Les algorithmes probabilistes

Johanne Cohen

Plan

Plus précisément

Probabilités élémentaires

- Notions de base
- Paradoxes des probabilités
- Différents principes
- Application : Vérification d'identités
- Techniques de réduction de l'erreur
- Loi conditionnelle
- Quelques inégalités utiles
- Quelques lois
- Application : Coupures minimales

Espace de probabilité

- Un **espace de probabilité** est donné par un triplet $(\Omega, \mathcal{A}, P)$, où
  - Ω est un ensemble (de toutes les issues/résultats possibles d'une expérence aléatoire).
  - $\mathcal{A}$ est une **tribu** :
    - $\mathcal{A}$ est une famille de parties de Ω qui contient l'ensemble vide, qui est close par union dénombrable, et qui est close par passage au complémentaire.
    - Les éléments de $\mathcal{A}$ sont appelés des **événements**.
  - $\Pr : \mathcal{A} \to [0, 1]$ est une fonction de probabilité :
    - $\Pr(\Omega) = 1$,
    - et pour toute suite d'éléments $A_1, A_2, \cdots A_n \in \mathcal{A}$ deux à deux disjoints,

$$\Pr(\bigcup_{i=1}^{n} A_i) = \sum_{i=1}^{n} \Pr(A_i). \tag{1}$$

Le cas discret

- En informatique, souvent Ω est soit fini, soit dénombrable.
- Dans ce cas,
  - **Univers Ω :** ensemble de toutes les issues/résultats possibles d'une expérence aléatoire
  - **Evénements :** l'ensemble des parties de Ω
  - **La probabilité d'un évènement A :**

$$\Pr A = \frac{|A|}{|\Omega|}$$

Le cas discret

- En informatique, souvent Ω est soit fini, soit dénombrable.

- Dans ce cas,

  - **Univers Ω :** ensemble de toutes les issues/résultats possibles d'une expérence aléatoire
    - Lancé de dé : $\{1, 2, 3, 4, 5, 6\}$
    - Naissance : $\{G, F\}$
  - **Evénements :** l'ensemble des parties de Ω
  - **La probabilité d'un évènement A :**

$$\Pr A = \frac{|A|}{|\Omega|}$$

Le cas discret

- En informatique, souvent Ω est soit fini, soit dénombrable.

- Dans ce cas,

  - **Univers Ω :** ensemble de toutes les issues/résultats possibles d'une expérence aléatoire
    - Lancé de dé : $\{1, 2, 3, 4, 5, 6\}$
    - Naissance : $\{G, F\}$
  - **Evénements :** l'ensemble des parties de Ω
  - **La probabilité d'un évènement A :**

$$\Pr A = \frac{|A|}{|\Omega|}$$

- Exemple : pour modéliser le tirage **uniforme** d'un dé,
  - $\Omega = \{1, 2, \cdots, 6\}$
  - $\mathcal{A} = \mathcal{P}(\Omega)$
  - $\Pr(\{i\}) = \frac{1}{6}$.
  - Pour $U \in \mathcal{A}$, $\Pr(U) = \frac{|U|}{6}$.

Plus précisément

Paradoxe de Monty Hall

Lors d'un jeu télévisé, un candidat est placé devant 3 portes dont derrière l'une d'elles se trouve une voiture.

Paradoxe de Monty Hall

Lors d'un jeu télévisé, un candidat est placé devant 3 portes dont derrière l'une d'elles se trouve une voiture.

1. L'emplacement du cadeau a été choisi de façon uniforme.

Paradoxe de Monty Hall

Lors d'un jeu télévisé, un candidat est placé devant 3 portes dont derrière l'une d'elles se trouve une voiture.

1. L'emplacement du cadeau a été choisi de façon uniforme.

2. Ensuite le présentateur systématique ouvre l'une des deux portes autre que celle qui a été choisie et autre que celle qui cache la voiture.

Paradoxe de Monty Hall

Lors d'un jeu télévisé, un candidat est placé devant 3 portes dont derrière l'une d'elles se trouve une voiture.

1. L'emplacement du cadeau a été choisi de façon uniforme.

2. Ensuite le présentateur systématique ouvre l'une des deux portes autre que celle qui a été choisie et autre que celle qui cache la voiture.

3. Le candidat a le choix entre maintenir son premier choix ou le modifier.

Paradoxe de Monty Hall

Lors d'un jeu télévisé, un candidat est placé devant 3 portes dont derrière l'une d'elles se trouve une voiture.

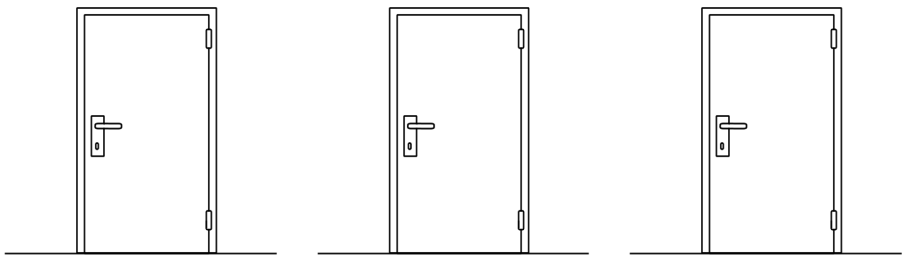
1. L'emplacement du cadeau a été choisi de façon uniforme.

2. Ensuite le présentateur systématique ouvre l'une des deux portes autre que celle qui a été choisie et autre que celle qui cache la voiture.

3. Le candidat a le choix entre maintenir son premier choix ou le modifier.

Que lui conseillez-vous de faire ?

Paradoxe de Monty Hall



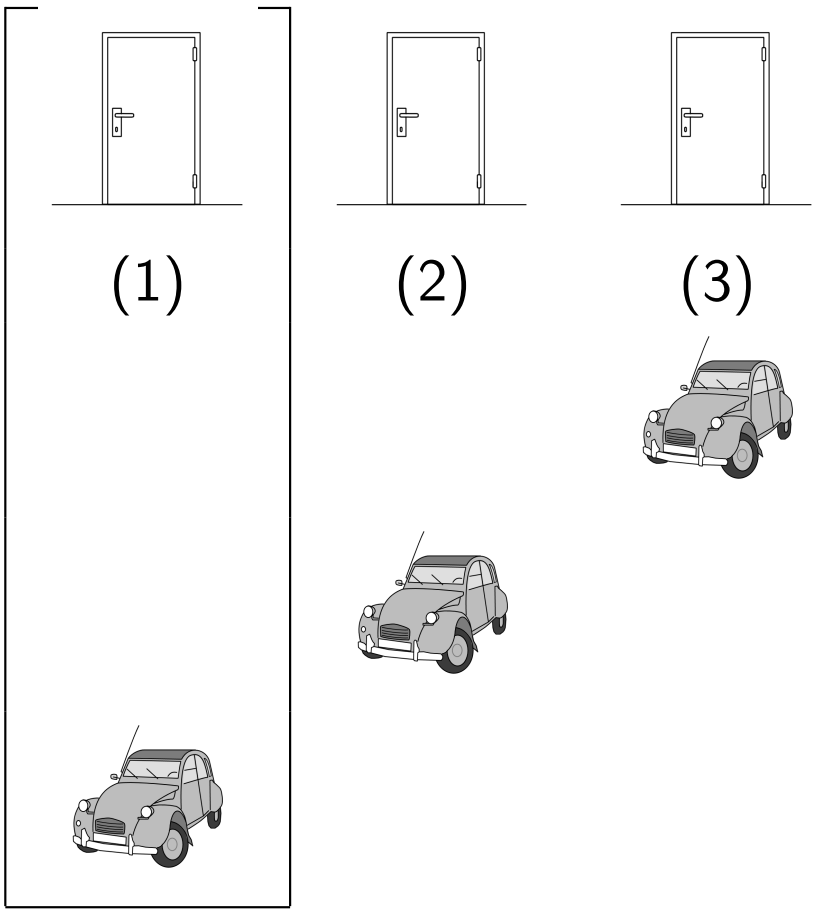
- Lorsque le candidat maintient son choix,

  sa probabilité de gagner est $1/3$.

- Cette probabilité ne dépend pas des actions du présentateur.

Paradoxe de Monty Hall

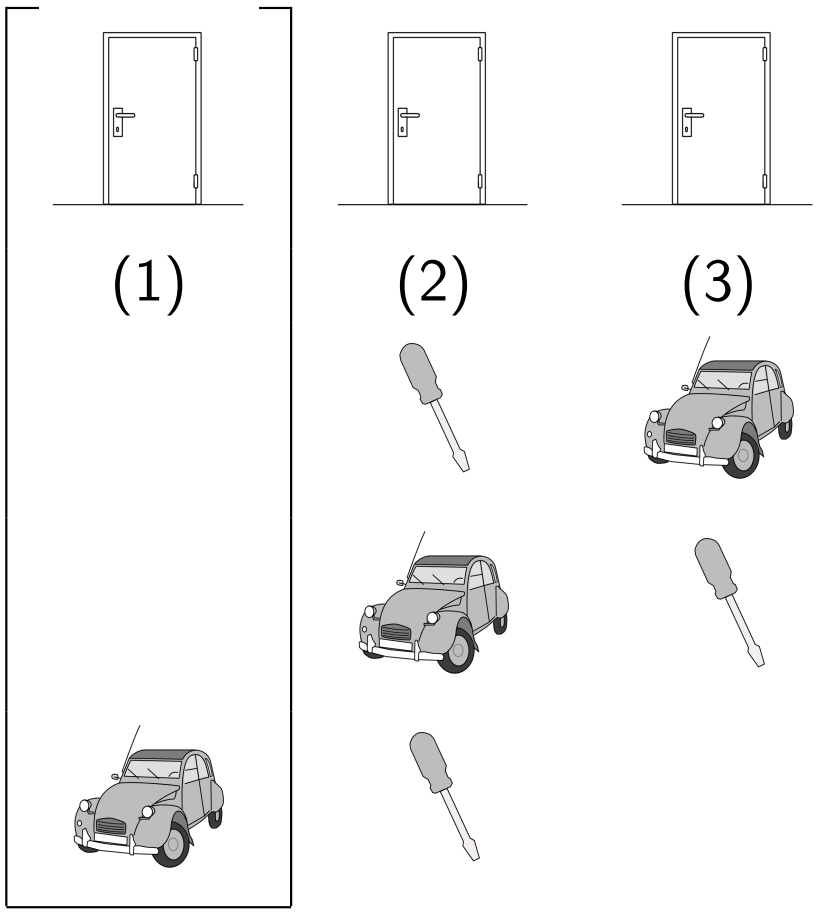
- Le candidat choisit la porte (1).



Paradoxe de Monty Hall

- Le candidat choisit la porte (1).

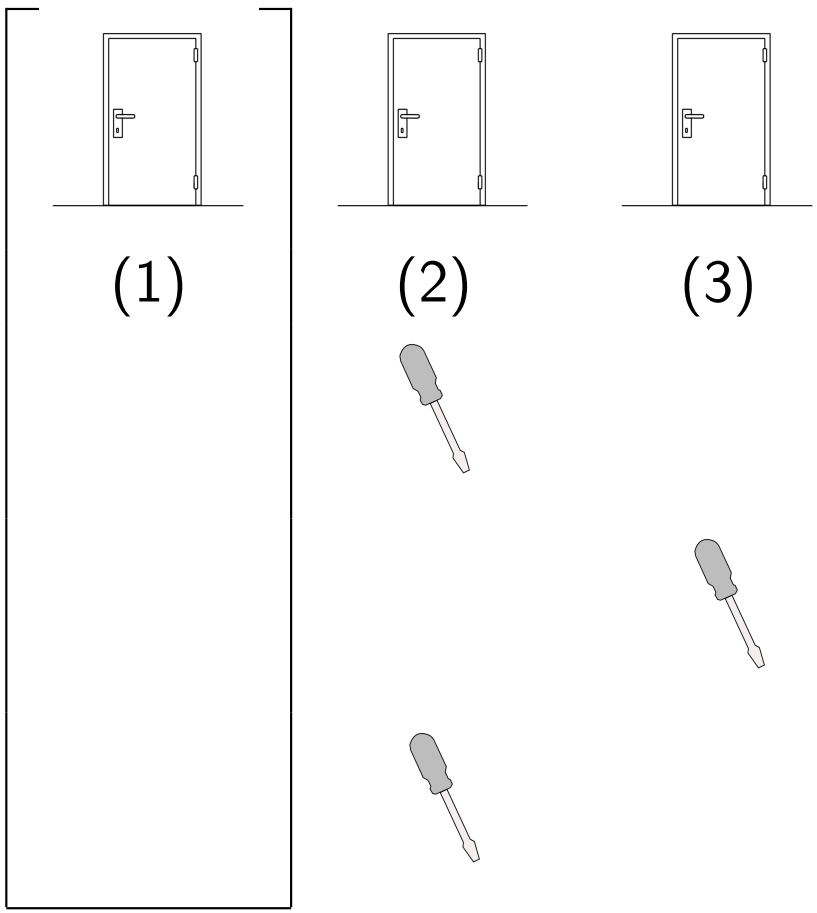
  Le présentateur lui ouvre une porte.



Paradoxe de Monty Hall

- Le candidat choisit la porte (1).

Le présentateur lui ouvre une porte.

