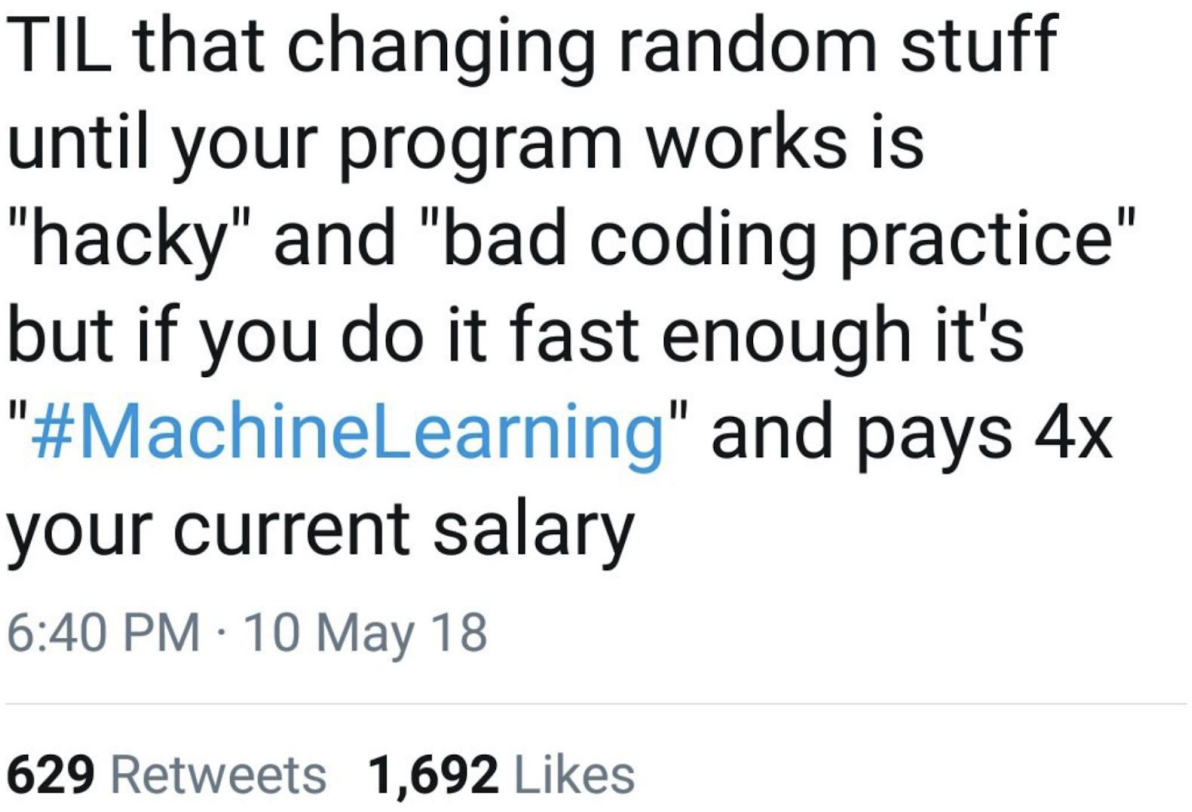


Article URL:
https://medium.com/@brianwahome254/pca-fmri-data-haxby-noise-filtration-fbc39259f472?sk=f06913d902801da7abaea1b950618ba0

Introduction

FMRI data collection entails exposing subjects to magnetic flux and monitoring the variability of the signal on exposure to some stimuli. This variability is what we call activation/signal. The overarching goal is usually mapping stimulus to regions of the brains that are activated when they are induced. Now all this sounds simple and straight forward, but the devil lies in the details. FMRI data is muddied with noise. The noise comes from a wide range of sources ranging from basic head movements, the body's background activation to perform basic body functions like breathing among others and even if we have a dead subject (Shout out to the Salmon experiment), Noise can come from outside interference and end up distorting the signal.

Preprocessing

One of the purposes of preprocessing is to tackle the noise problem. Some methods include slice timing correction, realignment, smoothing and alignment and normalization. Different approaches can be used to perform these steps including threshold setting for signals and Principal Component Analysis - PCA) This article will focus on PCA as a Noise Reduction tool.

Principal Component Analysis - PCA

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component, in turn, has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

Simply put, we project data to a different, lower dimension. As is expected with lower dimension projection, we lose data but, we usually retain most of the information as the order of the components is by variance, where the first component explains the most variance, and the subsequent components explain less variance. This makes PCA superb for data compression and dimensionality reduction which really speeds up our process of performing Machine Learning Analysis.

PCA for Noise Reduction

Now how could we apply PCA for Noise reduction? It has already been shown that using PCA to reduce the dimensionality of data entails information loss. Given that in Its raw format, most of the data collected from FMRI scans are usually Noise, simply performing PCA presents the risk that we will lose the true signal especially since the variance explained by the Signal is very little relative to the Noise. Now, what do we do about this? A key statement we have already made is the fact that the first principal component is usually the most significant, the second more significant and so on. What if we looked at the components that are less significant and Ignore the loud mouth components explaining most of the Variance?

