Nonlinear Measures of Correlation and Dimensionality Reduction with Application to Protein Motion

REU Site: Interdisciplinary Program in High Performance Computing

Nancy Hong[1], Emily Jasien[2], Christopher Pagan[3], Daniel Xie[4],
Graduate assistant: Zana Coulibaly[5], Faculty mentor: Kofi P. Adragni[5], Client: Ian F. Thorpe[6]

[1]Department of Mathematics, Stony Brook University
[2]Department of Mathematics, California State Polytechnic University, Pomona
[3]Department of Computer Science and Electrical Engineering, UMBC
[4]Division of Natural Sciences, New College of Florida
[5]Department of Mathematics and Statistics, UMBC
[6]Department of Chemistry and Biochemistry, UMBC

Technical Report HPCF–2014–11, www.umbc.edu/hpcf > Publications

Abstract

The study of allostery, a regulatory process that occurs in complex macromolecules such as proteins, is of particular interest as it has a key role in determining the function of these macromolecules. Allostery produces motional correlations that can be analyzed using different statistical methods. We implement a program in the statistical programming language R that uses polynomial regression and leave-one-out cross-validation to model relationships in data obtained from different sites in the protein, using the square root of the coefficient of determination to detect both linear and non-linear trends. The performance of the program will be studied on a simulated data set with linear and non-linear relationships and the effectiveness of the implemented methods as it relates to this problem will be assessed.

1 Introduction

Researchers today are very interested in the study of complex biomolecules, such as proteins, as they are immensely important to living organisms. Proteins are of particular interest as they perform many functions within an organism that are integral to its survival. Within a protein, a process called allostery can occur which produces a significant effect on the function of that protein. As this process is not completely understood, the study of allostery has become a focus in the field and has also brought about an interest in the study of motional correlations that occur in proteins as a result of allostery.

Statistics can be used to assist in determining the motional correlations in proteins to further our understanding of allostery. Various statistical methods can be applied to simulated data representing the motion at different protein sites, and one such method will be discussed in this paper. We will discuss and also implement polynomial regression and leave-one-out cross-validation to model relationships in data using a polynomial of optimal degree, and then uses the square root of the coefficient of determination to detect both linear and non-linear trends multiple linear regression to generate predictions for the data and then calculate the correlations between the predicted and the actual data. The data set that the

method will be tested on will be one in which the correlations between the variables are known so that the accuracy of this method can be tested.

2 Background

2.1 Allostery

Allostery is a process that allows complex macromolecules, such as proteins, to sense and react to changes within the environment. Different regions in a macromolecule are thermodynamically or structurally coupled, allowing information to be communicated between regions that are often distant from each other. An event that occurs at a particular region in the macromolecule can therefore create a change at the region to which it is coupled. In proteins, this allows for the properties at an active site to be altered by some action that occurs at the site that is coupled to the active site [**?**]. Allostery will therefore result in motional correlations between measured properties at distant sites in a protein. As an example of such activity, these movements can be linked with the binding and folding of proteins which have a strong role in determining their functional roles. Allostery is important because the properties of the macromolecules are altered, such as their structure and stability [**?**]. There is an interest in finding and explaining the allosteric processes of proteins as it can help us better understand these complex molecules. This paper will deal with finding and using methods to identify the motional correlations of particles within proteins and biomolecules, specifically for applications to the RNA polymerase from the Hepatitis C virus (HCV).

2.2 Measures of Correlation

To discover a relationship between the motions of different regions of macromolecules, we desire a suitable measure of correlation. The most commonly used correlation coefficient, the Pearson correlation coefficent, is suitable only for measuring linear trends between two random variables. Given two random variables, x and y, Pearson's population correlation coefficient ρ_{xy} is defined as

$$\rho_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{\mathrm{var}(x)\mathrm{var}(y)}} \tag{2.1}$$

The population correlation coefficient takes values between -1 and 1 and is a measure of a linear relationship between x and y. The higher the magnitude of ρ_{xy}, the stronger the relationship, where a magnitude of 1 indicates a perfect linear relationship between x and y.

