A PROGRAM FOR DETECTING CHORDALITY, WITH USER'S MANUAL

ALI AKBAR YAZDAN POUR

Abstract. Chordal clutters are natural generalization of chordal graphs and were first introduced in [4]. `ChordalityCheck.cpp` is a `C++` program that may determine whether a d-uniform clutter $\mathcal{C}$ is chordal or not. This short note aims to introduce briefly the class of chordal clutters and a manual for using the program `ChordalityCheck.cpp`. The source code of this program was provided by Iman Kiarazm.

1. Chordal clutters

Chordal clutters are natural generalization of chordal graphs based on Dirac theorem on characterization of chordal graphs [3]. This class of clutters were first introduced in [4]. One of the aspects of chordal graphs that makes it interesting, is perhaps that they may be described in different equivalent ways. Several mathematicians have generalized the definition of chordal graphs, to chordal clutters based on particular definition. Since such generalizations may be made in many different directions, no particular standard concerning the use of the word "chordal clutter" has been established. In [4], it is introduced a class of clutters $\mathfrak{C}_d$, that extends graph theoretical definition of chordal graphs in the case $d = 2$ and inherits the same algebraic properties of chordal graphs. This class is of great importance, because recently it has been shown that, this class contains other famous families of chordalities that have been introduced till now [2]. In the followings, we investigate the class $\mathfrak{C}_d$ with more details. For further information on this class, the reader may refer to [2, 4].

Definition 1.1 (Clutter)**.** Let $[n] = \{1, \dots, n\}$. A *clutter* $\mathcal{C}$ with vertex set $[n]$ is a collection of subsets of $[n]$, called *circuits* of $\mathcal{C}$, such that if F_1 and F_2 are distinct circuits, then $F_1 \nsubseteq F_2$. A *d-circuit* is a circuit consisting of exactly d vertices, and a clutter is called *d-uniform*, if every circuit has d vertices. A $(d-1)$-subset $e \subset [n]$ is called a *submaximal circuit* of $\mathcal{C}$, if there exists $F \in \mathcal{C}$ such that $e \subset F$. The set of all submaximal circuits of $\mathcal{C}$ is denoted by $\mathrm{SC}(\mathcal{C})$.

Let $\mathcal{C}$ be a clutter with vertex set $[n]$. For a subset $W \subseteq [n]$, the *induced subclutter* on W is denoted by $\mathcal{C}|_W$ and is defined as follows:

$$\mathcal{C}|_W = \{F \in \mathcal{C} : \quad F \subseteq W\}.$$

Let n, d be positive integers. For $n \geq d$, we define $\mathcal{C}_{n,d}$, the *complete d-uniform clutter* on $[n]$, as follows:

$$\mathcal{C}_{n,d} = \{F \subset [n] : \quad |F| = d\}.$$

In the case that $n < d$, we let $\mathcal{C}_{n,d}$ be some isolated points.

If $\mathcal{C}$ is a d-uniform clutter on $[n]$, we define $\bar{\mathcal{C}}$, the *complement* of $\mathcal{C}$, to be

$$\bar{\mathcal{C}} = \mathcal{C}_{n,d} \setminus \mathcal{C} = \{F \subset [n] : \quad |F| = d, \, F \notin \mathcal{C}\}.$$

Definition 1.2. Let $\mathcal{C}$ be a d-uniform clutter on $[n]$. A subset $V \subset [n]$ is called a *clique* in $\mathcal{C}$, if all d-subsets of V belong to $\mathcal{C}$. Note that a subset of $[n]$ with less than d elements is supposed to be a clique.

For any $(d-1)$-subset e of $[n]$, let

$$N_{\mathcal{C}}[e] = e \cup \{c \in [n] : \quad e \cup \{c\} \in \mathcal{C}\}.$$

We call $N_{\mathcal{C}}[e]$ the *closed neighborhood* of e in $\mathcal{C}$. We say that e is *simplicial* in $\mathcal{C}$, if $N_{\mathcal{C}}[e]$ is a clique in $\mathcal{C}$. Let us denote by $\operatorname{Simp}(\mathcal{C})$, the set of all simplicial elements of $\mathcal{C}$. More generally, for a subset $A \subset [n]$ with $|A| < d$, let

$$N_{\mathcal{C}}[A] = A \cup \{c \in [n] \colon \quad A \cup \{c\} \subseteq F, \text{ for some } F \in \mathcal{C}\}.$$

Example 1.3. Figure 1 displays a 3-uniform clutter $\mathcal{C}$ whose circuits are

$$\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{2,3,6\}, \{2,5,6\}, \{2,5,7\}, \{2,6,7\}, \{5,6,7\}.$$

In this clutter, $\{2,3\}$ and $\{2,6\}$ are not simplicial in $\mathcal{C}$, but all the other submaximal circuits of the set $\{1,\dots,7\}$ are simplicial in $\mathcal{C}$.

