Fast Kernel Density Estimation

https://github.com/jsisaacs/fastKDE

Application Towards Medical Journal Analysis

Joshua Isaacson, JT Wolohan

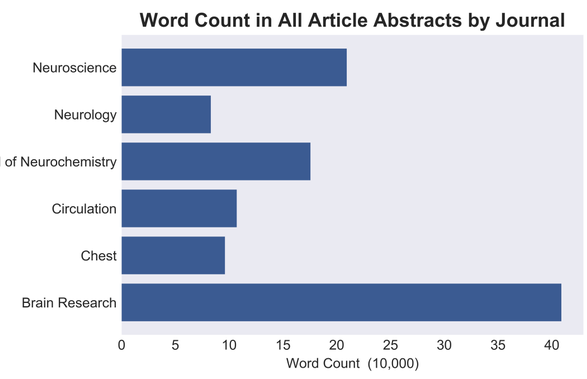
Overview

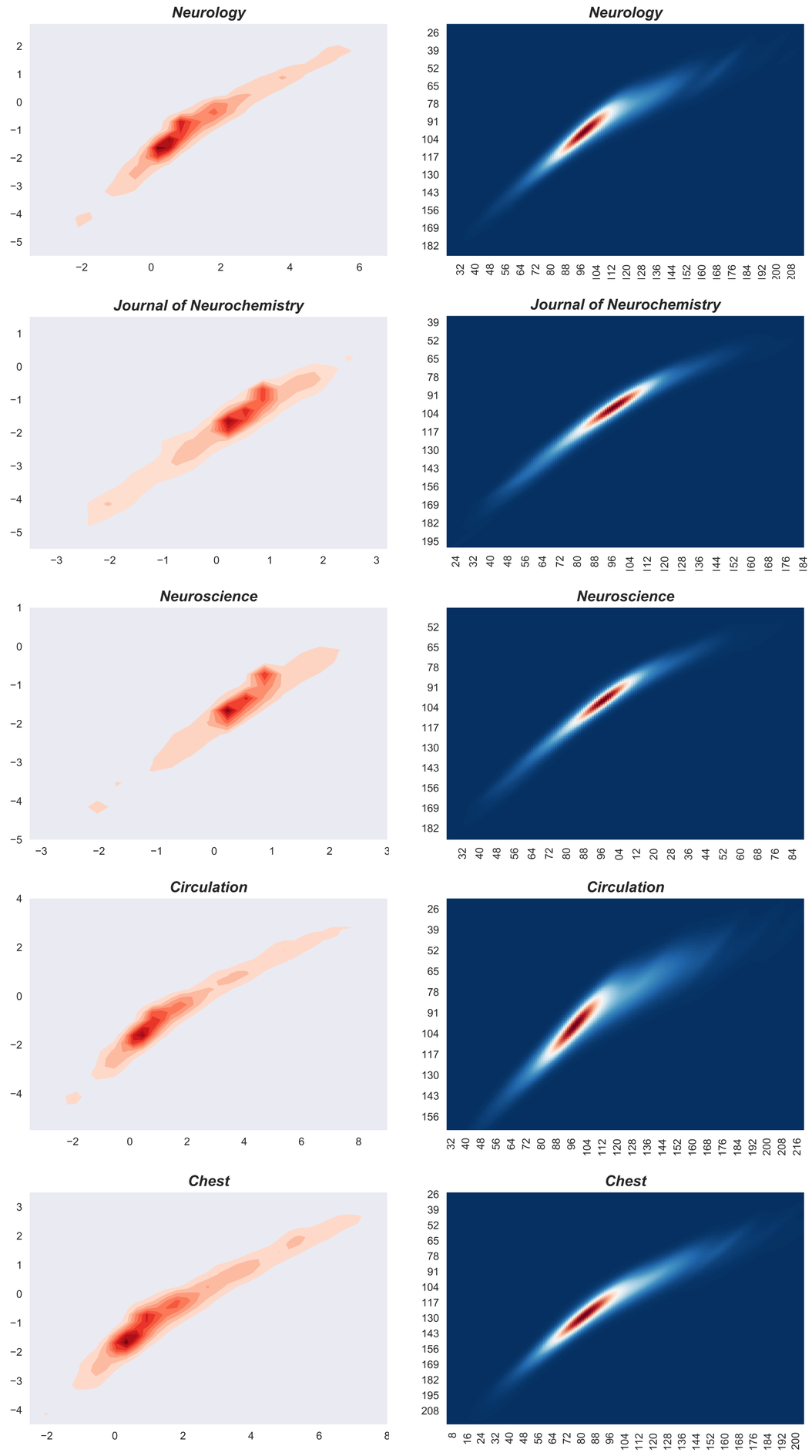
Kernel density estimation (KDE) is a technique for approximating the probability distribution of sample data. Histograms estimate local density, but points must fall into the same discrete bins to be counted correctly. With KDE, the distribution is mapped continuously, which avoids the problems of a rough estimate and data falling into bins that fail to represent their true position.

Currently, there isn't a reliable and fast KDE library implemented for JVM languages. The project's goal is to build such a library and compare it to an existing Python library. It is expected that the JVM library will be faster than the Python counterpart because numerical computing with the Java-based Nd4j library is shown to be faster than numpy, which is what most statistical Python packages use.

Data

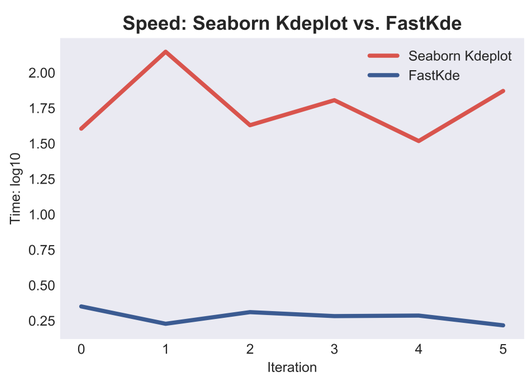
To test the library, we used the PubMed dataset from *data.gov,* which is made up of over 26 million citations for medical research literature. The entire data was roughly 50 GB, which was then parsed to contain only article abstracts and their corresponding journals. We then trained a Word2vec model with the article abstract texts. From there, 6 journals were chosen to be analyzed: *Brain Research*, *Neurology*, *Journal of Neurochemistry*, *Neuroscience*, *Circulation*, and *Chest*. The last two journals are of different parts of the body from the first 4, so we expected their distributions to be slightly different. For each of the 6 journals, we obtained the corresponding word vectors, which were the inputs to our JVM-based KDE library.





Implementation

Our library is based off of the FastKde algorithm created by T. O'Brien, Kashinath, Cavanaugh, Collins, and J. O'Brien in 2016. The inputs are 2 one-dimensional matrices, as well as optional grid dimensions, weights, and adjustment factor. There are seven main components to the algorithm: grid dimension optimization, bin setup, grid creation, bandwidth determination, kernel calculation, 2-dimensional convolution, and scaling by a normalization factor.



For the six journals, the speed to calculate the KDE was roughly 100 seconds for Seaborn Kdeplot and 2 seconds for the Java-based FastKde.

The above twelve figures are the KDE plots of the six journal's word vectors, with the left side being the default Seaborn Kdeplot, and the right being our FastKde library. As the plots show, a greater area correlates to a greater range of words in the associated document, in this case the journal. *Neurology*, *Circulation*, and *Chest* all have broad distributions, which suggests that the journals have a wider range of research articles with more diverse language, compared to densely packed KDE plots, like that of *Brain Research*, Journal of Neurochemistry, and Neuroscience, which could explain that their journals have more concentrated research content.