Tom McIlwain

A Review and Implementation of Manifold Learning Techniques on Single-Neuron Recordings Generated from a Memory Recall Task

Introduction

A common theme in science is the representation of difficult-to-understand concepts in simpler contexts. In neuroscience, there exists a theory in which the representation of thoughts and behaviors within the brain lie in a lower dimensional space than the number of neurons that determine those behaviors. These low-dimensional spaces, often called manifolds, take advantage of the population-level dynamics of high-dimensional recordings where single-neuron analysis may not elucidate useful information. For example, if there is a single-trial recording of three neurons over time, the dynamics can be plotted in three dimensions, with each axis representing the activity of the neurons. It may become evident that the activity of these neurons occupies a much smaller area within the entire three dimensional space. We can capture this space with dimensionality reduction by decreasing data from dimension D to a space with K latent variables that explain variance of the data. The K-dimensional space may represent how each neuron covaries and the activity patterns that are correlated between neurons. Figure 1 is a visualization of this process, where the neural activity of each neuron is represented as each axis, and the trajectory through time lies on a two-dimensional plane [22]. The latent dynamics of the neurons represent the time-dependence of the neuron's activity and the neural modes are estimated using a dimensionality reduction technique such as Principal Component Analysis (PCA) that define the manifold.

Many studies have previously used dimensionality reduction techniques for the construction of manifolds from high-dimensional neural recordings that have revealed interesting dynamics within the brain. For example, William Newsome's group at Stanford University used a recurrent neural network model on decision-making prefrontal cortex data from macaque monkeys [1]. The motor cortex has been a popular choice for neural manifold analysis, and has revealed low-dimensional representations of movement from high-dimensional recordings [2][3]. Another increasingly relevant region for neural manifold discovery is the olfactory system within the brain, where many studies have found lower-dimensional trajectories of neural responses to odors [4][5].

Often, single units vary in firing rate and do not have a strong correlation with stimulus type, especially when recordings are averaged over trials [6]. Averaging removes noise, but also may remove important dynamics that are specific to each trial. In complex tasks such as memory recall or decision making, the time course of the neural spikes may differ widely. Therefore, single-trial analysis holds more statistical power but yields a much higher-dimensional dataset for analysis.

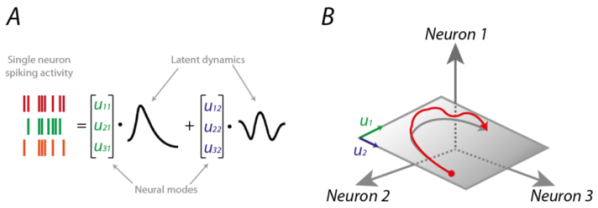

Figure 1: (A) Single-neuron spike trains can be represented as a weighted combination of neural modes and activation dynamics. **(B)** The trajectory of neurons' activity is plotted in three dimensions (red), where a two-dimensional plane (gray) captures most of the variance of the trajectory.

Many dimensionality reduction techniques exist, but there are a handful of them that are very useful for neural data analysis [7]. The method used for manifold discovery depends on the type of data used and the temporal dynamics of the task at hand. There are basic covariance methods that find how neural activity covaries over time such as PCA or Factor Analysis (FA) that are

usually applied to trial-averaged data. Other methods exist that are better used on single-trial time series data where spikes are binned into firing rates, such as Latent Linear Dynamical Systems or Gaussian Process Factor Analysis. This report includes a review of commonly used dimensionality reduction techniques in the context of neural data analysis, and an implementation of three of these techniques on single-trial recordings from individual neurons in the human medial temporal lobe when subjects are performing a memory recall task.

Dimensionality Reduction Techniques

Dimensionality reduction techniques are built on the concept of being able to capture a majority of covariance in the given data using a smaller amount of dimensions. Oftentimes, when dimensionality reduction techniques are mentioned, the first method to be brought up is PCA. A basic covariance method, PCA finds a lower-dimensional space by identifying orthogonal axes that capture maximum variance. PCA has commonly been used for neural data, for example extracting information from large datasets for tasks such as classification [8]. However, sometimes it may not be a good idea to capture the maximum amount of variance because firing rate variability and spiking variability will be included in the manifold discovery process [7]. To circumvent this issue, Factor Analysis (FA) is a similar technique that discards variance that is independent to each neuron but keeps covariance among neurons. FA is able to separate overall population variability from individual spike noise and has been useful for neural data analysis in the past for this reason [9].

