Comparaison de bornes théoriques pour l'accélération du clustering incrémental en une passe

Nicolas Labroche, Marcin Detyniecki, Thomas Baerecke

UPMC Paris 6, LIP6 UMR CNRS 7606
BC 169, 4 place Jussieu 75252 Paris Cedex 05, France
{prenom.nom@lip6.fr}

Résumé. Le clustering incrémental en une passe repose sur l'affectation efficace de chaque nouveau point aux clusters existants. Dans le cas général, où les clusters ne peuvent être représentés par une moyenne, la détermination exhaustive du cluster le plus proche possède une complexité quadratique avec le nombre de données. Nous proposons dans ce papier une nouvelle méthode d'affectation stochastique à chaque cluster qui minimise le nombre de comparaisons à effectuer entre la donnée et chaque cluster pour garantir, étant donné un taux d'erreur acceptable, l'affectation au cluster le plus proche. Plusieurs bornes théoriques (Bernstein, Hoeffding et Student) sont comparées dans ce papier. Les résultats sur des données artificielles et réelles montrent que la borne de Bernstein donne globalement les meilleurs résultats (notamment lorsqu'elle est réduite) car elle permet une accélération forte du processus de clustering, tout en conservant un nombre très faible d'erreurs.

1 Introduction

Le clustering permet l'exploration d'ensembles de données en les résumant sous la forme de groupes homogènes plus facilement caractérisables et interprétables. Récemment, de nouveaux algorithmes ont été proposés pour répondre aux problèmes de traitement des grands volumes ou des flux de données (Aggarwal et al., 2003; Cao et al., 2006; Philipp Kranen et Seidl, 2011). Ces méthodes reposent généralement sur des algorithmes qui ne réalisent qu'une seule passe sur les données initiales. Ceux-ci ne sont malheureusement applicables qu'à des données vectorielles, pour lesquelles des structures incrémentales de description des clusters existent (Zhang et al., 1996).

Dans ce travail, nous nous intéresssons au cas général où les données ne sont pas nécessairement vectorielles, et où il n'est donc pas possible d'utiliser de telles structures. Une solution consiste à résumer chaque cluster par un sous-ensemble des données qui le compose, possiblement un seul point, appelé médoide, qui est le plus similaire aux autres données du cluster. Le problème est que la détermination des médoides, et donc l'affectation d'un nouveau point aux clusters existants dans un contexte incrémental, possède une complexité quadratique avec le nombre de données. Cela n'étant pas envisageable dans des cas d'usage réels, les algorithmes implémentent généralement des mécanismes d'échantillonnage pour réduire le coût des calculs

de l'appartenance à un cluster. Cependant, échantillonner implique l'introduction d'une incertitude dans la représentation du cluster, qui, dans le cas d'un algorithme en une passe, peut se traduire par une erreur d'affectation d'un point à un cluster, et qui est aggravé par le fait que cette donnée peut ensuite être, à tort, utilisée pour représenter le cluster auquel elle appartient.

Dans ce papier, nous proposons une méthode stochastique d'affectation d'un point à un cluster, qui s'inspire des principes des inégalités de concentration en mettant en œuvre des bornes théoriques qui vont estimer la distance réelle d'un point à chaque cluster et ainsi gérer l'incertitude liée à l'échantillonnage. Nous comparons ici trois bornes théoriques : Bernstein, Hoeffding et Student. Nous proposons également de réduire artificiellement ces bornes à l'aide d'un pourcentage pour en accélérer la convergence, mais au prix d'erreurs plus nombreuses.

Les résultats expérimentaux sur des jeux de donnés artificiels ou réels issus du répertoire UCI Machine Learning Repository visent ici à évaluer l'accélération du processus de clustering en une passe tout en comptabilisant les erreurs d'affectation par rapport à un algorithme exhaustif, mais ne s'intéressent pas à évaluer la qualité de la partition obtenue en tant que telle par les indices habituels (Rand, erreur de confusion). Nos résultats montrent que les meilleurs performances sont obtenues par la borne de Bernstein qui offre dans tous les cas le meilleur ratio "nombre de comparaisons entre données par nombre d'erreurs observées". Les expérimentations montrent par ailleurs que la réduction des bornes théoriques permet d'en améliorer les performances en pratique.

Cet article est organisé comme suit : la section 2 présente un état de l'art succinct des principales méthodes d'échantillonnage utilisées pour l'accélération du clustering de données non vectorielles et leurs limites. La section 3 décrit les principes généraux de notre méthode de sélection de cluster et décrit les différentes bornes théoriques ainsi que leurs variantes réduites. Ensuite, la section 4 présente les résultats comparatifs expérimentaux entre la méthode exhaustive, qui est considérée comme la vérité terrain, et les approches basées sur les bornes théoriques et réduites sur l'ensemble des bases de tests. Finalement, la section 5 présente les conclusions et les perspectives de ce travail.

