Détection de changements de distribution dans un flux de données : une approche supervisée

Alexis Bondu*, Marc Boullé**

*EDF R&D ICAME/SOAD, 1 avenue du Général de Gaulle, 92140 Clamart.
alexis.bondu@edf.fr
**Orange Labs, 2 avenue Pierre Marzin, 22300 Lannion.
marc.boulle@orange-ftgroup.com

Résumé. L'analyse de flux de données traite des données massives grâce à des algorithmes en ligne qui évitent le stockage exhaustif des données. La détection de changements dans la distribution d'un flux est une question importante dont les applications potentielles sont nombreuses. Dans cet article, la détection de changement est transposée en un problème d'apprentissage supervisé. Nous avons choisi d'utiliser la méthode de discrétisation supervisée MODL car celle-ci présente des propriétés intéressantes. Notre approche est comparée favorablement à une méthode de l'état-de-l'art sur des flux de données artificiels.

1 Introduction

Ces dernières années, la quantité de données à traiter a considérablement augmenté dans de nombreux domaines applicatifs. Les méthodes d'apprentissage "classiques" passent difficillement à l'échelle sur de tels volumes de données. L'analyse de flux de données (ou *"Data stream mining"*) est une des réponses possibles au traitement des données massives. Le paradigme des "flux de données" prend en compte plusieurs contraintes : i) l'ordre d'arrivé des données (ou tuples [1]) n'est pas contrôlé ; les débits des flux d'entrée sont fluctuants et ne sont pas contrôlés ; les ressources matérielles disponibles sont limitées (mémoire RAM et CPU). Du fait de ces contraintes, les tuples ne peuvent pas être stockés exhaustivement et sont analysés à la volée. Une analyse sur flux de données fournit un résultat en "temps réel" qui évolue continuellement. L'objectif est de maximiser la qualité de l'analyse, compte tenu des contraintes imposées par les flux, et étant donnés les ressources matérielles disponibles. La mise au point de méthodes de détection de changements de distribution qui soient génériques, capables de passer à l'échelle, et pertinentes d'un point de vue statistique est un véritable challenge. La détection de changements dans un flux de données consiste à comparer la distribution des tuples observés sur deux fenêtres temporelles distinctes : la *"référence"* et la *"courante"*. Dans le cadre de cet article, nous traitons le cas de la détection relative à un régime de fonctionnement normal. Dans la littérature, il existe plusieurs manières d'aborder le problème de la détection de changements. Par exemple, un test statistique impliquant deux échantillons de tuples (Dries et Rückert, 2009) peut être utilisé. Le test de *"Wald-Wolfowitz et Smirnov"* a notamment été généralisé au cas des données multidimensionnelles (Friedman et Rafsky, 2006), ce qui est

1. Les données élémentaires émises dans un flux sont appelées des *"tuples"*

pertinent dans le cas du traitement de flux de données. D'autres approches basées sur la méthode des plus proches voisins (Hall, 2002), ou encore, sur des distances entre distributions (Bondu et al., 2010) ont été développées. Enfin, des travaux sur la détection de rupture ont été réalisés (Desobry et Davy, 2003). Dans cet article, la Section 2 propose une approche originale qui transpose la détection de changement en un problème d'apprentissage supervisé. Notre approche est comparée favorablement à une méthode de l'état-de-l'art lors d'une validation expérimentale menée à la Section 3. Enfin, les perspectives de nos futurs travaux sont discutées à la Section 4.

2 Une approche supervisée pour la détection de changement

Comme le montre l'étape 1 de la Figure 1, l'approche proposée dans cet article exploite deux fenêtres temporelles [2] définies a priori par un expert : i) la fenêtre de *référence* représente le fonctionnement normal du système observé ; ii) la fenêtre *courante* caractérise l'état actuel du système. La fenêtre de référence est **fixe** et la fenêtre courante est **glissante**. La définition de ces deux fenêtres constitue le seul paramétrage demandé à l'utilisateur. Les tuples sont ensuite étiquetés (*cf* étape 2 Figure 1). Les tuples appartenant à la fenêtre de référence [*respectivement* à la fenêtre courante] sont estampillés comme appartenant à la classe "0" [*respectivement* la classe "+"]. La qualité du classifieur caractérise le niveau de changement dans la distribution des tuples. Le contexte des flux de données impose de traiter les tuples le plus rapidement possible, c'est pourquoi nous avons choisi de simplifier le problème de classification initial en considérant K problèmes de classification univariés. Les tuples sont projetés sur chacune des variables (*cf* étape 3 Figure 1) et plusieurs classifieurs sont entraînés en parallèle.

