Parallel Algorithms in Data Mining
The ANU CSL Data Mining Approach

Peter Christen*
Computer Sciences Laboratory, RSISE
Australian National University

peter.christen@anu.edu.au
http://csl.anu.edu.au/~peter

18 July 2000

* Funded by grants from the Swiss National Science Foundation and Novartis Stiftung

Outline

• What is Data Mining?

• Why Parallel Data Mining?

• The ANU CSL Data Mining Approach:
  – Predictive Modelling
  – Parallel Techniques
  – Some Results
  – Data Mining Projects / Consultancies
Data Mining – Data Collections

- Large amounts of data are being collected in business and science
- Data sets are in the range of Terabyte, with first Petabyte collections in science
- Examples: Human Genome project, retail sales transactions, insurance and financial data, health data (Medicare), etc.
- Complexity of data sets (number of attributes) is growing
- Many organisations are *data rich but information poor*

Data Mining – Tasks

- Detect hidden patterns and relationships
- Extract useful, previously unknown information
- Find outliers (e.g. fraudulent behaviour)

- Techniques used:
Data Mining – Challenges

- Data Mining Algorithms/Applications have to deal with:
  - Scalability (size of data collections doubles every 18 month)
  - Dimensionality (curse of dimensionality)

- Parallel Data Mining:
  - Increased I/O, memory and computing power
  - Allows to attack bigger problems
  - Gives shorter response times

Curse of Dimensionality

Hundred closest neighbours in a two- and hundred-dimensional space
The ANU CSL Data Mining Approach

- Development of algorithms that are scalable both with the data size and the number of processes

- Using a finite element approach for predictive modelling and high-dimensional surface fitting

- Apply techniques like finite elements, thin plate splines, wavelets and additive models

Group members: M. Hegland, O. Nielsen, P. Christen, S. Roberts, T. Semenova

Collaborators: P. Strazdins, A. Torda, P. Hall, M. Kahn, I. Altas (CSU), K. Burrage (UQ) and others

G. Williams, S. Hawkins, R. Baxter, M. Fett (CSIRO CMIS Enterprise Data Mining group)
Predictive Modelling

- A predictive model in some way describes the typical behaviour of a data set by a function \( y = f(x_1, \ldots, x_d) \)
- Predictor variables \( x_i \) (attributes), response variable \( y \) (to be predicted)
  Currently: Continuous and categorical variables
- Data collection is split into training and test set

<table>
<thead>
<tr>
<th>Postcode</th>
<th>Gender</th>
<th>Age</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>2602</td>
<td>M</td>
<td>24</td>
<td>32,570</td>
</tr>
<tr>
<td>2608</td>
<td>M</td>
<td>42</td>
<td>56,420</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Example:

Our Three Methods

1. TPSFEM: Thin plate splines finite element method
   Piecewise multilinear finite elements, most accurate approximation at highest computational costs, suitable for two and three dimensional problems

2. HISURF: High dimensional surface smoothing
   Uses a hierarchical interpolatory wavelet basis, suitable for three to seven dimensional problems

3. ADDFIT: Additive models
   Lowest computational costs, but coarsest approximations, suitable for high-dimensional problems
Example: Modelling aeromagnetic data

TPSFEM: 16641 vars   HISURF: 833 vars   ADDFIT: 388 vars

Steps of our Data Mining Algorithms

Raw Data → Preprocess, Normalise → Binary Data

TPSFEM / HISURF / ADDFIT

Solve Linear System → Linear System → Assemble Linear System

Model y=f(x) → Visualize, Postprocess → Graphs, Reports
Advantages

- All three methods have two steps:
  1. Read data set and assemble a linear system
  2. Add constraints and solve linear system
- The first step is easy and efficient to parallelise
- Data set has to be read once only from secondary storage
- Size of the linear system is independent of the number of data records

Step 1: Read Data and Assemble Linear System

- Each data record adds some values into the final linear system
- Assembly of a data record is independent of all others
- Structure of the linear system depends on the method
- Dimension of the linear system is independent of the number of data records, but depends on the method and resolution of the model (e.g. grid points in each dimension)
Elementary Matrix Structure

Three dimensional data set, model resolution 9 grid points

Final Matrix Structure

Storage requirements
Parallel Architectures

[Diagram showing parallel architectures: Shared memory vs. Distributed memory]

- Sun Enterprise SMP Server
- Beowulf Linux Cluster

Parallel Assembly Step

- Each process reads a fraction $n/p$ of the whole data set and assembles a local linear system
- The complete matrix data structure is needed on each process
- Local linear systems are collected and summed after assembly
- Implemented both in Master-Worker and SPMD style (using MPI message passing)

SPMD: Single Program, Multiple Data
Master-Worker Assembly

- Master sends task-message to worker (containing filename, start position and number of records to read and assemble)
- Workers open file, read data and assemble into local linear systems
- Repeated until all data records are assembled
- Advantage: Automatic load balancing (faster processes assemble more records)
- Drawback: Small messages have to be communicated

SPMD Assembly

- Each process computes the part it has to assemble locally
- It then reads and assembles this part independently
- Advantage: No communication needed
- Drawback: Load imbalance may occur

SPMD: Single Program Multiple Data
Step 2: Add Constraints and Solve Linear System

- Constraints matrix has same dimension as data matrix, but the assembly is very fast because no data is involved
- Linear system can be solved sequential or parallel (depending on its size \( m \) and the hardware platform available)
- For parallel run: Locally assembled linear systems have to be collected and summed prior to solving (reduce operation)
- Work on parallel solver: Peter Strazdins

Parallel Assembly on Sun Enterprise 4500

CSIRO DM server, shared memory, 12 UltraSPARC 400MHz processors, 6,912 MB memory and 250 GB disk
**Parallel Assembly on Beowulf**

ANU DCS *Bunyip*, distributed memory, 96 Dual Intel P-III 550MHz processors, 36,864 MB memory, 1,306 GB disk

**Parallel Assembly on Beowulf: Speedup and Efficiency**

ANU DCS Seminar, 18 July 2000
ANU CSL Data Mining Projects & Consultancies

- Analysis of health care data collections
  Collaboration with CSIRO Enterprise Data Mining group

- SAUSAGE (Secondary protein structure)
  Collaboration with RSC Biomolecular Simulation/Calculation group

- Old projects:
  - MACHO project (Mount Stromlo star data base)
  - NRMA (fraud detection)
  - Australian Taxation Office

Conclusions & Outlook

- Data Mining helps to analyse and understand large data collections

- The ANU CSL Data Mining group develops scalable parallel algorithms for predictive modelling

- Future work:
  - Application and evaluation of our techniques within real world data mining projects
  - Extend our techniques for more complex attributes (e.g. sets and vectors)