0.1 Action List

last updated: 10/26/2017

0.1.1 Code

Currently the algorithms are being implemented with tensorflows python api. This will offer us the fastest performance and the software development time. Currently the algorithms are all being run on Shuchin's dell37 server, but I would like to run the software using slurm's scripting language to get full use of the cluster.

2 - tucker

1. trust region methods for choice of regularization constants (log scale search/ armijo's method)
2. load in and iterate over each of the time slices at the same time
3. implement a batch optimizer in tensorflow
4. implement a stochastic optimizer in tensorflow
5. use multiprocessing library to handle naively parallelizable tasks?
6. store the results of the loss function in a vector to make plots for the progression of the optimizers.

h Add in core matrix B into the factorizations

   (a) **Done** add in B symmetry encouraging objective functions

   (b) **Done** project into the eigenspaces associated with positive eigenvalues of B

   (c) using projection methods to minimize von-neumann entropy.(we have a new method for enforcing positive definiteness, however tensorflow has proximal gradient descent optimizers)

t-svd

1. implement t-svd in tensorflow, requires batch processing to compute the fft for each tube.(Will be straight forward once the batch processing routines are finished).

Interface software

1. Bug in the 3D-scatter plots, plotly won't plot large 3D- scatter.
2. remove words in input sentence from result of k-nearest words
3. **Done** properly normalize the rows of the SVD rows

0.1.2 Analysis

How can we properly measure how successful a given word embedding is for a given text corpus.

1. Find metrics for evaluating the word embeddings
    (a) interpolate between missing words?
    (b) Jaccard Index to measure similarity between k - nearest neighbors in svd embedding (could use minHash for speed if needed).

0.1.3 Open questions

1. does aggregating accross PMI matrices perform better than shared PMI slices? How do they differ? content...

0.1.4 Applications

1. Could we apply these methods to analysis of messages shared between peers for an individual to provide a personal embedding? My thoughts are that if we could provide a dynamic word embedding for a person, then it could perhaps be used to measure whether or not the individual needed some sort of intervention if they were suffering from a mental health condition.

2. Could we apply these embeddings to a large text corpus and then plot the path of a vector moving through the text corpus (reading it) and would the path reveal any information about the structure of the document?
    (a) Would have to be more careful to decide on how word representations are combined to represent the state of the path at the tth word.
    (b) end-to-end neural nets may be useful to help make these decisions.