Calculating the Lyapunov Exponent
Time Series Analysis of Human Gait Data

Rodney Gaines
Anthony Zittle

Louisiana State University

1. Introduction

Peripheral neuropathy causes pain and numbness in your hands and feet, usually described as tingling or burning. While the loss of sensation, is often compared to the feeling of wearing a thin stocking or glove. Peripheral neuropathy can result from such problems as traumatic injuries, infections, metabolic problems and exposure to toxins. One of the most common causes of the disorder is diabetes.

In many cases, peripheral neuropathy symptoms improve with time. Medications, initially designed to treat other conditions, are often used to reduce the painful symptoms of peripheral neuropathy.

Peripheral neuropathy is a progressive deterioration of peripheral sensory nerves in the distal extremities. This disease affects more than 20 Million Americans. Most commonly, peripheral neuropathy begins in the longest nerves. Specific symptoms vary, depending on which types of nerves are affected, but symptoms may include:

- Numbness and tingling in your feet or hands, which may spread to your arms and legs,
- Burning or a sharp, jabbing pain,
- Extreme sensitivity to touch,
- Lack of coordination,
- Muscle weakness or paralysis if motor nerves are affected
- Bowel or bladder problems if autonomic nerves are affected.

The first goal of treatment is to manage the condition causing your neuropathy. A number of factors can cause neuropathies. These factors include:

- Trauma or pressure on the nerve. Nerve pressure can result from using a cast or crutches, spending a long time in an unnatural position, repeating a motion many times or having a tumor or abnormal bone growth.
- Diabetes. When damage occurs to several nerves, the cause frequently is diabetes. At least half of all people with diabetes develop some type of neuropathy.
- Vitamin deficiencies. B-vitamins are particularly important to nerve health.

- Alcoholism. Many alcoholics develop peripheral neuropathy because they have poor dietary habits, leading to vitamin deficiencies.
- Autoimmune diseases.
- Other diseases. Kidney disease, liver disease and an underactive thyroid (hypothyroidism) also can cause peripheral neuropathy.
- Exposure to poisons. These may include some toxic substances and certain medications

If the underlying cause is corrected, the neuropathy often improves on its own. The second goal of treatment is to relieve the painful symptoms.

Brad Manor, the leader of this Kinesiology project, hopes to pinpoint the problem areas associated with Peripheral Neuropathy by studying human gait of both affected and non-affected patients. Learning the exact parts of the body which are malfunctioning could lead to improved ways of approaching and helping to treat peripheral neuropathy. Creating an accurate way to measure stability in human gait could lead to possible prevention of future accidents.

Data is collected in the lab from volunteers. Every person who participates is first fitted with several reflective sensors placed on their toes, ankle, knee, hip, wrist, elbow, shoulder, and forehead. Patients are then asked to walk on a treadmill, surrounded by 8 infrared cameras for approximately 80 seconds. Cameras capture 60 images/second and collect 4900 data points. Multiple cameras are used to reduce the error caused by a patient's arm blocking the sensor at the hip.

2. Time Series and Reconstruction

A time series is a set of consecutive data points that are taken over a specific, uniform interval of time. For this project, the time series will be a sequence of positions of the hip, ankle, and knee taken from a subject in the 1-dimensional space during the Kinesiology Human Gait experiment. With this time series, the Mathematics and Kinesiology department are attempting to locate a stable walking pattern by calculating and analyzing the Lyabunov exponent. With the study of chaotic discrete dynamical systems and attractors, the project will map a stable walking pattern to attractor, where as unstable walking patterns will deviate (repel) from a general attractor and therefore produce and large, positive Lyabunov exponent. In order to begin analyzing a time series Lyabunov exponent, the time series must be sent through a reconstruction process to properly capture the dynamics of the system from the 1-dimensional times series into an n-dimensional attractor. This project uses the method of delays, one of the most common reconstruction processes, to assemble the n-dimensional attractor. The method of delays is driven by a chosen embedding dimension of n, which will represent the number of state variables of the time series attractor. This method delays reconstruction begins with a time series X such that:

$$X = \{x_1, x_2, ...\}$$

we define new vectors X_i such that:

$$X = \left\{x_i, x_{i+j}, ..., x_{i+(n-1)j}\right\}$$

where j is the reconstruction lag and n is the embedding dimension. The reconstruction lag refers to the number of data points between each vector X_i that stretches the attractor by forcing the reconstruction to choose points later in the series. The reconstruction lag is calculated using the Rosenstein Lyapunov exponent reconstruction method. This method requires that the embedding dimension (determined from previous trials) be entered in as a parameter for the calculation. The GUI and plots for this project have used an embedding dimension and of 5. The reconstruction creates the vector $\bar{X}$ such that:

$$\bar{X} = \{X_1, X_2, ..., X_M\}$$

This reconstructed vector creates the attractor that will proceed into the nearest neighbor analysis for the trajectory.

