AMATH 563 HW 2

Katie Johnston

May 6, 2020

Abstract

Finding a model from data is an invaluable skill that has endless applications. In the project, we explore two data sets over a number of data driven methods. These methods include Dynamic Mode Decomposition (DMD), time delayed embedding DMD, and sparse regression. We also compared our models using KL divergence, AIC and BIC. We found that some of our models were good at interpolating our data, but none of our models were particularly good at extrapolating to new data.

1 Introduction and Overview

Being able to find a model from raw data is an invaluable skill with an endless number of applications. In this project we explore a number of methods and data sets in order to learn methods to find a model from data. We explored two oscillatory data sets, first on populations of hare versus lynx and a second on video from an oscillating chemical reaction. We also explored a number of methods for finding models including Dynamic Mode Decomposition (DMD), Time Delayed Embedding DMD, the Lotka-Volterra model, and Sparse Regression. The details of these models will be explained in subsequent sections along with explanations of the pro and cons of each method.

2 Theoretical Background

We explored our data on a variety of methods to find a model that represented the data well. These methods included: Dynamic Mode Decomposition (DMD), Time Delayed Embedding DMD, the Lotka-Volterra model, and Sparse Regression. This section will give a brief theoretical overview of each method.

Dynamic Mode Decomposition (DMD) is a method to predict the next time step of a model and is a highly data drive method. We will do exact DMD. First, we obtain many snapshots of the data and then we want to find a the best fit linear operation such that $X' \approx AX$, where X is a matrix of our data, and X' is a matrix of our data off set by one. The best fit operator will be $A = X'X^{\dagger}$, which we can use the SVD to help us calculate as $A = X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^*$. Next, we will find the eigenvalues of A as $AW = W\Lambda$, and the columns of these eigenvalues will help us find the DMD modes, Φ given by $\Phi = X'\tilde{V}\tilde{\Sigma}^{-1}W$. Once we have the DMD modes, we can construct a data-driven spectral decomposition given by $x_k = \Phi\Lambda^{k-1}b$, where Φ is the DMD modes, Λ is the eigenvalues, and b is the mode amplitude.

Many times, there could be latent variables impacting the data because we don't (or cannot) always measure all relevant variables. Time Delayed Embedding is one method to attempt to find these latent variables. In order to time delay embed, we take many realizations of our data, all offset by one time step. We can then look at the SVD of this matrix, and the singular values will help us see which and how many modes are important. Once we have the time delayed matrix, we can again preform DMD to find a spacial expansion of our system.

Another model that we might want to have is a model that can be written in a closed form with a limited number of terms. A common predator-prey model with a small number of terms is the Lotka-Volterra model. This model is given a system of differential equations:

$$\dot{x} = (b - py)x \tag{1}$$

$$\dot{y} = (rx - d)y \tag{2}$$

Another, more problem specific, method of finding a model is using sparse repression. The idea of sparse regression is to find a model that best fits the model with as limited number of terms as possible. A sparse model is ideal because we want to be on the Pareto Frontier, and we want a model that generalizes well to new data.

Once we have our models, we want to find the "best" model. "Best" is not a well defined word that we need to define. We consider our "best" model to be the model that performs the best on new data. Cross-validation is on way of finding the best model, where we calculate the error on a test data set that was not trained on, and the model with the lowest test data is the best model. Another method is to use KL divergence or information criterion. KL divergence is given by

$$I(f,g) = \int f(X,\beta)\log\left[\frac{f(X,\beta)}{g(X,\beta)}\right]dX \tag{3}$$

and measures the distance between two probability distributions. There are also two kind of information criterion which we will consider: Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). Both of these estimate the distance between the true model and the calculated model, as well as penalize for the number of terms (but in different ways). AIC is given by

$$AIC = 2K - 2\log[\mathcal{L}(\hat{\mu}|x)]$$

and BIC is given by

$$BIC = \log(n)K - 2\log[\mathcal{L}(\hat{\mu}|x)]$$

To calculate the log likelihood of a model, we can use the equation:

$$\log\mathcal{L}(\mu,\sigma) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2 \tag{4}$$

3 Algorithm Implementation and Development

First we did a little preprocessing of our data. For the hare and lynx data, we had a very limited number of data points. Thus, before doing anything, we used spline interpolation in order to increase our number of data points. We also adjusted our years, so that the data started at year 0, not year 1845. For the BZ tensor data, we didn't change anything, but only considered the first 200x200 square of x and y data points so that the algorithms did not take so long on my laptop.