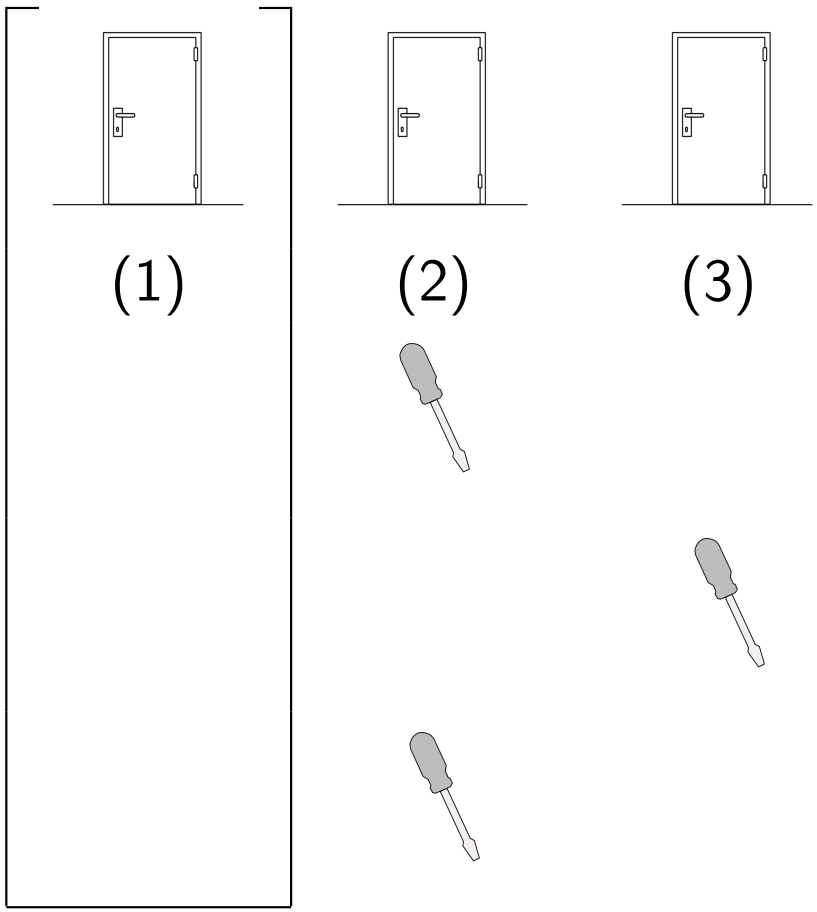


Paradoxe de Monty Hall

- Le candidat choisit la porte (1).

  Le présentateur lui ouvre une porte.

- Lorsque le candidat change de porte,

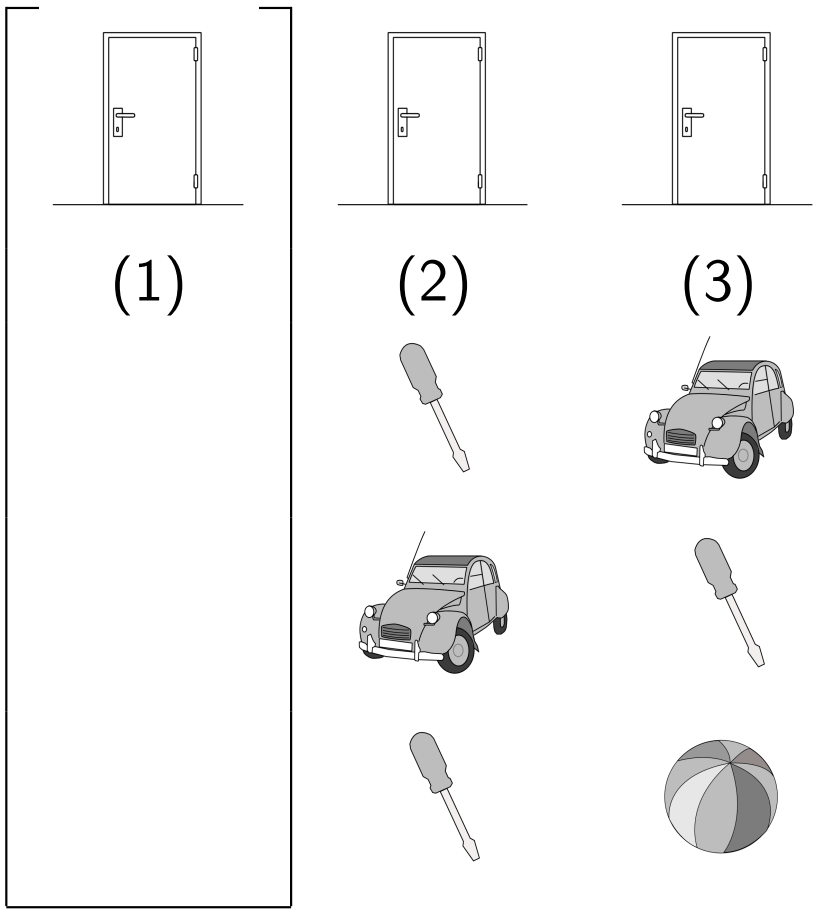


Paradoxe de Monty Hall

- Le candidat choisit la porte (1).

  Le présentateur lui ouvre une porte.

- Lorsque le candidat change de porte,



Le candidat gagne

Le candidat gagne

Le candidat perd

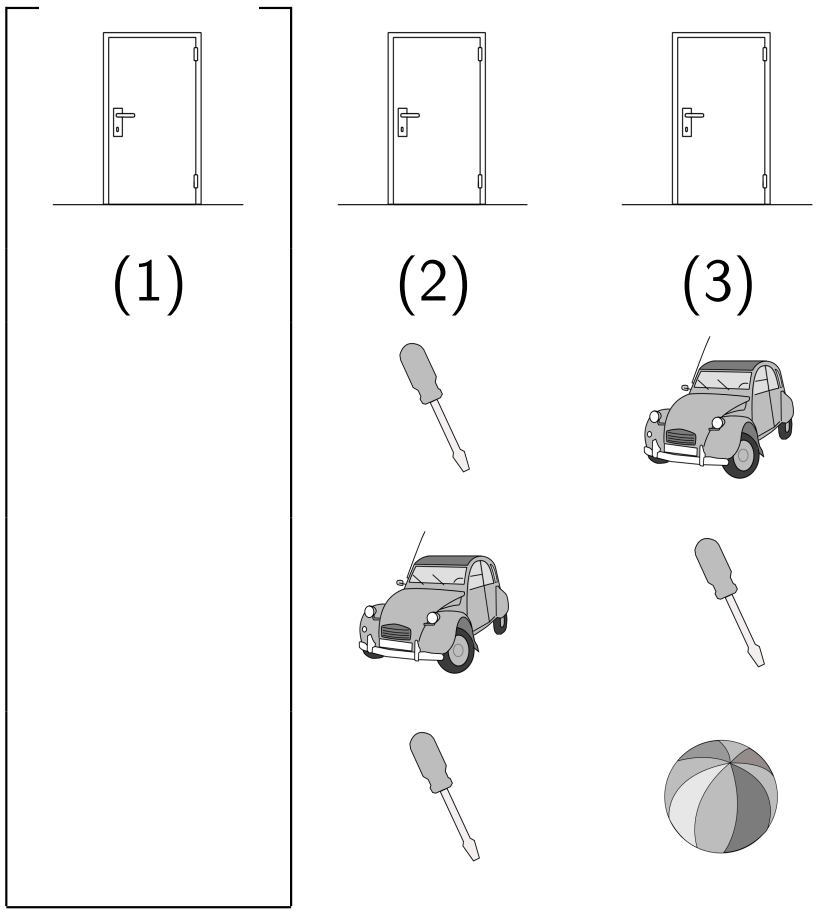
Paradoxe de Monty Hall

- Le candidat choisit la porte (1).

  Le présentateur lui ouvre une porte.

- Lorsque le candidat change de porte,

  La probabilité de gagner est donc 2/3.



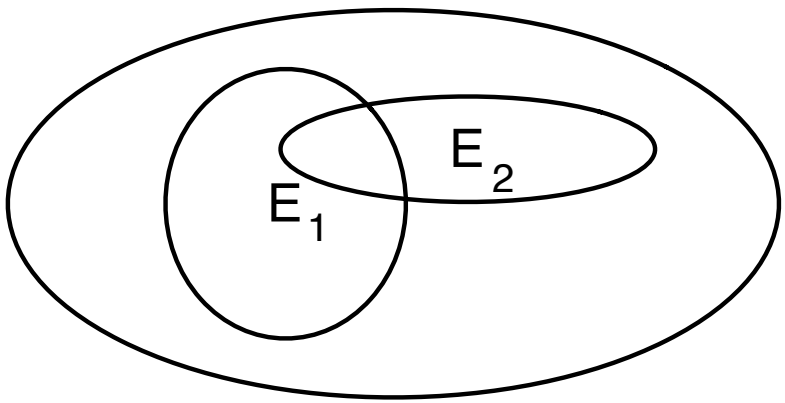
Plus précisément

Union-bound

Proposition [Union bound]

Pour toute suite finie ou dénombrablement infinie d'événements $E_1, E_2, \cdots$

$$\Pr(\bigcup_{i \geq 1} E_i) \leq \sum_{i \geq 1} \Pr(E_i).$$



Indépendance

- Deux événements E_1 et E_2 sont dit **indépendants** si

$$\Pr(E_1 \cap E_2) = \Pr(E_1)\Pr(E_2)$$

- Plus généralement, les événements $E_1, E_2, \cdots E_k$ sont dits **mutuellement indépendants** si et seulement si pour tout $I \subset \{1, 2, \cdots, k\}$,

$$\Pr(\bigcap_{i \in I} E_i) = \prod_{i \in I} \Pr(E_i).$$

Principe de décision différée

- Choisir selon une loi uniforme $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ dans $\{0,1\}^n$ est équivalent à choisir chaque x_i de façon indépendante et uniforme dans $\{0,1\}$.

Principe de décision différée

- Choisir selon une loi uniforme $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ dans $\{0,1\}^n$ est équivalent à choisir chaque x_i de façon indépendante et uniforme dans $\{0,1\}$.

- Preuve :
  - Dans les deux cas, la probabilité de choisir chacun des 2^n vecteurs possibles est 2^{-n}.

Plus précisément

Vérification d'identités

- Supposons que l'on se donne trois matrices $\mathbf{A}, \mathbf{B}$ et $\mathbf{C}$ de taille $n \times n$.
- Supposons que l'on veuille vérifier si $\mathbf{AB} = \mathbf{C}$.

Vérification d'identités

- Supposons que l'on se donne trois matrices $\mathbf{A}, \mathbf{B}$ et $\mathbf{C}$ de taille $n \times n$.
- Supposons que l'on veuille vérifier si $\mathbf{AB} = \mathbf{C}$.
- Pour simplifier la discussion, nous supposerons que ces matrices sont à coefficients dans $\mathbb{Z}_2$.

Vérification d'identités

- Supposons que l'on se donne trois matrices $\mathbf{A}, \mathbf{B}$ et $\mathbf{C}$ de taille $n \times n$.
- Supposons que l'on veuille vérifier si $\mathbf{AB} = \mathbf{C}$.
- Pour simplifier la discussion, nous supposerons que ces matrices sont à coefficients dans $\mathbb{Z}_2$.
- Une façon de faire consiste à multiplier $\mathbf{A}$ par $\mathbf{B}$, puis à tester l'égalité avec $\mathbf{C}$.
  - L'algorithme de multiplication le plus simple nécessite $\mathcal{O}(n^3)$ opérations.
  - Il existe des algorithmes plus efficaces en $\mathcal{O}(n^{2.37})$ opérations.

Algorithme randomisé

- Algorithme randomisé :
  - On choisit un vecteur aléatoire $\mathbf{x} \in \{0,1\}^n$.
  - On calcule alors $\mathbf{ABx}$ en calculant $\mathbf{Bx}$ puis $\mathbf{A}(\mathbf{Bx})$.
  - On calcule ensuite $\mathbf{Cx}$.
  - Si $\mathbf{ABx} \neq \mathbf{Cx}$ on répond que $\mathbf{AB} \neq \mathbf{C}$.
  - Sinon, on répond que $\mathbf{AB} = \mathbf{C}$.

Algorithme randomisé

- Algorithme randomisé :
  - On choisit un vecteur aléatoire $\mathbf{x} \in \{0,1\}^n$.
  - On calcule alors $\mathbf{ABx}$ en calculant $\mathbf{Bx}$ puis $\mathbf{A}(\mathbf{Bx})$.
  - On calcule ensuite $\mathbf{Cx}$.
  - Si $\mathbf{ABx} \neq \mathbf{Cx}$ on répond que $\mathbf{AB} \neq \mathbf{C}$.
  - Sinon, on répond que $\mathbf{AB} = \mathbf{C}$.
- On a besoin uniquement de trois multiplications matrice/vecteur :
  - temps $\mathcal{O}(n^2)$ par l'algorithme évident.

Probabilité d'erreur

Lemma

Si $\mathbf{AB} \neq \mathbf{C}$, *et si* $\mathbf{x}$ *est choisi uniformément dans* $\{0,1\}^n$ *alors*

$$\Pr(\mathbf{ABx} = \mathbf{Cx}) \leq \frac{1}{2}.$$

Preuve

- Soit $\mathbf{D} = \mathbf{AB} - \mathbf{C}$.
- $\mathbf{ABx} \neq \mathbf{Cx}$ implique que $\mathbf{Dx} \neq 0$, et puisque $\mathbf{D}$ n'est pas la matrice nulle, $\mathbf{D}$ doit avoir au moins un coefficient non nul.

Preuve

- Soit $\mathbf{D} = \mathbf{AB} - \mathbf{C}$.
- $\mathbf{ABx} \neq \mathbf{Cx}$ implique que $\mathbf{Dx} \neq 0$, et puisque $\mathbf{D}$ n'est pas la matrice nulle, $\mathbf{D}$ doit avoir au moins un coefficient non nul.
- Supposons sans perte de généralité que cela soit $d_{1,1}$.

Preuve

- Soit $\mathbf{D} = \mathbf{AB} - \mathbf{C}$.
- $\mathbf{ABx} \neq \mathbf{Cx}$ implique que $\mathbf{Dx} \neq 0$, et puisque $\mathbf{D}$ n'est pas la matrice nulle, $\mathbf{D}$ doit avoir au moins un coefficient non nul.

- Supposons sans perte de généralité que cela soit $d_{1,1}$.
- Pour que $\mathbf{Dx} = 0$, on doit avoir $\sum_{j=1}^{n} d_{1,j} x_j = 0$, et donc

$$x_1 = -(\sum_{j=2}^{N} d_{1,j} x_j)/d_{1,1}. \tag{2}$$

Preuve

- Soit $\mathbf{D} = \mathbf{AB} - \mathbf{C}$.
- $\mathbf{ABx} \neq \mathbf{Cx}$ implique que $\mathbf{Dx} \neq 0$, et puisque $\mathbf{D}$ n'est pas la matrice nulle, $\mathbf{D}$ doit avoir au moins un coefficient non nul.