2 Méthodes d'échantillonnage pour le clustering de données non vectorielles

Deux approches principales existent pour le traitement de données non vectorielles (Hammer et Hasenfuss, 2007) : les approches basées sur des médoides qui limitent les coordonnées des centres des clusters à des exemples du jeu de données, et les approches basées sur des données relationnelles qui travaillent directement à partir des matrices de distances ou de (dis)similarité. Dans les deux cas, des méthodes d'échantillonnage ont été proposées pour réduire la complexité quadratique de leur résolution.

Ainsi, l'algorithme CLARA (Kaufman et Rousseeuw, 1990) réduit la complexité en échantillonnant alétoirement l'ensemble du jeu de données. D'autres méthodes, comme CLARANS (Ester et al., 1995) ou CURE (Guha et al., 1998) limitent la recherche des candidats médoides aux voisins des médoides actuels, tout comme la méthode floue Linearized Fuzzy C-Medoids (Krishnapuram et al., 1999) qui utilise pour cela les degrés d'appartenance des points aux clusters. D'autres méthodes, comme l'algorithme Leader Ant (Labroche, 2006) remplace

la détermination exhaustive du médoide par un nombre fixé réduit de comparaisons aléatoires avec chaque cluster.

Dans (Zhu et al., 2012), les auteurs présentent des mécanismes d'accélération pour les données relationnelles de type approximation de Nyström, visant à réduire la dimension de la matrice de distance, ou de type "patch processing", ne considérant qu'un échantillon carré de la matrice de distances à la fois pour traiter de grands volumes de données.

Cependant, pour toutes ces méthodes, le taux d'échantillonnage est un paramètre, qui est non seulement difficile à déterminer a priori car il manque de sens, mais il est également le même tout au long du processus de clustering indépendamment par exemple de la taille des clusters ou de la complexité de leur forme. Pour éviter cela, (Domingos et al., 2001) proposent une variante de k-means, limitée aux données vectorielles, qui utilise une borne de Hoeffding (Maron et Moore, 1994) pour minimiser le nombre de données nécessaires à la détermination des centres de chaque cluster, tout en garantissant que l'erreur commise reste bornée pour un taux d'erreur fixé.

Similairement, l'idée de ce papier est de proposer un cadre général pour accélérer les algorithmes de clustering en une passe, dans le cas de données non nécessairement numériques, à l'aide d'un mécanisme de mise en compétition des clusters reposant sur des bornes théoriques, qui va permettre de gérer l'incertitude sur les distances issue de l'échantillonnage.

3 Accélération des algorithmes de clustering en une passe

3.1 Sélection des clusters dans les algorithmes en une passe

Dans le cas particulier des algorithmes de clustering en une seule passe, les données sont considérées séquentiellement et l'affectation à un cluster dépend uniquement de l'estimation de la distance de cette donnée aux clusters existants. Bien sûr, l'ordre des données a une influence directe sur la partition produite par ces méthodes. Généralement, celles-ci sont paramétrées à l'aide d'un seuil de distance qui est utilisé pour décider si une donnée est suffisamment proche des clusters existants pour en intégrer un ou bien si elle doit initier son propre cluster. Lorsque plusieurs (éventuellement tous) les clusters sont éligibles, le problème revient à déterminer le cluster qui optimise le mieux la fonction objectif de l'algorithme de clustering. Ce problème est encore plus compliqué lorsque la distance est estimée à partir d'un échantillon des données de chaque cluster comme dans ce travail.

Nous proposons un mécanisme de sélection des clusters inspiré du mécanisme de compétition ou "racing" introduit par (Heidrich-Meisner et Igel, 2009) qui peut s'appliquer au problème de clustering pour adapter automatiquement la taille de l'échantillon nécessaire à l'affectation d'un point à un cluster sur la base d'une erreur maximale tolérée d'affectation. Ce faisant, nous pouvons simultanément accélérer les algorithmes de clustering en une passe, en limitant le nombre de calculs de distances entre une nouvelle donnée et les données déjà classées, et également garantir une borne supérieure sur l'erreur d'affectation par rapport à un algorithme exhaustif qui réaliserait toutes les comparaisons possibles, sous l'hypothèse que toutes les comparaisons sont indépendantes. En pratique, comme le montre nos expérimentations, et bien que l'hypothèse d'indépendance ne soit pas nécessairement vérifiée, cela conduit à réduire drastiquement le nombre de comparaisons nécessaires tout en limitant les erreurs à des niveaux très faibles.

