Clustering a Large Number of Compounds. 2. Using the Connection Machine

ROBERT WHALEY† and LOUIS HODES*,‡

Thinking Machines Corporation, Cambridge, Massachusetts 02142, and National Cancer Institute, National Institutes of Health, Bethesda, Maryland 20892

Received December 18, 1990

About 230 000 compounds in the National Cancer Institute Repository are available for screening under a new protocol. This paper is the second on an project to extract a representative sample of these compounds by clustering. The clustering program was implemented on the Connection Machine, a massively parallel computer with 16K processing elements. This implementation reduced a formidable task to a relatively routine run.

INTRODUCTION

Clustering of over 230 000 compounds was undertaken to obtain a representative diverse sample of the National Cancer Institute's file. The resulting subset is to be used as a resource for random testing in the NCI cell culture screens.

The first paper[1] described the variable size fragment features that were used for matching compounds. A rationale was put forth for the use of the simple and fast leader algorithm for clustering. An effort was made to optimize cluster parameters and fragment weights for this application by results on an initial sample of the file.

Since clustering time can increase as the square of the data, it became clear that the full clustering could not conveniently or economically be done at the NIH IBM 3090 facility. This impasse was solved by the availability of the massively parallel Connection Machine at the Naval Research Laboratory.

Here we briefly review the method and how it fits the parallel operation of the Connection Machine.

REVIEW OF THE METHOD

Each compound is replaced by a set of distinct molecular fragments. Each fragment has a weight determined by its size, its frequency in the entire collection, and its multiplicity in the compound. Thus, each compound has a total weight, which is the sum of its fragment weights. For optimal operation of the algorithm, which enters compounds in sequence, the compounds are sorted in the order of increasing total weight.

Clustering is performed as follows by the leader algorithm. The clusters are represented by an initially empty list of leaders. Each entering compound is compared with all of the leaders. All matches are recorded by outputting the id's of the matching pairs. Multiple matches result in the compound belonging to more than one cluster. Only if there are no matches is the entering compound added to the list of leaders.

The comparison of an entering compound with a leader begins by determining the fragments in common. For these fragments we select the lower of the two weights. Notice that a given fragment may have a higher weight in another compound in which it occurs more often. These selected weights are then summed over the common fragments. If the sum exceeds the preset fraction of the total weight of the entering compound then the comparison yields a match. Notice that the total weight of the entering compound is at least as large as the total weight of any compound on the leader list because of the order of the entering compounds.

To be more precise, we can supply the dimensions of the data elements. The molecular fragments are converted to 32-bit hash codes at the time they are first generated. Their weights require 7 bits. Each compound has a 20-bit identifying NSC number and a total weight that is the sum of the fragment weights. The average number of fragments per compound is about 16, but it varies greatly and can reach over 100. The total weight requires 11 bits.

USE OF THE CONNECTION MACHINE

Here we describe some of the features of the Connection Machine[2] that are important for this application. The "small" machine we used contained 16K processors, each with its own 8K bytes of random access memory, for a total of 128 Mbytes. (The memory on this machine has since been upgraded for a total of 2 Gbytes of memory. A large Connection Machine could have up to 64K processors, each with 128 Kbytes of memory, for a total of 8 Gbytes.) There is also a front-end Sun computer which controls the Connection Machine.

In operation, the Sun computer broadcasts instructions and data to enabled processors of the Connection Machine. Processors are enabled with a context bit contained within each processor. The Sun computer can also read and write the memory of individually addressed processors. Besides the context bit, each processor has a test bit that is set upon the success of a test instruction (e.g., the comparison of two numbers).

To facilitate programming, each processor in the Connection Machine can simulate multiple virtual processors.[2] The ratio of virtual processors to physical processors is known as the virtual processor ratio. When using virtual processors, the memory of each physical processor is divided equally among its virtual processors.

The Connection Machine software provides an extensive repertoire of instructions that operate on local data within each enabled processor. In addition, there are also global instructions (e.g., the logical 'or' of a designated variable over every processor with the context bit set) and communications instructions such as 'scans'.[3] The scan operation takes a binary associative operator, such as addition, and an ordered set of elements [a0, a1, a2, ...] and returns the ordered set [a0, a0+a1, a0+a1+a2, ...]. In our clustering application, scan was used to sum a variable over a specified range of virtural processors. It operates simultaneously on all segments of processors within designated segment boundaries.

THE DATA PARALLEL LEADER ALGORITHM

The leader algorithm was implemented as a data parallel algorithm[4] for the Connection Machine. Data parallel algorithms generally assign one processing element to each data element.

One approach would be to store one leader compound in each processor using an array to store the leader's fragments. A simpler, more memory efficient approach is to store a single leader fragment together with its weight and associated variables in each virtual processor. A leader compound's frag-

† Thinking Machines Corp.
‡ National Institutes of Health.

