Résumé généraliste de flux de données

Projet MIDAS ANR07-MDCO-008, financé par l'ANR
Collectif d'auteurs participant au projet MIDAS*

*contacts : Georges Hébrail ou Christine Potier
Télécom ParisTech, Département INFRES
46 rue Barrault, 75634 Paris Cedex 13
{georges.hebrail, christine.potier}@telecom-paristech.fr
http://midas.enst.fr/

Résumé. Lorsque le volume des données est trop important pour qu'elles soient stockées dans une base de données, ou lorsque leur fréquence de production est élevée, les Systèmes de Gestion de Flux de Données (SGFD) permettent de capturer des flux d'enregistrements structurés et de les interroger à la volée par des requêtes permanentes (exécutées de façon continue). Mais les SGFD ne conservent pas l'historique des flux qui est perdu à jamais. Cette communication propose une définition formelle de ce que devrait être un résumé généraliste de flux de données. La notion de résumé généraliste est liée à la capacité de répondre à des requêtes variées et de réaliser des tâches variées de fouille de données, en utilisant le résumé à la place du flux d'origine. Une revue de plusieurs approches de résumés est ensuite réalisée dans le cadre de cette définition.

1 Introduction

Les nombreux travaux récents relatifs aux flux de données ont permis de dégager clairement les grandes différences entre "traitement des flux de données" et "traitement des bases de données". Une des différences structurantes est que le traitement des flux de données repose sur l'exécution de requêtes continues (valables sur la durée du flux) sur des données volatiles (potentiellement infinies par nature, le flux ne peut pas être stocké) alors que le traitement des bases de données repose sur l'exécution de requêtes volatiles ("one shot") sur des données persistantes (Golab et Özsu 2003).

A l'évidence, une requête continue sur un flux de données ne peut fournir de réponse que sur une période nécessairement limitée. Il est dans la nature des données volatiles de n'être plus disponibles pour les analyses après leur expiration, ce qui suppose de poser a priori sur un flux l'ensemble des requêtes dont on pourra avoir besoin par la suite. Il est clairement hors de question de poser des requêtes a posteriori sur le flux. Ici "a posteriori" doit être entendu comme portant sur des données ayant expiré.

Pour pallier cet inconvénient, différents algorithmes et structures de données ont été proposés dans la littérature, souvent (mais pas toujours) sous le nom de "résumé" ("summary") sans toutefois qu'émerge une notion cohérente de ce qu'est ou devrait être le résumé d'un flux de données.

Dans cet article, nous donnons d'abord la définition de "résumé généraliste" et nous analysons quelques techniques de résumé de flux en regard de cette définition. Puis dans le paragraphe 3, nous analysons le traitement de la dimension temporelle dans chacune de ces méthodes.

2 Définition d'un résumé généraliste de flux de données

2.1 Définition d'un flux de données

Un flux de données, noté F, est une suite infinie d'événements ordonnés dans le temps, chaque événement étant caractérisé par :

- l'identifiant de l'objet ayant émis l'événement (typiquement une adresse dans un espace A non nécessairement connu a priori);
- l'estampille temporelle de l'événement (temps physique ou temps logique, cela dépendra du flux). Soit T l'espace des estampilles temporelles;
- la description de l'événement, qui peut comporter des descripteurs quantitatifs ou qualitatifs ; formellement, l'espace de description est un produit cartésien $D = R^p \times Q$ où R^p est l'espace de description des variables quantitatives et Q celui des variables qualitatives.

A l'exception de l'estampille temporelle, les différents champs peuvent être non renseignés.

2.2 Notion idéale d'un résumé généraliste

Idéalement, un résumé généraliste est une structure de données mise à jour au fur et à mesure de l'arrivée des éléments du flux, et permettant de répondre a posteriori et approximativement à n'importe quelle requête portant sur le flux, de façon optimale compte tenu de contraintes de ressources. Cette structure doit permettre également de calculer des bornes sur la précision des réponses approchées à ces requêtes.

Cette définition idéale appelle quelques commentaires. Sur les contraintes de ressources tout d'abord : ces contraintes sont au moins de trois types, contraintes sur la mémoire disponible pour stocker le résumé, contrainte sur le temps de construction / mise à jour du résumé et contrainte sur le temps d'exécution d'une requête sur le résumé.

