Dompter les Distributions de Probabilité Géantes

Comment de nouveaux algorithmes pour estimer les paramètres de distributions sur des domaines gigantesques nécessitent considérablement moins de données.*

Ronitt Rubinfeld
(Traduction: Clément Canonne)

Septembre 2012

Ces derniers temps, le concept de *big data*, ainsi que l'absence d'outils efficaces pour gérer les énormes quantités d'information que cela entraîne, semblent être au centre de toutes les discussions. Bien souvent en « *big data* », les données considérées peuvent être vues comme des échantillons provenant d'une distribution de probabilité définie sur un très grand domaine de valeurs. Un tel scénario se retrouve dans l'immense majorité des domaines existants, des données de transactions financières aux enregistrements sismiques, en passant par les mesures d'activité neuronale, les réseaux de capteurs et les enregistrements d'activité sur les réseaux informatiques – pour ne citer qu'une fraction d'entre eux.

Dans la plupart des cas, la distribution de probabilité en question n'est pas explicitement fournie – il est uniquement possible d'en observer des réalisations. En vue de tirer parti de ces données, il est nécessaire d'estimer certains paramètres caractéristiques ainsi que des propriétés fondamentales de la distribution sous-jacente. Par exemple, combien de valeurs du domaine ont une probabilité non nulle d'apparaître ? Est-ce que la distribution de probabilité est une loi uniforme, Gaussienne, de Zipf ? Est-ce que les différentes variables des observations sont indépendantes ? Quelle est l'entropie de la distribution ? Il est possible de répondre à toutes ces questions de manière relativement aisée, *via* des méthodes statistiques classiques.

Cependant, à moins de faire des hypothèses supplémentaires sur la nature de la distribution de probabilité inconnue – par exemple, qu'il s'agit d'une loi de probabilité Gaussienne, ou suffisamment régulière – ces méthodes requièrent un nombre d'observations qui croît de manière (a minima) linéaire avec la taille du domaine de la distribution. Malheureusement, l'idée même de *big data* implique que les domaines en question sont de taille gigantesque, et par conséquent que le nombre d'échantillons nécessaire est lui-même énorme. Les algorithmes habituellement appliqués deviennent dès lors impossibles à utiliser du fait de leur lenteur.

Tout n'est pas désespéré, toutefois : dernièrement, d'importantes avancées ont été faites dans le développement d'algorithmes *sous-linéaires* (en termes d'observations) pour ces problèmes. Dans cet article sont ainsi décrits deux récents résultats qui illustrent les idées principales derrière ces progrès : le premier permet de tester si deux distributions de probabilité sont similaires, et le second d'estimer l'entropie d'une distribution. En dehors de l'hypothèse que leur domaine D est un ensemble fini de n éléments, les distributions considérées sont supposées *a priori* quelconques.

Similitude entre distributions

Comment déterminer si deux distributions sont identiques ? Beaucoup de variantes de cette question ont été formulées, mais considérons pour l'instant un problème plus simple, motivé par l'exemple suivant : de combien d'années de résultats de loterie aurions-nous besoin pour être convaincu que les résultats ne sont pas truqués ? Ou, en d'autres termes : ayant accès à des échantillons d'une distribution de probabilité p, combien d'entre eux nous faut-il pour déterminer si p est la loi uniforme ?

Pour formaliser ce problème correctement, il est nécessaire d'accepter une certaine forme d'approximation : en effet, p pourrait être arbitrairement proche d'une distribution uniforme (tout en n'étant pas *exactement* uniforme), auquel cas aucun algorithme n'utilisant qu'un nombre fini d'échantillons n'aurait assez d'information pour détecter la différence. C'est pourquoi nous nous placerons dans le cadre du *test de propriété*[1] : l'algorithme de test aura uniquement à « accepter » les distributions uniformes, et « refuser » celles qui sont loin de l'être. Ce qui nous amène à la question suivante : spécifier ce que « loin » signifie dans ce contexte. Bien qu'il existe de nombreuses mesures communément employées pour quantifier la distance entre distributions, nous nous focaliserons dans cet article sur la distance ℓ_1 entre deux distributions de probabilité p et q, définie de la manière suivante :

$$\|p - q\|_1 \stackrel{\text{def}}{=} \sum_{x \in D} |p(x) - q(x)|$$

*Traduction de *"Taming Big Probability Distributions"*, par Ronitt Rubinfeld [1] (Traduit en août 2013).