- Supposons sans perte de généralité que cela soit $d_{1,1}$.
- Pour que $\mathbf{Dx} = 0$, on doit avoir $\sum_{j=1}^{n} d_{1,j} x_j = 0$, et donc

$$x_1 = -(\sum_{j=2}^{N} d_{1,j} x_j)/d_{1,1}. \tag{2}$$

- Selon le **principe de la décision différée**, on peut voir choisir $\mathbf{x} \in \{0,1\}^n$ comme choisir chacune de ses composantes.

Preuve

- Soit $\mathbf{D} = \mathbf{AB} - \mathbf{C}$.
- $\mathbf{ABx} \neq \mathbf{Cx}$ implique que $\mathbf{Dx} \neq 0$, et puisque $\mathbf{D}$ n'est pas la matrice nulle, $\mathbf{D}$ doit avoir au moins un coefficient non nul.

- Supposons sans perte de généralité que cela soit $d_{1,1}$.
- Pour que $\mathbf{Dx} = 0$, on doit avoir $\sum_{j=1}^{n} d_{1,j} x_j = 0$, et donc

$$x_1 = -(\sum_{j=2}^{N} d_{1,j} x_j)/d_{1,1}. \tag{2}$$

- Selon le **principe de la décision différée**, on peut voir choisir $\mathbf{x} \in \{0,1\}^n$ comme choisir chacune de ses composantes.

- Considérons la situation juste avant que x_1 soit choisi :

  - à ce moment là le membre droit est fixé, et il y a au plus une possibilité pour x_1 qui rend l'égalité vraie.

Preuve

- Soit $\mathbf{D} = \mathbf{AB} - \mathbf{C}$.
- $\mathbf{ABx} \neq \mathbf{Cx}$ implique que $\mathbf{Dx} \neq 0$, et puisque $\mathbf{D}$ n'est pas la matrice nulle, $\mathbf{D}$ doit avoir au moins un coefficient non nul.
- Supposons sans perte de généralité que cela soit $d_{1,1}$.
- Pour que $\mathbf{Dx} = 0$, on doit avoir $\sum_{j=1}^{n} d_{1,j} x_j = 0$, et donc

$$x_1 = -(\sum_{j=2}^{N} d_{1,j} x_j)/d_{1,1}. \tag{2}$$

- Selon le **principe de la décision différée**, on peut voir choisir $\mathbf{x} \in \{0,1\}^n$ comme choisir chacune de ses composantes.
- Considérons la situation juste avant que x_1 soit choisi :
  - à ce moment là le membre droit est fixé, et il y a au plus une possibilité pour x_1 qui rend l'égalité vraie.
  - Puisqu'il y a deux choix pour x_1, cela se produit avec probabilité $\leq \frac{1}{2}$.

□

Plus précisément

- On appelle un tel algorithme un algorithme de **Monte Carlo** : l'algorithme est efficace, mais peut parfois se tromper.

- On appelle un tel algorithme un algorithme de **Monte Carlo** : l'algorithme est efficace, mais peut parfois se tromper.

- Il s'agit d'un algorithme à erreur unilatérale :

- On appelle un tel algorithme un algorithme de **Monte Carlo** : l'algorithme est efficace, mais peut parfois se tromper.

- Il s'agit d'un algorithme à erreur unilatérale :

  - Si l'on trouve un $\mathbf{x}$ tel que $\mathbf{ABx} \neq \mathbf{Cx}$, on est certain de la réponse : $\mathbf{AB} \neq \mathbf{C}$.

- On appelle un tel algorithme un algorithme de **Monte Carlo** : l'algorithme est efficace, mais peut parfois se tromper.

- Il s'agit d'un algorithme à erreur unilatérale :

  - Si l'on trouve un $\mathbf{x}$ tel que $\mathbf{ABx} \neq \mathbf{Cx}$, on est certain de la réponse : $\mathbf{AB} \neq \mathbf{C}$.
  - Sinon, il y a une possibilité de se tromper.

- On appelle un tel algorithme un algorithme de **Monte Carlo** : l'algorithme est efficace, mais peut parfois se tromper.

- Il s'agit d'un algorithme à erreur unilatérale :

  - Si l'on trouve un $\mathbf{x}$ tel que $\mathbf{ABx} \neq \mathbf{Cx}$, on est certain de la réponse : $\mathbf{AB} \neq \mathbf{C}$.
  - Sinon, il y a une possibilité de se tromper.

- La probabilité d'erreur est au pire cas de $1/2$.

- On appelle un tel algorithme un algorithme de **Monte Carlo** : l'algorithme est efficace, mais peut parfois se tromper.
- Il s'agit d'un algorithme à erreur unilatérale :
  - Si l'on trouve un $\mathbf{x}$ tel que $\mathbf{ABx} \neq \mathbf{Cx}$, on est certain de la réponse : $\mathbf{AB} \neq \mathbf{C}$.
  - Sinon, il y a une possibilité de se tromper.
- La probabilité d'erreur est au pire cas de $1/2$.
- Cependant, on peut la rendre assez facilement aussi petite que l'on veut.

Réduction de l'erreur

- Supposons que l'on recommence alors k fois le test,

Réduction de l'erreur

- Supposons que l'on recommence alors k fois le test,
  - tant que **l'on n'est pas certain de la réponse**, en testant à chaque fois un nouveau $\mathbf{x}$ tiré uniformément dans $\{0,1\}^n$.

Réduction de l'erreur

- Supposons que l'on recommence alors k fois le test,
  - tant que **l'on n'est pas certain de la réponse**, en testant à chaque fois un nouveau $\mathbf{x}$ tiré uniformément dans $\{0,1\}^n$.

- La probabilité que l'on se trompe à l'issu de k tests successifs est en 2^{-k}.

Réduction de l'erreur

- Supposons que l'on recommence alors k fois le test,
  - tant que **l'on n'est pas certain de la réponse**, en testant à chaque fois un nouveau $\mathbf{x}$ tiré uniformément dans $\{0,1\}^n$.

- La probabilité que l'on se trompe à l'issu de k tests successifs est en 2^{-k}.

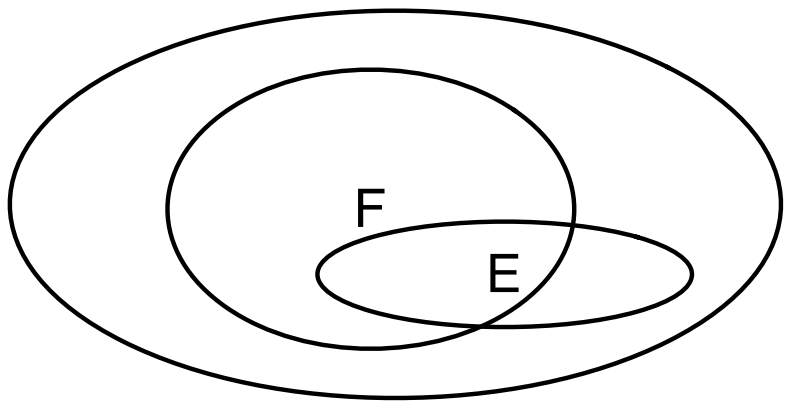
- Si l'on prend $k = 100$, cela donne une probabilité d'erreur 2^{-100} avec un temps $\mathcal{O}(kn^2) = \mathcal{O}(100n^2) = \mathcal{O}(n^2)$.

Plus précisément

Loi conditionnelle

- Rappelons la notion de **probabilité conditionnelle** :
- La probabilité de E sachant F est définie par

$$\Pr(E|F) = \frac{\Pr(E \cap F)}{\Pr(F)}.$$



Paradoxe des deux enfants

Le roi a deux enfants.

Paradoxe des deux enfants

Le roi a deux enfants.

L'espace des évènements $\Omega = \{FF, FG, GF, GG\}$

Paradoxe des deux enfants

Le roi a deux enfants.

L'espace des évènements $\Omega = \{FF, FG, GF, GG\}$

Quelle est la probabilité que les deux enfants soient des garçons ?

Paradoxe des deux enfants

Le roi a deux enfants.

L'espace des évènements $\Omega = \{FF, FG, GF, GG\}$

Quelle est la probabilité que les deux enfants soient des garçons ?

- La probabilité vaut $\mathbf{P(GG)} = \frac{|A|}{|\Omega|} = \frac{1}{4}$

Paradoxe des deux enfants

Le roi a deux enfants.

L'espace des évènements $\Omega = \{FF, FG, GF, GG\}$

Quelle est la probabilité que les deux enfants soient des garçons ?

- La probabilité vaut $\mathbf{P(GG)} = \frac{|A|}{|\Omega|} = \frac{1}{4}$
- sachant que l'ainé est un garçon.

Paradoxe des deux enfants

Le roi a deux enfants.

L'espace des évènements $\Omega = \{FF, FG, GF, GG\}$

Quelle est la probabilité que les deux enfants soient des garçons ?

- La probabilité vaut $\mathbf{P(GG)} = \frac{|A|}{|\Omega|} = \frac{1}{4}$
- sachant que l'ainé est un garçon.
  - les seuls cas possibles sont : GG et GF.

Paradoxe des deux enfants

Le roi a deux enfants.

L'espace des évènements $\Omega = \{FF, FG, GF, GG\}$

Quelle est la probabilité que les deux enfants soient des garçons ?

- La probabilité vaut $\mathbf{P(GG)} = \frac{|A|}{|\Omega|} = \frac{1}{4}$
- sachant que l'ainé est un garçon.
  - les seuls cas possibles sont : GG et GF.
  - La probabilité vaut : 1/2.

Paradoxe des deux enfants

Le roi a deux enfants.

L'espace des évènements $\Omega = \{FF, FG, GF, GG\}$

Quelle est la probabilité que les deux enfants soient des garçons ?

- La probabilité vaut $\mathbf{P(GG)} = \frac{|A|}{|\Omega|} = \frac{1}{4}$
- sachant que l'ainé est un garçon.
  - les seuls cas possibles sont : GG et GF.
  - La probabilité vaut : 1/2.
- sachant qu'au moins l'un des deux est un garçon.

Paradoxe des deux enfants

Le roi a deux enfants.

L'espace des évènements $\Omega = \{FF, FG, GF, GG\}$

Quelle est la probabilité que les deux enfants soient des garçons ?

- La probabilité vaut $\mathbf{P(GG)} = \frac{|A|}{|\Omega|} = \frac{1}{4}$
- sachant que l'ainé est un garçon.
  - les seuls cas possibles sont : GG et GF.
  - La probabilité vaut : 1/2.
- sachant qu'au moins l'un des deux est un garçon.
  - les seuls cas possibles sont : FG, GF et GG.

Paradoxe des deux enfants

Le roi a deux enfants.

L'espace des évènements $\Omega = \{FF, FG, GF, GG\}$

Quelle est la probabilité que les deux enfants soient des garçons ?

- La probabilité vaut $\mathbf{P(GG)} = \frac{|A|}{|\Omega|} = \frac{1}{4}$
- sachant que l'ainé est un garçon.
  - les seuls cas possibles sont : GG et GF.
  - La probabilité vaut : 1/2.
- sachant qu'au moins l'un des deux est un garçon.
  - les seuls cas possibles sont : FG, GF et GG.
  - La probabilité vaut : 1/3.

Plus précisément

Quelques inégalités utiles

Lemma
Soit $p \in [0, 1]$.

$$1 - p \leq e^{-p}.$$

Lemma
Soit $p \in [-1, 1]$.

$$1 + p \leq e^{p}.$$

Lemma
Soit $p \in \mathbb{R}$.

$$\frac{e^{p} + e^{-p}}{2} \leq e^{p^2/2}.$$

Chacune s'obtient en utilisant le développement en série de Taylor de e^{p} et de e^{-p} et de $e^{p^2/2}$.

Plus précisément

Quelques lois

- Loi de Bernoulli : $\Pr(X = 1) = p$, $\Pr(X = 0) = 1 - p$.
  - la variable aléatoire qui code le résultat d'une épreuve :

    1 pour " succès " avec la probabilité p,
    0 pour " échec" avec la probabilité $1 - p$

Quelques lois

- Loi de Bernoulli : $\Pr(X = 1) = p$, $\Pr(X = 0) = 1 - p$.
  - la variable aléatoire qui code le résultat d'une épreuve :
    1 pour " succès " avec la probabilité p,
    0 pour " échec" avec la probabilité $1 - p$
- Loi Géométrique : $\Pr(X = n) = (1 - p)^{n-1}p$.
  - On renouvelle une épreuve de Bernoulli de manière indépendante jusqu'au premier succès. Cette loi correspond à la variable aléatoire donnant le rang du premier succès.

Quelques lois

- Loi de Bernoulli : $\Pr(X = 1) = p$, $\Pr(X = 0) = 1 - p$.
  - la variable aléatoire qui code le résultat d'une épreuve :
    1 pour " succès " avec la probabilité p,
    0 pour " échec" avec la probabilité $1 - p$
- Loi Géométrique : $\Pr(X = n) = (1 - p)^{n-1} p$.
  - On renouvelle une épreuve de Bernoulli de manière indépendante jusqu'au premier succès. Cette loi correspond à la variable aléatoire donnant le rang du premier succès.
- Loi Binomiale de paramètres n et p :
  $\Pr(X = j) = C_n^j p^j (1 - p)^{n-j}$.
  - On renouvelle n fois de manière indépendante une épreuve de Bernoulli de paramètre p. Cette loi correspond à la variable aléatoire donnant le nombre de succès obtenus à l'issue des n épreuves.

Plus précisément

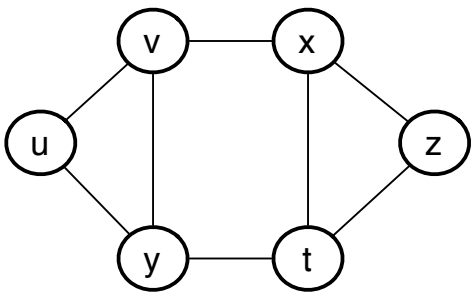
Probabilités élémentaires

- Notions de base
- Paradoxes des probabilités
- Différents principes
- Application : Vérification d'identités
- Techniques de réduction de l'erreur
- Loi conditionnelle
- Quelques inégalités utiles
- Quelques lois
- Application : Coupures minimales

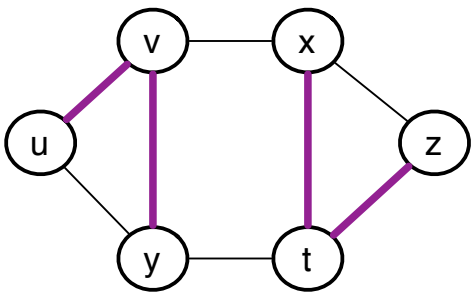
Coupures minimales

- Une **coupure d'un graphe** est un ensemble d'arêtes telles que les enlever coupe le graphe en deux ou plus composantes connexes.

