),

\[
\min_{\theta} \mathcal{L}, \quad ||\theta||_0 = z, \tag{1.1}
\]

where the \( \ell_0 \)-"norm" is equivalent to the number of features in \( z \)\(^1\). These features can then be analyzed independently, or used in future learning tasks. The function \( \mathcal{L} \) can be part of a learning model (known as embedded feature selection methods) or defined separately from the the model (known as filter methods). Of course, implicit in this statement is the important fact that the solution is found over the feature group \( \theta \), and is not a simple combination of individually optimal features—thus leading to the difficult nature of the problem.

\(^1\)While we use \( ||\theta||_0 = z \) throughout this work for consistency, other constraints (such as those based on the objective function value, or empirical risk) can be substituted.
1.2.2 Additional Considerations

This investigation is motivated by eventual application, so any solution to the above problem will also be judged by its suitability to the application in question. Specifically, we are interested in methods of solving for $\theta$ in conditions similar to those found in the computational simulation of protein movement and function (where $Y$ represents a behavior of interest)—an environment that presents its own unique suite of challenges and requirements. For example, molecular simulations of protein function are resource-intensive and produce large amounts of data (measuring in gigabytes and terabytes is not uncommon). Additionally, as it mirrors biological systems, the data produced by these simulations normally contains complex, noisy, and high-dimensional interactions. Further, as we would ideally like our solutions to advance some chemical understanding or experimental efficiency, we are also bottlenecked by human intuition and experimental throughput—and, as such, generally prefer a few highly informative features over a large number of less informative features.
Chapter 2

Theoretical Fundamentals

In this section, we lay out requisite theoretical fundamentals, as well as relevant estimation and approximation methodologies.

2.1 Information Theory

We begin with a brief introduction to information theoretical concepts and algorithms. We also introduce the MIST approximation—which we will leverage in Chapter 3. For an in-depth introduction to information theory, the reader is referred to [15].

2.1.1 Entropy

Given a discrete random variable $X \in \mathcal{X}$ with probability mass function $p_X(x) = \Pr(X = x)$ for $x \in \mathcal{X}$, we define $H(X)$, the entropy of $X$, as

$$H(X) = - \sum_{x \in \mathcal{X}} p_X(x) \log p_X(x).$$  \hspace{1cm} (2.1)

The logarithm above is typically to the base 2, with the corresponding entropy expressed in bits, although the entropy to any base $b$ can be taken, and is expressed $H_b(X)$. For the special cases $b = e$ and $b = 10$, the entropies $H_e(X)$ and $H_{10}(X)$ are measured in nats and hartleys, respectively.
We can arrive at this definition of entropy in a variety of ways (one of which is outlined in Shannon’s seminal 1948 paper “A Mathematical Theory of Communication”), but it can be intuitively understood as a measure of uncertainty—especially, as the amount of information required on average to describe the random variable $X$ [15].

**Joint Entropy**

If we consider the single random variable $X$ mentioned above to be a vector-valued random variable, we arrive naturally at the definition of the joint entropy $H(X, Y)$ of random variables $X$ and $Y$ with joint distribution $p_{X,Y}(x, y)$,

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x, y) \log p_{X,Y}(X, Y).$$

(2.2)

Of course, as it is an extension of the univariate entropy, the joint entropy measures the uncertainty of the set of variables $\{X, Y\}$. Further, we see that the joint entropy of the set of random variables $S$ is an upper-bound on the entropy of any individual entropy in the set,

$$H(S) \geq \max_{s \in S} H(s),$$

(2.3)

and also that the joint entropy $H(S)$ is a lower-bound on the sum of the individual entropies of the variables in $S$,

$$H(S) \leq \sum_{s \in S} H(s),$$

(2.4)

where $H(S) = \sum_{s \in S} H(s)$ if and only if all $s \in S$ are statistically independent.

**Conditional Entropy**

Next, we would like to define $H(Y|X)$, the entropy of random variable $Y$ given random variable $X$. Intuitively, we can understand this quantity as the bits of information required to fully describe the state of $\{X, Y\}$, given the value of $X$. If we first learn the value of $X$, we have (by the proceeding definitions) gained $H(X)$ bits of information. We therefore only require $H(X, Y) - H(X) = H(Y|X)$ more bits of information to
fully describe \((X,Y)\). We can use this intuition (known as the chain rule) to derive 
\(H(Y|X)\),

\[
H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y) \log p_{X,Y}(x,y)
\]

(2.5)

\[
= - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y) \log p_X(x)p_{Y|X}(y|x)
\]

(2.6)

\[
= - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y) \log p_X(x) - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y) \log p_{Y|X}(y|x)
\]

(2.7)

\[
= - \sum_{x \in X} p_X(x) \log p_X(x) - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y) \log p_{Y|X}(y|x)
\]

(2.8)

\[
= H(X) - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y) \log p_{Y|X}(y|x).
\]

(2.9)

Using \(H(X,Y) = H(X) + H(Y|X)\), we come to the definition of conditional entropy

\[
H(Y|X) = - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y) \log p_{Y|X}(y|x).
\]

(2.10)

Relative Entropy

The relative entropy, \(D(p||q)\), is a measure of the difference between two distributions \(p\) and \(q\),

\[
D(p||q) = \sum_{x \in X} p_X(x) \log \frac{p_X(x)}{q_X(x)}.
\]

(2.11)

The relative entropy is also called the Kullback-Leibler divergence, and represents the extra bits needed to code the distribution \(p\) if the code for distribution \(q\) was used. The relative entropy \(D(p||q)\) is not symmetric in \(p\) and \(q\), and is always greater than zero—except when \(p = q\), in which case it is equal to zero.

2.1.2 Mutual Information

The mutual information of two random variables, \(I(X;Y)\), measures the information shared by \(X\) and \(Y\)—or, equivalently, the reduction in uncertainty of one variable gained by the knowledge of the other. Representing this in information theoretic
terms,

\[ I(X; Y) = H(X) - H(X|Y) \]  \hfill (2.12)
\[ = H(Y) - H(Y|X). \]  \hfill (2.13)  

We can define mutual information using this (and the proceeding definitions),

\[
H(X) - H(X|Y) = -\sum_{x\in\mathcal{X}} p_X(x) \log p_X(x) + \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y) \log p_{X|Y}(x|y)  
\]
\[
= -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y) \log p_X(x)
+ \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y) \log p_{X|Y}(x, y)
\]
\[
= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y) \log \frac{p_{X|Y}(x|y)}{p_X(x)}
\]
\[
= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x, y) \log \frac{p_{X,Y}(x|y)}{p_X(x)p_Y(y)} = I(X; Y). \]  \hfill (2.15)  