0095-2338/91/1631-0345$02.50/0

Virtual Processor Number	NSC Number 20 bits	Total Weight 11 bits	Fragment Hash Id 32 bits	Fragment Weight 7 bits	Start 1 bit	End 1bit	Active 1 bit	Temporary Space 33 bits
...								
i	5707	11	335407	3	Yes	No	Yes	
i + 1	5707	11	165271	4	No	No	Yes	
i + 2	5707	11	142741	1	No	No	Yes	
i + 3	5707	11	67321	1	No	No	Yes	
i + 4	5707	11	38992	2	No	Yes	Yes	
i + 5	203598	12	525099	2	Yes	No	Yes	
i + 6	203598	12	471055	6	No	No	Yes	
i + 7	203598	12	207181	2	No	No	Yes	
i + 8	203598	12	97380	2	No	Yes	Yes	
i + 9	187661	12	34606	12	Yes	Yes	Yes	
i + 10							No	
i + 11							No	
...								

Figure 1. Memory layout for leader compounds 5707, 203598, and 187661.

```
for each compound N

  compare compound with all leaders

  if any matches are found

    for all leaders J matching compound N

      output match N,J

    endfor

  else


    add compound N to leader list

  endif

endfor
```

Figure 2. Parallel clustering algorithm.

ments are stored sequentially in adjacent virtual processors on the Connection Machine (Figure 1).

The parallel algorithm (Figure 2) is essentially the same as the serial algorithm (Figure 3) presented in ref 1, except the inner loop is removed and performed in parallel. The entering compounds are stored on the front-end Sun Computer and broadcast to the Connection Machine serially. As an entering compound is broadcast, it is compared with the leaders stored on the Connection Machine. If the compound does not match at least one leader, it is added to the list of leaders in the Connection Machine's memory. If the entering compound does match one or more leaders, the Sun computer records each of the matches.

The heart of the algorithm is the similarity measure[1] used to compare the entering compounds with the leader compounds. The similarity measure used is

$$\text{sum min}[c(i), l(i)]/\text{sum } c(i)$$

where $c(i)$ is the weight of the ith fragment of the compound, and $l(i)$ is the weight of the corresponding fragment of a leader. Sum $c(i)$ is the total weight of the entering compound.

The similarity measure is implemented on the Connection Machine in the following way. The current entering compound's fragments are broadcast one at a time. Each active virtual processor already contains one fragment from a leader. The leader fragment is compared with the broadcast fragment, and if they correspond, the entering compound's fragment weight is stored in the virtual processor. This is performed

```
for each compound N

  set flag F

  for each leader J

    compare compound N with leader J

    if compound N matches leader J

      output match N,J

      clear flag F

    endif

  endfor

  if flag F

    add compound N to leader list

  endif

endfor
```

Figure 3. Serial clustering algorithm.

simultaneously in all virtual processors.

Once all of the fragments for the compound have been broadcast, the minimum of the fragment weights is computed. The minimum value is 0 for all leader fragments with no equivalent entering fragments. Then the scan operation is used to sum the minimum over the set of virtual processors corresponding to each leader. Finally, this sum is compared with the entering compound's total weight to produce the similarity measure. The entire computation is performed simultaneously for each leader.

The similarity measure is compared to the preset fraction (in our case 0.65) to test for matches. The test bit is recorded so that a global 'or' detects matches for output, or if none, the program adds the entering compound as a new leader.

MANAGEMENT OF VPS

The program begins with 16K virtual processors (a virtual processor ratio of 1 on our 16K Connection Machine). Each time the virtual processors become filled with leader fragments, the virtual processor ratio is doubled to provide more processors for leader fragments.

With approximately 100 000 leaders, averaging 16 fragments each, the final number of virtual processors allocated is 2M (a virtual processor ratio of 128). A total of 1 754 404 of these processors were actually used. At a virtual processor ratio of 128, there were slightly less than 64 bytes of available memory per virtual processor, quite adequate for the required variables.

This strategy reduces the run time to 1/6–2/3 the time of the alternative of starting with the largest virtual processor ratio required. In addition, in cases where the approximate number of leaders is not known in advance, this is a more robust strategy, efficiently handling both cases with unusually high and unusually low proportions of leaders.

To reduce run time even further, leaders that are too small to match the current entering compound can be eliminated, and their processors reused. This is possible because the entering compounds are presorted by increasing total weight. Any leader less than 65% of the total weight of the entering compound cannot match the entering compound or any of the subsequent entering compounds.

The program eliminates unmatchable leaders each time the free virtual processors are exhausted. If enough processors are reclaimed, then processing continues without increasing the number of virtual processors. Otherwise, the number of virtual processors is increased as described above.

