CS328: Homework 3

Due date 17/2/2019 @11:59 pm

Submit a Jupyter notebook for each of the coding questions. For the other questions, submit a pdf. Any file that we cannot open will be regarded as not submitted. Likewise for any code that we cannot run.

Copying code is not allowed, from others or any sources. Discussion is fine, but please write down names of people you discussed with.

---

1. Consider the https://grouplens.org/datasets/movielens/ (a train-test partition will be uploaded here soon). We will use a low-rank approximation of Train to estimate the entries that are queried for in Test. Follow the procedure below.

   ```
   Train = set of triplets (i, j, T(i,j))
   Test = set of triplets (i, j, S(i, j))

   Represent Train as a matrix and find a rank-k approximation to Train.
   Let this matrix be named Pred.
   For every triplet (i, j, S(i,j)) in Test:
      err = err + (S(i, j) - Pred(i, j))^2
   Return err
   ```

   Vary k over a range of your choice and find out whether there is an optimal value. Plot k versus the error for k in a range $(0, 100)$. Compare this with the following baseline algorithm: every test entry $(i.j)$ is predicted as $\alpha * \mu_i + \beta * \eta_j$ where μ_i is the average rating of user i, and η_j that of movie j over all ratings in Train , α and β are fitted using training data.

2. Consider the following code for calculating personal pagerank based cuts. Given the undirected graph in the dataset, and the starting node s, implement this algorithm. The find out the plot of that is described below. Let $p \in \Re^n$ and $r \in \Re^n$ be two vectors intuitively capture the approximate pagerank vector and the residual vector (difference between true and approximate). Let v be the source node. Let χ_v be the vector that has 1 at position v and 0 everywhere else. The two methods that you have to implement are in page 6 of the linked paper. The procedures are named push and ApproximatePageRank. Fix ϵ to be 0.01 and $\alpha = 0.2$.

   We will consider the Amazon dataset from SNAP. For this network, as well each of the specified source nodes. call ApproximatePageRank. Let $q \in \Re^n$ be the vector returned. Report how many vertices are in the support of q (i.e. have non-zero value of q). Sort all the vertices in support of q by $\frac{q(u)}{d(u)}$ where $d(u)$ is the degree of node u. Let S_k be the first k nodes the sequence. Plot $\phi(S_k)$,the conductance of S_k, versus k. Also consider the ground truth community label of the starting point and plot the precision of each S_k in terms of this community (i.e. what fraction of S_k lies in this community), versus k.

3. Let $\mathbf{p}$ be a probability vector (non-negative components adding up to 1) on the vertices of an arbitrary connected graph which is sufficiently large that it cannot be stored in a computer. Define a transition matrix T by setting T_{ij} (the transition probability from i to j) to p_j for all $i \neq j$ which are adjacent in the graph, add in self-loops if needed (describe when/whether you have to add any). Show that the stationary probability vector is $\mathbf{p}$. If we wanted to sample from the distribution $\mathbf{p}$, is using this random walk an efficient way to do this? For instance, consider the vertices to be all possible 0/1-strings of length k. There are 2^k vertices and two vertices are connected by an edge if they differ in only one coordinate. Is doing a random walk on this graph more efficient than sampling from $\mathbf{p}$ directly?

4. In class, we studied reservoir sampling where we wanted to store one item. Consider the two following variants of it.

   - We run k independent copies of the algorithm done in class.

- We run the algorithm done in class, except now we keep a reservoir of size k.

(a) If you consider the possible samples from each of these variants, are they the same As in, if a set of items can be sampled by variant 1, can they also be sampled by variant 2, and vice-versa?

(b) Also prove that the second variant actually retains each item of the stream with probability k/m, m =length of stream.

(c) Does this show that the k samples are independently chosen?