SAMPLING ON THE FLY

General Framework

- Problem with massive data. Can be read from external memory, but too large to be stored in RAM.

- A natural approach

  Draw a sample (much smaller than the input) storable in RAM.

- Algorithm processes sample and yields good estimates of answer to whole problem.
Simplest kind of sampling is **Uniform Sampling**: every piece of data is equally likely to be picked.

**Advantages**:

- “Coins can be tossed” “blindly” - prior to a pass through the data. So, sample can be extracted in one (quick) pass through the data from external memory.

- Much recent work on things we can do with a uniform sample of fairly small size.

**Drawback**

Much we cannot do: Toy example - find the average of $n$ given real numbers, where there is a lot of cancellation

(i.e., $|\text{average}| \ll \text{average of } |\text{reals}|$.)

2
A central principle: No free lunch.

So, with a small sample size, only “global statistical properties” can be measured. Cannot hope to get “fine structure details”. One reason, then for drawing a small sample and computing only on the sample is that it may take us from impossibly large sizes to manageable ones.

Another motivation If the data (like the constraint matrix of a Linear Program) is expensive to obtain accurately, then in any case, a computation with all the data is error-prone. A sample may introduce the same order of error, but the cost of obtaining just the sampled data instead of the whole may be manageable.
What can we do with uniform sampling?

First Example: Linear Programming with (0,1) variables

\[ LP - 1 \] \quad Ax \leq b \quad 0 \leq x_i \leq 1 \quad i = 1, 2, \ldots n. \\

Q random subset of q variables among \( x_1, x_2, \ldots x_n. \)

\( A^Q, x^Q \) : Take only \( i \in Q. \)

\[ LP - Q \] \quad A^Q x^Q \leq \frac{q}{n} b \quad 0 \leq x_i \leq 1 \quad i \in Q.

Trivial \quad \text{If LP-1 is feasible, then whp, LP-Q is}
\quad \text{“nearly” feasible.} \quad \text{In fact the feasible solution}
\quad \text{restricted to } Q \text{ works for LP-Q whp.}
Converse holds too! (Easy, but, requires duality)

LP-1 infeasible \[ \iff \exists u \geq 0 : (uA)x \leq ub, \quad 0 \leq x_i \leq 1 \quad \text{is infeasible} \]

One inequality

\[ \iff \sum_i (uA)_i^- > ub \]
\[ \implies \sum_{i \in Q} (uA)_i^- > \frac{q}{n} ub - \epsilon \]
\[ \implies (LP - Q) \text{ is “nearly” infeasible.} \]
Quantitative Result for (0,1)-LP

\[ LP - 1 \] \quad Ax \leq b \quad 0 \leq x_i \leq 1 \quad i = 1, 2, \ldots n.

Q random subset of q variables among \( x_1, x_2, \ldots x_n \).

\[ LP - Q \] \quad A^Q x^Q \leq \frac{q}{n} b \quad 0 \leq x_i \leq 1 \quad i \in Q.

(1) If \( LP - 1 \) is feasible, then for 99/100 of the \( Q \)’s, there is an \( x^Q \) with \( 0 \leq x_j \leq 1 \forall j \in Q \) with

\[ A^Q x^Q \leq \frac{q}{n} b + 4\sqrt{q}||A||_\infty 1. \]

(2) If \( LP - 1 \) is infeasible, then for 99/100 of the \( Q \)’s, there is no \( x^Q, 0 \leq x_j \leq 1 \forall j \in Q \) such that

\[ A^Q x^Q \leq \frac{q}{n} b - 4\sqrt{q}||A||_\infty 1. \]

General Flavour:

\( \exists \) a good solution to whole problem \( \implies \exists \) a good solution to random sub-problem. (Easy)

\( \nexists \) a good solution to whole problem \( \implies \nexists \) a good solution to random sub-problem. (Harder)
Beyond Uniform Sampling

Frequency moments of streaming data

Given a “STREAM” $S$ of data consisting of pairs $(i, a)$ of integers.

