Lundi 7 février 2005 à 17 h

Logique et géométrie

Au milieu du XIX^e siècle, le mathématicien George Boole s'intéresse aux conditions logiques que doit satisfaire une expérience dans laquelle interviennent des probabilités. Ces conditions dépendent des fréquences relatives d'événements liés et s'expriment à l'aide d'inégalités. Pitowsky donne en 1989 une interprétation géométrique de ces inégalités. Considérons une urne contenant des boules de différentes couleurs et façonnées dans diverses matières. Admettons que chaque boule peut être caractérisée par deux propriétés, disons «blanche» et «en bois». Ainsi, une boule décrite est blanche, en bois, ou blanche et en bois. L'état de l'urne est donné par les probabilités associées au tirage d'une boule ayant une de ces propriétés. Soit p_1 la proportion de boules blanches dans l'urne, p_2 celle de boules en bois et p_{12} celle de boules blanches et en bois. Si l'urne contient suffisamment de boules, ces proportions sont les probabilités d'obtenir par tirage une boule ayant une de ces propriétés. Les inégalités :

$$0 \leq p_{12} \leq p_1 \leq 1 \qquad 0 \leq p_{12} \leq p_2 \leq 1 \qquad (1)$$

sont vérifiées par les proportions p_1, p_2 et p_{12} qui peuvent être considérées comme les probabilités de chaque événement et de leur réunion seulement si ces inégalités sont satisfaites. Mais ces inégalités sont insuffisantes : la probabilité de tirer une boule qui est, soit blanche soit en bois ($p_1 + p_2 - p_{12}$), doit être comprise entre 0 et 1, ce qui fournit une autre inégalité :

$$0 \leq p_1 + p_2 - p_{12} \leq 1 \qquad (2)$$

Les trois inégalités ci-dessus sont nécessaires et suffisantes pour que les nombres p_1, p_2 et p_{12} représentent dans l'ordre les probabilités de tirer de l'urne une boule blanche, une boule en bois, une boule blanche qui est en bois.
Formons la table de vérité dans laquelle p_1, p_2 et $p_1 \wedge p_2$ représentent respectivement les propositions :
« la balle tirée est blanche » « la balle tirée est en bois » « la balle tirée est blanche et elle est en bois ».

p_1	p_2	$p_1 \wedge p_2$
Vrai	Vrai	Vrai
Vrai	Faux	Faux
Faux	Vrai	Faux
Faux	Faux	Faux

p_1	p_2	$p_1 \wedge p_2$
1	1	1
1	0	0
0	1	0
0	0	0

Associons 1 à « Vrai » et 0 à « Faux » et considérons chaque ligne comme les composantes d'un vecteur. Ces vecteurs définissent un polyèdre convexe dont les sommets (points extrêmes) s'interprètent ainsi : (1, 1, 1) l'urne ne contient que des boules blanches en bois. (1, 0, 0) l'urne ne contient que des boules blanches qui ne sont pas en bois. (0, 1, 0) l'urne ne contient que des boules en bois qui ne sont pas blanches. (0, 0, 0) aucune boule blanche ni aucune boule en bois ne se trouve dans l'urne. Les inégalités décrivant l'urne peuvent alors être associées aux plans contenant les faces du polyèdre. Dans ce modèle très simple de l'urne qui ne fait intervenir que deux propriétés, les inégalités (1) et (2) peuvent être devinées facilement. Ce n'est plus possible dès qu'il y a davantage de propriétés, mais on peut trouver toutes les inégalités à partir des sommets d'un polyèdre convexe. Ce problème, connu sous le nom de problème de la frontière (hull problem), peut être résolu algorithmiquement.

