MAPSI – Examen réparti 1

Durée : 2 heures

Seuls documents autorisés : Calculatrice, antisèche recto-verso, tables de lois.
– Barème indicatif –

Exercice 1 (3 pts) – **Indépendance**

Deux logiciels X et Y implémentent deux fonctions structurellement indépendantes de l'infrastructure d'une grande organisation de vente en ligne. Soit trois variables aléatoires :

- A la variable associée à la proposition "les soldes sont en cours",
- B la variable "X est en marche",
- C la variable "Y est en marche".

	a_1		a_2	
	b_1	b_2	b_1	b_2
c_1	0,288	0,192	0,016	0,024
c_2	0,192	0,128	0,064	0,096

Les trois variables ont modalités $\{a_1, a_2\}$, $\{b_1, b_2\}$, $\{c_1, c_2\}$, correspondant à des évaluations vrai et faux de chaque proposition. La probabilité jointe de ces variables est fournie par le tableau ci-dessus.

Q 1.1 (1 pt) Indépendence. Déterminer si B et C sont indépendantes.

On peut tester indépendance en montrant si $P(B,C) = P(B)P(C)$. En marginalisant, on calcule ces tableaux de probabilités :

$P(B,C)$

	b_1	b_2
c_1	0,304	0,216
c_2	0,256	0,224

$P(B)$

b_1	0,56
b_2	0,44

$P(C)$

c_1	0,52
c_2	0,48

On trouve facilement que $P(B,C) \neq P(B)P(C)$, e.g. $P(B = b_1, C = c_1) = 0,304 \neq P(B = b_1)P(C = c_1) = 0,56 \cdot 0,52 = 0.2912$.

Q 1.2 (1 pt) Indépendence conditionnelle Déterminer le tableau $P(B,C|A)$ et vérifier que B et C sont indépendantes conditionnellement à A.

On calcule $P(B,C|A) = P(A,B,C)/P(A)$.

$P(A)$

a_1	0,80
a_2	0,20

$P(B,C|A)$

	a_1		a_2	
	b_1	b_2	b_1	b_2
c_1	0,36	0,24	0,08	0,12
c_2	0,24	0,16	0,32	0,48

Par marginalisation :

$P(B|A)$

	a_1	a_2
b_1	0,60	0,40
b_2	0,40	0,60

$P(C|A)$

	a_1	a_2
c_1	0,60	0,40
c_2	0,20	0,80

On observe que en effet, $P(B, C|A) = P(B|A)P(C|A)$. Alors B et C sont indépendantes conditionnellement à A.
==========
Correction alternative : on voit que les deux sous-tableaux (pour a1 et a2) sont chacun indépendant.

Q 1.3 (1 pt) Donner une explication plausible au fait que B et C ne soient pas indépendantes malgré l'indépendance structurelle de X et Y.

B et C sont dépendent en surface car il partagent la dépendance de A. Quand A est figé, on observe en effet qu'il n'y a pas de dépendance entre eux.

Exercice 2 (4.75 pts) – Tirage de roulette numérique et hypothèse fumeuse

Dans un casino, face à une roulette électronique, un client statisticien voudrait mieux comprendre le système pour maximiser ses chances... Il a lu sur le site du fabricant que le tirage de chaque numéro émis par la roulette électronique était effectué selon une binomiale de paramètres $p, N = 16$.

Q 2.1 (0.25 pt) Combien de tirages différents sont possibles selon cette loi ?

17

Q 2.2 (1 pt) Nous observons les tirages suivants : $10, 10, 5, 8, 12, 8, 7, 12, 6$. En faisant l'hypothèse que les tirages sont indépendants, donner une estimation du paramètre p au sens du maximum de vraisemblance.
Rappel $X \sim \mathcal{B}(p, N)$, $\quad p(X = k) = C_N^k p^k (1-p)^{N-k} = \frac{N!}{k!(N-k)!} p^k (1-p)^{N-k}$

p=78/(78+56) = 0.54167

Q 2.3 (1.5 pt) Le résultat ne tombe pas rond... Et notre statisticien est convaincu que le créateur de la machine a dû choisir une probabilité simple. Il pense que c'est 0.5 : cela vous semble-t-il raisonnable ? (à un niveau de confiance 0.95 et en prenant $\sigma^2 = 4$ et en se rappelant que l'espérance de la binomiale est Np)

$$\bar{X} = 8.667 \qquad \mu = Np = 8$$

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{(9)}} \sim \mathcal{N}(0, 1)$$