Examples  NetFlow information - Pair $(i, a)$ logs a message with addresses (source-destination) $i$ containing $a$ packets. Possibly logs of many messages written in by multiple agents in arbitrary order.

Data Bases : $(i, a)$ is an element of a relation.
Estimate “second moment”:

\[ \sum_i \left( \text{total number of packets for address } i \right)^2 \]

More generally, \( p \) th moment - replace 2 by \( p \) above.

\( p = 0 \) distinct number of values of attribute \( i \) (in a database).

\( p = 2 \) and higher - degree of skew of the data..

Dewitt, Naughton, .... - useful in query optimization methods in database applications.

\( a \) may be negative too !!
Streaming Model

We are allowed to make one “leisurely” pass through the data (slowly flowing stream) -

For each bit (or block) read, can do poly(log n) amount of processing (where n is the total number of bits).

Also have RAM of poly(log n).

Thus most data must be seen once and thrown away.

We must sample on the fly - after poly(log n) processing, must decide if current block read should be preserved as part of sample (or “folded into running sample” in some way).

Recent Results \((p =) 0, 1, 2\) moments can all be approximated to relative error \(\epsilon\) in the streaming model.
Random Projections

Want $\sum_i (\text{total number of packets for address } i)^2$.

Let $v_i = \text{total number of packets for address } i$. Really want $\sum_i v_i^2 = |v|^2$.

**Old Theorem**  In a random coordinate system, the first component of $v$ will be almost exactly $|v|/\sqrt{n}$ in absolute value.

This first component $= v \cdot u$ for a random unit length vector $u$.

If we randomly generated $u$ and kept in ($O(n)$ bits needed !!), then we can compute $v \cdot u$ as data streams - on seeing $(i, a)$, just add $u_i a$ to running total $u \cdot v$.

Avoiding $O(n)$ space - use a pseudo-random $u$; just store seed for pseudo-random generator and regenerate $u_i$ each time it is needed !!
Literature

Dewitt, Naughton, Schneider and Seshadri and Haas, Naughton, Seshadri, Stokes: Computation and application of frequency moments in databases.

Henzinger, Raghavan and Rajagopalan: The Streaming model.

Alon, Mathias, Szegedy: Relative error $\epsilon$ streaming algorithms for $p = 0$ (number of distinct values of an attribute) and $p = 2$ using only $O(\log n)$ RAM (all with only non-negative $a$).

Feigenbaum, Sampath Kannan, Strauss, Viswanathan: $p = 1$ when $a$ can be negative. (Useful for comparison of streams.) - Beware Cancellations.

Indyk (pseudo) Random Projections
Many other problems have been tackled in the streaming model.

For example, clustering problems:

Given a census, say, of incomes \((n \text{ real numbers})\), cluster them into \(k\) \((k \ll n)\) clusters. Some work on \(O(1)\) factor approximations for \(k\)--median and \(k\)--center measures - Guha, Mishra, Motwani and O’Callahan.

Open: With one or more passes, can we do more realistic clustering??
BEYOND A SINGLE PASS

Model

Data stored in external memory. We can read entire data by making a quick pass through the data.

Quick Pass - $O(1)$ time per block read. [Not poly($\log n$) as in Streaming model.]

Measure three resources

(i) number of (quick) **PASSES**.

(ii) Extra **TIME** (besides the passes)

(iii) Extra **RAM**

Early work on pass Vs space trade-offs - Munro and Patterson.

Drineas, Kannan
Why this modification to Streaming?

Disk storage has grown rapidly and I/O time has shrunk. So, large data sets can be stored in external memory and read once or twice or a few times.

But, we cannot hope to apply even low polynomial time algorithms on the whole data.
Matrices

Document-Term matrix $A$ of a (large) collection of documents has:

$A_{ij} =$ number of occurrences of term $j$ in document $i$ or a function of that number.

Many Information Retrieval applications based on the document-term matrix.

Generally, a collection of $m$ objects described by $n$ features.

