Assignment 1

Psych 253 (Stanford University, Spring 2019)

Released: 2019/04/03; Due: 2019/04/22

1 Assignment Questions

1.1 Reliability

In this section, we will show that by computing a reliability metric, you can both assess the quality of a dataset, as well as discover meaningful scientific findings. The dataset we will use includes visually evoked responses of neurons in macaque ventral cortical areas V4 and IT, from an experiment conducted by Najib Majaj, Ha Hong and Jim DiCarlo at MIT.[1] In total, the researchers measured electrophysiological responses from 296 multi-unit neural sites in V4 and IT, while animals were rapidly presented with a wide variety of complex images containing pictures of objects at multiple levels of object viewpoint variability (see section 2 for details).

This dataset has been used to perform a variety of powerful analyses, including quantitative predictions of human object recognition performance from neural substrates [2, 3], comparison of cortical neurons to deep neural networks [4, 5], and reformulations of our understanding of representation of "category-orthogonal properties" in higher visual cortex [6]. So this will be a cool dataset for the class to learn on.

However, as good scientists, we always want to remain skeptical of everything. Thus, before doing anything fancy on this dataset, we would like to estimate its internal quality by computing its reliability. To understand how we measure reliability, it's useful to think about the way the data arises in the first place. Electrophysiological data (essentially) arrives in the form of millisecond-scale "spiketrains" — that is to say, sequences indicating for each millisecond bin whether or not a given neural site registered a spike during that bin. Because this measurement process is noisy, for each neural site in the dataset, the same image was presented multiple times, so as to ensure that a good estimate of the neural site's true mean response to each image could be estimated. Our first task, therefore, will be to measure exactly how good these estimates are in our dataset.

1.1.1 Problem 1: Reliability computation on time-averaged responses

For most of the problems we will face this quarter, we will work with what we call "trial-averaged, time-averaged neural responses". As the name indicates, these are neural responses in which the raw spiketrain data has been processed by averaging spike counts within a larger time window, and then averaging further across all trials in which the same image has been presented. (The specific time window we will often be looking at is 70ms-170ms. You may want to know why the experimentalists choose this window; the next problem aims to answer this question.) To estimate the reliability of these trial-averaged time-averaged data, we need to work with *trial-wise* (but still time-averaged) neural responses. This data has been provided in the dataset (see section 2).

- **Reliability of each neuron:** In this problem, we would like you to first compute the reliability for each neuron using the methods described in the class, and then perform a significance test on the population of all the neurons to determine whether the population is significantly reliable or not. In the report, please describe your methods for computing the reliability, choose your own significance level, and report the results of significance test properly (refer to section 3 for details).
- **Reliability by areas:** Now that you've gotten a reliability measure for each neuron, one interesting thing to test is whether reliability of V4 neurons is significantly different from that of IT neurons. You can find the indexes of V4 neurons and IT neurons in the dataset, and as discussed in class. When you do this, again choose a significance level and report the significance test results properly. In performing this test, you should include variability due jointly to choices of sets of repetitions, as well as to choices of sets of neurons on which to calculate the mean reliability for each area. Plot a bar chart showing the mean and standard deviations of reliabilities for V4 and IT. To get an intuitive understanding of the potential differences between the areas, also plot a histogram of the reliability of the neurons in the two areas.

[1] For those of you who are not familiar with the neuroscience of the primate visual system, have a look at [1], which is a good introductory reference.

- **Reliability by animals:** Since in this experiment we have two animals and multiple electrode arrays in each animal, we also would like to validate that neurons in each animal and also each electrode array agenerating reliable responses. (For subgrouping of neuron by animals, the information of which animal neurons belong to is given by "ANIMAL_INFO" in "neural_meta" (see section 2).) After grouping neurons by animals, please perform significance test to both animals and report the results. Similarly, please also plot a histogram of reliability of neurons in two animals.
- **Reliability by electrode arrays:** As for grouping by electrode arrays, the information of which array neurons belong to is stored at "ARRAY_INFO" in "neural_meta". You need to combine this information with "ANIMAL_INFO" to group the neurons by six electrode arrays, e.g. the neurons belonging to "Chabo" and array "P". After grouping neurons by electrode arrays, please plot a histogram of reliability of neurons in six arrays and also barcharts of the averaged reliability of neurons in six arrays with error bars. You are NOT required to perform significance tests for this part of the problem, but you are welcome to do so if you want.

1.1.2 Problem 2: Reliability computation on 20ms responses

Although we might have confirmed the quality of this dataset through computing reliability on time-averaged neural responses, this process of time-averaging could conceal low-quality data by combining both high- and low-quality data. Therefore, in this problem, we will work on time-binned neural responses which are closer to the most original experimental data. Specifically:

- **Reliability by areas in each of 20ms bins:** Please first compute reliability for every neuron on data from each of 20ms bins from 0ms to 200ms post stimulus presentation. The time binned neural data is stored in the dataset using key "time_binned".

