Data Mining, Lecture 8

Mining Data Streams

S. Nõmm

[1] Department of Software Science, Tallinn University of Technology

29.10.2019

Introduction

- *Assumption:* it is not possible to store all the data.
- In reality this assumption is not true any more. There are big data based distributed storage techniques etc.
- Avoiding this assumption leads: enormous storage costs, loss of real time processing capabilities etc.
- One may say, that assumption is true.

Examples

- Transactions.
- Web clicks.
- Social streams.
- Networks streams.

Unique challenges

- One pass content: it is assumed that the data can be processed only once.
- Concept drift: the data may evolve over time.
- Resource constraints: it is not always possible to control the process generating the stream. Loadshedding - is the process of dropping tuples which can not be processed.
- Massive domain constraints.

Reservoir Sampling

- Sampling is one of the methods for stream summarization.
- Main advantage of the sampling: after the sample is drawn any offline algorithm may be applied.
- Reservoir sampling is the methodology to maintain a dynamic sample from the data.
- In this case the sample is referred as *reservoir sample*.
- The goal is to continuously maintain a dynamically updated sample of k points from a data stream without explicitly storing the stream.
- The sampling approach works with incomplete knowledge about the previous history of the stream at any given moment in time.

Admission control

- Sampling rule to decide whether to include the incoming data point in the sample or not?
- The rule to decide whether to eject a data point from the sample or not, to make room for the newly inserted data point?

Reservoir sampling algorithm

For the sample of size k:
Initialize: include first k points into the sample.

- Insert the nth incoming stream data point in the reservoir with probability k/n.
- If the newly incoming data point was inserted, then eject one of the old k data points in the reservoir at random to make room for the newly arriving point.

This method allows to maintain an unbiased reservoir sample from the data stream.

Theorem

After n stream points have arrived, the probability of any stream point being included in the reservoir is the same, and equal to k/n.

Concept Drift

- Assumption: recent data considered more important than older data.
- A uniform random sample from the reservoir will contain data points that are distributed uniformly over time.
- Decay-based framework used to regulate relative importance of data points.
- So called *bias functions are used*.

Concept Drift

Let $p(r,n)$ be the probability of the rth data point belong to the reservoir when nth point arrives. Define function $f(r,n)$ to be proportional to $p(r,n)$. In the frameworks of the reservoir sampling $f(r,n)$ is referred as *bias function*.

- $f(r,n)$ decreases monotonically with n whereas r is fixed.
- $f(r,n)$ increases monotonically with r whereas n is fixed.
- Recent data points have a higher probability of belonging to the reservoir.

Definition

Let $f(r,n)$ be the bias function. The sample $\mathcal{S}(n)$ of size n is said to be biased (or bias sensitive) with respect to the bias function $f(r,n)$ if $p(r,n)$ is proportional to $f(r,n)$.

Open problem

It is an open problem to perform reservoir sampling with an arbitrary bias function. There is number of methods exists for the exponential bias function.

$$f(r,n)^{=}e^{-\lambda(n-r)}$$

where λ defines bias rate, preferably in the range of $[0,1]$.

- The case when $\lambda = 0$ represents the unbiased case.
- The exponential bias function belongs to the class of memoryless functions.
- Interesting from the viewpoint of space-constrained scenarios, where reservoir size $k < 1/\lambda$.

- Assume reservoir size $k < 1/\lambda$.
- Start with an empty reservoir.
- Replacement policy:
  - Assume that before the nth point arrives the fraction reservoir filled is $F \in [0, 1]$.
  - Insertion probability of the point $n + 1$ is $\lambda \cdot k$.
  - Consider the reservoir is not full. Random generator with the success probability of $F(n)$ is used to decided if one of the older points should be randomly chosen to be removed from the reservoir.x

Theoretical bounds for sampling

- Reservoir sampling method provides data samples.
- Usually samples are used to *estimate* statistical aggregates.
- The accuracy of this aggregates may be provided by following bounds.

Markov and Chebyshev Inequalities

Theorem

Let X be a random variable that takes on only nonnegative random values. then , for any constant α satisfying $E[X]) < \alpha$, the following inequality holds:

$$P(X > \alpha) =\leq E[X]/\alpha.$$

Theorem

Let X be an arbitrary random variable. Then, for any constant α, the following inequality holds.

$$P(|X - E[X]| > \alpha) \leq V([X]/\alpha^2)$$

here V denotes variance.