There are other dimensionality reduction techniques besides PCA and FA that are more suited for time-series data, such as hidden Markov Models (HMMs), Gaussian Process Factor Analysis (GPFA), Latent Linear Dynamical Systems (LDS) and Latent Non-Linear Dynamical Systems (NLDS) . For example, HMMs are latent-state models that are used when neural population activity changes based on a discrete variable such as events that occur throughout the trial or experiment. HMMs have been used on neural population data for inferring the trial state of a monkey performing a center-out task [10]. When using HMMs, it is assumed that the population activity jumps between discrete states corresponding to the trial state. Autoregressive HMMs (AR-HMMs) are an extension of HMMs in which the Gaussian observations are replaced with an autoregressive model (the next observation depends on the previous state as well as the previous observations). For example, AR-HMMs have previously been used to train models using neural data recorded from the dorsolateral striatum of mice to relate to discrete and continuous features of mouse behavior [11][12]. While PCA is useful for neural activity averaged across trials to achieve average neural trajectories for different conditions, HMMs are useful for single-trial data in which events are occurring throughout the trial and neural data is assumed to correspond to the event state [7].

A model that has similar applications to the HMM in neural data analysis is the Gaussian Process Factor Analysis (GPFA) model. Yu et al. developed the GPFA method which unifies both smoothing and dimensionality reduction [13]. GPFA was created with single-trial analysis of neural population activity in mind and works similarly to other probability based models. Like FA, GPFA employs a specific noise model to account for across-trial single-neuron variability. However, neural states at different time points are related using Gaussian processes (GP) and the squared exponential (SE) covariance function is used for calculation of covariance matrices. The GP priors on the latent states allow for temporal smoothing due to the Gaussian kernel. GPFA simultaneously performs both smoothing and dimensionality reduction using the GP prior thereby allowing for joint optimization. GPFA is fit using training data and the expectation-maximization (EM) algorithm, which adjusts model parameters to maximize probability of

observed data given latent states. After fitting, GPFA can then be used to extract low-dimensional neural trajectories from high-dimensional neural population activity. However, GPFA is a linear dimensionality reduction technique which may not hold true for population-level activity of neurons.

Nonlinear dimensionality reduction methods exist for decreasing dimensions of data that covaries in a nonlinear manner. Isomap is a relatively older method for identifying nonlinear manifolds published in 2000 [14]. It combines the algorithmic features of PCA with nonlinear flexibility to detect nonlinear manifolds. Isomap uses a nearest-neighbour approach to determine points that are neighbors on a manifold, estimates the geodesic distances between each pair of observations, and applies classical multidimensional scaling (MDS) to construct a *d*-dimensional Euclidean space that preserves the intrinsic geometry of the manifold. Isomap was generated to find globally meaningful coordinates from complex data that did not have clear manifold geometry and has been used before in classification of neural activity [14][19]. Similar to Isomap, Locally Linear Embedding (LLE) uses a nearest-neighbour approach but uses Euclidean distance rather than geodesic distance. LLE was published around the same time as Isomap and has been used extensively in neural data dimensionality reduction for many applications including dimensionality reduction of olfactory system activity for odor detection [4][5][15].

Another method useful for neural data dimensionality reduction is the Gaussian Process Latent Variable Model (GPLVM). The GPLVM proposed by Neil Lawrence (2004) is a nonlinear dimensionality reduction technique that builds off of the algorithm of PCA but considers an alternative popular Radial Basis Function (RBF) kernel [16]. Compared to PCA, the GPLVM's mapping from latent space to the data space uses a nonlinear covariance method. GPLVM has been empirically shown to give useful visualizations of a range of different datasets and many methods have expanded upon the GPLVM's success, such as implementing a Variational Bayesian method for improved prediction to create a Bayesian GPLVM [17]. Jensen et al. at the University of Cambridge extended the GPLVM past the assumption of Euclidean latent states to include symmetric intrinsic manifolds such as spheres [18]. In the case where the manifold found lies within a Euclidean space, then the developed manifold GPLVM (mGPLVM) algorithm is identical to the Bayesian-GPLVM [17].