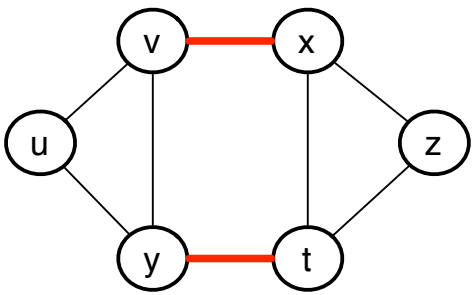
- Étant donné un graphe $G = (V, E)$, le problème de la coupure **minimale** consiste à déterminer une coupure de cardinalité minimale du graphe.



graphe G



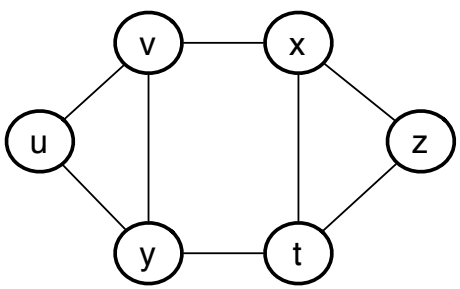
coupure



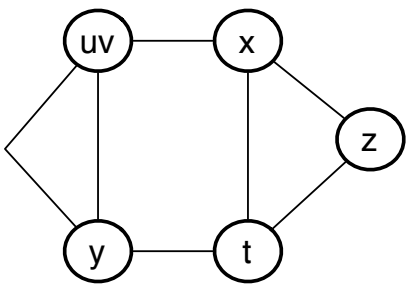
coupure minimale

Algorithme randomisé

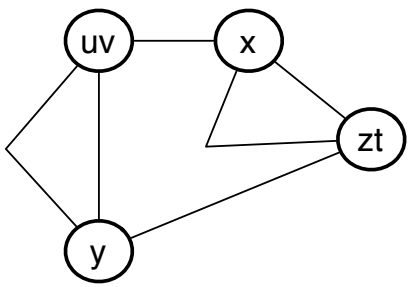
- On opère $n-2$ itérations, où n est le nombre de sommets.
- A chaque itération, on choisit aléatoirement selon une loi uniforme une arête $\{u, v\}$ du graphe, et on la contracte :
  - c'est-à-dire, on fusionne u et v en un unique sommet, on élimine toutes les arêtes entre u et v et on garde toutes les autres arêtes.
  - Le nouveau graphe obtenu peut avoir des multi-arêtes (plusieurs arêtes entre deux mêmes sommets) mais pas de boucle.



graphe G



Après fusion de $\{u, v\}$



Après fusion de $\{z, t\}$

Algorithme randomisé

- On opère $n-2$ itérations, où n est le nombre de sommets.
- A chaque itération, on choisit aléatoirement selon une loi uniforme une arête $\{u, v\}$ du graphe, et on la contracte :
  - c'est-à-dire, on fusionne u et v en un unique sommet, on élimine toutes les arêtes entre u et v et on garde toutes les autres arêtes.
  - Le nouveau graphe obtenu peut avoir des multi-arêtes (plusieurs arêtes entre deux mêmes sommets) mais pas de boucle.
- Chaque itération réduit le nombre de sommet de 1. Après $n-2$ itérations, il ne reste donc plus que deux sommets avec un ensemble d'arêtes C entre ces sommets.

Algorithme randomisé

- On opère $n-2$ itérations, où n est le nombre de sommets.
- A chaque itération, on choisit aléatoirement selon une loi uniforme une arête $\{u, v\}$ du graphe, et on la contracte :
  - c'est-à-dire, on fusionne u et v en un unique sommet, on élimine toutes les arêtes entre u et v et on garde toutes les autres arêtes.
  - Le nouveau graphe obtenu peut avoir des multi-arêtes (plusieurs arêtes entre deux mêmes sommets) mais pas de boucle.
- Chaque itération réduit le nombre de sommet de 1. Après $n-2$ itérations, il ne reste donc plus que deux sommets avec un ensemble d'arêtes C entre ces sommets.
- On retourne alors cet ensemble d'arêtes C.

Analyse

- Toute coupure d'un graphe obtenu à l'une des itérations est une coupure du graphe initial.

Analyse

- Toute coupure d'un graphe obtenu à l'une des itérations est une coupure du graphe initial.
- Par conséquent, la réponse C est toujours une coupure du graphe initiale, mais elle peut être de taille non minimale.

Analyse

- Toute coupure d'un graphe obtenu à l'une des itérations est une coupure du graphe initial.
- Par conséquent, la réponse C est toujours une coupure du graphe initiale, mais elle peut être de taille non minimale.

Analyse

- Toute coupure d'un graphe obtenu à l'une des itérations est une coupure du graphe initial.

- Par conséquent, la réponse C est toujours une coupure du graphe initiale, mais elle peut être de taille non minimale.

Qualité de l'algorithme

L'algorithme produit une coupure minimale
avec probabilité au moins $\frac{2}{n(n-1)}$.

Analyse

- Toute coupure d'un graphe obtenu à l'une des itérations est une coupure du graphe initial.

- Par conséquent, la réponse C est toujours une coupure du graphe initiale, mais elle peut être de taille non minimale.

Qualité de l'algorithme

L'algorithme produit une coupure minimale
avec probabilité au moins $\frac{2}{n(n-1)}$.

- Il s'agit donc encore ici d'un algorithme de Monte Carlo.

Idée de la preuve

- Soit k la taille de la coupure minimale.
- Le graphe peut avoir plusieurs coupures C de taille minimale.

On calcule la probabilité d'en trouver une.

Idée de la preuve

- Soit k la taille de la coupure minimale.
- Le graphe peut avoir plusieurs coupures C de taille minimale.

  On calcule la probabilité d'en trouver une.
- Puisque C est une coupure, supprimer C partitionne les sommets en deux ensembles S et $V - S$, tel qu'aucune arête connecte un sommet de S à un sommet de $V - S$.

Idée de la preuve

- Soit k la taille de la coupure minimale.
- Le graphe peut avoir plusieurs coupures C de taille minimale.
  On calcule la probabilité d'en trouver une.
- Puisque C est une coupure, supprimer C partitionne les sommets en deux ensembles S et $V - S$, tel qu'aucune arête connecte un sommet de S à un sommet de $V - S$.
- Supposons que pendant l'exécution, on contracte des arêtes de S et de $V - S$ mais jamais de C.

Idée de la preuve

- Soit k la taille de la coupure minimale.
- Le graphe peut avoir plusieurs coupures C de taille minimale.
  On calcule la probabilité d'en trouver une.
- Puisque C est une coupure, supprimer C partitionne les sommets en deux ensembles S et $V - S$, tel qu'aucune arête connecte un sommet de S à un sommet de $V - S$.
- Supposons que pendant l'exécution, on contracte des arêtes de S et de $V - S$ mais jamais de C.
- Dans ce cas, après $n - 2$ itérations, l'algorithme retourne un graphe avec deux sommets connectés par les arêtes de C.

Idée de la preuve

- Soit k la taille de la coupure minimale.
- Le graphe peut avoir plusieurs coupures C de taille minimale.
  On calcule la probabilité d'en trouver une.
- Puisque C est une coupure, supprimer C partitionne les sommets en deux ensembles S et $V - S$, tel qu'aucune arête connecte un sommet de S à un sommet de $V - S$.
- Supposons que pendant l'exécution, on contracte des arêtes de S et de $V - S$ mais jamais de C.
- Dans ce cas, après $n - 2$ itérations, l'algorithme retourne un graphe avec deux sommets connectés par les arêtes de C.
- On en déduit que si l'algorithme ne choisit jamais une arête de C dans ses $n - 2$ itérations, alors il retournera C.

Idée de la preuve

- Soit k la taille de la coupure minimale.
- Le graphe peut avoir plusieurs coupures C de taille minimale.
  On calcule la probabilité d'en trouver une.
- Puisque C est une coupure, supprimer C partitionne les sommets en deux ensembles S et $V - S$, tel qu'aucune arête connecte un sommet de S à un sommet de $V - S$.
- Supposons que pendant l'exécution, on contracte des arêtes de S et de $V - S$ mais jamais de C.
- Dans ce cas, après $n - 2$ itérations, l'algorithme retourne un graphe avec deux sommets connectés par les arêtes de C.
- On en déduit que si l'algorithme ne choisit jamais une arête de C dans ses $n - 2$ itérations, alors il retournera C.
- Cela donne l'intuition de l'algorithme :
  - en choisissant les arêtes uniformément, si C est petit, la probabilité de choisir une arête de C est faible.

Preuve (1/2) :

- Soit E_i l'événement que l'arête **contractée à l'itération i n'est pas dans C.**

Preuve (1/2) :

- Soit E_i l'événement que l'arête **contractée à l'itération** i **n'est pas dans** C**.**
- Soit $F_i = \bigcap_{j=1}^{i} E_j$ l'événement qu'**aucune** arête de C **n'**a été contractée pendant les premières i itérations.

Preuve (1/2) :

- Soit E_i l'événement que l'arête **contractée à l'itération i n'est pas dans C.**
- Soit $F_i = \bigcap_{j=1}^{i} E_j$ l'événement qu'**aucune** arête de C **n'**a été contractée pendant les premières i itérations.
- Puisque la coupure minimale est de taille k, tout sommet doit être de degré au moins k :

Preuve (1/2) :

- Soit E_i l'événement que l'arête **contractée à l'itération i n'est pas dans C.**
- Soit $F_i = \bigcap_{j=1}^{i} E_j$ l'événement qu'**aucune** arête de C **n'**a été contractée pendant les premières i itérations.
- Puisque la coupure minimale est de taille k, tout sommet doit être de degré au moins k :
  - en effet, si on enlève toutes les arêtes incidentes à un sommet, alors ce sommet devient sa propre composante connexe, et les arêtes incidentes en ce sommet constituent une coupure.

Preuve (1/2) :

- Soit E_i l'événement que l'arête **contractée à l'itération i n'est pas dans C.**
- Soit $F_i = \bigcap_{j=1}^{i} E_j$ l'événement qu'**aucune** arête de C **n'**a été contractée pendant les premières i itérations.
- Puisque la coupure minimale est de taille k, tout sommet doit être de degré au moins k :
  - en effet, si on enlève toutes les arêtes incidentes à un sommet, alors ce sommet devient sa propre composante connexe, et les arêtes incidentes en ce sommet constituent une coupure.
- Si chaque sommet est adjacent à au moins k arêtes, le graphe doit donc avoir au moins $\frac{nk}{2}$ arêtes.

Preuve (1/2) :

- Soit E_i l'événement que l'arête **contractée à l'itération i n'est pas dans C.**
- Soit $F_i = \bigcap_{j=1}^{i} E_j$ l'événement qu'**aucune** arête de C **n'**a été contractée pendant les premières i itérations.
- Puisque la coupure minimale est de taille k, tout sommet doit être de degré au moins k :
  - en effet, si on enlève toutes les arêtes incidentes à un sommet, alors ce sommet devient sa propre composante connexe, et les arêtes incidentes en ce sommet constituent une coupure.
- Si chaque sommet est adjacent à au moins k arêtes, le graphe doit donc avoir au moins $\frac{nk}{2}$ arêtes.
- La probabilité que l'on ne choisisse pas k arêtes parmi ces $\frac{nk}{2}$ arêtes vérifie donc

$$\Pr(E_1) = \Pr(F_1) \geq 1 - \frac{2k}{nk} = 1 - \frac{2}{n}.$$

Preuve (2/2) :

- Supposons que la première contraction n'a pas éliminé une arête de C : autrement dit, on conditionne sur F_1.

Preuve (2/2) :

- Supposons que la première contraction n'a pas éliminé une arête de C : autrement dit, on conditionne sur F_1.
- Après la première itération, on a un graphe avec $n-1$ sommets, de coupure minimale de taille k.

Preuve (2/2) :

- Supposons que la première contraction n'a pas éliminé une arête de C : autrement dit, on conditionne sur F_1.
- Après la première itération, on a un graphe avec $n-1$ sommets, de coupure minimale de taille k.
- On peut donc utiliser exactement le même raisonnement sur ce nouveau graphe, pour obtenir :

$$\Pr(E_2|F_1) \geq 1 - \frac{2k}{(n-1)k} = 1 - \frac{2}{n-1}.$$

Preuve (2/2) :

- Supposons que la première contraction n'a pas éliminé une arête de C : autrement dit, on conditionne sur F_1.
- Après la première itération, on a un graphe avec $n-1$ sommets, de coupure minimale de taille k.
- On peut donc utiliser exactement le même raisonnement sur ce nouveau graphe, pour obtenir :

$$\Pr(E_2|F_1) \geq 1 - \frac{2k}{(n-1)k} = 1 - \frac{2}{n-1}.$$

- De même $\Pr(E_i|F_{i-1}) \geq 1 - \frac{2k}{(n-i+1)k} = 1 - \frac{2}{n-i+1}$.

Preuve (2/2) :

- Supposons que la première contraction n'a pas éliminé une arête de C : autrement dit, on conditionne sur F_1.
- Après la première itération, on a un graphe avec $n-1$ sommets, de coupure minimale de taille k.
- On peut donc utiliser exactement le même raisonnement sur ce nouveau graphe, pour obtenir :

$$\Pr(E_2|F_1) \geq 1 - \frac{2k}{(n-1)k} = 1 - \frac{2}{n-1}.$$

- De même $\Pr(E_i|F_{i-1}) \geq 1 - \frac{2k}{(n-i+1)k} = 1 - \frac{2}{n-i+1}$.
- On écrit alors

$$\begin{aligned}
\Pr(F_{n-2}) &= \Pr(E_{n-2}|F_{n-3})\Pr(E_{n-3}|F_{n-4})\cdots\Pr(E_2|F_1)\Pr(F_1). \\
&\geq \prod_{i=1}^{n-2}\left(1 - \frac{2}{n-i+1}\right) \geq \prod_{i=1}^{n-2}\left(\frac{n-i-1}{n-i+1}\right) \\
&\geq \frac{n-2}{n}\frac{n-3}{n-1}\frac{n-4}{n-2}\cdots\frac{4}{6}\frac{3}{5}\frac{2}{4}\frac{1}{3} \geq \frac{2}{n(n-1)}.
\end{aligned}$$

Réduction de l'erreur

- L'algorithme produit une coupure minimale avec probabilité au moins $\frac{2}{n(n-1)}$.
- Puisque l'algorithme est à erreur unilatérale, on peut réduire l'erreur facilement comme précédemment :
  - supposons que l'on exécute l'algorithme $n(n-1)\log n$ **fois**, et que l'on produise en sortie la plus petite coupure trouvée dans toutes les itérations.