3.2 Méthode statistique de compétition (racing) entre clusters

La technique de "racing" est un outil qui permet de prendre une décision dans le cas d'une incertitude résultant d'au moins deux variables aléatoires ayant un recoupement partiel de leur intervalle de confiance, étant donné un taux d'erreur fixé. L'idée du racing est que la comparaison entre deux (ou plus) variables aléatoires peut être affinée lorsque plus de réalisations des variables aléatoires sont observées. Avec peu de réalisations, la variance est très large et la plupart des distributions attachées aux variables aléatoires se recoupent. Lorsque plus d'observations sont réalisées, la variance décroît et il existe un moment où les distributions se séparent (exception faite de distributions exactement identiques ou s'il n'y a pas assez de données à échantillonner pour chaque variable aléatoire). Après un certain nombre d'observations, il devient possible de prendre une décision sur la relation des deux variables aléatoires (plus grand / plus petit en fonction de l'objectif du problème) avec un certain niveau de confiance, qui correspond à la borne supérieure de l'erreur qui est tolérée. Sur la base des relations estimées (plus petit / plus grand) les mauvais clusters candidats sont éliminés plus tôt, même si la variance demeure assez grande. Cela conduit à la concentration des efforts de calcul sur les meilleurs candidats (Horvitz et Zilberstein, 2001; Beyer et Sendhoff, 2007a,b).

Plus formellement, dans notre algorithme de clustering en une seule passe, nous représentons la distance estimée entre le point actuel et chaque cluster i par une variable aléatoire X_i. Comme déjà indiqué, nous faisons par la suite l'hypothèse naïve que les X_i sont indépendantes. La distance entre la donnée et le cluster est supposée comprise entre deux bornes a et b, à partir desquelles il est possible de calculer une étendue (ou "range" en anglais) $R = |a - b|$. Nous supposons également que nous connaissons un seuil de confiance noté $1 - p$, et où $p \in [0, 1]$ est la probabilité d'erreur. Nous proposons ci-après trois méthodes principales pour estimer les bornes de l'intervalle de confiance associé à chaque variable aléatoire. De plus, nous proposons de modifier l'expression théorique des bornes en introduisant un facteur de réduction $r \in [0, 1]$ pour les rendre plus strictes au besoin. Il est intéressant de noter que si $r = 1$, on retrouve l'expression exacte des bornes théoriques.

La première borne est la borne de Hoeffding (Maron et Moore, 1994) :

$$\left|\widehat{X_{i,n}} - E(X_i)\right| \leq r \times R\sqrt{\frac{log\frac{2}{p}}{2n}}$$

où $\widehat{X_{i,n}}$ représente la moyenne empirique des distances au cluster X_i après n comparaisons et est définie comme suit :

$$\widehat{X_{i,n}} = \frac{1}{n}\sum_{n'=1}^{n} X_{i,n'}$$

, où $E(X_i)$ désigne la distance réelle au cluster X_i, et l'erreur de probabilité p indique les chances que la distance réelle soit hors des bornes.

La seconde borne est la borne plus récente de Bernstein (Heidrich-Meisner et Igel, 2009) qui repose sur l'écart-type empirique $\widehat{\sigma_{i,n}^2} = \frac{1}{n}\sum_{n'=1}^{n}(X_{i,n'} - \widehat{X_{i,n}})^2$ comme suit :

$$\left|\widehat{X_{i,n}} - E(X_i)\right| \leq r \times \widehat{\sigma_{i,n}}\sqrt{\frac{2log\frac{3}{p}}{n} + \frac{3Rlog\frac{3}{p}}{n}}$$

Bien que la borne de Bernstein soit connue pour être plus stricte que la borne de Hoeffding (Audibert et al., 2007; Mnih et al., 2008) et doit donc conduire à accélérer davantage le processus de compétition entre clusters, nous proposons de comparer le comportement des deux sur nos jeux de test.

Cependant, comme le montre les équations précédentes, une des limitations potentielles des bornes de Hoeffding et Bernstein est que l'étendue R est un paramètre nécessaire au calcul des valeurs de ces bornes. Même si les espaces de description des données sont souvent bornés, ce qui permet de déduire la valeur du paramètre R, dans de nombreux cas il n'est pas possible de connaître cette valeur a priori, ou il est peut-être trop complexe de le calculer exhaustivement, ce qui ferait perdre le gain de notre approche par ailleurs. Pour toutes ces raisons et également pour accélérer les calculs en resserrant la borne en la contraignant plus, nous proposons d'évaluer également la borne de Student qui est indépendante de l'étendue des données, et qui fait l'hypothèse que les distances aux clusters suivent une loi normale de variance inconnue.

$$\left|\widehat{X_{i,n}} - E(X_i)\right| \leq r \times t_{1-\frac{p}{2}}^{n-1} \sqrt{\frac{\widehat{S_{i,n}^2}}{n}}$$

où $\widehat{S_{i,n}^2} = \frac{1}{n-1}\sum_{n'=1}^{n}(X_{i,n'} - \widehat{X_{i,n}})^2$ désigne l'estimateur non biaisé de la variance de la distance au cluster X_i.

