B-SOiD Guide

Kailey Quesada

November 26, 2024

1 What B-SOiD Does

B-SOiD (Behavioral Segmentation of Open-field In Deep Learning) allows users to find behaviors using unsupervised learning - without the need for annotated data. Specifically, B-SOiD finds clusters in animal behavior using pose estimation data from another tool such as DeepLabCut. Seven low-covalence features are used in combination with the t-SNE and GMM algorithms to cluster the data [1]. Finally, the likelihood processing step helps to define poses and occlusions [1].

2 Data Requirements

Data for B-SOiD is split between control and experimental. You can choose to use just control data or you can use both control and experimental. An example of where you would use both control and experimental would be if you were comparing between two species of lizard. For both control and experimental, approximately 20-40 videos are needed to train a model for simple animal behavior, assuming each video is approximately 15 seconds long.

3 Usage Requirements

B-SOiD is available in both MATLAB and Python versions. Because B-SOiD relies upon the t-SNE algorithm for clustering during one of its processing steps, the specific implementation of t-SNE can affect B-SOiD's performance. Specifically, the MATLAB version of B-SOiD is able to rely on a MATLAB implementation of t-SNE with more tweak-able parameters than the Python version of t-SNE [1]. While the Python version of B-SOiD was used for this demonstration, it should be noted that, especially for larger datasets, the MATLAB version will likely outperform the Python version of B-SOiD. Should this direction look promising, it is likely that we could use a MATLAB license from Georgia Tech, since MATLAB does cost money.

4 Python Setup

The setup for B-SOiD's Python version is mostly straightforward. The instructions are listed in the project GitHub: `https://github.com/YttriLab/B-SOID`. The steps are as follows: download the GitHub repository, create an environment with the requirements listed in the appropriate requirements file, activate the new environment, and run a python script to start the GUI. For this demonstration, Windows was used. The only issue was that tables was required in the anaconda environment. While B-SOiD is recommended for Python 3.7 and 3.8, tables requires Python 3.9 or higher. The solution was to pip install tables instead of using conda to install. I would also recommend installing "pip install imageio-ffmpeg" up front so that you have full functionality later on.

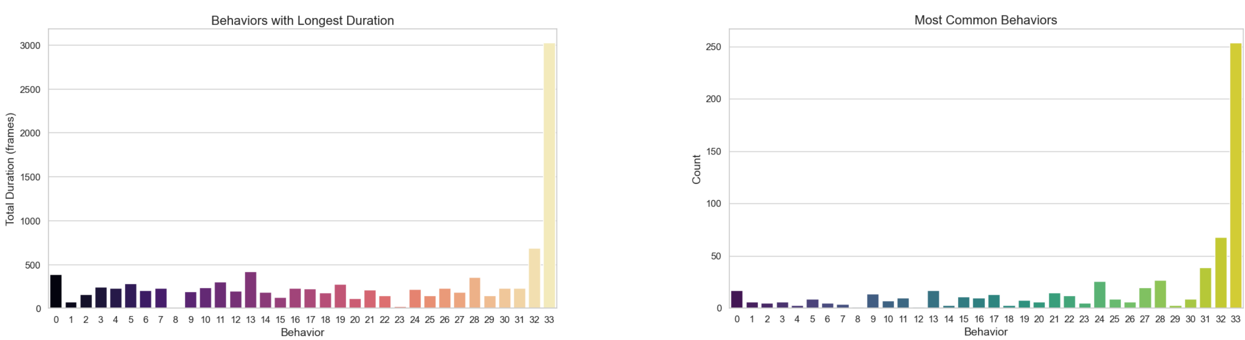
5 GUI Usage

There are both command line and GUI options available for this tool. I will be describing the GUI option here. The first thing to understand about the GUI option is that it assumes a root directory that has two folders, control and experimental. Both control and experimental have the necessary video files and data files. In our case, the videos generated by DLC are in the mp4 format, and the accompanying data files are in the h5 format. Other video formats are accepted by B-SOiD, but mp4 is preferred. For the accompanying data files, either h5 or csv can be used. Additionally, a config.yaml from DLC might be needed in some instances. While videos are not technically required, you will want to include them to be able to visualize the behaviors that B-SOiD is picking up on.

After activating your B-SOiD environment and running "streamlit run bsoid_app.py," the GUI will appear in your browser and the prompts are somewhat easy to follow with some trial and error. Pay attention to what you are checking and unchecking in the toolbar on the left. When you get to the "load previous item" step, make sure that you set your working directory to where the .sav file is located. Typically, this would be under your "output" sub-folder. When outputting the side-by-side videos shown on the B-SOiD GitHub, be sure to edit the default video lengths to get the full video.

6 Cluster Number Selection & Interpreting Results

A big part of the resulting clustering is the minimum cluster size range that the user defines during the clustering process. Minimum cluster sizes that are too small can result in an extreme number of clusters that might not break down into useful behaviors. For example, on the lizard data provided, a human annotator might annotate using only two clusters: moving and not moving. However, t-SNE finds 53 clusters when the minimum cluster size is between 0.5% and 1%, which is equivalent to roughly 2 seconds for the smallest cluster. By changing the minimum cluster size to between 1% and 5%, which is equivalent to roughly 4 seconds for the smallest cluster, t-SNE clustering results in 33 clusters. But what do these additional clusters represent? Using Python, we can take the bout_length csv's in "Bsoid / control / BSOID" and determine which behaviors were most common and which behaviors had the longest durations. This can give us some clues to what is going on. Based on the images in Figure 1, we can see that behavior 33 is the most common and the longest in duration. Behavior 32 is the second most common and the second longest in duration. The other behaviors are somewhat less common.



