# Détection de changements dans des flots de données qualitatives

Dino Ienco[*,***], Albert Bifet[**],
Bernhard Pfahringer[***], Pascal Poncelet[*]

[*]Irstea, UMR TETIS, Montpellier, France
LIRMM, Montpellier, France
{dino.ienco@irstea.fr, pascal.poncelet@lirmm.fr}

[**]Yahoo! Research Barcelona, Catalonia, Spain
abifet@yahoo-inc.com

[***]University of Waikato, Hamilton, New Zealand
bernhard@cs.waikato.ac.nz

**Résumé.** Pour mieux analyser et extraire de la connaissance de flots de données, des approches spécifiques ont été proposées ces dernières années. L'un des challenges auquel elles doivent faire face est la détection de changement dans les données. Alors que de plus en plus de données qualitatives sont générées, peu de travaux de recherche se sont intéressés à la détection de changement dans ce contexte et les travaux existants se sont principalement focalisés sur la qualité d'un modèle appris plutôt qu'au réel changement dans les données. Dans cet article nous proposons une nouvelle méthode de détection de changement non supervisée, appelée *CDCStream* (*Change Detection in Categorical Data Streams*), adaptée aux flux de données qualitatives.

## 1 Introduction

De nombreux domaines d'application génèrent en permanence d'énormes quantités de données. L'un des défis essentiels auquel les approches de fouilles de flots doivent faire face est la détection de changement dans ces données. En effet, l'information disponible dans les flots change et évolue au fil du temps et les connaissances acquises au préalable peuvent s'avérer non représentatives des nouvelles données. Dans un contexte d'apprentissage, cela se traduit par le fait que des classes ou des concepts sous représentés (resp. surreprésentés) peuvent apparaître surreprésentés (resp. sous représentés) après une période plus longue. Savoir détecter le plus tôt possible les réels changements dans le flot permet alors de pouvoir réévaluer automatiquement les apprentissages précédents et surtout garantir que la connaissance extraite à un moment donné est vraiment représentative des données disponibles sur le flot. Dans cet article, nous proposons une nouvelle méthode de détection de changement définie pour traiter des donnés qualitatives : *CDCStream* (*Change Detection in Categorical data Streams*). L'une

des originalité de notre approche est de pouvoir mettre en évidence de manière non supervisée les changements dans le flot des données en exploitant efficacement une partie de l'historique. Le modèle retenu de description du flot est sous la forme de lots de données : lorsque qu'un nouveau lot arrive, *CDCStream* construit un résumé informatif et calcule différents tests statistiques afin de vérifier si un changement a eu lieu dans la distribution des données.

Le reste de cet article est organisé de la manière suivante. *CDCStream* est décrit dans la section 2 et des expérimentations menées dans la section 3. Nous concluons dans la section 4.

## 2 L'approche *CDCStream*

Nous considérons une représentation classique de flots de données sous la forme d'un flot infini divisé en lots : $S = \{S_1, S_2, ..., S_n, ...\}$. En outre, nous considérons que chaque exemple appartenant au flot est défini sur un ensemble d'attributs qualitatifs, i.e. chaque attribut $X_j$ est défini sur un ensemble discret de valeurs nominales. Tout d'abord nous résumons les données qualitatives à l'aide de la méthode *DILCA* proposée dans (Ienco et al., 2012) afin de résumer la distribution des données sous-jacente. Nous proposons ensuite de surveiller les statistiques extraites des lots en utilisant *l'inégalité de Chebyshev* (Aggarwal, 2007).

Le principe de *DILCA* est de regrouper des données qualitatives entre elles via un ensemble de matrices, une pour chaque attribut, où chaque matrice contient les distances apprises entre chaque paire de valeurs d'un attribut spécifique. Dans notre contexte, considérons l'ensemble $F = \{X_1, X_2, \ldots, X_m\}$ de $m$ attributs qualitatifs pour le lot $S_i$. $|X_i|$ désigne la cardinalité de l'attribut $X_i$. On note $Y$ l'attribut cible, un attribut spécifique dans $F$ pour lequel nous devons calculer les différentes distances. *DILCA* calcule une distance basée sur le contexte entre n'importe quelle paire de valeurs $(y_i, y_j)$ de l'attribut cible $Y$ sur la base de la similitude entre les distributions de probabilité des $y_i$ et $y_j$, compte tenu des attributs de contexte, appelée $\mathcal{C}(Y) \subseteq F \setminus Y$. Pour chaque attribut de contexte $X_i$ il calcule la probabilité conditionnelle pour les deux valeurs $y_i$ et $y_j$ étant données les valeurs $x_k \in X_i$, puis il applique la distance euclidienne. La distance euclidienne est normalisée par le nombre total de valeurs considérées :

$$d(y_i, y_j) = \sqrt{\frac{\sum_{X \in \mathcal{C}(Y)} \sum_{x_k \in X} (P(y_i|x_k) - P(y_j|x_k))^2}{\sum_{X \in \mathcal{C}(Y)} |X|}} \tag{1}$$

