Information and Computation 242 (2015) 354–368

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Information retrieval with unambiguous output

Ville Junnila[1], Tero Laihonen*

Department of Mathematics and Statistics, University of Turku, FI-20014 Turku, Finland

ARTICLE INFO

Article history:
Received 10 December 2013
Received in revised form 13 February 2015
Available online 9 April 2015

Keywords:
Associative memory
Information retrieval
Extremal graph
Covering code
Forced vertex
Algorithmic complexity

ABSTRACT

The main problem in information storage has previously been how large amounts of data can be stored. However, the technological development over the years has been able to give rather satisfactory answers to this problem. Recently, the focus has shifted towards determining how stored information can be efficiently retrieved. This problem is addressed in an article by E. Yaakobi and J. Bruck (2012), where information retrieval in associative memories is studied. In this paper, we focus on the case where the retrieved information unit is unambiguous. In particular, we present characterizations and study various extremal properties of such associative memories. Moreover, the algorithmic complexity of certain naturally rising problems is considered.

1. Introduction

Let $G = (V, E)$ be a graph on at least two vertices without loops or multiple edges. We also assume that G is connected and undirected. As usual, the graphic distance $d(x, y)$ is the number of edges in any shortest path between the vertices x and y. If the vertices x and y are adjacent, we denote $x \sim y$. The degree of a vertex $x \in V$ is denoted by $\deg(x)$, the minimum degree of G by $\delta = \delta(G)$ and the maximum degree by $\Delta = \Delta(G)$. We use the notation $G[C]$ for the induced subgraph with vertex set $C \subseteq V$.

The idea behind associative memories is to try to mimic the human memory [11], which is fundamentally associative. One can remember a new piece of information much better if it is associated with previously obtained knowledge which is already firmly anchored in the memory. The way we process and retrieve (by questions or clues) information is mainly performed by associations. The concept of associative memories is important, for example, for cognitive computing and machine learning.

In this paper, we consider the model of information retrieval in associative memories introduced by Yaakobi and Bruck in [11]. An associative memory is modeled by a graph $G = (V, E)$ as follows. The vertices of the graph represent the memory entries containing stored information units. The edges between vertices represent the associations of information units to each other. Two distinct vertices $x, y \in V$ are said to be *t-associated* for a positive integer t if $d(x, y) \le t$. The set of vertices t-associated to $x \in V$ (that is, the ball of radius t centered at $x \in V$) is denoted by

$$B_t(x) = \{y \in V \mid d(x, y) \le t\}.$$

For a subset $T \subseteq V$, we use the notation

* Corresponding author.
E-mail addresses: viljun@utu.fi (V. Junnila), terolai@utu.fi (T. Laihonen).

[1] Research supported by the Väisälä Foundation of the Finnish Academy of Science and Letters, and the Finnish Cultural Foundation.

http://dx.doi.org/10.1016/j.ic.2015.04.002
0890-5401/© 2015 Elsevier Inc. All rights reserved.

$$S_t(T) = \bigcap_{c \in T} B_t(c)$$

for the set of vertices which are t-associated to all elements of T. Let m be a positive integer. A *reference set* (or a *code*) is a subset $C \subseteq V$ with $|C| \geq m$. We retrieve the information units from the associative memory in the following manner (see [11]). A set of m distinct elements $\{c_1, \ldots, c_m\} \subseteq C$, called *input clues* (or *codewords*), are given. As an output set we receive the set of vertices, which are t-associated to all the m input clues, that is, the set $S_t(\{c_1, \ldots, c_m\})$. The smaller the output set is, the more precisely we know the information unit which we were looking for with the aid of the m input clues. The maximum size of an output set

$$N_t(m, C) = \max_{T \subseteq C, |T|=m} \{|S_t(T)|\},$$

is called the *uncertainty* of an associative memory with a reference set C in G.

We also want to make sure that we have an access to every information unit $x \in V$ with at least one set of m input clues. Let us denote the set of elements of C which are t-associated to x by

$$I_t(x) = I_t(C; x) = B_t(x) \cap C.$$

If $|I_t(x)| < m$, then we cannot retrieve x using any set of m input clues. The vertices $x \in V$ with $|I_t(x)| \geq m$ are called *accessible*. It is natural to require that *all* the vertices (i.e. information units) are accessible. We also wish to give an upper bound N on the uncertainty. This gives rise to the following definition.