1. En anglais *property testing*.

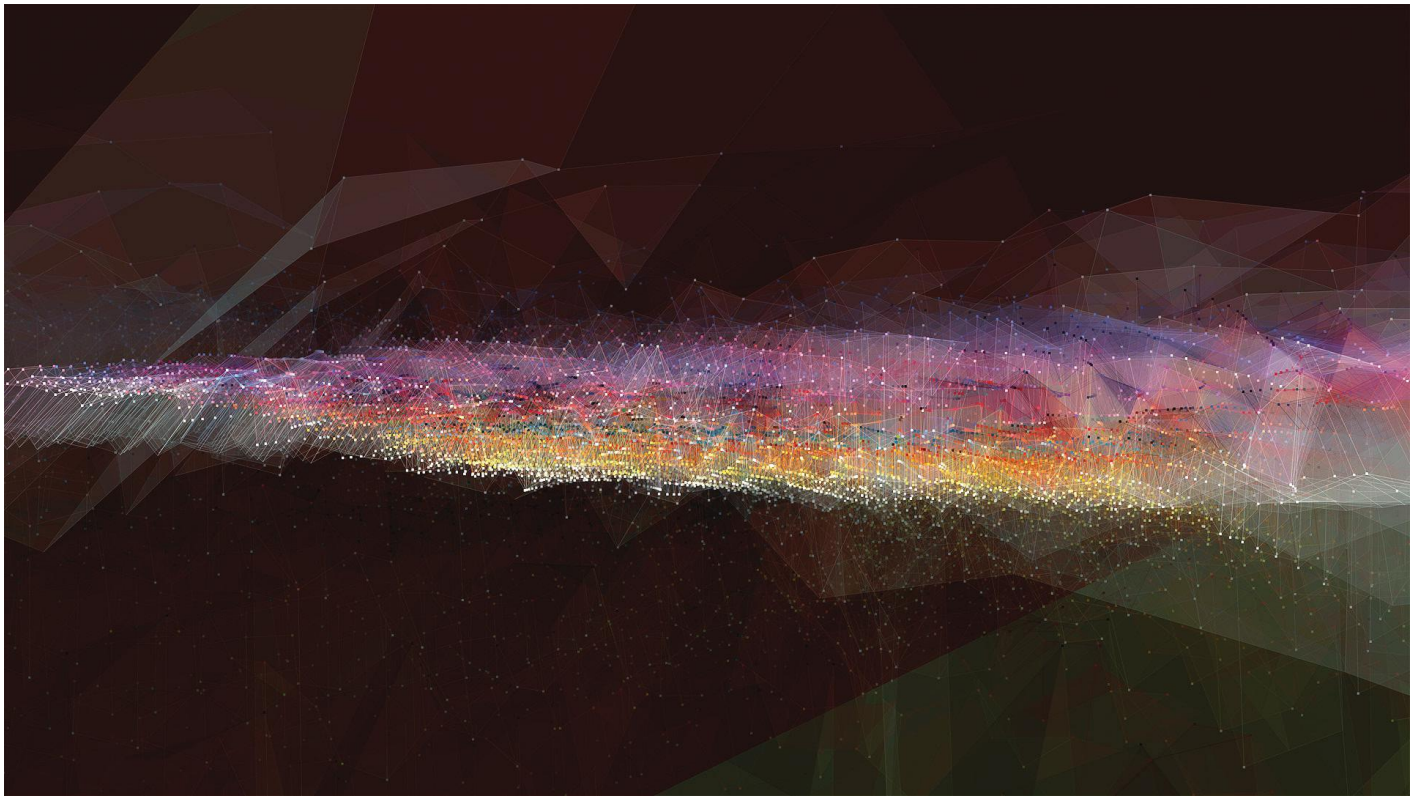


Illustration par Gwen Vanhee (générée *via* HTML5 et Javascript).

