Approche relationnelle de l'apprentissage de séquences

Clément Charnay*, Nicolas Lachiche*, Agnès Braud*

*ICube, Université de Strasbourg, CNRS
300 Bd Sébastien Brant - CS 10413
F-67412 Illkirch Cedex
{charnay,nicolas.lachiche,agnes.braud}@unistra.fr

Des flux de données aux sources de données ouvertes donnant accès à des informations en temps réel, de plus en plus de données peuvent être vues comme des séquences ordonnées. Les besoins en termes d'apprentissage automatique sur ces données séquentielles deviennent donc importants, comme pour des tâches de prévision du futur de la séquence. Dans ce contexte, nous nous intéressons à l'apprentissage supervisé hors ligne de la séquence : étant donnés des exemples ordonnés, nous voulons construire un modèle qui utilise dans ses hypothèses des propriétés des exemples passés.

Pour ce faire, nous introduisons une représentation relationnelle des données séquentielles, où chaque exemple, représenté dans une table, est associé avec tous ses prédécesseurs, relation représentée dans une autre table. Ces approches évitent une représentation attribut-valeur où à chaque exemple sont associés sur la même ligne les exemples précédents dans la limite d'une fenêtre choisie au préalable, approche plus lourde et ne permettant pas d'apprécier les tendances de la séquence.

Attribut-valeur	Relationnel
donnees(x_n, $\mathbf{y_n}$, x_{n-1}, y_{n-1}, x_{n-2}, y_{n-2}, …, x_{n-l}, y_{n-l})	donnees($\underline{id}$, x, $\mathbf{y}$) association($\underline{idPrincipal}$, $\underline{idAnterieur}$, $dist$)

Dans une représentation attribut-valeur, une seule table est utilisée pour représenter les données. Pour un exemple donné, le vecteur de variables prédictrices x_n associées à l'exemple ainsi que la valeur à prédire y_n sont stockées. De plus, on associe à l'exemple les valeurs des variables prédictrices et de la cible pour les exemples antérieurs à l'exemple courant, dans une limite de l exemples. Ainsi x_{n-1} et y_{n-1} représentent les informations associées à l'exemple précédant directement l'exemple courant. Plus généralement, x_{n-k} et y_{n-k} représentent les informations associées au k-ème exemple précédant l'exemple courant. C'est une approche par fenêtrage où $l + 1$ désigne la largeur de la fenêtre.

Cette approche présente un inconvénient : la taille de la fenêtre doit être définie à l'avance. Nous proposons de dépasser cette contrainte grâce à une autre représentation, relationnelle, des données séquentielles. Dans cette représentation, une table est utilisée pour représenter les attributs des exemples courants. Une deuxième table est utilisée comme table d'association pour mettre en correspondance deux lignes de la table des données.

Les colonnes *idPrincipal* et *idAnterieur* de la table *association* référencent toutes deux la colonne *id* de la tabledonnees. Ainsi, une ligne de la table *association* indique que l'exemple

référencé par la colonne *idAnterieur* est antérieur à l'exemple référencé par la colonne *idPrincipal*. La colonne *dist* de la table *association* désigne la distance au sein de la séquence entre les deux exemples.

Dans ce contexte, nous proposons l'utilisation d'agrégats complexes dans un arbre de décision, développée par Charnay et al. (2013). Les agrégats complexes constituent une nouvelle propriété des exemples principaux, construite comme une agrégation d'un attribut sur un sous-ensemble des exemples passés, sélectionné via une condition d'agrégation. Cette condition d'agrégation peut porter sur la distance au sein de la séquence entre l'exemple principal et l'exemple passé. Ceci constitue alors un fenêtrage dynamique étant donné que l'algorithme d'apprentissage choisit lui-même les exemples antérieurs pertinents pour répondre au problème posé. Dans la pratique, ceci se matérialise par des conditions sur la distance au sein de la séquence mais aussi sur les autres attributs, qui ne seront pas les mêmes aux différents nœuds de l'arbre. Une représentation attribut-valeur ne permet pas de construire de tels agrégats.

Nous introduisons également la gestion des granularités temporelles dans les séquences. Par exemple, la granularité de semaine recouvre un ensemble de 7 jours consécutifs. Si l'on considère que les éléments de la séquence sont représentés au niveau des jours et que l'on veut introduire le concept de semaine, ce support se traduit par l'ajout de deux colonnes à la table *association* de la représentation relationnelle :

— Une colonne booléenne *memeJourSemaine* indiquant si les jours associés par la ligne de la table correspondent au même jour de la semaine. Par exemple, si le jour référencé par *idPrincipal* est un lundi, pour toutes les lignes dont le jour référencé par*idAnterieur* est un lundi, la colonne vaudra *vrai*.

— Une colonne *distSemaine* indiquant la distance en nombre de semaines entre la semaine à laquelle appartient le jour *idPrincipal* et la semaine à laquelle appartient le jour *idAnterieur*.

Cette gestion des granularités dans les séquences enrichit la représentation relationnelle, lui permettant de tenir compte des similarités entre l'exemple courant et certains exemples antérieurs au sens d'une granularité temporelle supérieure à celle des exemples.

Références

Charnay, C., N. Lachiche, et A. Braud (2013). Incremental construction of complex aggregates : Counting over a secondary table. In *Late Breaking Papers of the 23rd International Conference on Inductive Logic Programming*, pp. 1–6.

Summary

We observe an increasing amount of sequential data, for instance open data sources provide real-time information. In order to apply classical learning algorithms, sequential data are often modelled in an attribute-value setting using a sliding window. In this paper, we propose a relational approach. A first advantage is to let the relational algorithm choose the length of the window. A second advantage is to allow to consider conditions based on the existential quantifier and aggregates. A third advantage is to be able to consider several granularities at the same time.