Reconstructing Set Partitions

Vladimir Grebinski and Gregory Kucherov *

1 Problem Formulation and Motivation

We study the following combinatorial search problem: Reconstruct an unknown partition of the set $[n] = \{1, \ldots, n\}$ into at most K disjoint non-empty subsets (classes) by making queries about subsets $Q_i \subseteq [n]$ such that the query returns the number of classes represented in Q_i. The goal is to reconstruct the whole partition with as few queries as possible. We also consider a variant of the problem where a representative of each class should be found without necessarily reconstructing the whole partition.

Besides its theoretical interest, the problem has practical applications. Paper [2] considers a physical mapping method based on hybridizing probes to chromosomes using the FISH technology. Under some reasonable assumptions, the method amounts to finding a partition of probes determined by the chromosome they hybridize to, and actually corresponds to the above formalization.

2 Reconstructing the Whole Partition

We first estimate the lower bound. Any algorithm must, in particular, reconstruct partitions of $[n]$ into two subsets. There are 2^{n-1} such partitions, and each query has two potential answers. By the information-theoretic argument, any algorithm requires in the worst case at least $\log_2 2^{n-1} = n - 1$ queries.

We now present a divide-and-conquer algorithm solving the problem in $\mathcal{O}(n \log\log K)$ queries. The algorithm proceeds as follows.

1. Split the set $[n]$ into two disjoint subsets of $n/2$ elements.
2. Reconstruct recursively the induced partition for each of the two subsets.
3. Choose one representative of each equivalence class in each subset.
4. Identify the pairs of representatives of the same class.

Clearly, the algorithm allows to reconstruct the whole structure. The key step is step 4. We show that it can be done in at most $2K$ non-adaptive queries using the following theorem which is a particular case of a more general result from [1]. Consider a bipartite graph $G = (V, W, E)$, $V \cap W = \emptyset$, $E \subseteq V \times W$, and assume that the degree of each vertex of V is 0 or 1 (there is no restriction on W). Assume the graph is unknown to us and we want to reconstruct it by making queries of the following kind: For two subsets $V' \subseteq V$ and $W' \subseteq W$, what is the number of edges connecting V' and W'?

THEOREM 2.1. *Assume that $|V| = n$, $|W| = m$. Then G can be reconstructed in $2n\frac{\log m}{\log n}$ non-adaptive queries asymptotically.*

At step 4, two sets of representatives, say V and W, can be thought of as two sides of a bipartite graph. Two vertices are connected by an edge iff they represent the same equivalence class. Since each set consists of representatives of distinct equivalence classes, this is indeed a bipartite graph. By the same argument, the degree of each vertex on each side is 0 or 1. We want to apply Theorem 2.1 and for that we must be able to answer queries of the type "How many edges are there between two subsets $V' \subseteq V$ and $W' \subseteq W$?".[1] Denote this number by $e(V', W')$. According to our initial problem, querying $V' \cup W'$ yields the number of distinct equivalence classes in which the elements of $V' \cup W'$ occur. Denote this number by $\mu(V' \cup W')$. It is easily seen that $e(V', W') = |V'| + |W'| - \mu(V' \cup W')$, and we can simulate a query e by one query μ. By Theorem 2.1, step 4 can be done in $2K$ non-adaptive queries, since the size of each set of representatives is bounded by K.

The overall complexity follows from the recurrence $T(n) = 2T(n/2) + 2K$ which has the solution $T(n) = \alpha n - 2K$ (K is assumed constant). That is, the algorithm above is linear on n.

Unfortunately, the coefficient α itself depends on K and is of order $\log K$. To see this, consider the behaviour of the algorithm for $n < K$. Here we have the recurrence $T'(n) = 2T'(\frac{n}{2}) + 2\frac{n}{2}$, and the solution $T'(n) = O(n \log n)$. Therefore, $T(K) = O(K \log K)$, and we deduce that $\alpha = O(\log K)$.

However, it is possible to reduce α to $\log\log K$. For $n < K$, we modify the algorithm as follows.

*INRIA-Lorraine/LORIA, 615, rue du Jardin Botanique, BP 101, 54602 Villers-lès-Nancy, France, e-mail: {grebinsk,kucherov}@loria.fr

[1] Note that Theorem 2.1 applies to a larger class of graphs, as here the degree of each vertex *on both sides* is bounded by 1.

