Cleaning dropout scores

In the neural GLM we are performing model selection and model evaluation by fitting the full model, containing many different regressors, then fitting a series of reduced models with each regressor selectively removed "dropped-out", or all but that regressor removed "single-drop out". This approach allows us to evaluate the relative value of different regressors in describing the df/f signal. We compute two dropout metrics. The cleaning steps below apply to both dropout types.

- The standard dropout tell us the total fractional change in variance explained when removing a kernel.
- The adjusted dropout tell us the total fractional change in variance explained where that kernel has support.

These dropout metrics are complicated by two types of noise. First, cross validation can lead to slight differences in the variance explained from equivalent models. Imagine a full model F and a dropout model D. When the dropout regressor is present, perhaps F can fit its weights in such a way that it generalizes slightly worse than when it is fit without the dropout regressor. This effect can be very tiny, and doesn't really inform us about the quality of F or D. The second type of noise is poor fits overall. If both F and D fit the model well, then the ratio difference between F and D is not largely effected by the cross validation noise. However, if F and D both fit very poorly, then the cross validation noise will be amplified when computing the dropout score. Here I propose a series of steps to clean the dropout scores to make them robust to these types of noise. Note that this doesn't mean there isn't a need for robust regularization and checking for over-fitting, but simply that the dropout metrics are not the proper choice for looking for over-fitting.

Principles for dropout cleaning

The dropout cleaning proceeds from simple rules. With these rules and one free parameter α which is the minimum amount of variance to be meaningful we can make the following cleaning procedure.

- The variance explained should not be negative. If cross-validation causes the variance explained to be slightly negative, we should just interpret that as "the model explains no variance" and clip the variance explained to 0.
- The dropout score is only meaningful if the full model or the dropout explains some minimum amount of variance.
- The dropout metric should be strictly between (-1,0).

Cleaning steps for leave-one-out dropouts

- Compute the variance explained for the full model F and dropout D.
- Force all negative variance explained values to 0.
- Compute the dropout scores
- For any dropout model that explains more variance than the full model, clip the dropout to 0. Note that a large difference here suggests over-fitting, which should be detected and dealt with separately.
- If the full model explains less than α variance, clip the dropout to 0. Note this case suggests poor fits overall, which should be detected and dealt with separately.

Cleaning steps for all-but-one "single" dropouts

- Compute the variance explained for the full model F and dropout F.
- Force all negative variance explained values to 0.
- Then compute the dropout scores.
- For any dropout model that explains more variance than the full model, clip the dropout score to be -1. Note that a large difference here suggests over-fitting, which should be detected and dealt with separately.
- For any dropout model that explains less than α variance, clip the dropout score to be 0. Note this case suggests poor fits overall, which should be detected and dealt with separately.
- If the full model explains less than α variance, clip the dropout score to be 0. Note this case suggests poor fits overall, which should be detected and dealt with separately.

Note the logic is inverted for the single dropouts, because if the dropout explains the same variance as the full model, we want that to be a good dropout score, because it reflects importance of that kernel. So the minimum threshold is applied to the variance explained by each model, not to the difference in model performance.

Important considerations

- How to set the minimum value α? I set it to 0.005 based on a judgement about the size of cross-validation noise I saw on a test session, but I should investigate a principled approach
- Should we add a minimum threshold to the fractional change? For example if the fractional changes is $+0.000001$, it would have gotten snapped to 0 from the cleaning steps above. But a change of -0.000001 would be preserved, despite that clearly being insignificant and likely due to cross- validation noise.

Dropout scores before and after cleaning

Need a visualization of how these cleaning steps impact the scores

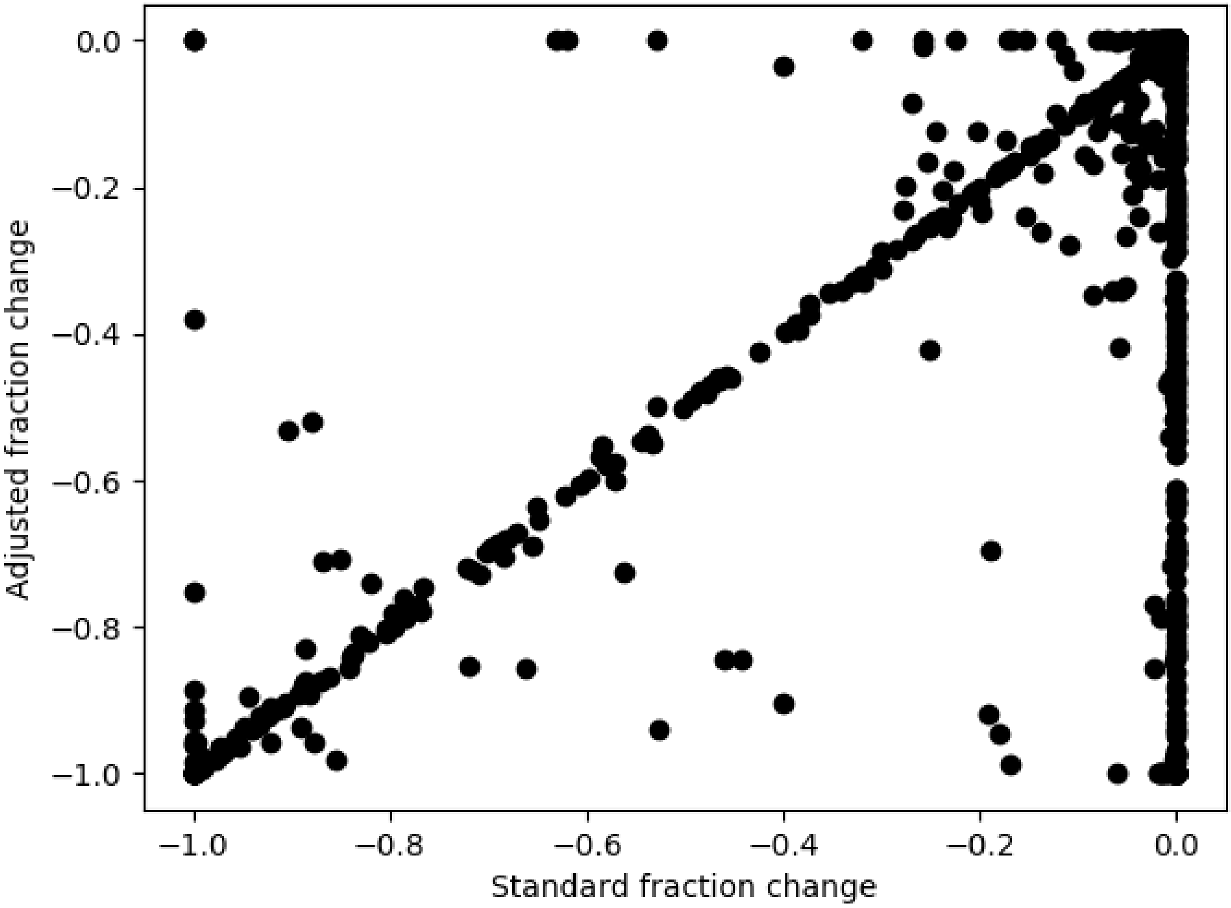

Figure 1: Scatter plot of each dropout score after cleaning.