10/31/2017 Research Notes

Currently using the tensorflow routine for computing the embedding minimizing the objective function with the Adagrad optimizer with an initial learning rate of .01 .

$$\arg\min_{U\in\mathbb{R}^{n\times d}} \|P - UBU^T\|_F - \lambda_1\|U\|_F$$

Currently hoping to impose B be symmetric positive definite. We impose symmetry by starting with a symmetric initial matrix and then noting that the gradient will be symmetric now that we're using U for both the word and context embeddings. So B will naturally stay symmetric. To impose positive definiteness, I have implemented a method to project B onto its positive eigenspaces at each iteration. I've done this by computing the eigendecomposition and only keeping the eigenvectors associated with the positive eigenvalues.

Projection Sanity Check:
Let $B = V\Lambda V^T$ be the eigendecomposition for B where V is an orthonormal basis for the eigenspaces. With this, note that we can partition.order the eigenpairs such that $V = [V_p\ \tilde{V}]$ where V_p are the eigenvectors associated with the positive eigenvalues and $\tilde{V}$ are all the other eigenvalues. Similarly we can partition/order the eigenvalues in the block diagonal matrix $\Lambda = \begin{bmatrix}\Lambda_p & \\ & \tilde{\Lambda}\end{bmatrix}$, using similar notation to denote the positive eigenvalues.

Thus we can write the projection onto the positive eigenspaces as $P = V_pV_p^T$ which means that we can compute PA as

$$\begin{aligned}
PA &= V_pV_p^T(V\Lambda V^T) = V_pV_p^T([V_p\ \tilde{V}]\Lambda V^T) \\
&= V_p[I_p\ \vec{0}]\begin{bmatrix}\Lambda_p & \\ & \tilde{\Lambda}\end{bmatrix}\begin{bmatrix}V_p^T \\ \tilde{V}^T\end{bmatrix} = [V_p\ \vec{0}]\begin{bmatrix}\Lambda_pV_p^T \\ \tilde{\Lambda}\tilde{V}^T\end{bmatrix} \\
&= V_p\Lambda_pV_p^T.
\end{aligned}$$

However this method is not proving successful in maintaining the positive definiteness of the matrix, as the next update re-introduces the negative eigenspaces. There is also another key issue with this method as even after removing the negative eigenvalued eigenspaces, which is that this makes the matrix B rank deficient, and thus it is positive-semi definite. Since the number of negative eigenvalues of B has been around 50. So at best, this imposes a semi-norm into our loss functions, where each of the equivalence classes will be quite large because of the size of the kernel of B.

However having discussed this problem with a friend of Shuchin and I, he suggested that we simply multiply the matrix to its transpose to enforce positive definiteness. So I have adjusted the objective functions to minimize

$$\arg\min_{U\in\mathbb{R}^{n\times d}} \|P - UBB^TU^T\|_F - \lambda_1\|U\|_F$$

. However because there are no constraints on U or B, this will not lead to very successful embeddings. I have since realized my mistake and currently implementing an objective function to learn a shared embedding across multiple PMI matrices P_k. minimizing the objective function

$$\arg\min_{U \in \mathbb{R}^{n \times d}} \sum_k \|P_k - U B_k B_k^T U^T\|_F - \lambda_1 \|U\|_F.$$

I will also add in regularizers for the B_k terms in the objective functions.

0.1 Experiment Results

The following subsections are labeled by their file name templates on the dell37 server.All of the embeddings are normalized row-wise. None of the embeddings computed produced significant results. I also produced experiments varying λ_1 on a logarithmic scale, but the also did not show any improvements. I believe the mistake is not using multiple time slices to create a proper core, and without a positive definite B, I have to use a row normalized U, rather than $UV\sqrt{\Lambda}$ which means when testing the embeddings I'm leaving out a critical part of the factorization.

0.1.1 wordPairPMI_2000_iterations_10000_lambda1_-1.0_lambda2_-1.0_dimensions_50_2tf[U,B]

This experiment ran for 10 times as long as any of the other embeddings that I've been running. This function included the projection method for B. However the extra iterations did not improve the quality of the embedding at all. Due to the fact that B was not positive definite, I could not compute a cholesky decomposition or compute the square root of the eigenvalues without introducing imaginary terms.

Note that the λ_2 term in the file name is a relic of the regularizer for the V matrix, which is no longer present in the most recent versions of the software pushed up to the repo.

0.1.2 wordPairPMI_2000_iterations_1000_lambda1_-1.0_lambda2_-1.0_dimensions_50_2tf[U,B].npy

This experiment was run before I had implemented the projection method for B. However this embedding did not produce significant results either.