What sort of metric is that? Representational Similarity Analysis, Benefits, Limitations, and Tradeoffs

Bruce Rushing

20 March 2020

1 Introduction

The proliferation of data from neuroimaging techniques has led to the development of a number of new methods for analyzing that data. Among those methods, representational similarity analysis (RSA) (Kriegeskorte, Mur, and Bandettini 2008) promises indirect comparisons of representations formed internally within brains, comparisons of the representations formed in brains of different species, and comparisons of representations between biological and artificial neural networks. RSA works by measuring per some established similarity metric the response difference to stimuli of distributed representations found in data collected either through various neuroimaging techniques or directly from models. In this way, representations can be compared and said to be "similar" to one another. But "similarity" is a vague concept. How scientists make that concept exact has consequences for what precisely RSA ends up measuring. It is those consequences that we aim to discuss in this paper. More specifically, our goal is to show that, *pace* (Kriegeskorte, Mur, and Bandettini 2008), the choice of similarity measure does make a difference and suggest that those measures track structural differences in models. This follows results given by (Maheswaranathan et al. 2019) that architectural differences often matter to the similarity of representations formed in different models.

Here is how the paper proceeds. First, we review how RSA works, its successes, as well as the different similarity metrics employed in RSA. Second, we outline our approach of training models to compare their representations utilizing RSA and testing whether the choice of similarity metric can affect the result of those analyses. Third, we provide the exact models trained, the data set used, and RSA tools utilized. Fourth, we present our results. Finally, we discuss our results and directions for future research.

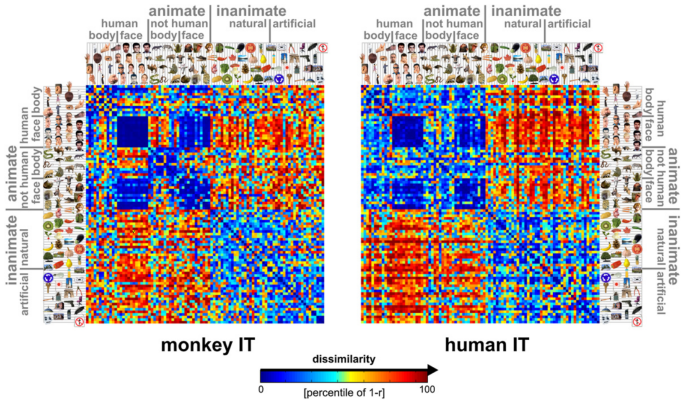


Figure 1: Two RDMs constructed from raw data collected in monkey and human inferotemporal cortex. Taken from (Kriegeskorte, Mur, Ruff, et al. 2008).

2 RSA Basics and Successes and Metrics

RSA was originally proposed in (Kriegeskorte, Mur, and Bandettini 2008) and has since been expanded as a tool to analyze representations found in both brains and models. In this section, we review RSA. First, we examine the basic steps employed in RSA. Second, we briefly canvas some successes of RSA. Third, we focus on the core component of RSA, the similarity metrics used to compare and order representations.

2.1 RSA Basics

RSA is a six step process through which distributed representations found in neuroimaging data or model data structures can be compared. The core tool is the representational dissimilarity matrix (RDM) (see figure 1 for example). The RDM is a two dimensional matrix that compares pairwise the raw numerical data of different representations. Each representation can be derived from neuroimaging data, such as fMRI voxel response, or taken directly from model activity patterns. Those representations are elicited by the presentation of stimuli and labeled according to those stimuli. Different representations can then be compared by measuring them against one another according to a similarity metric. The results of those measures are then stored in a matrix where each entry in the rows and columns corresponds to a specific stimuli. After being constructed, RDMs can then be compared against one another.

Constructing RDMs and utilizing RSA has six steps. The six steps are as follows:

1. Estimate Activity Patterns.
2. Measure Activity-Pattern Dissimilarity.

3. Predict Representational Similarity with Models.
4. Compare Brain and Model Dissimilarity Matrices.
5. Test Relatedness of Dissimilarity Matrices via Randomization.
6. Visualize the Similarity Structure.

Step one is to collect raw data for estimating activity patterns in either brains or models. The details of this step depend upon the data being collected and may require some additional interpolation such as fitting fMRI voxel response to a univariate linear model. Step two is the construction of RDMs from that data using an established similarity/dissimilarity metric. Step three involves building and training models and constructing RDMs from their activation patterns. Steps four through six involve doing statistical tests for comparing RDMs from both organics and models.