For DMD we used the example code from the textbook. We used the first half of the data (30 points for lynx and hares) as our training set and the second half (29 points) for our testing set. For time delayed embedding, we made our H matrix using a variety number of time delays. After we predicted our x matrix using our results of DMD, we used the first two modes as the hare and lynx respectively and used the first mode as the BZ tensor data.

Next, we wanted to find a good estimate for the Lotka-Volterra model. First we needed a good estimate for the derivatives of the two data sets. A few derivative methods were explored, without observably significant changes to the model. Thus, the following equation was used as an estimate for the derivative:

$$\frac{dx_i}{dt} \approx \frac{x_{i+1}-x_{i-1}}{2\Delta t} \tag{5}$$

Once we had our estimate for the derivative, which we stored in our b, we made a matrix A, which contained $[x_1, x_2, x_1x_2]$. We then did least squares to solve $x = A/b$, and set the x values to the desired parameters. Thus, we had our Lotka-Volterra model.

Next, we wanted to find a sparse regression models. Similar to above, we made our matrix b which contained our estimates of the derivatives. Then we built our A library which contained all polynomials of x_1 and x_2 up to fourth order as well as an assortment of sine and cosine functions. First we solved $x = A/b$ using least squares, which gave us a weighting of all our terms. However, we want a sparse model. One option for finding a sparse fitting would be to use Lasso with increasing α. However, this was producing undesirable results that would either blow up or go negative. Thus, we picked our top weights in our x vector, kept those terms, and redid least squares. This was repeated a few times until a desirable sparse model was found.

Next, we compare the different models to each other. For the hare and lynx data set, we used KL divergence, AIC and BIC. We calculated these based on the equations presented in the previous section. We trained our data on the first half of the data (30 data points) and tested on the second half of the data (29 points). Thus, we did these calculations on just our predicted data from our testing data set. For the BZ tensor data set, we just used the relative 2-norm error on the testing set, and found the rank of DMD with the smallest testing error.

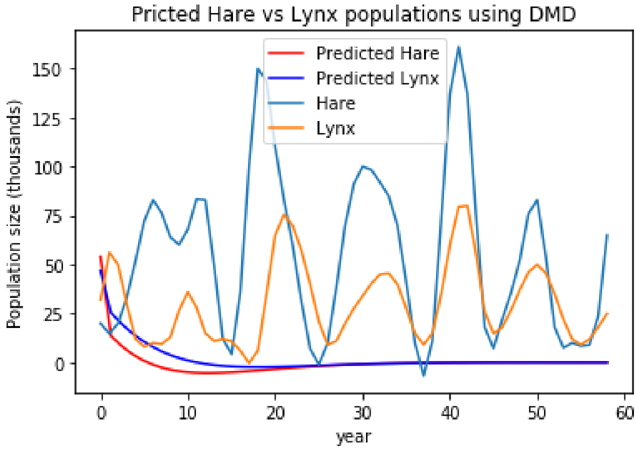


Figure 1: Predicting Hare and Lynx using DMD. First half was used for training and second half of data was used for testing.

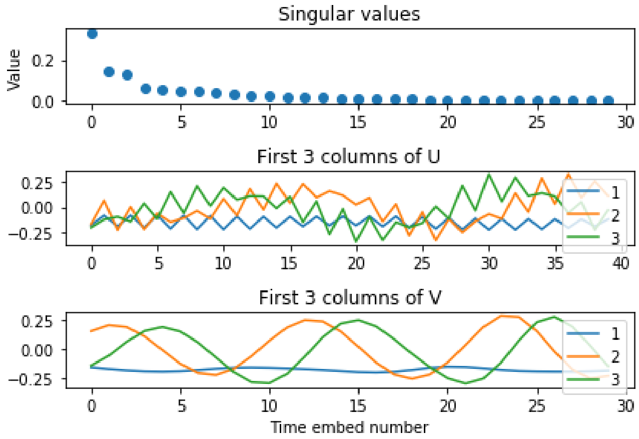


Figure 2: The SVD of the time delay embedding matrix H. In the first plot, we can see that there is one main mode, two also significant modes, and then many nonzero modes. From the first three column of V we can see that the main mode is constant, and the next two modes resemble the hare and lynx data, with a similar period and the lynx being slightly delayed from the hares.

