0368-3248-01-Algorithms in Data Mining **Fall 2012**

Lecture 3: Assignment 1

Lecturer: Edo Liberty

Warning: *This note may contain typos and other inaccuracies which are usually discussed during class. Please do not cite this note as a reliable source. If you find mistakes, please inform me.*

---

1 Approximating the size of a tree

setup

In this question we will try to approximate the number of leaves in a tree. A binary tree is a graph consisting of internal nodes and n leaves. Each internal node, u, has two children. A left child $l(u)$ and a right child $r(u)$. The only node which does not have a parent is the root of the tree u_{root}. For each node we also denote by $d(u)$ its depth in the tree which is the distance from the root. For example $d(u_{rood}) = 0$ and $d(r(u_{root})) = 1$.

We define a random walk on a tree as the process of starting at the root and then randomly moving to one of the children until we hit a leaf. More precisely:

1. $u \leftarrow u_{root}$
2. while u is an internal node
3. w.p. 1/2
4. $u \leftarrow l(u)$
5. otherwise
6. $u \leftarrow r(u)$
7. return u

questions

1. Let the leaf u be at depth $d(u)$. Calculate the probability, $p(u)$, that the random walk outputs u?
2. Let x be the output leaf of a random walk and let $f(x) = 2^{d(x)}$ be a function defined on the leaves. Compute the value of:
$$E_{x\sim w}[f(x)]$$
where $x \sim w$ denotes that x is chosen according to the distribution on the leaves generated by the random walk.
3. We say that a tree is c-balanced if $d(u) \leq \log_2 n + c$ for all leaves in the tree. Show that for a c-balanced tree
$$\mathrm{Var}_{x\sim w}[f(x)] \leq 2^c n^2$$
4. Let $Y = \frac{1}{s}\sum_{i=1}^{s} f(x_i)$ where x_i are output nodes of s independent random walks on the tree. Compute $E[Y]$ **and** show that $\mathrm{Var}[Y] \leq 2^c n^2/s$.

5. Use Chebyshev's inequality to find a value for s such that for two constants $\varepsilon \in [0,1]$ and $\delta \in [0,1]$:

$$\Pr[|Y - n| > \varepsilon n] < \delta.$$

   s should be a function of c, ε and δ.

2 Approximate histograms

setup

We are given a stream of elements $x_1, \ldots, x_N$ where $x_i \in \{a_1, \ldots, a_n\}$. Let n_i denote the number of times element a_i appeared in the stream, i.e., $n_i = |\{j | x_j = a_i\}|$. Our goal is to estimate n_i for all frequent elements. Let the sub stream y include every element in the stream x with probability p. let $\hat{n}_i = |\{j | y_j = a_i\}|$ be the number of times a_i appears in y.

questions

1. Let $z_i = \hat{n}_i / p$, compute $\mathbb{E}[z_i]$
2. Assume a_1 is such that $n_1 \geq \theta N$ for some fixed θ. Compute a value for p (as low as possible) which guaranties that $n_1(1+\varepsilon) \geq z_1 \geq n_1(1-\varepsilon)$ w.p. at least 1/2.
3. Assume a_1 is such that $n_1 < \theta N(1-2\varepsilon)$ for some fixed θ. Compute a value for p (as low as possible) which gives that $z_1 \leq \theta N(1-\varepsilon)$ w.p. at least 1/2.
4. Use the union bound to specify a value for p which guaranties that for every i, if $n_i \geq \theta N$ then $n_i(1+\varepsilon) \geq z_i \geq n_i(1-\varepsilon)$ and if $n_i < \theta N(1-2\varepsilon)$ then $z_i \leq \theta N(1-\varepsilon)$ with probability at least $1-\delta$.
5. Compare this result with the algorithm described in class for approximately counting frequent items in streams, which is better under what circumstances?