  One interesting pattern to investigate is how the reliabilities vary across time, especially whether this pattern differs between neurons in V4 and IT. Hence, please group the reliabilities of neurons for V4 and IT respectively for different time bins and then plot the the timecourse of reliabilities for these areas in the same plot. Describe your findings. Do you think everything is OK in the data? If not, explain what you find abnormal and why. If you think everything's OK, please check **HINT 0** below.

  HINT 0: The images are presented to the animals from 0ms to 100ms. It will usually take some time (at least 40ms) for neurons in V4 and IT to produce reliable responses.

- **Diagnose the problem:** OK, let's face reality: **There must be something wrong in our data!** Given the reliabilities for all neurons in all time bins, how should we locate this problem? What will you try to diagnose it? Please show your efforts in the notebook and explain your motivations behind these efforts. If you find it difficult to continue, we have provided you a hint (**HINT 1**) after section 1.2.1. If you believe you have found the problem, please also check **HINT 1** to verify your finding.

  If you have successfully located the problem, we believe you will already know what to do in the following problems to avoid using the wrong part of the data. But please check **HINT 2** after section 1.2.2 to make things clear. (Modifying your answers of previous problems to exclude the wrong data is **NOT** required.)

  - **(NOT required) Look back at reliabilities from time-averaged responses:** Answering this sub-problem is not required but it will give you a slightly better understanding about what is really happening with the wrong part of the data.

    One natural question you may want to ask is what influence the bad part of the data has on the reliabilities computed on time-averaged responses. To see this, average and show those reliabilites for both good part and bad part separately. What do you notice? Next, choose two neurons from the same area, one from good part and one from bad part, compute trial-to-trial correlation matrix for each of them on images from one variation level (choose any one you like). Visualize these two matrices — do you find any differences? Is the same conclusion true for all other neurons? What do you then conclude about the effect of the bad part of the data?

- **Difference in V4 and IT neurons:** Finally, exclude the wrong part of the data in the following analyses. Redraw the first plot you had for this problem (1.1.2). Describe and interpret your findings. How do V4 and IT differ in terms of when in time they become reliable? Or when reliability fades? What might any differences between V4 and IT you find mean neuroscientifically? (We don't require significance tests for your findings, but you are welcome to perform one if you want).

1.2 Classifiers

After measuring the reliability of the dataset, we can now explore some more interesting hypotheses. Researchers have found that simple linear classifiers based on neural responses from Inferior Temporal (IT) cortex can accurately perform object recognition tasks [7, 2]. In this section of the assignment, we would like you to first reproduce aspects of this result in a simpler context — that is classifying one of the eight basic object categories in the images (you can find category information in "category" of "image_meta"). You will exploring the effects of regularization, variation levels and visual areas on the classification results.

1.2.1 Problem 1: Regularization

As introduced in the class, the type and amount of regularization added to SVM model will greatly influence the results. In this problem, we will explore this influence by testing different types and amounts of regularization. Specifically:

- To explore the best parameters for regularization, please first compute SVM models with cross-validation but without any regularization as baseline. Then, try both L1 and L2 regularization on SVM weights using parameters from {1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1, 10}. To compare fairly between different parameters, you need to use the same cross-validation splits across all parameters. In constructing your splits, you want to (i) make sure that the number of splits is larger enough that the average will provide a good estimate of the mean tendency of performance, (ii) have enough images in both train and test components of the split so that the classifier is learned reasonably and tested effectively, and (iii) of course ensure that the train and test components do not overlap.

  Besides the accuracies on validation set, please also compute the d-primes from the confusion matrix on validation sets (averaging the confusion matrix over splits as discussed in class). For both accuracies and d-primes, please plot values versus parameters for L1 and L2 regularization respectively in one plot.
- Next, find the best parameter for both types of regularization, and run multiple cross-validation image splits for the best parameters of L1 and L2 regularization (as well as the SVM model without regularization). Finally, please plot two barcharts for both accuracies and d-primes of best L1, L2, and no-regularization model with error bars and use proper statistical tests on each of two measures to find the best type of regularization to use later (again, please choose your significance level and report the significance test results properly).

HINT 1: Please check the reliabilities of each neuron in early time bins. Some neurons have obviously wrong reliability numbers.