On se contente ici d'envisager un résumé sur un flux unique ; les notions de contrainte de communication (bande passante) dont il faut tenir compte dans le cas du traitement distribué de flux de données seront donc ignorées.

L'existence de ces contraintes et le caractère potentiellement infini du flux de données rendent inévitable une dégradation de l'information présente dans le résumé ; on ne peut qu'imposer que cette dégradation se fasse de façon "optimale". On remarque que le résumé devant être à même de traiter toutes les requêtes, la notion d'optimalité est à définir sans faire référence à des requêtes particulières ; la possibilité de calculer des bornes sur les approximations obtenues des réponses aux requêtes en fonction des contraintes est une exigence supplémentaire, portant sur le résumé et qui permet de l'utiliser de façon contrôlée.

On note finalement que cette définition idéale impose de répondre à toutes les requêtes, ce qui distingue d'emblée la notion de "résumé généraliste" de la notion de "sketch" ou "synopsis" qui sont des structures de données conçues pour répondre (de façon optimale sous contrainte de ressources) à un type particulier de requête (par exemple requête COUNT pour le Count Min Sketch (Cormode et Muthukrishnan 2004).

2.3 Définition réaliste d'un résumé généraliste

Une définition "réaliste" vise en particulier à guider l'inévitable perte d'information entre le résumé et le flux. Une première proposition est de restreindre la classe des requêtes sur le résumé aux seules requêtes de type SELECT AND COUNT où la sélection (prise ici au sens de *filtrage*) peut se faire sur toutes les variables, à l'exclusion des identifiants; les requêtes ponctuelles portant sur tel ou tel objet sont donc exclues du champ du résumé généraliste.

Cette restriction laisse une très grande expressivité au résumé généraliste : en effet, la capacité à répondre exactement à toute requête de type SELECT AND COUNT sur le temps et les variables descriptives revient à connaître exactement la densité jointe sur l'espace $T \times D$. En d'autres termes, le résumé généraliste permet une estimation approchée de la densité jointe. Alternativement, une estimation approchée de la densité jointe permet de répondre approximativement à toute requête de type SELECT AND COUNT.

Un résumé généraliste est donc une structure de données permettant une approximation optimale de la densité jointe sur $T \times D$, sous les contraintes d'espace mémoire, de temps de mise à jour / construction et d'exécution des requêtes.

La notion d'optimalité sous contrainte d'espace-mémoire peut être abordée en théorie sous l'angle de la théorie du Minimum Description Length (MDL) (Gründwald 2007). En dépit d'inconvénients pratiques évidents (l'optimum MDL est non calculable, la qualité de l'approximation ne décroît pas de façon monotone quand la longueur de description diminue, voir Adriaans et Vitanyi 2007), l'approche MDL reste une heuristique séduisante pour définir l'optimalité sous contrainte d'espace-mémoire. Une solution "pure MDL" permet en particulier de mettre en évidence les interactions complexes entre les contraintes d'espace-mémoire et les contraintes temporelles. En effet, les approches "pure MDL" peuvent reposer sur des techniques de compression des données très sophistiquées qui auront un impact important sur les temps de mise à jour / construction du résumé et d'exécution des requêtes.

Nous proposons de limiter la notion de "résumé généraliste" à une structure de données qui laisse l'espace $T \times D$ sous sa représentation native. Un avantage immédiat de cette restriction est que les requêtes s'exécutent sur le résumé de la même façon que sur le flux, ce qui fait disparaître la contrainte temporelle sur l'exécution des requêtes. Il ne reste donc que les contraintes d'espace-mémoire et de temps de mise à jour / construction.

La discussion qui précède nous amène à proposer la définition suivante pour un "résumé généraliste de flux de données".

Définition : Un résumé généraliste d'un flux de données (tel que défini plus haut) est une structure de données :

(1) dont la construction respecte des contraintes d'espace-mémoire et de temps de calcul,
(2) s'exprimant sous forme de variables appartenant à $T \times D$,
(3) permettant une approximation de toute requête de type SELECT AND COUNT sur $T \times D$,
(4) permettant le calcul de l'erreur d'approximation en fonction des ressources mémoire et CPU disponibles.