Réduction de l'erreur

- L'algorithme produit une coupure minimale avec probabilité au moins $\frac{2}{n(n-1)}$.
- Puisque l'algorithme est à erreur unilatérale, on peut réduire l'erreur facilement comme précédemment :
  - supposons que l'on exécute l'algorithme $n(n-1)\log n$ **fois**, et que l'on produise en sortie la plus petite coupure trouvée dans toutes les itérations.
  - La probabilité que l'on ne produise pas une coupure minimale est bornée par

$$\left(1 - \frac{2}{n(n-1)}\right)^{n(n-1)\log n} \leq e^{-2\log n} = \frac{1}{n^2},$$

où l'on a utilisé le fait que $1 - x \leq e^{-x}$.

Plan

Variable aléatoire et moyenne

- Une **variable aléatoire** sur un espace de probabilité Ω discret est une fonction $X : \Omega \to \mathbb{R}$.

- Une **variable aléatoire discrète** est une variable aléatoire qui prend un nombre fini ou dénombrable de valeurs.

Variable aléatoire et moyenne

- Une **variable aléatoire** sur un espace de probabilité Ω discret est une fonction $X : \Omega \to \mathbb{R}$.
- Une **variable aléatoire discrète** est une variable aléatoire qui prend un nombre fini ou dénombrable de valeurs.
- La moyenne d'une variable aléatoire discrète X, notée $E[X]$, est définie par

$$E[X] = \sum_i i \Pr(X = i).$$

Quelques lois

Espérance

- Loi de Bernoulli :

$$\Pr(X = 1) = p, \quad \Pr(X = 0) = 1 - p \qquad E[X] = p.$$

- Loi Géométrique :

$$\Pr(X = n) = (1 - p)^{n-1} p. \qquad E[X] = 1/p.$$

- Loi Binomiale :

$$\Pr(X = j) = C_n^j p^j (1 - p)^{n-j}. \qquad E[X] = np.$$

Linéarité de la moyenne

Theorem (Linéarité de la moyenne)

Pour toute famille finie de variables aléatoires $X_1, X_2, \cdots, X_n$ discrète de moyennes finies

$$E[\sum_{i=1}^{n} X_n] = \sum_{i=1}^{n} E[X_i].$$

Linéarité de la moyenne

Theorem (Linéarité de la moyenne)

Pour toute famille finie de variables aléatoires $X_1, X_2, \cdots, X_n$ discrète de moyennes finies

$$E[\sum_{i=1}^{n} X_n] = \sum_{i=1}^{n} E[X_i].$$

- Remarque importante :
  - aucune hypothèse sur l'indépendance des variables aléatoires.

Plus précisément

Variables aléatoires et moyennes

- Application : Problème du collectionneur
- Application : Tri Bucket Sort
- Application : Temps moyen du tri rapide

Problème du collectionneur

- Problème du collectionneur :
  - Supposons que des boites de céréales contiennent chacune un coupon parmi n coupons possibles.
  - Supposons que l'on veuille posséder au moins un exemplaire de chacun des coupons.
  - Combien faut il acheter en moyenne de boites de céréales pour cela ?

Problème du collectionneur

- Problème du collectionneur :
  - Supposons que des boites de céréales contiennent chacune un coupon parmi n coupons possibles.
  - Supposons que l'on veuille posséder au moins un exemplaire de chacun des coupons.
  - Combien faut il acheter en moyenne de boites de céréales pour cela ?
- Ce problème apparaît dans de nombreux contextes en informatique.

Problème du collectionneur : méthode

- Soit X le nombre de boîtes achetées avant de posséder tous les n coupons. **On s'intéresse donc à $E[X]$.**

Problème du collectionneur : méthode

- Soit X le nombre de boîtes achetées avant de posséder tous les n coupons. **On s'intéresse donc à $E[X]$.**
- X_i désigne le nombre de boîtes achetées en ayant exactement $i-1$ coupons pour posséder un coupon supplémentaire, alors

$$X = \sum_{i=1}^{n} X_i.$$

Problème du collectionneur : méthode

- Soit X le nombre de boîtes achetées avant de posséder tous les n coupons. **On s'intéresse donc à $E[X]$.**
- X_i désigne le nombre de boîtes achetées en ayant exactement $i-1$ coupons pour posséder un coupon supplémentaire, alors

$$X = \sum_{i=1}^{n} X_i.$$

- Lorsque l'on a exactement $i-1$ coupons, la probabilité d'obtenir un nouveau coupon en achetant une boite est

$$p_i = 1 - \frac{i-1}{n}.$$

Problème du collectionneur : méthode

- Soit X le nombre de boîtes achetées avant de posséder tous les n coupons. **On s'intéresse donc à $E[X]$.**
- X_i désigne le nombre de boîtes achetées en ayant exactement $i-1$ coupons pour posséder un coupon supplémentaire, alors

$$X = \sum_{i=1}^{n} X_i.$$

- Lorsque l'on a exactement $i-1$ coupons, la probabilité d'obtenir un nouveau coupon en achetant une boite est

$$p_i = 1 - \frac{i-1}{n}.$$

- X_i est une variable aléatoire géométrique de paramètre p_i et

$$E[X_i] = \frac{1}{p_i} = \frac{n}{n-i+1}.$$

Problème du collectionneur : méthode

- Soit X le nombre de boîtes achetées avant de posséder tous les n coupons. **On s'intéresse donc à $E[X]$.**
- X_i désigne le nombre de boîtes achetées en ayant exactement $i-1$ coupons pour posséder un coupon supplémentaire, alors

$$X = \sum_{i=1}^{n} X_i.$$

- Lorsque l'on a exactement $i-1$ coupons, la probabilité d'obtenir un nouveau coupon en achetant une boite est

$$p_i = 1 - \frac{i-1}{n}.$$

- X_i est une variable aléatoire géométrique de paramètre p_i et

$$E[X_i] = \frac{1}{p_i} = \frac{n}{n-i+1}.$$

- En utilisant la linéarité de la moyenne, on obtient

$$E[X] = E[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i}.$$

Problème du collectionneur : réponse

- Le résultat suivant est connu.
  - Le nombre $H(n) = \sum_{i=1}^{n} \frac{1}{i}$, connu sous le nom du $n$ème **nombre harmonique**, vérifie $H(n) = \log(n) + \Theta(1)$.
- Par conséquent, la réponse au problème du collectionneur de coupon est

$$n \log(n) + \Theta(n).$$

Plus précisément

Variables aléatoires et moyennes

- Application : Problème du collectionneur
- Application : Tri Bucket Sort
- Application : Temps moyen du tri rapide

Tri Bucket Sort

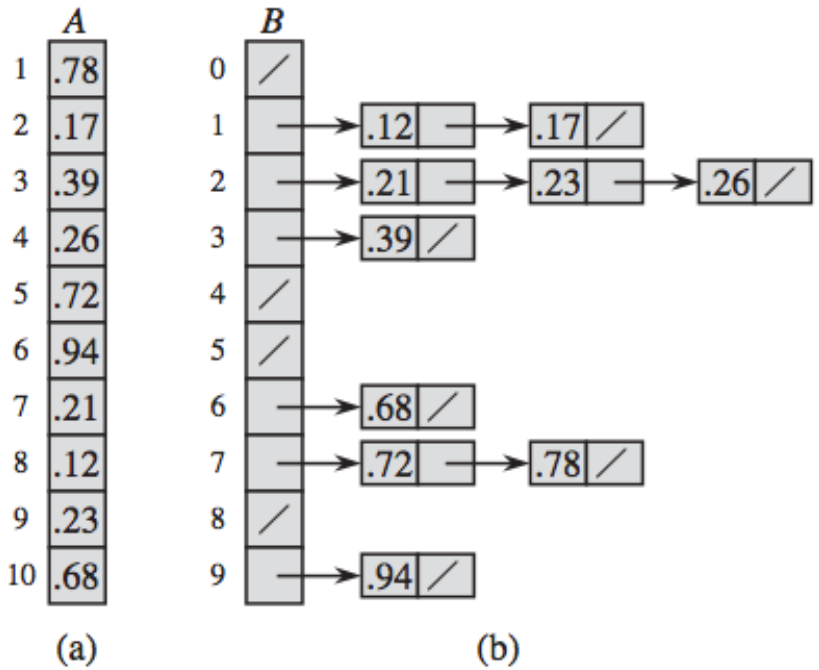
Le tri **Bucket Sort** est un exemple de méthode de tri qui brise la borne inférieure de $\Omega(n \log(n))$ opérations par comparaisons :

- supposons que l'on ait n éléments à trier, et que chaque élément est un entier choisit uniformément dans l'intervalle 0 à 1.
- Avec ce tri, on peut trier en temps moyen $\mathcal{O}(n)$.

Tri Bucket Sort

- Dans une première étape,
  - on partage l'intervalle $[0, 1]$ en n intervalles de même longueur.
  - on place les éléments dans n emplacements : l'emplacement j contient tous les éléments dont les m premiers bits correspondent au nombre j.
  - En supposant que placer un élément dans un emplacement se fait en temps $\mathcal{O}(1)$, cette étape nécessite un temps $\mathcal{O}(n)$.
- Dans une deuxième étape,
  - l'algorithme trie chaque emplacement, avec un algorithme de complexité quadratique.
- L'algorithme produit alors en sortie le résultat de la concaténation des listes triées.

Tri Bucket Sort : Illustration



$$\text{Nombre de comparaisons} = \mathcal{O}(n) + \sum_{i=0}^{n-1} \mathcal{O}(X_i^2)$$

avec X_i la variable aléatoire correspondant au nombre d'éléments dans l'emplacement j.

Analyse

- Le nombre d'éléments qui tombe dans chaque emplacement suit une loi binomiale $B(n, 1/n)$. Les emplacements peuvent s'implémenter par des listes par exemple.

Analyse

- Le nombre d'éléments qui tombe dans chaque emplacement suit une loi binomiale $B(n, 1/n)$. Les emplacements peuvent s'implémenter par des listes par exemple.
- X_j : le nombre d'éléments qui tombe dans l'emplacement j.

Analyse

- Le nombre d'éléments qui tombe dans chaque emplacement suit une loi binomiale $B(n, 1/n)$. Les emplacements peuvent s'implémenter par des listes par exemple.
- X_j : le nombre d'éléments qui tombe dans l'emplacement j.
- Le temps pour trier chaque emplacement est de la forme $c(X_j)^2$ pour une constante c.

Analyse

- Le nombre d'éléments qui tombe dans chaque emplacement suit une loi binomiale $B(n, 1/n)$. Les emplacements peuvent s'implémenter par des listes par exemple.
- X_j : le nombre d'éléments qui tombe dans l'emplacement j.
- Le temps pour trier chaque emplacement est de la forme $c(X_j)^2$ pour une constante c.
- Le temps total pour la deuxième phase est donné par

$$E[\sum_{j=1}^{n} c(X_j)^2] = c \sum_{j=1}^{n} E[X_j^2] = cnE[X_1^2],$$

où l'on a utilisé la linéarité de la moyenne et le fait que chaque emplacement joue un rôle symétrique.

Analyse

- Le nombre d'éléments qui tombe dans chaque emplacement suit une loi binomiale $B(n, 1/n)$. Les emplacements peuvent s'implémenter par des listes par exemple.
- X_j : le nombre d'éléments qui tombe dans l'emplacement j.
- Le temps pour trier chaque emplacement est de la forme $c(X_j)^2$ pour une constante c.
- Le temps total pour la deuxième phase est donné par

$$E[\sum_{j=1}^{n} c(X_j)^2] = c \sum_{j=1}^{n} E[X_j^2] = cnE[X_1^2],$$

où l'on a utilisé la linéarité de la moyenne et le fait que chaque emplacement joue un rôle symétrique.

- Puisque chaque X_i est une variable binomiale,

$$E[X_1^2] = 1 + \frac{n(n-1)}{n^2} = 2 - \frac{1}{n} < 2.$$

Plus précisément

Variables aléatoires et moyennes

- Application : Problème du collectionneur
- Application : Tri Bucket Sort
- Application : Temps moyen du tri rapide

Algorithme du tri rapide

L'algorithme **Quicksort** est un algorithme de tri récursif, qui consiste, étant donné une liste $S = \{y_1, \cdots, y_n\}$ d'éléments distincts aux opérations suivantes :

- retourner S si S ne possède qu'un ou zéro élément ;
- choisir sinon un élément y de S, appelé **pivot** ;
  - comparer chaque élément à y, pour diviser S en 2 sous-listes :

    S_1, ceux qui sont inférieurs à y,
    S_2 ceux qui sont plus grands.
  - utiliser récursivement Quicksort sur chacune des listes S_1 et S_2 pour les trier ;
  - retourner le résultat S_1 concaténé avec y concaténé avec le résultat S_2.

Analyse

- Dans le pire des cas : par exemple
  - si la liste est dans l'ordre décroissant
  - et si l'on prend comme pivot systématiquement le premier élément,

  l'algorithme nécessite $\mathcal{O}(n^2)$ comparaisons.

- Supposons que
  - dans Quicksort on choisisse le pivot systématiquement selon une loi uniforme
  - et des tirages indépendants parmi les possibilités.

  Alors pour toute entrée, l'algorithme effectue un nombre moyen de comparaisons donné par $2n \log n + \mathcal{O}(n)$.