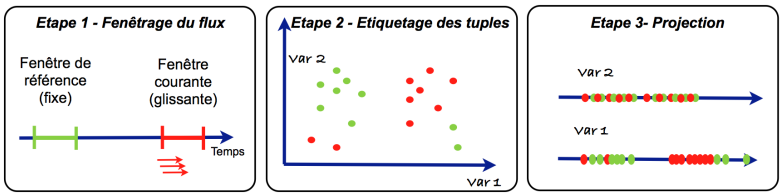

FIG. 1 – *Détection de changement posé comme un problème de classification supervisée.*

Le choix de la méthode de classification est critique dans notre démarche et doit répondre aux critères suivants :

- ✓ le classifieur doit estimer les densités conditionnelles des classes ;
- ✓ la méthode ne doit pas demander de connaissances a priori ;
- ✓ la méthode ne doit pas impliquer de paramètres utilisateur ;
- ✓ le classifieur doit être peu sensible aux valeurs atypiques ;
- ✓ la méthode doit être régularisée pour éviter le sur-apprentissage ;

Etant donnés ces critères, nous avons choisi d'utiliser l'approche de discrétisation supervisée MODL (Boullé, 2006) qui s'est distinguée lors de challenges (Guyon et al., 2006).

2. Une fenêtre temporelle est définie par une "date de début" et une "date de fin", elle contient les tuples émis dans cet intervalle de temps.

2.1 Détection et diagnostic

Le critère d'évaluation d'une discrétisation $\mathcal{C}_k(M) = -\log[P(M|D_k)]$ correspondant à la probabilité qu'un modèle de discrétisation M explique le type de régime du flux *(référence ou courant)* connaissant les données D_k, caractérisées par la variable k. On note "$\mathcal{M}ap$" le modèle qui minimise ce critère. Soit $M0$ le modèle qui discrétise la variable k en un seul intervalle. Le gain de compression (Boullé, 2009) est défini selon la formule : $\mathcal{G}ain_k(M) = 1 - C_k(M)/C_k(M0)$. Notre approche exploite $\mathcal{G}ain_k(\mathcal{M}ap)$, qui vaut 0 quand la variable k ne permet pas de discriminer le type de régime, et qui est strictement positive quand il existe une différence de distribution significative de k selon le type de régime. $\mathcal{G}ain_k(\mathcal{M}ap)$ vaut 1 quand la variable k permet de discriminer parfaitement les deux types de régime.

quantification du changement de distribution : Le critère proposé ici a pour objectif de quantifier de manière globale le changement de distribution des tuples sur les deux fenêtres temporelles. Ce changement est caractérisé indépendamment sur chaque variable $k \in K$ par $\mathcal{G}ain_k(\mathcal{M}ap)$. Il convient alors de définir un critère agrégeant ces indicateurs de changement :

$$\mathcal{C}hange = \frac{1}{K} \sum_{k \in [1,K]} \mathcal{G}ain_k(\mathcal{M}ap) \quad (1)$$

contribution de chaque variable : La somme des contributions $\mathcal{C}ontrib(k)$ sur l'ensemble des variables explicatives correspond au changement évalué par le critère $\mathcal{C}hange$. La contribution de la variable k est définie par :

$$\mathcal{C}ontrib(k) = \frac{\mathcal{G}ain_k(\mathcal{M}ap)}{K}$$

3 Validation expérimentale

Lors de cette validation expérimentale, nous utilisons deux flux de données artificiels qui partagent la même structure temporelle. Les tuples sont émis chaque seconde selon une distribution sous-jacente qui varie au cours du temps (voir Figure 2). Les distributions *"initiale"* et *"modifiée"* sont définies par le Tableau 1 pour les deux flux de données artificiels. Il s'agit de distributions normales notées $\mathcal{N}(m, v)$, où m est un vecteur à deux dimensions caractérisant la moyenne et v est la matrice de covariance.