Vu l'énoncé un peu flou, on peut faire le test symétrique ou juste à la hausse
(1) à la hausse : $p(Z > z_\alpha) = 0.05 \Rightarrow$ valeur limite $= 1.645 * 2./3 + 8 = 9.09666$
(2) symétrique : $p(Z > z_\alpha) = 0.025 \Rightarrow$ valeur limite $= 1.96 * 2./3 + 8 = 9.30666$
Dans les deux cas, on valide l'hypothèse. Même si évidemment, on ne peut pas calculer de puissance

Q 2.4 (0.5 pt) Calculer : $\dfrac{p(X = k+1)}{p(X = k)}$

$$\frac{p(X = k+1)}{p(X = k)} = \frac{p}{1-p} \frac{n-k}{k+1}$$

Q 2.5 (1.5 pt) Sur quel chiffre faut-il miser ? Formuler le problème, et en vous appuyant sur la question précédente, donner votre réponse.

Il faut trouver $k^\star = \arg\max_k p(X = k)$.
D'après la question précédente, et avec $p = 0,5$, on a

$$\frac{p(X = k+1)}{pX = k} = \frac{p}{1-p}\frac{n-k}{k+1} = \frac{16-k}{k+1} = \frac{17}{k+1} - 1$$

Du coup, on trouve naturellement $k^\star = 8$ car $\frac{17}{k+1} - 1$ supérieur à 1 pour $k \leq 7$, puis inférieur à 1 pour $k \geq 8$.

Exercice 3 (4.5 pts) – **Review Spam**

Les revues d'utilisateurs sur Internet constituent une ressource importante comme déclencheur d'achat. En conséquence, ce système est fréquemment attaqué, notamment par des rédactions massives de fausses revues visant à sur-valoriser ou au contraire dénigrer un produit. Les experts estiment à 2% le taux de spam parmi les revues. Des chercheurs ont mis au point un détecteur de fausses revues, dûment évalué par des experts, présentant les caractéristiques suivantes :

	Spam (s)	Non spam ($\bar{s}$)
Alarme (a)	95%	10%
Pas d'alarme ($\bar{a}$)	5%	90%

Q 3.1 (0.5 pt) Étant donné les deux variables aléatoires en présence, à savoir A (alarme du système de détection) et S (spam selon les experts), que représente le tableau ci-dessus ?

Proba conditionnelle :

$$p(A|S)$$

Q 3.2 (1.5 pt) Ce type de système s'évalue en précision et rappel : la précision correspond au taux de vrai spam parmi les alarmes et le rappel (aussi appelé couverture) qui mesure le taux de spam détecté.
Donner la formulation de la précision et du rappel puis effectuer l'application numérique.
En s'appuyant sur le rappel et la précision calculés, comment interpréter les performances d'un système qui retireraient toutes les revues qui ont levé une alarme ?

$$precision = p(S = s|A = a) = p(A = a|S = s)p(S = s)/p(A = a)$$

et

$$p(A = a) = p(A = a|S = s)p(S = s) + p(A = a|S = \bar{s})p(S = \bar{s}) = 0.950.02 + 0.10.98 = 0.117$$

$$precision = p(S = s|A = a) = 0.950.02/0.117 = 0.1624$$

$$rappel = p(A = a|S = s) = 0.95$$

Un système qui retire toutes les revues sous alarme élimine 95% du spam... Mais supprime alors $\sim 84\%$ de revues valides dans la purge !

