**When the SKY meets the CROWD:   
A probabilistic approach to cost reduction**

# Problem Statement

Attempt to reduce the cost of crowdsourcing using skyline points.

# Notations and Definitions

1. A skyline tuple is not dominated by any other tuple.
2. Tuple A is said to dominate tuple B iff the following holds:
3. A={a1,a2,a3} B={b1,b2,b3}
4. a1>=b1 and a2>=b2 and a3>=b3 and (a1>b1 or a2>b2 or a3>b3)
5. Tuple A is said to ignore B if any of the above condition fails. In that case A and B cannot be compared and relatively are skylines if not dominated by any other point.
6. A>B: A dominates B.
7. A<b: B dominates A.
8. Tuple⬄record.
9. Dimension⬄ attribute
10. Ranges will be denoted as R (a-b) denoting ‘range from a to b’.
11. D-score and I-score: Dominance and Ignore score respectively.
12. Ignore refers to ‘being dominated’. A point that is too recessive (probabilistic case under ranged dimensions) has a lower probability of being a skyline. So, if the partial dominance condition between two points is satisfied, then the probability of the dominated one being able to dominate the dominant point will lead to non-comparability, hence ignored.
13. D-score = (probability A dominates B based on unknown dimensions)\* (no. of unknown dimensions in A).
14. I-score = (probability A is dominated by B based on unknown dimensions)/ (no. of unknown dimensions in B).
15. **Partial dominance condition**: For points A= {10, 6, 7, R (7-9)} and B= {5, 5, 6, 4}, according to the partial dominance condition, A>=B because A dominates B on all the dimensions that are **known, i.e.** If we were to even consider the relationship between points A and B, then it will only be in case when A dominates B partially. Else in case where based on the known dimensions we cannot conclude the dominance relation, there’s no point in comparing them based on the ranged values.

# Intuition

The intuition behind using skyline points is that because **such points are not dominated by any other point**, they are more likely to appear as top search results depending on the scenario. Moreover, skyline are not only the points that are not dominated by any other point, but also the points that dominate others. Hence using them we can eliminate the points that need not be crowdsourced thereby reducing the overall cost of the process.

# Methods so far

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 OR that they ignore each other. So, let us try and give a higher weightage to A of dominating over B, i.e., 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. E.g. 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 as we use cascading technique to perform eliminations.
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.)
   * 1. dim4 (A>=B) = ((10-7)/5 )\* 1(no. of unknown dimensions in A)
     2. dim4(B>=A) = (7-5)/(5\*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 and not B to reduce the size of our initial set as it already dominates B on the known dimensions.)
     3. dim5(A>=B) = ((14-7)/10)\*1 (emphasizing dominance on A)
     4. dim5(B>=A) = (16-14)/(10\*1) (suppressing dominance on B or suppressing incomparability)
     5. Prob (A>=B) = Prob(dim4(A>=B)) \* Prob(dim5(A>=B))
     6. Prob(B>=A) = Prob(dim4(B>=A)) \* Prob(dim5(B>=A))
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 in magnitude, 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. E.g. 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.

In graph directed G= (V, E) the set of vertices are all the points in the dataset that haven’t been eliminated in the preprocessing stage. A directed edge exists from a point u to a point v if denoting the weighted probability that A dominated B. If no such edge is there then A ignores B (A and B are not comparable). In such a case there won’t be a back edge from B to A if there’s an edge from A to B. Bidirectional edges will occur only in cases when two points have all dimensions unknown. We have a faint idea of how good this approach of being not comparable would be. However, absence of edges denoting non-comparability is still information.

Issue with the model:   
1.You have edge u to v if u dominates v. but what does that have to say about u. Some other point may dominate u heavily.

Instead say you have an edge u to v if u is not dominated by v by prob something. Then higher the sum of outgoing edges, higher is the prob of u being a skyline. This will also give nodes that are skylines but dominate no point a very high sum prob.

2. In your present model, if there is no edge between u and v it indicates they are not comparable and no edge essentially means you have given a added edge value of 0. And if there is an edge from u to v’ it indicates u dominates v with some prob x which means you have an added edge value x. You then say higher prob, more chance of u being a skyline. But this is not true as if I have fewer number of edges out of me then even if the edge values are low it means I am not comparable to many points and hence maybe I have more chances of being a skyline.

I am suggesting that you experimentally also try the below model.

First steps are the same. You ignore points that are dominated. With the rest you create vertices. An edge exists from u to v with prob p if u is not dominated by v with prob p which is basically the sum of your R-score and ignore-score.

Hence the number of edges go up as there is almost always a bidirectional edge. A single direction edge to my understanding will never exist.

Highest sum prob of outgoing edges are most likely to be skyline.

Am I missing something in suggesting this?

If p1 is prob on edge v to u and p2 is prob on edge u to v then you can heuristically decide to eliminate an edge u to v if the p1-p2 > threshold.

This will again reduce the number of edges.  
  
Add details of “A point is eliminated if ….”

Let’s call your present model as Model1 and the suggested one as Model2.

Can you list the advantages, disadvantages of the two?

1. 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 not be a dense one (depends on the diversity of the dimensions having ranged values) and will be disconnected ( as for the points that are not comparable there won’t be any incoming or outgoing edges due to absence of a dominance-recessive relationship), 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. This was to counter the ripple effect.

 8. A point will be a potential ‘good point’ for crowdsourcing if it has a high D-score as well as I-score, as a high D-score will emphasize on eliminations, a high I-score will emphasize non-comparability thereby indicating a potential skyline point.

D-score, I-score formula mentioned in the first section of the document.

 9. To reduce the number of comparisons, one approach is to perform a piecewise brute force (**cascading**), i.e., to first compare all the points that have d unknown dimensions. Make the graph for them and, compute the D-score and the I-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 canbe merged with the points with d-1 unknown dimensions and the process can be repeated.

# Discussions

Q.)What is partial skyline condition?   
  
Partial skyline condition states that the skyline conditions for two points should hold for all the known dimensions of the two points. E.g. if dimensions 1, 2, and 4 are known for points A and B, then either A>=B or A<=B should hold for these three dimensions. If not, there’s no point comparing them on ranged dimensions.

Q.) What does “weightage reduction” mean? What would have happened if in the first three dimensions A did not completely dominate B?   
  
We ignore it. There won’t be a case (or very rare) where there’s a point that is not comparable to lots of points that are getting eliminated by skyline points as well as not comparable to a skyline point as well. At least that’s what we thought when Shivam and I were discussing about it. Always there’s a possibility of outliers where such a point could exist, but we were thinking of getting some results first, then trying to improve them by making changes to the methods that led us to the results. We are prioritizing the probability of a point dominating other. We thought that a major role will be played by the ability of a point dominating others rather than being dominated or being not comparable. So we decided to boost the probability of dominance. So, if a point is dominating 5-6 points, even if there is a fair chance, we are increasing its probability of being amongst the initial points that will be crowdsourced. Again, this is one way of dealing with the problem. We discussed that it would help reduce the size of the initial crowdsourced set by having maximum eliminations so that even though some point that may actually be a skyline might get missed, some point that should not be crowdsourced will have a very low chance of being a part of this set unless it’s an outlier kind of a point or very close to skylines.

Q.)Not clear. What is the 7, 15? Can we formalize this to make it easy to understand? Let’s say A={a1,a2,a3,R(x-y), a5} and B={b1,b2,b3,b4,b(x’-y’)} then ......  
  
Other points follow calculations based on weightage. Dim5 (A>=B) refers to A and B compared on dimension no. 5, representing the probability of A dominating B over that particular dimension. It’s based on the same approach that you were using when you dealt with ranges initially. We blurred the overlapping region instead of being so discrete about it, which was causing us a lot of computation problems like we previously discussed. Unlike your approach, we are not ignoring it completely. So we have considered it in a range but in such a way that it gives value to both but differently based on the difference in ranges. I hope the remaining points are a bit clear. We’ll still try to come up with a concrete idea of how to write all this more formally so that the same can be carried forward and need not be changed from time to time.

# The Procedure

1. Iterate through all the points and calculate the D-score, I-score and non-comparability (points that are clearly not comparable either through the partial dominance relationship or else).
2. Create a graph where a directed edge from A to B indicates the probability of A dominating B (D-score). Also another edge connecting A and B indicated the I-Score, i.e., the probability that A and B won’t be comparable.
3. The graph will look as follows:   
   a. All the points that are skyline will appear as disconnected points on the graph.  
   b. All the points that have been dominated won’t appear in the graph.  
   c. All the other points shall appear with edge relationship as above.
4. Use the above graph to gather a set of initial points for crowdsourcing using all the skyline points and potential skyline points. Crowdsource the set and use the results to perform further eliminations. Repeat till all the necessary points have been crowdsourced.

# Change Log

Changed the definition of R-score to I-score.

Defined I-score in the definition section.

Included a ‘Procedure’ section in the document.

Revised calculations of D-score and I-score in the method section.

# Comments

Please mention any heading you’d like to comment on along with the comment here. The update section will be included into the main body after you approve of it. Then it will be cleared so that the next updates can be mentioned there.

# XYZ