Using Embedding Algorithms to Find a Low Dimensional Representation of Neural Activity During Motor Planning

CS229 Final Project – Fall 2005

Afsheen "the Plumber" Afshar and John "the Whale" Cunningham

Introduction

Patterns of neural activity in certain brain areas are understood to drive motor behavior. In the time immediately preceding a movement, there is a period of preparatory neural activity, called the "plan period." This activity can be measured as cell firing patterns, where we record as data, for each neuron, the number of observed action potentials in small time bins. Thus, the recorded data is equivalent to a high dimensional firing rate vector as a function of time, with one dimension for each neuron.

Prior work has supported the hypothesis that neurons in the dorsal premotor cortex (PMd) region of the brain modulate their activity depending on the direction, distance, and speed of an upcoming movement [Churchland and Shenoy]. Numerous theories have attempted to explain what these neurons are representing by their firing patterns, but no single encoding scheme has proven universally successful in explaining the observed data.

Recently, various researchers have begun to propose models of plan period activity in PMd that do not restrict neurons to solely representing a small subset of the dimensions of the upcoming movement (e.g. parameters in a kinematic model of the arm) [Yu et al]. Specifically, they have proposed that, while planning, the population's firing rate vector is in a low dimensional manifold of the high dimensional firing rate space. Further, it posits that when the operation of planning involves altering firing rates so that they move to a subspace of that manifold. This is called the 'optimal subspace hypothesis'. Various experimental results, such as the finding that firing rate variance across similar trials decreases as a function of time, agree with this hypothesis [Churchland et al].

If firing rates during planning occupy a low dimensional manifold, there should be a way to represent these data using many fewer dimensions than the number of neurons. This low dimensional representation could reveal the fundamental neural signatures of motor planning as well as correlate with behavioral features of the impending movement. Such a representation could be very useful for neural prosthetics in addition to basic neuroscience.

Current work involves using expectation maximization to do exactly this [Yu et al]. This work investigates how well the simpler algorithms of Principle Component Analysis (PCA), Isomap, Local Linear Embedding (LLE), and Sensible PCA (SPCA) perform on this problem.

Background

Linear Methods – PCA and SPCA [Roweis]

PCA is a linear projection method that works on the simple principle of ordering the axes in terms of their variability (an eigenvector decomposition) and selecting the dimensions of most variability (highest eigenvalues). This well known analysis has the benefit of simple implementation, quick runtime, and optimal mean squared error over the class of linear methods. While it is very useful in assessing important dimensions in the data, it lacks a proper probabilistic model under which one can evaluate test data; that is, given a n dimensional (n=number of neural units) training set $\{x^{(i)}; i = 1, ..., m\}$, PCA does not learn a generative model for x. Hence, one can not calculate the likelihood of any training or test data.

SPCA addresses this problem by adding a proper probabilistic framework to PCA via a Factor Analysis approach. Calling the data **x** and the latent variables (iid) **y**, we assume the model:

$$\mathbf{x} = C\mathbf{y} + \mathbf{v} \qquad \mathbf{y} \sim N(\mathbf{0}, \mathbf{I}) \qquad \mathbf{v} \sim N(\mathbf{0}, \varepsilon\mathbf{I}) \qquad \Rightarrow \mathbf{x} \sim N(\mathbf{0}, CC^T + \varepsilon\mathbf{I})$$

We can calculate the likelihood of any data under this model using $p(x \mid \theta)$. To learn this model, we will employ the EM algorithm. Using Bayes's Rule and the rules for conditioning on Gaussian Random Vectors, we can write:

<u>E-step:</u> $Q_i(y^{(i)}) = p(y^{(i)} \mid x^{(i)}; C, \varepsilon) = N(\beta x^{(i)}, \mathbf{I} - \beta C)\,|_{y^{(i)}}$ $where\, \beta = C^T(CC^T + \varepsilon\mathbf{I})^{-1}$

Letting **X** be the n-x-m (number of units by size of training set) matrix of training data, further analysis then yields:

<u>M-step:</u> $C^{new} = \mathbf{X}\mathbf{X}^T\beta^T\Sigma^{-1}; \quad \varepsilon^{new} = tr(\mathbf{X}\mathbf{X}^T - C\beta\mathbf{X}\mathbf{X}^T)/mn; \quad where\, \Sigma = m\mathbf{I} - m\beta C + \beta\mathbf{X}\mathbf{X}^T\beta^T$

Thus, SPCA builds a Factor Analysis model around the data set. The drawback of this approach is that it requires the iterative EM algorithm, which can be computationally costly and can find only local optima. However, in our application, we found that this EM algorithm ran quickly and produced consistently the same optima, which are equal to the principle components found in PCA. In [Roweis], the author confirms that SPCA has never been found to have local optima, though no proof has been found. With slightly more computational and algorithmic complexity, SPCA produces the same projection results as PCA and yields a probabilistic model, which will be useful in comparing this method to other inference techniques.