In this paper, we will focus on steps two and three, because these steps involve the construction of RDMs and ordering of representations according to a similarity metric.

2.2 RSA Successes

Using RSA has led to a number of purported successes. First, RSA has allowed for representations to be compared across different parts of the brain. Comparison of RDMs from early in the human visual cortex to those later in the cortex purport to show increased categorical representation in the inferotemporal (IT) lobes as opposed to earlier in the ventral visual stream (Kriegeskorte, Mur, Ruff, et al. 2008, pp. 1134–1135). Second, RSA has allowed comparison of representations between homologous cortices across human and monkey brains (Kriegeskorte, Mur, Ruff, et al. 2008). Third, and perhaps most excitingly, comparisons of human IT representations and those found in deep neural network AIs used for image and audio classification suggest that those AIs process visual and audio data in a manner similar to the human brain (see Daniel LK Yamins, Hong, et al. 2014, Daniel L Yamins et al. 2013, Daniel LK Yamins and DiCarlo 2016, Khaligh-Razavi and Kriegeskorte 2014, Cadieu et al. 2014, Kell et al. 2018).

2.3 Similarity Metrics

As mentioned previously, the key step in RSA is the construction of RDMs. Critically, each element of a RDM has its value determined by a similarity metric, which is a function that takes as input the raw data of the relevant representations and outputs a real-value number. That output determines how "similar" the two representations are to one another. Consequently, what metric is used determines what exactly is being measured by RSA.

Most RSAs employ one of four different metrics. These metrics come in two different categories: similarity measures and distance measures.

Similarity measures are functions whose values are normalized with respect to a scale. The main two measures used in RSA are pearson correlation and spearman rank correlation. When applied in RSA, these measures are typically incorporated as the inverse similarity measure ($1 - measure$). The pearson correlation between two random variables X, Y is given by

$$\rho(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{1}$$

It measures the linear correlation between those random variables (how well the variance between the two variables can be fitted on a line). Spearman rank correlation is given by the rank, G, of two random variables X, Y and is given by

$$\rho_S(G_X, G_Y) = \frac{cov(G_X, G_Y)}{\sigma_{G_X} \sigma_{G_Y}} \tag{2}$$

This measures the degree to which the variance of the functions can be fit by a monotonically increasing function.

Distance measures are functions whose values are not normalized with respect to a scale. This means that distance measures are more sensitive to the magnitude of the differences they measure. The two main distance measures employed in RSA are euclidean distance and absolute difference. Euclidean distance is taken between two n-dimensional vectors p, q and is given by

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_j)^2} \tag{3}$$

Importantly, euclidean distance does not just consider just the magnitude difference between vectors but also the angle between the two vectors. Absolute difference ignores that information and computes the difference of the magnitudes between two vectors as

$$d(p, q) = \sum_{i=1}^{n} |p_i - q_i| \tag{4}$$

The sum in absolute difference is done to produce a scalar from two vectors.

Because similarity measures normalize their inputs, they throw away information. The information that they throw away is the magnitude of their input. Distance measures, however, keep that information (Friedman, Hastie, and Tibshirani 2001, pp. 503–504). It is this difference between the two types of measures that lead us to think that different metrics will give different RDMs. We turn to that now.

3 Hypotheses

Since similarity measures track a particular ordering of their inputs and distance measures track the magnitudes of their inputs, we hypothesized that different metrics can lead to incongruent RDMs. By incongruent RDMs, we mean that representations would be ordered differently between the different RDMs. RDMs that are congruent may assign different values for individual pairs of representations but the ordering of those values across the RDMs would be the same.

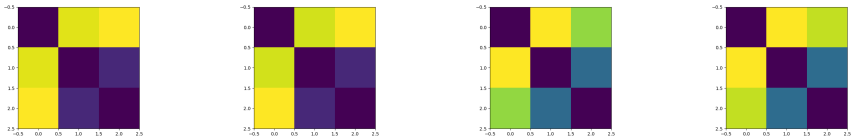
We also hypothesized that similarity metrics would give different results for data collected on stimuli at a coarse-grained as opposed to fine-grained level. Coarse-grained stimuli are those stimuli that are more abstract categorically than the fine-grained stimuli. For instance, the category *mammal* is more coarse-grained than the category *cat* since *cat* is included in *mammal*. The reason for testing this is that if metrics give different answers between coarse and fine-grained representations, some metrics might be better suited to different grains.

