Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100636

Bayesian Method for Source Local Deduplication in Cloud Backup Services

P. Neelaveni and M. Vijayalakshmi*

Department of Information Science and Technology, Anna University, Chennai, India.

Received: 1 Jul. 2016, Revised: 17 Sep. 2016, Accepted: 19 Sep. 2016
Published online: 1 Nov. 2016

Abstract: Data deduplication technique is widely deployed in cloud backup storage system to reduce storage space and to minimize the transmission of redundant data for proper utilization of network bandwidth. During cloud backup service, redundancy of typical backup data dominated heavily by duplicate chunks. The intrinsic drawback of this system is detecting the similar chunks.The storage server consists of large volume of chunks, making the duplicate detection process much more complicated which decreases deduplication efficiency and increases deduplication overhead. In this paper we propose Bayesian method for source local deduplication for finding out duplicate chunks. For finding chunk similarity, the learning based similarity metrics are developed. The data features are used to train Bayesian system. Our experimental results shows that precision, recall and F measure values are high compared to SVM and GP. Due to these high values the proposed Bayesian method increases deduplication efficiency and reduces deduplication overhead. Therefore the proposed Bayesian method yields better performance than Support Vector Machine Model and Genetic approach.

Keywords: Deduplication, cloud backup service, Bayesian method , duplicate detection.

1 Introduction

Cloud storage has a capability to deliver virtualized storage on demand over a network. It gains an extensive focal point on development and explore both in the industry community [1] and research area. Cloud storage refers to scalable and elastic storage potentials that are delivered as a service using Internet technologies with elastic provisioning. In recent years, cloud-based storage services [2] such as Dropbox, Google DriveApple, iCloud,JustCloud, Mozy and Microsoft skydive competitively provides easy to access, secure, reliable, and low cost remote storage spaces for file-sharing, document suites, and online-backup services for their users. These cloud based storage providers are popular due to easy data access. Cloud backup service has become a cost-effective solution for data protection in cloud storage system. Traditionally, dedicated external drivers were utilized for backup operations. These drivers are not efficient and expensive for the users. Now data backup [3] is emerged to be attractive application to the cloud storage providers because cloud clients can manage their data easily without any difficulty as they need not aware about maintaining the backup infrastructure. This is potential because the centralized cloud management has formed a competence and cost intonation point. It procures effective offsite storage and it has been always a significant concern for cloud data backup operations. The basic feature in provisioning cloud backup is quality of service [2] which is based on the management of handling the large amount of network bandwidth requirements from a user to cloud storages and techniques utilized effectively to reduce the storage space. Thus, top most popular cloud-based backup storage services use data deduplication techniques [3] at a source or client site to save the network bandwidth and storage space, which in turn accelerate the data upload process. Data deduplication technology identifies duplicate data, eliminate redundancy and reduce the need to transfer or store the data in the overall capacity. Deduplication is an effective technique to optimize the utilization of storage space. Data deduplication can greatly reduce the amount of data, thereby reducing energy consumption and reduce network bandwidth in cloud data centers [4, 5].

The deduplication module partitions a file into chunks, generates the respective summary information, which we call a fingerprint, and looks up fingerprint table to determine if the respective chunk already exists. If it does not exist, the fingerprint value is inserted into

* Corresponding author e-mail: srirang.neels@gmail.com

fingerprint table. Chunking and fingerprint management is the key technical constituents which governs the overall deduplication performance. There are a number of ways for chunking, e.g., variable size chunking, fixed size chunking, or mixture of both. Depending on the location where redundant data is eliminated, deduplication can be categorized [6] into two basic approaches A) In the target-based approach, deduplication is performed in the destination storage system. The client is not having knowledge about the deduplication strategies. This method have the advantage of increasing storage utilization, but does not save bandwidth. B) In Source based deduplication, elimination of duplicate data is performed close to where data is created, rather than where data is stored as in the case of target deduplication.