This final optimization of the program used only 1M virtual processors, and run time was reduced by more than a factor of 2 on our data.

THE APPLICATION

This work was performed using 230 092 compounds from the National Cancer Institute Development Therapeutics Program Repository. The 116 706 resulting cluster leaders comprise a resource of diverse compounds for testing in a new screen that is now in operation. Such a large set would be used in conjunction with a program to test for activity and novelty in the screen. This latter program can be based on sufficient early testing in the screen as described in ref 5. If desired, other members of selected clusters can be retrieved.

The estimated CPU time for this job on an IBM 3090 mainframe was 20–34 h. The total run time (wall clock time) was 2 h, 35 min on the 16K Connection Machine. If a larger Connection Machine with more processors had been used, the time would have been reduced to about 1 h, 18 min (for a 32K machine) or 39 min (for a 64K machine).

The use of cluster leaders as input to a prioritizing program circumvents the bunches of similar compounds that would effectively prevent the collection of a diverse subset. Moreover, the set of cluster leaders can be used repeatedly for different biological tasks.

It is now almost feasible to do a comprehensive literature surveillance by clustering the Chemical Abstracts Service 10 million compound file. This job is estimated to take about 800 h on the large Connection Machine. Fortunately, like the application reported here, it would only need to be done once.

ACKNOWLEDGMENT

Connection Machine is a registered trademark of Thinking Machines Corporation. Sun is a registered trademark of Sun Microsystems, Inc. IBM is a registered trademark of International Business Machines Corporation. We thank Henry Dardy for providing access to the Connection Machines at the Naval Research Laboratory.

REFERENCES AND NOTES

(1) Hodes, L. Clustering a Large Number of Compounds. 1. Establishing the Method on an Initial Sample. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 66–71.
(2) Hillis, W. Daniel. *The Connection Machine*. MIT Press: Cambridge, MA, 1985.
(3) Belloch, G. E. Scans as Primitive Parallel Operations. *Proceedings of the International Conference on Parallel Processing*; 1987.
(4) Hillis, W. Daniel; Steele, Guy L., Jr. Data Parallel Algorithms. *Commun. ACM* **1986**, *29* (12), 1170–1183.
(5) Hodes, L. Computer Aided Selection for Large Scale Screening. In *Comprehensive Medicinal Chemistry, Vol. 1. General Principles*; Hansch, Sammes, Taylor, Eds.; Pergamon Press: New York, 1990; Chapter 3.3; p 279.

Clustering a Large Number of Compounds. 3. The Limits of Classification

LOUIS HODES* and ALFRED FELDMAN
National Cancer Institute, National Institutes of Health, Bethesda, Maryland 20892

Received December 18, 1990

Clustering is normally used to group items that are similar. In this application of obtaining a diverse sample from the 230 000 compounds in the National Cancer Institute Repository, we cluster to select compounds that are different from the rest, to optimize screening for new leads. With these constraints, our approach yielded many singleton clusters. We can interpret these results as evidence for a limit to classification, contrary to the customary view of chemistry as a study of classes of compounds.

INTRODUCTION

The first paper[1] announced the intention of clustering a set of over 230 000 diverse compounds. The objective was to extract a representative sample of compounds from the National Cancer Institute file for testing in the new NCI primary screen.[2] That objective was accomplished with the aid of a Connection Machine.[3]

This paper makes some general observations on the classification of chemical structures, based primarily on the results of the large clustering. Our results show that a diverse set of compounds will exhibit a dual nature, some clusters and some scattered.

We also showed that the systematic clustering method can provide examples for an attempt at classifying the file. In the reported case, less than half the file was successfully classified in this way; the remainder being probably too diverse to classify.

CLASSIFICATION AND CLUSTERING

Classification of compounds is a form of clustering. By performing reasonable clustering on sets of diverse compounds we have obtained evidence with respect to the validity of classification. One can imagine that, if compounds fall into natural classes, then this phenomenon would show up upon clustering a large set of compounds, even if they are diverse. Instead, we get a persistent occurrence of singletons in addition to the expected increase in large clusters.

These singletons are important for testing novel compounds in our screening program. In contrast, Willett[4] avoids singletons by assigning them each to its closest cluster. Lawson and Jurs[5] achieve a similar effect by their choice of clustering method.

This forcible treatment of singletons is an example of an implicit belief that compounds ought to belong to classes. Clustering is a means for classifying, and classifying has worked well in many areas of chemistry.

Perhaps the most fundamental achievement in chemistry was the classification of elements according to the periodic table. The study of organic chemistry under the traditional, or nonsystematic, nomenclature is heavily linked to classes. At the macro level, there is the classification of biological species. Back in chemistry, the classification of natural products derives from that of species.