A Polymake Extension for Matroid Homology

Austin Alderete

Abstract

We introduce a new polymake extension designed to perform computations in the intersection ring of matroids. Matroid addition is carried out on the indicator vectors of maximal chains of flats and matroid product is the usual matroid intersection where the product is defined to be zero if it is not loopfree. The primary feature of this extension is the ability to compute the homology groups of the intersection ring induced by deletion and contraction as well as those of any subring generated by a minor-closed of matroids.

1 Introduction

There's been a growing interest in defining algebraic structures on the set of all matroids, of note are the intersection ring of matroids whose theory was developed by Simon Hampe and the matroid-minor Hopf algebra introduced by Crapo and Schmitt [2] [1]. The former arises out of tropical geometry and uses the structure of tropical linear spaces to gain insight into matroids. For example, it was shown that the $\mathcal{G}$-invariant induces a $\mathbb{Z}$-module homomorphism on the intersection ring. Likewise, the matroid operations of contraction and deletion give rise to a pair of $\mathbb{Z}$-module homomorphisms between intersection rings. When taking the direct sum over all intersection rings one can define a pair of boundary maps using alternating sums of contractions or deletions. This is the matroid generalization of Kontsevich graph homology [**?**].

Our polymake extension provides the tools for calculations in the intersection ring, but more importantly, allows the user to specify any minor-closed class of matroids recognized by the polyDB database and compute the homology groups associated to either boundary map.

2 Preliminaries: matroid theory and the intersection ring

For an introduction to matroid theory, we recommend Oxley [**?**].

Definition 1 (Intersection ring of matroids)**.** For $1 \leq r \leq n \in \mathbb{N}$, let $\mathfrak{C}_{r,n}$ be the set of all proper chains of length r of the form

$$\emptyset = F_0 \subset F_1 \subset ... \subset F_r = E.$$

Define $V_{r,n} = \mathbb{Z}^{\mathfrak{C}_{r,n}}$, the free $\mathbb{Z}$-module whose coordinates are indexed by elements of $\mathfrak{C}_{r,n}$.

Let $\mathcal{M}$ be a set of matroids. We define a set map as follows:

$$\Phi_{r,n} : \mathcal{M} \to V_{r,n} \quad ; \quad M \mapsto v_M$$

where v_M is defined coordinate-wise by

$$(v_M)_C := \begin{cases} 1, & \text{if } C \text{ is a chain of flats in } M \\ 0, & \text{otherwise} \end{cases}.$$

We call $\Phi_{r,n}$ and any extensions of it, the *indicator map*.

When $\mathcal{M}$ is a minor-closed class, we let $\mathbb{M}_{r,n}^{\text{free}}$ be the free $\mathbb{Z}$-module with generators the set of all loopfree elements of $\mathcal{M}$ of rank r on $[n]$. Then we can extend the indicator map to a $\mathbb{Z}$-module homomorphism by granting it linearity. *The intersection ring* of $\mathcal{M}$ is the $\mathbb{Z}$-module

$$\mathbb{M}_n = \bigoplus_{r=1}^{n} \mathbb{M}_{r,n}$$

with $\mathbb{M}_{r,n} = \mathbb{M}^{\text{free}}_{r,n}/\ker(\Phi_{r,n})$.

This is the *intersection ring of* $\mathcal{M}$.

Often we take $\mathcal{M}$ to be the set of all matroids, but for the example in the sequel we will take the uniform matroids. By taking $\bigoplus_n \mathbb{M}_n$ we form a bigraded algebraic structure.

There are two sets of $\mathbb{Z}$-module homomorphisms on $\mathbb{M}$ of interest:

$$d_i \mathbb{M}_{r,n} \to \mathbb{M}_{r,n-1} \quad ; \quad d_i(m) := \begin{cases} m\backslash i, & \text{if } i \text{ is not a coloop of } m \\ 0, & \text{otherwise} \end{cases}$$

$$c_i \mathbb{M}_{r,n} \to \mathbb{M}_{r-1,n-1} \quad ; \quad c_i(m) := \begin{cases} m/i, & \text{if } \mathrm{cl}_m(\{i\}) = \{i\} \\ 0, & \text{otherwise} \end{cases}.$$

These are the single element deletion and contraction matroid operations respectively. One then defines differentials of $\mathbb{M}$ ∂_d and ∂_c by taking alternating sums, e.g. $\partial_d = \sum_i (-1)^i d_i$.

