SAMPLING IN MASSIVE MATRICES, TENSORS

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 - all 0, except one massively big.

But, here running sum works.

Remark 1 In general, routine statistics argument : can estimate one fixed quantity by taking a sample.

Remark 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".

What can we do with uniform sampling ?

Example 1 There are n data objects. (n large). For each pair (i, j) of objects, we are given whether i, j are similar. Want to test the hypothesis "the objects can be divided roughly into k clusters", where each cluster contains similar objects.

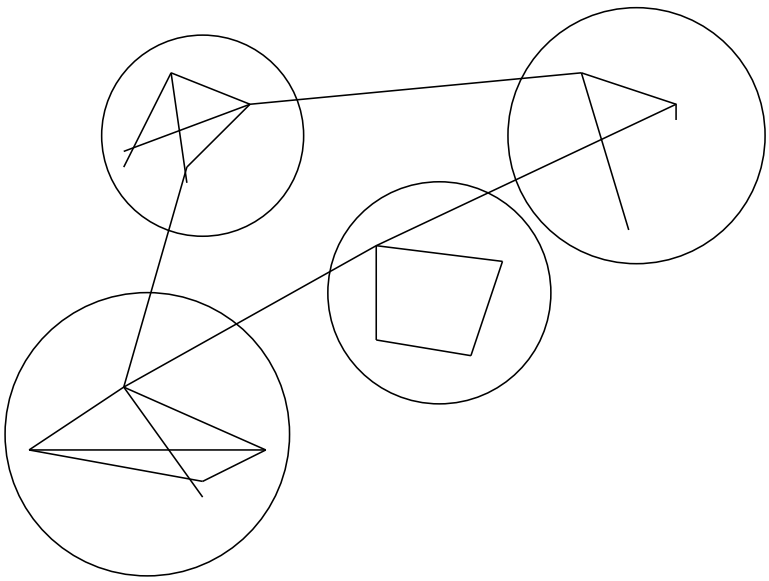
Also want a representative object from each cluster.

Often $k \ll n$, indeed, we assume $k \in O(1)$.

One formulation : what is the minimum number of new similarities we need to throw in so that there are k clusters with EVERY pair of objects inside each cluster being similar ?

Call this number ANS.

– Really a graph clique or coloring problem.



Result 1

We can find ANS to within $\pm\epsilon n^2$ given a random subset of $O(1/\epsilon^4)$ objects and all their pairwise similarities.

No good if ANS$\ll n^2$. (No free lunch).

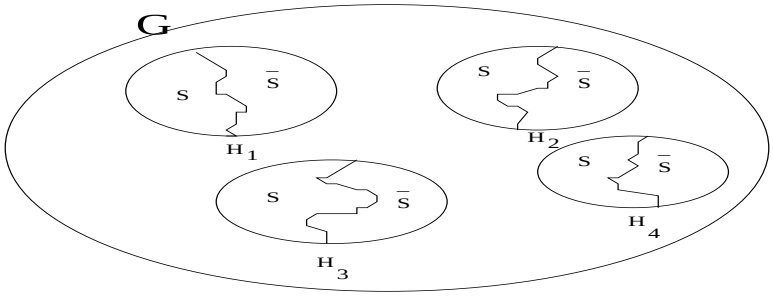
Easy Part : If there is a good clustering of the n objects, this induces a good clustering of sampled objects. – Traditional Statistical sampling arguments

Hard Part : Good clusterings of random subsets yield good clustering of the whole.

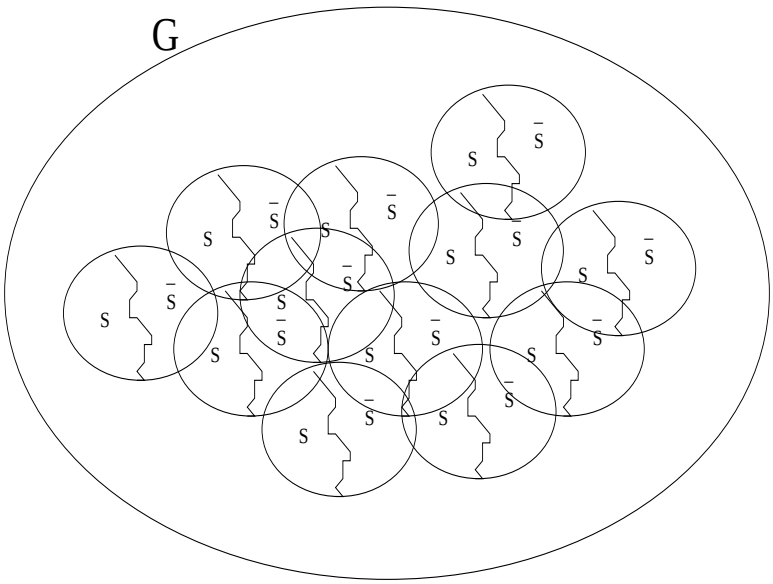
Sample is not overly optimistic - because it can be optimistic in one of exponentially many ways.

This example is a very special case of a general result.

Easy Part A good clustering on the whole induces a good clustering on random sub-parts. An illustration for $k = 2$ - clustering into $S, \bar{S}$.



