Métriques statistiques pour l'évaluation de performance en présence de vérité terrain imprécise[1]

Bart Lamiroy* — **Pascal Pierrot****

* *Université de Lorraine* – LORIA *(UMR 7503)*
Campus Scientifique – BP 239
54506 Vandœuvre-lès-Nancy CEDEX – FRANCE
bart.lamiroy@loria.fr
** *Université de Lorraine – Mines Nancy*
Campus Artem - CS 14 234
92 Rue Sergent Blandan
54042 Nancy – FRANCE

RÉSUMÉ. *Ce papier aborde l'évaluation de performances en présence de vérité terrain imprécise. En effet, lors de procédures de benchmarking il est généralement supposé que les données de référence sont parfaites. Nous avons démontré précédemment que cette hypothèse de travail n'est généralement pas satisfaite dans le contexte de problèmes d'interprétation perceptuelle, sauf dans les cas les plus triviaux. Nous présentons ici un approche et test statistiques qui permettent de mesurer la confiance que l'on peut avoir dans des classements issues de campagnes d'évaluation. Plus précisément, nous sommes capables d'exprimer la probabilité qu'un classement reste inchangé en fonction du taux d'erreur supposé dans les données d'évaluation.*

ABSTRACT. *This paper addresses performance evaluation in the presence of imprecise ground-truth. Indeed, the most common assumption when performing benchmarking measures is that the reference data is flawless. In previous work, we have shown that this assumption cannot be taken for granted, and that, in the case of perceptual interpretation problems it is most certainly always wrong but for the most trivial cases. We are presenting a statistical test that will allow to measure the confidence one can have in the results of a benchmarking test ranking multiple*

1. Traduction étendue d'un article présenté à GREC 2015 – *Eleventh IAPR International Workshop on Graphics Recognition – 22-23 August 2015, Nancy, France* et à paraître en anglais dans le volume Springer LNCS associé.

algorithms. More specifically, we can express the probability of the ranking not being respected in the presence of a given level of errors in the ground truth data.

MOTS-CLÉS : *Évaluation de performances, vérité terrain imprécise*

KEYWORDS: *Performance analysis, imprecise ground truth*

1. Introduction

Dans ce papier nous étudions des tests statistiques permettant d'évaluer le risque d'établir des classements d'algorithmes erronés sur des *benchmarks* lorsque la vérité terrain utilisée est entachée d'erreurs. Le consensus courant en analyse de performance est que les algorithmes sont testés sur une vérité terrain qui est totalement fiable. Nous avons montré précédemment que cette hypothèse est erronée et qu'il y a forcément un biais interprétatif dans les données expérimentales et la vérité terrain (Lamiroy, 2013b ; Lamiroy, 2013a).

À notre connaissance, il s'agit de la première tentative d'établir un formalisme mathématique permettant d'évaluer le risque d'utiliser de la vérité terrain imprécise. En effet, la probabilité de mal classer un algorithme dépend directement de sa performance sur les données d'évaluation d'une part, et sur le taux d'erreur dans la vérité terrain utilisée d'autre part. Ce formalisme peut être appliquée à des *benchmarks* et des campagnes d'évaluation comme (Liu, 2006 ; Al-Khaffaf *et al.*, 2010 ; Al-Khaffaf *et al.*, 2013 ; Bukhari *et al.*, 2014) ou encore (Lamiroy et Sun, 2013).

2. Description du problème

2.1. *Définitions et notations*

Dans cette section nous introduisons l'ensemble des définitions et notations que nous utiliserons par la suite.

Soient $\Delta = \{\delta_1, ..., \delta_p\}$ un ensemble de données, $\mathcal{I} = \{i_1, ..., i_q\}$ un ensemble fini d'interprétations possibles sur Δ et $\mathsf{A} = \{A_1, ..., A_n\}$ un ensemble d'algorithmes.

Dans un premier temps, nous définissons la notion de *Vérité Terrain*, qui associe une valeur de vérité à l'interprétation i d'une donnée d.

Définition 1 (Vérité Terrain) *Une vérité terrain est une fonction Ω telle que :*

$$\begin{array}{rcl} \Omega : \Delta \times \mathcal{I} & \rightarrow & \{0,1\} \\ (\delta, i) & \mapsto & \left\{ \begin{array}{l} 1 \text{ ssi } i \textit{ est une interprétation correcte pour } \delta \\ 0 \textit{ sinon} \end{array} \right. \end{array}$$

Dans le reste de ce document et pour tout donnée δ et interprétation i, nous noterons systématiquement :

$$\begin{array}{rcl} \delta \models_\Omega i & \text{si} & \Omega(\delta, i) = 1 \\ \delta \not\models_\Omega i & \text{si} & \Omega(\delta, i) = 0 \end{array}$$

Définition 2 (Algorithme) *Un algorithme A est une fonction qui associe une ou plusieurs interprétations à une donnée δ.*

$$\begin{aligned} A : \Delta &\rightarrow \{0,1\}^q \\ \delta &\mapsto (a_1, ..., a_q) \\ &\quad \textit{avec } a_k\textit{=1 si } \delta \textit{ a } i_k \in \mathcal{I} \textit{ comme interpretation} \\ &\quad \textit{and } a_k\textit{=0 sinon} \end{aligned}$$

Comme illustré dans le Tableau 1, nous adopterons la notation abrégée suivante : $A_j(\delta_k) = (a^j_{k1}, .., a^j_{kq})$, pour $k \in \{1, ..p\}$.

