

Le package tnsproba

Code source disponible sur <https://github.com/typensee-latex/tnsproba.git>.

Version 0.0.0-beta développée et testée sur Mac OS X.

Christophe BAL

2020-07-10

Table des matières

1	Introduction	2
2	Probabilité	2
2.1	Probabilité « simple »	2
2.2	Probabilité conditionnelle	2
2.3	Espérance	2
3	Arbres pondérés	3
3.1	Que se passe-t-il en coulisse ?	3
3.2	Sans cadre	3
3.3	Avec des cadres	4
4	Historique	6
5	Toutes les fiches techniques	7
5.1	Probabilité	7
5.1.1	Probabilité « simple »	7
5.1.2	Probabilité conditionnelle	7
5.1.3	Probabilité – Espérance	7
5.2	Arbres pondérés	7
5.2.1	Probabilité – Arbres pondérés	7

1 Introduction

Le package `tnsproba` propose des macros utiles quand l'on parle de probabilités.

2 Probabilité

2.1 Probabilité « simple »

Exemple 1

<code>\backslashproba{A}</code>	$p(A)$
--	--------

Exemple 2 – Choisir le nom de la probabilité

<code>\backslashproba[P]{A}</code>	$P(A)$
---	--------

2.2 Probabilité conditionnelle

Exemple 1 – Les deux écritures classiques

La 1^{re} notation, qui est devenue standard, permet de comprendre l'ordre des arguments.

<code>\backslashprobacond {B}{A}</code> <code>= \probacond*{B}{A}</code>	$p_B(A) = p(A \mid B)$
--	------------------------

Exemple 2 – Obtenir la formule de définition

Le suffixe `exp` est pour `exp-and` soit « *développer* » en anglais¹.

<code>\backslashprobacondexp {B}{A}</code> <code>= \probacondexp*{B}{A}</code>	$\frac{p(A \cap B)}{p(B)} = \frac{p(A \cap B)}{p(B)}$
--	---

Exemple 3 – Choisir le nom de la probabilité

<code>\backslashprobacond [P]{B}{A}</code> <code>= \probacond* [P]{B}{A}</code> <code>= \probacondexp* [P]{B}{A}</code> <code>= \probacondexp [P]{B}{A}</code>	$P_B(A) = P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A \cap B)}{P(B)}$
--	--

2.3 Espérance

Exemple

`expval` vient de `exp-ected val-ue` soit « *espérance* » en anglais.

1. Pour ne pas alourdir l'utilisation de `\probacond`, il a été choisi d'utiliser un suffixe au lieu d'un système de multi-options.

\expval{X} $E(X)$

Choisir le nom de l'espérance

 $\expval[E_1]{X}$ $E_1(X)$

3 Arbres pondérés

3.1 Que se passe-t-il en coulisse ?

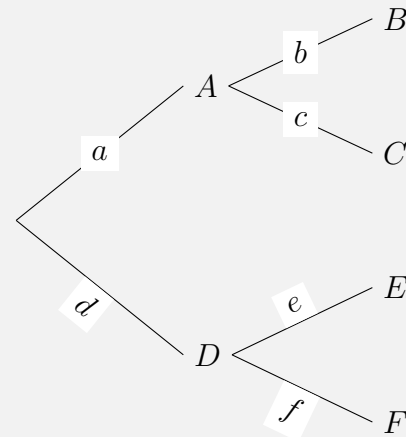
Le gros du travail est fait par le package `forest` qui utilise `TikZ`. On peut donc faire appel à la machinerie de ce dernier et obtenir des choses sympatiques comme nous allons le voir ci-dessous.

3.2 Sans cadre

Exemple 1 – Le cas type

Dans le code suivant l'environnement `probatree` utilise en coulisse celui nommé `forest` du package `forest`. Des réglages spécifiques sont faits pour obtenir le résultat ci-après. À cela s'ajoutent les styles spéciaux `pweight`, `apweight` et `bpweight` qui facilitent l'écriture des pondérations sur les branches².

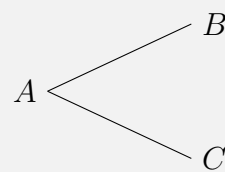
```
\begin{probatree}
[
  [A$, pweight = $a$
    [B$, pweight = $b$]
    [C$, pweight = $c$]
  ]
  [D$, bpweight = $d$
    [E$, apweight = $e$]
    [F$, bpweight = $f$]
  ]
]
\end{probatree}
```



Exemple 2 – Des poids cachés partout

On peut cacher tous les poids via l'environnement étoilé `probatree*` sans avoir à retaper un arbre où les pondérations ont déjà été indiquées.