Pour sélectionner un ensemble pertinent et non redondant de caractéristiques, les auteurs de (Ienco et al., 2012) proposent d'adopter *FCBF*, une approche de sélection d'attributs initialement proposée par (Yu et Liu, 2003). A la fin du processus, *DILCA* renvoie un modèle de distance $\mathcal{M} = \{M_{X_l} \mid l = 1, \ldots, m\}$, où chaque $M_{X_l}$ est la matrice contenant les distances entre toutes les paires de valeurs d'attribut $X_l$, calculées en utilisant l'équation 1. Chaque matrice générée $M_{X_l}$ présente des caractéristiques intéressantes : elle est symétrique, la diagonale contient forcément des 0 et chaque valeur est délimitée entre 0 et 1. En fait, cet ensemble de matrices constitue une information utile pour résumer la distribution sous-jacente. L'ensemble $\mathcal{M}$ de matrices peuvent alors être agrégées en une seule mesure par la formule suivante, qui ne tient compte que de la partie triangulaire supérieure de chaque matrice :

$$resumeLot(M) = \frac{\sum_{M_l \in M} \frac{2 \times \sqrt{\sum_{i=0}^{|X_l|} \sum_{j=i+1}^{|X_l|} M_{X_l}(i,j)^2}}{|X_l| * (|X_l| - 1)}}{|F|} \tag{2}$$

Cette formule correspond à un résumé de l'ensemble du lot en tenant compte à la fois de la corrélation entre les attributs et de la distribution des valeurs des attributs.

Pour déterminer si une distribution d'un lot s'écarte ou non d'une distribution initiale, nous utilisons des techniques de contrôles statistiques des processus (Gama et al., 2004) fondées sur l'inégalité de Chebyshev (Aggarwal, 2007). Ce choix ne nécessite aucune hypothèse sur les distributions des données et permet de montrer qu'une variable aléatoire prendra, avec une grande probabilité, une valeur relativement proche de son espérance :

**Definition 1 (Inégalité de Chebyshev)** *Soit X une variable aléatoire avec espérance $\mu_X$ et écart-type $\sigma_X$. Alors, pour tout k $\in R^+$,*

$$Pr(|X - \mu_X| \geq k\sigma_X) \leq \frac{1}{k^2} \tag{3}$$

L'inégalité nécessite de calculer la moyenne et l'écart-type et il n'est pas possible de stocker l'intégralité de ces valeurs. Nous proposons donc d'évaluer les changements à la volée et de ne remonter que les véritables modifications dans la distribution des données (en utilisant un mécanisme d'oubli) ou de ne faire remonter que les simples fluctuations sous forme d'alarmes.

# 3 Expérimentations

Nous comparons les résultats et le comportement de notre algorithme, *CDCS.*, avec la méthode $RSCD$ proposée dans (Cao et Huang, 2013) et reconnue comme la méthode la plus efficace pour détecter des changements dans des flots de données qualitatives. Nous comparons également notre approche avec la méthode supervisée, $SDriftC$, de (Gama et al., 2004) qui possède un mécanisme de détection de changement et de variation en fonction de la précision du classifieur. Les expérimentations menées avec les jeux de données suivants : *Electricity*, *Forest*, *Airlines*, *KDD99* ont été réalisées avec la plateforme MOA (Bifet et al., 2010). Dans le tableau 1, nous pouvons constater que l'approche supervisée n'est pas meilleure que les approches non supervisées. Ceci est un résultat intéressant dans la mesure où $SDriftC$ est la seule méthode qui exploite la précision afin de prendre une "décision de détection de changement" pour ré-apprendre le modèle sous jacent. Nous pouvons également constater que, la plupart du temps, le classifieur appris avec notre méthode obtient de bien meilleures performances que les autres approches non supervisées. Nous présentons également le comportement de *CDCStream* et $RSCD$ dans le tableau 2 où nous mesurons le pourcentage des changements détectés par les deux méthodes pour différentes tailles de lots. Nous pouvons constater que les deux méthodes ont des comportements très différents. Par exemple, si nous analysons les résultats de *Electricity*, pour une taille de lot de 50 le nombre de changements est similaire. Par contre lorsque les lots grandissent les tendances sont très différentes. En particulier, la taille du lot pour $b = 500$ et $b = 1\,000$ a un impact fort sur le pourcentage de changements découverts par $RSCD$.