	A_1				...	A_n			
	i_1	i_2	...	i_q	...	i_1	i_2	...	i_q
δ_1	a^1_{11}	a^1_{12}		a^1_{1q}		a^n_{11}	a^n_{12}		a^n_{1q}
δ_2		a^1_{23}	...				a^n_{23}	...	
...									
δ_p	a^1_{p1}								a^n_{pq}

Tableau 1. *Exemple de représentation de données, algorithmes et interprétations*

2.2. *Classer des algorithmes*

L'analyse de performances consiste généralement à établir des classements entre algorithmes en fonction de leurs résultats sur la vérité terrain. Afin de correctement développer la suite de notre raisonnement nous devons formaliser la notion de classement et d'ordre de classement d'algorithmes sur une vérité terrain donnée.

Définition 3 (Ordre de classement) *Un ordre de classement dépend d'une vérité terrain Ω et est défini pour un ensemble d'algorithmes* **A***, un ensemble de données Δ et un ensemble d'interprétations $\mathcal{I}$.*

On note $\prec_\Omega$ un ordre sur **A** *tel que* $A_1 \prec_\Omega A_2$ ssi

$$\left|\left\{(k,l)|a^1_{k,l} = \Omega\left(\delta_k, i_l\right)\right\}\right| \leq \left|\left\{(k,l)|a^2_{k,l} = \Omega\left(\delta_k, i_l\right)\right\}\right|$$

en d'autre termes, les algorithmes sont comparés selon la cardinalité de leur accord sur la vérité terrain.

3. Métriques de performance dans le cas de vérité terrain parfaite

La comparaison d'algorithmes en présence de vérité terrain parfaitement fiable correspond aux méthodes habituellement utilisées et ne nécessite pas d'approche sta-

	A_1				A_2				A_3				Ω			
	i_1	i_2	i_3	i_4	i_1	i_2	i_3	i_4	i_1	i_2	i_3	i_4	i_1	i_2	i_3	i_4
δ_1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1
δ_2	0	0	1	0	0	0	0	1	1	0	0	1	0	0	0	1
δ_3	0	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0
δ_4	0	1	1	0	0	1	0	0	1	1	0	0	0	1	0	0
δ_5	1	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0

Tableau 2. *Exemple de résultats algorithmes sur une vérité terrain parfaite* Ω

tistique spécifique. Le Tableau 2 montre un exemple de résultats d'algorithmes sur un ensemble de données, sous l'hypothèse que Ω représente une vérité terrain parfaitement fiable.

En utilisant la notation de la définition 3, la façon la plus triviale pour classer les algorithmes est de calculer le taux de bonnes réponses de chacun d'entre eux et de les trier selon ce critère.

Le taux de bonnes réponses d'un algorithme A_j est :

$$\tau_{A_j} = \frac{\sum_{k=1,l=1}^{k=p,l=q} 1_{a^j_{k,l}=\Omega(\delta_k,i_l)}}{pq}$$

τ_{A_j} permet alors d'instancier l'opérateur d'ordre de classement d'algorithmes en définissant $A_i \prec_\Omega A_j$ *ssi* $\tau_{A_i} <= \tau_{A_j}$.

Si on applique ceci aux données dans le Tableau 2, on obtient $\tau_{A_1} = 0.85$, $\tau_{A_2} = 0.95$, $\tau_{A_3} = 0.8$ et, par conséquent, que $A_3 \prec_\Omega A_1 \prec_\Omega A_2$.

4. Métriques de performance dans le cas de vérité terrain erronée

Nous avons démontré dans (Lamiroy, 2013a) qu'il est quasi-impossible d'obtenir une vérité terrain parfaite. Il est donc nécessaire que Ω contienne forcément un pourcentage de données incertaines. Nous allons prendre comme hypothèse de travail que la valeur de ce pourcentage est bornée par une valeur connue ε.

4.1. *Cas général*

Nous nous intéressons ici à la question de savoir quelle confiance avoir dans le classement d'un ensemble d'algorithmes lorsqu'il est basé sur une vérité terrain fiable à ε près. Pour ce faire, appelons $\overline{\Omega}$ la vérité terrain *absolue* à laquelle nous n'avons accès qu'à travers une approximation Ω_ε.

En d'autres termes,

$$\frac{\left|\left\{(\delta,i)\in\Delta\times\mathcal{I}\,\middle|\,\Omega_\varepsilon(\delta,i)\neq\overline{\Omega}(\delta,i)\right\}\right|}{|\Delta|\,|\mathcal{I}|}\leq\varepsilon$$

A partir de Ω_ε nous pouvons utiliser Défintion 3 pour définir $\prec_{\Omega_\varepsilon}$. La question restante à laquelle nous cherchons maintenant à répondre est de savoir s'il est possible de déterminer ci cette ordre de classement est une approximation fiable de $\prec_{\overline{\Omega}}$ (à laquelle nous n'avons pas accès).

La suite de ce papier sera consacré à quelques approches probabilistes qui nous permettront de quantifier cette différence.

4.2. *Notations*

Nous définissons les variables aléatoires suivantes :

Pour une vérité terrain Ω, un algorithme A_j et avec $j\in[1,...,n]$, $k\in[1,...,p]$ et $l\in[1,...,q]$, la variable aléatoire $X_{k,l}^{\Omega,j}$ exprime l'évènement que l'algorithme A_j interprète δ_k correctement comme i_l.