Pour une valeur $0 < \epsilon < 1$, nous dirons que p et q sont ϵ-*proches* en distance ℓ_1 si $\|p - q\|_1 \leq \epsilon$. Si l'on désigne par $\mathcal{U}_D$ la loi uniforme de domaine D, le rôle du testeur est alors, étant donné un paramètre $0 < \epsilon < 1$, d'accepter p s'il s'agit de la loi uniforme et de rejeter si $\|p - \mathcal{U}_D\|_1 \geq \epsilon$. Si p est entre les deux – pas uniforme, mais pas trop loin de l'être non plus – accepter ou rejeter sont tous les deux des réponses possibles (et pertinentes).

Une façon intuitive de résoudre ce problème, l'« algorithme naïf », est de demander suffisamment d'observations distribuées suivant p pour obtenir une bonne approximation de la valeur $p(x)$ pour chacun des éléments x du domaine. Il n'est pas difficile de se convaincre que dans certains cas cette méthode nécessitera un nombre d'échantillon au moins linéaire en $|D| = n$.

Toutefois, il existe une approche bien plus efficace, reposant sur une idée de Goldreich et Ron [2] et ne requiérant que $O(\sqrt{n}/\epsilon^4)$ échantillons (voir aussi Paninski [3] pour un algorithme plus récent, qui ne demande que $O(\sqrt{n}/\epsilon^2)$ observations). Cet algorithme n'essaie pas d'apprendre la probabilité que p alloue aux différents éléments du domaine ; à la place, il compte les collisions – le nombre de paires d'échantillons qui « tombent » sur la même valeur.

Plus précisément, pour une liste de k échantillons $x_1, \ldots, x_k$, et $i, j \in [\![1, k]\!]$ deux indices quelconques, nous dirons que i et j entrent en collision s'ils pointent vers le même élément du domaine, c'est-à-dire si $x_i = x_j$. Une observation importante est que la probabilité que i et j entrent en collision ne dépend pas du choix de i et j, et constitue un paramètre important de la distribution p, la *probabilité de collision* C_p. Il est aisé de remarquer que la fraction de paires i, j qui entrent en collision au sein de l'échantillon a pour espérance cette même quantité C_p ; en outre, un calcul assez simple montre que C_p est minimum lorsque p est la loi uniforme, auquel cas sa valeur est $C_p = 1/n$. Il est également possible de prouver que si p est loin d'être uniforme, C_p diffère de manière significative de $1/n$. Dès lors, il devrait être clair que cette probabilité de collision C_p est une quantité judicieuse à estimer. Ce qui la rend d'autant plus appropriée est le fait qu'un nombre étonnamment faible d'observations suffit à cela : en effet, avec k d'entre eux, c'est de $\binom{k}{2}$ paires dont l'on dispose pour estimer la probabilité de collision.

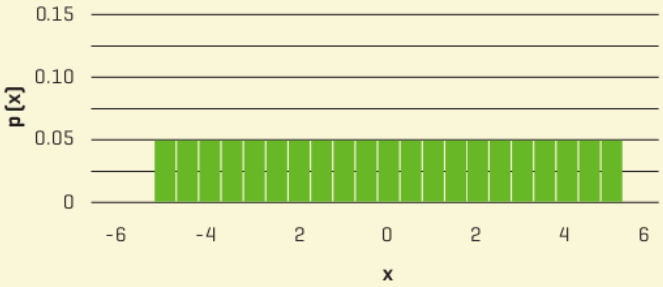
Bien que ces paires d'échantillons ne soient pas indépendantes, Goldreich et Ron ont montré qu'elles possédaient tout de même certaines agréables propriétés [2], et en déduisent un algorithme qui calcule approximativement la probabilité de collision, n'utilisant que $O(\sqrt{n} \log n/\epsilon^4)$ observations – apportant une réponse à notre problème d'équivalence à la loi uniforme. Il s'avère que la dépendance en n du nombre d'échantillons ne peut guère être améliorée : en effet, il est facile de se persuader que les probabilités de collision « généralisées » (c'est-à-dire les nombres de collisions entre sous-ensembles de ℓ échantillons, pour toutes valeurs de ℓ) constituent la seule information pertinente qu'un algorithme peut exploiter en vue de tester si une distribution est uniforme. En réalité, il s'agit même de la seule information disponible afin

FIGURE 1 – Distributions

(a) Loi uniforme

Dans le cas d'une loi uniforme, les n éléments consécutifs du domaine ont tous la même probabilité d'être observés.