3.3 Implémentation de la méthode de compétition

Algorithme 1 Algorithme de clustering en une passe avec compétition (X, D, T)

Entrée : X : jeu de données, D : matrice de distances, T : seuil de distance pour la construction d'un nouveau cluster

Sortie : P : partition de sortie du jeu de données X

1: **initialiser** l'ensemble des clusters $C = \emptyset$
2: **Pour Tout** $x \in X$ **Faire**
3: **déterminer** le meilleur cluster c_w pour x en utilisant le mécanisme de compétition de l'algorithme 2
4: **affecter** x à c_w ssi $\widehat{X_{c_w}} \leq T$
5: **Fin Pour**
6: **Retourner** la partition calculée à partir de C

L'algorithme 1 détaille le schéma global de notre méthode de clustering : à chaque itération, un nouvel objet est considéré et les clusters existants sont mis en compétition par le biais de l'algorithme 2. La compétition permet de filtrer graduellement l'ensemble des clusters candidats, jusqu'à ce qu'il ne reste plus qu'un seul candidat, ou bien qu'il n'y ait plus de données pour affiner la prise de décision. En effet, à chaque fois que la borne inférieure sur la distance empirique à un cluster est plus grande que la borne supérieure du meilleur cluster actuel, il est supprimé de la compétition. À l'opposé, si la borne supérieure d'un cluster est plus petite que la borne inférieure du vainqueur actuel, celui-ci le remplace et devient le nouveau meilleur cluster. Enfin, lorsqu'il n'y a pas de différences singitificatives entre les clusters restants, le vainqueur final de la compétition est celui qui minimise sa distance empirique moyenne avec

la donnée. À l'issue de la compétition, l'objet est affecté au cluster vainqueur c_w si sa distance $\widehat{X_{c_w}}$ est inférieure à un seuil T passé en argument. Dans le cas contraire, la donnée construit un nouveau cluster.

Algorithme 2 Algorithme de compétition entre clusters (x, C, D)

Entrée : x : objet du jeu de données X, C : ensemble des clusters existants, D : matrice de distances
Sortie : c_w : indice du cluster qui remporte la compétition

initialiser le cluster vainqueur $c_w = \emptyset$
Tant Que la compétition n'est pas finie **Faire**
  Pour Tout clusters $c \in C$ **Faire**
    selectionner aléatoirement une nouvelle donnée x_c dans le cluster c
    mettre à jour la distance moyenne empirique pour le cluster c avec la distance $D(x, x_c)$ ainsi que les bornes $[inf_c, sup_c]$ pour le cluster c
    Si $sup_c < sup_{c_w}$ ou $c_w == \emptyset$ **Alors**
      mettre à jour le cluster vainqueur $c_w = c$
    Fin Si
  Fin Pour
  supprimer tous les clusters $c \in C$ encore en compétition tels que $inf_c > sup_{c_w}$
  Si $|C| < 2$ **Alors**
    Retourner c_w
  Fin Si
Fin Tant Que
Si la compétition se termine sans différence significative entre les clusters de C **Alors**
  Retourner le cluster $c \in C$ qui minimise la distance moyenne empirique
Fin Si

4 Résultats expérimentaux

Nous présentons ici les résultats comparatifs entre la méthode de détermination exhaustive du cluster le plus proche et notre méthode de compétition basée sur des bornes théoriques. Nous discutons ensuite l'influence des paramètres de la méthode sur son efficacité.

4.1 Protocole expérimental

Les résultats sont présentés pour différentes valeurs du facteur de réduction r comprises entre 0 et 1 (quand $r = 1$ on se trouve dans le cas de la borne théorique originale), pour une probabilité d'erreur fixée pour l'ensemble des tests à 0.1 et une valeur du range R exacte. La valeur du seuil qui détermine si un nouveau cluster doit être construit ou non est estimée, similairement à ce qui a été proposé pour l'algorithme Leader Ant (Labroche, 2006), comme la moyenne des distances calculée sur un échantillon aléatoire d'une taille égale à 10% de l'effectif total du jeu de données. Enfin, du fait du calcul de la partition de manière exhaustive, seulement 5 tests ont été réalisés pour chaque jeu de données et chaque valeur du facteur de réduction de borne r.

