The German Neuroinformatics Node www.g-node.org

Information Theory and Decoding

1. **Training and testing a classifier** Load the dataset `task01.mat`. It contains observations from two classes. Each row in `X` is a single observation, each element in `Y` is the corresponding class label. The matrix `Xtest` has the same structure as `X`, but you do not have the corresponding labels.

   (a) Use `glmnet` to fit a logistic regression to the data. Do not forget to add the toolbox to your matlab path beforehand. Use `glmnetSet` to set a particular value of α and λ.

   (b) Use `glmnetPredict` to predict the class labels on `Xtest`.

   (c) Play with α and λ to tune your classifier. Once you think you have good parameters, predict the labels on `Xtest` with the corresponding classifier, save them as `Ytest` in a `.mat` file names `FIRST_LAST.mat` where you replace `FIRST` and `LAST` with your first and last name. Give the .mat file to me. We will make a ranking who has the best classifier.

   (d) **(optional)** Download another classifier from the web or use one provided by Matlab to predict the labels. Candidates are *k nearest neighbors (kNN)*, *perceptron*, *support vector machine (SVM)*, or *Fisher discriminant analysis*. You might need to change the values of the labels, since many implementations expect $y_i \in \{-1, 1\}$ instead of $y \in \{1, 2\}$
   $-$

Please wait before you start the next task. There will be a short lecture on model selection beforehand.

---

i

2. **Proper model selection** Use the same dataset as in the previous task.

   (a) Implement *cross validation (CV)* to implement a series of training and validation sets. It is up to you whether you generate the sets randomly (without replacement) in each iteration or whether you generate the sets from a partition of you dataset. However, make sure that the single splits are *balanced*, i.e. that the fraction of positive and negative labels reflect the respective fractions in the entire dataset.

   (b) Use the average CV classification accuracy (the fraction of correctly predicted class labels on the validation set) to optimize your parameters λ and α. Re-train the classifier on the entire dataset for the best set of parameters.

   (c) As before, use your best classifier to predict the labels of `Xtest` and give them to me in a .mat file as above.

   (d) Visualize the decision of the classifier along with the training and test points. Make sure to plot the decision line at which the predicted label changes.

   (e) If you have not done so already, put the code for the model selection into an extra function, that takes the data `X` and `Y`, the number of cross validation splits `N`, the fraction `q` of points that should end up in the validation set in each split (if you use random sampling for the CV), as well as two lists `alpha` and `lambda` of possible values for α and λ you would like to check. It should return the best model trained again on the entire dataset.

3. **Mutual Information** Write a script that loads each of the 25 datasets `task03_*.mat`.

   (a) For each dataset perform model selection on `X` and `Y` and use the best model to predict the labels on `Xval`. Compute the classification accuracy using `Yval` and the predictions from `Xval`.

   (b) Let $\hat{y}_i$ denote the predicted labels for y_i. Use `Y` and `Yval` to estimate the following probability mass functions $P(\hat{y})$, $P(y)$, and $P(\hat{y}, y)$. *Hint: I recommend implementing this and the next subtask in an extra Matlab function, since we will need the function again later.*

   (c) Use the probability mass functions to compute the two marginal entropies

$$H[\hat{Y}] = -\sum_{\hat{y}\in\{1,2\}} P(\hat{y})\log_2 P(\hat{y})$$

$$H[Y] = -\sum_{y\in\{1,2\}} P(y)\log_2 P(y)$$

   and the joint entropy

$$H[\hat{Y}, Y] = -\sum_{\hat{y}\in\{1,2\}, y\in\{1,2\}} P(\hat{y}, y)\log_2 P(\hat{y}, y).$$

   From that compute the mutual information

$$I[\hat{Y}, Y] = H[\hat{Y}] + H[Y] - H[\hat{Y}, Y].$$

(d) Plot the accuracies and the mutual information for all datasets against the variable `bayes_err` which contains the theoretical optimal error for each dataset. Do the errors and mutual information values make sense to you? What is the maximal mutual information we can achieve? What do you think is the better measure for reporting classification results, the classification error or the mutual information? Why?

4. **Time course of information** Load the dataset `neurons08.mat`. It contains filtered responses of 20 neurons to different orientations and contrasts.

   (a) Pick two orientations for discrimination. For each contrast and each of the 90 time bins: (i) take the response of the population as data, (ii) perform model selection, (iii) test, and (iv) estimate the mutual information form the predicted labels. Since we do not have an extra test or validation set anymore, we need to create one using cross validation again (This means, we use two nested cross validations: the outer for testing and computing the mutual information, and the inner for model selection). To speed up the process, use 5 instead of 10 splits and fix $\alpha = 0$.

   (b) Plot the mean mutual information at each time point and for each contrast against time (the array `times`). Use the several mutual information estimates per bin and contrast from the outer cross validation loop to visualize the variability of the mutual information estimates.

   (c) What can you observe in the plot? Does the maximal mutual information make sense to you? Does the time course make sense to you?

5. **Population vs. single neurons** For this exercise use the population responses at the time points between 50ms and 250ms. Ignore neuron 1 since it is silent in this time window. Pick two orientations and focus on contrast 100.

   (a) Now use the responses of each single neuron in the time window 50ms to 250ms as features for your classifier. Like above, use two nested cross validations to estimate the information that each *single* neuron has about the two different orientations.

   (b) Concatenate the responses of all the neurons to form long feature vectors representing the population activity during the above time window. Like for the single neurons, use the long feature vector to estimate the mutual information the population has about the orientations.

   (c) Compare the average information of each single neuron, the information of the population, and the sum of the informations over all neurons. How can you explain what you find?