Using a dimensionality reduction technique for manifold discovery usually involves a few steps. First, the data is pre-processed which usually includes spike sorting and binning. Additional preprocessing steps such as smoothing of the data may be required beforehand depending on the method used. ANOVA analysis may also be considered to discard neurons that do not have much effect on mean firing rate. The next step is to estimate and interpret the dimensionality of the data. Using a metric such as variance explained, reconstruction error, or cross-validated likelihood of data, one can iterate over K where $K = [1, \ldots, \textit{num neurons}]$ and $k \in K$ to quantify the performance of a given model using k latent states. Then a model can be trained using the k-value that exceeds a chosen threshold. A low-dimensional trajectory can then be extracted using the fitted model by transforming given data from the high-dimensional neural space to the low-dimensional latent space.

Implementation of Dimensionality Reduction Techniques

Implementation Methods

To further the analysis of dimensionality reduction techniques for manifold discovery, three methods were chosen for implementation on subject data. Single-neuron recordings were used in a single-trial analysis of human medial temporal lobe activity while 59 subjects performed recognition memory tasks. The data was adapted from Chandravadia et al. and was formatted according to Neurodata Without Borders: Neurophysiology 2.0 (NWB:N) and Python code functions were used for formatting

original data and working with NWB:N data [19]. The task that was performed by the subjects had two parts: an encoding images phase and a recognizing images phase. During the encoding phase, 100 novel images were presented to the, each fitting to a distinct category (houses, landscapes, mobility, phones, animals, fruits, kids, military, space, cars, food, people, spatial). During the recognition phase, subjects were presented with 100 images, 50 being novel images and 50 being old images that the subject had seen in the encoding phase. The subject was asked to indicate if the image was new or old on a 1-6 confidence scale rating.

The data was spike sorted when presented, where neurons' single-trial recordings were grouped with the corresponding subject. Signals were sampled at 32 kHz and were band-passed filtered with a range of 300-3000 Hz before spike sorting was performed. Data was then ported to Python and spike rate was calculated with a bin size of 50 ms or spike trains were formatted depending on dimensionality reduction method used. For each trial, 4 seconds were analyzed starting with stimulus onset. After data pre-processing, the next step was finding dimensionality reduction. This process consisted of three steps: finding optimum dimensions for each method, apply method to transform data from high-dimensional neural space to low-dimensional latent-space along with visualization of trajectories, and calculating covariance matrices between trajectories of trials when presented with new vs. old stimuli during the recognition phase.

Three different dimensionality reduction techniques were implemented using this data. The first was GPFA to explore the linear dynamics of the neural data. GPFA was used rather than other linear methods due to its effective GP prior on smoothing data. The next technique used was Isomap to explore how an established nonlinear dimensionality reduction technique would perform on high-dimensional neural data. The Bayesian-GPLVM approach presented by Lawrence and Titsias was the third method used due to the incorporation of both GPs and nonlinear manifold discovery. GPFA was implemented using *Elephant*, an open-source library for analyzing electrophysiological data in Python [20]. Isomap was implemented using Scikit-learn. Bayesian-GPLVM was implemented using the GPy library from the Sheffield Machine Learning Group [21].

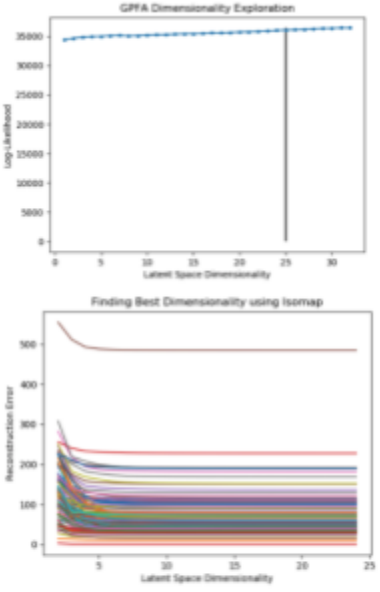

Figure 2: Dimensionality exploration of GPFA (top) and Isomap (bottom) methods. 80% of LL was achieved at 25 dimensions for GPFA, and reconstruction error reached asymptote starting at 5 latent space dimensions for Isomap.

