Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 348 (2005) 240–250

Theoretical Computer Science

www.elsevier.com/locate/tcs

The minimum-entropy set cover problem

Eran Halperin[a,1], Richard M. Karp[b,*]

[a] *Computer Science Department, Princeton University, Princeton, NJ 08544, USA*
[b] *International Computer Science Institute, 1947 Center St., Berkeley, CA 94704, USA*

Abstract

We consider the minimum entropy principle for learning data generated by a random source and observed with random noise.

In our setting we have a sequence of observations of objects drawn uniformly at random from a population. Each object in the population belongs to one class. We perform an observation for each object which determines that it belongs to one of a given set of classes. Given these observations, we are interested in assigning the most likely class to each of the objects.

This scenario is a very natural one that appears in many real life situations. We show that under reasonable assumptions finding the most likely assignment is equivalent to the following variant of the set cover problem. Given a universe U and a collection $\mathcal{S} = (S_1, \ldots, S_t)$ of subsets of U, we wish to find an assignment $f : U \to \mathcal{S}$ such that $u \in f(u)$ and the entropy of the distribution defined by the values $|f^{-1}(S_i)|$ is minimized.

We show that this problem is NP-hard and that the greedy algorithm for set cover s with an *additive* constant error with respect to the optimal cover. This sheds a new light on the behavior of the greedy set cover algorithm. We further enhance the greedy algorithm and show that the problem admits a polynomial time approximation scheme (PTAS).

Finally, we demonstrate how this model and the greedy algorithm can be useful in real life scenarios, and in particular, in problems arising naturally in computational biology.

1. Introduction

The Shannon entropy function is a measure of the concentration of a distribution which plays an important role in various fields of computer science, such as coding theory, compression, learning, speech recognition and others. In many applications, one is given a data set that has been corrupted by noise and wishes to extract the true data. In this paper we use a minimum entropy principle to attack such problems.

Data classification is an important problem in learning theory. Given a data set generated by a random source, one would like to learn the distribution of the source. Often, the data are generated by the source and then passes through a noisy channel which adds ambiguity to the data. In such cases, one would like to learn both the distribution of the source and the origin of each of the data points, thus removing the noise effects.

* Corresponding author. Tel.: +1 510 666 2973; fax: +1 510 666 2956.
E-mail addresses: heran@cs.princeton.edu (E. Halperin), karp@icsi.berkeley.edu (R.M. Karp).

[1] Some of this work was done while the author was in UC Berkeley and ICSI, Berkeley, CA. The research was partly supported by NSF ITR Grant CCR-0121555.

0304-3975/$ - see front matter © 2005 Published by Elsevier B.V.
doi:10.1016/j.tcs.2005.09.015

We consider the following scenario for noisy data generated by a random source. We are given a sequence of observations of objects drawn uniformly at random from a population. Each member of the population has a type. For each object drawn from the population, we perform an observation which determines that the object's type is one of a given set of types. Given these observations, we are interested in assigning the most likely type to each of the objects.

These types might be code words in an erasure code, phonemes, letters of an alphabet, words in a limited lexicon, insurance risk categories, genomic haplotypes, alleles of a gene, different types of a disease such as leukemia, or any phenotype or trait, as long as each object has only one type. In the case of code words for example, the observation we perform on each object might be the output of an erasure channel.

We show that under some reasonable assumptions the most likely assignment is the one that minimizes the entropy of the distribution of the types. The problem of finding the most likely assignment via minimum entropy is of great practical importance. A number of approaches to this and related problems have been suggested, including the EM algorithm, Markov Chain Monte Carlo and convex optimization (see e.g. [9,17,16,20]), but we are not aware of any prior work on the computational complexity of solving the problem exactly or approximately.

The problem of finding the assignment which minimizes the entropy of the distribution of the types can be formulated as the following variant of the well-known minimum-cardinality set cover problem. We are given a universe U and a collection $\mathcal{S} = (S_1, S_2, \ldots, S_t)$ of subsets of U. A cover of U is a function $f : U \to \mathcal{S}$ such that $u \in f(u)$. The objective of the problem is to find a cover f which minimizes the entropy of the distribution $(|f^{-1}(S_1)|/|U|, |f^{-1}(S_2)|/|U|, \ldots, |f^{-1}(S_t)|/|U|)$. Similarly, the minimum-cardinality set cover problems aims in finding a cover which minimizes the non-zero entries in of the above vector.

The minimum-cardinality set cover problem is well studied, and it is well known that the greedy algorithm achieves a $\ln n$ approximation [1] and that this is best possible unless $NP \subseteq ZTIME[n^{\text{polylog}(n)}]$ [14,4,11]. Although the greedy algorithm's worst-case performance for the minimum-cardinality set cover problem is far from optimal, when one looks closely at its behavior, it does not seem to give a totally unreasonable solution in the sense that most of the universe U is usually covered by relatively large sets. In fact, it has been shown that, for any t, the number of elements covered by the t largest sets in the greedy set cover is at least $1 - ((t-1)/t)^t$ of the number of elements covered by the t largest sets in any set cover. In this paper we explore the greedy algorithm further, and show that it approximates the minimum entropy cover within a small *additive* constant. Thus, in this sense, the greedy algorithm actually finds a cover which explains the data nearly as well as the optimal distribution.

We further show that one can actually enhance the greedy algorithm to a polynomial time approximation scheme (PTAS) for the minimum entropy cover problem. Finally, we show how we can use the PTAS and the greedy algorithm in various scenarios arising in computational biology, and we explore the theoretical and empirical behavior of the greedy algorithm in these special cases.

2. The minimum entropy cover problem

The problem we consider in this paper is a variant of the minimum-cardinality set cover problem. We begin by formally defining the problem. In the next section, we give the main motivation for the problem.