Hard Part How do we stich together clusterings in random sub-parts into a clustering of the whole ? In different random parts, the same object may be put into S or $\bar{S}$.



CONSTRAINT SATISFACTION PROBLEMS

n Boolean variables - $x_1, x_2, \ldots x_n$.

r constant.

m constraints given, each involving r literals.

(Many global variables. "Local" constraints, each involving only a fixed number).

Find a truth setting of $x_1, x_2, \ldots x_n$ which satisfies as many of the constraints as possible. Call this answer ANS.

The clustering problem above is a special cases of CSP with $r = 2$. So are many graph and Boolean problems.

Example Satisfy as many clauses as possible among :

$(x_1 + \bar{x}_2 + \bar{x}_3)(x_4 + \bar{x}_1 + x_2)(x_7 + \bar{x}_3 + x_2) \ldots$

Result 2

Given a uniform random subset R of $x_1, x_2, \ldots x_n$ of size c/ϵ^4 and all constraints involving only variables in R, we can find ANS to within $\pm\epsilon n^r$.

Answer to "induced" "sub-problem" on a random subset of variables is a good estimate of answer to whole problem.

Indeed, also a truth setting of all variables $x_1, x_2, \ldots x_n$ attaining this value can be constructed in $O(n)$ time from sample.

Literature

Arora, Karger, Karpinski: In poly time, we can approximate CSP to **additive error** ϵn^r.

[Can also be phrased as "we can solve dense instances with relative error ϵ.]

Goldreich, Goldwasser, Ron
r=2
First "constant" time algorithms for CSP problems for $r = 2$ with additive error ϵn^2. Good polynomial bounds (in $1/\epsilon$) on the size of the random sub-problem needed ("sample complexity") using structure of individual CSP problems..... Property Testing .....

Frieze, Kannan Poly time algorithms for all CSP problems (with fixed r) achieving additive error ϵn^r using Linear Algebra methods.

Result 2 is due (independently to)

Alon, de la Vega, Kannan, Karpinski using Linear Algebra methods and

Andersson and Engebretsen using combinatorial methods.

Beyond Uniform Sampling - Streaming

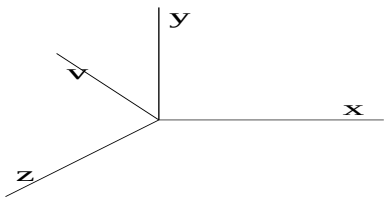
Random Projections

Let $v_1, v_2, \ldots v_n$ be n reals specified by a stream of instructions of the form

Increment / Decrease v_i by x

Want say, $|v|^2 = \sum_i v_i^2$.

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 "Increase v_i by 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 !!

Alon, Mathias, Szegedy; Feigenbaum, S. Kannan, Strauss, Vishwanathan; Piotr Indyk.

Random Projections An excellent tool for dimension reduction. Applications to Nearest Neighbor problems, Learning Alg's.....

Spectral Projections Another (entirely different) dimension reduction technique - to come

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.

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.

Frequency moment problems above can be viewed as dealing with one $n-$ vector (or two if we are comparing two streams).

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; Mahoney

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

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

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

Also, $E\left(||(A - CUR)||_F^2\right) \leq \frac{4}{s^{1/4}} ||A||_F^2 +$ Best we can

with a rank $\sqrt{s}$ matrix.

$$\left(\quad A \quad \right) \approx$$

$$\left(\quad C \quad \right) \cdot \left(\quad U \quad \right) \cdot \left(\quad R \quad \right)$$

What is wrong with the length-squared distribution ?

An Example : A has the first $m-1$ rows all equal and the last row orthogonal to them; all rows are of length 1.

Best rank 2 approximation : A itself. Error=0.

Repeated sampling only yields the first vector. Error $\not\leq O($ best error $)$!!

Two Issues : Relative Error Get a low-rank approximation $\widehat{A}$ to A so that

$$||A - \widehat{A}||_F \leq (1 + \epsilon)||A - A_k||_F.$$

(Recall : A_k best rank k approx to A.)

"Interpolative" approximation Get an $\widehat{A}$ which is in the span of at most s (s small) rows of A.

[For a "patient-gene" matrix, get a representative sample of patients/genes, not a linear combination of patients/genes.]

Deshpande, Rademacher, Vempala, Wong; Drineas, Mahoney, Muthukrishnan; Sarlos; Har-Peled; Martinson, Rokhlin and Tygert all address these questions.

Approximate Least Squares (Best Fit) :

$$\text{Min}|Ax - b|,$$

where A is $n \times d$, $n \geq d$.

Relative error allowed $= \epsilon$.

If A has one column equal to b, rest of the columns ortho to b and large, length-squared sampling will not achieve relative-error best fit.

Sarlos, based on Drineas, Mahoney, Muthukrishnan : Alg. with time bound :

$$O\left(nd \log n + \frac{d^3 \log d}{\epsilon^2}\right).$$

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

de la Vega, Kannan, Karpinski, Vempala : A general class of CSP including dense and metric problems can be approximated in poly time :

"Core Dense" problems : For example, a core of n^{α} vertices connected together by a dense graph, with n^{β} (weak) connections to the rest of the graph and $O(1)$ connections (very weak) among the non-core vertices.

Interesting question : Can preferential attachment type graphs (modelling the web) be treated in this fashion ??