While crude, this is the principle behind PCA for Noise reduction. Depending on what proportion of Noise you want to filter for, you organize your components by the variance they explain and then reconstruct your original data. Most true signal is usually weak and will not stand out in the face of the Noise thus by removing the most significant portion of the raw data, we are left with 'more concentrated soup' of data with the true signal being more visible.

Illustration Assumptions

To illustrate this application, we will demonstrate PCA's potential as a Noise clean up tool. It should, however, be noted that some assumptions we make do not necessarily hold in real life, while some do. These include the following

1. We already know the amount of Noise we have. In real data, we tend not to know the exact amount of variance accounted for by the noise. To filter out the noise, we tend to rely on a lot of experimentation, fitting in a number of parameters and testing different thresholds until we achieve a level we are satisfied with. With PCA, this might entail including a various number of components.
2. We assume our data is Gaussian. For our demo, we will randomly sample noise from a gaussian distribution. In real data too, noise tends to be normally distributed and the net noise signal is assumed to be 0 as positive and negative signals cancel out. Positive signals are spikes which may be similar to our activations albeit not induced by stimuli while negative signals.

Methods

To investigate this, we will make use of the Haxby dataset. This is a block-design fMRI dataset from a study on face and object representation in human ventral temporal cortex. It consists of 6 subjects with 12 runs per subject. In each run, the subjects passively viewed greyscale images of eight object categories, grouped in 24s blocks separated by rest periods. Each image was shown for 500ms and was followed by a 1500ms inter-stimulus interval. Full-brain fMRI data were recorded with a volume repetition time of 2.5s, thus, a stimulus block was covered by roughly 9 volumes.

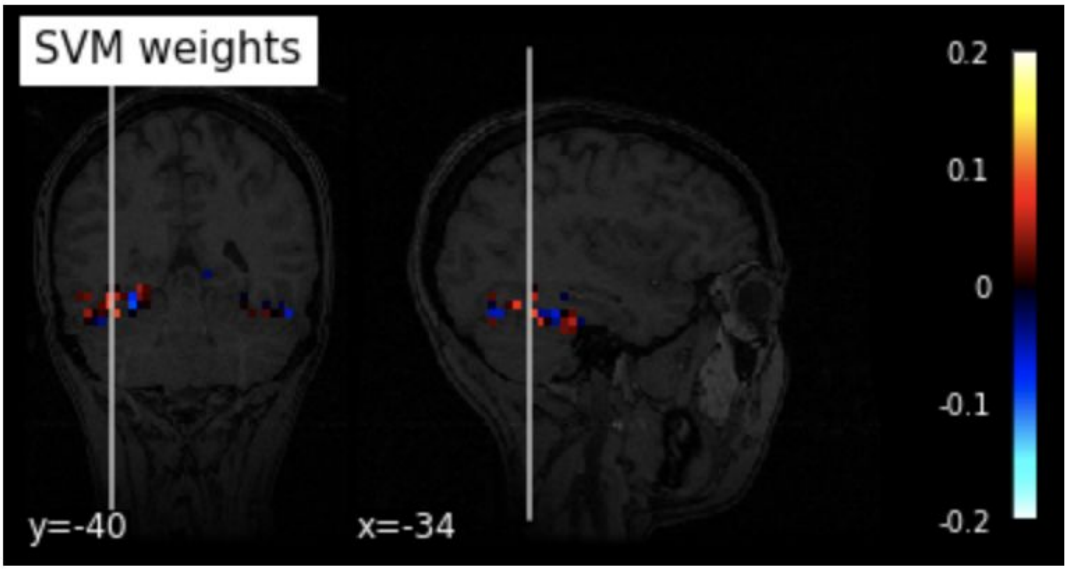
We will get the SAP exported version of the data. To load it, we will follow the nilearn SVM decoding tutorial below to completion. You may find the tutorial here (Nilearn)

In summary, the tutorial above guides us through the process of loading the data, masking it to perform some basic preprocessing steps like smoothing. Do not remove the noise using the nifty masker as we want to make use of PCA for that.

They then fit an SVM to the decode the data, giving us a 2D matrix of the coefficient that we can map onto an anatomical background image of the brain and visualize the activated regions. Some of the activations shown will be noise data.

The final outcome is an image of the overall activated region as follows:

Pre PCA filtration signal plot.



Our extension entails using PCA to get the components that explain about 80% of the variance, or whatever proportion you think noise accounts for. After doing this, we inverse transform the

dimensionally reduced data back to Its original shape, getting an estimate of the data as it would be if it was ‘mostly noise’.
We then find the difference between this reconstructed data and the original data. This difference is what we are most interested in, ideally, most of it would be the activation signal.