We see that the mutual information is equal to the relative entropy between \( p_{X,Y}(x, y) \) and \( p_X(X)p_Y(Y) \). It’s easy to see that

\[ I(X; Y) = H(X) + H(Y) - H(X, Y) \]  \hfill (2.16)

and also that

\[ I(X; X) = H(X). \]  \hfill (2.17)

\[ I(X; X) = H(X). \]  \hfill (2.18)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)

\[ I(X; X) = H(X). \]  \hfill (2.19)
known. Thus, it is defined by

\[
I(X; Y|Z) = H(X|Z) - H(X|Y, Z)
\]  
(2.20)

\[
= \sum_{z \in Z} p_z(z) \sum_{x \in X} \sum_{y \in Y} p_{X,Y|Z}(x, y|z) \log \frac{p_{X,Y,Z}(x, y|z)}{p_X(x|z)p_Y(y|z)}.
\]  
(2.21)

Using the definitions of conditional and joint entropy, we can also express the \(I(X; Y|Z)\) as a combination of joint entropies,

\[
I(X; Y|Z) = H(X, Z) + H(Y, Z) - H(X, Y, Z) - H(Z).
\]  
(2.22)

\subsection*{2.1.4 Differential Entropy}

Differential entropy is Shannon's extension of (discrete) entropy to continuous probability distributions. Given continuous random variable \(X\) with density \(f_X(x)\) and support \(S\), the differential entropy \(h(X)\) of \(X\) is defined as

\[
h(X) = -\int_S f_X(x) \log f_X(x) dx.
\]  
(2.23)

Differential entropy shares many important quantities with discrete entropy. For example,

\[
h(X_1...X_p) = \sum_{i=1}^{p} h(X_i|X_1...X_{i-1}) \leq \sum_{i=1}^{p} h(X_i).
\]  
(2.24)

However, there are some important differences between discrete and differential entropy—for example, differential entropy can be negative, and is not invariant to arbitrary invertible maps [15]. Calculations on continuous variables, therefore, must be done with care.
Joint Differential Entropy

The joint differential entropy of continuous random variables $X_1...X_p$ with corresponding density $f(x_1...x_p)$ is defined as

$$h(X_1...X_p) = -\int f_{X_1...X_p}(x_1...x_p) \log f_{X_1...X_p}(x_1...x_p) dx_1...dx_p.$$  \hspace{1cm} (2.25)

Using Bayes' theorem $f_{X|Y}(x|y) = f_{Y|X}(y|x) f_X(x)/f_Y(y)$, we see that

$$h(X|Y) = h(X,Y) - h(Y).$$  \hspace{1cm} (2.26)

Conditional Differential Entropy

Given continuous random variables $X$ and $Y$ with joint density $f(x,y)$, the differential entropy of $X$ conditioned on $Y$ is

$$h(X|Y) = -\int f_{X|Y}(x,y) \log f_{X|Y}(x,y) dxdy.$$  \hspace{1cm} (2.27)

Relative Differential Entropy

The relative differential entropy (also called the Kullback-Leibler distance) between densities $f_X(x)$ and $g_X(x)$ is defined to be

$$D(f||g) = \int f_X(x) \log \frac{f_X(x)}{g_X(x)}.$$  \hspace{1cm} (2.28)

If the support of $g_X(x)$ is not contained within the support of $f_X(x)$, then $D(f||g) = \infty$.

Relating Differential Entropy to Discrete Entropy

Now that we have introduced differential entropy, we briefly inspect the relationship between differential and discrete entropy as outlined in [15]. Understanding this relationship is critical when estimating these quantities.

We consider a continuous random variable $X \in \mathcal{X}$ with corresponding probability density function $f_X(x)$. If we discretize the space $\mathcal{X}$ by dividing it into equal-sized
bins of length $\Delta$, we have a new random variable $X^\Delta \in \mathcal{X}^\Delta$ defined by

$$X^\Delta = x_i, \; i\Delta \leq X < (i + 1)\Delta.$$  \hspace{1cm} (2.29)

By the mean-value theorem, there exists an $x_i$ such that

$$\int_{i\Delta}^{(i+1)\Delta} f_{X^\Delta}(x) \, dx = f_{X^\Delta}(x_i) \Delta = p_i,$$  \hspace{1cm} (2.30)

where $p_i$ is the probability of $X$ being in the $i$th bin. So, the entropy of this new, quantized random variable is

$$H(X^\Delta) = - \sum_i p_i \log p_i$$  \hspace{1cm} (2.31)

$$= - \sum_i f_{X^\Delta}(x_i) \Delta \log f_{X^\Delta}(x_i)$$  \hspace{1cm} (2.32)

$$= - \sum_i f_{X^\Delta}(x_i) \Delta \log f_{X^\Delta}(x_i) - \sum_i f_{X^\Delta}(x_i) \Delta \log \Delta.$$  \hspace{1cm} (2.33)

Using that $\sum_i f_{X^\Delta}(x_i) \Delta = \int f_X(x) = 1,$

$$= - \sum_i f_{X^\Delta}(x_i) \Delta \log f_{X^\Delta}(x_i) - \log \Delta.$$  \hspace{1cm} (2.34)

So, if the density $f_X(x)$ of $X$ is Riemann integrable (and thus the limit is well defined), the entropy of an $b$-bit quantization of the continuous random $X$ variable approaches $h(X) + b$ as $\Delta \to 0$.

### 2.1.5 Maximum Information Spanning Trees

The Maximum Information Spanning Tree, or MIST, approximation was introduced by King and Tidor in 2009, and provides a framework for the approximation of joint entropies [33]. The MIST approximation uses lower-order relationships to estimate higher-order terms, and as such is particularly useful when working with high-dimensional information theoretic quantities, which are difficult to estimate
directly.

The goal of the MIST framework is to find a $H_n^k$, a $k$-dimensional approximation for an $n$-dimensional joint entropy,

$$H_n^k(H_1...H_k) \approx H_n(x_1...x_n), \quad (2.35)$$

for true $i$th order entropies $H_i$. Briefly, this is accomplished by noting that

$$H_n(x_1...x_n) = \sum_{i=1}^{n} H(x_i|x_1...x_{i-1}) \quad (2.36)$$

$$= H_k(x_1...x_k) + \sum_{i=k+1}^{n} H_i(x_i|x_1...x_{i-1}). \quad (2.37)$$

$$\leq H_k(x_1...x_k) + \sum_{i=k+1}^{n} H_i(x_i|x_1...x_{k-1}), \quad (2.38)$$

where the inequality arises because conditioning on a random variable cannot increase entropy. Thus, $H_n^k$ is an upper-bound on $H_n$, and the goal is to find the sets $J$ and $C^1...C^{n-k}$ consisting of the indices that make up the joint entropy term $H_k(x_1...x_k)$, and each of the conditional entropy terms, $H_i(x_i|x_1...x_{k-1})$, respectively, such that the upper bound is minimized. Formally,

$$\min_{J,C^1...C^{n-k}} H_k(x_J_1...x_J_k) + \sum_{i=k+1}^{n} H_i(x_i|x_{C^1_1...x_{C^1_{i-1}}}). \quad (2.39)$$

One way to solve this problem is by phrasing it as a minimum spanning tree problem with $n$ nodes, where each node represents a variable and each edge represents the information between the connected nodes—leading to the approximation’s name. For full details, we refer the reader to [33].
Chapter 3

Approaches & Algorithms

In this chapter, we formulate two approaches to addressing the general problem statement outline in Section 1.2.1. One of these approaches is rooted in statistical learning theory, and the other is information theoretic in nature. We also outline algorithms for solving these problems—including GCMI, a novel information theoretic feature selection algorithm based on the approximate conditional mutual information-based minimization of conditional entropy.

