ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

Sketching of Big Data

[1] M. Parameswari
Department of Computer Science and Engineering, PG scholar,
Francis Xavier Engineering College, Tirunelveli, Tamilnadu.

Abstract: **- The Human beings create everything but the most innovative and creative one is the internet. The internet has allowed for very less transfer of data and information in a fraction of seconds. The next level of the internet with human innovation to increase the communication, data speed and a large amount of data gathering. The solution for a large amount of data gathering is big data. Big data is the very large amount of data it does not possible to fit in single machine main memory. The need for big data analysis in increased day by day. In this paper analysis and evaluate the sketching and streaming of big data algorithms. The advantages of sketching include less memory consumption, faster algorithms, and reduced bandwidth requirements in distributed computing environments. Now a day's sketching of big data is an essential one.**

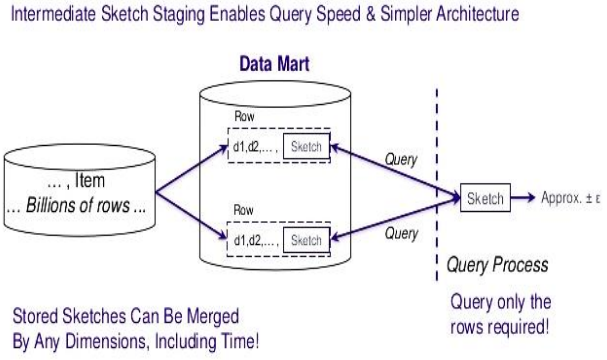
Keywords: **Human beings, big data, sketching, internet.**

I. INTRODUCTION

Big data is the very large amount of data that it does not fit in the single main memory of a single machine. To process big data analysis by space and cost-efficient algorithms arises in the computer industry and many other areas. This survey discussed mathematically calculation models for developing such algorithms, as well as some verifiable restriction of algorithms. In this analysis using following techniques. First one is Dimensionality reduction, it is common techniques and hopelessness results for data dimension reduction while still maintain the graphical structure. The second one is Numerical linear algebra, in this Algorithms used for big matrices. Regression, matrix calculation etc. The third one Compressed sensing, it is used to recover sparse signals approximately based on few linear measurements. The final one is Sparse Fourier Transform; it is approximately rapid algorithms for computing the Fourier Transform of signals.

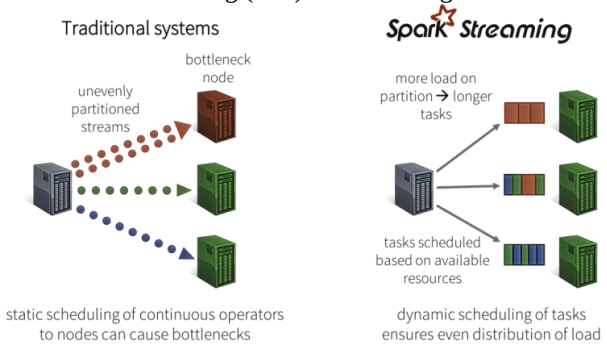
Sketching:

Sketching is the compression P(a) of some data set ' a ' that allows us to query S(a). There are some things we might want when we are designing such a P(a). Perhaps, we want S to take on two arguments instead of one, as is the case when we want to compute S(a; b) from P(a) and P(b). Often, we want P(a) to be composable. In other words, if a = $a_1a_2a_3$: : : an, we want to be able to compute P(aan+1) using just P(a) and an+1.



Streaming :

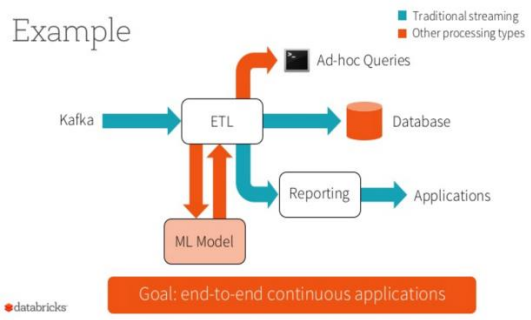
If a data set is very large, it may not be possible to store all the data and information in a single memory. A stream is a sequence of data attributes that come in bit by bit, like items on a conveyor belt in manufacturing factory. Streaming is the perform of processing these data essentials on they as they appear. The ambition of streaming is to answers queries within the restrictions of sublinear memory. Steaming has a following approaches Continuous monitoring (CM) and Chaining.



ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

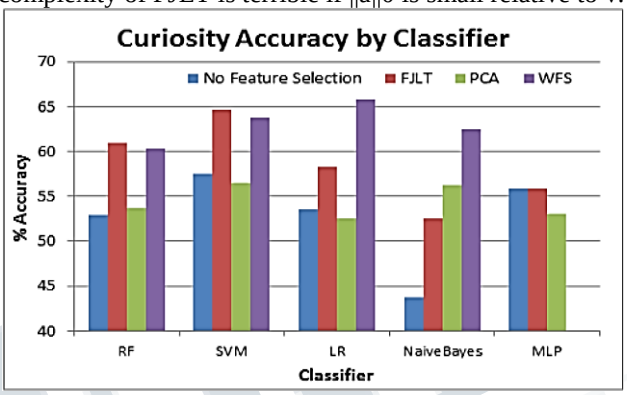
Continuous monitoring :
continuous monitoring (CM), we think about data streams comes nonstop, more involved querying. An an example, consider a router that sees a flow of IP address and at every point in time we are involved in what the deep hitters are, how twisted the allocation is, or whether we can spot tendency and anomalies in the traffic.



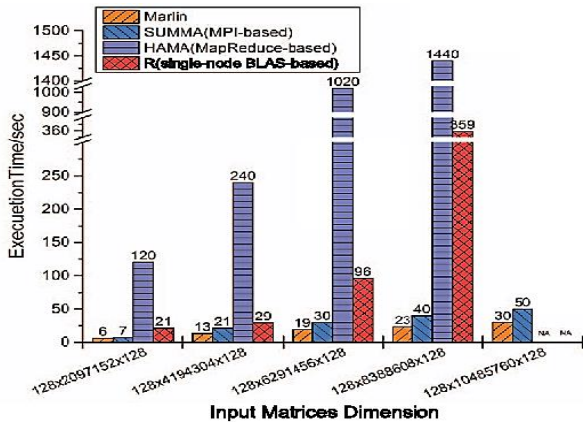
Chaining:
Chaining is a method for computing bounds on EsupcϵC C that leverages correlations across the C. For the lecture today, every C looks like <σ, a> for some a. We want to bound EsupaϵT <σ,a> def = r(T). The way of thinking is that if you bounce this and the unhelpful of this, we have a guarantee. If you have two vectors a and b where |a-b| is close to 0, then <σ,a> is close to <σ,b>. Bounding r(t) We are going to discover four ways of bounding r(T) that are gradually tighter. first one is Union Bound, the second one is - ϵ-net argument, third is Dudley's inequality, the final bound approach is - Last approach (not full proof).

Dimensionality Reduction :
There are lots of instances in the actual world where we meet data sets with high dimensionality. The example brought up in class was the difficulty of spam filtering. A simple method to spam detection is the bag-of-words model, where each email can be represented as a high dimensional vectors whose indices come from a dictionary of words and the value at each index is 1 the number of incidences of the equivalent word. In this conditions like these where we have a high dimensional computational geometry trouble, we may want to decrease the number of dimensions in pre-processing while preserving the estimated geometric structure. Typically we have some high-dimensional computational geometry trouble, and we use JL to hurry up our algorithm in two steps: (1) apply a JL map π to reduce the trouble to low dimension n, then (2) solve the lower-dimensional trouble. As n is made smaller, typically (2) becomes quicker However, ideally, we would also like the step (1) to be as quick as possible. In this section, we investigate two approaches to speed up the computation of πx.

Fast JL transform (FJLT):
We said that the time complexity is O(vlog v + n3), where v is the dimension of the vector a and n is the number of rows of the transform matrix π. However, in practice, the vector ' a ' is often a sparse vector, and we would expect that the time complexity for the transform a →πa is O(n||a||0), where ||a||0 =| {i: ai ≠ 0}|, and the time complexity of FJLT is terrible if ||a||0 is small relative to v.



Large-Scale Matrix Computations :
Many of the troubles that contract with big data end up connecting some large-scale matrix computation. Examples of large-scale matrix computations include regression & principal component analysis (PCA). For a least square regression, we estimate that b = f(a) for some vector a of illuminating variables. We want to learn f, so we think that f(a) = <β,a> for some β. Here, <•,•> indicates the internal product of two vectors. We collect data points (ai; bi) and assume that bi = f(ai) + ϵ i for some small clatter or error term ϵ i, and from these data points, it is possible to recover a vector of coefficients β = β LS that minimizes
$\sum_{i=0}^{n}(b_i - < \beta, a_i >)^2 = \|A\beta - b\|_2^2$, by computing $\beta^{LS} = (A^TA)^{-1}A^TB$. Here, $\|\cdot\|_2$ denotes the L^2 norm

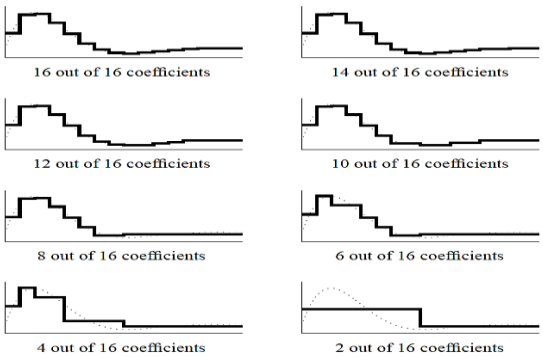


ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

Compressed Sensing :

As we have seen, many of the troubles faced in the actual world engage linear signals. Sometimes, when we modify the basis we use to explain this linear signal, it becomes sparse. This permits us to accumulate remote less samples than otherwise necessary. Compressed sensing also engage the process of finding a basis in which a linear signal is sparse, taking a small number of linear measurements, and later roughly rebuild the original signal from the measurements.