Q 3.3 Une autre étude portant sur 1 million de compte du site nozamA pointe l'importance de la prise en compte de la taille des comptes (le nombre de messages écrits par les utilisateurs). Les auteurs considèrent qu'un compte compromis n'émet que du spam (et respectivement qu'un compte sain n'en émet pas). Après pré-traitements et regroupement, l'étude propose de regrouper les utilisateurs en 6 catégories, et calcule les statistiques suivantes :

Catégorie de compte utilisateur :	1	2	3	4	5	6	Tot
Nombre moyen de messages par compte :	100	50	20	10	5	1	
Nombre de comptes :	10k	20k	50k	80k	100k	740k	1M
Taux de spam sur la catégorie :	0.01	0.004	0.003	0.001	0.0003	0.0001	

Q 3.3.1 (1 pt) En moyenne, combien de revues sont écrites par chaque utilisateur ? En moyenne toujours, combien de spam sont émis par chaque utilisateur (vous pourrez d'abord déterminer le nombre moyen de spam par compte dans chaque catégorie) ? Cela est-il cohérent avec la question précédente ?

Q 3.3.2 (1.5 pt) Un expert pense que le taux de spameurs est proportionnel au nombre de messages émis. Selon cette hypothèse, le nombre de spameurs attendu par catégorie serait le suivant :

Catégorie de compte utilisateur :	1	2	3	4	5	6	Tot
Nombre de spameurs dans la catégorie :	102	102	102	82	51	75	514

Tester cette hypothèse à un niveau de confiance 95% en détaillant le protocole utilisé.

Tableau 1, nb spameur :
array([100., 80., 150., 80., 30., 74.])
((ns2-ns)**2/ns2).sum() 36.08172166427547
DDL = 5
limite = 11.1
⇒ On rejette l'hypothèse

Exercice 4 (16.5 pts) – Modèle de langue

En recherche d'information, le but est de trouver les documents qui sont pertinents pour une question q formée de la suite de mots $\{m_{q1}, \ldots, m_{qn}\}$. Pour déterminer l'intérêt d'un document d, les moteurs calculent :

$$p(d \text{ pertinent pour } q) = p(q|\mathcal{M}_d) \tag{1}$$

où $\mathcal{M}_d$ est un modèle de langue défini par le document d. En pratique, un moteur de recherche présentera en premier les documents pour lesquels la probabilité $p(q|\mathcal{M}_d)$ est la plus haute. Dans un premier temps, un modèle de langue est simplement une distribution multinomiale donnant les probabilités des mots (et donc des documents en faisant une hypothèse d'indépendance naïve) :

$$\mathcal{M}_d = \{\theta_1, \ldots, \theta_T\}, \qquad \theta_i = p(m_i|\mathcal{M}_d) = \text{ probabilité d'observation du mot } i \tag{2}$$

Un modèle de langue permet de calculer la distribution de probabilité sur des séquences de mots. De manière formelle, un modèle de langue $\mathcal{M}$ permet d'estimer la probabilité $p\,(t|\mathcal{M})$ qu'un texte t soit généré suivant le modèle de langue $\mathcal{M}$, où un texte t est représenté comme une séquence de mots $m_{t1}, \ldots, m_{tl_t}$, où l_t est la longueur du texte t. En pratique, un texte t peut être un document, un ensemble de document ou une question. Nous utiliserons les notations suivantes :

T	nombre de mots dans le vocabulaire
$C = \{d_1, \ldots, d_N\}$	Corpus : ensemble des N documents de notre univers.
f_{dm}	nombre d'occurrences du mot m dans le document d
f_m	nombre d'occurrences du mot m dans tout le corpus C ($f_m = \sum_{k=1}^{N} f_{d_i} m$)
ℓ_d	longueur du document d (nombre de mots : $\ell = \sum_{k=1}^{N} \ell_{d_k}$)
ℓ	nombre de mots dans l'ensemble des documents (le corpus C)

Le processus général est le suivant :

0. $\forall d \in C$, calcul des $\mathcal{M}_d$ au sens du maximum de vraisemblance,
1. une requête $q = \{m_{q1}, \ldots, m_{qn}\}$ arrive,
2. calculs des $p(q|\mathcal{M}_d)$, $\forall d \in C$,
3. retour des documents maximisant la vraisemblance de la requête q.