3.1 Convex Relaxation Approach

Adopting a regularized statistical learning approach, we can formulate our problem as

$$
\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i) + \lambda R(f)
$$

for $f : \mathcal{X} \to \mathbb{R}$ from hypothesis space $\mathcal{H}$, regularizer $R : \mathcal{H} \to [0, \infty)$, loss function $V : \mathcal{Y} \times \mathbb{R} \to [0, \infty)$ and regularization parameter $\lambda > 0$ [48].

A natural regularization function $R(f)$, from a modeling perspective, is the $\ell_0$-norm $||f||_0$, which is equal to the number of features used by $f$ in prediction. The corresponding problem is

$$
\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i) + \lambda ||f||_0.
$$
Unfortunately, while this approach is intuitive, it is typically intractable. In fact, solving this problem is equivalent to trying all possible combinations of features—which has combinatorial complexity [69].

This intractability motivates the adoption of a convex relaxation. Specifically, we replace the \(\ell_0\)-norm with an \(\ell_1\)-norm. In the linear, least squares case, this is equivalent to the LASSO problem [57]. As the response variables are binary in our setting, we adopt a binomial model with a logistic link function [40]. Our problem, then, becomes

\[
\min_{w \in \mathbb{R}^p} - \left[ \frac{1}{n} \sum_{i=1}^{n} y_i \langle w, x_i \rangle - \log \left( 1 + e^{\langle w, x_i \rangle} \right) \right] + \lambda ||w||_1, \tag{3.3}
\]

where \(\langle w, x_i \rangle\) is the inner product between the model coefficients and the datum \(x_i\). We solve this problem for a grid of \(\lambda\) values using standard tools from non-smooth convex optimization, and then identify the maximum value of \(\lambda\) that satisfies our constraint (typically a function of \(||w||_0\), the number of non-zero parameters, or the model's error) [48, 57].

### 3.2 Greedy, Information Theoretic Approach

We can instead formulate our problem from an information theoretic perspective. As we seek the features \(\theta\) that best explain our output variable \(Y\), a natural choice for \(\mathcal{L}\) is \(H(Y|X_\theta)\), the entropy of the output variable \(Y\) conditioned on the the selected feature variables \(X_\theta\). Intuitively, we would like to identify the set of features that, when known, result in minimum uncertainty in \(Y\). Formally, we would like to solve

\[
\min_{\theta} H(Y|X_\theta), \quad ||\theta||_0 \leq z \tag{3.4}
\]

where the \(\ell_0\)-norm here allows no more than \(z\) features to be selected. Unfortunately, the naive, brute-force approach is again typically intractable. A complete search would require the estimation of \(2^{||\theta||_0+1}\) probabilities—as well as sufficient data to estimate
Thus, we instead adopt a greedy approach based on the conditional mutual information $I(X,Y|Z)$. We present here two distinct algorithms, both of which take different approximate approaches to solving Equation 3.4 via the application of conditional mutual information. This first algorithm, CMIM, uses an iterative greedy maximum-of-minimums formulation, and was presented by Fleuret in 2004 [21]. The second approach, GCMI, is proposed by us, and is a natural information theoretic approach that can be made to scale to with the use of entropy approximation techniques.

### 3.2.1 CMIM

The CMIM algorithm considers a particular feature $X_i$ good if $\hat{I}(X_i; Y|X_j)$ is large for every $X_j$ that has already been picked. The underlying goal is to capture features with information about $Y$ that is not carried in any of the previously selected features. Specifically, the GCMI algorithm is defined by the iterative procedure

$$\theta_1^{CMIM} = \arg\max_{i \in \{1..p\}} \hat{I}(X_i; Y)$$

$$\theta_u^{CMIM} = \arg\max_{j \in \{1..p\} \setminus \{\theta_1^{CMIM}, \theta_{u-1}^{CMIM}\}} \left[ \min_{j \in \{1..u-1\}} \hat{I}(X_i; Y|X_{\theta_j}) \right]$$

for $1 \leq u \leq z$. The score $\hat{I}(X_i; Y|X_j)$ is low when $X_i$ does not contain information about $Y$, or when the information it does contain has already been captured by $X_j$. So, by taking the $X_i$ with the maximum minimum score, the algorithm attempts to ensure that it only selects features that contain non-overlapping information about $Y$.

However, although CMIM selects relevant features while avoiding both redundant features and high-dimensional information theoretic calculations, its ability to identify sets of features that interact with the output as a group is limited. Since it prioritizes features with minimum conditional mutual information, CMIM does not necessarily select sets of variables with high complementarity or dependence with the output [64].

There have been several extensions of the CMIM algorithm that have attempted
to address this problem. For example, in 2010, Vergara proposed CMIM-2, which simply replaces the minimum of the conditional mutual information terms with their mean [64]. Most recently, in 2015, Bennasar proposed JMIM in [6], which replaces the conditional mutual information term with the joint mutual information,

\[ \theta_{\mu}^{JMIM} = \arg \max_{j \in \{1,...,p\} \setminus \{\theta_1, \theta_{\mu-1}\}} \left[ \min_{j \in \{1,..,u-1\}} \hat{I}(X_i, X_{\theta_j}; Y) \right]. \quad (3.7) \]

While interesting, these approaches do not attempt to explicitly model higher-order interactions—or to estimate them directly, should the resources be available. As such, we lack a general, adjustable framework that phrases the problem in terms of the optimal objective function and that only introduces a level of approximation commensurate with both the size and type of data under analysis.

### 3.2.2 GCMI

We propose the GCMI algorithm to address these concerns. The GCMI algorithm is also a forward-selection algorithm that iteratively identifies features with novel information about the variable \( Y \). However, rather than adopt the maximum-minimum heuristic, we condition on all of the selected features at each step—and then approximate high-dimensional quantities as necessary. Specifically, we find \( \theta_{GCMI} \) through the iterative scheme

\[ \theta_1 = \arg \max_{j \in \{1,...,p\}} \hat{I}(X_j; Y) \quad (3.8) \]

\[ \theta_u = \arg \max_{j \in \{1,...,p\} \setminus \{\theta_1, \theta_{\mu-1}\}} \hat{I}(X_j; Y \mid X_{\theta_1}...X_{\theta_{u-1}}). \quad (3.9) \]