# 4 Conclusion

Détecter les changements dans un flux de données qualitatives n'est pas simple. Dans cet article, nous avons présenté un nouvel algorithme qui extrait des résumés des différents lots et

| **Dataset** | | b=50 | | b=100 | | b=500 | | b=1 000 | |
|---|---|---|---|---|---|---|---|---|---|
| | SDriftC | CDCS. | $RSCD$ | CDCS. | $RSCD$ | CDCS. | $RSCD$ | CDCS. | $RSCD$ |
| Electricity | 70.58% | **73.84**% | 73.77% | 71.23% | 72.19% | 68.18% | 64.45% | 66.01% | 63.40% |
| KDD | **91.38**% | 90.81% | 90.65% | 90.04% | 90.06% | 90.59% | 90.06% | 90.87% | 90.06% |
| Forest | 80.27% | **82.99**% | 82.25% | 81.18% | 80.39% | 74.46% | 64.08% | 80.05% | 64.08% |
| Forest Sort. | 67.44% | 70.57% | 68.35% | 70.51% | 68.35% | **72.10**% | 68.35% | 69.33% | 68.35% |
| Airlines | 65.25% | 66.43% | 62.71% | 64.50% | 64.41% | 67.16% | 66.73% | **67.66**% | 67.64% |

TAB. 1 – *La précision des différents modèles appris dans le cadre d'une stratégie variation/changement avec différentes tailles de lots (paramètre b)*

| **Jeux de données** | b=50 | | b=100 | | b=500 | | b=1 000 | |
|---|---|---|---|---|---|---|---|---|
| | CDCS. | $RSCD$ | CDCS. | $RSCD$ | CDCS. | $RSCD$ | CDCS. | $RSCD$ |
| Electricity | 96.57% | 92.93% | 64.67% | 100% | 55.55% | 6.66% | 44.44% | 8.88% |
| KDD | 7.97% | 1.34% | 1.01% | 0.06% | 85.52% | 0.33% | 75.67% | 0% |
| Forest | 96.3% | 77.63% | 34.49% | 71.15% | 0% | 0.17% | 18.24% | 0.34% |
| Forest Sort. | 12.65% | 0% | 13.87% | 0% | 96.21% | 0.08% | 8.26% | 0.17% |
| Airlines | 14.68% | 100% | 94.12% | 100% | 2.78% | 100% | 58.07% | 44.52% |

TAB. 2 – *Changement de comportement : nombre de changements déclenchés sur le nombre de changements possibles (nombre de lots - 1)*

utilise l'inégalité de Chebyshev pour mettre en évidence des changements de distribution mais également des variations dans les données.

# Références

Aggarwal, C. C. (2007). *Data Streams - Models and Algorithms*. Advances in Database Systems. Springer.

Bifet, A., G. Holmes, R. Kirkby, et B. Pfahringer (2010). MOA : Massive Online Analysis. *J. Mach. Learn. Res. 11*(May), 1601–1604.

Cao, F. et J. Z. Huang (2013). A concept-drifting detection algorithm for categorical evolving data. In *PAKDD, LNAI 7819*, pp. 485–496.

Gama, J., P. Medas, G. Castillo, et P. Rodrigues (2004). Learning with drift detection. In *SBIA, LNAI 3171*, pp. 286–295.

Ienco, D., R. G. Pensa, et R. Meo (2012). From context to distance : Learning dissimilarity for categorical data clustering. *ACM TKDD 6*(1), 1 :1–1 :25.

Yu, L. et H. Liu (2003). Feature selection for high-dimensional data : A fast correlation-based filter solution. In *ICML*.

# Summary

In real world applications, data streams have categorical features, and changes induced in the data distribution of these categorical features have not been considered extensively so far. Previous work focused on detecting changes in the accuracy of the learners, but without considering changes in the data distribution. To cope with these issues, we propose a new unsupervised change detection method well suited for categorical data streams.