Extraction incrémentale de séquences fréquentes dans un flux d'itemsets

Thomas Guyet[*,***], René Quiniou[**,***]

[*]AGROCAMPUS-OUEST
[**]INRIA, Centre de Rennes - Bretagne Atlantique
[***]IRISA - UMR 6074 Campus de Beaulieu, F - 35 042 Rennes Cedex
thomas.guyet@agrocampus-ouest.fr, quiniou@inria.fr

1 Introduction

De nombreuses méthodes ont été proposées pour l'extraction de motifs séquentiels d'une base de transactions. La plupart détermine la fréquence d'un motif à partir du nombre de transactions contenant ce motif, sans tenir compte des répétitions dans une même transaction. Plus rares sont les approches visant à extraire les motifs, ou épisodes, fréquents dans une séquence d'itemsets unique. Le comptage du nombre d'occurrences d'un épisode dans une séquence est plus difficile dans la mesure où il faut tenir compte des répétitions et chevauchements entre les occurrences d'un même épisode. Plusieurs méthodes de comptage ont été proposées pour résoudre cette difficulté tout en conservant des propriétés de monotonie nécessaires à l'efficacité de la recherche d'occurrences (voir Achar et al. (2010)). Par exemple, Winepi (Mannila et al. (1997)) compte toutes les occurrences d'un épisode dans la séquence et la fréquence est le nombre de fenêtres contenant cet épisode lorsque l'on fait glisser la fenêtre sur toute la séquence. Minepi, des mêmes auteurs, compte le nombre d'occurrences minimales d'un épisode.

Des solutions algorithmiques spécifiques doivent être adaptées aux flux de données pour l'extraction de motifs ou épisodes fréquents. Dans un contexte de flux de données, la fenêtre glissante sur laquelle sont extraits les épisodes fréquents est une séquence d'itemsets en perpétuelle évolution : lorsque des nouvelles données arrivent, elles rendent obsolètes celles du début de la fenêtre. Une approche naïve réitèrant l'intégralité du processus de fouille pour la séquence à chaque modification serait trop coûteuse en temps de calcul. La plupart des méthodes d'extraction de séquences fréquentes dans un flux de données traite de la gestion des séquences fréquentes extraites dans des tampons successifs du flux ou "batches" (Marascu et Masseglia (2006)). Seuls quelques algorithmes se sont intéressés au problème de la fouille dans des fenêtres glissantes.

2 Algorithme incrémental de fouille d'un flux d'itemsets

Nous présentons un algorithme incrémental, complet et correct, d'extraction de séquences d'itemsets fréquentes basé sur un dénombrement des occurrences minimales d'une séquence. L'algorithme s'appuie sur la représentation des séquences fréquentes sous la forme d'un arbre

de préfixage inspiré de la méthode PSP (Masseglia et al. (1998)). Au lieu de la fréquence, nous associons à chaque nœud de l'arbre la liste des occurrences du motif représenté par le chemin de la racine à ce nœud. L'algorithme met à jour efficacement cette représentation de l'ensemble des séquences fréquentes à la suite de la suppression d'un itemset en début de fenêtre (données obsolètes) et de l'ajout d'un itemset en fin de fenêtre (nouvelles données) :

1. Suppression du premier itemset : toutes les instances débutant par la position 1 de l'ancienne fenêtre sont supprimées. Les nœuds dont la liste d'instances est de taille strictement inférieure au seuil de fréquence minimale sont ensuite supprimés de l'arbre.

2. Fusion de l'itemset courant à tous les nœuds de l'arbre : cette étape consiste à générer toutes les nouvelles séquences candidates dans la nouvelle fenêtre. Intuitivement, une séquence est une nouvelle candidate si elle est la concaténation d'un sous-itemset du nouvel itemset à une séquence fréquente de l'ancienne fenêtre décalée d'une position. Dans la représentation arborescente des séquences fréquentes, la concaténation se traduit en l'extension de chaque nœud par l'arbre représentant l'ensemble des sous-itemsets de l'itemset courant.

3. Complétion des listes d'instances : pour les nouveaux nœuds candidats, mais uniquement pour ceux-là, il est nécessaire de parcourir de nouveau la fenêtre pour compléter la liste des instances d'une séquence.

4. Élagage des non-fréquents : tous les nœuds de l'arbre dont la liste des instances est de taille strictement inférieure au seuil de fréquence sont supprimés.

Les expérimentations ont été menées sur des données simulées et sur des données réelles. Les résultats montrent que notre algorithme incrémental améliore les temps de calcul de 61% en moyenne par rapport à une approche naïve, non-incrémentale.

Références

Achar, A., S. Laxman, et P. S. Sastry (2010). A unified view of automata-based algorithms for frequent episode discovery. *CoRR abs/1007.0690*.

Mannila, H., H. Toivonen, et A. I. Verkamo (1997). Discovering frequent episodes in sequences. *Data Mining and Knowledge Discovery 1*(3), 210–215.

Marascu, A.-M. et F. Masseglia (2006). Mining sequential patterns from data streams : a centroid approach. *Journal for Intelligent Information Systems 27*, 291–307.

Masseglia, F., F. Cathala, et P. Poncelet (1998). The PSP approach for mining sequential patterns. In *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, pp. 176–184.

Summary

We present a complete and correct incremental algorithm for mining frequent sequences in a sliding window over a stream of itemsets. It relies on a representation of frequent sequences inspired by the PSP algorithm and on an original method for counting the minimal occurrences of a sequence. Experiments made on simulated and real data show that our incremental algorithm significantly improves the computation time compared to a non-incremental approach.