4 Computational Results

First we explored the hare and lynx data set. The first method we used to find model this data set was DMD, shown in Fig (1). Here you can see that the results are not matching the data very well at all, and both populations quickly decay to zero. Next, we used time delayed embedding to improve on our exact DMD. In Fig (2), you can see the results of the SVD from the matrix formed by the time embedding. You can see that there are is one main mode, and two modes that are also really large. There is also a significant number of terms that have a significant nonzero weight. This implies that there are latent variable in our data. In the plot that shows the columns of V, you can see that the main mode is roughly constant, and the next two modes are oscillatory, slightly delayed from each other much like how the raw data from the hare and lynx is, and has a period roughly equal to that of the raw data. In Fig (3), you can see the results of the time delayed embedding DMD, and you can clearly see that these results follow the data much more closely than the exact DMD. You can also see that as time increases, the model is following the data less and less. The time delayed embedding model fits considerable better than exact DMD, this also helps to show that there are latent variables affecting these populations.

Next, we want to find a sparse model that we can write in a closed form. First, we fit our model to the Lotka-Volterra model. You can see those results in Fig(4). From that figure, you can see that the results are not

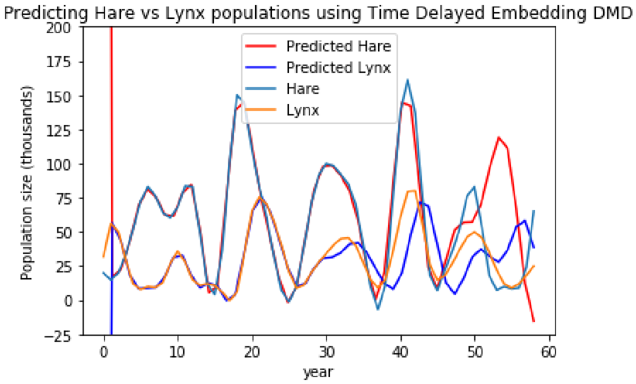


Figure 3: Predicting Hare and Lynx using time delayed embedding DMD. First half was used for training and second half of data was used for testing.

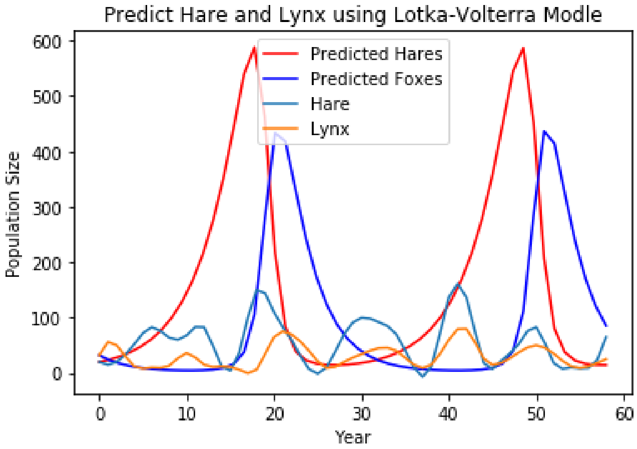


Figure 4: Predicting Hare and Lynx using the Lotka-Volterra Model. First half was used for training and second half of data was used for testing.

very great, but you can see the oscillatory behavior where the hare peak a little before the lynx. However, the period is much too large on our predicted model and the amplitude is much to large.

Next, we used sparse regression with a much larger library size. Figure (5) shows a plot of our model when all terms in the library are used. Here you can see that the model does an excellent job on the training data, but quickly diverges when trying to predict new data. This implies that we are overfitting and have too many terms. Figure (6) shows a plot of a model with only two terms. Here we can see that the model does not do a great job, but it also does not get worse when during the testing set. We also see that the population of hares quickly goes less than zero. Ideally we would want to have a model that restricted this kind of impossible behavior.

Once we have our models we want to compare them to each other to find the "best" model. Table (1) shows the KL divergence, AIC and BIC scores for our top 3 models. Based on KL divergence, the Lotka-Volterra model looks the best, very closely followed by the time delayed embedding DMD. However, when using either AIC or BIC, the sparse regression model is determined to be the best model. This makes it hard to choose a "best" model.