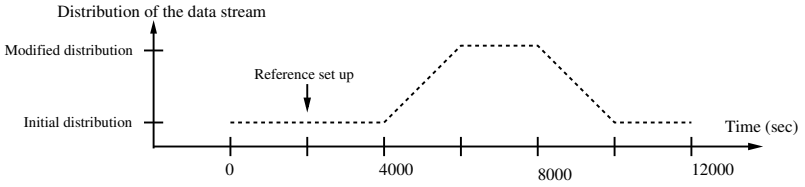

FIG. 2 – *Structure temporelle des deux flux de données artificiels.*

	distribution initiale	**distribution modifiée**
Flux 1 : changement de moyenne	$\mathcal{N}\left(\begin{array}{cc} 0 & 0 \end{array}, \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right)$	$\mathcal{N}\left(\begin{array}{cc} 4 & 8 \end{array}, \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right)$
Flux 2 : changement d'écart type	$\mathcal{N}\left(\begin{array}{cc} 0 & 0 \end{array}, \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right)$	$\mathcal{N}\left(\begin{array}{cc} 0 & 0 \end{array}, \begin{array}{cc} 4 & 0 \\ 0 & 9 \end{array}\right)$

TAB. 1 – *Définition des distributions "initiale" et "modifiée" pour les deux flux de données.*

3.1 Protocole expérimental

Le fenêtrage du flux d'entrée constitue le seul paramètrage requis par notre approche de détection de changements. Dans le cadre de nos expérimentations, la fenêtre de référence est fixe et comporte les 2000 premiers tuples du flux. L'ensemble de ces tuples est supposé être représentatif du fonctionnement normal du système observé. La fenêtre de référence est glissante [3] et comporte, à chaque instant, les 300 derniers tupes émis dans le flux.

3.2 Méthode concurrente

Notre stratégie de détection est comparée à une méthode de l'état-de-l'art basée sur l'estimation en ligne de la distribution des tuples (Bondu et al., 2010). Cette méthode met en jeu quatre étapes successives :

1 le résumé du flux par la méthode de micro-clustering "DenStream" qui exploite une pondération temporelle des tuples (Feng Cao et al., 2006) ;

2 l'estimation de la distribution courante par une version des fenêtres de Parzen adaptée aux micro-cluster ;

3 la comparaison de la distribution courante à une distribution de référence grâce à la divergence de Kullback-Leibler ;

4 l'estimation de la contribution de chaque variable grâce à un critère exploitant la divergence de Kullback-Leibler dans un sous-espace.

3.3 Résultat comparatifs

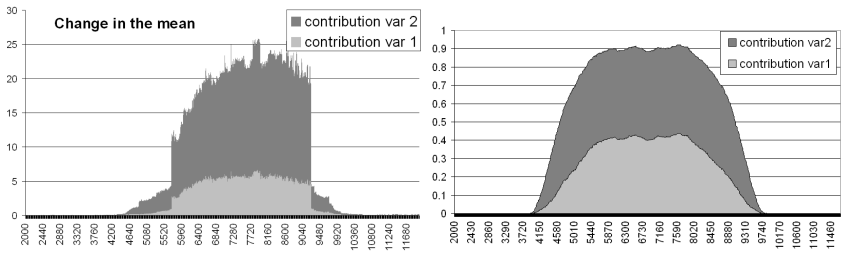

FIG. 3 – *Résultats comparatifs sur le 1er flux, où un changement de moyenne apparaît*

Les deux approches de détection sont évaluées selon trois critères : i) la précocité de la détection ; ii) la netteté de la détection ; iii) la capacité à évaluer les contributions. La Figure 3 présente les résultats obtenus sur le premier flux de données, où un changement de moyenne apparaît. Le graphique de gauche correspond à la méthode concurrente et le graphique de droite à notre approche. Dans les deux cas, l'axe horizontal représente le nombre de tuples émis depuis le début de l'expérience. L'axe vertical caractérise le niveau du changement détecté, soit grâce à la divergence de Kullback-Leibler (à gauche), soit par le critère $\mathcal{C}hange$ donné par

3. Afin de réduire le temps de calcul de nos expériences, nous avons choisi d'effectuer un point de mesure tous les dix tuples.

l'Equation 1 (à droite). Sur chaque graphique, la somme des contributions correspond au niveau global de détection. Comme le montre la Figure 3, notre approche commence à détecter le changement de distribution de manière très précoce (après 4100 tuples contre 4500 pour la méthode concurrente). Notre approche domine également la méthode concurrente du point de vue de la netteté de la détection, les courbes sont beaucoup moins bruitées. Dans le cas de notre approche, les contributions semblent être correctes uniquement pendant les régimes transitoires. Entre 6000 et 8000 tuples les deux contributions ont sensiblement la même valeur et forment un plateau. Cela s'explique par le fait que les deux classes sont parfaitement discriminées sur chacune des variables. Les deux variables apportent des quantités d'information équivalentes vis-à-vis de la discrimination des classes.

