**Reduction in cost of Crowd Sourcing using Skyline Queries**

Notations:

1. A tuple is called a skyline tuple if it is not dominated by any other tuple.
2. A tuple is dominating iff it holds the following condition:  
   For tuple A and B, A dominates B if for every attribute di belonging to A and B, A-di >= B-di  and for at least one d, A-d > B-d.
3. A=B :: Tuple A is equal to tuple B on all attributes. Hence they are not comparable.
4. A>B :: A dominates B.
5. A<B :: B dominates A.
6. The term ‘attribute’ and ‘dimension’ will be used interchangeably. So will the words ‘tuple’ and ‘record’.
7. Ranges will be denoted as R10-15.
8. D-score and R-score - Dominance and recessive score respectively.

----------------------------------------------------------------------------------------------------

**31st May, 2016**

1. Results of using manhattan distance for Movielens Dataset yet to be updated.
2. Using the results, we observed that in a dataset of 400 for NBA , we get an average of 50-55 results for the experiment repeated on the dataset of size 400 but with different values.
3. On comparison with the results obtained from the brute force approach on the standard complete dataset, we found that the manhattan distance approach gave approx 50-60% of the skyline points from the standard. The data was normalized. We used 5 dimensions for the NBA dataset.

Using probabilistic approach to skyline queries:

1. For the S-known set, where we know all the dimensions, we can use the brute force approach for calculating skyline queries.
2. We first judge two points on the known dimensions. If for the known dimensions, partial skyline condition holds, then we explore the unknown dimensions. Else we term them as ‘incomparable’.
3. Suppose out of 5 known dimensions of A and B, A dominates B (A>=B) on 3 of those. A is unknown on the 4th dimension while B is unknown on the 5th (assumption). Since A is dominating on 3 dimensions, the only chance is that A can dominate B entirely. So, let us try and give a higher weightage to A of being a dominating over B,ie, let us provide a weightage to the probability of A>=B by dividing its probability with the number of unknown dimensions as the more the  number of unknown dimensions, the lesser will be its probability to acquire values such that A>=B for all values. Note that this will only be done for conflicting dimensions. Eg. If dim4(A) = R10-15 (range 10 to 15) and dim4(B) =5, then A>=B on dim 4 irrespective of the value it takes. Such dimensions won’t count to the weightage reduction. Such an assumption is based on the fact that for an actual database, the known dimensions are of a higher value and shall contribute more towards deciding the skyline points. Infrequent will be the cases where in a database the unknown values are of more importance than the known values.
4. We are increasing the probability of a potential skyline point because initially our attempt is to cover as many skyline points as we can irrespective of the non skyline points that will be included in the set so that we can perform maximum reductions.We’ll also sort the points based on the maximum known dimensions, in an attempt to reduce the number of comparisons
5. Now, for points A={10,15,20,R5-10,14} and B={4,4,8,7,R6-16}, the method for probability calculation will be (notice that A already dominates B on all the known dimensions.) :

    i) dim4(A>=B) = ((10-7)/15 )\* 1(no.of unknown dimensions in A)

    ii) dim4(B>=A) = (7-5)/(15\*1(no.of unknown dimensions in B) ) (we are reducing the probability of B being able to dominate A as we would like A to be a potential skyline as it already dominates B on the known dimensions.)

iii) dim5(A>=B) = (14-7)\*1/10

iv) dim5(B>=A) = (16-14)/10\*1

v) Prob (A>=B) = Prob(dim4(A>=B)) \* Prob(dim5(A>=B))

   6.    We simply cannot ignore the probability of B>=A as skyline points themselves are incomparable. But in cases where the values of a tuple are low, if we provide equal probability to the unknown values of the dominating as well as the dominated tuple, then chances are that we might be bringing all the tuples to stage where most of them will be incomparable. Eg. if there are 4 unknown dimensions for each point, having an equally likely probability on each of the dimensions can cause either to be not comparable on any of the 4 dimensions with a probability of 25%. So instead of bringing all the points into the potential skyline set, we would like to incur some loss and reduce the set size for the initial stage, considering the fact that a point that is initially low(low on dimension values) will stay low even after crowdsourcing. We would like to minimize such points in the initial set.

 7. Let's assume we have all the probabilities of A>=B and B>=A for all the points.  Now, one thing is that the graph showing the dominance-recessive relation among points will be a dense one, with edges from every vertex to every vertex, and bidirectional. A dense graph saves us the need to traverse the graph in order to inculcate the effect of neighbors on the vertex.

 8. A point will be a potential ‘good point’ for outsourcing if it has a high D-score and a low R-score, with max difference between the D-score and R-score.

 9. To reduce the number of comparisons, one approach is to perform a piecewise brute force, ie, to first compare all the points that have d unknown dimensions. Make the graph for them and, compute the D-score and the R-score, then using this, perform crowdsourcing. After the obtained results, perform eliminations with all other points. The remaining points from the d unknown dimension set can be merged with the points with d-1 unknown dimensions and the process can be repeated.