Reservoir Sampling

- Assume that the data points have binary labels associated with them. The goal is to use stream sample to estimate the fraction of examples belonging to each class. To do this it is necessary to bound the probabilistic accuracy of the bounds.
- Maintain a reservoir sample S from the data stream.
- Apply a frequent pattern mining algorithm to the reservoir sample $\mathcal{S}$ and report the patterns.
- The probability of a pattern being a false positive can be determined by using the Chernoff bound

Theorem

Lower-Tail Chernoff Bound *Let X be a random variable which can be expressed as the sum of n independent binary random variables, each takes on the value of 1 with probability p_i. Then for any $\delta \in (0,1)$*

$$P(X < (1-\delta)E[X]) < e^{\frac{E[X]\delta^2}{2}}$$

Chernoff Bound

Theorem

Upper-Tail Chernoff Bound *Let X be a random variable which can be expressed as the sum of n independent binary random variables, each takes on the value of 1 with probability p_i. Then for any $\delta \in (0,1)$*

$$P(X < (1-\delta)E[X]) < e^{\frac{-E[X]\delta^2}{4}}$$

Theorem

Lower-Tail Chernoff Bound *Let X be a random variable which can be expressed as the sum of n independent binary random variables, each takes on the value of 1 with probability p_i. Then for any $\delta \in (0, 2e-1)$*

$$P(X > (1+\delta)E[X]) < e^{\frac{-E[X]\delta^2}{2}}$$

Hoeffding bound

Assume that underlying random variables are not necessary binary but bounded from certain interval.

Theorem

Let X be a random variable that can be expressed as the sum of n independent random variables, each of which is bounded in the range $[l_i, u_i]$.

$$X = \sum_{i=1}^{n} X_i.$$

then for any positive θ, the following inequalities hold.

$$\begin{aligned} P(X - E[X] > \theta) &\leq e^{-\frac{2\theta^2}{\sum_{i=1}^{n}(u_i - l_i)^2}} \\ P(E[X] - X > \theta) &\leq e^{-\frac{2\theta^2}{\sum_{i=1}^{n}(u_i - l_i)^2}}. \end{aligned}$$

Synopsis Structures for the Massive-Domain Scenario

- Bloom filter: Given a particular element, has it ever occurred in the data stream?.
- Count-Min Sketch
- AMS Sketch
- FlajoletMartin Algorithm for Distinct Element Counting

Bloom filter

- The bloom filter main applicaiton area is membership queries.
- A bloom filter is a binary array of length m indexed from 0 to $m-1$.
- It is associated with a set of independent hash functions $\{h_1(\cdot), h_\omega(\cdot)\}$. which arguments are the elements of the data stream.
- The hash function maps uniformly at random to an integer in the interval $[0, m-1]$
- To check membership of an item in the data stream compute the hash functions. Let element of interest is y. check if $h_i(y)$th element is equal to one. If at least one of these values is 0 it is guaranteed that the element has not occurred in the data stream.

Count-Min Sketch

- Count-Min Sketch for the case of count-based summaries.
- It may be seen as the matrix. Where indexing is similar to the previous case.
- Each row corresponds to a hash function which have the following properties:
  1. The ith has function maps a stream element similarly to the Bloom filter case.
  2. Hash are pairwise independent over different arguments.

Frequent Pattern Mining in Data Streams

- Reservoir sampling and Sketches may be use to leverage synopsis structure.
- Lossy Counting Algorithm

Clustering Data Streams

- STREAM Algorithm: The core idea is to break the stream into smaller memory-resident segments.
- CluStream Algorithm: Based on the idea of micro clustering.
- Massive Domain Stream Clustering

Microclusters

Denote multidimensional records in the data stream as $X_1, \dots X_k$ and corresponding time stamps $T_1, \dots, T_k$. Each $X_i = (x_i^1, \dots X_i^d)$.

Definition

A microcluster for a set of d-dimensional points $X_{i_1}, \dots, X_{i_n}$ with time stamps $T_{i_1}, \dots, T_{i_n}$ is the $(2d+3)$ tuple $(\overline{CF2^x}, \overline{CF1^x}, CF2^t, CF1^t, n)$ where

1. $\overline{CF2^x}$ the sum of squares of the data values for each dimension. The pth entry is equal to $\sum_{j=1}^{n} (x_{i_j}^p)^2$.
2. $\overline{CF1^x}$ the sum of the data values for each dimension.
3. $CF2^t$ the sum of the squares of the time stamps.
4. $CF1^t$ the sum of the time stamps.
5. n the number of the data points.

Usually used with $BIRCH$ Balanced Iterative Reducing and Clustering with Hierarchies.

Streaming Outlier Detection

- Outlier detection of individual records.
- Changes in the aggregate trends of the multidimensional data

Streaming Classification

Impact of the concept drift makes the problem extremely challenging.

- Very fast decision trees (VFDT).
- Supervised Microcluster Approach

Closed Book test II

Closed Book test will take place on November the 12th. Usual lecture time: 14:00 ICT-A1.