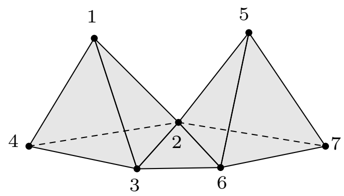

FIGURE 1. A 3-uniform clutter

Definition 1.4. Let $\mathcal{C}$ be a d-uniform clutter and let e be a $(d-1)$-subset of $[n]$. By $\mathcal{C} \setminus e$ we mean the d-uniform clutter

$$\{F \colon \quad F \in \mathcal{C}, \ e \nsubseteq F\}.$$

It is called the *deletion* of e from $\mathcal{C}$. In the case that e is not a submaximal circuit of $\mathcal{C}$, we have $\mathcal{C} \setminus e = \mathcal{C}$.

Definition 1.5. Let $\mathcal{C}$ be a d-uniform clutter. We call $\mathcal{C}$ a *chordal clutter*, if either $\mathcal{C} = \varnothing$, or $\mathcal{C}$ admits a simplicial submaximal circuit e such that $\mathcal{C} \setminus e$ is chordal.

We use $\mathfrak{C}_d$, to denote the class of all d-uniform chordal clutters. Our aim in this section is to show that, this definition is the natural generalization of chordal graphs. That is, with this definition, we show that the circuit ideal associated to these clutters has a linear resolution over any field. Note that the definition of chordal clutters can be restated as follows:

The d-uniform clutter $\mathcal{C}$ is chordal if either $\mathcal{C} = \varnothing$, or else there exists a sequence of submaximal circuits of $\mathcal{C}$, say $e_1, \dots, e_t$, such that e_1 is simplicial submaximal circuit of $\mathcal{C}$, e_i is simplicial submaximal circuit of $(((\mathcal{C} \setminus e_1) \setminus e_2) \setminus \cdots) \setminus e_{i-1}$ for all $i > 1$, and $(((\mathcal{C} \setminus e_1) \setminus e_2) \setminus \cdots) \setminus e_t = \varnothing$.

To simplify the notation, we use $\mathcal{C}_{e_1 \dots e_i}$ for $(((\mathcal{C} \setminus e_1) \setminus e_2) \setminus \cdots) \setminus e_i$.

Example 1.6. In Figure 2, the 3-uniform clutter $\mathcal{C}$ is chordal, while the 3-uniform clutter $\mathcal{D}$ is not.

$$\mathcal{C} = \{\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,5\}, \{1,2,6\}, \{1,5,6\}, \{2,5,6\}\}.$$
$$\mathcal{D} = \{\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,5\}, \{2,4,5\}, \{3,4,5\}\}.$$

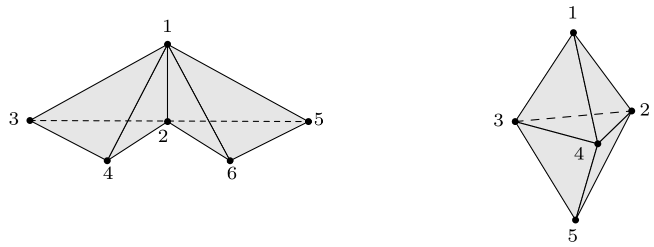

FIGURE 2. The Clutter $\mathcal{C}$ on the left and $\mathcal{D}$ on the right

Remark 1. It is worth to say that this definition of chordal clutters coincides with the graph theoretical definition of chordal graphs in the case $d = 2$. To see this, we recall that, a graph G is chordal, if and only if for every induced subgraph G' of G, one has $\mathrm{Simp}\,(G') \neq \varnothing$ (essentially [3]). It follows that, a graph G is chordal, if and only if there exists an order on vertices of G, say $v_1, \ldots, v_n$, such that v_i is simplicial in $G|_{\{v_1,\ldots,v_{i-1}\}}$. This is equivalent to say that $G \in \mathfrak{C}_2$.