Q 4.1 Lissage de Jelinek-Mercer

Q 4.1.1 (1.5 pt) Donner la log-vraisemblance d'un document $d = \{m_{d1}, \ldots, m_{d\ell_d}\}$ en fonction des paramètres d'un modèle $\mathcal{M}$ multinomial

[Je survolarise la question pour prendre en compte le temps d'assimilation des notations]
Cela a été vu en TD - il faut qu'ils définissent les notations, mais s'ils regardent la question d'après, cela donne.
Modèle multinomial : $p(m|\mathcal{M}) = \theta_m$ avec $\sum \theta_m = 1$, puis

$$\log p(d|\mathcal{M}) = \sum_m \log \theta_m = \sum_m f_{dm} \log \theta_m$$

Q 4.1.2 (1.5 pt) L'optimisation du modèle multinomial pour un document d, au sens du maximum de vraisemblance aboutit à :

$$\theta_{dm}^{MV} = p\left(m|\mathcal{M}_d^{MV}\right) = \frac{f_{dm}}{l_d}$$

Calculer la pertinence de la question "sorties à Paris" pour les deux documents suivants – on enlèvera la ponctuation et la casse des caractères (distinction majuscule/minuscule) – :

(d1) Sorties de Paris, Ginette et Paulette tournèrent à gauche

(d2) Sorties (Paris)

Quel problème se pose lorsqu'un terme de la question n'apparaît pas dans le document, en particulier un terme peu important (comme "à" dans "sorties à Paris") ? Pourquoi est-ce un problème dans le cadre de la recherche documentaire ?

$$p(q|d1) = \frac{1}{9} \times \frac{1}{9} \times \frac{1}{9} = 9^{-3}$$

et

$$p(q|d2) = \frac{1}{2} \times \frac{0}{2} \times \frac{1}{2} = 0$$

donc le document d1 sera préféré au document d2 - et surtout d2 a une probabilité nulle d'être pertinent, ce qui n'est pas souhaitable car "à" ne devrait pas être discriminant.
Si le document ne contient pas le mot (comme à), alors $\theta_d^{MV} = 0$ et

$$p(q|\mathcal{M}_d^{MV}) = 0$$

ce qui veut dire que le document n'est pas pertinent pour la question q. Cela n'est pas souhaitable pour les mots "peu importants".

Q 4.1.3 (1.5 pt) Afin de palier à ce problème, on utilise un modèle de lissage comme celui de Jelinek-Mercer, ce qui permet de définir le modèle de langue $\mathcal{M}_d^{JM}$ pour un document d de la manière suivante :

$$p(m|\mathcal{M}_d^{JM}) = \lambda p(m|\mathcal{M}_d^{MV}) + (1-\lambda)p(m|\mathcal{M}_C^{MV})$$

où le paramètre $\lambda \in [0,1]$, et $p(m|\mathcal{M}_C^{MV})$ est le modèle de langue du corpus C, à savoir le modèle qui maximise la probabilité d'observer l'ensemble des documents si on les mettait bout à bout. On utilise $\mathcal{M}_d^{JM}$ dans l'équation (1).

(a) Que vaut $\theta_m^{MV} = p\left(m|\mathcal{M}_C^{MV}\right)$?

(b) À quoi sert λ ? En particulier, comment l'interpréter de manière probabiliste ?

(c) En quoi cela permet-il de régler le problème lié aux mots absents ?

(1)

$$\theta_m^{MV} = \frac{f_m}{l}$$

(2) λ sert à contrôler la mixture de deux probabilités : le modèle de langue du document vs modèle de langue du corpus

(3) S'il apparaît dans le corpus, alors la probabilité n'est plus zéro ce qui posait problème dans l'équation (1). Par contre, s'il n'apparaît pas dans le corpus, le problème n'est pas réglé.

Q 4.2 Jelinek-Mercer : approfondissement

Q 4.2.1 (0.5 pt) Dire pourquoi le modèle $p\left(m|\mathcal{M}_d^{JM}\right)$ est multinomial

On le voit clairement en posant

$$\theta_m = \lambda\theta_{dm}^{MV} + (1-\lambda)\theta_m^{JM}$$

Q 4.2.2 (0.5 pt) Expliquer pourquoi ce modèle ne maximise plus forcément la probabilité d'observer le document d.