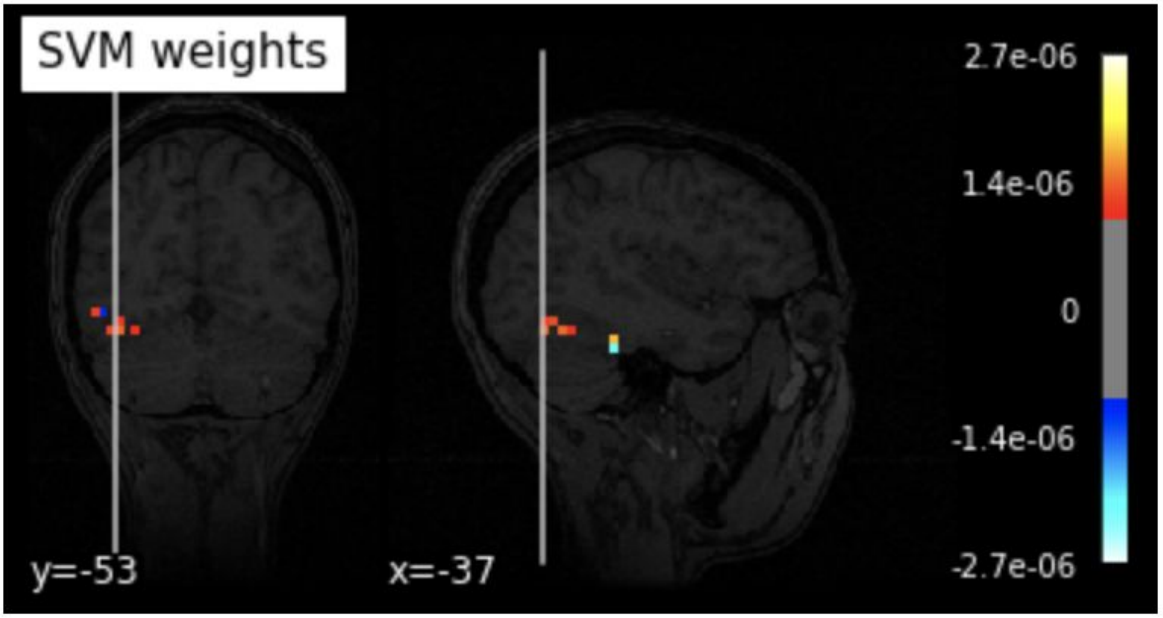
Analogically speaking, Think of evaporating a salt-water mixture, by losing most of the water, we are left will a more salty solution and if done just right, we can be left with pure salt, but a little water never killed anybody. The expected result is made up of only spikes in data with a more precise approximation of the activated region within the visual cortex given that our stimuli are all visual.

Results

We found that post PCA filtration, our signal became more concentrated in the central region and was almost all positive. This is what we expected as some of the signals we plotted in the uncleaned data was going to be noise.

Keep note, what we plot is actually the signal difference thus the scales. The background signal is filtered out leaving only the extra spike expected upon stimuli. The results were surprising as they suggest that only the positive spikes were different from the original hence why in our plot, all the signal was in the positive spectrum unlike for the original, where some of the signal plotted was blue. The region of activation was also more concentrated but the overall location stayed the same, the cortex.

Post PCA filtration signal plot.



Conclusion

It is safe to say we achieved the expected results. PCA stands to be a brilliant tool for Noise reduction in fmri images. This is especially true if we have prior knowledge about the overall proportion of the recorded signal that will be noise, in which case we can use it to tune the hyperparameters of our PCA model. One may think of this process in a similar light to independent component analysis, which has been shown to produce brilliant results when processing fmri data as identifying the true signal is 'as simple' as identifying the component(s) with the highest correlation with an approximation of the expected signal(Green, C. G., Nandy, R. R., & Cordes, D. (2002)). The PCA approach we take is more of a general swipe as we can eliminate noise in droves by just filtering out proportions of the recorded data attributed to noise which might span multiple components.

References

Nilearn: Machine learning for NeuroImaging in Python—Machine learning for NeuroImaging. (n.d.). Retrieved October 28, 2019, from https://nilearn.github.io/auto_examples/plot_decoding_tutorial.html

Haxby et al. (2001): Faces and Objects in Ventral Temporal Cortex (fMRI)—PyMVPA 2.6.5.dev1 documentation. (n.d.). Retrieved October 28, 2019, from http://www.pymvpa.org/datadb/haxby2001.html

Green, C. G., Nandy, R. R., & Cordes, D. (2002). PCA-Preprocessing of fMRI Data Adversely Affects the Results of ICA. Retrieved October 28, 2019, from https://pdfs.semanticscholar.org/9142/59e34b6f3584cbde318bf48db0825b342870.pdf

Code: