Analyzing a Greedy Approximation of MDL (Minimum Description Length) Summarization with Holes

Peter Fontana
Faculty Advisor: Dr. Sudipto Guha

Abstract:
This project analyzes a greedy algorithm for the MDL (Minimum Description Length) Summarization with Holes, and proves a tighter bound that changes the algorithm from a $\log(M)$-approximation (M is the size of the matrix) algorithm to a constant-factor algorithm that only is affected by the specification of rectangular regions (rectangles) that the algorithm can use to summarize the data matrix. This poster illustrates the special case where the matrix is 2-dimensional and the available rectangles are rows and columns. However, this bound has also been proven for a multi-dimensional matrices with an arbitrary specification of available rectangles.

Old Bound: $O(l_m \cdot \log(M))$ \rightarrow $2 \cdot \log(M)$ (Special Case)
My New Bound: $2(k - 2) \rightarrow O(k)$ \rightarrow $4$ (Special Case)

Definitions:
- $k$ is the largest number of rectangles which contain a common cell in the matrix. Two such rectangles, the entire matrix and the individual cell, are trivial. Excluding those rectangles gives ($k$-2) rectangles.
- $l_m$ Take each cell of the data matrix. For each cell $u$ of the matrix, consider all of the rectangles that contain $u$. The contains (subset) relation forms a partially ordered set over the rectangles, and more specifically a lattice, since the cell and the matrix are considered rectangles. $l_m$ is the size of the largest antichain of the lattice for all cells $u$.

While in the particular case ($k-2)=l_m=2$, in general this is not the case, with ($k-2$) often being larger than $l_m$. Here, the matrix and the individual cells are always available rectangles, but are considered trivial and are only considered in the definitions of $k$ and $l_m$.

Motivation of Algorithm and Project:
- OLAP (On-line Analytical Processing) applications produce data cubes that can convert a data table into an aggregated matrix that converts every cell to either a 0 or a 1 based on if that cell has a certain property. This project studies how to describe these aggregated matrices concisely.
- It is desirable to represent these matrices as compactly as possible, since the matrices can be large and with many dimensions, taking up a large amount of storage space.
- To compress the matrix, the matrix can be described by describing rectangular regions (rectangles) of the matrix that are all 1s. This is MDL (Minimum Description Length) Summarization. However, since every cell in the region had to be a 1, this limited the number of regions that could be chosen.

Description of Greedy Algorithm:
1. The Greedy Algorithm Reads in the Matrix
2. The Greedy algorithm picks an uncovered rectangle, and it looks at all its uncovered 1's and all the uncovered 0's and computes the following ratio: $1 + \#$ uncovered 0's
3. The Greedy algorithm repeats step #2 for all uncovered rectangle. While the greedy algorithm is doing this, it keeps track of the smallest rectangle and the rectangle that has that ratio.
4. The Greedy algorithm then looks at the smallest ratio, and checks to see if it is less than 1.
5. If the ratio is not less than 1,
   a. the algorithm terminates and returns the summarized Matrix and the cost of the summarized matrix.
   b. The summarized cost of the matrix is the #rectangles chosen + #uncovered 0's + #uncovered 1's.
6. Otherwise (the ratio is less than 1),
   1. The Greedy Algorithm chooses that rectangle. It marks that it chose the rectangle.
   2. The greedy algorithm covers each cell in that rectangle.
   3. The Algorithm goes back to step #2.

Conclusions:
- The New bound is not tight. Furthermore, I speculate that the factor of 2 in the new bound can be removed.
- The New bound is progress. It eliminates the size of the matrix as a parameter in the approximation ratio. This converts the algorithm from a $O(\log(M))$-approximation algorithm to a constant-factor-approximation algorithm for both, this is for the special and the general case.