Analyse

- Dans le pire des cas : par exemple
  - si la liste est dans l'ordre décroissant
  - et si l'on prend comme pivot systématiquement le premier élément,

  l'algorithme nécessite $\mathcal{O}(n^2)$ comparaisons.

- Supposons que
  - dans Quicksort on choisisse le pivot systématiquement selon une loi uniforme
  - et des tirages indépendants parmi les possibilités.

  Alors pour toute entrée, l'algorithme effectue un nombre moyen de comparaisons donné par $2n \log n + \mathcal{O}(n)$.

- Supposons que
  - dans Quicksort on choisisse systématiquement le 1er élément.
  - les entrées sont choisies de façon uniforme parmi les permutations de $\{1, 2, \cdots, n\}$,

  Alors l'algorithme effectue un nombre moyen de comparaisons de l'ordre de $2n \log n + \mathcal{O}(n)$.

Preuve du premier résultat (1/2)

- Soient $y_1, y_2, \cdots, y_n$ les mêmes valeurs que les valeurs en entrée $x_1, x_2, \cdots, x_n$ mais dans l'ordre trié.

- Pour $i < j$, soit X_{ij} la variable aléatoire qui prend la valeur 1 si x_i et x_j sont comparés par l'algorithme, et 0 sinon.

- Le nombre total de comparaisons est donné par

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}.$$

- On a donc par linéarité de la moyenne

$$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}].$$

- Puisque X_{ij} prend les valeurs 0 ou 1, $E[X_{ij}]$ est la probabilité que x_i soit comparé à x_j.

Preuve du premier résultat (2/2)

- x_i est comparé à x_j si et seulement si x_i ou x_j est le premier pivot choisit parmi l'ensemble $Y^{ij} = \{x_i, x_{i+1}, \cdots, x_j\}$.
  - Si x_i (ou x_j) est le premier pivot choisit dans cette liste, alors x_i et x_j seront dans la même liste et donc seront comparés.
  - Symétriquement, si aucun des deux n'est le premier pivot choisit dans cette liste, alors x_i et x_j seront séparés dans deux listes distinctes et donc ne seront jamais comparés.
- Comme les pivots sont choisis de façon uniforme et indépendante, la probabilité que cela se produise est $2/(j-i+1)$.
- En posant $k = j - i + 1$, on obtient

$$\begin{aligned} E[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k} \\ &= \sum_{k=2}^{n} \sum_{i=1}^{n+1-k} \frac{2}{k} = \sum_{k=2}^{n} (n+1-k)\frac{2}{k} \\ &= (2n+2)\sum_{k=1}^{n} \tfrac{1}{k} - 4n = 2n\log(n) + \Theta(n). \end{aligned}$$

Plan

- On va maintenant introduire un ensemble d'inégalités, qui visent toutes à mesurer de combien une variable peut s'écarter de sa moyenne.

- Nous irons des inégalités les plus grossières aux plus fines. Ces inégalités sont omniprésentes dans l'étude des algorithmes.

Plus précisément

Theorem (Inégalité de Markov)

Soit X une variable aléatoire à valeurs positive ou nulles. Alors pour tout $a > 0$,

$$\Pr(X \geq a) \leq \frac{E[X]}{a}.$$

preuve

- Pour $a > 0$, posons

$$I = \begin{cases} 1 & \text{si } X \geq a, \\ 0 & \text{sinon.} \end{cases}$$

- Puisque $X \geq 0$,

$$I \leq \frac{X}{a}. \tag{3}$$

- Puisque I est une variable à valeur dans $\{0, 1\}$, $E[I] = \Pr(I = 1) = \Pr(X \geq a)$.

- En passant à la moyenne dans (3), on obtient

$$\Pr(X \geq a) = E[I] \leq E\left[\frac{X}{a}\right] = \frac{E[X]}{a}.$$

- Observons que l'on a égalité par exemple pour une loi telle que $\Pr(X = a) = 1$.

- Une façon qui peut être plus intuitive de comprendre l'inégalité est d'écrire

$$\Pr(X \geq \mu a) \leq \frac{1}{a},$$

pour tout $a > 0$, où $\mu = E[X]$.

Exemple : tirage de pièces

- Considérons une suite de tirage de pièces.
- Posons

$$X_i = \begin{cases} 1 & \text{si la ième pièce est pile} \\ 0 & \text{sinon.} \end{cases}$$

- Notons par $X = \sum_{i=1}^{n} X_i$ le nombre de piles parmi les n tirages.
- On a $E[X] = \sum_{i=1}^{n} E[X_i] = \frac{n}{2}$.
- L'inégalité de Markov donne, pour $\lambda > 0$,

$$\Pr(X \geq \lambda n) \leq \frac{E[X]}{\lambda n} = \frac{n}{2\lambda n} = \frac{1}{2\lambda}.$$

Ou encore

$$\Pr(X \geq \lambda \frac{n}{2}) \leq \frac{1}{\lambda}.$$

Plus précisément

Moments et déviations

- Inégalité de Markov
- Inégalité de Tchebychev
- Application : Problème du collectionneur
- Application : Calcul de la médiane
- Bornes de Chernoff
- Application : Test d'hypothèses
- Meilleures bornes pour certains cas
- Application : Équilibrage d'ensembles

La variance

La **variance** d'une variable aléatoire X est définie par

$$\mathrm{Var}[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2.$$

Quelques lois

Espérance **Variance**

- Loi de Bernoulli :

$\Pr(X = 1) = p$,
$\Pr(X = 0) = 1 - p$ $\quad E[X] = p.$ $\quad \mathrm{Var}[X] = p(1-p).$

- Loi Géométrique :

$\Pr(X = n) = (1-p)^{n-1}p.$ $\quad E[X] = 1/p.$ $\quad \mathrm{Var}[X] = (1-p)/p^2.$

- Loi Binomiale :

$\Pr(X = j) = C_n^j p^j (1-p)^{n-j}.$ $\quad E[X] = np.$ $\quad \mathrm{Var}[X] = np(1-p).$

Inégalité de Tchebyshev

Lorsque l'on possède une information sur la variance, on peut utiliser l'inégalité suivante, qui est plus fine.

Theorem (Inégalité de Tchebyshev)

Pour tout $a > 0$,

$$\Pr(|X - E[X]| \geq a) \leq \frac{\mathrm{Var}[X]}{a^2}.$$

Inégalité de Tchebyshev

Lorsque l'on possède une information sur la variance, on peut utiliser l'inégalité suivante, qui est plus fine.

Theorem (Inégalité de Tchebyshev)

Pour tout $a > 0$,

$$\Pr(|X - E[X]| \geq a) \leq \frac{\mathrm{Var}[X]}{a^2}.$$

Preuve :

- Observons que $\Pr(|X - E[X]| \geq a) = \Pr((X - E[X])^2 \geq a^2)$.
- Puisque $(X - E[X])^2$ est une variable aléatoire à valeurs positives ou nulles, en appliquant l'inégalité de Markov, on obtient

$$\Pr((X - E[X])^2 \geq a^2) \leq \frac{E[(X - E[X])^2]}{a^2} = \frac{\mathrm{Var}[X]}{a^2}.$$

□

Remarques

- On a égalité par exemple pour une loi telle que

$$\Pr(X \in \{\mu - a, \mu + a\}) = 1,$$

puisque $E[X] = \mu$, $\Pr(|X - E[X]| \geq a) = 1$, $\mathrm{Var}[X] = 1$.

- Une façon qui peut être plus intuitive de voir l'inégalité est d'écrire

$$\Pr(|X - \mu| \geq a\sigma) \leq \frac{1}{a^2},$$

pour tout $a > 0$, où $\mu = E[X]$, $\sigma^2 = \mathrm{Var}[X]$.

Exemple : tirage de pièces

- Reprenons l'exemple des tirages de pièces.
- Puisque X_i est une variable aléatoire à valeur $0-1$,

$$E[X_i^2] = \Pr(X_i = 1) = \frac{1}{2}$$

- On a $\mathrm{Var}[X_i] = E[X_i^2] - E[X_i]^2 = \frac{1}{2} - \frac{1}{4} = \frac{1}{4}$.
- Puisque les X_i sont indépendants,

$$\mathrm{Var}[X] = \sum_{i=1}^{n} \mathrm{Var}[X_i] = \frac{n}{4}.$$

- L'inégalité de Tchebychev donne pour $\lambda > 0$

$$\begin{aligned} \Pr(|X - \tfrac{n}{2}| \geq \lambda n) &\leq \frac{\mathrm{Var}[X]}{\lambda^2 n^2} \\ &\leq \frac{n/4}{\lambda^2 n^2} \\ &\leq \frac{1}{4\lambda^2 n}. \end{aligned}$$

- C'est plus fin que pour Markov.

Plus précisément

Moments et déviations

- Inégalité de Markov
- Inégalité de Tchebychev
- Application : Problème du collectionneur
- Application : Calcul de la médiane
- Bornes de Chernoff
- Application : Test d'hypothèses
- Meilleures bornes pour certains cas
- Application : Équilibrage d'ensembles

Application : Problème du collectionneur

- Revenons sur le problème du collectionneur.
- Rappelons que le nombre moyen de coupons est donné par X de moyenne nH_n, où $H_n = \sum_{i=1}^{n} 1/n = \log(n) + \mathcal{O}(1)$.
- L'inégalité de Markov donne donc

$$\Pr(X \geq 2nH_n) \leq \frac{1}{2}.$$

Application : Problème du collectionneur (1/2)

- Pour utiliser l'inégalité de Tchebychev, il nous faut la variance de X.
  - Rappelons que $X = \sum_{i=1}^{n} X_i$, où chaque X_i est une variable aléatoire géométrique de paramètre $(n - i + 1)/n$.
  - Les variables X_i sont indépendantes puisque le temps nécessaire pour collecter le $i$ème coupon ne dépend pas du temps utilisé pour le $i - 1$.
  - Par conséquent,

$$\mathrm{Var}[X] = \mathrm{Var}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathrm{Var}[X_i].$$

  - La variance d'une variable aléatoire de loi géométrique de paramètre p est $(1 - p)/p^2$.

Application : Problème du collectionneur (1/2)

- Pour utiliser l'inégalité de Tchebychev, il nous faut la variance de X.
  - Rappelons que $X = \sum_{i=1}^{n} X_i$, où chaque X_i est une variable aléatoire géométrique de paramètre $(n - i + 1)/n$.
  - Les variables X_i sont indépendantes puisque le temps nécessaire pour collecter le $i$ème coupon ne dépend pas du temps utilisé pour le $i - 1$.
  - Par conséquent,

$$\mathrm{Var}[X] = \mathrm{Var}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathrm{Var}[X_i].$$

  - La variance d'une variable aléatoire de loi géométrique de paramètre p est $(1 - p)/p^2$.

- On obtient en utilisant $\sum_{i=1}^{\infty} \left(\frac{1}{i}\right)^2 = \frac{\pi^2}{6}$.

$$\mathrm{Var}[X] = \mathrm{Var}\left[\sum_{i=1}^{n} X_i\right] \leq \sum_{i=1}^{n} \left(\frac{n}{n-i+1}\right)^2 = n^2 \sum_{i=1}^{n} \left(\frac{1}{n}\right)^2 \leq \frac{\pi^2 n^2}{6}$$

Application : Problème du collectionneur (2/2)

- On obtient

$$\mathrm{Var}[X] \leq \frac{\pi^2 n^2}{6}.$$

Application : Problème du collectionneur (2/2)

- On obtient

$$\mathrm{Var}[X] \leq \frac{\pi^2 n^2}{6}.$$

- L'inégalité de Tchebychev donne alors

$$\Pr(|X - nH_n| \geq nH_n) \leq \frac{n^2\pi^2/6}{(nH_n)^2} = \frac{\pi^2}{6(H_n)^2} = \mathcal{O}\left(\frac{1}{\log^2(n)}\right).$$

Application : Problème du collectionneur (2/2)

- On obtient

$$\mathrm{Var}[X] \leq \frac{\pi^2 n^2}{6}.$$

- L'inégalité de Tchebychev donne alors

$$\Pr(|X - nH_n| \geq nH_n) \leq \frac{n^2\pi^2/6}{(nH_n)^2} = \frac{\pi^2}{6(H_n)^2} = \mathcal{O}\left(\frac{1}{\log^2(n)}\right).$$

- Une fois encore, l'inégalité de Tchebychev donne un meilleur résultat que l'inégalité de Markov.

Argument d'union-bound

- Cependant, on peut faire mieux, par un argument d'union-bound pour ce problème.

Argument d'union-bound

- Cependant, on peut faire mieux, par un argument d'union-bound pour ce problème.
- Cela sera l'occasion de présenter de tels types d'arguments :

Argument d'union-bound

- Cependant, on peut faire mieux, par un argument d'union-bound pour ce problème.
- Cela sera l'occasion de présenter de tels types d'arguments :
  - La probabilité de ne pas obtenir le $i$ème coupon après $n\ln(n) + cn$ étapes est donné par

$$\left(1 - \frac{1}{n}\right)^{n(\ln n + c)} \leq e^{-\ln(n)+c} = \frac{1}{e^c n}.$$

Argument d'union-bound

- Cependant, on peut faire mieux, par un argument d'union-bound pour ce problème.
- Cela sera l'occasion de présenter de tels types d'arguments :
  - La probabilité de ne pas obtenir le $i$ème coupon après $n\ln(n) + cn$ étapes est donné par

$$\left(1 - \frac{1}{n}\right)^{n(\ln n + c)} \leq e^{-\ln(n)+c} = \frac{1}{e^c n}.$$

  - En notant E_i cet événement, et en utilisant la proposition (union-bound) ....