4 Methods

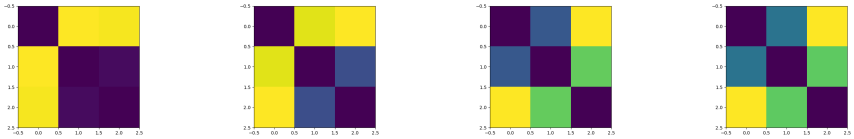
We built five different neural network models to test our two hypotheses. Each model was trained to classify probability distributions from a data set generated using Python's built in random module and pseudo-random number generator. Models would be presented a sample of 512 floating point numbers and asked to classify the generating probability distribution. Classification tasks could be either coarse or fine-grained. A coarse-grained classification task would involve labeling the functional form, e.g. Gaussian, while a fine-grained classification would be to label the actual probability distribution, e.g. a Gaussian with a mean of 0 and a standard deviation of 0.25. Each individual distribution has 10,000 samples generated for a total of 120,000 samples across all twelve distributions. There were three functional forms used: Gaussian, Beta, and Gamma. Each functional form had four different variations of parameters. Consequently, each coarse-grained category had 40,000 samples to draw from.

The model architectures chosen were not complex. Three of the models trained were simple networks that have a single hidden layer with sixty-four hidden units. Each simple network has a different activation functions: sigmoid, ReLU, and smoothstep. The smoothstep activation function involved a step activation function and then a smooth approximation of that function for computing the gradient. The remaining two models were dense networks with three hidden layers of size 256, 128, and 64 respectively. The difference between these networks was that one had a ReLU activation function and the other used a smoothstep.

Every model was trained separately on the coarse-grained and fine-grained task. Each task altered the output layer to the number of category labels units that were then run through a softmax function. When trained on a different



(a) Pearson (b) Spearman (c) Euclid (d) Absolute



(e) Pearson (f) Spearman (g) Euclid (h) Absolute

Figure 2: RDMs for simple networks with sigmoid and ReLU activation function. Figures (a)–(d) are for the simple Sigmoid network. Figures (e)–(h) are for the ReLU network. Each row and column in the RDM corresponds to the coarse-grained label of distributions with a Gaussian, Beta, and Gamma functional form (in that order). RDM values were computed by averaging of a set of four examples drawn from each distribution representative of that functional form. Darker values indicate representations that are more similar; lighter that they are more dissimilar.

task, models started with a random initialization of weights. Training occurred using the cross entropy loss function and was run for thirty epochs per model and task. 1,000 samples from each data set category were reserved for validation.

After training was complete, internal representations were accessed but cutting off the classification layer and feeding each model an exemplar from each fine-grained category. Direct activation values from the penultimate layer were then recorded as sixty-four dimensional vectors. Coarse-grained activation values were computed by averaging across the activation values for all relevant fine-grained samples.

Lastly, RDMs were constructed for each model at both the coarse-grained and fine-grained level. Coarse-grained levels involved averaging representational responses across all members of the coarse-grained label. Each level had four RDMs built using a different similarity metric: pearson correlation, spearman rank correlation, euclidean distance, and absolute difference distance. These RDMs were then qualitatively compared to detect whether they were congruent or incongruent.

(a) Pearson (b) Euclid

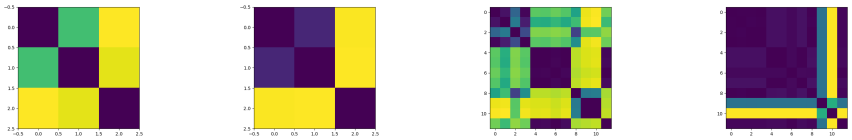
Figure 3: RDMs were incongruent at the fine-grained level. Figures (a) and (b) show the incongruency for a simple network with a ReLU hidden layer between pearson correlation and euclidean distance measures. The first four rows and columns correspond to individual Gaussian distributions, the next four to Beta distributions, and the last four to Gamma distributions. Darker values indicate representations that are more similar; lighter that they are more dissimilar.

5 Results

After training, all networks reliably achieved a testing accuracy of greater than ninety percent on the coarse-grained task and at least seventy percent on the fine-grained task (see figure 2 for examples). Training accuracy routinely outperformed testing accuracy.