Nonlinear Methods – General Approach

PCA will fail to find any lower dimensional space that is embedded non-linearly in a higher dimension (e.g. a manifold twisted up into higher dimension). For Euclidean manifolds, Isomap and LLE avoid this shortcoming of linear projection by arguing that, for a given point in a well sampled space, the point's nearest neighbors will lie only in that low dimensional space and will not exhibit the higher dimensional convolutions. Then, if we preserve the local geometry and dimension of each neighborhood, we should be able reconstruct the manifold using only the dimension of those neighborhoods. However, these methods suffer when neighborhoods do not well represent the manifold on which the data lies (e.g. due to sampling sparsity). While LLE and Isomap assess local geometry and reconstruct differently, their high level approach is quite similar.

LLE[Roweis et al.]

LLE examines the neighborhood of k points around each data point to approximate the high dimensional data X_i by a collection of linear subspaces. These linear subspaces are then used to embed the high dimensional X_i into a lower dimensional data set Y_i of predefined dimension.

Specifically, this is broken into two operations. The first, called reconstruction, involves following the weights needed to reconstruct each X_i by a linear combination of its k nearest neighbors:

$$E(W) = \left(\sum_i \left| X_i - \sum_j W_{ij} X_j \right|^2 \right) \qquad \begin{aligned} &W^* = \arg\min E(W) \\ &s.t. \sum_j W_{ij} = 1 \quad \forall i \end{aligned}$$

Note that $W_{ij} = 0$ for all points X_j that are not within the k-neighbors of point $X_{i.}$ These weights W are then used to embed X_i in the lower dimensional space in the second, embedding step to produce Y. This maintains the relative positions of the original points when performing the embedding. Note that the constraints that each row of W sum to 1 makes LLE invariant to rotation of the original data set X.

$$E(Y) = \left(\sum_i \left| Y_i - \sum_j W_{ij}^* Y_j \right|^2 \right) \qquad Y^* = \arg\min E(Y)$$

Isomap[Tenenbaum et al.]

Isomap also involves two steps, and it also tries to maintain the relative positions of data points X when performing the embedding to points Y. It goes about doing this by first creating a neighborhood graph in which all points are connected to their k neighbors (k is again a chosen parameter). Distances between all pairs of points are calculated by traversing this graph instead of using Cartesian distance. The concept here is that, with enough sampling, distance in the manifold between any two points can be well approximated by a series of hops along the shortest path of a neighborhood graph. This approach makes Isomap preserve the relative positions of points when performing the embedding operation.

The algorithm embeds the X_i's in a lower dimensional space of given dimension by trying to maintain the distances between all pairs of points as best as possible:

$$\begin{aligned} &E = \left\| \tau(D_G) - \tau(D_Y) \right\|_{L^2} \\ &Y = \arg\min E \end{aligned}$$

where D_G is the matrix of pairwise distances in the original space; D_Y is the matrix of pairwise distances in the embedded space; and the tau operator converts these distances to inner products for improving optimization.

Methods

An electrode array (Cyberkinetics, Inc.) was used to record from the PMd of a rhesus macaque monkey during the delay period prior to forty reaches to a single target. The targets were presented on a fronto-parallel screen about 20 cm from the monkey. Spike sorting, the process by which broadband neural data is processed to a set of discrete firing events, was done by hand using time amplitude hoops. Only those units that were deemed, by an expert spike-sorter, to be (i) of high quality, and (ii)dedicated to motor planning were kept for analysis. A total of 47 single and multi-units were used for the following study, where a unit represents a distinct pattern of neural activity that can be attributed to one or possibly multiple neurons. Each neural unit then has a pattern of discrete firings indicative of a continuously changing firing rate, which we view as a data set with dimensionality equal to the number of units. The TEMPO behavioral acquisition system (Measurement Computing) was used for all trial timing and behavioral control.

Code from the authors' of LLE and Isomap was downloaded and tailored to the data structures used. Code for PCA and SPCA's EM algorithm was written by Afshar and Cunningham.

