RSA assignment

For this assignment, we are going to compare some neural data and computational models to see how well they match. In particular, we are going to split the brain into 7 separate regions of interest (ROIs): EVC, OPA, PPA, LOC, PFS, OFA, and FFA. For our computational models, we are going to look at a vision based neural network AlexNet, and a word encoding model word2vec. The goal of this assignment is to run an RSA analysis to compare each model to each ROI.

Let’s start with Figure 1 from Representational Geometry (Kriegeskorte and Kievit 2013).

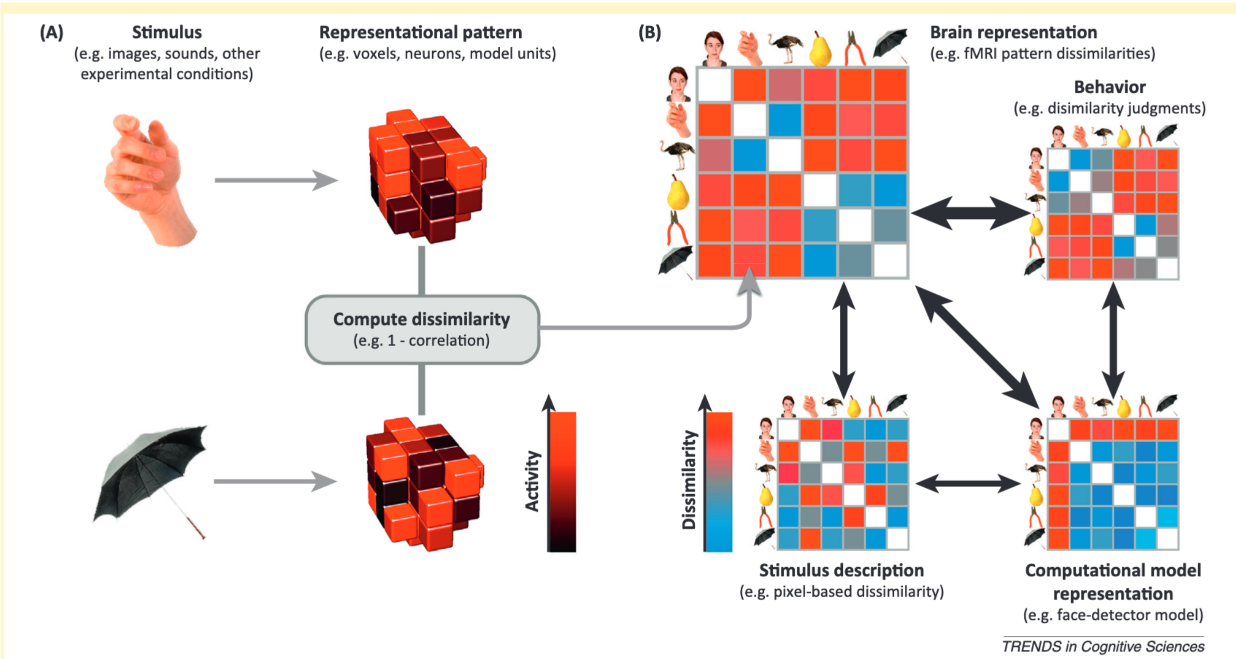


Figure I. Representational similarity analysis. Illustration of the steps of RSA for a simple design with six visual stimuli. **(A)** Stimuli (or, more generally, experimental conditions) are assumed to elicit brain representations of individual pieces of content (e.g., visual objects). Here the representation of each item is visualized as a set of voxels (an fMRI region of interest) that are active to different degrees (black-to-red color scale). We compute the dissimilarity for each pair of stimuli, for example using 1–correlation across voxels. **(B)** The representational dissimilarity matrix (RDM) assembles the dissimilarities for all pairs of stimuli (blue-to-red color scale for small-to-large dissimilarities). The matrix can be used like a table to look up the dissimilarity between any two stimuli. The RDM is typically symmetric about a diagonal of zeros (white entries along the diagonal). RDMs can similarly be computed from stimulus descriptions (bottom left), from internal representations in computational models (bottom right), and from behavior (top right). By correlating RDMs (black double arrows), we can then assess to what extent the brain representation reflects stimulus properties, can be accounted for by different computational models, and is reflected in behavior.