Ceci implique que

$$\begin{array}{l}X_{k,l}^{\Omega,j}=1 \text{ si } \ A_j(\delta_k)|_l=\ \Omega(\delta_k,i_l)|\\ X_{k,l}^{\Omega,j}=0 \text{ sinon}\end{array}$$

ce qui peut également être exprimé de la façon suivante :

$$\begin{array}{l}X_{k,l}^{\Omega,j}=1 \text{ si } a_{kl}^j=\Omega(\delta_k,i_l)\\ X_{k,l}^{\Omega,j}=0 \text{ sinon}\end{array}$$

Nous noterons les réalisations de $X_{k,l}^{\Omega,j}$ comme $x_{k,l}^{\Omega,j}$, et les probabilités associées

$$\mathbf{P}\left(X_{k,l}^{\Omega,j}=1\right)=p_{k,l}^{\Omega,j}$$

et

$$\mathbf{P}\left(X_{k,l}^{\Omega,j}=0\right)=1-p_{k,l}^{\Omega,j}$$

avec $\forall i,j,k\ p_{k,l}^j\in[0,1]$.

5. Solution simplifiée avec deux algorithmes

Compte-tenu du fait que nous n'avons aucune information *a priori* de $\overline{\Omega}$, nous ne pouvons pas avoir recours à un test d'adéquation multinomiale χ^2. En revanche, il est

possible, dans des cas simplifiés, de faire un développement formel des statistiques. Dans cette section nous allons considérer le cas de deux algorithmes et un seule interprétation possible par algorithme. Ceci nous donnera une première catégorie de tests statistiques.

	A_1	A_2	Ω_ε	$\overline{\Omega}$
	i	i	i	i
δ_1	0	1	0	$x_1^{\overline{\Omega}}$
δ_2	0	1	0	$x_2^{\overline{\Omega}}$
δ_3	0	1	0	$x_3^{\overline{\Omega}}$
δ_4	0	1	0	$x_4^{\overline{\Omega}}$

5.1. *Hypothèses de travail et notations*

Nous supposons que nous avons deux algorithmes A_1 et A_2 à comparer, et que seule une interprétation est possible $\mathcal{I} = \{i\}$.

Par conséquent, nous noterons $\{a_k^j\}$ l'ensemble des valeurs d'un algorithme sur Δ et nous noterons $\Omega_\varepsilon(\delta_k)$ l'ensemble correspondant de vérité terrain. Comme précédemment $\overline{\Omega}$ représente la vérité terrain parfaite, mais inconnue et pour laquelle nous ne disposons pas d'ordre de classement. De même, Ω_ε est la vérité terrain dont les valeurs sont connues, pas pour laquelle il est connu qu'elle diffère de $\overline{\Omega}$ d'au plus ε. Elle nous permet de définir et calculer l'ordre de classement $\prec_{\Omega_\varepsilon}$ entre A_1 et A_2.

Afin de pouvoir développer la suite de notre raisonnement, nous devons introduire la notion de *divergence* entre deux algorithmes A_1 et A_2 :

Définition 4 (Ensemble de désaccord) *Soient A_1 et A_2 deux algorithmes pour lesquels les résultats par rapport à une vérité terrain Ω sont connus. Soient résultats sont respectivement $a_{k,l}^{\Omega,1}$ et $a_{k,l}^{\Omega,2}$ avec $k \in [1..p]\,, l \in [1..q]$.*

Nous définissons l'ensemble de désaccord entre A_1 et A_2 comme

$$\mathcal{D}(A_1, A_2) = \left\{ (k,l) \,\middle|\, a_{k,l}^{\Omega,1} \neq a_{k,l}^{\Omega,2} \right\}.$$

Cette notation peut être étendue pour exprimer le désaccord entre un algorithme et une vérité terrain, ou entre deux vérités terrain :

$$\begin{aligned} \mathcal{D}\left(A_i, \Omega\right) &= \left\{ (k,l) \,\middle|\, a_{k,l}^{\Omega,i} \neq x_{k,l}^{\Omega} \right\} \\ \mathcal{D}\left(\overline{\Omega}, \Omega_\varepsilon\right) &= \left\{ (k,l) \,\middle|\, x_{k,l}^{\overline{\Omega}} \neq x_{k,l}^{\Omega_\varepsilon} \right\} \end{aligned}$$

Définition 5 (Ensemble d'accord) *Soient A_1 et A_2 deux algorithmes pour lesquels les résultats par rapport à une vérité terrain Ω sont connus. Soient résultats sont respectivement $a_{k,l}^{\Omega,1}$ et $a_{k,l}^{\Omega,2}$ avec $k \in [1..p]\,, l \in [1..q]$.*

Nous définissons l'ensemble d'accord entre A_1 et A_2 comme

$$\mathcal{A}(A_1, A_2) = \left\{ (k,l) \,\middle|\, a_{k,l}^{\Omega,1} = a_{k,l}^{\Omega,2} \right\}.$$

Il est immédiat d'observer que $\mathcal{A}$ and $\mathcal{D}$ sont des compléments l'un de l'autre :

$$\mathcal{A}(X,Y) = \overline{\mathcal{D}(X,Y)}.$$

Définition 6 (Divergence entre deux algorithmes) *Soient A_1 et A_2 deux algorithmes. Étant donnés leurs ensembles de désaccord par la Définition 4, nous définissons la divergence entre A_1 et A_2 comme*

$$\mathbf{D}(A_1, A_2) = |\mathcal{D}(A_1, A_2)|.$$

Comme dans le cas des ensembles de désaccord, cette définition peut être étendue pour exprimer la divergence entre algorithmes et vérité terrain, ou entre vérités terrain :

$$\begin{aligned} \mathbf{D}(A_i, \Omega) &= |\mathcal{D}(A_i, \Omega)| \\ \mathbf{D}\left(\overline{\Omega}, \Omega_\varepsilon\right) &= \left|\mathcal{D}(\overline{\Omega}, \Omega)\right| \end{aligned}$$