Notes on Implementation/Comparison vs. Milestone

It is pertinent to note differences in approach from those reported in the project milestone. After many different approaches to PCA, LLE, and Isomap yielded no useful results (see results shown on milestone), we changed the representation of our data structure. We first used binned spike counts (bin width of 20ms across ~100 neural units). Owing to the sparsity of spikes, bins almost always contain zero spikes, and in no case more than six. Thus, each data point (spike counts for one time bin) in ~100 dimensions was effectively confined to lie on the non-negative orthant, integer lattice, from zero to six. Even in high dimensional space, it seems doubtful that there would exist structure here.

To correct for this problem, we returned to the recorded spike times for each neural unit and convolved this event signal with a unit Gaussian of standard deviation 50ms. On the very reasonable assumption that neural firing rates are modulated continuously, this convolution produces continuously changing firing rates across the time course of a trial. This operation preserves the information in the data but enforces continuity and smoothness. With this alteration, we produce appealing results.

Also, PCA and SPCA have been included in the analysis. PCA was initially added as a cross comparison with Isomap and LLE. We found that it performed as well as Isomap and better than LLE (see Discussion). Having been frustrated by the lack of probabilistic framework in these techniques, we also sought a more theoretical learning model. This search produced SPCA. Using SPCA also has the benefit of learning a hidden linear system that generates this data. This allows us to compare model quality vs other, more complex models being developed in our research that include non-linearities and dynamics.

Results

The following figures show plan period firing rate trajectories that were projected onto a 3-D manifold using the four dimensionality reduction techniques. Red circles indicate the beginning of the trial, 200ms before the target appears (i.e., a time of undirected neural activity where we know no plan has formed); black circles indicate when the go cue was given to move (e.g., the *end* of the plan period). One blue trajectory connects each pair of red and black circles.

Figure 1. (a) Firing Rate Trajectories to one reach target projected onto 3-D using PCA. Note consistency of trajectories through firing rate space to a 'optimal subspace' of planning. (b) Residual Error curve of PCA showing effective dimensionality of around 6.

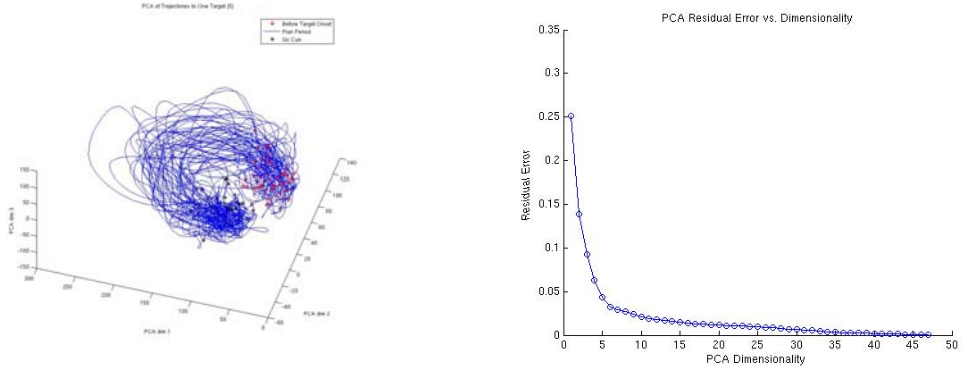


Figure 2. (a) Firing Rate Trajectories to one reach target projected onto 3-D using SPCA. (b) EM learning algorithm of SPCA.

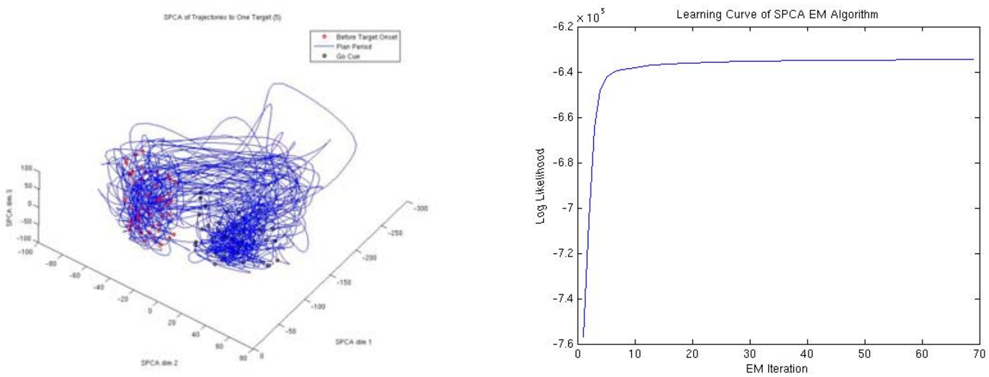


Figure 3. (a)Firing Rate Trajectories to one reach target projected onto 3-D using Isomap (k=41). (b) Residual reconstruction error curve showing intrinsic dimensionality of roughly 6.