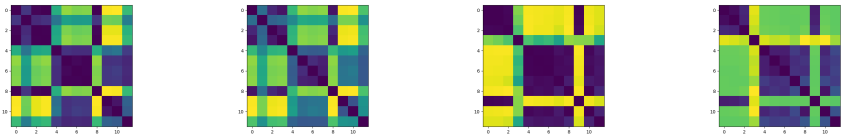
RDMs constructed from the models trained on the coarse-grained task at both a coarse-grained and fine-grained level show remarkable differences between models. The main difference that was found was between models with different activation functions. For example, RDMs constructed from the hidden layer of a simple sigmoid network are congruent while RDMs constructed from the hidden layer of a simple ReLU network are incongruent between similarity and distance measures (figure 2). The incongruency was most pronounced at the fine-grained level (figure 3). Similarily, smoothstep networks showed incongruencies at the coarse-grained level but failed to show it at the fine-grained level. Furthermore, dense networks RDMs differed from their simple net relatives at both coarse and fine-grained levels. Importantly though, the same incongruencies were present between similarity measures and distance measures for the ReLU dense networks as those in the simple networks (see figure 4)—indicating that the difference was due to the activation function.

The incongruency between sigmoid and ReLU activation functions suggested that the sigmoid's squishing of values between zero and one might be playing a normalizing effect on the output of the similarity metrics. We hypothesized that renormalizing the activation values before measuring them in the RDM would eliminate the incongruency between RDMs with similarity and distance measures. This was the case for networks with ReLU activation functions at coarse-grained levels (figure 5a-b). However, it still led to incongruency at a fine-grained level for those networks (figure 5c-d). The same effects were found for networks trained at the fine-grained task and examined at a fine-grained level: renormalization eliminated the incongruency at the coarse-grained level



(a) Pearson (b) Euclid (c) Pearson (d) Euclid

Figure 4: RDMs from dense networks showed the same incongruency as simple networks at both coarse and fine-grained levels. Figures (a) and (b) show the coarse-grained incongruency for a ReLU network between pearson and euclidean measures. Figures (c) and (d) show the fine-grained incongruency for that same network. Darker values indicate representations that are more similar; lighter that they are more dissimilar.



(a) Pearson (b) Euclid (c) Pearson (d) Euclid

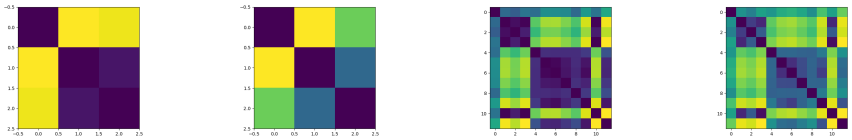
Figure 5: RDMs after renormalization. Figures (a) and (b) show the RDMs for ReLU networks at a coarse-grained level and figures (c) and (d) show for ReLU networks at a fine-grained level. Note that the similarity of some representations are reversed for pearson and euclidean measures. Darker values indicate representations that are more similar, lighter that they are more dissimilar.

but failed to eliminate it at the fine-grained level. Consequently, renormalization did not completely eliminate incongruency.

Comparing different metrics across networks at both coarse-grained and fine-grained levels revealed few differences. Networks trained on the coarse-grained task typically showed the same patterns across each similarity metric; metrics failed to diverge in their results when comparing aggregates of fine-grained representations to their coarse-grained counterparts. Greater variety was found between RDMs constructed from coarse and fine-grain levels for networks trained at the fine-grain task (figure 6). Nevertheless, this variety appears small.

6 Discussion

The results of our experiments lend credence to the first hypothesis: different similarity metrics can result in incongruent RDMs. In particular, our experiments show that there is a real divide in how representations are ordered at both the coarse and fine-grained levels between similarity and distance mea-



(a) Pearson (b) Euclid (c) Pearson (d) Euclid

Figure 6: RDMs from networks trained at the fine-grained task. Figures (a) and (b) show the RDMs for sigmoid networks at a coarse-grained level and figures (c) and (d) show for sigmoid networks at a fine-grained level. The ordering of representations at the fine-grained level largely reproduce at the coarse-grained level. Darker values indicate representations that are more similar, lighter that they are more dissimilar.

sures. While normalizing the raw data of representations can mitigate this incongruency, it does not totally eliminate it.

Importantly, this difference seems traceable to the activation function of the various network architectures. The most striking incongruencies were found in RDMs constructed from networks with a ReLU activation function. More moderate incongruencies were found at the coarse-grained level for RDMs from smoothstep networks. Sigmoid networks failed to show any incongruencies between similarity and distance measures. While individual representations differed between dense and simple networks, the same patterns of incongruencies showed up between ReLU and smoothstep dense networks. This suggests that activation function plays an important role in determining how the similarity metrics order representations.

