CS 574: Randomized Algorithms

Lecture 4. Occupancy Problems, Balls in Bins

September 3, 2015

Preliminaries that we saw in previous lectures (or/and you should know already)

- Variance, Standard Deviation.

Preliminaries that we saw in previous lectures (or/and you should know already)

- Variance, Standard Deviation.
- Properties of Variance (product/sum)

Preliminaries that we saw in previous lectures (or/and you should know already)

- Variance, Standard Deviation.
- Properties of Variance (product/sum)
- Bernoulli distribution.

Preliminaries that we saw in previous lectures (or/and you should know already)

- Variance, Standard Deviation.
- Properties of Variance (product/sum)
- Bernoulli distribution.
- Binomial distribution.

Preliminaries that we saw in previous lectures (or/and you should know already)

- Variance, Standard Deviation.
- Properties of Variance (product/sum)
- Bernoulli distribution.
- Binomial distribution.
- Union Bound.

Preliminaries that we saw in previous lectures (or/and you should know already)

- Variance, Standard Deviation.
- Properties of Variance (product/sum)
- Bernoulli distribution.
- Binomial distribution.
- Union Bound.
- Stirling's approximation.

Preliminaries that we saw in previous lectures (or/and you should know already)

- Variance, Standard Deviation.
- Properties of Variance (product/sum)
- Bernoulli distribution.
- Binomial distribution.
- Union Bound.
- Stirling's approximation.
- Markov, Chebychev from last lecture.

Happy Birthday!

Is it more likely that two of you share the same birthday or that no two of you share the same birthday?

Happy Birthday!

Is it more likely that two of you share the same birthday or that no two of you share the same birthday?

- **Birthday Paradox:** If you have 30 people in the room and you ask them for the date (month and day) of their birthday, then the probability that all birthdays are distinct is less than 30 percent.

Probability to have a pair with the same birthday

- 1 person : 0.0%
- 5 people: 2.7%
- 10 people: 11.7%
- 20 people: 41.1%
- 23 people: 50.7%
- 30 people: 70.6%
- 50 people: 97.0%
- 70 people: 99.9%
- 200 people: 99.9999999999999999999999999998%
- 366 people: 100% (pigeonhole)

Occupancy Problems

- Throw m balls into n bins randomly. We are interested in the following problems:

Occupancy Problems

- Throw m balls into n bins randomly. We are interested in the following problems:
- Can you formulate the birthday problem as a question in balls-in-bins?

Occupancy Problems

- Throw m balls into n bins randomly. We are interested in the following problems:
- Can you formulate the birthday problem as a question in balls-in-bins?
- What is the max number of balls in any bin?

Occupancy Problems

- Throw m balls into n bins randomly. We are interested in the following problems:
- Can you formulate the birthday problem as a question in balls-in-bins?
- What is the max number of balls in any bin?
- What is the number of empty bins?

Occupancy Problems

- Throw m balls into n bins randomly. We are interested in the following problems:
- Can you formulate the birthday problem as a question in balls-in-bins?
- What is the max number of balls in any bin?
- What is the number of empty bins?
- How many balls we have to throw such that all the bins are non empty with reasonable probability?

Occupancy Problems

- Throw m balls into n bins randomly. We are interested in the following problems:
- Can you formulate the birthday problem as a question in balls-in-bins?
- What is the max number of balls in any bin?
- What is the number of empty bins?
- How many balls we have to throw such that all the bins are non empty with reasonable probability?

Occupancy Problems

Previous analysis showed that if there are $m = \Omega(\sqrt{n})$ balls into n bins, then the probability that some bin contains more than 1 ball is high.

Occupancy Problems

Previous analysis showed that if there are $m = \Omega(\sqrt{n})$ balls into n bins, then the probability that some bin contains more than 1 ball is high.

Claim

When $n = m$, if X_i is the number of balls in bin i then $E(X_i) = 1$.