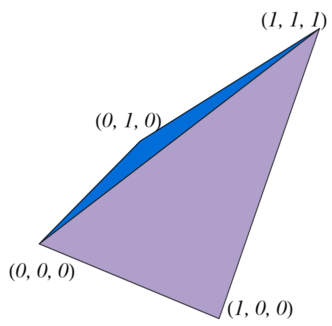


Fig. 1 : Polyèdre construit à partir d'une table de vérité comportant deux propositions et leur réunion.

Prochaine réunion : lundi 7 mars 2005 à 17 h

■ Travaux pratiques

■ Exercice 1

a) Construisez les tables de vérité des opérations logiques And, Nand, Nor, Or, Xor pour 2 propositions.
b) Dessinez les polytopes (on appelle polytope un polyèdre borné) associés à ces tables.

■ Exercice 2

a) Dessinez la frontière convexe d'un ensemble quelconque de points du plan.
b) Effectuez une triangulation de Delaunay de cet ensemble de points (une triangulation de Delaunay d'un ensemble de points est une triangulation des points de l'ensemble telle que les cercles circonscrits aux triangles ne contiennent aucun point de l'ensemble).
N. B. La solution de ce problème est utile par exemple pour déterminer la longueur minimale d'une clôture. L'aire délimitée par la clôture de longueur minimale qui englobe tous les points est convexe : la ligne reliant deux points intérieurs quelconques est toujours dans l'enclos.

■ Exercice 3

Une urne contient des boules de différentes couleurs et façonnées dans diverses matières. On utilise, pour décrire les boules, 3 propriétés : « blanche », « lisse » et « en bois ». Désignons par p_1 la proportion de boules blanches dans l'urne, par p_2 celle de boules lisses, par p_3 celle de boules en bois, par p_{12} celle de boules blanches et lisses, par p_{13} celle de boules blanches en bois, par p_{23} celle de boules lisses en bois et par p_{123} celle de boules blanches lisses et en bois. Quelles conditions ces proportions doivent-elles satisfaire pour donner les probabilités d'obtenir une boule ayant les propriétés correspondantes lors d'un tirage au sort ?

Programmes à installer (disponibles sur le web) pour résoudre l'exercice 3
Komei Fukuda (ETHZ, Zurich et EPFL, Lausanne) a développé depuis 1996 un algorithme permettant de générer le système linéaire d'inégalités décrivant les faces du polytope à partir de ses sommets et d'obtenir, réciproquement, tous les sommets d'un polytope convexe à partir du système linéaire d'inéquations représentant ses faces. Cet algorithme, écrit en C++, implémente la méthode de la double description, d'où son nom « cdd ».
Il existe un module *Mathematica* (CddIF.m), écrit par Stefan Filipp en 2001, qui permet d'exécuter le code binaire (cddmathlink2) de l'algorithme de la double méthode via *MathLink* et de résoudre l'exercice 3.

■ Pour en savoir plus

• Pour obtenir la frontière convexe d'un ensemble de points si vous ne disposez pas de *Mathematica* http://128.32.125.151/riot/Applications/ConvexHull/CHDetails.html
• George Boole, The Calculus of Logic. Cambridge and Dublin. *Mathematical Journal*, Vol. III (1848), pp. 183-98. Disponible à http://www.maths.tcd.ie/pub/HistMath/People/Boole/CalcLogic/CalcLogic.html
• I. Pitowsky. Correlation polytopes: Their geometry and complexity. *Mathmatical Programming*, 50: 395–414, 1991. Disponible sur http://hypatie.ge.ch
• Komei Fukuda, *Lecture Notes on Oriented Matroids and Geometric Computation*, Winter 2000 (revised June 16, 2004). Disponible sur http://hypatie.ge.ch
• Stefan Filipp and Karl Svozil, *Boole-Bell-type inequalities in Mathematica*, 17 may 2001.
Disponible sur http://hypatie.ge.ch

