Data Streams

Mardi 21 octobre 2003

Joannès Vermorel

Sommaire

- ✓ Polytechnic University
- ✓ Data streams
  - Introduction
  - Frequent item mining
  - Echantillonnage et histogrammes
- ✓ Autres activités

Polytechnic University

Polytechnic University

- Fondée en 1854
- Située dans le *Metrotech Center*

- Petite taille (~2000 étudiants)

Ci-contre en haut le Othmer Hall (résidence étudiante de Poly). Dessous, une vue aérienne du Brooklyn Bridge.

Data Streams – contexte 1/2

- Flux très importants de données : paquets TCP/IP, gros systèmes transactionnels, capteurs physiques.
- L'accès rapide à l'intégralité des données est impossible.
- Modélisation : l'accès aux éléments est strictement séquentiel. La quantité de mémoire disponible est très inférieure au nombre d'éléments du flux.

Data Streams – contexte 2/2

Origine du problème: explosion des temps de latence pour l'accès aux mémoires de grandes capacités.

Temps de latence (ordres de grandeur) :

- cache processeur : 1ns / 500ko
- ram : 10ns / 500mo
- disque durs : 10ms / 500go

Data Streams - problématique

- Réécriture des algorithmes classiques de la statistique dans le contexte des data streams.

Exemples :

- Calcul de la médiane (des quantiles).
- Calcul des objets fréquents.

Data streams - difficultés

- Résultats en lignes nécessairement approximatifs. Difficultés : garantir la qualité des approximations fournies.
- Flux rapide : temps de traitement pour chaque objet très limité (*O(1)* si possible).

Data streams - applications

- Optimisation des requêtes pour les bases de données.
- Surveillance / optimisation de réseaux TCP/IP.
- Data mining sur des gros systèmes transactionnels.

Frequent item mining 1/2

Problème : on dispose d'un flot d'objets x_1, x_2, …, x_N appartenant à E. On veut pouvoir déterminer l'ensemble des points de fréquences supérieures à ε.

Contrainte : l'algorithme doit faire appel à une mémoire en o(N).

Frequent item mining 2/2

Algorithme **COUNT-MIN** (Cormode, Muthukrishnan, juin 2003):

- ε : erreur d'estimation de fréquence.
- $w = e / \varepsilon$: taille des tables de hachages
- d : nombre de tables de hachages

Chaque table fournit un estimateur. En faisant le MIN de tous les estimateurs, on obtient une estimation correcte à ε près avec une probabilité en $O(e^{-d})$.

Data streams - travail de stage

- Méthode « universelle » de mise en ligne : **échantillonner le flot**
- Problème : l'échantillon est-il représentatif du flot tout entier ?
- Note: dans le cadre des data streams, même l'algorithme d'échantillonnage aléatoire n'est pas trivial.

Point de départ

- Papier (KDD03) de **Hervé Brönnimann** sur **EASE**, une méthode d'échantillonnage garanti appliquée aux *frequent itemsets* pour le *market basket data analysis*.
- Objectif : généraliser la méthode pour les données tabulaires (modèle vectoriel mixte, dimensions discrètes et continues).

Qualité d'échantillonnage

- Objectif de EASE : garantir simultanément à ε près les fréquences d'un nombre quelconque de fonctions caractéristiques dans l'échantillon.

Problèmes :

- Comment garantir ces fréquences ?
- Quelles fonctions caractéristiques choisir ?

Garantir des fonctions caractéristiques

- EASE procède par *halving*, i.e. par partition du flot en deux ensembles (bleu et rouge) suffisamment « ressemblants ».
- A chaque FC, on associe deux pénalités bleu et rouge. La pénalité totale est la somme de toutes les pénalités.
- Lorsqu'un point du flot arrive, on choisit le coloriage qui minimise la pénalité totale.
- EASE (théorème) garantie une erreur d'estimation maximale en $O(N^{-½}ln(k))$ pour chaque FC.

Choisir les fonctions caractéristiques : cas numérique

- La segmentation est une méthode naturelle pour générer des fonctions caractéristiques.
- Problème : définir la segmentation optimale (à nombre de segments fixé) pour une distribution.
- Interprétation de cette segmentation optimale sous la forme d'histogrammes optimaux pour la *freedom cost function*.

Histogrammes optimaux

- Le calcul exact des histogrammes optimaux en $O(n^2 ln(n))$ est trop coûteux.
- Introduction de l'algorithme **ECHAP** qui permet d'obtenir en $O(n)$ un histogramme sup-optimal en introduisant un nombre limité de segments supplémentaires.

Histogrammes optimaux en ligne

- Double problème du passage en ligne de ECHAP : mémoire limitée (classique) et nécessité de figer rapidement l'histogramme (spécifique).
- Introduction de l'algorithme **SECHAP** (version en ligne de ECHAP) qui fournit des garanties sur ces deux contraintes.

Implémentation de SECHAP

- L'algorithme SECHAP a été implémenté et validé pour un modèle abstrait de données tabulaires (bibliothèque C#).
- Implémentation additionnelle d'une interface pour un système de base de données spécifique (Microsoft Access).

Références

- *Efficient data reduction with EASE*, Herve Bronnimann, …, KDD2003, http://photon.poly.edu/~hbr/publi/kdd03.pd
- *Improved data stream summaries: Count-Min sketch and its applications*. Cormode and Muthukrishnan. DIMACS TR 2003-20, http://www.cs.rutgers.edu/~muthu/

Une expérience unique: réussir à prendre l'avion en plein **blackout**

- Les portes, les ascenseurs, le téléphone ne fonctionnaient pas…
- Les pompes pour l'eau et le carburant (taxis, avions) fonctionnaient pas…
- La voie lactée fut visible au dessus de Manhattan.