We first need some notations and definitions. Throughout the paper, all logarithms are taken to base 2. For the sake of clarity, we will assume that the value of the function $x \log x$ at zero is zero (formally, we could define a different function which is $x \log x$ for $x > 0$ and zero for $x = 0$). The concentration of a multiset $n_1, n_2, \ldots, n_k$ of natural numbers is defined as $\sum_{i=1}^{k} n_i \log n_i$. If $N = n_1 + \cdots + n_k$, then the entropy of $\{n_i\}$ is $\sum_{i=1}^{k} n_i/N \log N/n_i$, which is simply the entropy of the distribution $(p_1, \ldots, p_k)$ where $p_i = n_i/N$.

A set system is a universe U and a collection $\mathcal{S} = (S_1, \ldots, S_t)$ of subsets of U. A *cover* is a function $f : U \to \mathcal{S}$ such that, for all $u \in U$, $u \in f(u)$. The entropy of the cover f, denoted by $ENT(f)$, is the entropy of the sequence of numbers $\{|f^{-1}(S_i)|\}$. Similarly, the concentration of the cover f, denoted by $CON(f)$ is the concentration of $\{|f^{-1}(S_i)|\}$.

We are now ready to define the minimum entropy cover problem.

Definition 1. The Minimum Entropy Cover Problem (MIN-ENT)
Input: A set system $(U, \mathcal{S})$.
Output: A cover $f : U \to \mathcal{S}$.
Goal: Minimize $ENT(f)$.

Informally, in the minimum entropy cover problem we are interested in finding a cover such that the distribution of the cover is as concentrated as possible. Thus, a related problem is the maximum concentration problem, which is formally defined as follows.

Definition 2. The maximum concentration cover problem.
Input: A set system $(U, \mathcal{S})$.
Output: A cover $f : U \to \mathcal{S}$.
Goal: Maximize $CON(f)$.

Clearly, a cover of maximum concentration is also a cover of minimum entropy and vice versa, since there is an affine relationship between the entropy and the concentration. In particular, $ENT(f) = \log N - CON(f)/N$.

3. A random generative model

In this section we introduce a probabilistic model for classification or identification problems with noisy data, and show that these problems can be formulated as instances of the maximum concentration problem. The setting for this model is as follows. We are given a set of *objects* drawn uniformly at random from a population. Each member of the population has a *type*. We are not told the types of the given objects, but we perform an *observation* on each object which determines that its type lies within some set of types. Given these observations we would like to find the most likely assignment of types to the objects.

Let T be the set of types, Ω the set of objects, and A the set of possible observations. Each observation $a \in A$ consists of a pair $(\omega(a), COMPAT(a))$, where $\omega(a) \in \Omega$ is an object and $COMPAT(a) \subseteq T$ is a subset of the types. If $i \in COMPAT(a)$ then type i is said to be *compatible* with observation a. Let $P(a|i)$ be the conditional probability of observation a, given that the object observed is of type i. Our key assumption is that for each $a \in A$ there is a positive real number $q(a)$ such that, for every $i \in COMPAT(a)$, $P(a|i) = q(a)$, and for every $i \notin COMPAT(a)$, $P(a|i) = 0$. Thus, we assume that, for all types compatible with observation a, the conditional probability of observation a is the same. We also assume that these conditional probabilities are fixed (but not necessarily known). In the important case where each type is specified by a vector of attributes and a randomly chosen subset of the attributes get observed, our assumption holds provided that the random choice of attributes to be observed is independent of the type of the object.

Suppose N objects are drawn from the population and a_j is the observation of object j. An *assignment* is a function f which assigns to each object j a type compatible with its observation. Let p_i be the (unknown) frequency of type i in the population. Then the joint probability of the observations $(a_1, a_2, \ldots, a_N)$ and the event that each object j is of type $f(j)$ is given by $\Pi_{j=1}^{N} q(a_j)p(f(j))$. We call this quantity the *joint likelihood* of the assignment of types and the observations of the objects. Note that $\Pi_{j=1}^{N} q(a_j)$ is fixed, by the assumption that the sets $COMPAT(a)$ are part of the specification of the model, and that the probabilities $q(a)$ are fixed. Thus the joint likelihood is maximized by maximizing the product $\Pi_{j=1}^{N} p(f(j))$. For each type i, let $n_i = |f^{-1}(i)|$ Then we wish to maximize $\Pi_i p_i^{n_i}$. Using simple calculus, one can verify that this quantity is maximized by choosing $p_i = n_i/N$. With this choice the function to be maximized becomes $\Pi_i (n_i/N)^{n_i}$. Taking logarithms and using the fact that the n_i sum to N, this is equivalent to maximizing the concentration $\sum_i n_i \log n_i$. Thus, the problem of maximizing the joint likelihood is an instance of the maximum concentration problem where, for each i, $S_i = \{j \mid i \in COMPAT(a_j)\}$.

4. The complexity of MIN-ENT

As noted above, a maximum concentration cover is also a minimum entropy cover, and thus, if one of these problems is solvable in polynomial time then so is the other. Unfortunately, the problems are NP-hard. In fact, we prove the following stronger theorem:

Theorem 1. *Maximum concentration cover is APX-hard.*

Proof. We use a reduction from the 3-set packing problem. In the 3-set packing problem we are given a 3-uniform hypergraph $H = (V, E)$, where $|V| = n$, and the goal is to find a maximum pairwise disjoint set of edges. The problem is equivalent to finding a maximum matching in a 3-uniform hypergraph (recall that a matching is a set of pairwise disjoint edges). In [13] (see also [8]) it is proven that there is a constant $c < 1$, such that it is NP-hard to distinguish between the case that there exists a perfect matching, that is a matching of size $n/3$ and the case that the maximum matching is of size at most $c \cdot (n/3)$.

Given an input to the 3-set packing problem, we treat it as an input to the maximum concentration cover problem, where the universe is V and the sets are the edges. Consider first the case in which there is a perfect matching of size $n/3$. In this case the concentration of the corresponding cover is $n \log 3$.