Next, we looked at the BZ tensor data. We used exact DMD and time delayed embedding DMD to explore this data set. In Fig (7) you can see the results at time 150, which was the end of the training data. You can see that both DMD and time delayed embedding produced nice results at the end of the training time. In Fig (8), you can see the results after 300 times steps. Here you can see that the results are not looking very great. This is somewhat expected because we would anticipate that our model would do much better interpolating than extrapolating. You can see that there is a little of the feature in the bottom right corner, but not very many of the other features. You can see that the model which used time delayed embedding has sharper lines than

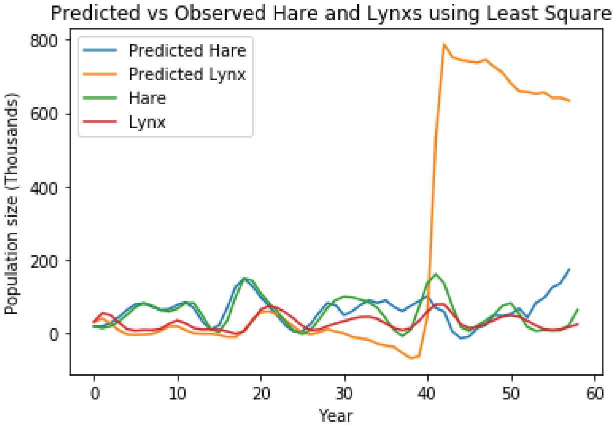


Figure 5: Predicting Hare and Lynx using Least Squares Regression. First half was used for training and second half of data was used for testing. You can see that this model interpolates very well, but then blows up on the testing set. This implies that our model is overfitting.

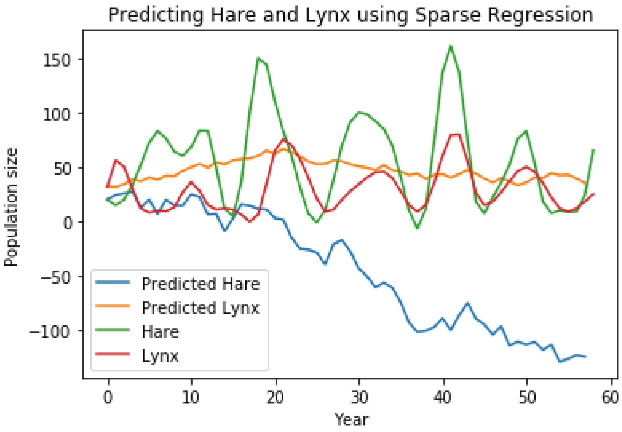


Figure 6: Predicting Hare and Lynx using sparse regression with 2 terms. First half was used for training and second half of data was used for testing. This model is predicting negative hares, which we know is impossible.

Model	KL Divergence		AIC		BIC	
	Hare	Lynx	Hare	Lynx	Hare	Lynx
Time Delayed	0.214	0.222	4309.615	2207.415	5456.829	3354.629
Lotka-Volterra	0.207	0.222	12520.307	4615.705	12527.955	4623.352
Sparse Regression	1.419	1.283	2366.910	2155.219	2441.512	2229.823

Table 1: Table of KL divergence, AIC, and BIC for top 3 models of Lynx and Hare model

regular DMD. However, there is not a drastic difference with and without time delay embedding. This implies that we did not discover any latent variables for this data set

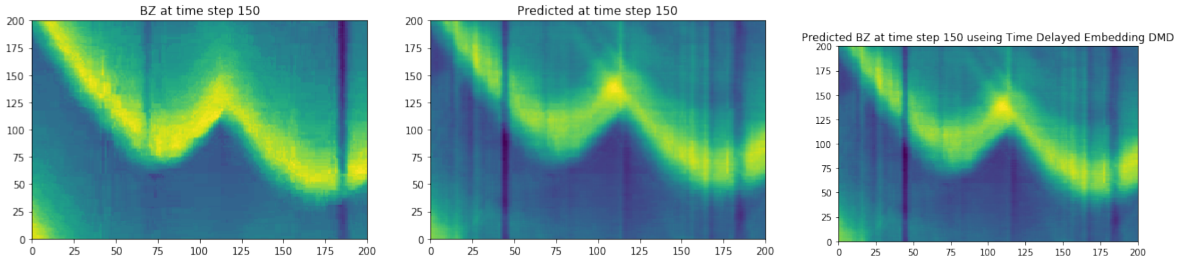
Also with DMD, there is a large question of what rank to use. Fig (9) shows the testing error on our data set. Here you can see that rank 26 has the lowest testing error before the error blows up. This blow up happens because we are overfitting our model. Thus, the above plots were made using DMD with rank 26. It is also interesting that the error does not greatly decrease with higher rank before 26. This implies that there are a few very important variables in this data set. For time delayed embedding, we looked at the SVD of the time delayed matrix. From the singular values of this matrix, we were able to see that there was one very important mode, and many much less important modes. For time delayed embedding DMD, we pick a rank of 30 because there was a drop in the magnitude of the singular values after 30 modes.