Our second hypothesis did not fare as well. That hypothesis held that similarity metrics could give different results between coarse-grained and fine-grained levels depending upon the metric. This was not born out by the experiments. While networks trained on the fine-grained task did show greater heterogeneity between coarse-grained and fine-grained level RDMs, there were no clear cases of incongruency. These results do not completely eliminate the hypothesis as tested. Instead, they may be an artifact of how the coarse-grained RDMs are constructed: coarse-grained representations are averages of representations across the four sub-categories that constitute them. It is not surprising that this would result in the coarse-grained level RDMs replicating to a great extent their fine-grained counterparts. This result suggests the following question: are averages a good measure of the coarse-grained representation? There are other ways to compute the relevant representations, such as sums, that may more accurately capture the networks actual response to the coarse-grained labels. The upshot is that an important area of future research would be to address how does one identify coarser representations in models.

Returning to the main problem about choosing similarity metrics, the results confirm that choice of similarity metric matters. The connection between

similarity metric and activation function suggests that said metrics do not necessarily select "similarity" of the underlying representations but mechanical features of the models and brains that produce the relevant representational data. This adds credence to the hypothesis that what matters for similarity analysis is model features (Maheswaranathan et al. 2019). Methodologically, this reiterates an important caveat by statisticians when doing similarity analysis. Similarity analysis is fundamentally a form of clustering analysis, which is in turn a type of unsupervised learning (Murphy 2012, p. 900). There is no objective function to minimize or pre-learning category labels to work with; instead, "central to all of the goals of cluster analysis is the notion of the degree of similarity (or dissimilarity) between the individual objects being clustered. A clustering method attempts to group the objects based on the definition of similarity supplied to it. This can only come from subject matter considerations" (Friedman, Hastie, and Tibshirani 2001, p. 502). Consequently, the work of the analysis is done by the chosen similarity metric and that metric had better be carefully selected for the analysis to target the thing researchers want it to target. And one cannot default to what has worked in the past because

> Although simple generic prescriptions for choosing the individual attribute dissimilarities [...] can be comforting, there is no substitute for careful thought in the context of each individual problem. Specifying an appropriate dissimilarity measure is far more important in obtaining success with clustering than choice of clustering algorithm. This aspect of the problem is emphasized less in the clustering literature than the algorithms themselves, since it depends on domain knowledge specifics and is less amenable to general research (Friedman, Hastie, and Tibshirani 2001, p. 506)

In short, we hope that this paper has contributed to identifying some of that "domain knowledge specifics" that can improve usage of the RSA. Future work would be to better refine what exactly about each activation function is leading to incongruencies in RDMs constructed from networks and whether those considerations actually latch on to the features of computation that neuroscientists are actually interested in.

References

Cadieu, Charles F et al. (2014). "Deep neural networks rival the representation of primate IT cortex for core visual object recognition". In: *PLoS Comput Biol* 10.12, e1003963.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York.

Kell, Alexander JE et al. (2018). "A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy". In: *Neuron* 98.3, pp. 630–644.

Khaligh-Razavi, Seyed-Mahdi and Nikolaus Kriegeskorte (2014). "Deep supervised, but not unsupervised, models may explain IT cortical representation". In: *PLoS computational biology* 10.11.

Kriegeskorte, Nikolaus, Marieke Mur, and Peter A Bandettini (2008). "Representational similarity analysis-connecting the branches of systems neuroscience". In: *Frontiers in systems neuroscience* 2, p. 4.

Kriegeskorte, Nikolaus, Marieke Mur, Douglas A Ruff, et al. (2008). "Matching categorical object representations in inferior temporal cortex of man and monkey". In: *Neuron* 60.6, pp. 1126–1141.

Maheswaranathan, Niru et al. (2019). "Universality and individuality in neural dynamics across large populations of recurrent networks". In: *Advances in neural information processing systems*, pp. 15603–15615.

Murphy, Kevin P (2012). *Machine learning: a probabilistic perspective*. MIT press.

Yamins, Daniel LK and James J DiCarlo (2016). "Using goal-driven deep learning models to understand sensory cortex". In: *Nature neuroscience* 19.3, p. 356.

Yamins, Daniel LK, Ha Hong, et al. (2014). "Performance-optimized hierarchical models predict neural responses in higher visual cortex". In: *Proceedings of the National Academy of Sciences* 111.23, pp. 8619–8624.

Yamins, Daniel L et al. (2013). "Hierarchical modular optimization of convolutional networks achieves representations similar to macaque IT and human ventral stream". In: *Advances in neural information processing systems*, pp. 3093–3101.