Consider the other case where the maximum matching is of size at most $c(n/3)$. Let f be a maximum concentration cover in this set system. Let A be the set of edges which cover exactly three vertices. The concentration of the cover f is at most $3|A| \log 3 + (n - 3|A|) \log 2$. On the other hand, $|A| \leqslant c(n/3)$. Therefore, the concentration of f is at most

$$cn(\log 3 - \log 2) + n \log 2 = c'n \log 3,$$

where $c' = c + (1 - c) \log 2/ \log 3 < 1$ is a fixed constant.

Therefore, it is NP-hard to distinguish between the case that the concentration is $n \log 3$ and the case that the concentration is smaller that $c'n \log 3$. Thus, the maximum concentration problem is APX-hard. □

Note that the fact that approximating the concentration within an arbitrary constant is hard does not imply that approximating MIN-ENT within an arbitrary constant is hard! It simply implies that MIN-ENT is NP-hard. In fact, we will actually show that MIN-ENT admits a PTAS.

4.1. The greedy algorithm

Although it is hard to approximate the maximum concentration cover within an arbitrarily small constant factor, we shall prove a surprising property: the greedy algorithm provides an approximation with a small *additive* error.

The greedy algorithm constructs a cover $f_G : U \to \mathcal{S}$ in the following way. We iteratively add a set $S_i \in \mathcal{S}$ which covers the maximum number of elements of U. We remove all its elements from U and from the other sets of $\mathcal{S}$, and recurse on the resulting set system. Thus, if $S_{i_1}, S_{i_2}, \ldots,$ are the sets chosen by the greedy algorithm, then $f_{\mathrm{G}}^{-1}(S_{i_1}) = S_{i_1}$, $f_{\mathrm{G}}^{-1}(S_{i_2}) = S_{i_2} \setminus S_{i_1}$, and in general, $f_{\mathrm{G}}^{-1}(S_{i_k}) = S_{i_k} \setminus (S_{i_1} \cup \cdots \cup S_{i_{k-1}})$.

Let $N = |U|$. We now prove the following theorem.

Theorem 2. *Let f_{OPT} be a cover of maximum concentration. Let f_{G} be the cover produced by the greedy algorithm. Then $ENT(f_{\mathrm{G}}) \leqslant ENT(f_{OPT}) + 3$. Equivalently, by the definition of $CON(f_{\mathrm{G}})$ and $ENT(f_{\mathrm{G}})$, $CON(f_{\mathrm{G}}) \geqslant CON(f_{OPT}) - 3N$.*

Theorem 2 may not seem intuitive at first sight in view of the $\log n$ approximation factor for the performance of the greedy algorithm on the minimum-cardinality set cover problem. The theorem gives a new interpretation for the greedy algorithm: it finds a cover with an almost minimum entropy. In many real life situations, a minimum-entropy cover seems more 'natural' than a minimum-cardinality cover.

Before proving Theorem 2 we need to introduce some more notations and definitions. For two non-increasing sequences $\{n_i\}$ and $\{m_i\}$ of non-negative real numbers, we say that $\{n_i\}$ majorizes $\{m_i\}$ if for every $k \geqslant 1$, their partial sums satisfy $n_1 + \cdots + n_k \geqslant m_1 + \cdots + m_k$. The following is a standard fact about convex functions, and it will be repeatedly used in our proof (see e.g. [7]):

Lemma 1. *Let F be a non-decreasing convex function such that $F(0) = 0$, and let $\{n_i\}$ and $\{m_i\}$ be two non-increasing sequences of non-negative real numbers such that $\{n_i\}$ majorizes $\{m_i\}$. Then $\sum_i F(n_i) \geqslant \sum_i F(m_i)$, where each sum is taken over all the elements of the sequence.*

Let $S_{i_1}, S_{i_2}, \ldots,$ be the sets chosen by the greedy algorithm. Furthermore, let $g_j = |f_G^{-1}(S_{i_j})|$ be the size of the jth set covered by the greedy algorithm. By definition of the greedy algorithm, $g_1 \geqslant g_2 \geqslant \cdots$. Let $B_1, B_2, \ldots,$ be the sets chosen by an optimal cover f_{OPT}, that is, for each j, there exists some i such that $B_j = f_{OPT}^{-1}(S_i)$. Finally, let $n_j = |B_j|$

and assume without loss of generality that $n_1 \geqslant n_2 \geqslant \cdots$. Theorem 2 states that $\sum g_i \log g_i \geqslant \sum n_i \log n_i - 3N$. In order to prove the theorem, we show that $\{g_i\}$ majorizes a certain multiset which is directly defined by $\{n_i\}$, and we then bound the concentration of that multiset.

Lemma 2. *For all i, $g_{i+1} \geqslant \left\lceil \max_k \left[\left(\sum_{j=1}^{k} n_j - \sum_{j=1}^{i} g_j \right) / k \right] \right\rceil$.*

Proof. For every k, the number of elements covered by the largest k sets of f_{OPT} is $n_1 + \cdots + n_k$, where the sets of f_{OPT} are $B_1, B_2, \ldots$, as before. On the other hand, the number of elements covered by the first i sets of the greedy algorithm is $g_1 + \cdots + g_i$. Therefore, before the $i+1$th iteration of greedy, there are at least $\sum_{j=1}^{k} n_j - \sum_{j=1}^{i} g_j$ uncovered elements in $B_1 \cup \cdots \cup B_k$. By averaging, there is at least one set B_l for some $l \in \{1, \ldots, k\}$ such that the number of uncovered elements in B_l is at least $\left(\sum_{j=1}^{k} n_j - \sum_{j=1}^{i} g_j\right)/k$, and thus $g_{i+1} \geqslant \frac{\sum_{j=1}^{k} n_j - \sum_{j=1}^{i} g_j}{k}$. Since this is true for every k, the lemma follows. □

Motivated by Lemma 2, we define a multiset $\{m_i\}$ in the following way. Let $m_1 = n_1$, and for $i \geqslant 2$ let

$$m_{i+1} = \left\lceil \max_k \left[\frac{\sum_{j=1}^{k} n_j - \sum_{j=1}^{i} m_j}{k} \right] \right\rceil.$$

We call this multiset the *extremal greedy multiset*.