Argument d'union-bound

- Cependant, on peut faire mieux, par un argument d'union-bound pour ce problème.
- Cela sera l'occasion de présenter de tels types d'arguments :
  - La probabilité de ne pas obtenir le $i$ème coupon après $n\ln(n) + cn$ étapes est donné par

$$\left(1 - \frac{1}{n}\right)^{n(\ln n + c)} \leq e^{-\ln(n)+c} = \frac{1}{e^c n}.$$

  - En notant E_i cet événement, et en utilisant la proposition (union-bound) ....
  - ... on obtient que la probabilité qu'au moins un des coupons ne soit pas collecté se majore par

$$\Pr(\bigcup_{i\geq 1} E_i) \leq \sum_{i\geq 1} \Pr(E_i) = n\frac{1}{e^c n} = \frac{1}{e^c}.$$

Argument d'union-bound

- Cependant, on peut faire mieux, par un argument d'union-bound pour ce problème.
- Cela sera l'occasion de présenter de tels types d'arguments :
  - La probabilité de ne pas obtenir le $i$ème coupon après $n\ln(n) + cn$ étapes est donné par

$$\left(1 - \frac{1}{n}\right)^{n(\ln n + c)} \leq e^{-\ln(n)+c} = \frac{1}{e^c n}.$$

  - En notant E_i cet événement, et en utilisant la proposition (union-bound) ....
  - ... on obtient que la probabilité qu'au moins un des coupons ne soit pas collecté se majore par

$$\Pr(\bigcup_{i\geq 1} E_i) \leq \sum_{i\geq 1} \Pr(E_i) = n\frac{1}{e^c n} = \frac{1}{e^c}.$$

  - En prenant $c = \ln n$, la probabilité que tous les coupons ne soient pas collectés après $2n\ln n$ étapes est au plus $1/n$, ce qui donne une borne plus fine que l'inégalité de Tchebychev.

Plus précisément

Moments et déviations

- Inégalité de Markov
- Inégalité de Tchebychev
- Application : Problème du collectionneur
- Application : Calcul de la médiane
- Bornes de Chernoff
- Application : Test d'hypothèses
- Meilleures bornes pour certains cas
- Application : Équilibrage d'ensembles

Application : Calcul de la médiane

- Étant donné un ensemble $S = \{s_1, s_2, \cdots, s_n\}$ de n éléments parmi un univers totalement ordonné,
  - on appelle **médiane** de S un élément m de S tel que
    1. au moins $\lfloor n/2 \rfloor$ éléments de S soit inférieurs ou égaux à m,
    2. et au moins $\lfloor n/2 \rfloor + 1$ éléments de S soit supérieurs à m.

Application : Calcul de la médiane

- Étant donné un ensemble $S = \{s_1, s_2, \cdots, s_n\}$ de n éléments parmi un univers totalement ordonné,
  - on appelle **médiane** de S un élément m de S tel que
    1. au moins $\lfloor n/2 \rfloor$ éléments de S soit inférieurs ou égaux à m,
    2. et au moins $\lfloor n/2 \rfloor + 1$ éléments de S soit supérieurs à m.

- Autrement dit, si S est trié, m est l'élément d'indice $\lceil n/2 \rceil$.

Application : Calcul de la médiane

- Étant donné un ensemble $S = \{s_1, s_2, \cdots, s_n\}$ de n éléments parmi un univers totalement ordonné,
  - on appelle **médiane** de S un élément m de S tel que
    1. au moins $\lfloor n/2 \rfloor$ éléments de S soit inférieurs ou égaux à m,
    2. et au moins $\lfloor n/2 \rfloor + 1$ éléments de S soit supérieurs à m.

- Autrement dit, si S est trié, m est l'élément d'indice $\lceil n/2 \rceil$.

- Bien entendu, l'idée est d'utiliser un algorithme plus efficace que $\mathcal{O}(n \log(n))$, ce qui peut être atteint en triant les données, puis en retournant cet élément.

Application : Calcul de la médiane

- Étant donné un ensemble $S = \{s_1, s_2, \cdots, s_n\}$ de n éléments parmi un univers totalement ordonné,
  - on appelle **médiane** de S un élément m de S tel que
    1. au moins $\lfloor n/2 \rfloor$ éléments de S soit inférieurs ou égaux à m,
    2. et au moins $\lfloor n/2 \rfloor + 1$ éléments de S soit supérieurs à m.

- Autrement dit, si S est trié, m est l'élément d'indice $\lceil n/2 \rceil$.

- Bien entendu, l'idée est d'utiliser un algorithme plus efficace que $\mathcal{O}(n \log(n))$, ce qui peut être atteint en triant les données, puis en retournant cet élément.

- On connaît une solution déterministe en $\mathcal{O}(n)$ opérations. On présente ici une version randomisée simple de même complexité.

- L'idée de l'algorithme est d'utiliser de l'échantillonnage :

- L'idée de l'algorithme est d'utiliser de l'échantillonnage :
  - on essaye de trouver deux éléments d et u tels que $d \leq m \leq u$,

- L'idée de l'algorithme est d'utiliser de l'échantillonnage :
  - on essaye de trouver deux éléments d et u tels que $d \leq m \leq u$,
  - et tel que le nombre d'éléments entre d et u soit faible, c'est-à-dire en $o(n/\log(n))$.

- L'idée de l'algorithme est d'utiliser de l'échantillonnage :
  - on essaye de trouver deux éléments d et u tels que $d \leq m \leq u$,
  - et tel que le nombre d'éléments entre d et u soit faible, c'est-à-dire en $o(n/\log(n))$.

- Notons $C = \{s | d \leq s \leq u\}$ les éléments entre d et u.

- L'idée de l'algorithme est d'utiliser de l'échantillonnage :
  - on essaye de trouver deux éléments d et u tels que $d \leq m \leq u$,
  - et tel que le nombre d'éléments entre d et u soit faible, c'est-à-dire en $o(n/\log(n))$.

- Notons $C = \{s | d \leq s \leq u\}$ les éléments entre d et u.

- Si l'on arrive à trouver deux éléments comme cela, il est facile de trouver la médiane en temps linéaire :

- L'idée de l'algorithme est d'utiliser de l'échantillonnage :
  - on essaye de trouver deux éléments d et u tels que $d \leq m \leq u$,
  - et tel que le nombre d'éléments entre d et u soit faible, c'est-à-dire en $o(n/\log(n))$.

- Notons $C = \{s | d \leq s \leq u\}$ les éléments entre d et u.

- Si l'on arrive à trouver deux éléments comme cela, il est facile de trouver la médiane en temps linéaire :

  - on parcours la liste S, et

- L'idée de l'algorithme est d'utiliser de l'échantillonnage :
  - on essaye de trouver deux éléments d et u tels que $d \leq m \leq u$,
  - et tel que le nombre d'éléments entre d et u soit faible, c'est-à-dire en $o(n/\log(n))$.

- Notons $C = \{s | d \leq s \leq u\}$ les éléments entre d et u.

- Si l'on arrive à trouver deux éléments comme cela, il est facile de trouver la médiane en temps linéaire :

  - on parcours la liste S, et
  - on compte le nombre ℓ_d d'éléments inférieurs à d,

- L'idée de l'algorithme est d'utiliser de l'échantillonnage :
  - on essaye de trouver deux éléments d et u tels que $d \leq m \leq u$,
  - et tel que le nombre d'éléments entre d et u soit faible, c'est-à-dire en $o(n/\log(n))$.

- Notons $C = \{s | d \leq s \leq u\}$ les éléments entre d et u.

- Si l'on arrive à trouver deux éléments comme cela, il est facile de trouver la médiane en temps linéaire :

  - on parcours la liste S, et
  - on compte le nombre ℓ_d d'éléments inférieurs à d,
  - et on trie l'ensemble C : puisque $|C| = o(n/\log(n))$, trier C se fait en temps $o(n)$ par n'importe quel algorithme de tri qui fonctionne en temps $\mathcal{O}(m\log(m))$ pour m éléments.

- L'idée de l'algorithme est d'utiliser de l'échantillonnage :
  - on essaye de trouver deux éléments d et u tels que $d \leq m \leq u$,
  - et tel que le nombre d'éléments entre d et u soit faible, c'est-à-dire en $o(n/\log(n))$.

- Notons $C = \{s | d \leq s \leq u\}$ les éléments entre d et u.

- Si l'on arrive à trouver deux éléments comme cela, il est facile de trouver la médiane en temps linéaire :

  - on parcours la liste S, et
  - on compte le nombre ℓ_d d'éléments inférieurs à d,
  - et on trie l'ensemble C : puisque $|C| = o(n/\log(n))$, trier C se fait en temps $o(n)$ par n'importe quel algorithme de tri qui fonctionne en temps $\mathcal{O}(m\log(m))$ pour m éléments.
  - L'élément d'indice $\lfloor n/2 \rfloor - \ell_d + 1$ de C est alors m.

- Pour trouver d et u, on échantillonne avec remplacement un (multi-)ensemble R de $\lceil n^{3/4} \rceil$ éléments :

- Pour trouver d et u, on échantillonne avec remplacement un (multi-)ensemble R de $\lceil n^{3/4} \rceil$ éléments :
  - c'est-à-dire que l'on pioche au hasard uniformément ce nombre d'éléments parmi S.

- Pour trouver d et u, on échantillonne avec remplacement un (multi-)ensemble R de $\lceil n^{3/4} \rceil$ éléments :
  - c'est-à-dire que l'on pioche au hasard uniformément ce nombre d'éléments parmi S.
- On souhaite que chaque étape fonctionne avec **grande probabilité**,

- Pour trouver d et u, on échantillonne avec remplacement un (multi-)ensemble R de $\lceil n^{3/4} \rceil$ éléments :
  - c'est-à-dire que l'on pioche au hasard uniformément ce nombre d'éléments parmi S.
- On souhaite que chaque étape fonctionne avec **grande probabilité**,
  - c'est-à-dire avec une probabilité au moins $1 - \mathcal{O}(1/n^c)$ pour une constante c.

- Pour trouver d et u, on échantillonne avec remplacement un (multi-)ensemble R de $\lceil n^{3/4} \rceil$ éléments :
  - c'est-à-dire que l'on pioche au hasard uniformément ce nombre d'éléments parmi S.
- On souhaite que chaque étape fonctionne avec **grande probabilité**,
  - c'est-à-dire avec une probabilité au moins $1 - \mathcal{O}(1/n^c)$ pour une constante c.
- Pour garantir qu'avec grande probabilité m soit entre d et u, on fixe d et u comme étant respectivement les $\lfloor n^{3/4}/2 - \sqrt{n} \rfloor$ème et $\lfloor n^{3/4}/2 + \sqrt{n} \rfloor$ ème éléments de R.

L'algorithme

1. Choisir $\lceil n^{3/4} \rceil$ éléments dans S, avec un tirage uniforme et avec remise. Soit R les éléments obtenus.
2. Trier R.
3. Soit d le $\lfloor n^{3/4}/2 - \sqrt{n} \rfloor$ élément de R.
4. Soit u le $\lfloor n^{3/4}/2 + \sqrt{n} \rfloor$ élément de R.
5. En comparant chaque élément de S à d et u, calculer les ensembles $C = \{x \in S | d \leq x \leq u\}$, $\ell_d = |\{x \in S | x < d\}|$ et $\ell_u = |\{x \in S | x > u\}|$.
6. Si $\ell_d > n/2$ ou $\ell_u > n/2$ alors échouer.
7. Si $|C| \leq 4n^{3/4}$ alors trier S sinon échouer.
8. Retourner le $\lfloor n/2 \rfloor - \ell_d + 1$ élément de C.

Proposition

La probabilité que l'algorithme échoue est en $\mathcal{O}(n^{-1/4})$.

- On considère les trois événements

$$E_1 : Y_1 = |\{r \in R | r \leq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_2 : Y_2 = |\{r \in R | r \geq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_3 : |C| > 4n^{3/4}.$$

- On considère les trois événements

$$E_1 : Y_1 = |\{r \in R | r \leq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_2 : Y_2 = |\{r \in R | r \geq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_3 : |C| > 4n^{3/4}.$$

- L'algorithme termine si au moins l'un des trois événements E_1, E_2 ou E_3 se produit :

- On considère les trois événements

$$E_1 : Y_1 = |\{r \in R | r \leq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_2 : Y_2 = |\{r \in R | r \geq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_3 : |C| > 4n^{3/4}.$$

- L'algorithme termine si au moins l'un des trois événements E_1, E_2 ou E_3 se produit :
  - en effet, un échec à l'étape 7 correspond à l'événement E_3.

- On considère les trois événements

$$E_1 : Y_1 = |\{r \in R | r \leq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_2 : Y_2 = |\{r \in R | r \geq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_3 : |C| > 4n^{3/4}.$$

- L'algorithme termine si au moins l'un des trois événements E_1, E_2 ou E_3 se produit :
  - en effet, un échec à l'étape 7 correspond à l'événement E_3.
  - Un échec à l'étape 6 implique $\ell_d > n/2$ ou $\ell_u > n/2$.

- On considère les trois événements

$$E_1 : Y_1 = |\{r \in R | r \leq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_2 : Y_2 = |\{r \in R | r \geq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_3 : |C| > 4n^{3/4}.$$

- L'algorithme termine si au moins l'un des trois événements E_1, E_2 ou E_3 se produit :
  - en effet, un échec à l'étape 7 correspond à l'événement E_3.
  - Un échec à l'étape 6 implique $\ell_d > n/2$ ou $\ell_u > n/2$.
  - Chacun de ces cas est équivalent à E_1 ou E_2.

- On considère les trois événements

$$E_1 : Y_1 = |\{r \in R | r \leq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_2 : Y_2 = |\{r \in R | r \geq m\}| < n^{3/4}/2 - \sqrt{n}$$

$$E_3 : |C| > 4n^{3/4}.$$

- L'algorithme termine si au moins l'un des trois événements E_1, E_2 ou E_3 se produit :
  - en effet, un échec à l'étape 7 correspond à l'événement E_3.
  - Un échec à l'étape 6 implique $\ell_d > n/2$ ou $\ell_u > n/2$.
  - Chacun de ces cas est équivalent à E_1 ou E_2.

- Par une union-bound, il suffit de montrer que $\Pr(E_i)$ est en $\mathcal{O}(n^{-1/4})$.

- On a $\Pr(E_1) \leq \frac{1}{4} n^{-1/4}$.