2. A USER MANUAL FOR `ChordalityCheck.cpp`

Since chordal clutters are defined recursively, it is not unreasonable to expect a computer program for detecting whether a given clutter is chordal or not. A `C++` program for detecting chordality, `ChordalityCheck.cpp`, was prepared by Iman Kiarazm and the source code is available at [1]. In the following, it is introduced a step by step guide, for using this program:

Step 1: Provide the input. The input of this program is a d-uniform clutter $\mathcal{C} = \{F_1, \ldots, F_r\}$ where $F_i = \{a_{i,1}, \ldots, a_{i,d}\}$, for $i = 1, \ldots, r$. Make a text file `input.txt`, and write the circuits of $\mathcal{C}$ in this file as follows:

$$x[a_{1,1}]{*}x[a_{1,21}] * \cdots * x[a_{1,d}]$$
$$x[a_{2,1}]{*}x[a_{2,2}] * \cdots * x[a_{2,d}]$$
$$\vdots$$
$$x[a_{r,1}]{*}x[a_{r,2}] * \cdots * x[a_{r,d}]$$

Indeed, separate $x[a_{i,j}]$s with a star $(*)$ and circuits with a new line. For example, the file `input.txt` for clutter $\mathcal{C}$ in Example 1.3 is:

```
x[1]*x[2]*x[3]
x[1]*x[2]*x[4]
x[1]*x[3]*x[4]
x[2]*x[3]*x[4]
x[2]*x[3]*x[6]
x[2]*x[5]*x[6]
x[2]*x[5]*x[7]
x[5]*x[6]*x[7]
x[2]*x[6]*x[7]
```

Step 2: Change some arguments in the source file. At the line 22 of the source code, change `ARRAY_SIZE` to d, and at line 23, change the `ARRAY_COUNT` to r (or any positive integer greater than r). If `ARRAY_SIZE` is not set correctly, the program may give a wrong output. For instance, in the above example, we must set `ARRAY_SIZE` to 3 and `ARRAY_COUNT` to 9 (or any number greater than 9).

Step 3: Running the program. Place `input.txt` in the same directory as the source code and use a `C++` editor (e.g. Dev-C++) to compile and run the program[1]. Then the file `output.txt` contains the result concerning chordality of the clutter $\mathcal{C}$.

If the clutter $\mathcal{C}$ is chordal, the file `output.txt` gives you a sequence $e_1, \ldots, e_s$, such that e_1 is a simiplicial submaximal circuit of $\mathcal{C}$, e_i is a simplicial submaximal circuit of $\mathcal{C}_{e_1 \cdots e_{i-1}}$, for $i = 2, \ldots, s$ and $\mathcal{C}_{e_1 \cdots e_s} = \varnothing$. Otherwise, the output file tells you that $\mathcal{C}$ is not chordal.

Example 2.1. Let $\mathcal{C}$ be the 3-uniform clutter as in Example 1.3. After running `ChordalityCheck.cpp` on this clutter we see that $\mathcal{C} \in \mathfrak{C}_3$. Here is the contents of the output file:

Input is chordal.
e[1]= 1 2
e[2]= 1 3
e[3]= 2 4
e[4]= 2 3
e[5]= 2 5
e[6]= 2 6
e[7]= 5 6
Running time : 0 Seconds

One may check that e_1 is a simiplicial submaximal circuit of $\mathcal{C}$ and e_i is a simplicial submaximal circuit of $\mathcal{C}_{e_1 \cdots e_{i-1}}$, for $i = 2, \ldots, 7$.

References

[1] The source code for detecting chordality. Available at:
www.iasbs.ac.ir/∼yazdan/chordalitycheck.cpp

[2] M. Bigdeli, A. A. Yazdan Pour and R. Zaare-Nahandi, *Stability of Betti Numbers under Reduction Processes: Toward Chordality of Clutters*, preprint, (2015). arXiv:1508.03799v1

[3] G. A. Dirac, *On rigid circuit graphs*, Abh. Math. Sem. Univ. Hamburg, **38**, pp. 71–76, (1961).

[4] M. Morales, A. Nasrollah Nejad, A. A. Yazdan Pour and R. Zaare-Nahandi, *Monomial ideals with 3-linear resolutions*, Annales de la Faculté des Sciences de Toulouse, Sér. **6**, 23: (4), pp 877–891, (2014). arXiv:1207.1790v1

DEPARTMENT OF MATHEMATICS, INSTITUTE FOR ADVANCED STUDIES IN BASIC SCIENCES (IASBS), P.O.BOX 45195-1159, ZANJAN, IRAN

E-mail address: `yazdan@iasbs.ac.ir`

[1]It is recommended to use GCC 4.8.2 or higher version for compiling this program