3. Nearest Neighbors and Calculating $d_j(i)$

The reconstructed vector forms an attractor of one trajectory and finitely many cycles. To continue the calculation to the Lyapunov exponent, the distances between the nearest points in the cycles must be found in order determine the rate of separation of the cycles, the Lyapunov exponent. This process is known as determining the nearest neighbor. The distance $d_j(0)$ is as follows:

$$d_j(0) = min||X_j - X_{\hat{j}}||$$

for a reference point X_j in the reconstructed vector provided $|j - \hat{j}| >$ mean period. The mean period is a user-defined constant that makes sure the distances separate in a sufficient number of data point. $d_j(1)$ is calculated as

$$d_j(1) = ||X_{(j+1)} - X_{(\hat{j}+1)}||$$

and this continues to $d_j(i)$ in the form:

$$d_j(i) = ||X_n - j - X_{(n-\hat{j})}||$$

The nearest neighbor algorithm is used with each vector in $\bar{X}$ as a reference point.

4. Curve Fitting and the Lyapunov Exponent

The Lyapunov exponent is the rate the nearest neighbor differences separate. Therefore, in order to calculate the Lyapunov exponent the slope of the average distances must be calculated with a respectable goodness of faith fitting. Over the course of this experiment the Lyapunov exponent has been calculated using the linear model and finding a short term and long term Lyapunov exponent for each patient. In addition, this semester the project has fitted the curve to double exponential model explored by other researchers. The algorithm to both models is explained below.

Linear Model

For each i, the average of $d(i)$ is taken as follows:

$$d_j(i) = avg[d_1(i), d_2(i), ..., d_n(i)]$$

Graphing the $d_j(i)$‘s on the Cartesian plane can be approximated by the function:

$$d_j(i) > C_j e^{\lambda_1(i*\Delta(t))}$$

such that C_j is the initial separation and $\lambda 1$ is the Lyapunov exponent. To make this fit the linear model the natural logarithm is used as follows:

$$ln[d_j(i)] > ln[C_j] + \lambda 1(i\Delta(t))$$

where the Lyapunov exponent is the slope of the linear function. This project produces a linear fit for both a short term and a long term Lyapunov exponent. The short term Lyapunov exponent is referring to the data point of only the first stride and the long term Lyapnuov exponent refers to data points between the fourth and the tenth stride.

Double Exponential Model

The double exponential model follows the same path as the linear model to the average of the $d_j(i)$‘s. The methods in the fact that there is no need to compute the natural logarithm of the $d_j(i)$‘s since the plot will be fitted to an exponential model.

$$d_j(i) > A - B_s e^{\frac{-t}{\tau_s}} - B_L e^{\frac{-t}{\tau_L}}$$

where $\tau_l > \tau_s$ and A is upper limit of the data points. Although this semester the prior fit was applied, the actual significance of B_s, B_L, τ_L, and τ_s have not been determined. Instead, a graphical user interface that will make analyzing the variables of the fitting was pushed further into production.

5. Progress of the GUI

The GUI developed by previous semesters was updated for usability and additional options. Users can now follow a system that is broken into segments, providing unique messages and progress bars as the fit their data to the linear Lyapunov model. Options for the single and double exponential have been added and Matlab code and functions enabling the data to be plotted using the exponential models have achieved, but implementing the code to apply the exponential fitting has not been achieved. To implement the code for the exponential fitting, it is suggested to first use the curve fitting toolbox to generate the M-code needed for the fitting and then alter the M-code in order for it to easily be placed into the options already created in the GUI.