5.2. *Estimation de la divergence*

Il est trivial de démontrer que les deux équations suivantes bornent $\mathbf{D}(A_1, A_2)$ quelle que soit la vérité terrain Ω :

$$\mathbf{D}(A_1, A_2) \leq \mathbf{D}(A_1, \Omega) + \mathbf{D}(A_2, \Omega) \quad [1]$$

$$\mathbf{D}(A_2, \Omega) - \mathbf{D}(A_1, \Omega) \leq \mathbf{D}(A_1, A_2) \quad [2]$$

Comme, par ailleurs, la divergence entre $\overline{\Omega}$ et Ω_ε est bornée par $\varepsilon\, p$, on peut affirmer que

$$\mathbf{D}(A_i, \overline{\Omega}) - \varepsilon\, p \leq \mathbf{D}(A_i, \Omega_\varepsilon) \leq \mathbf{D}(A_i, \overline{\Omega}) + \varepsilon\, p \quad [3]$$

On en déduit, en combinant [2] et [3], que

$$\mathbf{D}(A_2, \overline{\Omega}) \leq \mathbf{D}(A_1, A_2) + \varepsilon\, p + \mathbf{D}(A_1, \Omega_\varepsilon) \quad [4]$$

$$\mathbf{D}(A_2, \overline{\Omega}) - \mathbf{D}(A_1, \overline{\Omega}) \leq \mathbf{D}(A_1, A_2) + 2\varepsilon\, p \quad [5]$$

5.3. *Estimation de la probabilité d'un changement d'ordre de classement*

À ce point, nous disposons d'une vérité terrain approximative Ω_ε et deux algorithmes A_1 et A_2, que l'on est capable de comparer avec $\prec_{\Omega_\varepsilon}$. La question qui se pose

maintenant est de savoir si l'ordre entre A_1 et A_2 aurait été différent s'il avait été basé sur $\overline{\Omega}$ (*i.e.* sans le taux d'erreur ε) ?

Nous allons poser la question de l'influence d'un changement dans la valeur de ε sur le classement des algorithmes en des termes probabilistes. Soit $\mathbf{P}\left(A_1 \prec_{\overline{\Omega}} A_2 \left|_{A_1 \prec_{\Omega_\varepsilon} A_2}\right.\right)$ la probabilité que l'ordre de classement reste inchangé pour la vérité terrain $\overline{\Omega}$ étant donné l'ordre pour Ω_ε.

Un cas trivial est celui où $|\mathbf{D}(A_1, \Omega_\varepsilon) - \mathbf{D}(A_2, \Omega_\varepsilon)| > 2\varepsilon p$ (p étant le nombre d'éléments dans Δ). Dans ce cas, l'ordre $\prec_{\Omega_\varepsilon}$ sur A_1 et A_2 sera strictement équivalent que pour $\prec_{\overline{\Omega}}$.

En effet, supposons (sans perte de généralité, à cause de la symétrie du problème) que $A_1 \prec_{\Omega_\varepsilon} A_2$. Dans le pire des cas, nous aurons :

$$\mathcal{D}\left(\overline{\Omega}, \Omega_\varepsilon\right) \subset \mathcal{D}\left(A_1, \overline{\Omega}\right) \cap \mathcal{D}\left(A_2, \overline{\Omega}\right)$$

où, par construction du pire scénario, $\mathbf{D}\left(\overline{\Omega}, \Omega_\varepsilon\right) = p\varepsilon$. On obtient donc

$$\mathbf{D}(A_1, \overline{\Omega}) = \mathbf{D}(A_1, \Omega_\varepsilon) + p\varepsilon$$

et

$$\mathbf{D}(A_2, \overline{\Omega}) = \mathbf{D}(A_2, \Omega_\varepsilon) - p\varepsilon$$

En remplaçant les termes appropriés, on en déduit que $\mathbf{D}(A_1, \overline{\Omega}) - \mathbf{D}(A_2, \overline{\Omega}) > 0$, et que, par conséquent, $A_1 \prec_{\overline{\Omega}} A_2$.

En conclusion

$$\mathbf{P}\left(A_1 \prec_{\overline{\Omega}} A_2 \left|_{A_1 \prec_{\Omega_\varepsilon} A_2}\right.\right) = 1$$

lorsque $\mathbf{D}(A_2, \Omega_\varepsilon) - \mathbf{D}(A_1, \Omega_\varepsilon) \geq 2\varepsilon p$.

De façon un peu plus générale, si l'on prend l'ensemble d'accord de A_2 (*cf.* Définition 4) $\mathcal{A}\left(A_2, \Omega_\varepsilon\right)$ contenant les interprétations communes A_2 et Ω_ε, et l'ensemble de désacord $\mathcal{D}\left(A_1, \Omega_\varepsilon\right)$ l'ensemble des interprétations ou A_1 et Ω_ε diffèrent, alors

$$\mathbf{P}\left(A_1 \prec_{\overline{\Omega}} A_2 \left|_{A_1 \prec_{\Omega_\varepsilon} A_2}\right.\right) = 1 \qquad \text{if}$$

$$\left|\mathcal{A}\left(A_2, \Omega_\varepsilon\right) \cap \mathcal{D}\left(A_1, \Omega_\varepsilon\right)\right| < \frac{\mathbf{D}(A_2, \Omega_\varepsilon) - \mathbf{D}(A_1, \Omega_\varepsilon)}{2}$$

Explication

La seule configuration où A_1 pourrait inverser son classement avec A_2 serait lorsqu'il y a un nombre suffisant de valeurs de $\mathcal{A}\left(A_2, \Omega_\varepsilon\right) \cap \mathcal{D}\left(A_1, \Omega_\varepsilon\right)$ pour lesquelles Ω diffère de Ω_ε. Il en faut au moins $\frac{\mathbf{D}(A_2, \Omega_\varepsilon) - \mathbf{D}(A_1, \Omega_\varepsilon)}{2}$.

