CSE 4502/5717 Big Data Analytics. Fall 2024

Exam III Solutions

1. Since each item of I occurs in $\leq c$ transactions, it follows that $\sum_{t \in DB} |t| \leq cd$, i.e., $\sum_{t \in DB} |t| = O(d)$. We form a sequence S of all the items in all the transactions in DB. Note that $|S| = O(d)$. Also, we can think of each item as an integer in the range $[1, d]$. Thus we can sort S in $O(d)$ time. We scan through the sorted S to identify all the frequent items. The total runtime will be $O(d)$.

2. Pick a random sample S of s transactions from DB (The value of s will be finalized in the analysis). Let q and m be the number of transactions in DB and S, respectively, that contain X. m can be computed in $O(sk)$ time. We output $\mu = m\frac{n}{s}$. The expected value of m is $s\frac{q}{n}$. Using Chernoff bounds, $Prob.\left[m \geq 1.1\frac{sq}{n}\right] \leq \exp\left(-0.01\frac{sq}{3n}\right)$. This probability will be $\leq \frac{n^{-\alpha}}{2}$ if $s \geq 600\alpha\sqrt{n}\log_e n$. Also, using Chernoff bounds, $Prob.\left[m \leq 0.9\frac{sq}{n}\right] \leq \exp\left(-0.01\frac{sq}{2n}\right)$. This probability will be $\leq \frac{n^{-\alpha}}{2}$ if $s \geq 400\alpha\sqrt{n}\log_e n$.

   In summary, if we pick s to be $\geq 600\alpha\sqrt{n}\log_e n$, μ will be in the interval $[0.9q,\ 1.1q]$ with a probability of $\geq (1 - n^{-\alpha})$. The tun time of this algorithm will be $O(k\sqrt{n}\log n)$.

3. Note that there are $\binom{d}{k} < d^k$ possible k-itemsets. We can generate all possible k-itemsets in $O(1)$ time using $\binom{d}{k}$ processors. These are the candidates. Followed by this, we count the support for each possible k-itemset. We assign $\frac{n}{\log n}$ processors for each k-itemset candidate. The support can be computed in $O(\log n)$ time using a prefix computation. Details follow.

```
1) **for** each candidate $c$ **in parallel do**
2)     **for** each transaction $t$ **in parallel do**
3)         Processor $P_{c,t}$ computes $b_{c,t}$ as 1 if $c$ is in $t$ and zero otherwise;
4)     $\frac{n}{\log n}$ processors perform a prefix sums computation on $b_{c,1}, b_{c,2}, \ldots, b_{c,n}$.
5)     If this sum is $\geq minSupport$, output $c$ as a frequent $k$-itemset;
```

   Analysis: All the candidates can be generated in $O(1)$ time. For a given candidate c, if we have n processors, we can complete step 3 in $O(1)$ time. Using the slow-down lemma, this can be done in $O(\log n)$ time using $\frac{n}{\log n}$ processors. Step 4 takes $O(\log n)$ time and step 5 takes $O(1)$ time. Thus the total run time is $O(\log n)$.

4. Consider a complete binary tree with k leaves where each leaf has one of the input polynomials. Perform a computation up the tree as follows. Each internal node multiplies the two children polynomials and sends the result to its parent. When the root completes its operation we get the product of the k ploynomials. There are $\log k$ levels in the tree and the time spent at each level is $O(n \log n)$. Thus the run time of the algorithm is $O(n \log n \log k)$.

5. Sort A and B. Let a_i be the number of elements $x \in A$ such that $x = i$. Let b_i be the number of elements $y \in B$ such that $y = i$. Let $P_1(x) = a_{5n}x^{5n} + \ldots + a_0, P_2(x) = b_{5n}x^{5n} + \ldots + b_0$.

Let $P_3(x) = P_1(x)P_2(x) = c_{10n}x^{10n} + \ldots + c_0$. Then $c_i = |C_i|$. Sorting A and B takes $O(n)$ time, computing a_i, b_i for $i = 0, \ldots, 5n$ takes $O(n)$ time, and computing c_i for $i = 0, \ldots, 10n$ takes $O(n \log n)$ time.

6. Evaluate the given polynomial at each integer in the range $[1, cn]$. If the polynomial evaluates to zero at any v, then v is a root. We can evaluate the polynomial at cn points in $O(n \log^2 n)$ time as has been mentioned in class.