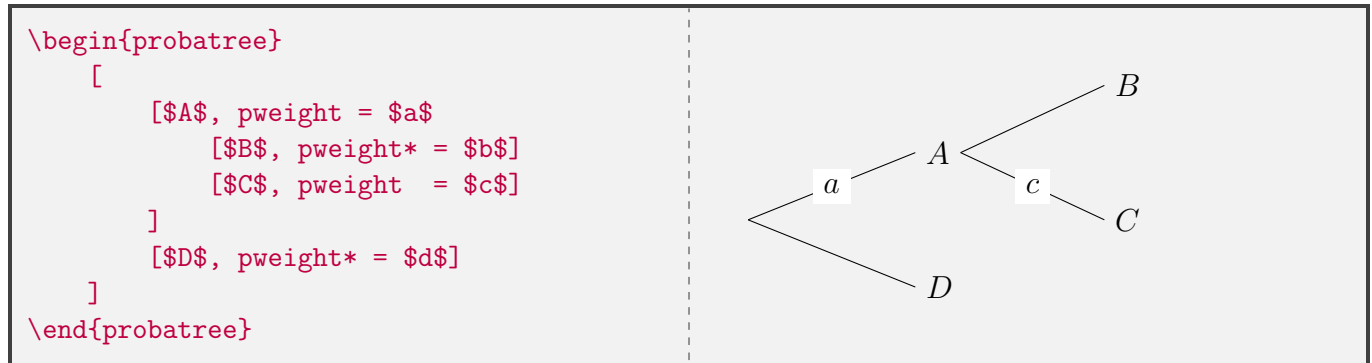
```
\begin{probatree*}
  [A$, pweight = $a$
    [B$, pweight = $b$]
    [C$, pweight = $c$]
  ]
\end{probatree*}
```



2. `pweight` vient de « *probability* » et « *weight* » soit « *probabilité* » et « *poids* » en anglais. Quant à `a` et `b` au début de `apweight` et `bpweight` respectivement, ils viennent de « *above* » et « *below* » soit « *dessus* » et « *dessous* » en anglais.

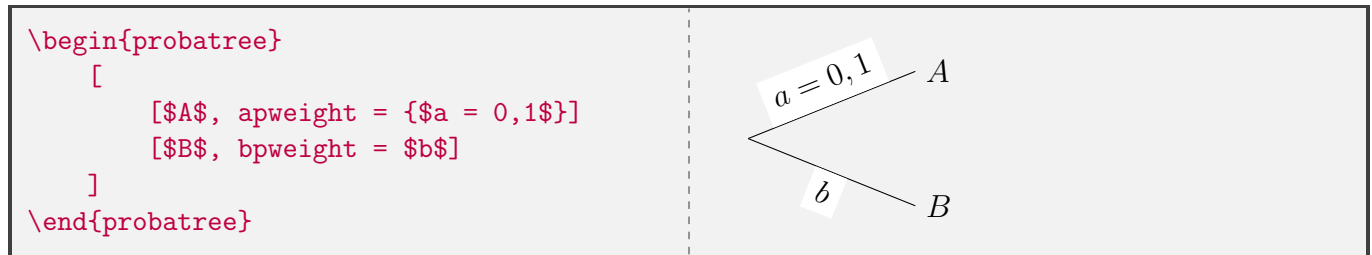
Exemple 3 – Des poids cachés localement

Pour ne cacher que certains poids, il faudra utiliser, à la main, le style `pweight*` comme dans l'exemple ci-dessous.



Exemple 4 – Un signe = et/ou une virgule dans les étiquettes

Vous ne pouvez pas utiliser directement un signe = ou une virgule dans les étiquettes des branches. L'astuce pour contourner cette limitation consiste juste à mettre le contenu de l'étiquette dans des accolades.