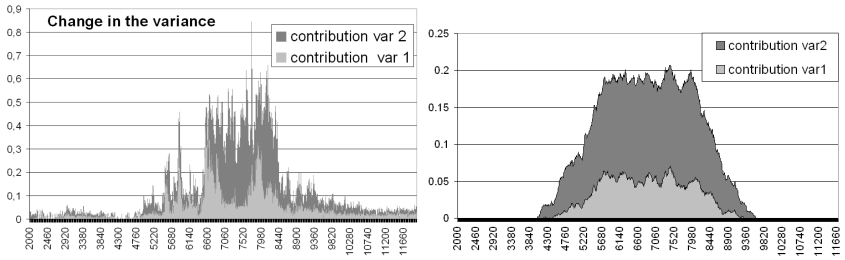

FIG. 4 – *Résultats comparatifs sur le 2ème flux, où un changement de variance apparaît*

De la même manière que précédemment, la Figure 4 présente les résultats obtenus par les deux méthodes de détection sur le 2ème flux de données, où un changement de variance apparaît. Encore une fois, notre approche détecte le changement de distribution très précocement (4050 tuples contre 4800 pour la méthode concurrente). Notre approche domine la méthode concurrente du point de vue de la détection, qui est beaucoup plus nette. L'effet de seuil souligné précédemment lors du calcul des contributions n'apparaît pas ici. Cela s'explique par le fait que les classes sont moins faciles à discriminer dans le cas de ce flux de données.

Pour conclure, les expériences comparatives présentées dans cette section montrent que notre approche domine la méthode de l'état-de-l'art. Notre approche est pourtant plus simple à mettre en oeuvre car elle n'implique aucun ajustement de paramètres utilisateur.

4 Conclusion et perspectives

Cet article propose de transposer la détection de changement dans la distribution d'un flux de données en un problème d'apprentissage supervisé. Deux fenêtres temporelles qui correspondent chacune à une classe sont définies sur le flux : i) la fenêtre de référence représente le fonctionnement normal du système observé ; ii) la fenêtre courante représente l'état actuel du système. La qualité du classifeur entrainé sur ces données caractérise le niveau de changement observé entre les deux fenêtres temporelles. Nous choisissons d'utiliser la méthode de discrétisation supervisée MODL (Boullé, 2006) en raison de ses bonnes propriétés (*cf* Section 2). Notre approche est comparée favorablement à une méthode de l'état-de-l'art (Bondu et al., 2010) sur deux flux de données artificiels. Il apparaît que notre approche détecte les changements de distribution plus tôt et que la détection est plus nette. Lors de travaux futurs, les contraintes matérielles (RAM et CPU) et applicatives (délais de détection) pourraient être

prises en compte explicitement par notre approche. L'objectif serait d'optimiser automatiquement la taille de la fenêtre de référence et le pas de calcul des indicateurs de changement et de diagnostic. Enfin, une version incrémentale de l'algorithme utilisé pour l'optimisation du critère $\mathcal{C}(M)$ pourrait être proposée.

Références

Bondu, A., B. Grossin, et M. Picard (2010). Density estimation on data streams : an application to Change Detection. In *EGC (Extraction et Gestion de l'Information)*.

Boullé, M. (2006). MODL: A bayes optimal discretization method for continuous attributes. *Machine Learning 65*(1), 131–165.

Boullé, M. (2009). Optimum simultaneous discretization with data grid models in supervised classification: a bayesian model selection approach. *Advances in Data Analysis and Classification 3*(1), 39–61.

Desobry, F. et M. Davy (2003). Support Vector-Based Online Detection of Abrupt Changes. In *Proc. IEEE ICASSP, Hong Kong*, pp. 872–875.

Dries, A. et U. Rückert (2009). Adaptive Concept Drift Detection. In *SIAM Conference on Data Mining*, pp. 233–244.

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

Friedman, J. et L. Rafsky (2006). Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *Annals of Statistic 7*(4), 697–717.

Guyon, I., A. Saffari, G. Dror, et J. Bumann (2006). Performance prediction challenge. In *International Joint Conference on Neural Networks*, pp. 2958–2965. http://www.modelselect.inf.ethz.ch/index.php.

Hall, P. (2002). Permutation tests for equality of distributions in high-dimensional settings. *Biometrika 89*(2), 359–374.

Summary

Data stream mining process massive data processing treating pieces of data on the fly. The detection of changes in a data stream distribution is an important issue. This article turns the changes detection into a supervised learning problem. The MODL supervised discretization method has been chosen because of its interesting properties. Our approach is favorably compared with a competitor method on artificial data streams.