Lemma 3. *The concentration of the greedy cover is at least the concentration of the extremal greedy multiset.*

Proof. We prove by induction on i that $\sum_{j=1}^{i} m_j \leqslant \sum_{j=1}^{i} g_j$ for all i. By Lemma 2, we get that $m_1 \leqslant g_1$. Assume for induction that $\sum_{j=1}^{i} m_j \leqslant \sum_{j=1}^{i} g_j$. Let k be such that $m_{i+1} = \left\lceil \left(\sum_{j=1}^{k} n_j - \sum_{j=1}^{i} m_j\right)/k \right\rceil$. Then, by Lemma 2,

$$\begin{aligned} m_{i+1} &= \left\lceil \frac{\sum_{j=1}^{k} n_j - \sum_{j=1}^{i} m_j}{k} \right\rceil \leqslant \left\lceil \frac{\sum_{j=1}^{k} n_j - \sum_{j=1}^{i} g_j}{k} \right\rceil + \left\lceil \frac{\sum_{j=1}^{i} g_j - \sum_{j=1}^{i} m_j}{k} \right\rceil \\ &\leqslant g_{i+1} + \left\lceil \frac{\sum_{j=1}^{i} g_j - \sum_{j=1}^{i} m_j}{k} \right\rceil, \end{aligned}$$

and so

$$\begin{aligned} \sum_{j=1}^{i+1} m_j &\leqslant \sum_{j=1}^{i} m_j + g_{i+1} + \left\lceil \frac{\sum_{j=1}^{i} g_j - \sum_{j=1}^{i} m_j}{k} \right\rceil \\ &= \sum_{j=1}^{i+1} g_j + \left(\sum_{j=1}^{i} m_j - \sum_{j=1}^{i} g_j \right) + \left\lceil \frac{\sum_{j=1}^{i} g_j - \sum_{j=1}^{i} m_j}{k} \right\rceil \leqslant \sum_{j=1}^{i+1} g_j, \end{aligned}$$

where the last inequality follows from the induction hypothesis and the fact that g_j and m_j are integers. Since $\sum_{j=1}^{i} m_j \leqslant \sum_{j=1}^{i} g_j$, then by Lemma 1, $\sum g_j \log g_j \geqslant \sum m_j \log m_j$, that is, the concentration of greedy is greater than the concentration of the extremal greedy multiset. □

We now describe another intermediate multiset $\{r_i\}$ whose concentration is at most that of the extremal greedy multiset. We then proceed to show that the concentration of $\{n_j\}$ exceeds that of $\{r_i\}$ by at most N. For each i, r_i will be equal to $\left\lceil \left(\sum_{j=1}^{k_i} n_j - \sum_{j=1}^{i-1} r_j\right)/k_i \right\rceil$, where the choice of the index k_i is as follows. Let $J_l = \{j \mid 2^{l-1} < N/n_j \leqslant 2^l\}$, let $W_l = \sum_{j \in J_l} n_j$ and let $t_l = \max_{j \in J_l} j$. Then, we set $k_i = \min\{t_l \mid W_1 + W_2 + \cdots + W_l > r_1 + r_2 + \cdots + r_{i-1}\}$.

Lemma 4. *The concentration of the extremal greedy multiset is greater than or equal to $\sum_i r_i \log(r_i) - N$. In other words, $\sum_i r_i \log(r_i) \leqslant N + \sum_j m_j \log(m_j)$.*

Proof. For every $l \geqslant 1$, let $R_l = \{i \mid k_i = t_l\}$. Let $r_{l,1} \geqslant r_{l,2}, \ldots$ be the set of $r_i \in R_l$. Then, $r_{l,i+1} = \lceil W_l - (r_{l,1} + \cdots + r_{l,i})/t_l \rceil$.

We consider another intermediate multiset $\{a_i\}$ which is defined by applying the following modifications to the multiset $\{m_i\}$. We define a breakpoint at m_i if for some l, $m_1+\cdots+m_{i-1} < W_1+\cdots+W_l \leqslant m_1+\cdots+m_i$. We replace the element m_i by a new element m'_i such that $m_1+\cdots+m'_i = W_1+\cdots+W_l$. We then replace m_{i+1} by $m_{i+1}+m_i-m'_i$. It is easy to see that the resulting multiset $\{a_i\}$ satisfies that $\sum m_i \log m_i \geqslant \sum a_i \log a_i - N \log 2 = \sum a_i \log a_i - N$ and that in every interval W_l, if $a_{l1} \geqslant a_{l2} \geqslant \cdots$ are the elements of $\{a_i\}$ in that interval, then $a_{l,i+1} \geqslant \lceil W_l-(a_{l1}+\cdots+a_{li})/t_l \rceil$.

Since for every $l \geqslant 1$, $\{a_{li}\}$ majorizes $\{r_{l,i}\}$ then by Lemma 1, $\sum a_i \log a_i \geqslant \sum r_i \log r_i$, and thus the lemma follows. □

Since the multiset $\{r_i\}$ is explicitly given, we can lower bound its concentration by a simple calculation.

Lemma 5. *For every* $l \geqslant 1$, $\sum_{i \in R_l} r_i \log r_i \geqslant W_l \log W_l - W_l \log t_l - W_l$.

Proof. First, recall that if $r_{l,1} \geqslant r_{l,2}, \ldots$ are the set of $r_i \in R_l$, then $r_{l,i+1} = \lceil \big(W_l - (r_{l,1}+\cdots+r_{l,i})\big)/t_l \rceil$. Therefore, one can show by induction that $r_{1,1}+\cdots+r_{l,i} \geqslant \sum_{j=1}^{i} W_l/t_l(1-(1/t_l))^{j-1}$. Thus, by Lemma 1 we have

$$\begin{aligned}
\sum_{j \in R_l} r_j \lg r_j &\geqslant \sum_{i=1}^{\infty} \frac{W_l}{t_l}\left(1-\frac{1}{t_l}\right)^{i-1} \log\left(\frac{W_l}{t_l}\left(1-\frac{1}{t_l}\right)^{i-1}\right) \\
&= W_l \log\left(\frac{W_l}{t_l}\right) + \frac{W_l}{t_l}\log\left(1-\frac{1}{t_l}\right)\sum_{i=1}^{\infty}(i-1)\left(1-\frac{1}{t_l}\right)^{i-1} \\
&= W_l \log\left(\frac{W_l}{t_l}\right) + W_l(t_l-1)\log\left(1-\frac{1}{t_l}\right) \\
&\geqslant W_l \log\left(\frac{W_l}{t_l}\right) - W_l. \qquad \square
\end{aligned}$$

The proof of the following claim is straightforward by the definition of W_l and t_l.

