|  |  |  |
| --- | --- | --- |
| **No** | **Issues** | **Date** |
| 1. | Bring back 13 pages of CIKM version | Done |
| 2. | Fixing the reference format, especially the author name  Read more papers related to visualization recommendation systems | Done |
| 3. | Fixing big flaws Max-Min Greedy algorithm   * Sort based on diversity like top-1 which is implemented on Swap * Update minimum bound * Observe the pruning performance | Done |
| 4. | Looking for another dataset | Done |
| 5. | Studying and implementing KL Divergence distance to our experiments.   * Impact distance for pruning performance, it may have different performance compare to current approach. | Done |
| 6. | Looking for mathematically proven the maximum bound of Euclidean distance = √ 2 | Done |
| 7. | Max-sum and Max-min diversification | Done |
| **Meeting** | | |
| 8. | Observing impact of K of two DiVE schemes (Greedy and dSwap technique)   * Observe the impact of increasing K while the λ is constant to pruning performance | Done |
| 7. | Apply pruning on Flights dataset, update the total cost figure with the cost after pruning | Done |
| 9. | Rectifying bound mistake while running pruning schemes | Done |
| 11. | Understanding Swap complexity   * CPU and I/O cost especially for the dataset which has large number of attributes. * Calculating the number of distance computation on Swap algorithm | Done |
| 12. | Add more figures in the paper draft   * Paper should has more figures such as Figure to compared between Greedy and Swap | Not sure |
| 13. | Applying multi queries shared computations   * Understanding shared computation of SeeDB * Implementing multi queries shared computation to our DiVE schemes * Compare the performance between shared computation in advanced and shared computation after sorted by diversity | Done |
| 14. | Proposing another distance function of contextual similarity |  |
| 15. | Comparing among all subsets in the dataset instead of only compare each subset with the whole dataset.  Two types of query load in the experiments:   1. Compare between two subset (e.g., disease vs. no disease) – *targeted/ we know what we want to do* 2. Compare between one subset to whole dataset (get all subsets from dataset then compare to whole dataset)   Do we need to compare among all possible subsets in the dataset?  For instance, Flights dataset has attribute = ‘carrier’ and there are a lot of carrier (e.g., AA, US, XX, UU), each carrier can be one subset.  While we only compare between each subset to whole dataset, it only show the trend of each subset compared to whole dataset.  However, It seems interesting while we compare between each subset to others. But it increases the number of combination significantly.  It might rely mainly on user intention. For instance, user want to compare the performance of carrier AA vs XX in terms of arrival delay. Of course it depends on user want. We did not mention about this scenario in our experiments. This issue is also related to “User intention” issue below. |  |
| 16. | Read more related works and write the summarize  Read more papers related to query similarity (Edit distance)  Read more papers related to PI, skewed distribution, etc  Update the related works and future plan of the research  Read AIDE paper, active learning in recommender system |  |