1.2.2 Problem 2: Variation levels

As discussed in class, the experiment presented images to animals at differing levels of variation: a *low-variation* condition in each which each object is presented at a fixed size, viewpoint, and position across multiple images; a *medium-variation* condition in which each of these variables ranges a bit; and a *high-variation* condition in which the variables range widely. Now that you know how to train SVM classifiers on neural responses of all images for eight-category classification, an interesting question to ask is how performance depends on variation level, and especially whether SVMs trained on one variation level can generalize to another. To answer this question:

- First group images into different variation levels (using "variation_level" stored in "image_meta"). For each level, please train a SVM model with cross-validation and the best regularization type and parameter found in the previous problem. Similarly, plot the validation accuracies and d-primes in two barcharts (you are not required to run multiple splits to generate error bars).
- Next, let's explore the generalization performances across different levels. Train SVM model with regularization on all images belonging to one variation level and test it on images of another level. Put the validation accuracies in the same matrix, show it, and explain your findings.

HINT 2: Please exclude the last 40 neurons in V4 in all following problems.

1.2.3 Problem 3: Visual areas

Given that the neurons in our dataset are drawn from both V4 and IT areas, one question we can ask is whether how performances compare between neurons from the two cortical areas. Specifically:

- Please first group the neurons by areas using "IT_NEURONS" and "V4_NEURONS" in "neural_meta". Then train SVM models with regularization on V4 and IT neurons separately. Please plot the validation accuracies in a barchart, generating error bars for each area across cross-validation image splits. Are the accuracies for two areas different? Using significance test to give an answer and report the test result properly. How do you interpret these results?
- Although accuracy is a good measure for distinguishing models, we can reveal more details about the differences through comparing their confusion matrices. Therefore, please compute the average confusion matrices over multiple splits for both V4 and IT neurons. Visualize the confusion matrices separately. Are the matrices different? One way to answer this question quantitatively is to flatten these matrices and then compute the Pearson or Spearman correlation values. Please compute and report these.
- Even given these correlation values, we still lack a deeper and more intuitive understanding about the differences, e.g. where do the results of the V4 and IT neural populations differ exactly? To answer this question, make plot a scatter plot of confusion matrix values for corresponding items between the V4 and IT matrices. Make the scatter plots separately for the diagonal terms and the off-diagonal terms of the confusion matrices, and interpret your findings.

2 Dataset

In this section, we will introduce the ventral neural dataset [2], which will be used throughout the quarter for testing methods. As described both in class, this dataset contains neuronal responses of 296 neurons from two animals Chabo and Tito through presenting 5760 images using standard rapid visual stimulus presentation (RSVP). Each image was constructed by rendering one of 64 3-dimensional objects at some chosen position, pose, and size, on a randomly chosen background photograph. Specifically, each image was only presented for 100ms to the animals.

We have stored this dataset in one hdf5 file. In order to load it, we will use package `h5py`. Below is an example of loading the data and showing its contents.

```
In [1]: import h5py

In [2]: fin = h5py.File('ventral_neural_data.hdf5', 'r')

In [3]: fin.keys()
Out[3]:
[u'image_meta',
u'images',
u'neural_meta',
u'time_averaged',
u'time_averaged_trial_averaged',
u'time_binned',
u'time_binned_trial_averaged']
```

These keys mean:

- "images": an array containing the actual images of the dataset, of which the shape is [5760, 256, 256]
- "image_meta": a group of arrays describing the meta data for each image:
  - "category": an array containing the string name of the category of the object for each image, e.g. one of Animal, Boat, Car, Chair, Face, Fruit, Plane or Table.
  - "object_name": an array containing the string name of the object (of 64 possible objects) in the image.
  - "variation_level": containing the string name of the variation level of the image parameters from which the image was drawn. Specifically, these include "V0" (low variation), "V3" (medium variation), and "V6" (high variation). There are 640 V0 images (10 images per object), 2560 V3 images (40 images per object), and 2560 V6 images (40 images per object), for a total of 90 images per object.
  - "translation_y": the horizontal position of the object in the image, in type float64.
  - "translation_z": the vertical position of the object in the image, in type float64.
  - "size": the size of the object in the image, in type float64.
  - "rotation_[xy,yz,yz]": the rotation of the object in the r_{yz}, r_{xy} or r_{xz} planes (corresponding to in-plane rotation, rotation around the horizontal axis, and rotation around the vertical axis, respectively).
- "neural_meta": a group of arrays describing the meta data for each neuron:
  - "V4_NEURONS": containing indexes of neurons belonging to V4 area