Le problème est que

$$\lambda\theta_{dm}^{MV} + (1-\lambda)\,\theta_m^{MV}$$

maximise la vraisemblance seulement si $\lambda = 1$ ou si $\theta_{dm}^{MV} = \theta_m^{MV}$

Q 4.2.3 (0.5 pt) Pour éviter ce problème, on définit un nouveau modèle de langue $\mathcal{M}_d^{JM^*}$ pour un document d :

$$p(m|\mathcal{M}_d^{JM^*}) = \lambda p(m|\mathcal{M}_d^{JV}) + (1-\lambda)p(m|\mathcal{M}_C^{JV})$$

où $\theta_{dm}^{JV} = p\left(m|\mathcal{M}_d^{JV}\right)$, $\theta_m^{JV} = p\left(m|\mathcal{M}_C^{JV}\right)$ et λ sont les paramètres à apprendre.
En vous souvenant de l'interprétation probabiliste 4.1.3(b), il est possible de considérer λ comme la probabilité d'une variable Z non observée. Quelle est-elle ?

Il s'agit du choix du modèle de langue : document ($Z = 1/0$) ou corpus ($Z = 0/1$).

Q 4.2.4 (0.5 pts) La vraisemblance à maximiser correspond à la probabilité d'observer l'ensemble des documents, i.e.

$$\mathcal{L} = p(d_1, \ldots, d_n|\mathcal{M}^{JM^*})$$

En supposant les documents indépendants, on peut l'écrire comme

$$\mathcal{L} = \prod_d p(d|\mathcal{M}_d^{JM^*})$$

Écrire $\log \mathcal{L}$ en fonction de f_{dm}

$$\log \mathcal{L} = \sum_{d,m} f_{dm} \log p(m|\mathcal{M}_d^{JM^*})$$

Q 4.2.5 (3.5 pts) Proposer un algorithme pour estimer les paramètres. Vous utiliserez explicitement la variable Z définie précédemment pour l'estimation.

On utilise EM : on introduit $Q(Z;d,m)$ qui correspond à la probabilité de générer le mot m du document d avec le modèle document (on pourrait en considérer un pour chaque occurrence d'un mot m, mais dans la pratique ça ne change rien).

$$\mathcal{L} \geq \sum_{d,m,z} f_{dm} Q(z;d,m) \log \frac{p(z,m|\mathcal{M}_d^{JV^*})}{Q(z;d,m)}$$

étape E On maximise par rapport à Q :

$$Q_t(Z;d,m) = p(Z|m,d) = \frac{p(m|Z,\mathcal{M}_d^{JV^*})p(Z)}{p(m|\mathcal{M}_d^{JV^*})}$$

d'où

$$Q_t(Z;d,m) = \frac{\lambda^{(t-1)}\theta_{dm}^{JV(t-1)}}{\lambda^{(t-1)}\theta_{dm}^{JV(t-1)} + \left(1-\lambda^{(t-1)}\right)\theta_m^{JV(t-1)}}$$

étape M On maximise par rapport aux paramètres :

$$\theta_{dm}^{JV(t)} = \frac{Q_t\left(Z;d,m\right) f_{dm}}{\sum_{m'} Q_t\left(Z;d,m'\right) f_{dm'}}$$

$$\theta_m^{JV(t)} = \frac{\sum_d \left(1 - Q_t\left(Z;d,m\right)\right) f_{dm}}{\sum_{m',d} \left(1 - Q_t\left(Z;d,m'\right)\right) f_{dm'}}$$

$$\lambda^{(t)} = \frac{\sum_{d,m} Q_t\left(Z;d,m\right) f_{dm}}{l}$$

Q 4.2.6 (1 pt) Lorsqu'on cherche une solution à ce problème, λ converge vers 0 ou 1. Dire ce que valent les autres paramètres dans ce cas, et expliquer pourquoi cela pose problème. Proposez une solution (simple) pour éviter ce problème.

Lorsque λ est proche de 0 ou de 1, les paramètres sont proches de la solution du maximum de vraisemblance (on peut le voir car $Q_t \approx \lambda$ pour λ très proche de 1 ou 0).
Il suffit de ne pas apprendre le λ, et vu les principes de EM (inégalité de Jensen), on va converger vers une solution.