The four matrices are each RDMs, and the arrows connecting them are separate RSA correlations.

Warm-up Questions

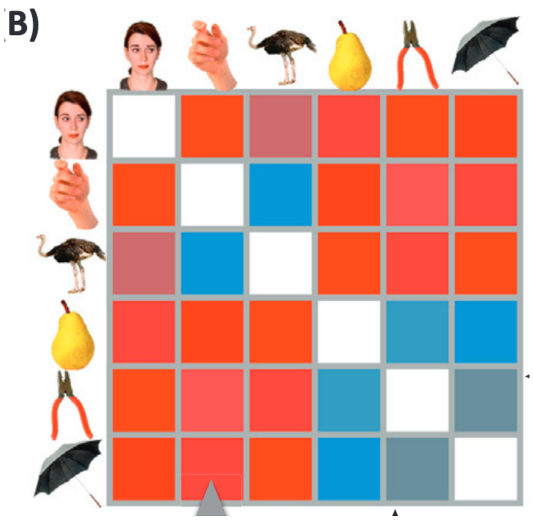
1. How many RDMs do we need to create?
2. How many total RSA correlations are possible?

For our analysis, we are going to restrict the number of correlations that you need to calculate. You will only need to calculate the correlation between the computational models and the separate ROIs.

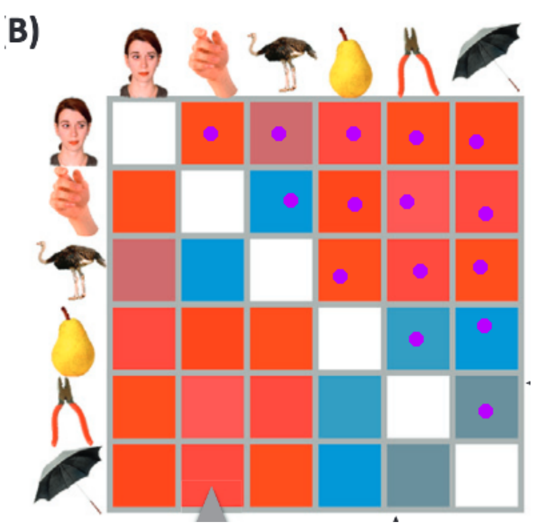
3. How many RSA correlations will you end up with?

Now let’s turn to the specific requirements for the coding part.

1. Create RDMs using Pearson distance
2. Compare RDMs using Spearman correlations (these are referred to as “RSA correlations” below)
    a. There’s one gotcha here that you should know while you are comparing the RDMs. Let’s zoom in on one of the RDMs from above.

    The items along the diagonal are all 0 because we are comparing the distance between a stimulus and itself (I suggest to make sure that this is true when you are calculating your RDMs). This is true of every RDM regardless of the representation. Beyond the diagonal, we actually have every element duplicated twice because the values are mirrored over the diagonal. When calculating the correlation between two RDMs, you should only consider the unique data above (or below, the order doesn’t matter) the diagonal. You should only consider the data marked with a purple dot

    b. **Question**: In our data, there are 81 stimulus conditions which produce a full 81x81 RDM. How many elements will the RDMs you compare have once we account for the diagonal and duplicated data?

3. Create two bar graphs: one showing RSA correlations for alexnet in each ROI, and a second showing RSA correlations for word2vec in each ROI. These do not need to be separate plots; they could be shown side-by-side.

You will need to turn in your code along with the figures that you generated from this analysis. Your code should be documented and commented so that we can follow the logic of your analyses.

It is important to understand the questions for implementing your analysis. If any of the questions confuse you, please come by office hours.

Other helpful information

- You DO NOT need to compute p-values or confidence intervals.
- You CAN work with others, but the final code that you turn in should be your own. This means that you should be able to describe each line of your code and what it does.
- You are welcome to explore other approaches to these analyses, but any alternative approaches should be reported IN ADDITION to the analyses outlined above.