It is nearly impossible to calculate the population correlation coefficient ρ, so a sample correlation coefficient is calculated to estimate ρ. The sample correlation coefficient r is defined as

$$r_{xy} = \frac{s_{xy}}{\sqrt{s_x^2 s_y^2}} \tag{2.2}$$

where s_{xy} is defined as the sample covariance of x and y and s_x^2 and s_y^2 are the sample variance of x and y respectively.

The Pearson correlation coefficient is suitable only for measuring linear trends between two random variables. When dealing with spatial information, however, we need a measure of dependency that is able is able to detect evolving, nonlinear relationships between two given random variables.

This problem can be mitigated by noting that in the case of a simple linear regression, the absolute value of the Pearson correlation coefficient between the regressor and the true value of the regressand will be the same as that of the predicted values generated for the regressand and the true values. This results from the Pearson correlation coefficient being invariant up to a sign change from scaling and invariant under the addition of a constant value to a random variable. In general, the correlation coefficient between the predicted values from a regression and the observed values is known as R, the square root of the commonly used coefficient of determination. With this knowledge, one can design a correlation coefficient more adept at detecting complex relationships by performing several multiple linear regressions where the regressors are transformed versions of one of the random variables that are able to approximate a larger class of functional forms. By choosing the R value from the best combination of regressors as chosen by cross-validation and using that as our measure of correlation, we now have a correlation coefficient robust enough to pick up linear and non-linear trends. Note that in doing so, we now sacrifice the symmetry of the correlation coefficient; this new measure of correlation will in general return different results when the random variable chosen for the regressors and regressand are switched.

Having decided this, we should choose basis functions for our transformations that are able to approximate broad classes of different functions. A natural choice is a polynomial basis: it is well known that functions defined on a closed interval with any minor degree of smoothness can be uniformly approximated by polynomials of increasing degree (the famed Weierstrass approximation theorem). Thus, the regressions we perform will be polynomial regressions, where the order of the polynomial used to calculate the correlation coefficient will be chosen using the mean squared predictive error from k-folds cross-validation.

The data that is tested for correlations will be simulations of the movement of p molecules within a single protein. We have p random variables, $x_1, x_2, ..., x_p$ that represent the random locations of each molecule in the protein. The data contains n observations for each x_i so that our data can be represented as a (nxp) matrix. The goal of this project is to obtain a measure of the correlation among the p molecules which will be presented in a pxp matrix R of that correlation between each x_i and x_j where $i \neq j$.

3 Methodology

3.1 Random Vectors

Given a finite sequence of random variables $x_1, x_2, \ldots, x_p$, we will find it convenient to express them in the form of a vector $\mathbf{x}$, where $\mathbf{x} = (x_1, x_2, \ldots, x_p)^T$. Then for such an $\mathbf{x}$, define

$$\mathrm{E}(\mathbf{x}) = (\mathrm{E}(x_1), \mathrm{E}(x_2), \ldots, \mathrm{E}(x_p))^T \tag{3.1}$$

and define

$$\text{cov}(\mathbf{x}) = \begin{pmatrix} \text{var}(x_1) & \text{cov}(x_1, x_2) & \cdots & \text{cov}(x_1, x_p) \\ \text{cov}(x_2, x_1) & \text{var}(x_2) & \cdots & \text{cov}(x_2, x_p) \\ \vdots & \vdots & & \vdots \\ \text{cov}(x_n, x_1) & \text{cov}(x_n, x_2) & \cdots & \text{var}(x_p) \end{pmatrix} \tag{3.2}$$

We will refer to $\text{cov}(\mathbf{x})$ as the *covariance matrix* for the random variables, which will usually be denoted by $\mathbf{\Sigma}$. Note that $\mathbf{\Sigma}$ will always be symmetric as $\text{cov}(x_i, x_j) = \text{cov}(x_j, x_i)$.

3.2 Linear Regression

Simple linear regression attempts to model a linear relationship between a dependent variable y and a single independent variable x. We will frequently refer to y as the regressand and x as a regressor. We assume the relationship between y and x is of the form

$$y = \beta_0 + \beta_1 x + \varepsilon \tag{3.3}$$

where ε represents error accompanying the model due to random fluctuations or errors in measurement and β_0 and β_1 are unobservable parameters. For n observations, our model then becomes.

$$y_i = \beta_0 + \beta_1 x + \varepsilon \tag{3.4}$$

for $i = 1, 2, ..., n$.

