SCALING GEOMETRIC MONITORING OVER DISTRIBUTED STREAMS

by

Alexandros D. Keros

A thesis submitted in partial fulfillment
of the requirements for the degree

of

UNDERGRADUATE

in

Electronic and Computer Engineering

Approved:

Dr. Vasilis Samoladas Major Professor	first reader Committee Member
second reader Committee Member	dean

Technical University of Crete
Chania, Crete, Greece

2015

Scaling Geometric Monitoring over Distributed Streams

by

Alexandros D. Keros, Undergraduate

Technical University of Crete, 2015

Abstract

BLAH BLAH

Thesis Supervisor: Dr. Vasilis Samoladas
Department: Electronic and Computer Engineering

(27 pages)

Public Abstract

BLAH BLAH

Acknowledgments

my mum

Contents

List of Tables

List of Figures

Part I

INTRODUCTION AND PRELIMINARIES

Chapter 1

Introduction

1.1 Overview

1.2 Motivation

1.3 Contributions

1.4 Thesis Outline

Chapter 2

Theoretical Background

The present chapter contains the necessary background knowledge used throughout the length of this thesis. Section 2.1 describes the *"Geometric Approach to Monitoring Threshold Functions over Distributed Data Streams"* in detail, as formulated by I.Sharfman, A.Shuster, D.Keren [**?**]. Section 2.2 presents *multi-objective optimization* and dives into the algorithms used in our implementation. Section 2.4 discusses *graph maximum weight matching* used for node pairing and, finally, in Section 2.3 we explain the *Savitzky-Golay filtering* used for velocity approximation.

2.1 Geometric Monitoring of Distributed Streams

The Continuous Distributed Monitoring Model, a.k.a. Data Stream System

idea: having a real-time overview over the system

application examples: ISP network traffic, distributed sensors etc

complexity: monitoring value or threshold monitoring over the whole set of observations, in real time

goal: minimize communication while retaining the highest accuracy possible

possible solutions:

1.centralize

- suffers from network overload, storage overload

2.poll

- not real time, update frequency-accuracy trade-off

3.GM monitoring

-apply convex opt theory in order to reduce communication while retaining accuracy bounds

details of geometric monitoring model

2.1.1 System Architecture

fully distributed node topology

.no coordinator-center node

.communication between nodes

image

coordinator based node topology

.coordinator-center node

.nodes communicate only with coordinator

image

2.1.2 Computational Model

stream and node notation

weights

statistics vectors

global statistics vector

monitored function

threshold

estimate vector

drift vector

general operation of distributed algorithm

drift vector definition

general operation of coordinator based algorithm

balancing process

slack vector

drift vector definition

2.1.3 Geometric Interpretation

node local constraints make sure global violation is accurately monitored

how?

convexity property of drift vectors

theorem of bounded convex hull by local constraints (balls)

monochromaticity of balls

balls monochromatic means threshold upheld

2.1.4 Protocol

decentralized algorithm (in short, for completeness)

centralized algorithm (in detail)

we will focus on that

2.2 Multiobjective Optimization

what is mop

use examples

kinds:

a.numerical

b.evolutionary

2.2.1 Sohr's algorithm a.k.a. ralg

algorithm description

2.3 Savitzky-Golay Filtering

filtering generals

examples of uses of filters

filters:

Kalman

+,- Moving Average

+,- Savitzky-Golay a.k.a. ??? +,-

algorithm description

2.4 Maximum Weight Matching in Graphs

general graph theory (introductory)

what is max weight matching

algorithm description

Chapter 3

Related Work

cite papers working on the original metioned above

function specific stuff

bounding ellipsoids

reference vector change (estimate vector)

safe zones

prediction

matching

*no work on slack vector distribution during balancing, we do!

Part II

PROBLEM DEFINITION AND IMPLEMENTATION

Chapter 4

Problem Statement

papers in chapter 3 do not scale well

why?

where exactly?

we try our luck at it, how? (one liner)

Chapter 5

Implementation

implementation details

assumptions made (before we dive deep)

5.1 Geometric Monitoring

*in short!, reference original paper (and other using same stuff)

system arch (centralized)

computational model (main vectors)

!exclude balancing process!

5.2 Heuristic Balancing

balancing process geometric interpretation

image

balancing process algorithm

5.2.1 Velocity Estimation via Savitzky-Golay

S-G implementation

image

5.3 Distance Based Node Matching

node matching interpretation and why I did it (i.e. follow global statistics vector)

image

distance based node matching algorithm

5.4 Implementation Challenges

training data

complexity of optimization (i.e. optimal point location)

complexity of optimization (i.e. node matching)

Part III

RESULTS AND CONCLUSIONS

Chapter 6

Experimental Results

experimental result showcase

6.1 Experimental Setting

dataset used

reference appendix for tools, mention in short

6.2 Heuristic Balancing

comparison with classic balancing

!random matching!

explain

how S-G affects results

6.3 Distance Based Node Matching

comparison with random matching

comparison with distribution node matching

!use same balancing, both classic and heuristic! (i.e. 1st all with classic, then all with heuristic)

explain

6.4 Overall Results

summarise results

compare classic random and classic distribution optpair with heuristic distance optpair

observe how S-G affects results again

explain

Chapter 7

Conclusions and Future Work

conclusions

7.1 Conclusions

problem statement in short

what has been done in short

our contributions

short explanation of contributions

7.2 Future Work

References

Appendix

Chapter A

Geometric Monitoring Python Implementation

A.1 Python

what is python

why python

A.2 Numpy and Scipy

what are they

why use them and how

A.3 Openopt

what is it

details about framework

A.4 NetworkX

what is it

details about framework

A.5 Putting It All Together

code description

UML

how to run