Données	$Art1$	$Art2$	$Art5$	$Art6$	$Statlog_{shuttle}$	$Letter_{recognition}$
n	20 000	20 000	45 000	40 000	14 500	20 000
n_{att}	2	2	2	8	8	16
k	4	2	9	4	7	26

TAB. 1 – *Caractéristiques principales des jeux de données artificielles et réelles issues du répartoire UCI Machine Learning Repository (Asuncion et Newman, 2007).*

Évaluation de la qualité des résultats : l'objectif de notre évaluation n'est pas de déterminer la qualité de la partition de manière classique (indice de Rand ...), car, dans notre cas, la référence est donnée par la qualité de la partition de la méthode exhaustive. Nos expérimentations visent donc à montrer l'impact de notre méthode selon deux dimensions principalement : l'accélération par la réduction du nombre de comparaisons et le nombre d'erreurs par rapport à la méthode exhaustive. Nous n'évaluons pas, dans ce travail, les temps de calcul des différentes méthodes pour mesurer l'accélération, car nous souhaitons nous affranchir des optimisations d'implémentation liées au langage choisi (Java) et du contexte d'exécution (bibliothèques liées).

De façon à pouvoir comparer identiquement au cours du temps nos méthodes basées sur les bornes, s'affranchir des éventuelles erreurs précédentes, et également déterminer à quel moment surviennent les erreurs d'affectation par rapport à la méthode exhaustive, nos expérimentations sont basées sur une mesure d'erreur avec correction de l'affectation à chaque nouvelle donnée traitée. Ainsi, pour chaque point des jeux de données, notre protocole détermine le cluster idéal à l'aide de la méthode exhaustive puis prédit le cluster qui serait choisi pour une affectation avec chacune des bornes. En cas de différence entre le cluster prédit et le cluster idéal, une erreur est comptabilisée pour la borne concernée. Dans tous les cas, le point est affecté au cluster idéal.

Jeux de données : du fait de l'utilisation de la méthode exhaustive qui a une complexité quadratique comme référence pour notre mesure d'erreur, nos tests ont été effectués sur des jeux de données d'une taille intermédiaire, permettant de finir le calcul en un temps raisonnable, tout en donnant une idée du comportement des méthodes sur de grands jeux de données (notamment l'accélération). De plus, de façon à varier les difficultés, les tests ont été conduits d'une part sur des données artificielles générées à l'aide de distributions de données normales avec un recouvrement plus ou moins important entre les groupes, et d'autre part sur des données réelles issues du UCI Machine Learning Repository (Asuncion et Newman, 2007). Le détail des jeux de données retenus est présenté dans le tableau 1 qui indique pour chacun son nombre d'objets (n), sa dimensionalité (nombre d'attributs n_{att}) et le nombre de clusters k attendus.

4.2 Résultats comparatifs

L'application de notre algorithme avec les trois bornes (Bernstein, Hoeffding et Student) sur les différents jeux de données produit les mêmes résultats avec différentes amplitudes