Implementation Results

For each method, the optimum dimensions of the latent space was found using cross-validated log-likelihood to fit GPFA or reconstruction error of Isomap to fit both Isomap and Bayesian-GPLVM. Dimensions searched spanned from 1 to N with N = number of neurons in the trial. The data was then transformed using each model from the high-dimensional neural space to the low-dimensional latent space resulting in a trajectory through latent space for each trial. Covariance matrices were then calculated for comparing the covariance of latent states found across trials using activity when presented with old stimuli and new stimuli.

The first method implemented was GPFA using the *Elephant* library [20]. Using cross-validation, the dimensionality of the latent space that captured at least 90% variance was found to be $k = N$-1 where N = number of neurons. GPFA dimensionality optimization can be seen

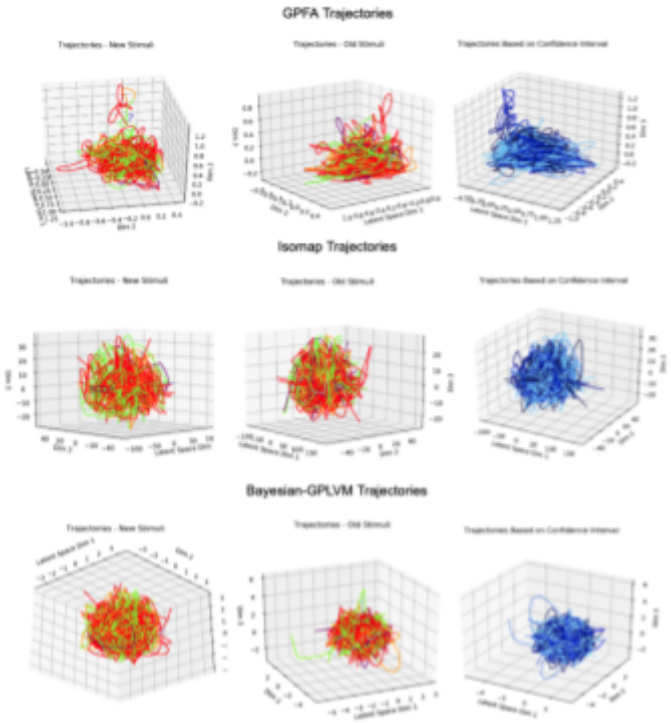

Figure 3: Neural trajectories through latent space for each trial found using each method above. The first two columns are colored according to events occurring during the trial (red = stimulus on, orange = stimulus off, green = question onset, blue-purple = subject given response, yellow = end of trial)

in figure 2. A final GPFA was then trained on the data using $k = 5$ to perform dimensionality reduction to move the data to the latent space. Covariance matrices were calculated for trials when subjects were presented with new stimuli, old stimuli, and all stimuli. Trajectories of GPFA can be seen in figure 3 and covariance matrices in figure 4. The 3-dimensional plots showing the first three dimensions of the latent state reveal a pyramid/cone type geometry with the tip of the cone at 0 magnitude. Based on the 3D plots, differentiability between the events throughout the trials is difficult as well as the confidence rating for each trial. The difference in mean covariance across trials for each stimulus group (new, old) was approximately double that of the covariance across all trials, indicating that the trajectories found may reveal a difference between 'new' and 'old' trajectories. However, based on 1000 iterations of random Monte Carlo sampling of 50 trajectories (25 from 'new' group and 25 from 'old' group), the difference is not statistically significant as Monte Carlo often results in mean covariance values between the two groups.