Instead of splitting the set into two subsets at step 1, we split it into $\sqrt{n}$ subsets $A_1, \ldots, A_l$ ($l \approx \sqrt{n}$) of approximately $\sqrt{n}$ objects each. After the induced partitions of $A_1, \ldots, A_l$ are reconstructed recursively, we choose a set of representatives R_i in each, and then merge them successively to obtain the partition of the set. To analyze the complexity, assume $a_i = |R_i| \leq \sqrt{n}$ and $b_i = \sum_{j=1}^{i-1} a_i \leq i\sqrt{n} \leq n$. By Theorem 2.1, the complexity of merging A_i with the current partition can be estimated from above by $2\sum_{i=1}^{\sqrt{n}} \log b_i \cdot \frac{a_i}{\log a_i} \leq 2\sum_{i=1}^{\sqrt{n}} \log n \frac{\sqrt{n}}{\log \sqrt{n}} \leq 4n$.

For the whole algorithm, we then get the recurrence $T'(n) = \sqrt{n} \cdot T'(\sqrt{n}) + 4n$ which gives the solution $T'(n) = O(n \log\log n)$. We then obtain $T(K) = O(K \log\log K)$, and $\alpha = O(\log\log K)$. This also shows that in case no information on the number of classes is available beforehand, $O(n \log\log n)$ queries suffice.

THEOREM 2.2. *A partition of $[n]$ into at most K subsets can be reconstructed in $O(n \log\log K)$ queries. In the general case when K is unknown, the partition can be reconstructed in $O(n \log\log n)$ queries.*

3 Finding a Representative of Each Class

Assume we need to find one representative of each class without necessarily reconstructing the entire partition. Compute first the information-theoretic lower bound. Assume we have an algorithm solving the problem. What is the set of different possible answers? We claim that *every* subset of $(K-1)$ elements occurs in some answer. It is seen by considering the partition under which the chosen $(K-1)$ elements form singleton classes and all the other $(n-K+1)$ objects form the K-th class. Since a K-set contains K distinct $(K-1)$-subsets, we can estimate from below the number of different answers by $\frac{1}{K}\binom{n}{K-1}$. Since each query can yield at most K answers, any algorithm makes at least $\log_K \left(\frac{1}{K}\binom{n}{K-1}\right)$. For constant K, this gives $\Omega(K \frac{\log n}{\log K})$ queries.

On the other hand, a simple binary search algorithm allows to find one representative of each class in $O(K \cdot \log_2 n)$ queries. Assume that we have already found $r_1, \ldots, r_i$ that are representatives of i equivalence classes. To find a representative of an $(i+1)$-st class, we split the set of remaining $n-i$ objects into two sets of equal size, then test each of them together with $\{r_1, \ldots, r_i\}$ to find the set which has elements of classes other than those represented by $\{r_1, \ldots, r_i\}$, and then iterate the procedure. Clearly, $\log_2(n-i)$ queries are needed to find the $(i+1)-st$ representative, and $O(K \log n)$ overall queries solve the problem.

4 Discussion

The presented results leave two open questions: For the partition reconstruction problem, can a linear algorithm be found (with a multiplicative constant independent on K in case K is known)? For the representative problem, does there exist an algorithm that matches the lower bound $\Omega(K \frac{\log n}{\log K})$? We conjecture a positive answer to the first question.

Another direction for future work is to study the following question that corresponds to Job 2 in [2]: Given a set of representatives of each of K classes, how to reconstruct efficiently the whole partition.

It turns out that the partition reconstruction problem gives an example of combinatorial search problem for which non-adaptive algorithms are considerably less efficient than adaptive ones. This phenomenon is another subject to study.

References

[1] Vladimir Grebinski and Gregory Kucherov. Optimal reconstruction of graphs under the additive model. In R. Burkard and G. Woeginger, editors, *5th Annual European Symposium on Algorithms, Graz (Austria)*, volume 1284 of *Lecture Notes in Computer Science*, pages 246–258. Springer Verlag, 1997.

[2] Fengzhu Sun, Gary Benson, Norman Arnheim, and Michael S. Waterman. Pooling strategies for establishing physical genome maps using FISH. *Journal of Computational Molecular Biology*, 4(4):467–485, 1997.