for \( 1 \leq u \leq z \). That is, we first identify the single feature with the highest mutual information with reactivity. Then, we find the feature that, conditioned on our first feature, has highest mutual information with reactivity, and continue this process until we reach the desired number of variables. Clearly, this approach seeks to identify features with information about \( Y \) that has not been captured by any of the previously selected features.
We emphasize that, unlike in the CMIM algorithm, the information theoretic quantities calculated by the GCMI algorithm may be high-dimensional—requiring very large datasets and ample computational power for even relatively small values of \( z \). We address this shortcoming via an entropy expansion of \( \hat{I}(X_j; Y \mid X_{\theta_1}...X_{\theta_{u-1}}) \),

\[
\hat{I}(X_j; Y \mid X_{\theta_1}...X_{\theta_{u-1}}) = \hat{H}(X_j, X_{\theta_1}...X_{\theta_{u-1}}) + \hat{H}(Y, X_{\theta_1}...X_{\theta_{u-1}}) - \hat{H}(X_j, Y, X_{\theta_1}...X_{\theta_{u-1}}) - \hat{H}(X_{\theta_1}...X_{\theta_{u-1}}) + \hat{H}(X_j|X_{\theta_1}...X_{\theta_{u-1}}) - \hat{H}(X_j|Y, X_{\theta_1}...X_{\theta_{u-1}}). \tag{3.10}
\]

After choosing our maximum directly estimable dimensionality \( d \) based on the data and computational power available, we approximate entropies of dimensionality \( j > d \) via a \( d \)th order MIST approximation,

\[
\hat{H}_d^j = \hat{H}_d(X_1...X_d) + \sum_{i=k+1}^{n} \hat{H}_i(X_i|X_1...X_{k-1}), \tag{3.13}
\]

where the ordering of the indices is solved for so as to minimize the expression (see Section 2.1.5 for more details) [33].

### 3.3 Other Benchmark Approaches

We compare the algorithms above with various other benchmark feature selection methods as well, which we briefly describe below.

#### 3.3.1 Random Sampling

Perhaps the most simple feature selection algorithm, random sampling denotes simply choosing a feature set of the desired size uniformly at random and without replacement from the set of available features.
3.3.2 Mutual Information

Selecting a feature $X_i$ purely based on $\hat{I}(X_i, Y)$, its estimated mutual information with the output, is a common approach (sometimes called the Mutual Information Maximization, or MIM, approach). While this method does ensure that the features selected contain information about reactivity, it does not require that selected features contain different information. As such, the information contained in $X_0$ can be redundant and sub-optimally informative.

3.3.3 Kolmogorov–Smirnov Test

The Kolmogorov-Smirnov statistic is a nonparametric quantification of the distance between two empirical distribution functions. Given two empirical distribution functions $\hat{F}_1$ and $\hat{F}_2$, each with independent, identically distributed observations $x_i$, where

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} 1(X_i \leq x),$$

the Kolmogorov-Smirnov statistic is defined as the maximum absolute difference between the two empirical distribution functions,

$$D_n = \max_x |\hat{F}_1(x) - \hat{F}_2(x)|.$$

For a feature selection task, the Kolmogorov-Smirnov statistic can be used to quantify the difference between the empirical distributions of a feature for the different values of $Y$—after which the top $z$ features with the most significantly distinct distributions are taken.
Chapter 4

Comparative Analysis of Algorithms on Synthetic Data

In this chapter, we investigate the performance of the algorithms discussed above on synthetic datasets. We generate the sets of datasets $\Gamma$, in which the output variable is a complex, non-linear function of some of the features, and $\Delta$, in which the output variable is a random number that is sampled from a distribution determined by some of the features. In both cases, we introduce randomness to make the problem more difficult. We also explore the relationship between feature selection and classification performance and, for the $\Delta$ set, analytically calculate the value of our information theoretic objective at each step in the feature selection process for comparison with the optimal value.

4.1 Synthetic Dataset $\Gamma$

We begin by discussing the data generation methodology used to create $\Gamma$. We then analyze the features selected by each algorithm, as well as the classification performance enabled by each feature set.
4.1.1 Data Generation Methodology

We define $\Gamma = \{(X^1, Y^1) \ldots (X^{\gamma_d}, Y^{\gamma_d})\}$ as a set of $\gamma_d$ synthetic datasets. Each dataset $(X^j, Y^j)$ has $\gamma_{d}^j$ features and $\gamma_{n}^j$ samples. Each feature is generated from a uniform distribution with $\gamma_{s}$ states, thus creating design matrix $X^j$, and the first four features from $X^j$ are taken to be the true feature vector $\theta = (1, 2, 3, 4)$. We create $Y^j = (y_1^j \ldots y_{\gamma_n}^j)$ by calculating the output $y_i^j$ for each sample using the feature values $x_1 \ldots x_4$ such that

$$y_i^j = \begin{cases} 
\cos(x_{i,3}^j + x_{i,4}^j) > 0 & \text{if } x_{i,1}^j > x_{i,2}^j \\
\sin(x_{i,3}^j + x_{i,4}^j) > 0 & \text{otherwise.}
\end{cases}$$

(4.1)

We finish the data generation process when we have the set

$$\Gamma = \{(X^i, Y^i) | 1 \leq i \leq \gamma_d, \ i \in \mathbb{Z}\}$$

containing $\gamma_d$ datasets.

4.1.2 Features Selected

After generating $\Gamma$, we ran the feature selection algorithms discussed above over each dataset. For each dataset, we stored the top 4 features identified by each algorithm. As shown in Figure 4-1, all of the algorithms (with the exception of the random feature sampler) were able to identify features 3 and 4. However, only the CMIM and GCMI algorithms were able to correctly elucidate the importance of the first two feature in the determination of $Y$. Further, only the GCMI algorithm was able to do so consistently. These results emphasize the importance of framing the feature selection problem in a way that considers the interactions between features—even if those interactions occur in a non-linear fashion.

4.1.3 Classification Performance

We evaluated classification performance by attempting to predict $Y$, our binary output vector, using only the feature vectors identified by each algorithm from $\Gamma$. We predicted
Figure 4-1: The normalized frequency of selection of each feature in \( \Gamma \) by each algorithm. The first 4 features constitute the true feature set, and all other features are randomly sampled.

\( Y \) using both linear and non-linear classification models—including logistic regression, classification tree, linear kernel SVM, and radial basis kernel SVM. Every model underwent 10-fold cross-validation, and the results were used to calculate a mean AUC score for every dataset in \( \Delta \). Figure 4-2 shows the results for every combination of feature selection and classification methods investigated (see Figures A-2 for more detail).

The linear classification models (logistic regression and linear kernel SVM) are outperformed by the non-linear models (classification tree, radial basis kernel SVM) for all feature sets tested—unsurprisingly, given the non-linear nature of our data. However, while the linear models achieve roughly equivalent performance across all of the feature sets, the non-linear models exhibit significant variation in performance—with the information theoretic metrics that account for interactions between features leading to superior classification performance. Specifically, the features identified by the CMIM and GCMI algorithms result in 93.7% and 100.0% mean AUC, respectively, when paired with the classification tree model, and 86.5% and 100.0% mean AUC,
Figure 4-2: Mean area under the ROC curve for logistic regression (LR), classification tree (CT), linear kernel support vector machine (LSVM) and radial basis kernel SVM (RSVM) on $\Gamma$ using the features identified by the indicated feature selection algorithms.

respectively, when paired with the radial basis kernel SVM.