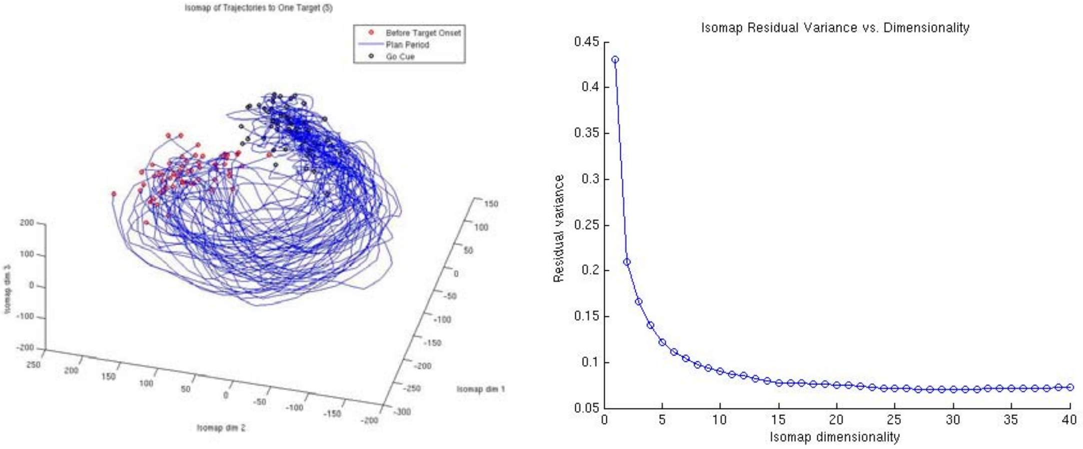


Figure 4. 98-dimensional noisy S-curve projected onto 3-D using LLE (k=40) (used for control to ensure correct algorithm implementation).

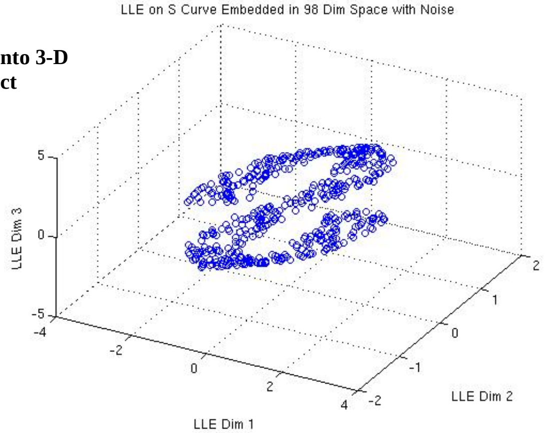


Figure 5. Firing Rate Trajectories to two reach targets projected onto 3-D using SPCA. Trajectories to each target begin at (red,yellow) circles, follow (black, green) paths, and end at (blue,magenta) circles in their respective 'optimal subspaces'

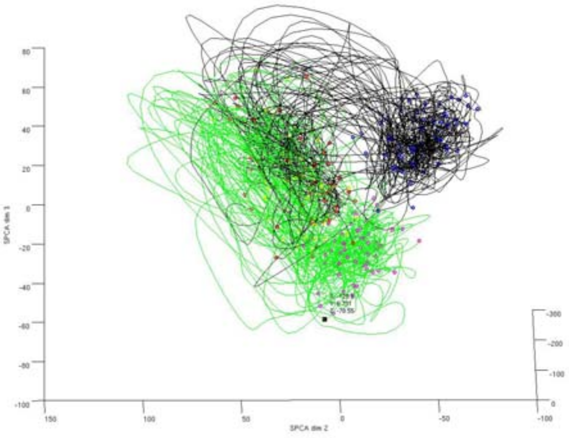
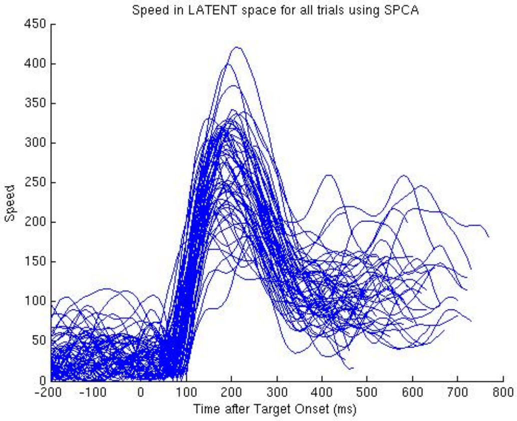


Figure 6. Speed of trajectories through latent space found using SPCA.