Logique et géométrie

■ Exercice 1

a) Tables de vérité pour les opérateurs « And », « Nand », « Nor », « Or », « Xor » pour deux propositions.
La fonction **TruthTable1** est due à Eric W. Weisstein. From *MathWorld*--A Wolfram Web Resource.

```
TruthTable1[op_, n_] := Module[
  {
   l = Flatten[Outer[List, Sequence @@ Table[{True, False}, {n}]], n - 1],
   a = Array[a, n]
  },
  DisplayForm[
   GridBox[Prepend[Append[#, op @@ #] & /@ l, Append[a, op @@ a]],
    RowLines -> True, ColumnLines -> True]
  ]
 ]
```

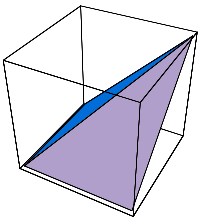
```
TruthTable1[And, 2] /. {True → 1, False → 0} // TraditionalForm
TruthTable1[Nand, 2] /. {True → 1, False → 0} // TraditionalForm
TruthTable1[Nor, 2] /. {True → 1, False → 0} // TraditionalForm
TruthTable1[Or, 2] /. {True → 1, False → 0} // TraditionalForm
TruthTable1[Xor, 2] /. {True → 1, False → 0} // TraditionalForm
```

b) Les lignes d'une table de vérité définissent des vecteurs donnant les sommets d'un polytope. Pour dessiner les polytopes associés à ces tables, nous utilisons la commande **TriangularSurfacePlot** du module **DiscreteMath`ComputationalGeometry`** :

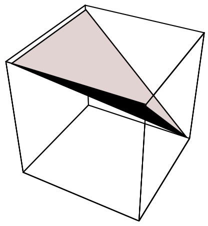
```
Needs["DiscreteMath`ComputationalGeometry`"]
TriangularSurfacePlot[{{1, 1, 1}, {1, 0, 0}, {0, 1, 0}, {0, 0, 0}}]
```

■ Sorties

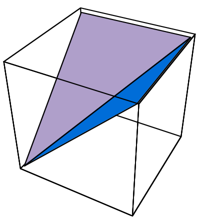
p_1	p_2	$p_1 \wedge p_2$
1	1	1
1	0	0
0	1	0
0	0	0



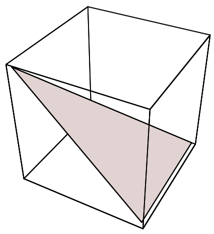
p_1	p_2	$p_1 \bar{\wedge} p_2$
1	1	0
1	0	1
0	1	1
0	0	1



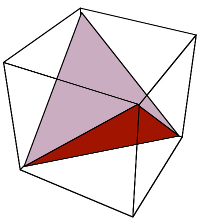
p_1	p_2	$p_1 \vee p_2$
1	1	1
1	0	1
0	1	1
0	0	0



p_1	p_2	$p_1 \bar{\vee} p_2$
1	1	0
1	0	0
0	1	0
0	0	1



p_1	p_2	$p_1 \veebar p_2$
1	1	0
1	0	1
0	1	1
0	0	0



■ Exercice 2

Différentes commandes du module **DiscreteMath`ComputationalGeometry`** permettent d'obtenir la frontière convexe d'un ensemble quelconque de points du plan. **ConvexHull** fournit l'index des points définissant la frontière lorsqu'on la parcourt dans le sens contraire aux aiguilles d'une montre. Cette frontière et la triangulation de Delaunay de l'ensemble de points peuvent être dessinées avec **PlanarGraphPlot** :

```
n = 10; (* nombre de points dont les coordonnées sont tirées au hasard *)
ConvexHull[pts = Table[{Random[], Random[]}, {n}]]
PlanarGraphPlot[pts, %,
  TextStyle → {FontFamily → Times, FontSlant → Italic, FontSize → 10}];
PlanarGraphPlot[pts, TextStyle →
   {FontFamily → Times, FontSlant → Italic, FontSize → 10}];
```

```
{7, 4, 10, 8, 2, 3, 5}
```