(a) Comparing Behavior Duration

(b) Comparing Behavior Frequency

Figure 1: Analyzing Behaviors Discovered by t-SNE

Now, the easiest way to interpret the groupings that t-SNE is finding is to look at the grouping examples that B-SOiD exports to the "Bsoid / control / mp4s / subfolder-name" directory. Alternatively, the "Bsoid / output / bsoid_videos" directory can also be consulted. Very quickly, the more common classes will stick out and, if analysis has gone well, B-SOiD will have grouped similar behaviors together based on the pose estimation data.

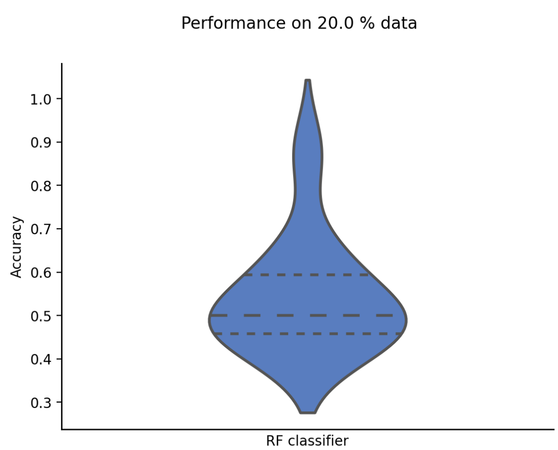
In our case, looking at examples of cluster 32 and 33, we can see that these are running behaviors. Cluster 30 is a pausing behavior with the limbs splayed and the head straight forward or the lizard being at the top of the course unmoving with no key points visible. Cluster 20 is a pausing behavior with the body curled a bit to the right. Cluster 14 is a pausing behavior with the head cocked to the left. Cluster 13 is a pausing behavior with the right foot higher. Cluster 7 is pausing behavior with the left foot higher. Clusters 0, 2, and 4 are where the experimenter hand is blocking all points. Cluster 1 is where the model is jittery can is losing some of the key points. Clusters 27 and 28 are other pausing behaviors with different limb angles. Other clusters represent pausing, occlusion, or movement. See Table 1. Depending on the level of analysis desired and the complexity of the behaviors observed in different groups, the number of clusters should be increased or decreased via the minimum cluster size range.

Cluster #	**Associated Behavior**
Cluster 0	Experimenter Hand Blocking Lizard
Cluster 1	Model Jittery, Losing Some Key Points
Cluster 2	Experimenter Hand Blocking Lizard
Cluster 4	Experimenter Hand Blocking Lizard
Cluster 7	Pausing (left foot higher)
Cluster 13	Pausing (right foot higher)
Cluster 14	Pausing (head left)
Cluster 20	Pausing (body curled right)
Cluster 27	Pausing
Cluster 28	Pausing
Cluster 30	Pausing (limbs splayed, head straight)
Cluster 32	Running
Cluster 33	Running

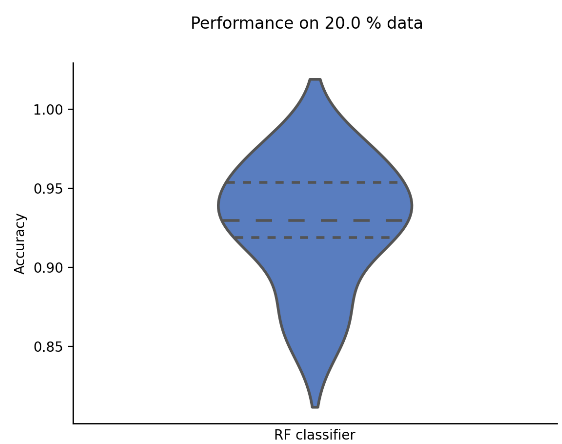
Table 1: Selected Clusters and Associated Behaviors

7 Model Improvement and Performance Evaluation

To determine if the amount of data you are providing is enough for B-SOiD to consistently assign behaviors to the correct cluster, one can evaluate the performance plots produced by B-SOiD, as shown in Figure 2. A model lacking sufficient data will have low accuracy, like the plot in Figure 2a. In contrast, a higher performing model indicates that the model is able to consistently find an appropriate cluster for the data, as shown in Figure 2b.



(a) Learning on 3 Videos



(b) Learning on 22 Videos

Figure 2: Performance at Identifying Simple Lizard Behaviors

From here, the various plots provided using the B-SOiD GUI or command line can be used to make further improvements to the model. These include visualizations of clustering, directed graphs, trajectory visuals, confusion matrices, and k-fold accuracy plots. Additionally, combined videos can be used to evaluate the progression of different behaviors. A screenshot of one such video can be seen in Figure 3. As the video progresses, each frame is associated with a new point in a cluster and the cluster for that frame is printed to the screen.

Figure 3: Example Frame of Side-by-Side Video

8 Further Usage

With the inclusion of additional data from, say, another species of lizard, distinctions between lizard movement based on species could be explored using this unsupervised approach. The beauty of B-SOiD is that it has a very easy-to-use GUI compared to other tools of its type while also allowing users to analyze non-annotated data to learn about animal behavior. For datasets such as the Lizard dataset, which have less complicated behaviors to be learned, this tool would be highly applicable.

References

[1] A. Hsu, *B soid demonstration and use 4/29/20*, YouTube user metalitia4, 2020. [Online]. Available: `https://www.youtube.com/watch?v=wFZFDpUBPjI`.