A Probabilistic Algorithm for Molecular Clustering

Frédéric Cazals
Algorithm Project, Inria

December 15, 1997

[summary by Bruno Salvy]

In order to design a drug curing a given pathology, an approach which is commonly used consists in first selecting those that work best among molecules known to treat similar symptoms. In view of the number of known molecules, an exhaustive approach is often impossible. Instead, it is a common strategy to pick at random a number of molecules in a large database; and then concentrate on those that are chemically close to the ones that performed well. It is therefore important to be able to find those molecules in such a database. The aim of this work is to present a probabilistic algorithm for this task.

1. Molecules and similarity

The database is represented as an array of $n \simeq 10,000$ molecules, each molecule being characterized by the presence or absence of $d \simeq 1,500$ molecular fragments. Molecules are close when they differ by few molecular fragments. More precisely, one defines the *size* $s(m)$ of a molecule m as the number of its fragments and the *similarity* $\mathrm{sim}(m, M)$ between two molecules as the number of common fragments. Finally, two molecules m and M are called (α, β)-similar for $\alpha \in [0, 1]$ and $\beta \geq 1$ when

$$\mathrm{sim}(m, M) \geq \alpha \min(s(m), s(M)), \qquad \max(s(m), s(M)) \leq \beta \min(s(m), s(M)). \tag{1}$$

Note that this is not an equivalence relation. Other measures of similarity might also be of interest in practice.

2. Algorithm

The aim of this work is to find efficiently as many (α, β)-similar pairs in the database as possible. Obviously, an exhaustive search in $n(n-1)/2$ operations is possible, but expensive. The idea instead is to use a divide-and-conquer partitioning process. A fragment is selected at random and the database is partitioned into two subsets according to the presence or absence of this fragment. When such a subset has less than a fixed number K of elements an exhaustive search is performed; otherwise, the same process is applied recursively.

This technique finds a proportion τN of the number N of (α, β)-similar pairs. Heuristically, repeating the same process from scratch yields $\tau(1-\tau)^{i-1}N$ new pairs at the ith iteration. Thus a few iterations of this idea yield a very large proportion of N.

3. Implementation

The parameter K plays an important part in the efficiency of the algorithm. When K is small the search is faster but finds less pairs, so that the number of times it has to be repeated to obtain

the same number of pairs can be larger than for higher values of K. The optimal value of K also depends on the efficiency of the different stages of the algorithm. Any improvement on the partitioning and exhaustive search part shift the optimum to larger values of K, while a good data-structure for checking whether a pair has already been found shifts it in the other direction. Here are implementation ideas that lead to an efficient program:

- The database is stored as an array of bits;
- the entries in the database are accessed by chunks (bytes or words), a constant array making it fast to count the number of bits equal to 1 in a chunk;
- computing the similarity between to molecules is then performed by bitwise and;
- the sizes of the molecules are computed once at the beginning;
- the partitioning is done like in Quicksort, on an array of pointers to the molecules;
- the set of pairs is stored as an array of binary search trees (a hash table would also do).

In practice, with this implementation and K around 150, then 4 or 5 runs of the partitioning yield more than 90% of the pairs in a matter of minutes. The exhaustive search would take several hours.

Conclusion

It would be nice to find the optimal value of K by a complexity analysis. However, the Bernoulli distribution for the bits in the database does not give a good model. It is necessary to take into account the fact that the database was arrived at by a historical process where many of the molecules are variants of each other.

Bibliography

[1] Cazals (Frédéric). – Effective nearest neighbours searching on the hyper-cube, with applications to molecular clustering. In *ACM Symposium on Computational Geometry*. pp. 222–230. – ACM Press, 1998. Also available at the url `http://www.inria.fr/prisme/personnel/cazals/xfc_research.html`.