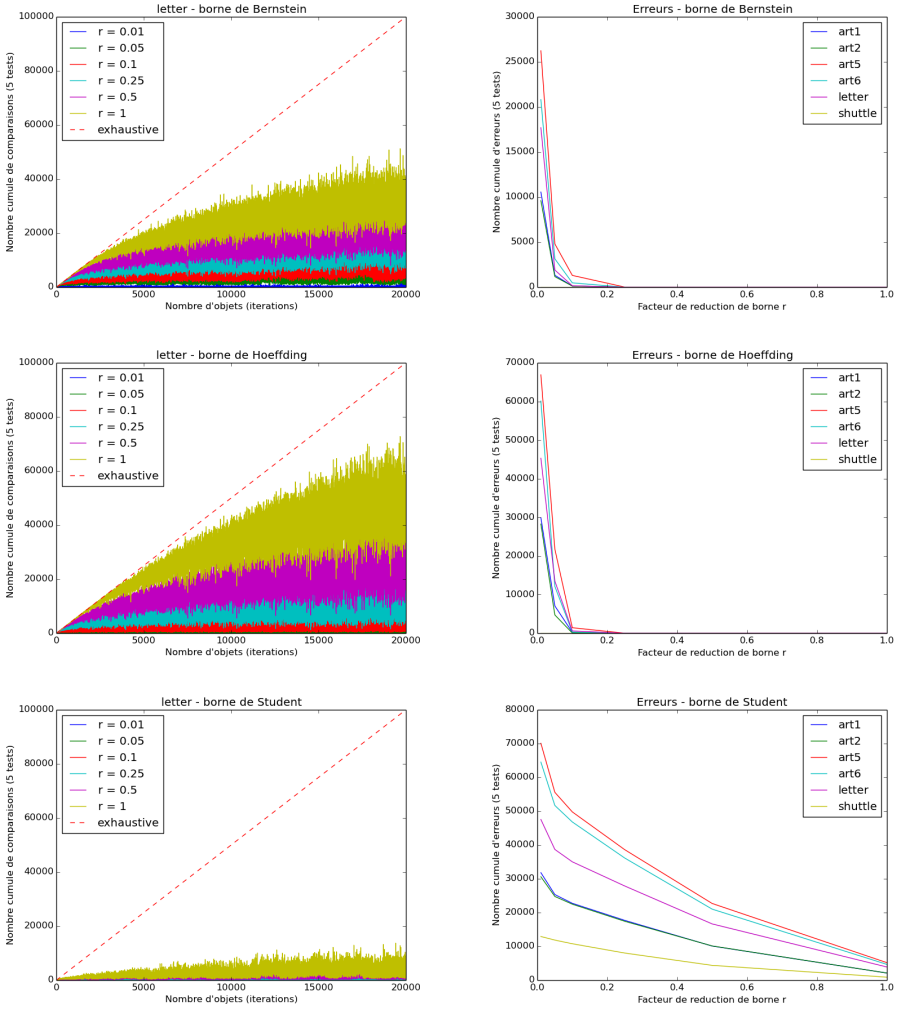

FIG. 1 – *Nombre cumulé (sur 5 tests) de comparaisons pour le jeu de données Letter (gauche) et d'erreurs d'affectation pour l'ensemble des jeux de données (droite) pour différentes valeurs du facteur de réduction* $r \in]0, 1]$.

comme le montre la figure 1. La borne de Bernstein permet d'obtenir dans tous les cas de meilleurs résultats que la borne de Hoeffding à la fois en terme de nombre de comparaisons et de nombre d'erreurs. Cela est probablement dû à l'utilisation de la variance empirique qui permet de resserer un peu plus la borne de Bernstein par rapport à celle de Hoeffding. La force de cette amélioration dépend du jeu de données mais l'exemple de la figure 1 est représentatif avec un nombre de comparaisons inférieur d'environ 30% pour Bernstein par rapport à Hoeffding.

La borne de Student est plus contrainte que les précédentes car elle fait l'hypothèse d'une distribution normale des données. Elle réalise ainsi beaucoup moins de comparaisons en général, mais au prix d'un nombre d'erreurs beaucoup plus élevé. En effet, sur nos jeux de données de tests, les bornes de Bernstein et Hoeffding, dès lors que le facteur de réduction $r \geq 0.25$ ne génèrent quasiment plus aucune erreur par rapport à la méthode exhaustive, alors que la borne de Student, même non réduite ($r = 1$) commet des erreurs (voir la colonne de droite de la figure 1).

Enfin, l'accélération augmente avec le nombre d'itérations. Plus le nombre de données déjà traitées augmente, meilleures sont les estimations, et donc plus les clusters candidats sont rapidement éliminés de la compétition, ce qui accélère le processus. Enfin, d'autres tests non rapportés dans cet article, suggèrent que le mécanisme d'accélération est également possible et bénéfique pour de petits jeux de données (comme Iris par exemple).

4.3 Discussion autour du paramétrage

L'accélération basée sur les bornes théoriques présentées dans la section 3 admet comme paramètre la probabilité d'erreur p. Du point de vue du problème de clustering, cette erreur p indique qu'il y a une probabilité non nulle qu'un mauvais cluster soit retenu. L'erreur d'affectation est donc supposée être inférieure à cette probabilité p. En pratique, comme le montre la figure 2-Haut, sur nos données de test nous observons que l'augmentation de la probabilité d'erreur réduit le nombre de comparaisons. En revanche, seule la borne de Student voit son erreur d'affectation augmenter, les bornes de Berstein et Hoeffding ne générant pas d'erreurs avec la borne originale ($r = 1$). Ce résultat doit encore être étudié, mais nous pensons pour le moment que cela est dû au fait que les bornes de Bernstein et Hoeffding sont lâches et se retrouvent par conséquent souvent dans un cas de décision ambigue entre plusieurs clusters candidats, l'affectation se faisant alors généralement au cluster correct sans garantie, mais après avoir éliminé plusieurs candidats.

Les bornes de Bernstein et Hoeffding nécessitent également de connaître à l'avance l'étendue (ou "range" R) des distances entre données. La figure 2-Bas illustre le comportement observé sur nos jeux de données à l'aide du jeu Art_1 lorsque l'étendue R est multiplié par des puissances de 2. On observe dans ce cas que la borne de Bernstein résiste mieux à une surestimation de l'étendue que la borne de Hoeffding, mais que dans tous les cas, il est toujours possible d'accélérer les calculs même avec une étendue 2 fois supérieure à ce qui était prévue initialement.