4.2 Synthetic Dataset $\Delta$

In this section, we describe the creation and analysis of $\Delta$, a set of synthetic datasets wherein $\mathcal{Y}$, the space of possible values for the output variable $Y$, as well as the probabilities with which those values are taken, are random functions of a subset of the feature space $\mathcal{X}$. This dataset is additionally interesting because we can derive the optimal value of our information theoretic objective function—allowing us to compare the performance of our feature selection algorithms in a principled manner.
4.2.1 Data Generation Methodology

We define \( \Delta = \{(X^1, Y^1),..., (X^{\delta_d}, Y^{\delta_d})\} \) as our set of \( \delta_d \) synthetic datasets. Each dataset \( (X^j, Y^j) \) has \( \delta^p \) features and \( \delta^m \) samples. Then, \( \delta^t \) features (an even number) are chosen as the true feature vector \( \theta^j = (\theta^j_1, ..., \theta^j_{\delta^t}) \). The remaining \( \delta^p - \delta^t \) are either randomly sampled (creating \( \delta^f \) random features) or calculated by taking the true features and adding random noise (creating \( \delta^c \) corrupted features). Output random variable \( Y^j \) is then sampled such that its value is taken as one of \( x^j_1, x^j_2, ..., x^j_{\delta^t} \), this first half of the true features, according to a distribution determined by \( x^j_{\frac{1}{2}\delta^t+1}, x^j_{\delta^t} \), the second half of the true features:

\[
Y^j(x^j_{\theta_1},...,x^j_{\theta_{\delta^t}}) = \begin{cases} 
  x^j_{\theta_1} & \text{with probability } \frac{x^j_{\frac{1}{2}\delta^t+1}}{\sum_{k=\frac{1}{2}\delta^t+1}^{\delta^t} x^j_k} \\
  \vdots \\
  x^j_{\theta_{\delta^t}} & \text{with probability } \frac{x^j_{\delta^t}}{\sum_{k=-\frac{1}{2}\delta^t+1}^{\delta^t} x^j_k} 
\end{cases} \tag{4.2}
\]

At the conclusion of this process, we have the set

\[
\Delta = \{(X^i, Y^i) \mid 1 \leq i \leq \delta_d, \ i \in \mathbb{Z}\}
\]

containing \( \delta_d \) datasets, each of which consists of an output vector \( Y^j \) and a design matrix \( X^j \) with \( \delta^t \) true features, \( \delta^f \) random features, and \( \delta^c \) noise-corrupted features.

4.2.2 Features Selected

We proceed by running each of the feature selection algorithms discussed on \( \Delta \) with \( \delta_p = 32 \) features, of which the first \( \delta_v = 6 \) are uniformly distributed over \( \delta_s \) states and comprise the true feature set \( \theta \), the subsequent \( \delta_r = 20 \) are randomly sampled from the same distribution as the true features, and the final \( \delta_c = 6 \) are noise corrupted versions of each of the true features. For each dataset \( (X, Y) \in \Delta \), we store the top \( \delta^c \) features identified by each feature selection method.
Figure 4-3: The normalized frequency of selection of each feature in $\Delta$ by each algorithm. The first $\delta_v = 6$ features constitute the true feature set, the final $\delta_c = 6$ are the corresponding noise corrupted features, and all other features are randomly sampled.

As shown in Figure 4-3 (and in more detail in Figure A-1), all of the algorithms (except the random feature sampler) identified the first three features (which determine the value of the output) consistently. However, only the GCMI algorithm regularly selected the three subsequent features (which determine the probability distribution over the possible output values). In fact, besides the CMIM algorithm, no other feature selection method exhibited any enrichment of selection frequency for these distribution-determining features.

The mutual information and Kolmogorov–Smirnov test repeatedly chose the first three features, as well as the noise-corrupted versions of the same features—highlighting the importance of only selecting features with novel information. Interestingly, while the LASSO method consistently identified the first three true feature and exhibited mild enrichment for the corresponding noise corrupted features, it was the only intelligent algorithm to frequently select totally random features.
4.2.3 Comparison with Analytical $H(Y|X_1...X_d)$

In this section, we derive an analytical expression for $H(Y|X_1...X_d)$, the entropy of $Y$ conditioned on the $d$ selected features $X_1...X_d$. Then, we use this to evaluate the performance of our feature selection algorithms in a quantitative and model-free manner. Specifically, we quantify the reduction in uncertainty in our output variable from each feature selection step—and compare the final results with $H(Y|X_0)$, the true minimum possible uncertainty of $Y$.

4.2.4 Calculation of $H(Y|X_1...X_d)$

We have that

$$H(Y|X_1...X_d) = H(Y) - H(X_1...X_d).$$

Because $X_1...X_d$ are independently and uniformly distributed over $\delta_s$ states,

$$H(X_1...X_d) = \sum_{i=1}^{d} H(X_i)$$

$$= -\sum_{i=1}^{d} \sum_{x \in X_i} p(x) \log p(x)$$

$$= -d \log \frac{1}{\delta_s}.$$ 

By the definition of joint entropy,

$$H(Y, X_1...X_d) = -\sum_{y \in Y} \sum_{x_1 \in X_1} ... \sum_{x_d \in X_d} p(y, x_1...x_d) \log p(y, x_1...x_d).$$

Applying Bayes’ theorem,

$$= -\sum_{y \in Y} \sum_{x_1 \in X_1} ... \sum_{x_d \in X_d} p(y|x_1...x_d) p(x_1...x_d) \log p(y|x_1...x_d) p(x_1...x_d).$$
Due to the independence of $X_1...X_d$,

\[
= - \sum_{y \in \mathcal{Y}} \sum_{x_1 \in \mathcal{X}_1} \ldots \sum_{x_d \in \mathcal{X}_d} p(y|x_1...x_d) \left( \prod_{i=1}^{d} p(x_i) \right) \log p(y|x_1...x_d) \left( \prod_{i=1}^{d} d(x_i) \right).
\]

Again using that $X_1...X_d$ are uniformly distributed over $\delta_s$ states,

\[
= - \sum_{y \in \mathcal{Y}} \sum_{x_1 \in \mathcal{X}_1} \ldots \sum_{x_d \in \mathcal{X}_d} p(y|x_1...x_d) \frac{1}{\delta_s^d} \log p(y|x_1...x_d) \frac{1}{\delta_s^d}.
\]

Finally, we have that

\[
H(Y|X_1...X_d) = d \log \frac{1}{\delta_s} - \sum_{y \in \mathcal{Y}} \sum_{x_1 \in \mathcal{X}_1} \ldots \sum_{x_d \in \mathcal{X}_d} p(y|x_1...x_d) \frac{1}{\delta_s^d} \log p(y|x_1...x_d) \frac{1}{\delta_s^d}.
\]

This expression can be easily evaluated for selected feature set $X_1...X_j$ with $j \leq d$, and for the optimal feature set $X_δ$. It can also be extended to other distributions besides the uniform distribution.