$$p(x) = \frac{1}{n}$$

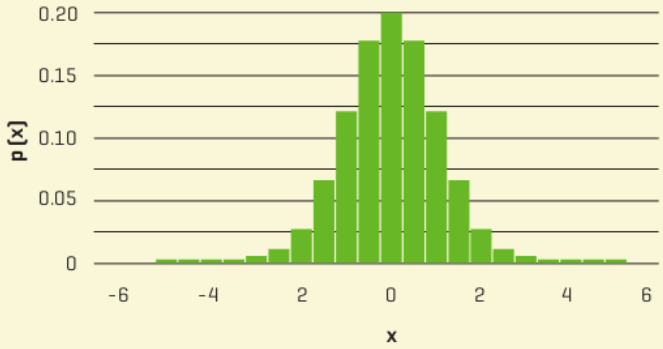


(b) Loi Gaussienne

La distribution Gaussienne (ou normale) est souvent décrite par l'expression *courbe en cloche*, en raison de la forme de sa fonction de densité

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

où μ est la moyenne et σ l'écart-type. Elle constitue l'une des distributions les plus fréquemment utilisées en statistiques.

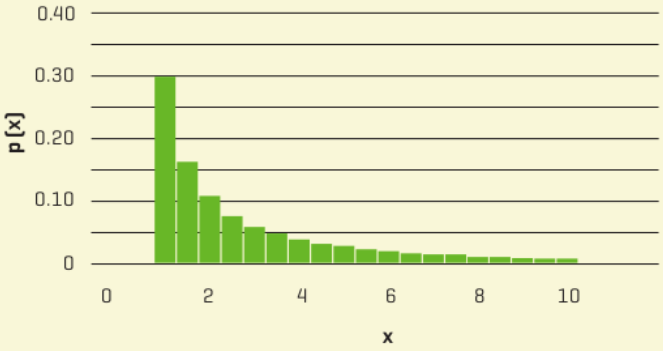


(c) Loi de Zipf

La loi de Zipf trouve des applications en linguistique, finance, et dans la modélisation de phénomènes rares. Sa fonction de densité est

$$p(x) = \frac{x^{-(p+1)}}{\zeta(p+1)}$$

où p est une constante positive et ζ dénote la fonction zêta de Riemann.



de décider si p possède n'importe laquelle d'un grand nombre de propriétés – plus précisément, n'importe quelle propriété (dite « symétrique ») qui ne dépend pas de la façon dont les éléments du domaine sont numérotés. En se basant sur cette observation, il suffit d'exhiber une distribution qui ne génère aucune collision tant que moins de $\Omega(\sqrt{n})$ sont observés, mais qui en dépit de cela est loin d'être d'uniforme. Une telle distribution de probabilité peut être obtenue en choisissant aléatoirement un sous-ensemble S contenant la moitié des éléments du domaine, et définissant la loi uniforme sur S [4].

Et *quid* de savoir si p est une loi Gaussienne standard ? Ou plus généralement si p est égale à une autre distribution q, qui elle est connue et explicitement fournie à l'algorithme – en d'autre termes, ce dernier a librement accès aux valeurs $q(i)$ pour tout élément i ? Un tel cas de figure se présente si par exemple q est une distribution Gaussienne, de Zipf ou exponentielle de paramètres (espérance et variance) donnés. Batu et al. décrivent un algorithme qui répond à cette question pour n'importe quelle distribution q fixée qui utilise $O(\sqrt{n}\log n)$ échantillons de p et évalue $O(\log n)$ probabilités de collision sur certains sous-ensembles spécifiques de D [5].

Enfin, que dire du cas où p et q sont tous deux inconnues, et où la seule façon d'obtenir des informations à leur sujet est d'en observer des réalisations ? Jusqu'ici, et bien que l'analyse des algorithmes évoqués ne soit pas immédiate, leur complexité en termes d'échantillons n'est pas si surprenante que cela pour quiconque a déjà rencontré des raisonnements du type « paradoxe des anniversaires » (collisions, fonctions de hachages . . .).