Il faut encore spécifier la nature des contraintes. Ainsi, la contrainte sur l'espace-mémoire peut prendre différentes formes selon qu'on souhaite ou non tenir compte du volume du flux écoulé à l'instant courant ($|F(t)|$): la contrainte la plus simple impose une borne fixe au volu-

me du résumé mais on pourrait aussi introduire une borne maximale en $O(log(|F(t)|)$, voire d'autres dépendances plus complexes (polylog par exemple). Ceci resterait cohérent avec la définition au-dessus : on ne ferait que préciser une classe de résumé généraliste en indiquant le type de contrainte auquel il est soumis. On peut ainsi définir un résumé généraliste à espace constant, à espace logarithmique (en fonction du temps) etc. De la même façon, rien n'interdit d'envisager des contraintes de puissance de calcul évoluant dans le temps.

De façon générale, tout algorithme de flux possède un mode "naturel" de dégradation face à la contrainte temporelle : l'échantillonnage aléatoire à réception des événements, dont l'effet sur la précision des réponses aux contraintes SELECT AND COUNT peut être borné. L'exigence de respect de la contrainte sur la puissance de calcul consiste donc essentiellement à pouvoir produire une borne sur le temps de mise à jour / construction du résumé. La connaissance de cette borne permet de dimensionner a priori l'échantillonnage en entrée en fonction des caractéristiques du flux qu'on supposera connues (débit maximal par exemple).

2.4 Analyse d'approches de résumés de la littérature

Nous avons sélectionné quelques "résumés généralistes" de la littérature afin de les passer au crible des contraintes de notre définition.

- Echantillonnage à réservoir : (Vitter 1985), (Féraud *et al*. 2009), (Raïssi *et al* 2007).
- Echantillonnage progressif : StreamSamp (Csernel *et al.* 2006)
- Construction de micro-classes : Clustream (Aggarwal *et al.* 2003)
- Extraction de motifs fréquents : (Vinceslas et al. 2009)
- Cubes de données et flux : (Pitarch *et al* 2008).
- Algorithme REGLO : (Marascu *et al* 2010).

Une description détaillée des résultats ainsi que la construction et l'utilisation de résumés de flux de données se trouve sur le site du projet (http://midas.enst.fr/) : http://midas.enst.fr/wakka.php?wiki=PublicationS/download&file=resume_generaliste.pdf

3 Analyse du traitement de la dimension temporelle

L'estimation de densité sur $T \times D$ peut se faire : (1) en considérant globalement l'espace $T \times D$; (2) en découplant la dimension temporelle dans l'estimation, par la définition d'un système de *fenêtres temporelles* au sein desquelles on réalise une estimation de la densité sur D (ou sur $T_f \times D$, T_f étant la partie de T correspondant à la fenêtre). La caractéristique majeure de l'approche (2) est que la structure des fenêtres temporelles est connue à l'avance indépendamment des données, ce qui permet la mise en place d'algorithmes plus efficaces pour la construction et la mise à jour du résumé mais au détriment de l'optimalité de la représentation du résumé. Afin de permettre la reconstitution d'un résumé sur n'importe quelle période du passé, il est nécessaire de pouvoir construire l'estimation de la densité de l'union de deux fenêtres temporelles à partir du résumé des deux fenêtres (et non du flux détaillé).

Dans les approches présentées précédemment, 3 systèmes de fenêtrage sont considérés :

- Une *fenêtre unique* couvrant toute la période d'observation du flux jusqu'à l'observation de l'instant présent.
- Des fenêtres dites de type « *batch* » : il s'agit de fenêtres adjacentes de taille fixe, en général définies au niveau logique (nombre d'événements) plutôt qu'au niveau physique afin de maîtriser leur taille.

- Des fenêtres dites à structure *logarithmique* (aussi appelées *tilted time windows*) dont la couverture temporelle varie de façon exponentielle au fur et à mesure qu'elles vieillissent.

Le tableau ci-dessous synthétise notre analyse des différentes approches de résumé suivant ce critère de traitement de la dimension temporelle.