An example from Wavelets for Computer Graphics: A Primer [1]

Sparse Fourier Transform

For a little predetermined integer n, let SFn = [SFjk] be the matrix hand over by the terms SFjk = e-2πijk/n and let x = (x1; x2; : : : ; xn) be a series of compound numbers. The discrete Fourier transform (DFT) sends x to SFnx. In 1942, Danielson and Lanczos published an algorithm that computed the DFT in O(n log n) floating-point operations (FLOPS) [DL42]. In 1965, Cooley & Tukey published a more general version of the fast Fourier transform (FFT) [CT65]. Cooley and Tukey are often credited with the discovery of the modern generic FFT algorithm, but in an unpublished manuscript from 1805, Gauss had already found a similar algorithm and used it to interpolate the orbits of Pallas and Juno. For more information, please see refer to the Theoria Interpolationis Methodo Nova Tractata. The sparse Fourier transform (SFT) is an algorithm that computes the Fourier transform in time O(k log n) if the output of the DFT is exactly k-sparse. If the output of the DFT is approximately k-sparse, SFT can approximate the Fourier transform in time that is approximately but a little worse than O(k log n).

II. CONCLUSION

Big data is a very large amount of data set , it does not possible to store a single RAM so we using some sketching and streaming algorithm , advantages of the sketching and streaming algorithm is to reduce the memory usage and easy to access . in this discussion we clearly known about seven techniques of sketching algorithms.

REFERENCES

[1] Noga Alon, Yossi Matias, Mario Szegedy. The Space Complexity of Approximating the Fre-quency Moments. Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC), pp. 20{29, 1996.

[2] Kasper Larsen, Jelani Nelson. Optimality of the Johnson-Lindenstrauss Lemma. Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS) 2017.

[3] [CT65] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. Math. Comp., 19(90):297{301, 4 1965.

[4] [DL42] Gordon C. Danielson and Cornelius Lanczos. Some improvements in practical fourier analysis and their application to x-ray scattering from liquids. J. Franklin Inst., 233(4):365{ 380, 4 1942.

[5] [Mor78] Robert Morris. Counting large numbers of events in small registers. Commun. ACM, 21(10):840{842, 10 1978.8

[6] [AC09] Nir Ailon and Bernard Chazelle. The fast Johnson{Lindenstrauss transform and ap- proximate nearest neighbors. SIAM J. Comput., 39(1):302{322, 2009.

[7] [BDF+11] Jean Bourgain, Stephen Dilworth, Kevin Ford, Sergei Konyagin, and Denka Kutzarova. Explicit constructions of RIP matrices and related problems. Duke Mathematical Jour- nal, 159(1):145{185, 2011.

[8] [Bou14] Jean Bourgain. An improved estimate in the restricted isometry problem. Geometric Aspects of Functional Analysis, 2116:65{70, 2014.

[9] [CT06] Emmanuel J. Cand_es and Terence Tao. Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inform. Theory, 52(12):5406{ 5425, 2006.

[10] [HR16] Ishay Haviv and Oded Regev. The restricted isometry property of subsampled fourier matrices. In Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 288{297, 2016.

[11] [KW11] Felix Krahmer and RachelWard. New and improved Johnson-Lindenstrauss embeddings via the Restricted Isometry Property. SIAM J. Math. Anal., 43(3):1269{1281, 2011.

[12] [NPW14] Jelani Nelson, Eric Price, and Mary Wootters. New constructions of RIP matrices with fast multiplication and fewer rows. In Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 1515{1528, January 2014.

[13] [RV08] Mark Rudelson and Roman Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. Comm. Pure Appl. Math., 61(8):1025{1045, 2008.

[14] [Ach01] Dimitris Achlioptas. Database-friendly random projections. In Proceedings of the twen- tieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 274{281. ACM, 2001.6

[15] [BOR10] Vladimir Braverman, Rafail Ostrovsky, and Yuval Rabani. Rademacher chaos, random eulerian graphs and the sparse johnson-lindenstrauss transform. arXiv preprint arXiv:1011.2590, 2010.

[16] [DKS10] Anirban Dasgupta, Ravi Kumar, and Tam_as Sarl_os. A sparse johnson: Lindenstrauss transform. In Proceedings of the forty-second ACM symposium on Theory of computing, pages 341{350. ACM, 2010.

[17] [DlPG12] Victor De la Pena and Evarist Gin_e. Decoupling: from dependence to independence. Springer Science & Business Media, 2012.