StreamDiv Project Report

Steve, Nick, Nathan Ong, Zaeem, Panos Chrysanthis and Alexandros Labrinidis
Department of Computer Science
University of Pittsburgh, Pittsburgh, Pennsylvania 15260
Email:

***Abstract*—Data mountains continue to grow in size, increasing the difficulty in retrieving useful analysis from them. In addition, generated data is frequently plagued by sameness; the data that is retrieved tends to follow the same pattern, helpful in simple prediction tasks or outlier detection, but not necessarily helpful in characterizing the data from a stream. Users tend to know in a general sense what they are searching for in their datasets and streams, but may not know the full space of relevant results. Diversity helps alleviate this problem, but faces its own challenges when combined with relevancy and recency in streaming contexts.**

In our work, we explore different designs for a diversity operator in a stream processing environment. In addition, we factor in the impact of recency and user-defined relevancy by ranking incoming data. Users will specify a size for a representative top-k set for the data seen from the stream. This top-k set is maintained by the system and updated based on recency, relevancy, and diversity from continuous streaming data. We consider two types of tuple-replacement schemes, Incremental and Batch, for maintaining the top-k set, and find CONCLUSION.

I. Introduction

Data creation and collection only continues to increase, leaving a large burden on analysts to come up with a way to quickly and efficiently comb through the data to locate trends and correlations. There tends to be a focus on data-specific approaches with frameworks that work well for certain kinds of data (but not well on others CITATION NEEDED). There also is a need in data analysis to provide relevant but diverse data points as a digest of the full data set. Several offline versions have been created [1] CITATIONS NEEDED, but lack the speed to work in a streaming environment. We present StreamDiv, a streaming framework built on top of a previously developed diversity system for offline database systems, PrefDiv[1], to provide queriers of a data streaming systems a top-k set that is most relevant and diverse, adding recency as a possible influence on the digested set.

The system model of our work is a stand-alone stream processing system, with a single stream of data as input. Processing is modeled as a pipeline of operators, which process incoming tuples and forward results to the next operator. The output is not a single result (as in the case of traditional databases), but periodical results that reflect the most sensible outcome (according to the algorithm) based on time-windows.

The pipeline is as follows: from a single input stream, a timestamp is appended to the tuple based on the time the system received it. The tuple is then sent to a Relevancy Operator where it is given a score based on its relevance, which is appended to the tuple. Then the Diversity Operator takes the tuple, and uses it to maintain the top-k representative list, which is then sent to the user based on his or her defined window interval. Maintenance of the top-k set is governed by the replacement policy, *Incremental* or *Batch*. The Incremental Replacement Scheme attempts to replace tuples in the top-k set as tuples come into the system, while the Batch Replacement Scheme replaces buffered tuples in a user-specified time or tuple-based window.

II. Related Work

Citations and stuff

III. Problem Definition

A. Input Output Relationship

Let k be the number of tuples of S, where $S = \{t_0, t_1, \ldots, t_{k-1}\}$ representing the top-k set. Let n be the number of tuples of N, where $N = \{t'_0, t'_1, \ldots, t'_{n-1}\}$ representing the set of incoming tuples from the stream. (Note for the Incremental Replacement Scheme, $n = 1$.) Our goal is to take some tuples from N to replace some tuples in S based on recency, relevancy, and diversity.

B. Recency

Recency is defined as a monotonically decreasing score, computed as some formula relating the tuple's timestamp of arrival to the system T_A and the current system time T_C. We choose two different formulas to capture a stronger and weaker emphasis on the recency score of a tuple $Rel(t)$.

1) $Rec(t) = -|T_C - T_A|$
2) $Rec(t) = e^{-|T_C - T_A|}$

The first provides a linear decay, allowing older tuples the possibility to persist longer in the top-k set. The second provides an exponential decay, which heavily prefers newer tuples.

C. Relevancy

Relevancy $Rel(t)$ is a user-defined function regarding what tuples are considered relevant in a user's query. We provide an explanation as to what constitutes Relevancy in Section IV, as the definition is dependent on the query.

D. Diversity

As far as different dimensions of the problem are concerned, ranking is considered as a set of user-defined properties that are either attached to incoming tuples (annotate), or are used for filtering out tuples. Turning to the diversity operator, we abstract it as a stateful operator, with time- based sliding windows. The operator's frequency of result production can be one of the following: (i) a temporary snapshot of the most diverse tuples in the current time-window produced every time

a tuple is received, (ii) a set of the most diverse tuples among all the tuples that have been received in a user-defined epoch. In the latter case, we consider how many tuples are replaced and which ones every time a new tuple arrives. For the former case, we consider a static approach (a pre-defined fraction p of the tuples are replaced after every epoch), and a dynamic one (the fraction p of the tuples replaced is proportional to the δ in diversity gained compared to the replacement fraction of the previous epoch). As regards to replacement policy, we have the alternatives of random replacement, and least recently tuple received replacement.

IV. Experiment

asdlkfjaow

V. Results

VI. Conclusion

VII. Future Work

sadlkfjafk

Acknowledgment

The authors would like to thank...

References

[1] X. Ge, P. K. Chrysanthis, and A. Labrinidis, "Preferential diversity," in *Proceedings of the Second International Workshop on Exploratory Search in Databases and the Web*, ser. ExploreDB '15. New York, NY, USA: ACM, 2015, pp. 9–14. [Online]. Available: http://doi.acm.org/10.1145/2795218.2795224