### 4.2.5 Evaluation of Algorithms

We again ran each feature selection algorithm (with the exception of the random algorithm) on $\Delta$. This time, we tracked the order in which features were selected\(^1\), and, with each new feature, calculated the entropy of $Y$ conditioned on the selected features. The results, showing the iterative reduction in uncertainty as more features are selected, are shown in Figure 4-4.

The LASSO algorithm exhibited the worst average reduction in conditional entropy, as well as the highest average variance. The mutual information maximization approach and the Kolmogorov–Smirnov test method performed quite similarly, with both outperforming LASSO for the full feature set. The CMIM algorithm reduced the conditional entropy of the output better than all three of these approaches. The GCMI algorithm consistently outperformed all other methods, and was the only algorithm to

\(^1\)For the LASSO algorithm, we iteratively identified the maximum lambda value with $||w||_0 = i$, for $i = 1...d$ to create an ordered feature vector.
4.2.6 Classification Performance

We turn now to the problem of predicting $Y$ based on the features identified by each algorithm. As with $\Gamma$, we used logistic regression, classification tree, linear kernel SVM, and radial basis kernel SVM models. We calculated 10-fold cross-validated AUC on every dataset in $\Delta$ using each feature selection and classification algorithm. The cross-validated AUCs for each combination are shown in Figure 4-5 (see Figure A-3 for precise values).

The linear models (both logistic regression and linear kernel SVM) perform equally well for all the feature selection algorithms (besides the random method). On the other hand, the non-linear methods, including the classification tree and the radial basis kernel SVM, were able to make more accurate predictions with the better feature sets.
Figure 4-5: Mean area under the ROC curve for logistic regression (LR), classification tree (CT), linear kernel support vector machine (LSVM) and radial basis kernel SVM (RSVM) on Δ using the features identified by the indicated feature selection algorithms.
Chapter 5

Learning Sparse Models of KARI Reactivity

5.1 Experimental Set-Up

We turn now to evaluating our feature selection algorithms on real data. Because we are interested in developing new methods for enzyme engineering, we aim to understand how the feature selection methods outlined previously perform within the unique context of hybrid quantum mechanics/molecular mechanics simulations. Further, due to both the large amounts of data produced by these simulations, as well as the highly correlated nature of atoms within proteins, we expect that methods of identifying features sets with minimal overlapping information will be both important and computationally tractable. A notable underlying hypothesis of this study is that the variable $Y$, representing enzyme reactivity, can be described by a small set of features—a question that, to our knowledge, has not been directly addressed in this manner.

5.1.1 Ketol-Acid Reductoisomerase

We chose the ketol-acid reductoisomerase (KARI) enzyme as a model system (Figure 5-1). KARI is the second enzyme (proceeded by acetolactate synthase and
succeeded by dihydroxyacid dehydratase) in the biosynthesis of the branched-chain amino acids valine, leucine, and isoleucine. Specifically, KARI catalyzes the reductive isomerization of 2-acetolactate to 2,3-dihydroxyisovalerate in the biosynthesis of valine and leucine [19, 11]. For the synthesis of isoleucine, KARI produces 2,3-dihydroxy-3-methylvalerate from 2-aceto-2-hydroxybutyrate [1]. In both cases, the enzyme requires both Mg$^{2+}$ and NADPH as cofactors [14].

KARI is an appealing model system because it is an important enzyme in the production of biofuels, because there exist high-quality crystal structures (which are necessary for realistic simulations), and because it has been the subject of numerous previous studies [56, 53, 1, 3, 36, 58, 59, 19, 11, 14]. We focused our investigation (both simulations and subsequent analysis) on the alkyl migration step (Figure 5-2).

Figure 5-1: PyMol cartoon visualization of KARI structure [16]. The active site of the enzyme (which was modeled quantum mechanically) is shown as a blue blob. Details of the alkyl migration reaction are shown in Figure 5-2.

### 5.1.2 Transition Path Sampling

After resolving our model system, we move to computational techniques of sampling the KARI catalysis process. While the majority of research in the field of enzyme
engineering ignores protein dynamics, we seek to create a sparse molecular model of catalysis with as few biasing preconceptions as possible [2]. As such, we leverage Transition Path Sampling (TPS).

The TPS sampling methodology provides an algorithm that, given an initial path, generates new paths through the corresponding energy landscape [63, 41, 17]. These new paths are then accepted or rejected, as in all Monte Carlo procedures, so as to generate a statistically correct ensemble. A critical component of this algorithm is called the shooting move. A shooting move takes as input a molecular dynamics simulation $\mathcal{M}$ of length $T$ from state $A$ to state $B$, which consists of the coordinates $r$ and momenta $p$ of each atom in the system for each time-step $t$ in the transition,

$$
\mathcal{M} = \{(r_t, p_t)\}_{t=0}^{T}, \ (r_0, p_0) \in A, \ (r_T, p_T) \in B.
$$

A timestep $t'$ is chosen at random, and the corresponding momenta $p_{t'}$ are modified slightly by a random perturbation $\delta$, such that $p_{t'} = p_{t'} + \delta p_{t'}$. A new trajectory is then simulated based on this point, both backwards and forwards in time, until either $A$ or $B$ is reached. If this new trajectory also connects $A$ and $B$, then it is accepted into the ensemble—otherwise, it is rejected. For more details, the reader is referred to [63, 41, 17]. The TPS algorithm was run on our KARI model system to generate $D$, an ensemble of reactive and non-reactive trajectories.

Figure 5-2: The alkyl migration catalyzed by the KARI enzyme which was simulated using TPS and studied in this work.
5.1.3 Molecular Features & Hypotheses

We began our analysis with \( D \), the results of the TPS simulations of the KARI enzyme as outlined in Section 5.1.2. We then defined our feature space \( X \) with \( n = 69 \) structural molecular features which we extracted from \( D \). Our resulting dataset spans \( o = 14 \) discrete order parameters and is approximately 13 gigabytes. The average number of reactive and non-reactive trajectories sampled at each order parameter was 425,012 and 410,829, respectively, with almost 1 million total trajectories sampled at order parameter 14 (see Figure A-4 for more detail).

For consistency across order parameters, we post-processed this dataset to include only those \( n = 81,148 \) trajectories that were sampled at every order parameter. Thus, our final dataset (schematized in Figure 5-3) consists of \( x_{i,j}^k \in \mathbb{R} \) for trajectory \( i = 1...m \), feature \( j = 1...n \), and order parameter \( k = 1...o \), as well as \( Y = (y_1...y_m)^T \), \( y_i \in \mathcal{Y} = \{0,1\} \), a boolean vector indicating whether each simulated trajectory successfully completed the reaction.