Given data that fits the model 3.4, we will derive estimates of β_0 and β_1 using the method of least squares. The following assumptions are used in the classical linear regression model to provide favorable results about our regression estimates.

1. $\text{E}(\varepsilon_i | x_1, x_2, \ldots, x_p) = 0$
2. $\text{var}(\varepsilon_i | x_1, x_2, \ldots, x_p) = \sigma^2$
3. $\text{cov}(\varepsilon_i, \varepsilon_j | x_1, x_2, \ldots, x_p) = 0$ for $i \neq j$

The assumptions state that given $x_1, x_2, \ldots, x_p$, y_i is only dependent on x_i, the variance of y and ε are constant and thus not dependent on x_i(homoscedasticity), and ε_i is not correlated with ε_j if $i \neq j$.

Multiple linear regression extends simple linear regression and attempts to model a dependent variable y on the assumption that it has a linear relationship with p independent variables $x_1, x_2, ..., x_p$. Again, y will commonly be referred to as the regressand and $x_1, x_2, ..., x_p$ as regressors. We assume the relationship between y and $x_1, x_2, ..., x_p$ takes the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \varepsilon \tag{3.5}$$

Given n observations, we represent our model as

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_p x_{pi} + \varepsilon_i \tag{3.6}$$

Simple linear regression is now seen to be a case of multiple linear regression when only one regressor is included in the model. The multiple linear regression equations for n observations may be expressed in matrix form thusly:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

or, more compactly,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.7}$$

where $\mathbf{X}$ is an $n \times (p+1)$ matrix known as the design matrix. When generalized to multiple linear regression, the classical assumptions become

1. $\mathrm{E}(\boldsymbol{\varepsilon}|\mathbf{X}) = \mathbf{0}$
2. $\mathrm{cov}(\boldsymbol{\varepsilon}|\mathbf{X}) = \sigma^2\mathbf{I}$

Our goal is derive estimates for the unobserved parameters $\boldsymbol{\beta}$. The estimator we will used will be based on minimizing the sum of squares of the prediction errors derived from our estimates. We must additionally assume that $\mathbf{X}$ has full column rank to carry out our calculations. If $\hat{\boldsymbol{\beta}} = (\beta_1, \beta_2, \ldots, \beta_p)^T$ is an estimator of β, then the value of $\hat{\boldsymbol{\beta}}$ that minimizes $(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$ is given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\mathbf{y} \tag{3.8}$$

This specific estimator $\hat{\boldsymbol{\beta}}$ is referred to as the *ordinary least squares estimator* for $\boldsymbol{\beta}$. As long as the classical assumptions hold, $\hat{\boldsymbol{\beta}}$ has the minimum variance of all unbiased estimators for β.

There is a special type of multiple linear regression known as *polynomial regression* where we consider a model of the form

$$y = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + ... + \beta_p x_i^p + \varepsilon_i \tag{3.9}$$

All of the standard results about multiple linear regression hold with polynomial regression; the models that we use to generate predictions and calculates measures of correlation will all be of this form.

3.3 Correlation Coefficients

The multiple correlation coefficient R is a generalization of r. R can be defined as

$$R = r_{y\hat{y}} \tag{3.10}$$

where $r_{y\hat{y}}$ is the simple correlations of observed y_i variables and predicted $\hat{y}_i$ variables. The multiple correlation coefficient R can also be expressed geometrically as the cosine of θ, the angle between $\mathbf{y} - \bar{y}\mathbf{j}$ and $\hat{\mathbf{y}} - \bar{y}\mathbf{j}$ which are the centered forms for $\mathbf{y}$ and $\hat{\mathbf{y}}$, where $\mathbf{j}$ is the vector containing all 1's in its entries. The cosine of θ can be written as

$$\cos\theta = \frac{\sqrt{(\hat{\mathbf{y}} - \bar{y}\mathbf{j})'(\hat{\mathbf{y}} - \bar{y}\mathbf{j})}}{\sqrt{(\mathbf{y} - \bar{y}\mathbf{j})'(\mathbf{y} - \bar{y}\mathbf{j})}} = R \tag{3.11}$$