5 Summary and Conclusions

In this project, we explored two data sets, one on populations of hare vs lynx and another on BZ tensor. Both data sets were oscillatory in nature, and we were able to find some of that oscillatory behavior using DMD and sparse regression.

For the hare and lynx data set, we found a number of models which predicted the testing data to a varying degree. Some of the models did a good job interpolating the data, but none of the models did a particularly good job extrapolating the data. We calculated the KL divergence and AIC and BIC scores of our models and discovered that based on KL divergence the Lotka-Volterra model is the "best" while using AIC and BIC, the sparse regression model is the "best" for extrapolating data. However, neither of these models fit very well. For interpolating data, the Time Delayed embedding DMD model and the least squares regression model looked to be preforming the best.

We also explored the BZ tensor data set. We found that DMD with a rank of 26 had the lowest testing

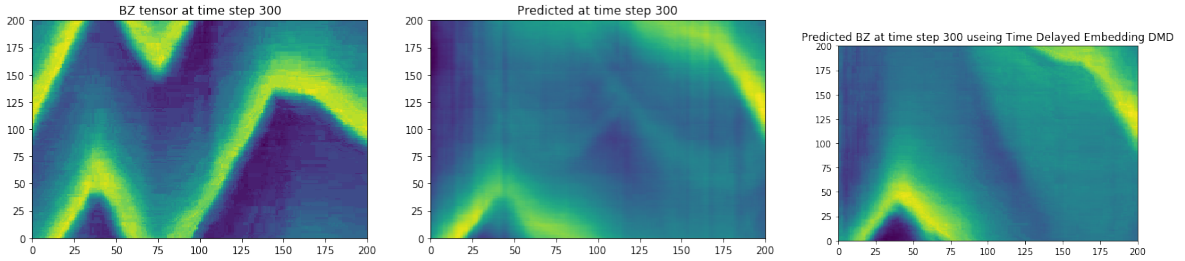


(a) True Data

(b) Predicted using DMD

(c) Predicted using time delayed embedding DMD

Figure 7: Predicting the BZ tensor at time step 150, the end of the training set



(a) True Data

(b) Predicted using DMD

(c) Predicted using time delayed embedding DMD

Figure 8: Predicting the BZ tensor at time step 300

error and then overfitting occurred. We also found a model using time delayed embedding. We did not notice significant difference between these two models, which suggests that we did not find an latent variables. We also see very good agreement at the last time step of our training set, which implies that we are interpolating our data well. However, we are not observing as good of behavior on our testing set, although some features are still picked up. This implies that we are not doing a very good job extrapolating the data.

If I had the time to continue this project, there are many areas that we could add. For the hare and lynx data set, we could explore using a variety of method to estimate the derivative, try different/larger libraries for our sparse regression, and try and find a sparse model with restrictions so the population cannot go negative. For the BZ tensor data, we only used a small snip pit of the data in order to increase speed of the calculations. It would be good to try to model a larger area or see if our model fit on different areas well.

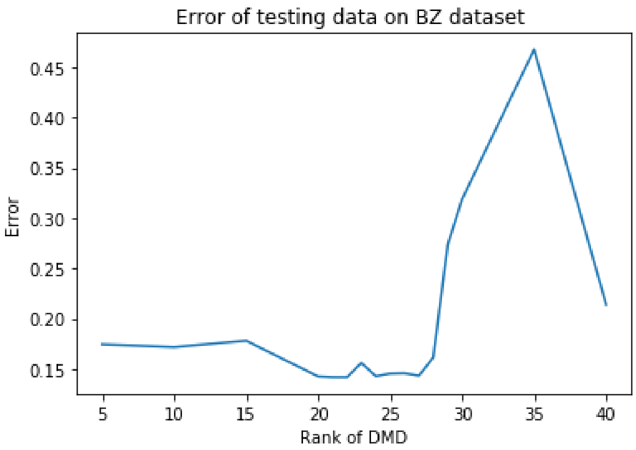


Figure 9: Plot of the testing error of the BZ tensor data set calculated using DMD with varying rank.

A Python functions used and brief implementation explanation

Not really any custom functions made. Used DMD from the example code. Used a variety of numpy and scipy methods for various parts of the calculations. Used matplotlib.pyplot and pcolor for graphics.

B Python Code

Please see Github for code: `https://github.com/KatieJ16/AMATH563/tree/master/HW2`

Notebooks included:

- `HW2-Spline` Contains code for the hare vs lynx data set using spline interpolation.
- `HW2-BZ` Contains code relevant to the BZ tensor data set.