Water Quality Database Project Report

by: Caden Oslund

Introduction:

For my final project, I decided to make a system for tracking and potentially predicting water quality based on data gathered by the EPA on water sources across the nation. This system could be potentially useful for detecting drops in water quality before they happen, allowing counties to prepare for problems before they happen and minimizing the damage. My plan to do this is by creating a database that relates the water quality measured by different tracking stations so that when one has a change or problem it can predict which others will be affected. This database could be incorporated into an automated system that can issue warnings if there is likely to be a problem and give people extra warning to stock up on water filters or bottled water before there is a significant dip.

Task and Questions:

My task was creating a database that could efficiently find relations between water sources based on water quality data from them, and process this information to find likely and/or common correlations between them. For example, if Station 3 is closely related to Station 7, and the water quality at Station 3 suddenly drops, it can be predicted that Station 7 is likely to have a problem in the near future. In theory, this system is sound, but it has several problems such as a large pre-processing workload and a small resulting dataset. I elaborate on these issues in the Evaluation of the Methodology section of this report.

Methodology:

My methodology for this task is to create an item-item dataset of the different tracking stations based on similarities in their water quality. I used the Apriori algorithm to calculate similarities and produce the result. Once I got a usable dataset, I put the different tracking stations into buckets based on their water quality. If their Dissolved Solids were within 100 mg/l of each other, they were put in a bucket together. The majority of the work went into pre-processing the data. It was difficult to get a useful dataset from the EPA website. They have a system for looking up individual values or small groups, but it was harder to get a large dataset to use for my algorithm. I ended up getting a dataset of all water quality samples from 1950 through 1998 because I needed the largest initial dataset that I could get, and samples from before 1950 are sparse and inaccurate.

Algorithm Pseudo-code:

```
for every datapoint
    for every other datapoint
        if water quality of datapoint 1 is within 100 of datapoint 2
            add datapoint 2 to the datapoint 1 bucket

runs Apriori algorithm on the buckets
returns output triples
```

Evaluation of the Methodology:

The overall methodology is sound but is limited in its usability by the large effort that is required for pre-processing as well as a small resulting dataset. Once I had the dataset from the EPA I had to remove all unnecessary data, which took a long time because out of over 65000 rows and over 30 columns, I only ended up being able to use 419 rows and 2 columns. I then had to convert all of the units of measurement to be the same and removed all decimals as they aren't very useful for the final code and make it take longer to process. In the end, less than 1% of the data was actually usable in my algorithm. Once all of that work had been done, however, the Apriori algorithm worked very well at efficiently finding matches, and most of those found were very high confidence. I used Dissolved Solids as my metric for water quality as it's a single value that gives a good indicator of general quality. I decided to only keep triples resulting from the algorithm as there were very few pairs or anything over triples, and they were not as accurate.

Results and Conclusion:

Here is an example of the output data:
CALWR_WQX-A0922000 CALWR_WQX-DMC06716 CALWR_WQX-KA030341 1.0

The way that this would be interpreted is that if CALWR_WQX-A0922000 experiences a drop in water quality, CALWR_WQX-DMC06716 will likely experience a similar effect soon as well. The number at the end is confidence, which is 100% certain that this correlation is true. While that number is higher than it should be, it's still a good result as all 3 tracking stations are in the same area because they all start with CALWR. This means that the algorithm is working as intended, as most of the high-confidence groupings are in the same state or adjacent state, such as stations in California being grouped with ones in Oregon. The only main problem with the output is having very high confidence values, which is likely a problem caused by having a small initial dataset.