- On a $\Pr(E_1) \leq \frac{1}{4} n^{-1/4}$.
  - En effet, considérons la variable aléatoire X_i qui vaut 1 si le $i$ème échantillon est inférieur ou égal à la médiane m, et 0 sinon.
  - Les X_i sont indépendants, puisque l'on procède avec remplacement.
  - Puisqu'il y a $(n-1)/2+1$ éléments inférieurs ou égaux à l, on a $\Pr(X_i = 1) = \frac{(n-1)/2+1}{n} = \frac{1}{2} + \frac{1}{2n}$.
  - L'événement E_1 est équivalent à

$$Y_1 = \sum_{i=1}^{n^{3/4}} X_i < \frac{1}{2} n^{3/4} - \sqrt{n}.$$

    Y_1 est la somme de tirages de Bernoulli, il suit donc une loi binomiale de paramètres $n^{3/4}$ et $1/2 + 1/2n$.
  - L'inégalité de Tchebychev donne alors

$$\begin{aligned} \Pr(E_1) &= \Pr(Y_1 < \tfrac{1}{2} n^{3/4} - \sqrt{n}) \\ &\leq \Pr(|Y_1 - E[Y_1]| > \sqrt{n}) \\ &\leq \frac{\mathrm{Var}[Y_1]}{n} < \frac{1/4 n^{3/4}}{n} \leq \frac{1}{4} n^{-1/4}. \end{aligned}$$

- On a $\Pr(E_3) \leq \frac{1}{2} n^{-1/4}$.

- On a $\Pr(E_3) \leq \frac{1}{2} n^{-1/4}$.

  - si E_3 se produit, soit au moins $2n^{3/4}$ éléments sont plus grands que m, soit au moins $2n^{3/4}$ sont plus petits que m.
  - Bornons le premier, puisque l'autre est symétrique.
  - L'ordre de u dans l'ensemble S trié doit être au moins $\frac{1}{2}n + 2n^{3/4}$, et donc R possède au moins $\frac{1}{2}n^{3/4} - \sqrt{n}$ éléments parmi les $\frac{1}{2}n - 2n^{3/4}$ éléments les plus grands de S.
  - Soit X le nombre d'éléments parmi les $\frac{1}{2}n - 2n^{3/4}$ éléments les plus grands de S.
  - X s'écrit $X = \sum_{i=1}^{n^{3/4}} X_i$, où X_i vaut 1 si le $i$ème élément pioché est parmi les $\frac{1}{2}n - 2n^{3/4}$ éléments les plus grands de S, et 0 sinon.
  - X, somme de lois de Bernoulli, suit une loi binomiale, et l'inégalité de Tchebychev permet de majorer la probabilité de cet événement par

$$\begin{aligned} \Pr(X \geq \tfrac{1}{2}n^{3/4} - \sqrt{n}) &\leq \Pr(|X - E[X]| \geq \sqrt{n}) \\ &\leq \frac{\mathrm{Var}[X]}{n} \\ &< \frac{1/4 n^{3/4}}{n} \leq \tfrac{1}{4} n^{-1/4}. \end{aligned}$$

De Monte Carlo à Las Vegas

- Observons qu'en répétant cet algorithme jusqu'à ce qu'il réussisse, on obtient un algorithme de **Las Vegas** :

De Monte Carlo à Las Vegas

- Observons qu'en répétant cet algorithme jusqu'à ce qu'il réussisse, on obtient un algorithme de **Las Vegas** :
  - il retournerait toujours une réponse correcte, mais son temps de réponse est aléatoire.

De Monte Carlo à Las Vegas

- Observons qu'en répétant cet algorithme jusqu'à ce qu'il réussisse, on obtient un algorithme de **Las Vegas** :
  - il retournerait toujours une réponse correcte, mais son temps de réponse est aléatoire.
- En fait, le nombre d'itérations de l'algorithme serait alors donné par une loi géométrique.

De Monte Carlo à Las Vegas

- Observons qu'en répétant cet algorithme jusqu'à ce qu'il réussisse, on obtient un algorithme de **Las Vegas** :
  - il retournerait toujours une réponse correcte, mais son temps de réponse est aléatoire.
- En fait, le nombre d'itérations de l'algorithme serait alors donné par une loi géométrique.
- On peut assez facilement se convaincre que l'algorithme obtenu fonctionnerait en temps moyen linéaire.

Plus précisément

Moments et déviations

- Inégalité de Markov
- Inégalité de Tchebychev
- Application : Problème du collectionneur
- Application : Calcul de la médiane
- Bornes de Chernoff
- Application : Test d'hypothèses
- Meilleures bornes pour certains cas
- Application : Équilibrage d'ensembles

Bornes de Chernoff

- Les bornes de Chernoff, sont en fait une famille d'inégalités obtenues sur un même principe.

Principe de base :(1/2)

- La fonction génératrice des moments d'une variable aléatoire X est $M_X(t) = E[e^{tX}]$.

Principe de base :(1/2)

- La fonction génératrice des moments d'une variable aléatoire X est $M_X(t) = E[e^{tX}]$.

- La propriété élémentaire que l'on va utiliser est la suivante :

Principe de base :(1/2)

- La fonction génératrice des moments d'une variable aléatoire X est $M_X(t) = E[e^{tX}]$.

- La propriété élémentaire que l'on va utiliser est la suivante :

  - Si X et Y sont des variables indépendantes, alors

$$M_{X+Y}(t) = M_X(t)M_Y(t).$$

Principe de base :(1/2)

- La fonction génératrice des moments d'une variable aléatoire X est $M_X(t) = E[e^{tX}]$.

- La propriété élémentaire que l'on va utiliser est la suivante :

  - Si X et Y sont des variables indépendantes, alors

$$M_{X+Y}(t) = M_X(t)M_Y(t).$$

- Preuve :

$$M_{X+Y}(t) = E[e^{t(X+Y)}] = E[e^{tX}e^{tY}] = E[e^{tX}]E[e^{tY}]$$

$$M_{X+Y}(t) = M_X(t)M_Y(t).$$

□

Principe de base :(2/2)

- Les bornes de Chernoff sont alors obtenues en appliquant l'inégalité de Markov sur e^{tX} pour un t bien choisi.

- En effet, par l'inégalité de Markov, on obtient : pour tout $t > 0$,

$$\Pr(X \geq a) = \Pr(e^{tX} \geq e^{ta}) \leq \frac{E[e^{tX}]}{e^{ta}}.$$

- En particulier,

$$\Pr(X \geq a) \leq \min_{t>0} \frac{E[e^{tX}]}{e^{ta}}.$$

Principe de base :(2/2)

- Les bornes de Chernoff sont alors obtenues en appliquant l'inégalité de Markov sur e^{tX} pour un t bien choisi.

- En effet, par l'inégalité de Markov, on obtient : pour tout $t > 0$,

$$\Pr(X \geq a) = \Pr(e^{tX} \geq e^{ta}) \leq \frac{E[e^{tX}]}{e^{ta}}.$$

- En particulier,

$$\Pr(X \geq a) \leq \min_{t>0} \frac{E[e^{tX}]}{e^{ta}}.$$

- De façon similaire, pour $t < 0$,

$$\Pr(X \leq a) = \Pr(e^{tX} \geq e^{ta}) \leq \frac{E[e^{tX}]}{e^{ta}}.$$

- Donc $\Pr(X \leq a) \leq \min_{t<0} \frac{E[e^{tX}]}{e^{ta}}$.

Theorem

On considère des variables $X_1, X_2, \cdots, X_n$ à valeurs dans $\{0,1\}$ indépendantes telles que $\Pr(X_i) = p_i$. Soit $X = \sum_{i=1}^{n} X_i$, et $\mu = E[X]$. Alors on a les bornes de Chernoff suivantes.

- *Pour tout $\delta > 0$*

$$\Pr(X \geq (1+\delta)\mu) \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\mu}.$$

- *Pour tout $0 < \delta \leq 1$*

$$\Pr(X \geq (1+\delta)\mu) \leq e^{-\mu\delta^2/3}.$$

- *Pour $R \geq 6\mu$,*

$$\Pr(X \geq R) \leq 2^{-R}.$$

Preuve

- Écrivons

$$M_{X_i}(t) = E[e^{tX_i}] = p_i e^t + (1-p_i) = 1 + p_i(e^t - 1) \leq e^{p_i(e^t-1)}.$$

- Par conséquent,

$$M_X(t) = \prod_{i=1}^{n} M_{X_i}(t) \leq \prod_{i=1}^{n} e^{p_i(e^t-1)} = exp(\sum_{i=1}^{n} p_i(e^t-1)) = e^{(e^t-1)\mu}.$$

- Par l'inégalité de Markov, pour tout $t > 0$, on a

$$\begin{aligned} \Pr(X \geq (1+\delta)\mu) &= \Pr(e^{tX} \geq e^{t(1+\delta)\mu}) \\ &\leq \frac{E[e^{tX}]}{e^{t(1+\delta)\mu}} \\ &= \frac{M_X(t)}{e^{t(1+\delta)\mu}} \\ &\leq \frac{e^{(e^t-1)\mu}}{e^{t(1+\delta)\mu}}. \end{aligned}$$

- Fixons $t = \ln(1+\delta) > 0$ pour obtenir

$$\Pr(X \geq (1+\delta)\mu) \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu,$$

c'est-à-dire la première inégalité recherchée.

Preuve

- La deuxième inégalité s'obtient en montrant que

$$\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \leq e^{-\mu\delta^2/3}$$

  pour $0 < \delta \leq 1$ par étude de fonction.

- La troisième, consiste à poser $R = (1+\delta)\mu$. Pour $R \geq 6\mu$, on a $\delta = R/\mu - 1 \geq 5$. Donc,

$$\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \leq \left(\frac{e}{1+\delta}\right)^{(1+\delta)\mu} \leq \left(\frac{e}{6}\right)^{R} \leq 2^{-R}.$$

En dessous de la moyenne...

Theorem

On considère des variables $X_1, X_2, \cdots, X_n$ à valeurs dans $\{0,1\}$ indépendantes telles que $\Pr(X_i) = p_i$. Soit $X = \sum_{i=1}^{n} X_i$, et $\mu = E[X]$. Alors on a les bornes de Chernoff suivantes.

- *Pour tout* $0 < \delta < 1$

$$\Pr(X \leq (1-\delta)\mu) \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu};$$

- *Pour tout* $0 < \delta < 1$

$$\Pr(X \leq (1-\delta)\mu) \leq e^{-\mu\delta^2/2}.$$

- Par un “union-bound”, on obtient :

- **Corollary**

  On considère des variables $X_1, X_2, \cdots, X_n$ à valeurs dans $\{0, 1\}$ indépendantes telles que $\Pr(X_i) = p_i$. Soit $X = \sum_{i=1}^{n} X_i$, et $\mu = E[X]$. Pour tout $0 < \delta < 1$,

$$\Pr(|X - \mu| \geq \delta\mu) \leq 2e^{-\mu\delta^2/3}.$$

Tirage de pièces

- Reprenons l'exemple des tirages de pièces.
- Pour $\lambda < \frac{1}{2}$

$$\Pr(|X - \frac{n}{2}| \geq \lambda n) \leq 2\exp(-\frac{1}{3}\frac{n}{2}4\lambda^2) = 2e^{-2/3n\lambda^2},$$

ce qui est infiniment mieux que Tchebychev et Markov.

Tirage de pièces

- Reprenons l'exemple des tirages de pièces.
- Pour $\lambda < \frac{1}{2}$

$$\Pr(|X - \frac{n}{2}| \geq \lambda n) \leq 2\exp(-\frac{1}{3}\frac{n}{2}4\lambda^2) = 2e^{-2/3n\lambda^2},$$

ce qui est infiniment mieux que Tchebychev et Markov.

- En fait, on peut aussi voir que X est fortement centré autour de sa moyenne.

$$\Pr(|X-\frac{n}{2}| \geq \lambda\sqrt{n\ln n}) \leq 2\exp(-\frac{1}{3}\frac{n}{2}\frac{4\lambda^2\ln n}{n}) = 2n^{-2/3\lambda^2} = \frac{2}{n^{2/3\lambda^2}}$$

Ce qui donne $2/n$ pour $\lambda = \sqrt{6}/2$.

Plus précisément

Moments et déviations

- Inégalité de Markov
- Inégalité de Tchebychev
- Application : Problème du collectionneur
- Application : Calcul de la médiane
- Bornes de Chernoff
- Application : Test d'hypothèses
- Meilleures bornes pour certains cas
- Application : Équilibrage d'ensembles

Application : Test d'hypothèses

- Supposons que l'on veuille évaluer une probabilité p de mutation de virus dans une population.
- Étant donné un virus, on peut déterminer s'il a muté, mais le test est coûteux.
- Supposons que l'on fasse le test sur une population de taille n, et que l'on observe que parmi cet échantillon que $X = \overline{p}n$ virus ont muté.
- Si n est suffisamment grand, on peut espérer que $\overline{p}$ est proche de p.

- Formellement, cette intuition est capturée par la notion d'intervalle de confiance.

- **Definition (Intervalle de confiance)**

  Un $1-\gamma$-**intervalle de confiance** pour un paramètre p est un intervalle $[\overline{p}-\delta, \overline{p}+\delta]$ tel que

$$\Pr(p \in [\overline{p}-\delta, \overline{p}+\delta]) \geq 1-\gamma.$$

Theorem

On peut déterminer un $1-\gamma$*-intervalle de confiance en utilisant un échantillon de taille*

$$n = \lceil 3\frac{1}{\delta^2} \ln \frac{2}{\gamma} \rceil.$$

Preuve

- On considère $X = \sum_i X_i$, avec $\Pr(X_i = 1) = p$, $\Pr(X_i = 0) = 1 - p$.
- Sur n expériences, on aura $E[X] = np$, et donc on va considérer $\overline{p} = X/n$ comme estimation de p.
- Les bornes de Chernoff disent que

$$\Pr(X/n > p+\delta) = \Pr(X > np+n\delta) = \Pr(X > np(1+\delta/p)) \leq e^{-np\delta}$$

  Soit $\Pr(X/n > p + \delta) \leq e^{-n\delta^2/(3p)}$
- Symétriquement $\Pr(X/n < p - \delta) \leq e^{-n\delta^2/(2p)}$
- On ne connaît pas p, mais on sait qu'il est plus petit que 1.
- Donc

$$\Pr(X/n > p + \delta) \leq e^{-n\delta^2/(3)}$$

$$\Pr(X/n < p - \delta) \leq e^{-n\delta^2/(2)}$$

- Soit, par un union bound, $\Pr(p \notin [X/n - \delta, X/n + \delta]) \leq 2e^{-n\delta^2/3}$.
- Il suffit de prendre $n \geq \lceil 3\frac{1}{\delta^2} \ln \frac{2}{\gamma} \rceil$, pour que cette quantité soit inférieure à γ.