The Source deduplication approach works on the client machine before it is transmitted specifically, the client software communicates with the backup server (by sending hash signatures) to check for the existence of files or chunks. Duplicates are replaced by pointers and the actual duplicate data is never sent over the network. Further Source de-duplication method [4] is classified based on different deduplication granularities as 1) source local chunk-level de-duplication 2) source global chunk-level de-duplication [6,7]. In the local chunk level, the redundant data chunks are removed before sending them to the remote backup destination within the same client. In the global chunk level, the duplicate chunks are removed globally across different clients. Duplication detection of chunks at source level is crucial component in the deduplication process. During Backup operation, when local deduplication is performed, redundancy of typical backup data is conquered largely by duplicate chunks. however, storage server consist of huge volume of chunks, making the duplicate detection process much more complicated which increases deduplication computational overhead and decreases deduplication efficiency. Recognizing similar Chunking mechanisms need to be devised to effectively exploit the cloud backup storage space.

In this work, we dedicate our efforts in reducing the performance overhead in finding the duplicate chunks. We propose Bayesian method for finding out duplicate chunks which increases deduplication efficiency and reduces computational overhead through precision, recall and F-Measure values. The Proposed Bayesian method outperforms an existing state of-the-art method found in the literature which is proved through our experimental results and comparative analysis. The rest of the paper is organized as follows. In Section II we discuss related work for duplicate detection. The Theoretical and probabilistic model is discussed in section III. The system architecture is explained in Section IV. Section V evaluates Bayesian method through experiments driven by real-world datasets. Section VI concludes the paper.

2 Related Work

In Duplicate detection process, record matching is a basic approach for determining duplicate data, also known as merge-purge, data deduplication and instance identification. It identifies whether different files consists of same data. Duplicate files detection has become very essential as more and more volume of data to be backed up in the Cloud storage systems by local or global deduplication system. The methods to solve record matching problem can be broadly classified into two categories as Probabilistic models and supervised or semisupervised learning [8] based on learning and training data to match files. Approaches such as Rule-based and Distance-based techniques [9] that rely on domain knowledge or distance metrics to match records. In the Probabilistic-based technique to find similar data, a maximum likelihood estimate is computed which is used to determine whether record pair is matching or non-matching. Unsupervised Expectation Maximization (EM) algorithm [10] can be used when training data is not available. The EM algorithm needs about details regarding data to calculate maximum likelihood estimate. However, the performance of EM algorithm is good when more than duplicates in the dataset, matching data are segregated from non-matching dataset, the typographical error rate is low, adequate attributes to compensate for errors and the conditional independence assumption results in good classification performance [11].

The Rule-based approach [12] is also a potential method since it produces high accuracy in finding duplicate data. However, the high accuracy is obtained through significant effort and time of an expert to precisely devise a matching ruleset. It also requires an analysis of the dataset to give a better idea to the expert how to fine-tune the rules. Designing efficient matching rule set and analysis of the dataset without human intervention is required due to the reasons as an expert may not be available all the time, the dataset may be private and two datasets with similar domains may behave drastically different. Therefore, manually the ruleset constructed for the first dataset may not be applicable to the second dataset.

For large volume of dataset in cloud, record matching system must produce techniques for matching process and through which it must increase the efficiency of the duplicate detection system. To maintain accuracy of the system, subspaces creation method is utilized to decrease the number of candidate datasets. For example, blocking method [12] and the sorted neighborhood (SN) method, employ candidate keys to sort the dataset. Then a block or window is applied to restrict the number of candidate data sets. The group of dataset attributes are involved in creating these candidate keys. Accordingly, it is crucial to select the suitable attributes to detect a proper duplicate record.

Support Vector Machine [13] for deduplication procedure, the similarity function, which are used Dice

coefficient, DamerauLevenshtein distance, Tversky index for similarity measurement. Using these similarity function, testing is enforced whether data record is duplicate or not. A set of data generated from several similarity measures are used as the input to the system. The training phase and the testing phase are two processes which distinguish the proposed deduplication technique. The deduplication efficiency is low for large volume of data and it is not scalable.