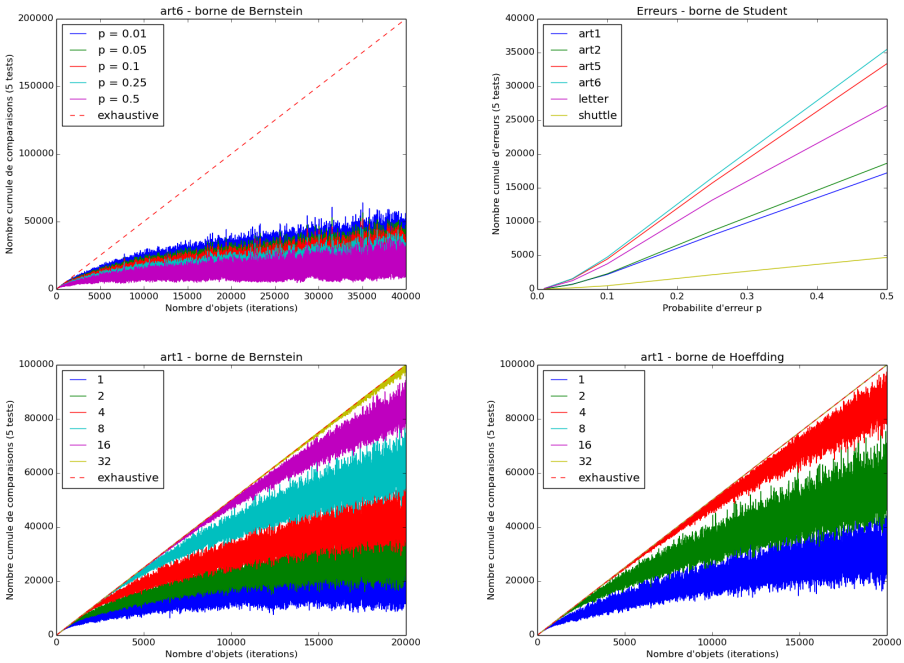

FIG. 2 – *Haut : analyse de l'influence de la probabilité d'erreur p. Haut-Gauche : exemple représentatif avec la borne de Bernstein pour le jeu de données Art_6, qui montre l'influence de la probabilité d'erreur sur le nombre de comparaisons. Haut-Droite : influence de la probabilité d'erreur p sur le nombre d'erreurs dans le cas de la borne de Student. Bas : influence de la sur-estimation de l'étendue des distances ("range") R en fonction des bornes de Bernstein (gauche) et Hoeffding (droite) pour le jeu de données Art_1.*

5 Conclusion et perspectives

Ce papier présente une nouvelle méthode de clustering en une passe pour des données non vectorielles, qui repose sur le principe des inégalités de concentration pour définir un mécanisme de compétition (ou "racing") qui estime la distance d'un nouveau point aux clusters tout en minimisant le nombre de comparaisons nécessaires. Trois bornes, Bernstein, Hoeffding et Student, sont comparées ainsi qu'une version réduite de chacune d'entre elles. Nos résultats montrent que notre algorithme permet de réduire drastiquement le nombre de comparaisons nécessaires par rapport à une méthode de clustering en une passe exhaustive en pratique, bien que les garanties théoriques des bornes ne puissent être assurées du fait de la possible dépendance des observations. Le facteur de réduction permet d'améliorer encore les résultats observés, notamment pour les bornes de Bernstein et Hoeffding qui sont, par construction, plus lâches que la borne de Student. Nous observons également que cette accélération augmente avec le nombre de données déjà classées, ce qui nous laisse à penser que notre méthode est particulièrement adaptée pour le traitement de grands jeux de données. En conclusion, d'un point de

vue général, la borne de Bernstein offre le meilleur compromis entre l'accélération (meilleur que Hoeffding) et le nombre d'erreurs (meilleur que Hoeffding et Student).

Dans le futur, de nouveaux tests doivent être conduits sans la correction des erreurs à partir de la "vérité terrain" de l'approche exhaustive, de façon à évaluer la qualité des partitions ainsi que l'impact des erreurs d'affectation sur les affectations suivantes. Ces tests pourront être conduits sur des données plus grandes, comme des données d'usage sur Internet qui sont produites continuellement et pour lesquelles il n'est pas possible d'utiliser les algorithmes incrémentaux classiques limités aux données numériques. D'autres modèles statistiques (U-statistics) et d'autres bornes (borne de Serfling) pourront également être envisagés pour obtenir des garanties théoriques.

A plus long terme on pourra s'intéresser à l'utilisation de l'information d'ambiguité, c'est-à-dire les cas où il n'est pas possible de différencier les clusters sur la base des bornes estimées, dans un contexte incrémental pour déterminer notamment : quand un cluster doit être créé, quand plusieurs clusters peuvent être fusionnés ou plus généralement pour identifier les points les moins importants d'un cluster dont l'importance peut en conséquence être réduite.