Claim 1. $(t_l - t_{l-1})N/2^l < W_l \leqslant (t_l - t_{l-1})N/2^{l-1}$.

We now upper bound the concentration of f_{OPT} in each of the intervals W_l.

Lemma 6. $\sum_{j \in J_l} n_j \log n_j \leqslant W_l \log(W_l/(t_l - t_{l-1}))$.

Proof. For a set of $t = t_l - t_{l-1}$ numbers $a_1, \ldots, a_t$ such that $a_1 + \cdots + a_t = W_l$, it is easy to see that $\sum a_i \log a_i$ is maximized when for every i, $a_i = W_l/t$, and in that case, $\sum a_i \log a_i = W_l \log W_l/t$. Therefore, the lemma follows. □

Lemmas 6 and 5 allow us to bound the difference between the concentration of $\{r_i\}$ and that of $\{n_i\}$.

Lemma 7. $\sum_i n_i \log n_i - \sum_j r_j \log r_j \leqslant 2N$.

Proof. By the lemmas above,

$$\begin{aligned}
\sum_i n_i \log n_i - \sum_j r_j \log r_j &\leqslant \sum_l W_l\left(\log\left(\frac{t_l}{t_l - t_{l-1}}\right)+1\right) \\
&= N + \sum_l W_l \log\left(\frac{t_l}{t_l - t_{l-1}}\right) \\
&\leqslant N + \sum_l W_l \frac{t_{l-1}}{t_l - t_{l-1}} \leqslant N + N\sum_l \frac{t_{l-1}}{2^{l-1}},
\end{aligned}$$

where the last inequality follows from Claim 1. But note that $\sum_l (t_l - t_{l-1})/2^{l-1} \leqslant \sum_j n_j/N = 1$, and thus, $\sum_l t_l/2^l \leqslant 1$. □

We can now prove Theorem 2:

Proof. By Lemmas 3 and 4, $CON(f_G) \geqslant \sum r_i \log r_i - N$. On the other hand, by Lemma 7, $CON(f_{OPT}) - 2N \leqslant \sum r_i \log r_i$. Thus, $CON(f_G) \geqslant CON(f_{OPT}) - 3N$. □

Theorem 2 shows that the greedy algorithm comes within an additive constant of the optimal entropy. In order to implement the greedy algorithm, one has to solve the subroutine that finds a set $S \in \mathcal{S}$ which covers the maximum number of elements of U. If the collection $\mathcal{S}$ is given explicitly, then this subroutine can be done by enumerating over all possible sets. But in some scenarios, the sets are given implicitly, and then finding the set which covers the maximum number of uncovered elements may be NP-hard. If this subroutine admits an α-approximation algorithm for some $\alpha < 1$, then by tracing the proof of Theorem 2, one can verify that $CON(f_G) \geqslant CON(f_{OPT}) - (3 + \log(1/\alpha))N$. Examples where this result is applicable include covering the edges of a graph by cut-sets, covering the vertices of a graph by dominating sets, and covering a finite set of points in R^n by balls of a given radius.

4.2. A PTAS for MIN-ENT

The greedy algorithm finds a cover with relatively small entropy, but there is a family of instances in which the ratio between the optimal entropy and the entropy of the greedy cover is bounded above by a constant smaller than one. Consider for example the following instance. Let $U = \{1, \ldots, n\}$ and let $\mathcal{S} = \{S_0, S_1, \ldots, S_t\}$, where S_i is a random subset of U of size $n/2$ and S_0 is the complement of S_t. The optimum cover uses S_0 and S_t, and its entropy is $\log 2$. On the other hand, the solution of the greedy algorithm may result in the sequence of sets $S_1, S_2, \ldots, S_t$ and in this case the entropy would be $\frac{1}{2}\log 2 + \frac{1}{4}\log 4 + \frac{1}{8}\log 8 + \cdots > \log 2$. Note that in this case the difference between the entropy of the greedy and the optimal entropy is still bounded above by 3, and therefore there is no contradiction with Theorem 2.

In this section we show how can one enhance the greedy algorithm and find a polynomial time approximation scheme for MIN-ENT, that is, we show that for every constant $\varepsilon > 0$ one can approximate MIN-ENT within a factor of $1 + \varepsilon$.

We keep the notations from the previous section. We let $OPT = ENT(f_{OPT})$, and $a = 3/\varepsilon$. We say that f is a *large partial* cover of U, if the following three properties hold:

- The domain of f (denoted D_f) is a subset of U (that is, the cover does not have to cover all of U).
- For every $S \in \mathcal{S}$, either $f^{-1}(S)$ is empty or $|f^{-1}(S)| \geqslant N/2^a$.
- If $f^{-1}(S)$ is not empty, then $S \subseteq D_f$.