Approche de résumé	**Type de fenêtres / taille du résumé**	**Découplage de la dimension temporelle**	**Méthode d'estimation de la densité sur D**	**Estimation de la densité sur l'union de 2 fenêtres**
Réservoir	Fenêtre unique / fixe	Non	Echantillonnage aléatoire	Sans objet
StreamSamp	Logarithmique / logarithmique	Oui	Echantillonnage aléatoire	Oui : union des échantillons avec pondération
Clustream	Logarithmique / logarithmique au mieux	Partiel	Construction de micro-classes	Oui : soustraction des clichés correspondant aux bornes de la fenêtre
Motifs fréquents	Batch / linéaire	Oui	Co-occurrences d'items au-dessus d'un support	Non, ou de façon approximative
Cuboïdes	Batch / logarithmique	Non	Agrégation multi-dimensionnelle	Oui : agrégation multi-dimensionnelle
REGLO	Fenêtre unique / fixe	Non	Coefficients de régression	Sans objet

L'approche Clustream réalise un découplage partiel de la dimension temporelle. En effet, Clustream conserve une estimation de la densité du flux par un ensemble évolutif de micro-classes dont des clichés sont archivés régulièrement. L'introduction de la dimension temporelle *T* au côté de *D* dans la construction des micro-classes permet de conserver une estimation de cette densité sur *T x D*. Pour n'importe quelle fenêtre du passé, il est possible de reconstituer cette densité à condition que des clichés correspondant aux bornes de ces fenêtres aient été conservés.

Les cuboïdes sont alimentés en considérant des fenêtres de type batch traitées de façon consécutive indépendamment les unes des autres. Il peut donc sembler en première analyse qu'il y a un découplage de la dimension temporelle. En réalité, l'estimation de densité s'opère bien sur toute la dimension temporelle car des hiérarchies sur des dimensions autres que le temps peuvent être utilisées pour agréger les données anciennes lorsque le cube occupe trop de place. Ainsi la dimension temporelle permet de piloter l'oubli des données mais les agrégations réalisées pour l'oubli ne portent pas uniquement sur la dimension temporelle.

En comparant certains "résumés" de la littérature à cette définition, nous avons montré comment cette définition peut constituer une base pragmatique pour la comparaison de ces différentes structures de données.

Références

Adriaans P., Vitanyi P. (2007). *The Power and Perils of MDL*, IEEE International Symposium on Information Theory, ISIT 2007, 2216-2220.

Aggarwal C.C., Han J., Wang J., Yu P.S. (2004). *A Framework for clustering evolving data streams*, Conférence VLDB.

Cormode G., Muthukrishnan S. (2004). *An improved data stream summary: The count-min sketch and its applications*, in proceedings of Latin American Theoretical Informatics (LATIN), 29-38.

Csernel B., Clérot F., Hébrail G. (2006). *StreamSamp: Data stream clustering over tilted windows through sampling*, Workshop ECML/PKDD Knowledge Discovery from Data Streams, Berlin.

Féraud R., Clérot F., Gouzien P. (2009). *Sampling the join of streams*, IFCS'2009, Dresden.

Golab L., Özsu M.T. (2003). *Issues in Data Stream Management*. SIGMOD Record, Vol. 32, No. 2.

Grünwald P.D. (2007), *The Minimum Description Length Principle*. MIT Press.

Marascu, A., Masseglia, F., Lechevallier Y. (2010). *A Fast Approximation Strategy for Summarizing a Set of Streaming Time Series*, à paraître dans Proceedings of the ACM Symposium on Applied Computing (Sierre, Switzerland), SAC '10, ACM, NY.

Pitarch Y., Laurent A., Plantevit M., Poncelet P. (2008). *Fenêtres sur cube*. BDA 2008, Guilherand Granges, France, 1-20.

Raïssi C., Poncelet P. (2007): *Sampling for Sequential Pattern Mining: From Static Databases to Data Stream*. Proceedings of ICDM 07, Omaha NB, USA.

Vinceslas L.,Symphor J.E., Mancheron A., Poncelet P. (2009). *SPAMS: Une nouvelle approche incrémentale pour l'extraction de motifs séquentiels fréquents dans les data streams*. EGC 2009, 205-216.

Vitter, J. S. (1985). *Random Sampling with a Reservoir*. ACM TOMS, Vol. 11, No. 1, 37-57.

Summary

When the volume of data is too high to be stored in a database at a reasonable cost or when data is arriving at a high speed, Data Stream Management Systems (DSMS's) can capture streams of structured records and query them on the fly by defining permanent queries. But DSMS's do not save the history of the streams which is lost forever. This paper proposes a formal definition of what should be a generic summary of the history of a structured data stream. Generic refers here to the possibility of answering various queries or performing various mining tasks from the summary instead of the whole stream. Then it reviews several summarizing approaches according to this definition.