A genetic programming [14] approach to implement deduplication that clusters a number of dissimilar pieces of information extracted from the data content to determine a deduplication function. It is able to distinguish whether two entries in a storage are similar or not. When the committee majority voting is not enough to predict the class of the data pairs, a user is called to solve the conflict. The method was applied to three datasets and compared with supervised GP based deduplication strategy. Results show that quality of the deduplication is obtained while reducing the number of labeled examples needed. The method based on GP for the data deduplication task is used to find record-level similarity functions that combine single-attribute similarity functions, aiming to improve the identification of duplicate records and, at the same time, avoiding errors.

None of the above mentioned works actually addressed the issue of improving the precision, recall and F- measure value for very large volume of storage datasets and these methods are not scalable. We propose Bayesian method for finding out duplicate chunks which increases precision, recall and F-Measure values which are more suitable for large scale cloud backup storage data.

3 Bayesian method for Source Local Chunk Duplicate Detection

We use A and B to denote the chunks that we want to match. In the duplicate detection problem, each chunk (α,β) where $\alpha \in A$ and $\beta \in B$ is assigned to one of the two classes M and N. The class M contains the chunks that represent the same data (match) belongs to single client and the class N contains the record pairs that represent two different data (nonmatch or differ) that for the given client. We represent each (α,β) as a random vector x= $[x_1, x_2, \ldots x_n]$.

3.1 Bayes Decision Rule for finding chunk similarity

Let x be a comparison vector, arbitrarily drawn from the comparison space that corresponds to the chunk pair (α,β). The main objective is to determine whether $(\alpha,\beta) \in$ M or $(\alpha,\beta) \in$ N. A decision rule, based on probabilities, can be written as follows:

$$(\alpha,\beta) = \begin{Bmatrix} M & if\ p(M/x) \geq p(N/x) \\ N & otherwise \end{Bmatrix} \quad (1)$$

This decision rule states that, if the probability of the match class M, given the comparison vector x, is greater than the probability of the nonmatch class N, then x is classified to M, and vice versa. By using the Bayes theorem, the previous decision rule may be expressed as:

$$(\alpha,\beta) \in \left\{ M \quad if q(x) = \frac{p(x/M)}{p(x/N)} \geq \frac{p(M)}{p(N)} \right\} \quad (2)$$

The ratio q(x) is called likelihood ratio.The ratio $\frac{p(M)}{p(N)}$ denotes the threshold value of δ the likelihood ratio for the decision. The decision rule in eq 2 is known as bayes test.

Let δ represents a threshold value. It can be selected as random variable or a fixed value. The input to a decision rule is the comparison vector x that assigns x to M or to N. The main assumption is that x is a random vector whose density function is different for each of the two classes. Then, if the density function for each class is known, the duplicate detection problem becomes a Bayesian inference problem. Therefore probability density function is calculated in the following section.

3.2 Probability Density Function

The random variable e is generated by a normal probability distribution. A Normal distribution can be absolutely characterized by its mean and its standard deviation σ

$$p(x_g) \equiv \quad lim \quad \varepsilon \to 0 \frac{1}{\varepsilon} p(x_o \leq \ x \ \leq \ x_o + \varepsilon) \quad (3)$$

To demonstrate Bays rule for duplicate detection, the hypothesis (h) is set as match (M) or differ (N). The maximum likelihood hypothesis must be obtained to get the least-squared error hypothesis. This can be shown by deriving the maximum likelihood hypothesis as shown in eq 4.

$$h_{M_L} = ^{argmax}\ h \in H \ \ p(D/h) \quad (4)$$

where p refers to the probability density. It is assumed a fixed set of instances $(x_1 \ldots x_m)$ and therefore the data D considered to be the corresponding sequence of target values $D = (d_1 \ldots d_m)$. Here $d_i = f(x_i) + e_i$. These are mutually independent given h, we can write $p(d/h)$ as the product of the various $p(d_i/h)$.

$$h_{M_L} = ^{argmax}\ h \in H \ \prod_{i-1}^{m} p(D_i/h) \quad (5)$$

Given that the noise ℓ_i obeys a normal distribution with zero mean and unknown variance σ^2, each d_i must also obey a normal distribution with variance σ^2 centered around the true target value $f(x_i)$ rather than zero.