Definition 1. Let $G = (V, E)$ be a simple, undirected and connected graph on at least two vertices. Let also $C \subseteq V$. Assume further that t, m and N are positive integers. We say that a pair (G, C) is a (t, m, N)-*associative memory with the reference set* C if

(i) $|I_t(x)| \geq m$ for any $x \in V$ and
(ii) $|S_t(T)| \leq N$ for any subset $T \subseteq C$ of size m.

In this paper, we focus on unambiguous output by setting $N = 1$. In other words, we demand that given m input clues the sought information unit is uniquely determined. Furthermore, we restrict ourselves to the radius $t = 1$. We omit the subscript t when $t = 1$ from both $B_t(x)$ and $I_t(x)$. We denote a $(1, m, 1)$-associative memory in short by $\mathcal{AM}(m)$. We say that C *gives* an $\mathcal{AM}(m)$ if it is a reference set of it. We also say that G *admits* an $\mathcal{AM}(m)$ if there exists a reference set C giving $\mathcal{AM}(m)$ in G.

It is convenient to use the following additional definitions. If $|I(C; x)| \geq s$ for any $x \in V$, then we say (see [1]) that C is an s-*fold* 1-*covering*. We also say that $x \in V$ *covers* $y \in V$ if $d(x, y) \leq 1$.

Previously, information retrieval in associative memories has been examined in [11] and [12], where the problem has been studied in binary Hamming spaces and Grassmann graphs, respectively. The study of information retrieval in binary Hamming spaces has been continued in [5]. There the problem is also considered in the case where the underlying graph is the infinite square grid.

In the following two remarks, we consider two applications of the concept discussed in Definition 1, namely, the sensor network monitoring and the Levenshtein's sequences reconstruction problem.

Remark 2. Consider a *sensor network monitoring* with RF-based localization introduced in [3,10] in indoor environments. Sensors in a building are mapped to vertices of a graph $G = (V, E)$ and a pair of vertices is connected by an edge if the two corresponding sensors are within each other's communication range. A small portion of all sensors $C \subseteq V$ are kept active while the others can be put in energy-saving mode. An observer (located at $x \in V$) periodically receives ID packets from neighboring sensors of C (in other words, from $I(x)$). Suppose C gives an $\mathcal{AM}(m)$ in G. The observer can determine her location uniquely based only on (at least) m ID packets from $I(x)$.

The codes giving $\mathcal{AM}(m)$ have the following advantage over the usual identifying codes [6,8] for sensor networks. It is enough to receive any m codewords from $I(x)$ to determine x uniquely in contrast to the usual identifying codes where one has compare $I(x)$ to all other $I(y)$, $y \in V$, to guarantee that the right location x is found.

Remark 3. Let our underlying graph G be the binary Hamming space $\mathbb{F}^n$ (see [11]) and C gives a (t, m, N)-associative memory. Levenshtein's *sequences reconstruction problem* [7,11] is motivated by situations in the fields like chemistry and biology, where the only way to overcome errors is to repeatedly transmit in a channel (say M times) the same codeword. In other words, a codeword $x \in C$ is transmitted through M channels where at most t errors can occur in each. Based on the M different outputs $y_1, \ldots, y_M \in V$ of the channels, a list decoder $\mathcal{D}_{\mathcal{L}}$ gives estimations $\{x_1, \ldots, x_\ell\}$ (where $\ell \leq \mathcal{L}$) on the transmitted word x. In [11] the minimum number of channels to guarantee the existence of a *successful* decoder is studied — a successful decoder naturally means that $x \in \{x_1, \ldots, x_\ell\}$. The reference sets discussed above provide a code for a successful decoder with $M = N + 1$ channels where N is the uncertainty (in this paper the focus is on $N = 1$, so we study situation where we need *only two* channels) and the parameter m gives an upper bound on the length of the list provided by the decoder, namely $\mathcal{L} < m$. For more details, see [11].