3.3 Avec des cadres

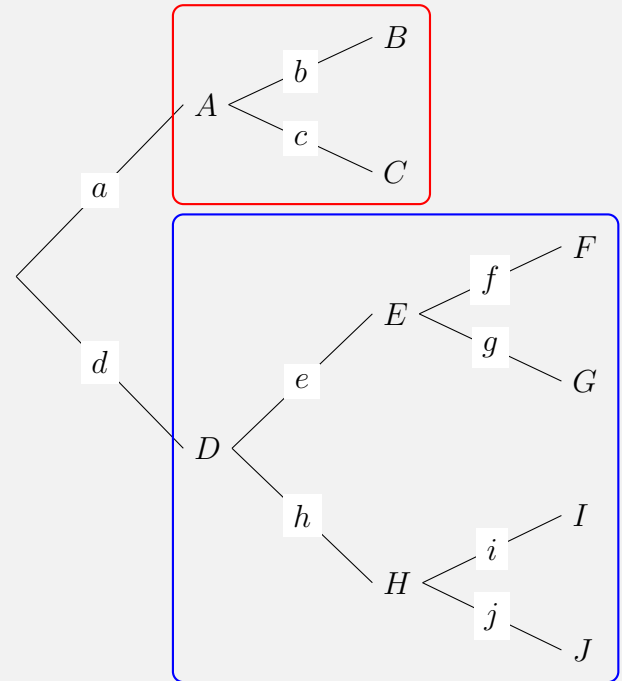
Exemple 1 – Des cadres facilement

Via la clé `frame`, il est très aisé d'encadrer un sous-arbre comme le montre l'exemple suivant. Dans l'exemple ci-après nous utilisons la bidouille `\s sep = 1.3cm` qui évite que les cadres se superposent.

```

\begin{probatree}
[{}], s sep = 1.3cm
% Astuce pour espacer les cadres.
[$A$, pweight = $a$,
    frame = red
[$B$, pweight = $b$]
[$C$, pweight = $c$]
]
[$D$, pweight = $d$,
    frame = blue
[$E$, pweight = $e$
[$F$, pweight = $f$]
[$G$, pweight = $g$]
]
[$H$, pweight = $h$
[$I$, pweight = $i$]
[$J$, pweight = $j$]
]
]
\end{probatree}

```



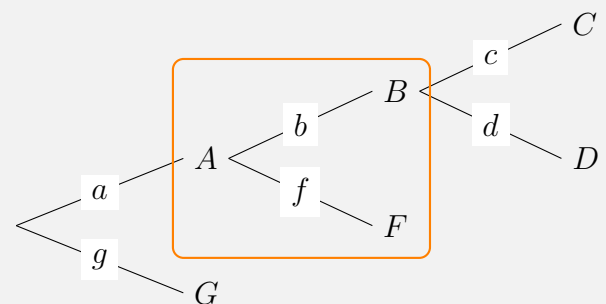
Exemple 2 – Des cadres faits à la main

En utilisant la machinerie de TiKz il est facile de décorer un arbre de probabilité comme ci-dessous où le cadre s'appuie sur trois noeuds nommés. Notons que cet exemple est tout simplement infaisable avec la clé `frame`.

```

\begin{probatree}
[
    [$A$, pweight = $a$,
        name = nA
    [$B$, pweight = $b$,
        name = nB
    [$C$, pweight = $c$]
    [$D$, pweight = $d$]
    ]
    [$F$, pweight = $f$,
        name = nF]
]
[$G$, pweight = $g$]
]
\node[draw = orange,
    thick,
    rounded corners,
    fit = (nA)(nB)(nF)] {};
\end{probatree}

```



4 Historique

Nous ne donnons ici qu'un très bref historique récent ³ de **tnsproba** à destination de l'utilisateur principalement. Tous les changements sont disponibles uniquement en anglais dans le dossier **change-log** : voir le code source de **tnsproba** sur **github**.

2020-07-10 Première version **0.0.0-beta**.

3. On ne va pas au-delà de un an depuis la dernière version.

5 Toutes les fiches techniques

5.1 Probabilité

5.1.1 Probabilité « simple »

`\proba <macro> [1 Option] (1 Argument)`

— `Option`: le nom de la probabilité. La valeur par défaut est `p`.

— `Argument`: l'ensemble dont on veut calculer la probabilité.

5.1.2 Probabilité conditionnelle

`\probacond <macro> [1 Option] (2 Arguments)`

`\probacond* <macro> [1 Option] (2 Arguments)`

`\probacondexp <macro> [1 Option] (2 Arguments)`

`\probacondexp* <macro> [1 Option] (2 Arguments)`

— `Option`: le nom de la probabilité. La valeur par défaut est `p`.

— `Argument 1`: l'ensemble qui donne la condition.

— `Argument 2`: l'ensemble dont on veut calculer la probabilité.

5.1.3 Probabilité – Espérance

`\expval <macro> [1 Option] (1 Argument)`

— `Option`: le nom de la fonction espérance. La valeur par défaut est `E`.

— `Argument`: la variable aléatoire dont on veut calculer l'espérance.

5.2 Arbres pondérés

5.2.1 Probabilité – Arbres pondérés

`probatree <env>`

`probatree* <env>`

— `Contenu`: un arbre codé en utilisant la syntaxe supportée par le package `forest`.

— `Option "pweight"`: pour écrire un poids sur le milieu d'une branche.

— `Option "apweight"`: pour écrire un poids au-dessus le milieu d'une branche.

— `Option "bpweight"`: pour écrire un poids en-dessous du milieu d'une branche.

— `Option "frame"`: pour encadrer un sous-arbre depuis un noeud vers toutes les feuilles de celui-ci.