Divide and Recombine (D&R) for Analysis of Large and Complex Datasets

Computational Environments

Background: The Problem

Large, complex datasets are everywhere in science, engineering, and business

- overwhelming our statistical methodology and our computer environments for the analysis of data

Goal: comprehensive, detailed analysis of large, complex datasets that does not lose critical information in the data

Two Categories of Methods for Data Analysis

Mathematical methods: automated learning by the computer

- formulas and algorithms are computed
- the output is numeric and categorical values
- e.g., fitting a regression model
- e.g., support vector machines for classification

Visualization methods: human intervention to guide the automated methods

- graphs are made of raw data
- graphs are made of results from mathematical methods
- the output is a visual display
- e.g., scatterplot
- e.g., normal quantile plot

Two Categories of Methods for Data Analysis

Math and vis methods

- symbiotic: need each to do the other
- both are needed for all datasets, large and small

Visualization: plays the bigger role in preventing loss of critical information

Large, Complex Datasets

Must visualize the data at its greatest level of detail, but not necessarily all of the data

A good way to lose important information in a large, complex dataset

- apply a statistical or machine learning method to detailed data
- visualize just the data-reduced output
- use the excuse that because there are too many detailed data to visualize all of them, none of them should be visualized

The Dynamic Language for Data Analysis

Must be easy to "program with the data" and yet have extensibility

- methodology: prototype new methods
- data analysis: tailor the analysis to the data and not be stuck with just off-the-shelf routines

Goal for the data analyst

- minimize human time
- work exclusively eternally within the dynamic language
- not engage a lower-level programming language

Goal for the methodologist

- minimize human time
- engage a lower-level programming language for speed of computation only after prototyping

Why dynamic and not batch

- prototyping of methods involves typically many trials and revisions
- data analysis needs to be a sequence of steps with analysis decisions at step k dependent on steps $1, \ldots, k-1$

The Computational Environment for Data Analysis and Methods Development

Must provide for calls to the 1000s of methods of statistics and data analysis

Want the environment to serve any dataset

Want the environment to provide for development of any method

Methods written in

- the dynamic language
- a lower-level language like C, and called by the dynamic language
- hybrids
- research in methods is enhanced when all source code for methods is open

The Computational Environment: Large Complex Datasets

It's all a beautiful story if we have a small amount of data

But large, complex datasets are now ubiquitous

Overwhelming our computational environments and our methods, and forcing inefficient moves to lower-level programming and one-off's

Lee Edlefsen: "We have taken a step backward in our effectiveness to analyze data."

The computational environments need to change

The machine learning and statistical methods need to change too

Toward a Solution: Large Complex Datasets

We absolutely do not want to give up the power of the dynamic language just because we have lot of data.

Toward a Solution

1. Hardware

Cluster or cloud (modest will work)

Software

2. Parallel, external-memory algorithms for mathematical methods: see Lee Edlefsen

Toward a Solution

How can we apply any statistical method to a large, complex dataset right now, e.g., any R package?

Toward a Solution: Divide and Recombine (D&R)

Divide the data into subsets

Spread the subsets across the machines of a cluster or cloud running our dynamic language

Apply each of many method to each subset

Mathematical methods: typically apply to all subsets

Visualization methods: typically apply to a guided sample of the subsets

Recombination of the results

- a summary of analyses
- takes many forms

D&R is a parallelization of the analysis.

D&R

Subsets for a large, complex dataset typically defined so that there are between-subset variables (BSVs), one value per subset

Some of these variables can define the division

There are also within-subset variables (WSVs) that have many values per subset

Method of an analysis is applied to the WSVs of each subset creating output for each subset

Recombination for the method includes an analysis of how the outputs depend on the BSVs

The subset analysis is parallelized, but the computations on the subsets “communicate” through further data analysis of the results

D&R

There are typically a number of divisions for a large, complex dataset

Sometimes use “naive division”:

- divide the data without a BSV
- e.g., for n observations and p variables form k subsets, each with n/k observations

GBLX VoIP Dataset

B. Xi and H. Chen and W. S. Cleveland and T. Telkamp (2010) Statistical Analysis and Modeling of Internet VoIP Traffic for Network Engineering, **Electronic Journal of Statistics, 58–116**

VoIP packet traffic traffic from the Global Crossing (GBLX) international network

Monitor in Newark NJ on link between IP-PSTN Gateway and IP network edge router

- packet arrival timestamps
- 48 hr of packet timestamps & headers in both directions of 138,770 calls
- 27 sending sites: Newark gateway and 26 other gateways around the world
- 277,540 semi-calls
- 1.315 billion packets
- each header has 6 fields
- derive many other variables from the 6 fields, some with about 1.3 billion values

Purpose of analysis

- develop statistical model for traffic generation in one direction of a link
- simulation studies for traffic engineering and VoIP engineering factors

GBLX VoIP: One Division

277,540 semi-calls are subsets

WSVs

- packet arrival times a_k
- lengths of alternating transmission lengths and silence lengths
- packet jitter $a_k - a_{k-1} - 20$ ms within each transmission interval

BSVs

- sending gateway
- call direction: caller-to-callee or callee-to-caller
- number of transmission intervals
- duration
- start-time

Example: studied alternating transmission lengths and silence lengths and how statistical properties changed with the BSVs: duration, start-time, call direction, and sending gateway

GBLX VoIP: Another Division

Subsets are 14 million transmission intervals

2 WSVs: jitter values and traffic rate on link for each value

1 BSVs: sending site and link-direction

for each sending site, “near-replicate” subsets of WSVs of size 30000, “guided naive subset”

Altogether about 980,000 subsets

Studied how jitter changes with traffic rate and sending site

D&R for Visualization Methods: Subset Sampling

In most of analyses, numeric methods can be applied to all subsets

Apply a visualization method such as normal quantile plot to each subset in a sample of subsets

Cannot typically apply to all subsets: too many to view

Sample of subsets can be chosen to be representative using sampling frame based on BSVs

Sample of subsets can be chosen to have unusual values of BSVs variables

What D&R Is Not

Best not to think of it as MapReduce

The D and the R go way beyond the traditional conceptual framework of MapReduce (i.e., the way it has been used)

D&R can exploit MapReduce computational ideas (e.g., Hadoop)

RHIPE: R and Hadoop Integrated Programming Environment

Saptarshi Guha: a merger of R and the Hadoop distributed computing environment

Makes D&R easy to carry out

Can run any R code/package on a large, complex database

Open source: www.stat.purdue.edu/ sguha/rhipe

The data analyst works wholly within R to specify the subsets and the computation to be done on each

Hadoop distributes k copies of subsets around a cluster, and attempts to keep the computation for subset on a node close to where it is stored

RHIPE R commands provide an easy interface between Hadoop and R

RHIPE has vastly increased our ability achieve comprehensive, detailed analysis of our large complex datasets

D&R and Dynamic Languages

D&R

Research needed in division methods and recombination methods

Great success for some datasets so far

More research is needed to see how widely applicable it is

Dynamic Languages

Research to foster computation for large, complex datasets