C'est là que la situation prend une tournure inattendue : dans ce dernier cas, la complexité du problème est significativement différente de $n^{1/2}$. La raison ? Il est désormais possible que p et q coïncident sur certains éléments assez « lourds », suffisamment pour

que les chances d'observer une collision sur l'un d'entre eux occultent ce qui se passe sur le reste du domaine. Formaliser de manière rigoureuse cette intuition quant au minimum d'observations requises (« *sample lower bound* ») est tout sauf immédiat, et a résisté aux efforts des chercheurs durant de nombreuses années ; cependant, en 2008, Paul Valiant a été en mesure de prouver que $\Omega\left(n^{2/3}\right)$ échantillons étaient nécessaires à ce problème [6, 7]. L'algorithme proposé en 2000 par Batu et al. [4, 8], qui en demande $O\left(n^{2/3}\log n\,\mathrm{poly}(1/\epsilon)\right)$ et distingue les distributions p, q identiques de celles qui sont au moins à une distance ϵ l'une de l'autre procède de la manière suivante :

1. Tout d'abord, il détermine les éléments « lourds » du domaine, ceux ayant une probabilité au moins $1/n^{2/3}$ d'apparaître. Cette définition implique en particulier qu'il n'y aura au plus que $n^{2/3}$ de ces éléments lourds, puisque que la somme des probabilités de l'ensemble des points du domaine est égale à 1. L'algorithme naïf consistant à observer $O\left(n^{2/3}\log n\,\mathrm{poly}(1/\epsilon)\right)$ échantillons de p et q afin d'estimer les probabilités de chacun de ces éléments lourds a alors de grandes chances de fournir de bonnes approximations de celles-ci.
2. Si p et q ont l'air semblables à l'issue de cette première étape, l'algorithme vérifie alors que c'est aussi le cas sur le reste du domaine, en éliminant des échantillons qu'il obtient les éléments lourds et appliquant un test basé sur les probabilités de collision – cette fois-ci, pas seulement celles de p et q, mais également les collisions entre échantillons de p et échantillons de q. Puisqu'à présent aucun des éléments considérés n'est lourd, il est possible de démontrer qu'à nouveau $O\left(n^{2/3}\log n\,\mathrm{poly}(1/\epsilon)\right)$ observations sont suffisantes pour cette tâche.

Des idées semblables ont par la suite été appliquées à d'autres problèmes ; notamment, afin d'obtenir des algorithmes testant si une distribution a une densité de probabilité croissante ou bimodale sur son domaine de définition [9], ou bien si les variables marginales définies par une distribution jointe sont indépendantes [5]. La complexité de beaucoup de ces problèmes, en termes d'échantillons, a en outre été étudiée pour d'autres métriques que la distance ℓ_1 [2, 10, 11], mais les mêmes techniques basées sur le nombre de collisions sont souvent mises en œuvre. Tester l'équivalence de deux distributions a fait l'objet de recherches approfondies, et bien d'autres résultats existent à ce sujet [10, 12, 13].

Un testeur *tolérant* est un algorithme qui, étant donnés des paramètres $\epsilon_1 < \epsilon_2$, accepte les distributions p qui sont ϵ_1-proches de q et rejette celles qui n'en sont même pas ϵ_2-proches. Malheureusement, même dans le cas plus simple où l'on cherche à savoir si p est la distribution uniforme, Valiant a démontré que pour ϵ_1 suffisamment grand le problème devient bien plus ardu, et requiert un minimum de $n^{1-o(1)}$ échantillons [6, 7] (par comparaison, [14] établit que $O(n/(\epsilon^2\log n))$ sont suffisants). Néanmoins, les algorithmes de test classiques, plus efficaces, permettent d'obtenir une certaine marge de tolérance – bien que souvent infime. Il serait intéressant de voir si et jusqu'à quel point ceci peut être amélioré.

Estimer l'entropie d'une distribution

L'entropie d'une distribution est une mesure caractéristique de la « quantité d'aléatoire » qu'elle comporte, ainsi que de la compressibilité des données qui en proviennent. Pour cette raison, l'entropie joue un rôle prépondérant en statistique, théorie de l'information, compression de données et *machine learning*. L'entropie d'une distribution p de domaine (discret) D est définie de la manière suivante :

$$H(p) \stackrel{\text{def}}{=} \sum_{x\in D} -p(x)\log p(x)$$

L'estimation de l'entropie d'une distribution de probabilité et des quantités apparentées que sont la divergence de Kullback-Leibler et l'information mutuelle ont suscité beaucoup d'intérêt du fait de leurs applications dans l'analyse de données en machine learning et dans les sciences expérimentales [15, 16]. Combien d'observations indépendantes d'une distribution sont-elles nécessaires en vue d'obtenir une bonne approximation de son entropie ?