Pour les cas où la probabilité ne vaut pas 1, il est possible de faire une similation de Monte-Carlo (Metropolis et Ulam, 1949) dans lasuelle les valeurs de $\Omega_\varepsilon(\delta_k)$ sont changées avec une probabilité de ε. Ceci permet alors d'estimer $\mathbf{P}\left(A_1 \prec_{\overline{\Omega}} A_2 \left|_{A_1 \prec_{\Omega_\varepsilon} A_2}\right.\right)$.

	A_1	A_2	Ω_ε	$\overline{\Omega}$
	i	i	i	i
δ_1	0	0	0	$x_1^{\overline{\Omega}}$
δ_2	1	1	1	$x_2^{\overline{\Omega}}$
δ_3	1	1	1	$x_3^{\overline{\Omega}}$
δ_4	1	0	0	$x_4^{\overline{\Omega}}$
δ_5	1	0	0	$x_5^{\overline{\Omega}}$
δ_6	0	1	1	$x_6^{\overline{\Omega}}$
δ_7	0	1	0	$x_7^{\overline{\Omega}}$
δ_8	0	0	1	$x_8^{\overline{\Omega}}$
δ_9	0	0	1	$x_9^{\overline{\Omega}}$
δ_{10}	1	1	0	$x_{10}^{\overline{\Omega}}$

Tableau 3. *Example trivial de deux algorithmes vis-à-vis une vérité terrain de type binaire.*

Algorithme

Itérer N foix (pour de grandes valeurs de N)

– pour tout $k \in [1..p]$, changer la valeur de $\Omega_\varepsilon(\delta_k)$ avec une probabilité de ε ;

– calculer $\mathbf{D}(A_2, \Omega_\varepsilon) - \mathbf{D}(A_1, \Omega_\varepsilon)$;

– si $\mathbf{D}(A_2, \Omega_\varepsilon) - \mathbf{D}(A_1, \Omega_\varepsilon) \geq 0$, incrémenter un compteur c de 1.

Après N itérations, $\frac{c}{N}$ contient une approximation de la probabilité recherchée.

Exemple

nous avons utilisé une implémentation de Monte-Carlo en Matlab sur les données du Tableau 3. Dans cet exemple, $A_1 \prec_{\Omega_\varepsilon} A_2$ et $p = 10$.

Avec $N = 10^5$ tests et un niveau de confiance de 95% nous obtenons :

– $\varepsilon = 0.5$: $\mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2) = 0.3135 \pm 0.002876$

– $\varepsilon = 0.2$: $\mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2) = 0.5893 \pm 0.003049$

– $\varepsilon = 0.1$: $\mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2) = 0.7517 \pm 0.002677$

– $\varepsilon = 0$: comme attendu $\mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2) = 1$

5.4. *Solution algébrique*

Si la méthode Monte-Carlo précédente se généralise aisément à plusieurs algorithmes, il n'est pas évident de trouver une démonstration formelle et une solution algébrique. Nous la posons néanmoins pour le cas de deux algorithmes.

Il est en effet possible d'observer les points suivants :

1) Comme Ω_ε ne diffère de $\overline{\Omega}$ que par ε, la probabilité que $\Omega_\varepsilon(\delta_i)$ diffère de $\overline{\Omega}(\delta_i)$, exprimée comme $\mathbf{P}\left(x_i^{\overline{\Omega}} \neq x_i^{\Omega_\varepsilon}\right)$, peut être considérée comme suivant une loi de Bernoulli de paramètre ε. (Ceci est exactement ce qui se passe dans le tirage de Monte-Carlo dans la section précédente)

2) Notre objectif est de mesurer l'impact d'un désaccord entre Ω_ε et $\overline{\Omega}$ sur l'ordre de classement entre A_1 et A_2. Pour chaque δ_i où A_1 et A_2 sont en accord entre eux (indépendamment du fait qu'ils soient en accord ou non avec Ω_ε) un changement de $\Omega_\varepsilon(\delta_i)$ n'affectera aucunement l'ordre de classement $A_1 \prec_{\Omega_\varepsilon} A_2$ puisque les deux algorithmes seront affectés de la même façon.

Par conséquent, $\mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2)$ dépend uniquement de la probabilité que Ω_ε soit en désaccord avec $\overline{\Omega}$ sur les seuls δ_i où A_1 est en désaccord avec A_2.

Avec ces observations et utilisant les mêmes notations que précédemment, on peut calculer la probabilité $\mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2)$ de façon algébrique comme décrit ci-dessous.