Previous analysis showed that if there are $m = \Omega(\sqrt{n})$ balls into n bins, then the probability that some bin contains more than 1 ball is high.

Claim

When $n = m$, if X_i is the number of balls in bin i then $E(X_i) = 1$.

Class Assignment: (1) Prove the above claim. (2) What is the probability that the first bin has exactly i balls? Show that it is less than $(\frac{e}{i})^i$ by Stirling/Taylor expansion. (3) What is an upperbound on the probability that the j bin has k or more balls in it (call this event $C_j(k)$ so that we are all on the same page)?

Balls in Bins

Theorem

With probability at least $1 - 1/n$*, no bin has more than* $k^* = \lceil (3 \ln n)/\ln \ln n \rceil$ *balls in it.*

Balls in Bins

Theorem

With probability at least $1 - 1/n$, *no bin has more than* $k^* = \lceil (3 \ln n)/\ln \ln n \rceil$ *balls in it.*

Exercise

Show that for $m = n \ln n$ *with probability* $1 - o(1)$ *every bin has* $O(\log n)$ *balls.*

Balls in Bins

Theorem

With probability at least $1 - 1/n$, no bin has more than $k^* = \lceil (3 \ln n)/\ln \ln n \rceil$ balls in it.

Exercise

Show that for $m = n \ln n$ with probability $1 - o(1)$ every bin has $O(\log n)$ balls.

Also look at max value of Binomial distribution.

We can do much better max load, will maybe prove later in class.

Theorem

Suppose that n balls are sequentially placed into n bins in the following manner. For each ball, $d \geq 2$ bins are chosen independently and uniformly at random (with replacement). Each ball is placed in the least full of the d bins at the time of placement, with ties broken randomly. After all the balls are placed, the maximum load of any bin is at most $\frac{\ln \ln n}{\ln d} + O(1)$, with probability at least $1 - o(1/n)$.

Application: Bucketsort

- Bucket sort breaks the $\Omega(n \log n)$ bound for comparison sorting under certain assumptions.

Application: Bucketsort

- Bucket sort breaks the $\Omega(n \log n)$ bound for comparison sorting under certain assumptions.
- Assume we have set of $n = 2^m$ elements to sort, each one is a u.a.r. integer from the range $[0, 2^k)$, where $k \geq m$.

Application: Bucketsort

- Bucket sort breaks the $\Omega(n \log n)$ bound for comparison sorting under certain assumptions.
- Assume we have set of $n = 2^m$ elements to sort, each one is a u.a.r. integer from the range $[0, 2^k)$, where $k \geq m$.
- First place elements into n buckets. The j-th bucket contains elements whose first m binary digits are the number j. Then sort the buckets using any $O(n^2)$ algorithm.

Application: Bucketsort

- Bucket sort breaks the $\Omega(n \log n)$ bound for comparison sorting under certain assumptions.
- Assume we have set of $n = 2^m$ elements to sort, each one is a u.a.r. integer from the range $[0, 2^k)$, where $k \geq m$.
- First place elements into n buckets. The j-th bucket contains elements whose first m binary digits are the number j. Then sort the buckets using any $O(n^2)$ algorithm.
- We can sort in expected running time $O(n)$, expectation over input (since bucketsort is deterministic).

Application: Bucketsort

- Bucket sort breaks the $\Omega(n \log n)$ bound for comparison sorting under certain assumptions.
- Assume we have set of $n = 2^m$ elements to sort, each one is a u.a.r. integer from the range $[0, 2^k)$, where $k \geq m$.
- First place elements into n buckets. The j-th bucket contains elements whose first m binary digits are the number j. Then sort the buckets using any $O(n^2)$ algorithm.
- We can sort in expected running time $O(n)$, expectation over input (since bucketsort is deterministic).
- We will use the fact that for X $Bin(n, p)$ we have $E(X^2) = n(n-1)p^2 + np$.