En premier lieu, il convient de définir ce que l'on entend par « bonne approximation ». Commençons par considérer le cas où l'on désire estimer l'entropie de manière additive – c'est-à-dire obtenir une valeur y vérifiant

$$H(p) - \epsilon < y < H(p) + \epsilon$$

où ϵ est un paramètre fourni en entrée. Une méthode couramment utilisée pour cela, connue sous le nom de « estimateur par substitution » [2], s'appuie sur l'obtention préalable d'une hypothèse sur l'ensemble de la distribution p. Plus précisément, si $\hat{p}(x)$ dénote la fraction des échantillons qui tombent sur un élément x du domaine, l'estimateur correspondant pour l'entropie sera l'entropie de $\hat{p}$, c'est-à-dire

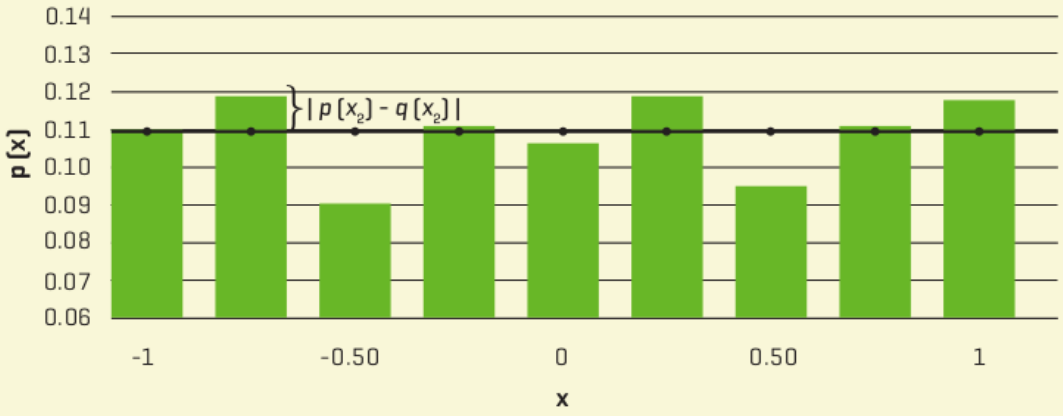
$$\hat{H} \stackrel{\text{def}}{=} \sum_{x\in D} -\hat{p}(x)\log \hat{p}(x)$$

Sans trop de surprise, si l'on souhaite que cette approximation ne soit pas trop mauvaise, il est nécessaire d'avoir suffisamment d'observations pour obtenir une bonne estimation de $p(x)$ pour la plupart des x ; ce qui en général implique d'utiliser un nombre d'échantillons au moins linéaire en n.

Les autres estimateurs les plus fréquemment rencontrés (Miller-Madow, méthode du jackknife ...), présentent également cette dépendance linéaire en n dans le nombre

2. *Plugin estimate* en anglais.

FIGURE 2 – La distance ℓ_1 entre p et q est la somme de ces quantités pour chaque élément x_i du domaine



d'observations ; ceci provient du fait que toutes ces techniques ne prennent pas convenablement en compte la contribution à l'entropie provenant des éléments du domaines non observés lors de l'échantillonnage.

Un cap important a été franchi lorsque Paninski a démontré (de manière non-constructive) qu'il existait un estimateur pour l'entropie d'une distribution n'utilisant qu'un nombre d'échantillons négligible (sous-linéaire) par rapport à la taille de son domaine [17]. Plus récemment, un résultat de Gregory et Paul Valiant a fait grand bruit, en résolvant définitivement la question du nombre exact d'observations nécessaires à ce problème [14, 18, 19, 20]. D'une part, ils décrivent un algorithme estimant l'entropie d'une distribution définie sur un domaine de cardinal n, à un facteur additif ϵ près, qui n'utilise que $O(n/(\epsilon^2 \log n))$ échantillons ; d'autre part, ils démontrent qu'il est impossible de le faire avec moins de $O(n/(\epsilon \log n))$ d'entre eux (améliorant ainsi l'état de l'art [7, 21]). Afin d'établir le premier point, ils reformulent la question en tant que problème d'optimisation linéaire (*linear programming*) dont la solution est une distribution présentant des probabilités de collision similaires à celles de p. Cette distribution, bien que potentiellement très différente de p en termes de distance ℓ_1, partage au moins avec elle un aspect crucial – elles ont toutes deux des entropies voisines. La démonstration du second point, quant à lui, repose sur la construction de deux familles de distributions qui, bien qu'ayant des probabilités de collision très proches, sont néanmoins éloignées en matière d'entropie ou de taille de support effectif.

