02_poliedros

June 24, 2021

1 Uso de polymake: Trabajando con poliedros

1.1 Representación de $\mathcal{V}$-poliedros

Internamente, `polymake` representa a los poliedros a través de sus conos de homogeneización. Conocemos que si $P = \operatorname{conv}(V) + \operatorname{cone}(Y) \subset \mathbb{R}^d$, entonces el cono de homogeneización de P está dado por:

$$\operatorname{homog}(P) := \operatorname{cone}\begin{pmatrix} \mathbf{1}^T & \mathbf{0}^T \\ V & Y \end{pmatrix} \subset \mathbb{R}^{d+1}.$$

Los vectores (o puntos) en $\mathbb{R}^{d+1}$ para generar $\operatorname{homog}(P)$ se especifican por medio de la propiedad `POINTS`.

Por ejemplo, definimos a continuación `$p` como la envolvente convexa del conjunto $\{(-1,-1),(-1,1),(1,-1),(1,1),(0,0)\}$:

[35]:
```
$p=new Polytope(POINTS=>[[1,-1,-1],[1,-1,1],[1,1,-1],[1,1,1],[1,0,0]]);
### mostrar las propiedades de $p
$p->properties;
```

[35]:
```
name: p
type: Polytope<Rational>

CONE_AMBIENT_DIM
3

POINTS
1 -1 -1
1 -1 1
1 1 -1
1 1 1
1 0 0
```

Podemos obtener una representación gráfica de `$p` llamando al método `VISUAL`.

[36]:
```
### mostrar una representación gráfica de $p
$p->VISUAL;
```

Consultando la propiedad `VERTICES` podemos determinar cuáles son los puntos no redundantes requeridos para definir el polítopo (es decir, sus vértices):

[37]:
```
print $p->VERTICES;
```

[37]:
```
1 -1 -1
1 -1 1
1 1 -1
1 1 1
```

La propiedad `DIM` nos devuelve la dimensión del polítopo:

[38]:
```
print $p->DIM;
```

[38]:
```
2
```

La propiedad `FACETS` nos devuelve las desigualdades que definen las facetas del polítopo. Al consultarla, `polymake` invoca internamente un algoritmo tipo Fourier-Motzkin para transformar la representación $\mathcal{V}$ en la representación $\mathcal{H}$ del polítopo, a partir del cálculo de la envolvente convexa. Con la instrucción `prefer` podemos requerir específicamente que se utilice alguno de los métodos disponibles. Por ejemplo, en el siguiente fragmento de código se emplea el algoritmo de búsqueda en reversa revisado `lrs` desarrollado por David Avis:

[39]:
```
prefer "lrs";
print_constraints($p->FACETS);
```

[39]:
```
0: -x1 >= -1
1: -x2 >= -1
2: x2 >= -1
3: x1 >= -1
```

Con la propiedad `VERTICES_IN_FACETS` consultamos la incidencia entre vértices y facetas:

[40]:
```
print ($p->VERTICES_IN_FACETS);
```

[40]:
```
{2 3}
{1 3}
{0 2}
{0 1}
```

Agregaremos ahora una parte cónica al poliedro anterior. Según la representación del cono de homogeinización, esto se consigue añadiendo un punto (o vector) con la primera coordenada igual a cero al especificar la propiedad `POINTS`.

En el ejemplo siguiente definimos a `$q` como $\text{conv}(\{(-1,-1),(-1,1),(1,-1),(1,1),(0,0)\}) + \text{cone}(\{(1,1)\})$

[41]:
```
$q = new␣
 ↪Polytope(POINTS=>[[1,-1,-1],[1,-1,1],[1,1,-1],[1,1,1],[1,0,0],[0,1,1]]);
$q -> VISUAL;
```

Al consultar la propiedad `VERTICES` podemos ver que el nuevo poliedro tiene tres vértices (puntos no redundantes en la combinación convexa) y un rayo (vector no redundante en la combinación cónica). Los vértices tienen la primera coordenada igual a 1, los rayos tienen la primera coordenada igual a 0:

[42]:
```
print($q->VERTICES);
```

[42]:
```
1 -1 -1
1 -1 1
1 1 -1
0 1 1
```

Podemos consultar también las desigualdades que definen las facetas de `$q`, así como la incidencia de facetas en vértices:

[43]:
```
### mostrar facetas de $q
print_constraints($q->FACETS);
print("---\n");

### mostrar incidencia de vértices en facetas
print ($q->VERTICES_IN_FACETS);
```

[43]:
```
0: x2 >= -1
1: x1 >= -1
2: -1/2 x1 + 1/2 x2 >= -1
3: 1/2 x1 - 1/2 x2 >= -1

---
{0 2}
{0 1}
{2 3}
{1 3}
```