The ring multiplication of $\mathcal{M}$ is

$$M \cdot N := \begin{cases} M \wedge N = (M^* \cup N^*)^*, & \text{if } M \wedge N \text{ is loopfree,} \\ 0, & \text{otherwise.} \end{cases}$$

Both the ring multiplication and the differentials above have a tropical meaning when considering matroid fans.

Conjecture 1 (Hampe)**.** *If* $\mathbb{M} = \bigoplus_n \mathbb{M}_n$ *where* $\mathbb{M}_n$ *is the intersection ring of all matroids on* $[n]$ *and* ∂_d, ∂_c *are the boundary maps induced by deletion and contraction respectively, then* $H^{\partial_d}_{r,n}(\mathbb{M}) = H^{\partial_c}_{r,n}(\mathbb{M}) = \mathbf{0}$ *for all* $r \leq n$*, where we define*

$$H^{\partial_c}_{r,n}(\mathbb{M}) := \ker\left(\partial_c : \mathbb{M}_{r,n} \to \mathbb{M}_{r-1,n-1}\right) / \mathrm{Im}\left(\partial_c : \mathbb{M}_{r+1,n+1} \to \mathbb{M}_{r,n}\right),$$

$$H^{\partial_d}_{r,n}(\mathbb{M}) := \ker\left(\partial_d : \mathbb{M}_{r,n} \to \mathbb{M}_{r,n-1}\right) / \mathrm{Im}\left(\partial_d : \mathbb{M}_{r,n+1} \to \mathbb{M}_{r,n}\right),$$

and $\mathbb{M}_{r,n}$ *is the component of* $\mathbb{M}$ *generated by rank* r *matroids on* $[n]$.

As an introduction to this problem, we compute the matroid homology of $\mathbb{U} = \mathbb{U}_n$, where $\mathbb{U}_n$ is the intersection ring of uniform matroids. We first note that $\mathbb{U}_{r,n} \cong \mathbb{Z}$ with generator $[U^r_n]$. Because of this it's enough to consider $\partial_d(U^r_n)$ and $\partial_c(U^r_n)$. We have

$$\partial_d(U^r_n) = \sum_{i\in[n]} (-1)^i d_i(U^r_n)$$

and since i is never a coloop of U^r_n so long as $r < n$ and $U^r_n \backslash i = U^r_{n-1}$,

$$\partial_d(U^r_n) = \sum_{i\in[n]} (-1)^i U^r_{n-1} = \begin{cases} -U^r_{n-1} & n \text{ is odd} \\ 0 & n \text{ is even} \end{cases}$$

Therefore, writing $n = 2k, 2k+1$ for some $k \in \mathbb{Z}$ respectively, we have

$$\ker(\partial_d : \mathbb{U}_{r,2k} \to \mathbb{U}_{r,2k-1}) = \langle U^r_{2k} \rangle \quad , \quad \ker(\partial_d : \mathbb{U}_{r,2k+1} \to \mathbb{U}_{r,2k}) = 0$$

and

$$\mathrm{Im}(\partial_d : \mathbb{U}_{r,2k+1} \to \mathbb{U}_{r,2k}) = \langle \partial_d(U^r_{2k+1}) \rangle = \langle U^r_{2k} \rangle \quad , \quad \mathrm{Im}(\partial_d : \mathbb{U}_{r,2k+2} \to \mathbb{U}_{r,2k+1}) = 0$$

which yields

$$H^{\partial_d}_{r,n}(\mathbb{U}) = 0.$$

Meanwhile, for

$$H^{\partial_c}_{r,n}(\mathbb{U}) = \ker(\partial_c : \mathbb{U}_{r,n} \to \mathbb{U}_{r-1,n-1})/\mathrm{Im}(\partial_c : \mathbb{U}_{r+1,n+1} \to \mathbb{U}_{r,n})$$

we note that

$$\partial_c(U^r_n) = \sum_{i\in[n]} (-1)^i c_i(U^r_n)$$

and that $\mathrm{cl}(\{i\}) = \{i\}$ for all i and $U_n^r/i = U_{n-1}^{r-1}$. So we have

$$\partial_c(U_n^r) = \sum_{i\in[n]} (-1)^i U_{n-1}^{r-1}.$$

and so conclude

$$H_{r,n}^{\partial_c}(\mathbb{U}) = 0.$$

An unusual property of the intersection ring of uniform matroids is that $\mathbb{U}_{r,n} = \mathbb{U}_{r,n}^{\mathrm{free}}$. However, in the more general case the indicator map induces relationships between matroids. These relationships can be analyzed using the polymake extension.