Tournons-nous à présent vers un autre cas de figure, où – revoyant nos ambitions à la baisse – l'on ne cherche à estimer l'entropie que de manière *multiplicative*. En d'autres termes, étant donné un paramètre $\gamma > 1$, l'algorithme doit renvoyer une valeur y telle que

$$H(p)/\gamma < y < \gamma H(p)$$

Il s'avère qu'obtenir une approximation de ce type nécessite radicalement moins d'échantillons, en particulier relativement à la taille du domaine : il existe pour cela des algorithmes qui n'en utilisent que $O\left(n^{(1+o(1))/\gamma^2}\right)$ [22]. (Ceci est à vrai dire légèrement incorrect, et ne s'applique en réalité qu'aux distributions dont l'entropie est supérieure à une certaine valeur indépendante de n et connue *a priori*) Qui plus est, il a été établi qu'un minimum de $O\left(n^{1/\gamma^2}\right)$ observations était nécessaire pour cette tâche [6, 7]. Pour donner un exemple concret, cela signifie qu'il est possible d'approximer l'entropie à un facteur deux près en n'utilisant que légèrement plus de $O\left(n^{1/4}\right)$ échantillons, soit significativement moins que ce qu'il faudrait pour le cas d'une estimation additive. L'algorithme lui-même est remarquabelement simple : il fait appel à la méthode de substitution pour tous les points du domaine ayant une grande probabilité d'apparaître, et se contente de supposer que le reste de la distribution est uniforme.

Des résultats similaires peuvent être obtenus pour un problème voisin : celui d'approximer la taille du support effectif de la distribution. Cette question, soulevée au début des années 1940 par Fisher et Corbet qui cherchaient à d'estimer le nombre d'espèces de papillons d'une certaine région, a depuis fait l'objet de beaucoup d'attention et été étudiée de manière intensive (une liste considérable de raisons expliquant l'intérêt pour cette question est disponible sur [23]). Des avancées récentes à ce sujet ont permis de conclure que $\Theta(n/\log n)$ observations étaient à la fois nécessaires et suffisantes pour obtenir une estimation additive de la taille du support [14].

Résumé et mot de la fin

Les défis suscités par les *big data* ont poussé la communauté des chercheurs en informatique à accomplir d'excitants progrès sur des problèmes de Statistique notoires. Cependant, dans certaines situations, la quantité

minimale de données nécessaire en vue d'obtenir une réponse acceptable est trop grande pour être utilisable en pratique.

Une façon de remédier à cela est de mettre au point des algorithmes spécialisés qui tireraient parti des propriétés de certaines distributions – par exemple, celles dont la densité de probabilité est suffisamment régulière, monotone, ou qui appartiennent à une classe spécifique telle que les lois normales. Des hypothèses supplémentaires de ce type permettent bien souvent d'obtenir des algorithmes considérablement plus efficaces.

Une seconde méthode pour contourner cette limite est de se pencher plus avant sur les paramètres du problème lui-même : dans un certain nombre de cas, il est naturel de supposer qu'en sus d'observations de la distribution inconnue, l'algorithme a accès à d'autres sources d'information – par exemple, qu'il lui est possible de déterminer rapidement la valeur de $p(x)$ pour n'importe quel élément fixé x du domaine. Cela peut se produire lorsque l'on cherche à étudier les propriétés de la répartition de données stockées de manière ordonnée (e.g, triées) : bien qu'il soit toujours possible d'obtenir un échantillon aléatoire de cet ensemble de données, il est également facile de calculer le nombre de collisions, ou le nombre d'occurrences d'une valeur particulière. Cette source d'information supplémentaire peut être exploitée afin d'obtenir des algorithmes sous-linéaires (par rapport au nombre d'échantillons autrement nécessaire), et donc de réduire significativement la complexité algorithmique du problème.