Soit $\mathcal{D}_{A_1}$ (resp. $\mathcal{D}_{A_2}$) le sous-ensemble $\mathcal{D}(A_1, A_2)$ où A_1 (resp. A_2) est en accord avec Ω_ε tout en étant en désaccord avec A_2 (resp. A_1).

$$\mathcal{D}_{A_1} = \mathcal{D}(A_1, A_2) \cap \mathcal{A}(A_1, \Omega_\varepsilon)$$
$$\mathcal{D}_{A_2} = \mathcal{D}(A_1, A_2) \cap \mathcal{A}(A_2, \Omega_\varepsilon)$$

Puisque nous ne considérons que deux algorithmes et étant donné l'observation 2 ci-dessus, ceci est équivalent à

$$\mathcal{D}_{A_1} = \mathcal{D}(A_1, A_2) - \mathcal{D}(A_2, \Omega_\varepsilon)$$
$$\mathcal{D}_{A_2} = \mathcal{D}(A_1, A_2) - \mathcal{D}(A_1, \Omega_\varepsilon)$$

Il est trivial de montrer que $\mathcal{D}_{A_1} \cap \mathcal{D}_{A_2} = \emptyset$ et que $A_1 \prec_{\Omega_\varepsilon} A_2$ *ssi* $\mathbf{D}_{A_1} \leq \mathbf{D}_{A_2}$ (où $\mathbf{D}$ exprime la cardinalité de $\mathcal{D}$).

De plus, $\mathcal{D}_{A_1} \cap \mathcal{D}_{A_2} = \emptyset$ implique que $\mathbf{D}_{A_2} = \mathbf{D}(A_1, A_2) - \mathbf{D}_{A_1}$ et par conséquent, que

$$A_1 \prec_{\Omega_\varepsilon} A_2 \quad \textit{ssi} \quad \mathbf{D}_{A_1} \leq \frac{\mathbf{D}(A_1, A_2)}{2}. \qquad [6]$$

Comme $\mathbf{D}(A_1, A_2)$ est indépendant de Ω, on peut en conclure que $\mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2)$ correspond à la probabilité que

$$D = \mathbf{D}_{A_1} - \frac{\mathbf{D}(A_1, A_2)}{2}$$

ne change pas de signe.

Sans perte de généralité et à cause de la symétrie du problème, on peut choisir A_1 et A_2 tels que $A_1 \prec_{\Omega_\varepsilon} A_2$, et donc $\mathbf{D}_{A_1} \leq \mathbf{D}_{A_2}$ et $D < 0$.

D changera de signe si au moins $\hat{D} = \frac{\mathbf{D}(A_1,A_2)}{2} - \mathbf{D}_{A_1}$ évènements de $\mathcal{D}_{A_2}$ sont en désaccord avec $\overline{\Omega}$ (et si aucun de ceux de $\mathcal{D}_{A_1}$ le sont). Étant donné que nos évènements suivent une loi de Bernouli law de paramètre ε, la probabilité d'avoir au moins $\hat{D}$ évènements (et donc D de changer de signe) est

$$\sum_{i=\hat{D}}^{\mathbf{D}_{A_2}} \left(\begin{array}{c} \mathbf{D}_{A_2} \\ i \end{array} \right) \varepsilon^i \left(1-\varepsilon\right)^{\mathbf{D}_{A_2}-i} = \sum_{i=\hat{D}}^{\mathbf{D}_{A_2}} \mathcal{B}\left(\mathbf{D}_{A_2}, i\right). \quad [7]$$

Cette probabilité est conditionnée par le fait que tous les évènements de $\mathcal{D}_{A_1}$ doivent être en accord avec $\overline{\Omega}$. Si k évènements de $\mathcal{D}_{A_1}$ étaient en désaccord avec $\overline{\Omega}$, la probabilité ci-dessus deviendrait

$$\sum_{i=\hat{D}+k}^{\mathbf{D}_{A_2}} \mathcal{B}\left(\mathbf{D}_{A_2}, i\right). \quad [8]$$

Par conséquent, la probabilité globale, couvrant tous les cas, que D change de signe est

$$\mathbf{P}_{\text{switch}} = \sum_{k=0}^{\mathbf{D}_{A_1}} \mathcal{B}\left(\mathbf{D}_{A_1}, k\right) \sum_{i=\hat{D}+k}^{\mathbf{D}_{A_2}} \mathcal{B}\left(\mathbf{D}_{A_2}, i\right). \quad [9]$$

Finalement, puisque nous cherchons la probabilité que l'ordre de classement reste inchangé, nous obtenons que

$$\mathbf{P}\left(A_1 \prec_{\overline{\Omega}} A_2 \,\middle|_{A_1 \prec_{\Omega_\varepsilon} A_2}\right) = 1 - \mathbf{P}_{\text{switch}} \quad [10]$$

Exemple numérique

En utilisant le Tableau 3, nous observons que $A_1 \prec_{\Omega_\varepsilon} A_2$. De plus,

$$\mathcal{D}\left(A_1, A_2\right) = \{\delta_4, \delta_5, \delta_6, \delta_7\}$$

$$\mathcal{A}\left(A_1, \Omega_\varepsilon\right) = \{\delta_1, \delta_2, \delta_3, \delta_7\} \qquad \mathcal{A}\left(A_2, \Omega_\varepsilon\right) = \{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6\}$$

$$\mathcal{D}_{A_1} = \{\delta_7\} \qquad \mathcal{D}_{A_2} = \{\delta_4, \delta_5, \delta_6\}$$