  - "IT_NEURONS": containing indexes of neurons belonging to IT area
  - "AIT_NEURONS": containing indexes of neurons belonging to anterior IT area
  - "CIT_NEURONS": containing indexes of neurons belonging to central IT area
  - "PIT_NEURONS": containing indexes of neurons belonging to posterior IT area
  - "ANIMAL_INFO": containing animal names for all neurons
  - "ARRAY_INFO": containing array information for all neurons. There are three arrays for each animal. These arrays are named as "P" (posterior), "M" (middle), and "A" (anterior). (see Figure 2.B in [2] for details)
- "time_averaged_trial_averaged": an array containing time averaged and trial averaged neural responses in shape of [5760, 296], in type float64
- "time_binned_trial_averaged": an array containing trial averaged neural responses for each of the 20ms time bins starting from 0ms to 200ms, in shape of [5760, 11, 296] and type float64
- "time_averaged": a group of arrays containing time averaged neural responses for different trials. As numbers of trials are different for images in different variation levels, so the responses are separated by the variation levels:
  - "variation_level_0": an array containing the neural responses in type float64 and shape of [28, 640, 296], where 28 is the number of trials
  - "variation_level_3": an array containing the neural responses in type float64 and shape of [51, 2560, 296], where 51 is the number of trials
  - "variation_level_6": an array containing the neural responses in type float64 and shape of [47, 2560, 296], where 47 is the number of trials
- "time_binned": a group of groups of arrays containing neural responses for different trials and different 20ms time bins. The neural responses are first stored by different starting time points (0ms, 20ms, 40ms, ..., 200ms). Each 20ms time bin will contain a group of arrays structured in the same way as "time_averaged" data.

3 Lab Report

In the IPython notebook you are going to submit, please assume that the data hdf5 file is stored with the same folder, which means that when you refer to the data hdf5 file, please use relative path rather than absolute path. We will run the submitted IPython notebook cell by cell from top to bottom. To make this operation successful, please ensure that functions used are either defined within the notebook (and can be found by running the cells in order) or imported from another python module that is also submitted. As far as possible please don't use complex external Python packages — write any code you use yourself whenever possible. If you'd like to use any third-party Python packages, please ask course staff for permission first.

For info on how to report the results of significance test properly, please refer to `https://my.ilstu.edu/~jhkahn/apastats.html`.

4 Further reading

Some additional reading you might find useful:

- "An Introduction to the Bootstrap" [8], a great book on the basis of bootstrapping by faculty here at Stanford.
- "The Elements of Statistical Learning" — PDF at `http://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf`. This is a classic in presentation of material about SVMs and many other types of classifiers and regressors.
- Two classic papers on representational similarity analysis (RSA) spearheaded by Niko Kriegeskorte [9, 10]. This assignment doesn't have you do RSA *per se*, but these papers are a great way to get into thinking about high-dimensional analysis of the visual system.
- "Untangling invariant object recognition" [1], a review of ideas about decoding in the ventral visual stream.
- [7], one of the earliest convincing uses of linear readouts to decode object recognition from electrophysiology data.
- A good review of the use of multi-voxel pattern analysis in fMRI [11].
- A more recent review of computational approaches to fMRI data analysis [12].
- `Nilearn`, a package for neuro-imaging in Python — `http://nilearn.github.io/`. Similarly, have a look at `PyMVPA` at `http://www.pymvpa.org/`. Be aware that these sort of high-level APIs can make it easy to get results without actually understanding what you're doing.
- `Scikit-Learn`'s comparison of different classifier types: `http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html`

References

[1] DiCarlo, J. J. & Cox, D. D. Untangling invariant object recognition. *TICS* (2007).

[2] Majaj, N. J., Hong, H., Solomon, E. A. & DiCarlo, J. J. Simple learned weighted sums of inferior temporal neuronal firing rates accurately predict human core object recognition performance. (*In press.*). *J. Neurosci.* (2015).

[3] Rajalingham, R., Schmidt, K. & DiCarlo, J. J. Comparison of object recognition behavior in human and monkey. *J. Neurosci.* **35** (2015).

[4] Cadieu, C. F. *et al.* Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology* **10**, e1003963 (2014).

[5] Yamins*, D. *et al.* Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences* (2014).

[6] *Hong, H., *Yamins, D. L., Majaj, N. J. & DiCarlo, J. J. Explicit information for category-orthogonal object properties increases along the ventral stream. *Nature neuroscience* **19**, 613–622 (2016).

[7] Hung, C. P., Kreiman, G., Poggio, T. & DiCarlo, J. J. Fast readout of object identity from macaque inferior temporal cortex. *Science* **310**, 863–6 (2005).

[8] Efron, B. & Tibshirani, R. J. *An introduction to the bootstrap* (CRC press, 1994).

[9] Kriegeskorte, N. *et al.* Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron* **60**, 1126–41 (2008).

[10] Kriegeskorte, N., Mur, M. & Bandettini, P. Representational similarity analysis - connecting the branches of systems neuroscience. *Front Syst Neurosci* **2**, 4 (2008).

[11] Haxby, J. V., Connolly, A. C. & Guntupalli, J. S. Decoding neural representational spaces using multivariate pattern analysis. *Annual review of neuroscience* **37**, 435–456 (2014).

[12] Cohen, J. D. *et al.* Computational approaches to fmri analysis. *Nature neuroscience* **20**, 304 (2017).