The support of a large partial cover f is $\mathcal{X}_f = \{S \in \mathcal{S} \mid f^{-1}(S) \neq \emptyset\}$. Note that if the support of f is $\mathcal{X}_f$, then $\cup_{S \in \mathcal{X}_f} S = D_f$. Let f be a large partial cover, and let $\mathcal{X}_f = \{S'_1, \ldots, S'_l\}$ be its support and D_f its domain. f is called a maximal partial cover if for every $x, y \in D_f$ such that $f(x) \neq f(y)$ there is $i \leqslant l$, such that $x \in S'_i$, $y \notin S'_i$ or $x \notin S'_i$, $y \in S'_i$. A cover g of U is an extension of f if for every $i \in D_f$, $g(i) = f(i)$. The algorithm is the following:

1. Apply the greedy algorithm. Let the concentration of the resulting cover be CON_0.
2. For every large maximal partial cover f, find an extension g of f by applying the greedy algorithm to all the sets that are not entirely covered by f.
3. Output the cover with maximum concentration among CON_0 and all the covers found in step 2

We first prove that the algorithm indeed gives a $1+\varepsilon$ approximation. First note that if $OPT > 3/\varepsilon$, then by Theorem 2, the greedy algorithm finds a cover f such that $ENT(f) \leqslant OPT + 3 < (1 + \varepsilon)OPT$. We thus assume that $OPT \leqslant 3/\varepsilon$.

Let $k = \max_{n_j > N/2^a} j$, that is k is the maximal index such that $n_j > N/2^a$. Let $X = \sum_{j \geqslant k+1} n_j$. Then,

$$N \log N - N \cdot OPT = CON(f_{OPT}) \leqslant X(\log N - a) + (N - X)\log N,$$

and thus, $X \leqslant N \cdot OPT/a$.

It is easy to see that if B_j is the set corresponding to n_j in the optimal solution, then the projection of the optimal cover to $B_1 \cup \cdots \cup B_k$ is a large maximal partial cover. Therefore, in step 2 of the algorithm, one possible large maximal partial cover is the one defined by the multiset $n_1, n_2, \ldots, n_k$. For this specific partial cover, the algorithm extends it

to a cover g such that its concentration satisfies

$$CON(g) \geqslant \sum_{j=1}^{k} n_j \log n_j + \sum_{j \geqslant k+1} n_j \log n_j - 3 \sum_{j \geqslant k+1} n_j \geqslant CON(f_{OPT}) - 3\frac{N \cdot OPT}{a}.$$

Thus,

$$ENT(g) = \log N - \frac{CON(g)}{N} \leqslant \log N - \frac{CON(f_{OPT})}{N} + 3\frac{OPT}{a} = OPT\left(1 + \frac{3}{a}\right) = OPT(1+\varepsilon).$$

Finally, it remains to show that the algorithm can be implemented in polynomial time. Clearly, the greedy algorithm can be implemented in polynomial time. Thus, it suffices to show that one can enumerate over all large maximal partial covers in polynomial time.

Note that the support of a large partial cover contains at most 2^a sets. Hence, we can enumerate over all possible supports of these covers since there are at most $t^{2^a} = t^{2^{3/\varepsilon}}$ such supports. Let $\mathcal{X} = \{S'_1, \ldots, S'_l\}$, where $l \leqslant 2^a$. We bound the number of large maximal partial covers with support $\mathcal{X}$ and domain $D = S'_1 \cup \cdots \cup S'_l$. Let $\mathcal{A} = \{A_1, A_2, \ldots, A_{2^l}\}$ be the subsets of D defined by the possible intersections of sub-collections of $\mathcal{X}$. It is easy to see that by enumerating over all partitions of D by sets of $\mathcal{A}$, we enumerate over all large maximal partial covers with support $\mathcal{X}$. There are at most $2^{2^l} \leqslant 2^{2^{2^a}}$ such partitions. We thus get the following theorem:

Theorem 3. *For every $\varepsilon > 0$, there is a $(1+\varepsilon)$-approximation algorithm for MIN-ENT which runs in time $O(2^{2^{2^{3/\varepsilon}}} \cdot t^{2^{3/\varepsilon}} \cdot (Nt)^{O(1)})$.*

5. Applications

In this section, we introduce two scenarios where the random generative model is helpful.

5.1. The haplotype resolution problem

We introduce an application which naturally arises in computational biology, but can also be viewed as a more general string-oriented problem.

A partial haplotype is a string over $\{0, 1, *\}^k$. A complete haplotype is simply a binary string of size k. A complete haplotype h is compatible with a partial haplotype h' if and only if for each i, if $h'(i) \neq *$ then $h(i) = h'(i)$.

In the haplotype resolution problem, we are given a set $U = \{h_1, h_2, \ldots, h_m\}$ of partial haplotypes of length k. For each complete haplotype $h \in \{0,1\}^k$, let $S_h = \{h_i \in U \mid h \text{ is compatible with } h_i\}$. The set U together with its collection of subsets $\mathcal{S} = \{S_h \mid h \in \{0,1\}^k\}$ forms a set system. We wish to find a minimum-entropy cover for this system.

The problem arises in the following biological context. A geneticist conducts an experiment, in which one of the steps is to sequence the DNA of a sample of individuals from the population. Unfortunately, current sequencing technology often gives the DNA sequence with some missing nucleotide bases at some positions. Our goal is to complete these missing bases. In terms of the notations above, each partial haplotype $h_i \in U$ corresponds to the DNA sequence of one individual, and the $*$ values correspond to missing bases. Clearly, the data observed by the geneticist follows the random generative model described in Section 3, where the types are the complete haplotypes, the observations are the partial haplotypes in U, and for each $h_i \in U$, $COMPAT(h_i) = \{h \in \{0,1\}^k \mid h_i \in S_h\}$. Thus, by the analysis given in Section 3, the most likely completion of the partial haplotypes is the one defined by the minimum entropy cover.

Since the haplotype resolution cover is a special case of MIN-ENT, there is hope to find a polynomial-time algorithm for it. We now show that this is not possible in the general case.

Theorem 4. *The haplotype resolution problem is APX-hard.*

Proof. In [10], Khanna et al. introduce a graph $G = (V, E)$, such that there exists $r = |V|^{1/c}$ for some $c > 1$, for which distinguishing between the following two cases is NP-hard:

- The case that G is r-colorable and the maximum independent set of G is of size r^{c-1}.
- The case where the maximum independent set of G is of size at most $r^{\varepsilon(c-1)}$, for some constant ε.