Q 4.3 Lissage de Dirichlet
Un autre modèle de lissage est celui basé sur le lissage de Dirichlet. Pour cela, on suppose que la probabilité a priori des termes est donné par une loi de probabilité de Dirichlet

$$p(\theta_1,\ldots,\theta_T;\boldsymbol{\alpha}) = Dir\left(\theta_1,\ldots,\theta_T;\boldsymbol{\alpha}\right) = \frac{1}{B(\boldsymbol{\alpha})}\prod_{i=1}^{T}\theta_i^{\alpha_i-1}$$

où $\boldsymbol{\alpha}$ est un vecteur de dimension T ($\boldsymbol{\alpha} \in \mathbb{R}^{+T}$) et $B(\boldsymbol{\alpha})$ permet d'obtenir une distribution de probabilité valide, i.e.

$$B(\mathbf{alpha}) = \int_{[0,1]^T} \prod_{i=1}^{T} \theta_i^{\alpha_i-1} d\theta$$

Q 4.3.1 (0.5 pt) Exprimer le principe de maximum a posteriori pour un document d (formulation).

$$\theta_d^{MAP} = \underset{\theta}{\operatorname{argmax}} \log p\left(\theta|d\right) = \underset{\theta}{\operatorname{argmax}} \log p\left(d|\theta\right) p\left(\theta\right)$$

Q 4.3.2 (2 pts) Donner le θ^{MAP} qui maximise le maximum a posteriori pour chaque mot m

Avec $\theta_{dm^*} = 1 - \sum_{m \neq m^*} \theta_{dm}$, et en annulant le gradient, on obtient

$$\theta_{dm}^{MAP} = \frac{f_{dm} + \alpha_m - 1}{l_d + \sum_{t=1}^{T} \alpha_t - 1}$$

Q 4.3.3 (3 pts) On suppose que $\alpha_m = \mu p\left(m|\mathcal{M}_C^{MV}\right)$.

(a) Ré-écrire la solution θ_{dm}^{MAP}

(b) En analysant la solution, comment peut-on interpréter μ ?

(c) On suppose que μ est de l'ordre de 1000. Quelle est la valeur de θ_{dm}^{MAP} si le document est très court ($l_d \ll \mu$) ? Si le document est long ($l_d \gg \mu$) ? En déduire l'intérêt du lissage de Dirichlet par rapport au lissage de Jelinek-Mercer.

Par substitution, on a

$$\theta_{dm}^{MAP} = \frac{f_{dm} + \mu p_{MV}\left(m|\theta_C\right) - 1}{l_d + \mu - 1}$$

Ce qui permet d'interpréter μ : cela correspond à un "nombre de mots" virtuels qu'on observe quel que soit le document ; ces mots "virtuels" sont générés selon le modèle de MV du corpus.
D'où

— Quand le document est très court, on a $f_{dm} \leq l_d \ll \mu$, d'où

$$\theta_{dm}^{MAP} = \frac{f_{dm} - 1}{l_d + \mu - 1} + \frac{\mu}{\mu + l_d - 1} p\left(m|\mathcal{M}_C^{MV}\right) \approx p\left(m|\mathcal{M}_C^{MV}\right)$$

On retrouve le modèle de corpus (maximum de vraisemblance).

— Dans un document long, on a $l_d \gg \mu$ et donc

$$\theta_{dm}^{MAP} \approx \frac{f_{dm}}{l_d} = p\left(m|\mathcal{M}_d^{MV}\right)$$

on retrouve donc le modèle de maximum de vraisemblance
Donc, lorsqu'un document contient peu de mots, on est proche du modèle de corpus ; lorsqu'un document contient beaucoup de mots, on est proche du modèle de maximum de vraisemblance $\mathcal{M}_d^{MV}$. C'est un comportement souhaitable car lorsqu'il y a peu de mots, les estimations sont plus bruitées ; et lorsqu'il y en a beaucoup, le modèle de corpus n'est plus vraiment nécessaire pour lisser les probabilités.