Therefore $p(d_i/h)$ can be written as a Normal distribution with variance σ^2 and mean $\mu = f(x_i)$. Let us write the formula for this Normal distribution to describe $p(d_i/h)$, beginning with the general formula for a normal distribution and substituting the appropriate μ and σ^2. Because we are writing the expression for the probability of d_i given that h is the correct description of the target function f, we will also substitute $\mu = f(x_i) = h(x_i)$ yielding,

$$h_{M_L} = ^{argmax}\ h \in H \ \prod_{i-1}^{m} \frac{1}{\sqrt{2\prod\sigma^2}} \ell^{-\frac{1}{2\sigma^2}(d_i-\mu)^2} \quad (6)$$

$$h_{M_L} = ^{argmax}\ h \in H \ \prod_{i-1}^{m} \frac{1}{\sqrt{2\prod\sigma^2}} \ell^{-\frac{1}{2\sigma^2}(d_i-h(x_i))^2} \quad (7)$$

Now transformation is applied which is common in maximum likelihood calculations. Instead of maximizing the above complicated expression we can maximize its (less complicated)logarithm

$$h_{M_L} = ^{argmax}\ h \in H \ \sum_{i-1}^{m} -\frac{1}{\sqrt{2\prod\sigma^2}} - \frac{1}{2\sigma_2}(d_i-h(x_i))^2 \quad (8)$$

The first term in this expression is a constant independent of h, and can therefore be discarded, which gives

$$h_{M_L} = ^{argmax}\ h \in H \ \sum_{i-1}^{m} -\frac{1}{2\sigma_2}(d_i-h(x_i))^2 \quad (9)$$

Minimizing the corresponding positive quantity is equivalent to maximizing this negative quantity

$$h_{M_L} = ^{argmax}\ h \in H \ \sum_{i-1}^{m} -\frac{1}{2\sigma_2}(d_i-h(x_i))^2 \quad (10)$$

Finally, Constants are discarded are independent of h.

4 Layered Architecture of Bayesian Method

The proposed Deduplication layered system consists of three levels of layers named as Chunking layer, Bayesian layer and Storage layer as shown in the figure 1.

Cloud client is a end user who inputs the files for cloud storage system and these files to be backed up. File Agent is a functional module that provides a functional interface (file backup/restore) to users. It is responsible for gathering datasets and sending or restoring them to and from underlying layer.

4.1 Chunking Layer

Chunking layer consist of 2 components called chunker and file recipe. Content store divides the file into variable sized chunks. Secure Hash Algorithm (SHA-1) finds the hash value of each chunk. The second component file recipe is maintained to construct the file when it is read. File recipe contains the sequence of ChunkID which constitutes that file. Each chunk is checked for duplicates against a set of chunk indices maintained at the chunk store. Chunk index is the metadata that includes chunkID and the location of actual chunks in storage.

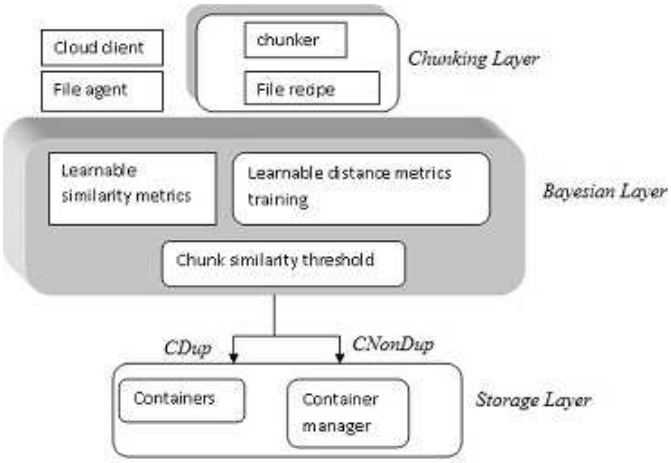

Fig. 1: Layered Architecture of Bayesian method for Source Local chunk level Deduplication

4.2 Bayesian Layer

The proposed Bayesian method finds similar chunks during source local deduplication process in cloud back operations. Based on Similarity function the learning similarity metrics are created as the startup procedure in this local deduplication method. The cosine similarity function is utilized for this procedure. The input is the value generated from the similarity distance measures.During Bayesian training phase, for each record in chunk, the learnable distance metrics are trained. Then the set of paired duplicate chunks generates training corpus. Finally it creates field-level duplicate chunks and non-duplicated chunks. Now distance for each field of duplicate and non duplicated chunk pairs is calculated using learned similarity metrics which creates training data for Bayesian component. These distance features are represented as vector for Bayesian method.