Consider such a graph G. We construct an instance of the haplotype resolution problem from this graph. The construction is similar to a construction given by [18] for a different problem. In this instance, we set the length of the strings to be $k = |V|$, where each position corresponds to a vertex in G. For each vertex v in G, we add a partial haplotype h_v to U which has a value 1 in v, a value 0 in every neighbor of v, and a value $*$ in any other position. It is easy to see that an independent set in G corresponds to a complete haplotype covering the set of partial haplotypes corresponding to the vertices of the independent set. Thus, if G is r-colorable and the maximum independent set of G is of size r^{c-1}, one can find a cover of entropy $\log r$, and if the maximum independent set of G is of size $r^{\varepsilon(c-1)}$, then the entropy of the minimum entropy cover is at least $(c - \varepsilon(c-1)) \log r$. Thus, it is NP-hard to distinguish between the case that the minimum entropy is $\log r$, and the case that the minimum entropy is $(c - \varepsilon(c-1)) \log r$, and therefore, finding the minimum-entropy cover is APX-hard. □

In the context of haplotype resolution, the greedy algorithm iteratively finds the complete haplotype which covers the maximum number of partial haplotypes in the data set. It then completes these partial haplotypes to that haplotype, and removes them from the data set. When $k = O(\log m)$, finding the complete haplotype can be done in polynomial time, simply by enumerating over all possible complete haplotypes. For an arbitrary k, this is NP-hard [18]. For practical data sets, the length of the DNA sequences is quite short (around 10) due to some practical considerations. [2] Therefore, for such regions, one can efficiently apply the greedy algorithm. In Section 6 we report some successful results over real biological data.

5.2. The genotype phasing problem

The genotype phasing problem is another problem arising from the field of computational biology. The motivation of this problem is similar to the haplotype resolution problem—the setting is where a geneticist is sequencing DNA sequences randomly sampled from a population. Recall that each chromosome in our cells has two copies, one transmitted from the mother and the other from the father. Current technology applicable for large-scale sequencing gives the information from both copies at the same time and not the information from each chromosome separately. That is, in each position, if both copies share the same value we know which value it is, and if they differ, we know their two values, but we do not know which copy has which value. The goal is to resolve these ambiguities.

Recall that a haplotype is a string over $\{0, 1\}^k$. A genotype is a string over $\{0, 1, 2\}^k$. A genotype g is compatible with the (complete) haplotypes (h_1, h_2) if in every position i, if $g(i) \in \{0, 1\}$ then $h_1(i) = h_2(i) = g(i)$, and otherwise $h_1(i) \neq h_2(i)$. In this case we say that g is covered by h_1 and h_2. A cover f of a set of genotypes G, is a function $f : G \to \{0, 1\}^k \times \{0, 1\}^k$ where for each $g \in G$, if $f(g) = (h_1, h_2)$ then g is compatible with (h_1, h_2). For each haplotype $h \in \{0, 1\}^k$, the coverage of h (denoted $COV(h, f)$) is the number of genotypes g such that for some haplotype h', $f(g) = (h, h')$ or $f(g) = (h', h)$, where g is counted twice if $f(g) = (h, h)$.

In the genotype phasing problem we are given a set of genotypes, $G = (g_1, \ldots, g_n)$ of length k each. We are interested in finding a cover $f : G \to \{0, 1\}^k \times \{0, 1\}^k$ such that the entropy of $\{COV(h, f)\}$ is minimized. This problem does not correspond exactly to our random generative model, but we next show how our results imply an approximation algorithm for the minimum entropy cover of genotypes when $k = O(\log n)$.

Theorem 5. *Assume $k = O(\log n)$. Then there is a polynomial-time algorithm for the maximum concentration genotype problem, such that if OPT is the maximum possible concentration, then the concentration given by the algorithm is at least $OPT/2 - O(n)$.*

Proof. For every genotype g and haplotype h where there is h' such that (h, h') is compatible with g, we say that h' is the complement of h under g. Consider the following algorithm. In each step find a haplotype which is compatible with the maximum number of uncovered genotypes. Cover these genotypes using this haplotype and its complement. Let CON be the concentration of the algorithm, and let f_{OPT} be the optimal cover. Let X_1 be a multiset derived by taking the first haplotype covering each genotype, and X_2 be the multiset derived by taking the second haplotype covering each

[2] The number of sequenced individuals is usually not very large, and a long sequence would mean that each DNA sequence appeared only once in the data set (and thus, there is no information). Another reason to use short regions is that there are strong correlations among different positions in the DNA that are physically close to each other.

Data Set	Total missing	Added missing	Greedy error rate
Daly et al.	26%	10%	2.8%
Gabriel et al.	10.5%	0.5%	8.1%
Gabriel et al.	15%	5%	7.4%
Gabriel et al.	20%	10%	7.8%

Fig. 1. The performance of the greedy algorithm under the different data sets and the different missing data ratio. The first column specifies the data set on which the experiment was done. The second column specifies the total missing data given to the algorithm—this missing data contain the added missing data and the missing data of the original data. The third column specifies the value of p, and the fourth column specifies the error rate of the algorithm, that is the number of incorrectly reconstructed masked positions divided by the total number of masked positions.

	HAPLOTYPER	PHASE	HAP	GREEDY
Gabriel et al.	–	4.4 %	3.7 %	7.3 %
Daly et al.	4%	1.65 %	1.27 %	0.82 %

Fig. 2. The results for the genotype phasing algorithm. Each column corresponds to a different algorithm and each row corresponds to a different data set. Evidently, on the Daly et al. data set, the greedy algorithm outperforms the other algorithms. On the other hand, on the Gabriel data set the greedy algorithm does not perform as well, although its error rate is comparable to the other algorithms.

genotype. Then, the concentration of f_{OPT} is at most $CON(X_1)+CON(X_2)+2n$. Assume that $CON(X_1) \geqslant CON(X_2)$. By Theorem 2, $CON \geqslant CON(X_1) - 3n$. Thus, $CON \geqslant (CON(f_{OPT}) - 7n)/2$. □

6. Experimental results

We measured the performance of the greedy algorithm in practice, both for genotype phase reconstruction and for haplotype missing data completion. Our results show that the greedy algorithm, which is very simple to state and to implement, performs reasonably well, and for certain data it is even better then previous phase reconstruction algorithms such as PHASE [19], HAPLOTYPER [12] and HAP [3,6].

