Parallel Implementation of Mining Highly Interacted Attribute Pairs

Jianmei Yang
Dec 9th, 2010
For two attribute variables $X_1$ and $X_2$ and a class variable $Y$, when relationship between $X_1$ and $Y$ depends on $X_2$, $X_1$ and $X_2$ are said to be *interact*.

Interactions are outcomes that occur when all the variables are observed together

- Interaction between two variables exists when the joint effect of both is different from that obtained by additively combining the individual effects.

Different interactions: independence, synergy, redundancy.
Interaction Mining using Information Theory

- Let \( \omega \) denote the set of all random variables:
  \[
  \omega = \{ X_1; X_2; \ldots X_i; \ldots; X_N \}.
  \]
  \( X_i \): A random variable representing an attribute or class label

- Entropy
  \[
  H(X_i) = -\sum_x p(X_i = x) \log_2 (p(X_i = x))
  \]

- KWII: Amount of information present in a set of variables, which is not present in any subset of the variables.
  - For set of variables \( S = \{ X_1; X_2; \ldots, X_K \} \)
    \[
    KWII(S) \equiv -\sum_{T \subseteq S} (-1)^{|S \setminus T|} H(T)
    \]
    - e.g. KWII \((A;B;C) = - H(A) - H(B) - H(C) + H(AB) + H(AC) + H(BC) - H(ABC)\)
Experiment Setting

- **Input:** Data set of $n$ attribute variables and class variable, number of sample is $m$
- **Computation:** Compute the KWII values for all possible attribute pairs
  - for $N$ attributes, # of attribute pairs will be $n^*(n-1)/2$
- **Output:** Attribute pairs with highest KWII value, which is the most significant interacted pairs
- **Sequential running time:** $O(n^2m)$
  - Can be very time consuming when $n$ is large
  - Turn to parallel solution!
Parallel Implementation

- **Dataset**

  - Attribute Variables (N-dimensional)
  - Class Variable

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

  - Total number of attribute pairs: \( \frac{N(N-1)}{2} \)

- **PE\(_0\)**
- **PE\(_1\)**
- **PE\(_p\)**

  - Read Data
  - Compute the Kwii of attribute pairs assigned to each processor
  - Get local highly interacted pair for each processor
  - Get global highly interacted pair

---

*University at Buffalo*
*The State University of New York*
The computation of KWII for all attribute pairs is evenly distributed across all the processors.

```
int pairs_per_node=(attr_num)*(attr_num-1)/(2*size) +1;
    ...... ...... 
for(int attr1=0;attr1<attr_num;attr1++)
{
    for(int attr2=attr1+1;attr2<attr_num;attr2++)
    {
        count_current=(2*attr_num-attr1)*attr1/2+attr2-attr1;
        //decide whether the KWII computation of current pair is assigned to this node or not
        if( count_current>= (pairs_per_node*rank +1)  &&  count_current<= (pairs_per_node*(rank +1)))
        {
            printf("attr1 is: %d, attr2 is: %d, count_current is: %d, rank is: %d \n",attr1,attr2,count_current,rank);
            kwii.kwii(D,sample_num,v);
            ...... 
        }
    }
}
```
Each processor picks up the attribute pair with the local highest KWII values and send it to \( P_0 \)

- Define a derived data types `Result` using triplet of (int, int, double) to store the results of attribute pair and KWII values.

```c
MPI_Datatype myresult,old_types[2]={MPI_INT,MPI_DOUBLE};
MPI_Aint indices[2];
int blocklens[2]={2,1};
MPI_Address(&r,&indices[0]);
MPI_Address(&r.kwii,&indices[1]);
indices[1] -= indices[0];indices[0]=0;
MPI_Type_struct(2,blocklens,indices,old_types,&myresult);
MPI_Type_commit(&myresult);

P_0 \text{ receives the Result from all other processors and picks up the one with the highest KWII value as the global highly interacted attribute pair.}
Parallel Running Time

# of attributes = 5000, # of samples = 1000
Parallel Speedup

# of attributes = 5000, # of samples = 1000
Parallel Efficiency

E(p) = \frac{T(S)}{(p \times T(P))}

# of attributes = 5000, # of samples = 1000
Parallel Speedup VS Dataset Size

![Graph showing parallel speedup vs dataset size with lines for different dataset sizes numbered 1000*5000, 1000*500, and 100*500. Each line represents a different dataset size, with the number of processors on the x-axis and parallel speedup on the y-axis. The graph shows an increasing trend in speedup as the number of processors increases.]
Running Time VS Dataset Size

# of nodes = 128
# of samples = 1000

# of nodes = 128
# of attributes = 1000
Thank you!