3 Polymake implementation

Everything that follows is done within the matroid enviornment of polymake. As the number of chains of length r from $\emptyset$ to $[n]$ increases combinatorially, performing any calculation by hand using the vector representations of elements of $\mathbb{M}_{r,n}$ becomes unfeasible. The first goal of the extension therefore is to handle the representations for us. The extension calculates the indices of $V_{r,n}$ each time when computing the indicator vector of a single matroid in the interactive shell and only once when the matroid homology script is called. The indices are generated using a depth-first. There are two ways of accessing the indicator vector: as a property of the matroid or via a function.

polymake example: obtaining indicator vector of matroid

```
matroid > $m = uniform_matroid(2,4);
matroid > print indicator_vector($m);
1 1 0 1 0 0 0 1 0 0 0 0 0 0
matroid > $v = $m->INDICATOR_VECTOR;
matroid > print $v;
1 1 0 1 0 0 0 1 0 0 0 0 0 0
```

While polymake already has the capability to perform matroid intersection, we've added to this by setting the product equal to the uniform matroid of rank 0 on $[n]$ whenever the intersection contains a loop. This is only used for ease of identification. Matroid addition takes in an array of matroids as well as a vector of coefficients and makes no assumption on the matroids having the same rank or being loopfree. However, the first matroid in the array must be loopfree and determines the dimension of the vector. We demonstrate this using an example found in [2]. Given the four matroids of rank 2 on $[n]$ defined by their flats:

$$\begin{aligned}
\mathcal{F}(m_1) &:= \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, [n]\},\\
\mathcal{F}(m_2) &:= \{\emptyset, \{1,4\}, \{2,3\}, [n]\},\\
\mathcal{F}(m_3) &:= \{\emptyset, \{1,4\}, \{2\}, \{3\}, [n]\},\\
\mathcal{F}(m_4) &:= \{\emptyset, \{1\}, \{2,3\}, \{4\}, [n]\},
\end{aligned}$$

one has $m_1 + m_2 = m_3 + m_4$ in $\mathbb{M}_{2,4}$.

It is important to note that the arguments of matroid addition are `Array<Matroid>` and `Vector<Int>` while those of matroid product are simply a pair of matroids.

The next pair of operations are derivative and are merely combinations of existing functions within polymake with the indicator vector function in this extension. We have `contraction_vector` and `deletion_vector`. They output the vector representing the image of ∂_c or ∂_d respectively. As polymake begins indexing with 0, the alternating sum begins with a positive term.

The largest feature of this extension is a script which allows for matroid homology to be computed. It makes use of the polyDB database and allows the user to select any class of matroids recognized by the database: laminar, paving, binary, ternary, regular, nested, etc. There are a few restrictions on

polymake example: matroid operations in the intersection ring

```
matroid > $m1 = new Matroid(BASES=>[[0,1],[0,2],[0,3],[1,2],[1,3],[2,3]],N_ELEMENTS=>4);
matroid > $m2 = new Matroid(BASES=>[[0,1],[0,2],[1,3],[2,3]],N_ELEMENTS=>4);
matroid > $m3 = new Matroid(BASES=>[[0,1],[0,2],[1,3],[2,3],[1,2]],N_ELEMENTS=>4);
matroid > $m4 = new Matroid(BASES=>[[0,1],[0,2],[0,3],[1,3],[2,3]],N_ELEMENTS=>4);
matroid > $arrMat = new Array<Matroid>($m1,$m2,$m3,$m4);
matroid > $vecCoef = new Vector<Int>(1,1,-1,-1);
matroid > print matroid_addition($arrMat,$vecCoef);
0 0 0 0 0 0 0 0 0 0 0 0 0 0
matroid >
matroid > $fano = fano_matroid;
matroid > $uni = uniform_matroid(6,7);
matroid > $fanoProdUni = matroid_product($fano,$uni);
matroid > print $fanoProdUni->RANK;
2
matroid > print $fanoProdUni->INDICATOR_VECTOR;
1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

polymake example: contraction and deletion differentials