![Figure 5-3: Schematic of tensor with extracted features and response variables for each simulated trajectory and order parameter.](image-url)
5.2 Comparing Approaches

We turn now to a comparative analysis of the feature selection algorithms outlined above within the context of TPS simulations of the KARI model enzyme. In this section, we explore the differences between the selected features, as well as the classification accuracy enabled by each algorithm.

5.2.1 Features Selected

The top three features found by each algorithm across all order parameters are shown in Table 5.1. Some interesting trends are immediately apparent. First, the identification of feature 19 as the first feature is largely consistent across algorithms and order parameters. While this is striking, it is relatively unsurprising, as the first step of the CMIM, GCMI, and mutual information algorithms are all equivalent. Second, and also due to similarities between the algorithms, we notice that both of the greedy information theoretic approaches select the same second feature for every order parameter—feature 69 early in the reaction, and feature 65 later on. It is only at the third feature when the two algorithms begin to diverge. Third, we see that the algorithms that do not enforce sparse overlap in information are both similar in the features they select, and, as a group, largely different than the algorithms that do implement such a constraint.

5.2.2 Predictive Power

To investigate the ability of the identified features to describe KARI reactivity, we calculated the classification accuracy achieved with each set of features for each order parameter. Since our set of algorithms include feature selection methodologies that assume a linear model, such as LASSO, as well as methods that are sensitive to non-linear interactions, such as GCMI, we evaluated our features ability to predict reactivity using both logistic regression and classification tree models.

For each algorithm, the feature vector \( \theta \) of size \( z = 3 \) was identified. Then, both a 10-fold cross-validated logistic regression, as well as a 10-fold cross-validated
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMIM</td>
<td>1 19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>2 69</td>
<td>69</td>
<td>69</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td>GCMI</td>
<td>1 19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>2 69</td>
<td>69</td>
<td>69</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>3 10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>K-S</td>
<td>1 19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>2 5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>3 6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>25</td>
<td>19</td>
</tr>
<tr>
<td>LASSO</td>
<td>1 19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>31</td>
</tr>
<tr>
<td>MI</td>
<td>1 19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>Random</td>
<td>1 29</td>
<td>10</td>
<td>20</td>
<td>29</td>
<td>22</td>
<td>64</td>
<td>51</td>
<td>64</td>
<td>53</td>
<td>27</td>
<td>15</td>
<td>45</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>2 26</td>
<td>6</td>
<td>33</td>
<td>63</td>
<td>39</td>
<td>17</td>
<td>26</td>
<td>38</td>
<td>22</td>
<td>55</td>
<td>59</td>
<td>10</td>
<td>18</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3 63</td>
<td>69</td>
<td>9</td>
<td>33</td>
<td>18</td>
<td>37</td>
<td>35</td>
<td>1</td>
<td>51</td>
<td>31</td>
<td>5</td>
<td>36</td>
<td>4</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 5.1: The top three features identified by each algorithm for each order parameter \( k = 1 \ldots 14 \). Numbers indicate unique molecular features.
classification tree, were fit to $X_\theta$, the corresponding data. As our data is relatively well-balanced (Figure A-4), we quantified predictive ability by calculating the area under the ROC curve for each feature set, and plotted them in heatmaps. The results are shown in Figure 5-4.

Figure 5-4: Classification ability on the simulated KARI dataset using the top $z = 3$ features identified by each algorithm. The top row in each heatmap indicates the AUC achieved with just the top feature, while the subsequent rows are calculated using the indicated feature and all the features above (that is, the AUC shown in the second row, for example, is calculated using the top two features).

We note several takeaways from these results. First, we see that the classification tree is more accurate, on average, than the logistic regression model. Second, we note that the LASSO model performs best for the logistic regression task. This is not necessarily surprising, as LASSO also assumes a linear model. However, in the non-linear, classification tree setting, our GCMI algorithm is the best performing method—achieving over 95% average AUC for every order parameter with only three features.

To further investigate the relative performance between our GCMI algorithm and
the popular LASSO method, we calculated $\theta^{\text{GCMI}}$ and $\theta^{\text{LASSO}}$ of size $z = 5$ (Shown in Figure 5-5). We see that, in this non-linear setting, the GCMI algorithm continues to outperform all other feature selection methods for this increased feature set size.

In conclusion, we have shown not only that we are capable of identifying small sets of molecular features that can predict enzymatic reactivity well, but also that the GCMI algorithm represents a promising and high-performing framework with which to do so. Interestingly, this work highlights the likely importance of complex, non-linear interactions within our protein—which, as we saw in Chapter 3, the simpler feature selection algorithms are unable to detect.
Figure 5-5: Classification ability for $\theta^{GCMI}$ and $\theta^{LASSO}$ on the simulated KARI dataset for the $z = 5$ top features. Each row indicates the AUC achieved using the corresponding feature and all features above.
Chapter 6

Summary, Conclusion, & Future Work

In this chapter we present a summary of our work, outline the most notable conclusions, and mention potential avenues for future work.

6.1 Summary

In this thesis, we have explored the problem of feature selection in the context of protein engineering. In Chapter 1, we introduced a formalization of the problem, including the importance of identifying features that interact with the output only as a group, and discuss the unique requirements of our application domain. In Chapter 2, we outlined requisite theoretical fundamentals, including entropy, differential entropy, mutual information, and conditional mutual information. We also discussed the relationship between discrete and differential entropy, and introduced the MIST approximation. In Chapter 3, we outlined both statistical learning theoretical and information theoretical approaches to solving our problem, and discussed the intractability of the naïve approach in both cases. We then introduced the CMIM algorithm (and its descendents), as well as our GCMI algorithm, as greedy feed-forward approximations of the optimal solution. In Chapter 4, we conducted a comparative evaluation of the algorithms discussed on synthetic datasets. We first compared the features selected, and then
quantified the reduction in uncertainty achieved by each algorithm—after which we evaluated how this reduction in uncertainty translated to predictive power. We found that only our GCMI algorithm was able to consistently identify the correct features, and that this increase in accuracy resulted in both a decrease in $H(Y|X_o)$, the entropy of the output variable conditioned on the features selected, as well as an increase in classification performance. We also observed that the other feature selection algorithms investigated often failed to correctly account for information overlap between features. Finally, in Chapter 5, we applied our methods to a large dataset containing the dynamics of reactive and non-reactive enzymatic trajectories. We found not only that we were able to predict enzyme reactivity using a small, well-chosen subset of the feature space, but also that the GCMI algorithm, when coupled with non-linear classification models, resulted in the highest performance when predicting reactivity.

6.2 Conclusion

We have shown that the analysis of enzymatic trajectories can be accomplished in a principled way, and have proposed a new feature selection algorithm for the analysis of such data that has both nice theoretical properties and leading performance. We have also shown that that highly interacting groups of features are important not only in hypothetical situations, but also in real-world protein simulation datasets. Therefore, we conclude that the methods used in similarly complex, high-dimensional settings should address the quantification (and minimization) of information overlap between selected features—and that the GCMI framework presents an appealing avenue for such efforts.