Ces nouveaux types d'approches en modélisation statistique peut permettre la conception d'algorithmes considérablement plus rapides pour traiter des distributions définies sur des domaines de plus en plus larges. Il devient dorénavant crucial de tirer parti de ces algorithmes et modèles statistiques plus riches – ce sont les instruments qui nous permettront de dompter *big data*.

Remerciements

L'auteur tient à remercier Reut Levi et Ning Xie pour leurs précieux commentaires et remarques lors de l'écriture de cet article.

Biographie

Ronitt Rubinfeld est professeur d'*electrical engineering* et d'informatique au MIT, ainsi que professeur d'informatique à l'Université de Tel-Aviv. Ces 20 derniers années, ses recherches se sont concentrées sur l'étude d'algorithmes sous-linéaires pour tous types de vastes ensembles de données.

Références

[1] Ronitt Rubinfeld. Taming big probability distributions. `http://xrds.acm.org/article.cfm?aid=2331052`, September 2012.

[2] O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. *Electronic Colloqium on Computational Complexity*, 7(20), 2000.

[3] L. Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Trans. Inf. Theor.*, 54(10) :4750–4755, October 2008.

[4] Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing that distributions are close. In *IEEE Symposium on Foundations of Computer Science*, pages 259–269, 2000.

[5] T. Batu, E. Fischer, L. Fortnow, R. Kumar, R. Rubinfeld, and P. White. Testing random variables for independence and identity. pages 442–451, 2001.

[6] Paul Valiant. Testing symmetric properties of distributions. pages 383–392, 2008.

[7] P. Valiant. Testing symmetric properties of distributions. *SIAM Journal on Computing*, 40(6) :1927–1968, 2011.

[8] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing closeness of discrete distributions. Technical Report abs/1009.5397, 2010. Version intégrale de [4].

[9] T. Batu, R. Kumar, and R. Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. pages 381–390, 2004.

[10] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian. Sublinear estimation of entropy and information distances. *ACM Trans. Algorithms*, 5(4) :35 :1–35 :16, November 2009.

[11] Khanh Do Ba, Huy L. Nguyen, Huy N. Nguyen, and Ronitt Rubinfeld. Sublinear time algorithms for earth mover's distance. *Theor. Comp. Sys.*, 48(2) :428–442, February 2011.

[12] Jayadev Acharya, Hirakendu Das, Ashkan Jafarpour, Alon Orlitsky, Shengjun Pan, and Ananda Theertha Suresh. Competitive classification and closeness testing. pages 22.1–22.18, 2012.

[13] Reut Levi, Dana Ron, and Ronitt Rubinfeld. Testing properties of collections of distributions. In *Innovations in Computer Science*, pages 179–194, 2011. Voir également ECCC TR10-157.

[14] Gregory Valiant and Paul Valiant. The power of linear estimators. In *FOCS*, 2011.

[15] Shang-Keng Ma. Calculation of entropy from data of motion. *Journal of Statistical Physics*, 26(2) :221–240, 1981.

[16] S. P. Strong, Roland Koberle, Rob R. de Ruyter van Steveninck, and William Bialek. Entropy and information in neural spike trains. *Physical Review Letters*, 80(1) :197–200, Jan 1998.

[17] L. Paninski. Estimating entropy on m bins given fewer than m samples. 50(9) :2200–2203, 2004.

[18] G. Valiant and P. Valiant. A CLT and tight lower bounds for estimating entropy. Technical Report TR10-179, 2010.

[19] G. Valiant and P. Valiant. Estimating the unseen : A sublinear-sample canonical estimator of distributions. Technical Report TR10-180, 2010.

[20] G. Valiant and P. Valiant. Estimating the unseen : an $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs. pages 685–694, 2011.

[21] S. Raskhodnikova, D. Ron, A. Shpilka, and A. Smith. Strong lower bonds for approximating distributions support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3) :813–842, 2009.

[22] T. Batu, S. Dasgupta, R. Kumar, and R. Rubinfeld. The complexity of approximating the entropy. *SIAM Journal on Computing*, 35(1) :132–150, 2005.

[23] J. Bunge. Estimating the number of classes in a population.