6. Reverse Dynamics

In general, if you use embedding dimension equal to m, there will be m Lyapunov exponents, just like a m-by-m linear system has m eigenvalues, that are not necessarily distinct. When using embedding theory to build chaotic attractors in a reconstruction space, extra ”spurious” Lyapunov exponents arise that are not Lyapunov exponents of the original system. By computing the exponents in reverse dynamics we are trying to identify potential spurious exponents. In the way we computed reverse dynamics, we are expecting to have the same exponents under normal and reverse dynamics, which is what the results show in Table 1 on the following page.

7. Conclusion

During the course of the semester, updates to the graphical user interface (GUI) and methods to updating both the single and double exponential, accompanied with a study on reverse dynamics was established. Although implementing the single and double exponential models in the GUI were not completed this could be added as a further update to the GUI.

We would like to thank Professor Li Li, Professor Lee Hong, Brad Manor, Alvaro Guevara, and Professor Wolenski for all of their helpful comments and advice over the course of the semester.

References

[1] Asplund, E. and Bungart, L. *A first course in integration,* Holt, Rinehart and Winston, 1966. Springer-Verlag Publishing Company, Heidelberg, 2000.

[2] Attouch, H. and Wets, R. J-B *A convergence theory for saddle functions*, Trans. Amer. Math. Soc., 280(1), 1983, 1-41.

[3] Rockafellar, R.T. "Dual problems of Lagrange for arcs of bounded variation," in *Calculus of Variations and Control Theory*, D.L.Russel(ed.), Academic Press, 1976, 155-192.

[4] Goebel, R. *Self-dual smoothing of convex and saddle functions*, J. Convex Analysis 15(1), 2008, to appear.

[5] Rockafellar, R.T. *Existence and duality theorems for convex problems of Bolza*, Trans. Amer. Math. Soc. 159, 1971, 1-40.

[6] Rockafellar, R.T. and Wolenski, P.R. *Convexity in Hamilton Jacobi theory 1: dynamics and duality*, SIAM J. Control Opt. 40, 2001, 1323-1350.

[7] Rockafellar, R.T. and Wolenski, P.R. *Convexity in Hamilton Jacobi theory 2: envelope representations*, SIAM J. Control Opt. 40, 2001, 1351-1372.

Table 1. Reverse Dynamics

Book	Normal/Reverse		Hip	Knee	Ankle
1	Normal	Short term	0.0311652	0.0345363	0.0241968
		Long term	0.00133618	0.0015417	0.00102537
	Reverse	Short term	0.0305798	0.0352694	0.0276853
		Long term	0.00169744	0.00172776	0.000978649
2	Normal	Short term	0.0319671	0.0338487	0.0278014
		Long term	0.00125689	0.00182689	0.00100679
	Reverse	Short term	0.0331502	0.0349964	0.0268293
		Long term	0.000871532	0.00170239	0.00132908
3	Normal	Short term	0.0233028	0.032567	0.0253493
		Long term	0.002046	0.00206504	0.00158643
	Reverse	Short term	0.0238859	0.0338533	0.0272519
		Long term	0.0017516	0.00178766	0.00152506
4	Normal	Short term	0.0321058	0.0366798	0.0203069
		Long term	0.0012862	0.00170522	0.00157981
	Reverse	Short term	0.0312252	0.0343921	0.0185343
		Long term	0.00120572	0.00210367	0.00156514
5	Normal	Short term	0.0244825	0.0316393	0.0241988
		Long term	0.00181546	0.00167798	0.00108574
	Reverse	Short term	0.0238859	0.0338533	0.0272519
		Long term	0.0017516	0.00178766	0.00152506
6	Normal	Short term	0.0233028	0.032567	0.0253493
		Long term	0.002046	0.00206504	0.00158643
	Reverse	Short term	0.0238859	0.0338533	0.0272519
		Long term	0.0017516	0.00178766	0.00152506
7	Normal	Short term	0.0400742	0.0413905	0.0300532
		Long term	0.000551648	0.000777868	0.000470957
	Reverse	Short term	0.0380803	0.0417392	0.0315075
		Long term	0.000945089	0.000753988	0.000265903
8	Normal	Short term	0.0283466	0.0316259	0.0266123
		Long term	0.000790844	0.00108036	0.000792237
	Reverse	Short term	0.027781	0.0319224	0.0270148
		Long term	0.00101281	0.000977129	0.00087323