1.2 Representación de $\mathcal{H}$-poliedros

Un $\mathcal{H}$-poliedro de la forma $P := \{x \in \mathbb{R}^d : Ax \leq b\}$ se representa a través se su cono de homogeinización, dado por el sistema de desigualdades:

$$\begin{pmatrix} 1 & \mathbf{0}^T \\ b & -A \end{pmatrix} \begin{pmatrix} x_0 \\ x \end{pmatrix} \geq \mathbf{0}.$$

Las filas de la matriz de coeficientes se especifican por medio de la propiedad `INEQUALITIES`; no es necesario incluir la primera fila, la cual es añadida automáticamente por polymake.

Por ejemplo, el siguiente fragmento de código asigna a la variable `$r` el poliedro en $\mathbb{R}^2$ definido por el sistema de desigualdades:

$$\begin{cases} x_1 + x_2 \leq 4 \\ x_1 + 2x_2 \leq 7 \\ 0 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq 3 \end{cases}$$

[44]:
```
$r = new Polytope(INEQUALITIES=>[[4,-1,-1],[7,-1,-2],[2,-1,0],[3,0,-1],[0,1,0],
↪[0,0,1]]);
### mostrar las propiedades de $r
$r->properties;
```

[44]:
```
name: r
type: Polytope<Rational>

CONE_AMBIENT_DIM
3

INEQUALITIES
4 -1 -1
7 -1 -2
2 -1 0
3 0 -1
0 1 0
0 0 1
1 0 0
```

Notar que polymake añadió automáticamente la fila `[1, 0, 0]`, correspondiente a la desigualdad $x_0 \geq 0$.

La función `print_constraints` escribe el sistema de desigualdades en un formato más amigable para el usuario. Notar que esta función deshace además la homogeneización. En particular, la nonegatividad de x_0 se transforma en la desigualdad trivial $0 \geq -1$.

[45]:
```
print_constraints($r->INEQUALITIES);
```

[45]:
```
0: -x1 - x2 >= -4
1: -x1 - 2 x2 >= -7
2: -x1 >= -2
3: -x2 >= -3
4: x1 >= 0
5: x2 >= 0
6: 0 >= -1
```

Consultando la propiedad `FACETS` obtenemos las desigualdades no redundantes del sistema. En el ejemplo anterior, la segunda desigualdad no es listada, porque la misma se obtiene como suma de la primera y la cuarta desigualdades. De igual forma, la desigualdad trivial $0 \geq -1$ es eliminada del sistema.

[46]:
```
print_constraints($r->FACETS);
```

[46]:
```
0: -x1 - x2 >= -4
1: -x1 >= -2
2: -x2 >= -3
3: x1 >= 0
4: x2 >= 0
```

Con el método `VISUAL` obtenemos una representación gráfica de `$r`:

[47]:
```
$r->VISUAL;
```

La propiedad `VERTICES` nos indica los vértices y rayos necesarios para expresar al polítopo en la forma $\mathcal{V}$. Al consultar esta propiedad, `polymake`invoca automáticamente un algoritmo tipo Fourier-Motzkin para cambiar la representación del poliedro mediante la enumeración de sus vértices.

Recordar que los vértices tienen la primera coordenada igual a 1, mientras que los rayos tienen la primera coordenada igual a 0.

[48]:
```
### mostrar vértices y rayos de $r
print($r->VERTICES);
```

[48]:
```
1 0 0
1 2 0
1 2 2
1 0 3
1 1 3
```

1.3 Sumas de Minkowski

Definamos al polítopo `$p` como la envolvente convexa del conjunto de puntos $\{(1,1),(2,1),(1,2)\}$:

[49]:
```
$p=new Polytope(POINTS=>[[1,1,1],[1,2,1],[1,1,2]]);
$p->VISUAL;
```

Definamos ahora a `$q` como el cono generado por los vectores $\binom{10}{9}$ y $\binom{9}{10}$. Notar que es necesario indicar explícitamente el vértice del cono entre los elementos del parámetro `POINTS` al llamar al constructor `Polytope`. Esto es requerido para obtener un cono de homogeneización de dimensión completa que pueda ser intersecado con el hiperplano $x_0 = 1$ para recuperar nuestro poliedro original.