Les autres paramètres observables sont : $\mathbf{D}_{A_1} = 1$, $\mathbf{D}_{A_2} = 3$ and $\hat{D} = 1$. L'équation 10 peut donc être ré-écrite comme

$$\begin{aligned} \mathbf{P}\left(A_1 \prec_{\overline{\Omega}} A_2\right) &= 1 - \sum_{k=0}^{1} \left(\begin{array}{c} 1 \\ k \end{array} \right) \varepsilon^k \left(1-\varepsilon\right)^{1-k} \sum_{i=1+k}^{3} \left(\begin{array}{c} 3 \\ i \end{array} \right) \varepsilon^i \left(1-\varepsilon\right)^{3-i} \quad [11] \\ &= 1 - \left(\left(1-\varepsilon\right) \left(3\varepsilon \left(1-\varepsilon\right)^2 + 3\varepsilon^2 \left(1-\varepsilon\right) + \varepsilon^3 \right) + \varepsilon \left(3\varepsilon^2 \left(1-\varepsilon\right) + \varepsilon^3 \right) \right) \\ &= 1 - \varepsilon \left(3 - 6\varepsilon + 7\varepsilon^2 - 3\varepsilon^4\right) \end{aligned}$$

On obtient donc :

– $\varepsilon = 0.5 : \mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2) = 0.313$
– $\varepsilon = 0.2 : \mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2) = 0.589$
– $\varepsilon = 0.1 : \mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2) = 0.753$
– $\varepsilon = 0$: comme attendu $\mathbf{P}(A_1 \prec_{\overline{\Omega}} A_2) = 1$

Ceci corrobore les estimations obtenues précédemment avec Monte-Carlo.

6. Extension à des interprétations multiples et niveaux de confiance

Jusqu'ici nous n'avons considéré que deux algorithmes exprimant les interprétations sous forme booléenne. Dans cette section nous considérons toujours deux algorithmes, mais cette fois-ci, exprimant des valeurs de confiance sur plusieurs possibles valeurs d'interprétation. Pour ce faire, nous allons utiliser la divergence de Kullback-Leibler divergence (Kullback et Leibler, 1951). La divergence de Kullback-Leibler est une mesure de dissimilarité entre deux distributions de probabilité P et Q, où P représente une série d'observations, ou une distribution de probabilité connue, et Q un modèle ou une approximation de P.

Définition 7 (Kullback-Leibler Divergence) *Soient P et Q deux distributions de probabilité. La divergence Kullback-Leibler de Q par rapport à P est définie par*

$$D_{KL}(P||Q) = \sum_i P(i)\, ln\left(\frac{P(i)}{Q(i)}\right)$$

Note : $D_{KL}(P||Q) = D_{KL}(Q||P) = 0$ *ssi* $P = Q$.

6.1. *Application à l'évaluation d'ordres de classement*

Afin de pouvoir appliquer Kullback-Leibler à notre cas, il est nécessaire de disposer de distributions de probabilité, et donc de reformuler notre problème et de le restreindre à des cas particuliers.

Hypothèses de travail

1) Nous considérons toujours deux algorithmes.

2) De multiples interprétations sont possibles (*i.e.* $q \geq 1$).

3) La vérité terrain n'attribue qu'une seule interprétation à chaque donnée mais peut exprimer une confiance plus ou moins forte dans cette "vérité".

4) Les algorithmes retournent une mesure de confiance entre $[0..1]$ par possible interprétation pour chaque donnée.

Nous étendons donc la définition d'algorithme comme suit :

Définition 8 (Algorithme) *Un algorithme A est une fonction qui associe une valeur de confiance à une ou plusieurs interprétations à une donnée δ.*

$$\begin{aligned} A : \Delta &\rightarrow [0..1]^q \\ \delta &\mapsto (a_1, ..., a_q) \\ & \quad avec \sum_{l=1}^{q} a_l = 1 \end{aligned}$$

Telle que formulée, la confiance d'interprétation d'une donnée particulière peut être assimilée à une distribution de probabilité.

Pour une vérité terrain donnée Ω et une donnée δ_k nous obtenons donc la divergence Kullback-Leibler :

$$D_{KL}\left(A_i\left(\delta_k\right)||\Omega\right) = \sum_{l=1}^{q} a_{kl}^{j} \, ln\left(\frac{a_{kl}^{j}}{\Omega(\delta_k, i_l)}\right)$$

Application numérique

Soient A_1 et A_2 deux algorithmes à comparer et Ω_ε la vérité terrain approximative, mais connue. D'abord, nous établissons un ordre de classement entre A_1 et A_2 en mesurant leur divergence au sens de Kullback-Leibler avec Ω_ε.

$$\mathbf{D}\left(A_i, \Omega_\varepsilon\right) = \sum_{k=1}^{p} D_{KL}\left(A_i\left(\delta_k\right)||\Omega_\varepsilon\right)$$

Il est ensuite possible d'appliquer les mêmes définitions et techniques que précédemment. $A_1 \prec_{\Omega_\varepsilon} A_2$ *ssi* $\mathbf{D}\left(A_1, \Omega_\varepsilon\right) >= \mathbf{D}\left(A_2, \Omega_\varepsilon\right)$ ou, en d'autres termes, *ssi*

$$\sum_{k=1}^{p} D_{KL}(A_1(\delta_k)||\Omega_\varepsilon) \geq \sum_{k=1}^{p} D_{KL}(A_2(\delta_k)||\Omega_\varepsilon)$$

$\mathbf{P}\left(A_1 \prec_\Omega A_2 \left|_{A_1 \prec_{\Omega_\varepsilon} A_2}\right.\right)$ peut ensuite être calculé en utilisant une technique de simulation de Monte-Carlo comme vu précédemment.