6.3 Future Work

There is significant promise in the application of techniques from artificial intelligence to the field of protein design. Incorporation of dynamics and the leveraging of feature selection methods to identify critical drivers of the desired phenotype constitute an
important step forward. However, there remains a need to build out this capability—including, for example, methods of translating feature vectors to objective functions, optimizing over possible mutations based on these objective functions, and validating the results computationally.

Future work on feature selection, especially in large and highly-interaction applications, could focus on improving our ability to reliably and efficiently estimate high-dimensional information theoretic quantities—currently a bottleneck in our methodology. Entropy approximations, such as the MIST framework, could present good starting points for such efforts, and developing tools to quantify a dataset’s complexity—and then intelligently adjusting the approximation as appropriate—could significantly improve both the performance and popularity of such approaches.
Bibliography


Appendices
THIS PAGE INTENTIONALLY LEFT BLANK
Appendix A

Figures

Figure A-1: The normalized frequency of selection of each feature by each feature selection algorithm on $\Delta$. The first $\delta_v = 6$ features constitute the true feature set, and the final $\delta_c = 6$ are the corresponding noise corrupted features.
Figure A-2: Mean area under the ROC curve for logistic regression (LR), classification tree (CT), linear kernel support vector machine (LSVM) and radial basis kernel SVM (R SVM) on $\Gamma$ using the features identified by each indicated feature selection algorithm.
Figure A-3: Mean area under the ROC curve for logistic regression (LR), classification tree (CT), linear kernel support vector machine (LSVM) and radial basis kernel SVM (RSVM) on $\Delta$ using the features identified by each indicated feature selection algorithm.

Figure A-4: The number of reactive and non-reactive trajectories initially sampled at each order parameter.
THIS PAGE INTENTIONALLY LEFT BLANK
## Appendix B

### Tables

<table>
<thead>
<tr>
<th>Order Parameter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMMI</td>
<td>1</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>62</td>
<td>10</td>
<td>62</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>46</td>
<td>46</td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>39</td>
<td>62</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>5</td>
<td>5</td>
<td>46</td>
<td>36</td>
<td>3</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>8</td>
<td>39</td>
<td>39</td>
<td>54</td>
<td>54</td>
<td>7</td>
<td>54</td>
<td>3</td>
<td>3</td>
<td>25</td>
<td>25</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>39</td>
<td>7</td>
<td>3</td>
<td>24</td>
<td>54</td>
<td>54</td>
<td>3</td>
<td>33</td>
<td>22</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>68</td>
<td>65</td>
<td>65</td>
<td>64</td>
<td>3</td>
<td>54</td>
<td>3</td>
<td>24</td>
<td>31</td>
<td>25</td>
<td>5</td>
<td>22</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>10</td>
<td>54</td>
<td>54</td>
<td>10</td>
<td>39</td>
<td>64</td>
<td>10</td>
<td>22</td>
<td>24</td>
<td>36</td>
<td>46</td>
<td>5</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>33</td>
<td>46</td>
<td>18</td>
<td>8</td>
<td>22</td>
<td>10</td>
<td>57</td>
<td>57</td>
<td>50</td>
<td>54</td>
<td>22</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>GMMI</td>
<td>1</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>46</td>
<td>46</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>62</td>
<td>67</td>
<td>2</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>51</td>
<td>62</td>
<td>51</td>
<td>62</td>
<td>51</td>
<td>51</td>
<td>69</td>
<td>59</td>
<td>21</td>
<td>16</td>
<td>61</td>
<td>42</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>59</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

69
<table>
<thead>
<tr>
<th>K-S</th>
<th>LASSO</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>58 60 60 58 58 20 24 59 59 16 60 60 60 58</td>
<td>68 68 18 68 68 39 18 31 24 36 36 36 31 31</td>
</tr>
<tr>
<td>9</td>
<td>17 17 16 67 29 58 21 14 67 21 29 21 12 63</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>63 55 1 63 36 24 55 63 58 17 62 23 55 67</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>19 19 19 19 19 19 19 19 19 19 19 19 19 36</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5 5 5 6 6 6 6 62 62 62 62 62 62 25 19</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6 6 6 5 5 62 6 24 24 25 25 25 25 32 31</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>39 39 39 39 39 24 5 5 31 31 31 31 31 31 36 32</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>23 8 8 8 24 39 39 25 25 6 36 36 36 62 62</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8 23 50 50 27 27 27 32 32 24 6 30 30 30 30 30</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>50 50 54 54 41 41 41 61 5 36 30 53 53 37</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>33 33 33 27 50 50 50 31 61 5 5 22 37 20</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>26 54 23 41 8 48 61 27 50 61 53 37 20 53</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>19 19 19 19 19 19 19 19 19 19 19 19 19 25</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5 62 62 62 62 39 24 24 31 62 36 36 31 19</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>62 6 6 6 6 6 6 39 31 24 24 62 62 36 30</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6 5 8 8 24 24 6 50 50 32 32 30 30 30 37</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>23 8 18 18 50 20 27 27 36 36 45 45 62 62</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7 18 7 50 8 27 31 6 45 45 50 50 45 32</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8 7 50 7 18 50 50 39 53 50 24 53 50 53</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>18 23 23 24 20 8 8 45 46 53 30 46 53 45</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>50 50 46 46 46 18 18 8 27 46 53 37 50</td>
<td></td>
</tr>
</tbody>
</table>
Table B.2: The top \( z = 10 \) features identified by each algorithm for each order parameter \( k = 1...14 \). Numbers indicate unique molecular features.
<table>
<thead>
<tr>
<th>Order Parameter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order Parameter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order Parameter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order Parameter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>62</td>
</tr>
<tr>
<td>5</td>
<td>23</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order Parameter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order Parameter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>59</td>
<td>48</td>
<td>40</td>
<td>26</td>
<td>62</td>
<td>69</td>
<td>17</td>
<td>61</td>
<td>34</td>
<td>18</td>
<td>53</td>
<td>66</td>
<td>49</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
<td>20</td>
<td>20</td>
<td>36</td>
<td>24</td>
<td>65</td>
<td>38</td>
<td>34</td>
<td>58</td>
<td>58</td>
<td>49</td>
<td>40</td>
<td>43</td>
<td>56</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>54</td>
<td>54</td>
<td>24</td>
<td>60</td>
<td>64</td>
<td>45</td>
<td>38</td>
<td>41</td>
<td>41</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>66</td>
<td>46</td>
<td>37</td>
<td>48</td>
<td>19</td>
<td>53</td>
<td>20</td>
<td>57</td>
<td>12</td>
<td>52</td>
<td>41</td>
<td>20</td>
<td>58</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>14</td>
<td>26</td>
<td>3</td>
<td>34</td>
<td>3</td>
<td>30</td>
<td>16</td>
<td>22</td>
<td>11</td>
<td>22</td>
<td>29</td>
<td>46</td>
<td>7</td>
</tr>
</tbody>
</table>

Table B.1: The top $z = 5$ features identified by each algorithm for each order parameter $k = 1...14$. Numbers indicate unique molecular features.