CS1952Q HW 2 Code

David Heffren

April 2023

1 Assignment

The assignment was given a nonnegative matrix M, find the NMF of rank r of this matrix. That is, given r and M, find A, W s.t $M \approx AW$, $A \in \mathbf{R} >= \mathbf{0}^{mxr}, B \in \mathbf{R} >= \mathbf{0}^{rxm}$. To do this, i used alternating minimization, with each minimization done with gradient descent. In our case, m was 2000 and r was 20, meaning we were most likely going to lose a lot of information.

2 Algorithm

Specifically, I initialized A and W with the rank r svd of M, with all the negative values set to 0, which has a loss of 899.576. Note that the SVD before setting negative values to 0 has a loss of 81.189. Note that this is the same loss that the eigenvalue decomposition (with r largest eigenvalues) gets, which most likely implies that this is the overall optimal factorization of rank r. Using the SVD as inputs, I ran my algorithm to try to A and W which minimize the loss by taking turns minimizing W with fixed A, then A with fixed W. The idea is that doing this will hopefully get the values to converge to some optimal nonnegative A and W.

For each minimization of W or A, I used gradient descent, taking the gradient of the loss function $|(M - AW)|^2$ with respect to A and W, and using the regularizers $|A < 0|^2$, $|W < 0|^2$, where $A < 0$ represents A where values greater than or equal to 0 are set to 0 and negative values are left alone, and likewise for $W < 0$. What this does is that for an optimal nonnegative A or W, after this operation the matrix would be all zeros, since no values are negative. So, this penalizes the "distance" of our matrix from this ideal nonnegativity. So, this regularization term quantifies how negative A and W are, as we want to eliminate their negative values. Note that this alone doesn't ensure that there aren't negative values in the final result. However, to fix this I set the regularization coefficient for these regularization terms to be very high, 30 to be exact.

The important part of this process for getting a good result is to assure that the coefficient terms on the regularizers are so high that the gradients are skewed away from negative values in the matrices. Thus, when at the end you make all the negative values zero, the product will be very similar. This is actually what happened, as making the A and W values nonnegative at the end increased the loss by like .5, which wasn't much at all.

As stated above, at the end of the algorithm, once we've found some optimal A and W, what we do is set all the negative values in A and W to be 0. However, the reason why this is ok is because the intuition is that the regularization term has discouraged negative values to such a degree that the degree of change these matrices will have will be minimal. We're hoping to naturally enforce A and W with almost no negative values, and if they do exist they are small in magnitude.

For the algorithm, we used 1000 iterations of the alternating minimization, with each gradient descent having 100 steps. We used an epsilon of .001 for grad descent, a regularization coefficient of 30 for both regularization terms, and didn't do any fancy grad descent, just default.

3 Results

For results, we got a loss (without regularization terms calculated, but used in descent) of 85.9, and a rounded loss (loss after rounding negative values to 0) of 86.3. As you can see, the algorithm ensured that the non rounded loss and the rounded loss converged, as opposed to the SVD which had a non-rounded loss of 81 but a rounded loss of 900. The regularization term helped fix this. This is most likely very near optimal, considering the last 800 iterations of the minimization only changed it by a tiny fraction. This is actually impressive, as the rounded loss of 86.3 is somewhat close to the most likely "optimal" loss of 81 obtained by the non-rounded SVD. This is reinforced by the fact that when we set the ac and wc values to 0 (removing regularization), the non rounded loss converges to around 81 as well.