The data sets: We applied our algorithm to two haplotype data sets from [2,15] and population D of [5]. Both these data sets have a significant portion of the genotype data (about 10%) missing. The data of [2] are partitioned into blocks. We arbitrarily partitioned the data of [5] into blocks of length 10. We performed our experiments on each of these blocks.

Completing missing haplotypes: We measured the performance of the greedy algorithm on haplotypes with about 10% missing data. We added random missing data by masking each position independently with probability p for different values of p. We gave the greedy algorithm as input the resulting haplotypes, and then compared the resulting haplotypes of the greedy algorithm to the original haplotypes. We considered each masked position and we observed if it was correctly reconstructed or not. We found that the error rate in the reconstruction is only a few percent in both data sets, even when the missing data consists of about 25% of the data. The results are given in Fig. 1.

Phasing genotype data: We used the greedy algorithm to phase genotype data. We compared our results to the results given by three other phasing algorithms, namely HAP [3,6], PHASE [19] and HAPLOTYPER [12]. The results of the comparison are given in Fig. 2.

The results achieved by the greedy algorithm are competitive with the previous results. In fact, for the data taken from [2], the performance of the greedy algorithm is superior to the performance of all the other algorithms. For the Gabriel [5] data, the greedy algorithm is inferior to the other algorithms, but it still gives reasonable results, given its simplicity with respect to the other algorithms.

References

[1] V. Chvátal, A greedy heuristic for the set-covering problem, Math. Oper. Res. 4 (1979) 233–235.
[2] M.J. Daly, J.D. Rioux, S.F. Schaffner, T.J. Hudson, E.S. Lander, High-resolution haplotype structure in the human genome, Nature Genetics 29 (2) (2001) 229–232.

[3] E. Eskin, E. Halperin, R. Karp, Efficient reconstruction of haplotype structure via perfect phylogeny, J. Bioinformatics Comput. Biol. 1 (1) (2003) 1–20.
[4] U. Feige, A threshold of $\ln n$ for approximating set cover, J. ACM 45 (1998).
[5] G.B. Gabriel, S.F. Schaffner, H. Nguyen, J.M. Moore, J. Roy, B. Blumenstiel, J. Higgins, M. DeFelice, A. Lochner, M. Faggart, S.N. Liu-Cordero, C. Rotimi, A. Adeyemo, R. Cooper, R. Ward, E.S. Lander, M.J. Daly, D. Altshuler, The structure of haplotype blocks in the human genome, Science 296 (2002) 2225–2229.
[6] E. Halperin, E. Eskin, Haplotype reconstruction from genotype data using imperfect phylogeny, Bioinformatics, 2003.
[7] G.H. Hardy, J.E. Littlewood, G. Polya, Inequalities, Cambridge University Press, Cambridge, England, 1934.
[8] E. Hazan, M. Safra, O. Schwartz, On the complexity of approximating k-dimensional matching, in: Proc. of the sixth Internat. Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX), 2003.
[9] E.H. Herskovits, G.F. Cooper, Kutato: an entropy-driven system for construction of probabilistic expert systems from database, in: Proc. Sixth Conf. on Uncertainty in Artificial Intelligence, 1990, pp. 54–62.
[10] S. Khanna, N. Linial, S. Safra, On the hardness of approximating the chromatic number, in: Proc. second Israel Symp. on Theory and Computing Systems, Natanya, Israel, 1993, pp. 250–260.
[11] C. Lund, M. Yannakakis, On the hardness of approximating minimization problems, in: Proc. 25th Annu. ACM Symp. on Theory of Computing, San Diego, CA, 1993, pp. 286–293.
[12] Niu, Qin, Xu, Liu, In silico haplotype determination of a vast set of single nucleotide polymorphisms, Technical report, Department of Statistics, Harvard University, 2001.
[13] E. Petrank, The hardness of approximation: gap location, Comput. Complexity 4 (1994) 133–157.
[14] R. Raz, S. Safra, A sub-constant error-probability low-degree test, in: Proc. 29th Annu. ACM Symp. on Theory of Computing, El Paso, TX, 1997, pp. 475–484.
[15] J.D. Rioux, M.J. Daly, M.S. Silverberg, K. Lindblad, H. Steinhart, Z. Cohen, T. Delmonte, K. Kocher, K. Miller, S. Guschwan, E.J. Kulbokas, S. O'Leary, E. Winchester, K. Dewar, T. Green, V. Stone, C. Chow, A. Cohen, D. Langelier, G. Lapointe, D. Gaudet, J. Faith, N. Branco, S.B. Bull, R.S. McLeod, A.M. Griffiths, A. Bitton, G.R. Greenberg, E.S. Lander, K.A. Siminovitch, T.J. Hudson, Genetic variation in the 5q31 cytokine gene cluster confers susceptibility to Crohn disease, Nature Genetics 29 (2) (2001) 223–228.
[16] S. Roberts, R. Everson, I. Rezek, Minimum entropy data partitioning, in: Proc. Ninth Internat. Conf. on Artificial Neural Networks, 1999, pp. 844–849.
[17] S.J. Roberts, C. Holmes, D. Denison, Minimum-entropy data partitioning using reversible jump markov chain Monte carlo, IEEE Trans. Pattern Anal. Mach. Intell. 23 (8) (2001) 909–914.
[18] R. Sharan, Personal communication, 2003.
[19] M. Stephens, N. Smith, P. Donnelly, A new statistical method for haplotype reconstruction from population data, Amer. J. Human Genetics 68 (2001) 978–989.
[20] Y. Xiang, S.K.M. Wong, N. Cercone, A "microscopic" study of minimum entropy search in learning decomposable markov networks, Mach. Learn. 26 (1) (1997) 65–92.