INTRODUCTION

“That’s a lot of data to make any sense! we will have to resort to information filtering”

PROBLEM

- The current approaches to diversification follow the general process-first-diversity-later approach in which
  - a query Q is first executed on the database to generate a result X
  - Result X is then diversified to generate a small subset S
  - The similarity between different points in result X is computed using distance function along all D-dimensions
- This generic approach hinders the performance of applying diversification in high dimensional large databases

CHALLENGE is to efficiently produce diverse query result set by reducing
- CPU Cost
- Disk I/O Cost

APPROACH

- Utilize partial distance computations to reduce CPU cost of diversification
- Compute partial distances between candidate points in X and points in diverse subset S in a h-dimensional subspace where h<<D
- On the basis of the partial distances prune the points that are not likely to be added to S
- Progressively add new dimensions for remaining candidate points till only one most distant point from S is left

- Integrate diversification with query processing to reduce disk I/O cost
- Leverage the features of vertically partitioned data stores to apply progressive diversification based on partial distances

RESULTS