```
matroid > $m = new Matroid(REVLEX_BASIS_ENCODING=>"000*******",N_ELEMENTS=>5,RANK=>2);
matroid > print deletion_vector($m);
1 1 0 0 0 0 0 0 0 0 0 1 0 0
matroid >
matroid > $vDel0 = deletion($m,0)->INDICATOR_VECTOR;
matroid > $vDel1 = deletion($m,1)->INDICATOR_VECTOR;
matroid > $vDel2 = deletion($m,2)->INDICATOR_VECTOR;
matroid > $vDel3 = deletion($m,3)->INDICATOR_VECTOR;
matroid > $vDel4 = deletion($m,4)->INDICATOR_VECTOR;
matroid > print $vDel0-$vDel1+$vDel2-$vDel3+$vDel4;
1 1 0 0 0 0 0 0 0 0 0 1 0 0
matroid > print contraction_vector($m);
0
```

use, however. The first is that the class be minor-closed. This is a theoretical requirement as otherwise there is no guarantee that the homology groups are well-defined. If $r > n/2$, then the class must be also dual-closed. While not mathematically necessary, the way by which the extension accesses matroids with large rank requires the use of matroid duals. If the class of matroids selected is empty, e.g. when the class is not recognized by the database, then the script will print a message and abort. It should also be noted that the intersection ring is generated by the nested matroids, in fact they form a basis of the free $\mathbb{Z}$-module $\mathbb{M}_{r,n}^{\text{free}}$. If the class contains the nested matroid, it is recommended to input just the nested matroids. For this reason the script defaults to the nested matroids if the class of all matroids is considered. The script itself only computes the ambient dimension, the dimension of the kernel in $\mathbb{M}_{r,n}$ and the dimension of the image of $\mathbb{M}_{r+*,n+*}$ in $\mathbb{M}_{r,n}$. Running the script twice on successive (r, n) pairs allows one to compare the dimensions of the kernel and image involved in the homology group. The preferred script to run is matroid_homology_fast.pl which uses the C++ code for speedy computations. It takes advantage of an S_n-representation which permutes indicator vectors when computing $\mathbb{M}_{r,n}$.

polymake example: matroid homology

```
matroid > $m = new Matroid(REVLEX_BASIS_ENCODING=>"000*******",N_ELEMENTS=>5,RANK=>2);
matroid > print deletion_vector($m);
1 1 0 0 0 0 0 0 0 0 0 1 0 0
matroid >
matroid > $vDel0 = deletion($m,0)->INDICATOR_VECTOR;
matroid > $vDel1 = deletion($m,1)->INDICATOR_VECTOR;
matroid > $vDel2 = deletion($m,2)->INDICATOR_VECTOR;
matroid > $vDel3 = deletion($m,3)->INDICATOR_VECTOR;
matroid > $vDel4 = deletion($m,4)->INDICATOR_VECTOR;
matroid > print $vDel0-$vDel1+$vDel2-$vDel3+$vDel4;
1 1 0 0 0 0 0 0 0 0 0 1 0 0
matroid > print contraction_vector($m);
0
```

4 To-do:

The following are works-in progress.

- Currently the script operates on $\mathbb{M}_{r,n} \otimes \mathbb{R}$. This means that quotients such as $\langle[1, 0, 0]\rangle/\langle[2, 0, 0]\rangle$ are considered as trivial when in fact they represent nontrivial homology groups (such as $\mathbb{Z}/\mathbb{Z}/2$) here. In other words, the script can only determine the free rank of the homology groups. If they are torsion groups then no information is gleaned. To correct for this an integral version of the code which computes a lattice basis for the kernel and image is in progress.
- Extend the permutation action on indicator vectors to the contraction and deletion cases.
- Replace S_n with the respective automorphism group of each matroid.
- Allow for polynomials in matroids to be evaluated without having to nest functions.

References

[1] H. Crapo and W. Schmitt. Primitive elements in the matroid-minor hopf algebra. 2005.

[2] S. Hampe. Intersection ring of matroids. *Journal of Combinatorial Theory Series B*, 122:578–614, 2017.