Exemple numérique

En utilisant les données du Tableau 4 (en remplaçant les valeurs 0 par des ϵ petits, pour des raisons numériques), avec $N = 10^5$ et une confiance de 95% nous obtenons :

- $\varepsilon = 0.5 : \mathbf{P}(A_1 \prec_\Omega A_2) = 0.6663 \pm 0.0029$
- $\varepsilon = 0.4 : \mathbf{P}(A_1 \prec_\Omega A_2) = 0.7353 \pm 0.0027$
- $\varepsilon = 0.2 : \mathbf{P}(A_1 \prec_\Omega A_2) = 0.8665 \pm 0.0021$

	A_1				A_2				Ω_ε			
	i_1	i_2	i_3	i_4	i_1	i_2	i_3	i_4	i_1	i_2	i_3	i_4
δ_1	0.3	0.1	0.4	0	0.2	0.6	0.2	0	1	0	0	1
δ_2	0.5	0.2	0.1	0.2	0	0.1	0.8	0.1	0	0	1	0
δ_3	0.6	0.2	0.1	0.1	0.5	0.3	0.1	0.1	1	0	0	0
δ_4	0.4	0.2	0.4	0	0.4	0.3	0.2	0.1	0	1	0	0
δ_5	0.1	0.8	0.1	0	0.9	0.1	0	0	1	0	0	0
δ_5	0.6	0.3	0	0.1	0.2	0.2	0.2	0.4	0	0	0	1

Tableau 4. *Exemple numérique pour l'évaluation de notre méthode basée sur la divergence Kullback-Leibler*

7. Conclusion and Perspectives

Dans cet article nous avons exploré quelques méthodes pour évaluer les performances d'algorithmes en présence de vérité terrain incertaine, notamment en exprimant la probabilité que le classement calculé puisse être modifié et en établissant des bornes en fonction de la qualité de la vérité terrain.

Notre modèle est capable d'exprimer la probabilité que l'ordre de classement entre deux algorithmes reste inchangée, malgré la connaissance de la vérité de terrain absolue. Nous l'avons exprimée de façon algébrique pour le cas de deux algorithmes et des données binaires, et validé une approche d'approximation de Monte-Carlo qui peut être étendue à des cas de multiples algorithmes. Nous avons également formalisé l'utilisation de la divergence de Kullback-Leibler dans le cas d'algorithmes d'interprétation n'opérant pas sur des valeurs binaires mais sur des valeurs de confiance sur les données.

Les limitations principales de nos modèles est qu'ils doivent être étendus à cas de classement de plusieurs algorithmes ($n > 2$) d'une part, et que le domaine des interprétations possibles puisse être étendu à d'autres types de valeurs que binaire ou niveau de confiance. Ces résultats théoriques et de simulation (bien que démontrés formellement) doivent encore être validés à une échelle plus grande et sur de vraies données (Lamiroy et Sun, 2013). Malheureusement, la plupart des résultats de *benchmarking* ne sont publiés que sous forme de courbes de rappel/précision, tandis que nos approches ont besoin de l'ensemble des résultats expérimentaux.

De futures pistes consistent également à étendre nos conclusions actuelles à des notions de précision et rappel.

8. Bibliographie

Al-Khaffaf H., Talib A., Osman M., « Final Report of GREC'11 Arc Segmentation Contest : Performance Evaluation on Multi-resolution Scanned Documents », *in* Y.-B. Kwon, J.-M. Ogier (eds), *Graphics Recognition. New Trends and Challenges*, vol. 7423 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 187-197, 2013.

Al-Khaffaf H., Talib A., Osman M., Wong P., « GREC'09 Arc Segmentation Contest : Performance Evaluation on Old Documents », *in* J.-M. Ogier, W. Liu, J. Lladós (eds), *Graphics Recognition. Achievements, Challenges, and Evolution*, vol. 6020 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 251-259, 2010.

Bukhari S., Al-Khaffaf H., Shafait F., Osman M., Talib A., Breuel T., « Final Report of GREC'13 Arc and Line Segmentation Contest », *in* B. Lamiroy, J.-M. Ogier (eds), *Graphics Recognition. Current Trends and Challenges*, vol. 8746 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 234-239, 2014.

Kullback S., Leibler R. A., « On Information and Sufficiency », *Ann. Math. Statist.*, vol. 22, n° 1, p. 79-86, 03, 1951.

Lamiroy B., « Interpretation, Evaluation and the Semantic Gap ... What if we Were on a Side-Track ? », *in* B. Lamiroy, J.-M. Ogier (eds), *10th IAPR International Workshop on Graphics Recognition, GREC 2013*, vol. 8746, Springer, Bethlehem, PA, United States, p. 213-226, August, 2013a.

Lamiroy B., On the Limits of Machine Perception and Interpretation, Habilitation à diriger des recherches, Université de Lorraine, December, 2013b.

Lamiroy B., Sun T., « Computing Precision and Recall with Missing or Uncertain Ground Truth », *in* Y.-B. Kwon, J.-M. Ogier (eds), *Graphics Recognition. New Trends and Challenges. 9th International Workshop, GREC 2011, Seoul, Korea, September 15-16, 2011, Revised Selected Papers*, vol. 7423 of *Lecture Notes in Computer Science*, Springer, p. 149-162, February, 2013.

Liu W., « The Third Report of the Arc Segmentation Contest », *in* W. Liu, J. Lladós (eds), *Graphics Recognition. Ten Years Review and Future Perspectives*, vol. 3926 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 358-361, 2006.

Metropolis N., Ulam S. M., « The Monte Carlo Method », *Journal of the American Statistical Association*, vol. 44, n° 247, p. 335-341, September, 1949.