3.4 Leave-One-Out Cross-Validation

Cross-validation divides data into two categories in order to compare the accuracy of different models. One category is used to train the model while the other category tests the model created from the training category. K-fold cross-validation is one of the most common used forms of cross-validation and it requires the data to be divided into k bins with equal amounts of data. There are k iterations that occur in k-fold cross-validation. For each iteration, one bin of data is taken to be the testing category while the other $k-1$ bins are part of the training category [**?**]. The leave-one-out cross-validation (LOOCV) is a type of k-fold cross-validation where k equals the number of observations of the data set.

In this paper, k-fold cross-validation is used to determine the optimal degree of a linear regression model. Using the data from the $k-1$ bins of the training category, a linear regression model is created. The linear regression model is then used to predict the dependent variables associated with the independent variables in the testing category. After the cross-validation is performed, the predicted dependent variables are used along with the true dependent variables to find the prediction error of that linear regression model. The prediction error is determined by

$$pe = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n} \tag{3.12}$$

The optimal degree can be found by finding the linear regression model with the minimal prediction error.

3.5 Implementation of the Algorithm in R

We use the statistical programming language R in order to calculate the optimal degree of the linear models and to produce the matrix of correlations between all random variables to be tested. In order for the code to run properly, we use the libraries "MPV" for the cross validation, "Rmpi" and "snow" to enable the code to run on a parallel computing system, and "gplots" to create heatmaps. The code is split up into functions that all work together within one function called "cor.mat". The subfunctions that work together within "cor.mat" are named as follows:

1. polyreg
2. polyreg.kfolds.pe
3. optimal.degree
4. unicor

All of the functions listed above are called within each other, with "cor.mat" calling "unicor", "unicor" calling "optimal.degree", etc. The first function that performs the calculations is "polyreg". Given a regressor, regressand, and degree, a linear model and the degree of the polynomial are calculated and combined into a list. The code then proceeds to find the optimal degree for which the polynomial models are fitted to the data by "optimal.degree".

This function calls "polyreg.kfolds.pe" to calculate the prediction error. "polyreg.kfolds.pe" obtains this result by calculating the PRESS (prediction residual sum of squares) and dividing PRESS by the total number of observations. The PRESS score is the sum of squares that are a result of LOOCV. After the PRESS scores for each model are found, the model of with the degree that generates the lowest prediction error is returned. With this degree, "unicor", calculates the correlations between this optimal model with the actual values obtained and returns this correlation.

The code's inner workings are all contained in the "cor.mat" function. The first task that R must perform is to create two matricies that hold the values of interest, which are the correlations between the set of random variables and the optimal degree of polynomials that would best correspond to their linear regression models.

A random variable's correlation between itself is always 1, and it is trivial to perform a regression of a random variable on itself. Because of this the entries along the diagonals of the matrix for the regression scores, named "scores", and the matrix for the optimal degrees, named "opt.deg", are all 1 and NA, respectively. After both the matrix for correlation scores and optimal degree have been calculated, the results are printed to the screen for users to see.

4 Results

5 Conclusions

Acknowledgments

These results were obtained as part of the REU Site: Interdisciplinary Program in High Performance Computing (www.umbc.edu/hpcreu) in the Department of Mathematics and Statistics at the University of Maryland, Baltimore County (UMBC) in Summer 2014. This program is funded jointly by the National Science Foundation and the National Security Agency (NSF grant no. DMS–1156976), with additional support from UMBC, the Department of Mathematics and Statistics, the Center for Interdisciplinary Research and Consulting (CIRC), and the UMBC High Performance Computing Facility (HPCF). HPCF is supported by the U.S. National Science Foundation through the MRI program (grant nos. CNS–0821258 and CNS–1228778) and the SCREMS program (grant no. DMS–0821311), with additional substantial support from UMBC. Co-author Christopher Pagan was supported, in part, by the UMBC National Security Agency (NSA) Scholars Program through a contract with the NSA. Graduate assistant Zana Coulibaly was supported during Summer 2014 by UMBC.