Isomap was the next method to be implemented. The reconstruction error was the metric to determine the best dimensionality of the latent space which was based on the first dimension to reach an asymptote from the starting reconstruction error, determined to be $k = 5$ (figure 2). The data was smoothed beforehand using a Gaussian kernel with $\sigma = 2.25$. Trajectories and covariance matrices were found using $k = 5$ and results can be seen in figures 2,3, and 4. The 3D plots of the trajectories did not reveal any meaningful geometry of the activity for each trial. Similarly to GPFA, Isomap trajectories were not differentiable based on confidence rating of the subject or trial stage. The mean of the covariance matrices for each stimulus group were again higher than the mean of the covariance matrix calculated for all trials but Monte Carlo simulations proved the difference was not significant.

After Isomap, Bayesian-GPLVM was implemented using the same *k*-value. The data was again transformed to the latent space then plotted and analyzed. The results of the Bayesian-GPLVM dimensionality reduction technique can be found in figures 2, 3, and 4. Similarly to Isomap, Bayesian-GPLVM did not reveal much meaningful geometry based on the trajectories through time.

Conclusion

There are many dimensionality reduction techniques that are well-suited for high-dimensional neural data analysis. For across-trial averaged activity, PCA and FA are effective at identifying axes that capture the most variance of the observations. For single-trial high-dimensional data, HMM, GPFA, LDS, and NLDS are best. Nonlinear methods such as Isomap, LLE, or GPLVM (and extensions) have also been

used successfully in the past on neural data analysis. Based on the success of these methods, three dimensionality reduction techniques were performed on human data when subjects were performing a memory recall task. GPFA revealed a cone-like geometry while Isomap and Bayesian-GPLVM did not reveal meaningful geometries.

There are a few limitations of these implementations. For example, nonlinear dimensionality reduction methods are often not robust to noise which impedes single-trial analysis. These methods also often use nearest-neighbour techniques that do not evenly explore the high-dimensional neural space therefore may not capture how each neuron covaries with others. This may explain why the nonlinear methods used may not be as successful at elucidating a meaningful manifold geometry.

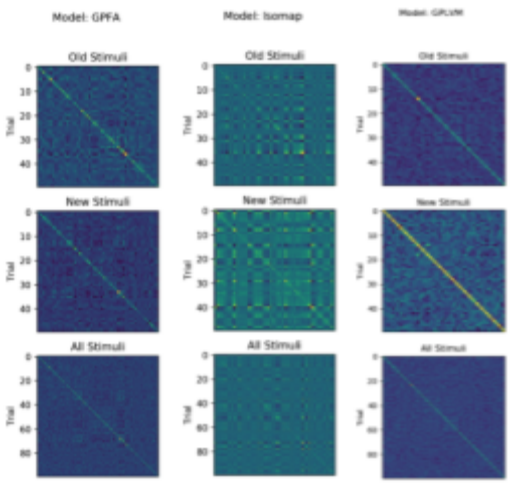

Figure 4: Covariance matrices for GPFA, Isomap, and GPLVM (left to right) with old, new, and all stimuli.

Future directions of this work would include combining all files of the given data and analyzing trajectories for each group across all individuals and sessions. Another addition would be to use HMM to detect changes in trial stage throughout the recording. Dimensionality reduction techniques for neural data analysis have been used successfully in the past to reveal geometries of high-dimensional recordings for less complex tasks such as center-out tasks and odor detection. However, the dynamics of memory recall may be too difficult to untangle using simple implementation of these methods using existing Python libraries. This work may be improved upon by further exploring the spatiotemporal dynamics of memory recall via a dimensionality reduction technique for manifold discovery.