[50]:
```
$q=new Polytope(POINTS=>[[1,0,0],[0,10,9],[0,9,10]]);
$q->VISUAL;
```

Para calcular la suma de Minkowski entre `$p` y `$q` podemos emplear la función `minkowski_sum`, la misma que retorna como resultado un nuevo poliedro:

[51]:
```
$m = minkowski_sum($p, $q);
$m -> VISUAL;
```

Podemos ahora consultar las diferentes propiedades del nuevo poliedro creado:

[52]:
```
print ($m->VERTICES);
```

[52]:
```
1 1 1
1 2 1
1 1 2
0 1 9/10
0 1 10/9
```

[53]:
```
print_constraints($m->FACETS);
```

[53]:
```
0: x1 >= 1
1: x2 >= 1
2: -9/8 x1 + 5/4 x2 >= -1
3: 5/4 x1 - 9/8 x2 >= -1
4: 0 >= -1
```

[54]:
```
print($m->VERTICES_IN_FACETS);
```

[54]:
```
{0 2}
{0 1}
{1 3}
{2 4}
{3 4}
```

1.4 Conos de recesión

Dado un poliedro, `polymake` puede emplearse para calcular su cono de recesión. Considerar el último poliedro `$m`que hemos definido:

[55]:
```
$m->VISUAL;
```

Para recuperar el cono de recesión de `$m` llamamos a la función `recession_cone`:

[56]:
```
$q2= recession_cone($m);
$q2 -> VISUAL;
```

Al contrario de lo que ocurre con los poliedros en general, este cono se representa sin homogeneización. La propiedad `RAYS` permite consultar los vectores que lo generan:

[57]:
```
print($q2->RAYS);
```

[57]:
```
1 9/10
1 10/9
```

1.5 Ejercicio

1. Definamos un polítopo `$h` como un hexágono incrustado en el hiperplano $x_3 = 0$ en $\mathbb{R}^3$:

[58]:
```
### $h := conv({(1,0,0), (1/2,2/3,0), (-1/2,2/3,0), (-1,0,0), (1/2,-2/3,0), (-1/
↪2,-2/3,0)})
$h=new Polytope(POINTS=>[[1,1,0,0],[1,1/2,2/3,0],[1,-1/2,2/3,0],[1,-1,0,0],
[1,1/2,-2/3,0],[1,-1/2,-2/3,0]]);
$h->VISUAL;
```

Definamos ahora `$c` como el cono tridimensional generado por los vectores $\begin{pmatrix} 0 \\ 1 \\ 10 \end{pmatrix}$, $\begin{pmatrix} 2/3 \\ -1/2 \\ 10 \end{pmatrix}$ y $\begin{pmatrix} -2/3 \\ -1/2 \\ 10 \end{pmatrix}$:

[59]:
```
### $c:= cone({(0,1,10), (2/3,-1/2,10), (-2/3,-1/2,10)})
$c=new Polytope(POINTS=>[[1,0,0,0],[0,0,1,10],[0,2/3,-1/2,10],[0,-2/3,-1/
↪2,10]]);
$c->VISUAL;
```

Definamos `$p` como la suma de Minkowski de `$h` y `$c`:

[60]:
```
$p=minkowski_sum($h,$c);
$p->VISUAL;
```

Consultemos las desigualdades que definen las facetas de `$p`:

[61]:
```
print_constraints($p->FACETS);
```

[61]:
```
0: -3/2 x2 + 3/20 x3 >= -1
1: x3 >= 0
2: x1 - 3/4 x2 + 3/40 x3 >= -1
3: -x1 - 3/4 x2 + 3/40 x3 >= -1
```

```
4: -x1 + 3/4 x2 + 5/48 x3 >= -1
5: x1 + 3/4 x2 + 5/48 x3 >= -1
6: -x1 - 4/9 x2 + 2/45 x3 >= -1
7: x1 - 4/9 x2 + 2/45 x3 >= -1
8: 3/2 x2 + 3/40 x3 >= -1
9: 0 >= -1
```

Consultemos los vértices y rayos de `$p`:

[62]:
```
print($p->VERTICES);
```

[62]:
```
1 1 0 0
1 1/2 2/3 0
1 -1/2 2/3 0
1 -1 0 0
1 1/2 -2/3 0
1 -1/2 -2/3 0
0 0 1 10
0 1 -3/4 15
0 -1 -3/4 15
```

Consultemos las incidencias entre los vértices, rayos y facetas de `$p`:

[63]:
```
print($p->VERTICES_IN_FACETS);
```

[63]:
```
{1 2 6}
{0 1 2 3 4 5}
{2 3 6}
{0 1 6}
{0 4 7}
{3 5 8}
{0 6 7}
{3 6 8}
{4 5 7 8}
{6 7 8}
```

Recuperemos el cono de recesión de `$p`:

[64]:
```
$c2=recession_cone($p);
$c2->VISUAL;
```

[ ]:
```

```