$A_{ij} =$ “intensity” of feature $j$ in object $i$
Similarity of two objects - dot product.

**Problem**  Process a large $A$, store “summary” so that later, for a query object (an $n$–vector) $x$, we can find (approximately) the similarity of $x$ to each original object.

**No free lunch**  - cannot hope to get it right if only 1 or 2 or very few of the original objects are mildly similar to the query.
Given an \( m \times n \) matrix \( A \), can we find an approximation \( A' \) to \( A \) (also an \( m \times n \) matrix) so that

(i) \( A' \) is storable in much less space than \( A \).

(ii) \( A' \) can be found from \( A \) in few passes of \( A \) from external memory.

(iii) \( A' \approx A \).

Yes; indeed enough to store \( O(1) \) randomly picked columns and \( O(1) \) randomly picked rows of \( A \), but picked according to a certain probability distribution. Also, the columns and rows can be picked in two passes through \( A \).

– Drineas, Kannan, earlier joint work with Frieze, Vempala, Vinay
$A$ any $m \times n$ matrix. $C$ an $m \times s$ matrix of $s$ random columns of $A$ picked in i.i.d. trials; in each trial,

\[
\text{Prob of picking column } j = \frac{\sum_i A_{ij}^2}{\sum_{k,l} A_{kl}^2}.
\]

[Probabilities are proportional to length squared of the columns.]

Similarly $R = s \times n$ from $s$ random rows of $A$. From $C, R$, we can compute a $s \times s$ matrix $U$ so that

\[
E \left( \max_{x:|x|=1} |(A - CUR)x|^2 \right) \leq \frac{4}{\sqrt{s}} \sum_{ij} A_{ij}^2.
\]

Rewritten $E \left( \|A - CUR\|_2^2 \right) \leq \frac{4}{\sqrt{s}} \|A\|_F^2$.

Also, $E \left( \|(A - CUR)\|_F^2 \right) \leq \frac{4}{s^{1/4}} \|A\|_F^2 + \text{Best we can with a rank } \sqrt{s} \text{ matrix.}$
\[
\left( \begin{array}{cc}
A & \\
\end{array} \right) \approx \\
\left( \begin{array}{c}
C \\
\end{array} \right) \cdot \left( \begin{array}{c}
U \\
\end{array} \right) \cdot \left( \begin{array}{c}
R \\
\end{array} \right)
\]
Matrix Reconstruction

$m$ users and $n$ products. $A_{ij}$ measures the preference of user $i$ for product $j$.

Suppose we have observed some entries of the matrix. Can we infer the other entries? [So, having observed some market behaviour, we want to recommend to users what they would like.]

[Recommendations Systems / Collaborative filtering]

Azar, Fiat, Karlin, McSherry and Saia

Achlioptas and McSherry

Drineas, Kerenidis and Raghavan
Achlioptas and McSherry’s algorithm:

$p$ probability. Independently for each entry $A_{ij}$ of matrix, replace it with $A_{ij}/p$ with probability (w.p) $p$ and 0 with probability $1 - p$. So, number of non-zero entries reduced by a factor of $p$.

$$\tilde{A}_{ij} = \begin{cases} 
0 & \text{w.p.} 1 - p \\
A_{ij}/p & \text{w.p.} p 
\end{cases}$$

$$\begin{pmatrix}
5 & 3 & 3 & -2 & -7 & 8 & 9 \\
1 & 2 & 2 & -17 & 1 & -8 & 9 \\
21 & 41 & 22 & -2 & 0 & 0 & 0
\end{pmatrix} \rightarrow
\begin{pmatrix}
10 & 6 & 0 & 0 & -14 & 16 & 0 \\
2 & 4 & 0 & 0 & 0 & -16 & 18 \\
0 & 0 & 44 & 0 & 0 & 0 & 0
\end{pmatrix}$$

If $|A_{ij}| \leq 1$, WHP, $\|A - \tilde{A}\|_2$ is small.