Discussion and Conclusions

For this data to be consistent with our hypothesis, we expected to see distinguishable trajectories over time through the low dimensional manifold, settling into the 'optimal subspace.' Indeed, in all of PCA, SPCA, and Isomap, we see these trajectories quite nicely (Figs 1, 2, 3, and 5). To go further with this hypothesis, we expect that different reach targets would settle to different optimal subspaces, representing different reach plans. We see this clearly in Figure 5, where we group all trials to two different reach targets. From the red and yellow circles, we see that each trial begins in the same large, noisy 'null subspace,' and then, when the target is presented, the brain quickly traverses to one of two optimal subspaces (determined by which target was presented), ending in the blue and magenta circles. It has also been hypothesized that, once a target is presented, the brain will form a motor plan quickly, reaching the optimal subspace and then staying roughly confined to that space. By plotting the velocity of the trajectories in Figure 6, we see exactly this effect: before the target onset, the brain state is slowly moving around a noisy null state; when the target is presented (after a small latency attributable to the visual system), the brain rushes predictably to the optimal subspace, where shortly after it slows and remains.

The above discussion focuses on the neuroscientific results of this analysis. There were also notable differences in the performance of the dimensionality reduction techniques. Linear methods (PCA and SPCA) have done as well visualizing the structure as the nonlinear methods. Isomap, with an appropriate choice of neighborhood size, can readily recover the structure, but its computational complexity does not justify its use given the similar outcome. Interestingly, we have not been able to get meaningful results from LLE for any choice of parameters. We show in Figure 4 a control, the noisy 3-D S-curve embedded into 98 dimensions, to illustrate that our algorithm is working correctly. One might suspect that the embedding via neighborhood mappings may be a source of this error. Indeed, for naïve choices of neighborhood sets and data sampling rates, this fact can cause failure in both LLE and Isomap. While we were able to overcome these roadblocks for Isomap, we have not been able to do so with LLE. Our working hypothesis for this failure is that, due to the sparsity in high dimension of this data set, modeling each point as a linear combination of its nearest neighbors is very inaccurate, thereby destroying the structure in the data. Unfortunately, the lack of a proper model framework for LLE and Isomap make it difficult to further analyze their successes and failures.

For this reason, we conclude with the opinion that, of all the dimensionality reduction techniques we have investigated, SPCA stands out as the most suitable for this application, both in terms of having a well defined machine learning framework and in terms of efficiently producing quality results. Being able to visualize trajectories through the brain's state space suggests many new avenues of research, including the determination of laws of motion and behavioral correlates in this state space. Having a sound probabilistic framework and an efficient dimensionality reduction algorithm, well tested against its peers, is critical in approaching any of these larger issues, and thus we consider this work a successful and crucial step towards that goal.

Acknowledgments

We thank Mark Churchland and Byron Yu for valuable discussions and data collection; and Missy Howard and Mackenzie Risch for expert veterinary care.

References

Churchland MM and Shenoy KV. Movement speed alters distance tuning of plan activity in monkey pre-motor cortrex. Soc. For Neurosci. Abstractss (2003).

Churchland MM, Yu BM, Ryu S, Santhanam G. Shenoy K. Reaction time and the time-course of cortical pre-motor processing. Soc.For Neurosci. Abstracts (2004).

Roweis S & Saul L. Nonlinear dimensionality reduction by locally linear embedding. Science v.290 no.5500, Dec.22, 2000. pp.2323-2326.

Roweis S. EM Algorithms for PCA and SPCA. Neural Information Processing Systems 10 (NIPS'97)

Tenenbaum JB, de Silva V, Langford JC. A Global Geometric Framework for Nonlinear Dimensionality Reduction Science 290 (5500): 2319-2323, 22 December 2000

Yu BM, Afshar A, Santhanam G, Ryu SI, Shenoy KV, Sahani M (talk and poster). Extracting dynamical structure embedded in neural activity. Neural Information Processing Society (NIPS '05)