References

1. Mante, V., Sussillo, D., Shenoy, K. et al. Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature 503, 78–84 (2013). https://doi.org/10.1038/nature12742
2. Churchland, M., Cunningham, J., Kaufman, M. et al. Neural population dynamics during reaching. Nature 487, 51–56 (2012). https://doi.org/10.1038/nature11129
3. Yu BM, Cunningham JP, Santhanam G, Ryu SI, Shenoy KV, Sahani M. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. J Neurophysiol. 2009 Jul;102(1):614-35. doi: 10.1152/jn.90941.2008.
4. Broome BM, Jayaraman V, Laurent G. Encoding and decoding of overlapping odor sequences. Neuron. 2006 Aug 17;51(4):467-82. doi: 10.1016/j.neuron.2006.07.018. PMID: 16908412.
5. Saha, D., Leong, K., Li, C. et al. A spatiotemporal coding mechanism for background-invariant odor recognition. Nat Neurosci 16, 1830–1839 (2013). https://doi.org/10.1038/nn.3570
6. Ryan J. Low, Sam Lewallen, Dmitriy Aronov, Rhino Nevers, David W. Tank. (2018). Probing variability in a cognitive map using manifold inference from neural dynamics. bioRxiv 418939; doi: https://doi.org/10.1101/418939
7. Cunningham, J., Yu, B. Dimensionality reduction for large-scale neural recordings. Nat Neurosci 17, 1500–1509 (2014). https://doi.org/10.1038/nn.3776
8. K. Das, S. Osechinskiy and Z. Nenadic, "A Classwise PCA-based Recognition of Neural Data for Brain-Computer Interfaces," 2007 29th Annual International Conference of the IEEE Engineering

in Medicine and Biology Society, Lyon, France, 2007, pp. 6519-6522, doi: 10.1109/IEMBS.2007.4353853.
9. Churchland, M., Yu, B., Cunningham, J. et al. Stimulus onset quenches neural variability: a widespread cortical phenomenon. Nat Neurosci 13, 369–378 (2010).
10. Kemere C, Santhanam G, Yu BM, Afshar A, Ryu SI, Meng TH, Shenoy KV. Detecting neural-state transitions using hidden Markov models for motor cortical prostheses. J Neurophysiol. 2008 Oct;100(4):2441-52. doi: 10.1152/jn.00924.2007.
11. Markowitz, Jeffrey E et al. "The Striatum Organizes 3D Behavior via Moment-to-Moment Action Selection." Cell vol. 174,1 (2018): 44-58.e17. doi:10.1016/j.cell.2018.04.019
12. Wiltschko, Alexander B et al. "Mapping Sub-Second Structure in Mouse Behavior." Neuron vol. 88,6 (2015): 1121-1135. doi:10.1016/j.neuron.2015.11.031
13. Yu, Byron M et al. "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity." Journal of neurophysiology vol. 102,1 (2009): 614-35. doi:10.1152/jn.90941.2008
14. Joshua B. Tenenbaum, Vin de Silva, John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*. 22 Dec 2000 : 2319-2323
15. Stopfer M, Jayaraman V, Laurent G. Intensity versus identity coding in an olfactory system. Neuron. 2003 Sep 11;39(6):991-1004. doi: 10.1016/j.neuron.2003.08.011. PMID: 12971898.
16. Neil Lawrence (2003). Gaussian Process Latent Variable Models for Visualization of High-Dimensional Data. *Advances in Neural Information Processing Systems 16 (NIPS 2003)*.
17. Titsias, M. & Lawrence, N.D.. (2010). Bayesian Gaussian Process Latent Variable Model. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research 9:844-851
18. Kristopher T. Jensen, Ta-Chu Kao, Marco Tripodi, Guillaume Hennequin. (2002). Manifold GPLVMs for discovering non-Euclidean latent structure in neural data. *Cornell University*. arXiv:2006.07429.
19. S. K. Prabhakar and H. Rajaguru, "Factor analysis, Hessian Local Linear Embedding and Isomap for epilepsy classification from EEG," 2016 Electrical Engineering Conference (EECon), Colombo, Sri Lanka, 2016, pp. 19-24, doi: 10.1109/EECon.2016.7830929.
20. Denker M, Yegenoglu A, Grün S (2018) Collaborative HPC-enabled workflows on the HBP Collaboratory using the Elephant framework. *Neuroinformatics* 2018, P19. doi: 10.12751/incf.ni2018.0019
21. The Sheffield Machine Learning Group (2018). Gpy: The Gaussian Processes Framework in Python. *Github*. github.com/SheffieldML/GPy
22. Gallego JA, Perich MG, Miller LE, Solla SA. Neural Manifolds for the Control of Movement. Neuron. 2017 Jun 7;94(5):978-984. doi: 10.1016/j.neuron.2017.05.025.