Références

Aggarwal, C. C., T. J. Watson, R. Ctr, J. Han, J. Wang, et P. S. Yu (2003). A framework for clustering evolving data streams. In *Proc. of VLDB*, pp. 81–92.

Asuncion, A. et D. Newman (2007). UCI machine learning repository – University of California, Irvine, School of Information and Computer Sciences. http ://www.ics.uci.edu/∼mlearn/MLRepository.html.

Audibert, J.-Y., R. Munos, et C. Szepesvari (2007). Tuning bandit algorithms in stochastic environments. In *Algorithmic Learning Theory*, Sendai, Japon, pp. 150–165.

Beyer, H.-G. et B. Sendhoff (2007a). Evolutionary algorithms in the presence of noise : To sample or not to sample. In *Proc. IEEE Symp. on Foundations of Computational Intelligence (FOCI)*.

Beyer, H.-G. et B. Sendhoff (2007b). Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering 196*(33-34), 3190–3218.

Cao, F., M. Ester, W. Qian, et A. Zhou (2006). Density-based clustering over an evolving data stream with noise. In *Proc. of SIAM Conference on Data Mining*, pp. 328–339.

Domingos, P., G. Hulten, P. C. W. Edu, et C. H. G. W. Edu (2001). A general method for scaling up machine learning algorithms and its application to clustering. In *Proc. of the 18th Int. Conf. on Machine Learning (ICML)*, pp. 106–113. Morgan Kaufmann.

Ester, M., H.-P. Kriegel, et X. Xu (1995). Knowledge discovery in large spatial databases : focusing techniques for efficient class identification. In *Int. Symposium on Advances in Spatial Databases*, Volume 951, Portland, ME, pp. 67–82. Springer.

Guha, S., R. Rastogi, et K. Shim (1998). Cure : An efficient clustering algorithm for large databases. In *Proc. of ACM SIGMOD Int. Conf. on management of Data*, pp. 73–84.

Hammer, B. et A. Hasenfuss (2007). Relational neural gas. In *Proc. of the 30th Annual German Conf. on Advances in Artificial Intelligence*, KI '07, pp. 190–204. Springer-Verlag.

Heidrich-Meisner, V. et C. Igel (2009). Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *Proc. Int. Conf. on Machine Learning (ICML)*.

Horvitz, E. et S. Zilberstein (2001). Computational tradeoffs under bounded resources. *Artificial Intelligence - Special Issue on Computational Tradeoffs under Bounded Resources 126*(1-2), 1–4.

Kaufman, L. et P. Rousseeuw (1990). Finding groups in data : An introduction to cluster analysis. In *John Wiley and Sons*.

Krishnapuram, R., A. Joshi, et L. Yi (1999). A fuzzy relative of the k-medoids algorithm with application to document and snippet clustering. In *Proc. of FUZZIEEE 99*, Korea.

Labroche, N. (2006). Fast ant-inspired clustering algorithm for web usage mining. In *Proc. IPMU 2006 Conference*, Paris, France, pp. 2668–2675.

Maron, O. et A. Moore (1994). Hoeffding races : Accelerating model selection search for classification and function approximation. In *Proc. Advances in Neural Inf. Proc. Systems*.

Mnih, V., C. Szepesvári, et J.-Y. Audibert (2008). Empirical bernstein stopping. In *Proceedings of the 25th Int. Conf. on Machine learning*, Proc. ICML '08, pp. 672–679. ACM.

Philipp Kranen, Ira Assent, C. B. et T. Seidl (2011). The clustree : indexing micro-clusters for anytime stream mining. *Knowledge and Information Systems 29*(2), 249–272.

Zhang, T., R. Ramakrishnan, et M. Livny (1996). BIRCH : an efficient data clustering method for very large databases. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, pp. 103–114.

Zhu, X., A. Gisbrecht, F.-M. Schleif, et B. Hammer (2012). Approximation techniques for clustering dissimilarity data. *Neurocomput. 90*, 72–84.

Summary

Single-pass incremental clustering relies on the efficient assignment of each new data point to one of the existing clusters. In the general case, where it is not necessarily possible to represent the clusters by a mean, the exhaustive assignment of a point a cluster has a quadratic complexity in term of the number of data objects. This paper proposes a novel stochastic assignment method that minimizes the number of comparisons between the new data and each cluster to guaranty, given an acceptable error rate, that the point is assigned to its nearest cluster. Several theoretical bounds are considered (Bernstein, Hoeffding and Student) and compared in this paper. Results observed on artificial and real data sets show that Berntein bound give the overall best results (especially when it is reduced) as it provides the best acceleration of the clustering while maintaining a very low number of errors.