The duplicate detection process associated with Bayesian layer generates potential duplicate chunk pairs. As cloud backup handles large amount of dataset, producing all possible pairs of records and computing similarity between chunks takes high computational cost since it would require $O(n^2)$ distance computations. To eliminate this problem, the canopies clustering method is

utilized using Jaccard similarity. This strategy is very efficient to handle large amount of data in cloud back system and scalable thereby reducing computational cost. Canopy method is metric based strategy on large index. The canopies are formed by segregating chunks into overlapping clusters. These canopies creates all possible duplicate chunks. The pair of chunks that are available under each cluster is a candidate for training a absolute similarity chunk pairs. To segregate duplicate and non duplicate chunks, setting chunk similarity threshold is required. For each field a chunk similarity threshold is defined as given in eq 11.

$$\delta = 1 - \sum_{i-1}^{m} \binom{n}{k} x^k a^{(n-k)} \tag{11}$$

Where δ is the threshold of a particular field, and n is the number of chunks of a given input file.

The proposed Bayesian designed for the deduplication technique generates two output values CDup and CNonDup. The value CNonDup is specific for the non-duplicate chunks and CDup is specific for duplicate chunks. Some data features are necessary to categorize the duplication and nonduplication records in chunk dataset which is useful for training the Bayesian layer. The data features train the Bayesian model. The cosine similarity measures and Jaccard similarity function are selected as data features. After calculating all the data features, values are fed into the system. After computing all the data features for every chosen duplicate record, result is given to Bayesian model. Using those results the Bayesian layer is trained to identify the duplicate and non-duplicate chunks from the given dataset. After the training the Bayesian, we can give a new chunk to find whether it has duplicate or non-duplicate. Thereafter, the similarity function is recomputed for the new chunk.

4.3 Storage Layer

Container is the unit of storage. The container in the storage layer consist of only unique chunks. The file is deduplicated using this container after loading the container and new chunk IDs are inserted into this container. Then new chunks are stored in disk, and then the file metadata containing all information to reconstruct this file is also stored in disk. Container manager is responsible for storing, allocating, deallocating, reading, writing and reliably storing containers.

5 Results and Discussion

5.1 Experimental setup

A private cloud is set using Eucalyptus open source software. The storage space in private cloud needs to be optimally utilized during cloud back up services. Hence, deduplication technique has been incorporated to create optimized storage system.

Eucalyptus consists of five functional components namely Cloud Controller (CLC), Cluster controller (CC), Storage controller (SC), Walrus and Node controller (NC) [15]. The resource allocation is performed by Cloud controller and also it maintains all client accounts. The Storage controller provides block storage services similar to Amazons Elastic Block Service (EBS) [16]. The client can interact with cloud storage through Walrus via S3 interface or REST based tools similar to Amazons [16] Simple Storage Service (S3). The Walrus store [17] the data in the installed machine. The Node controller monitors and pedals the hypervisor on each compute node and establishes the virtual machines. Walrus process the request given by Cloud controller. Walrus is accustomed with Amazons S3 which permits users to execute essential operations on the data.

Gluster File System(GFS) [18] is used to establish and set up a storage with many servers that use GlusterFS. Similar to local file system access, the cloud client can access the storage in GFS. Gluster File System is used to merge the storage resources of different machines. It gives permission for a cloud client to accumulate the consolidated storage at a single mount point. Further, it gives the privilege to the clients user to control the storage and retrieval of the files.

Four machines are configured as CC, SC, CLC and Walrus. Rest of the machines are configured as Node controllers. GFS is installed on the machines which are part of the cloud to consolidate their storage resources. Walrus allows the users to store persistent data organized as buckets and objects. Users may use third party tools to interact with walrus.

Bayesian method is implemented by interacting with walrus. Eucalyptus gives privilege to users through which cloud clients can store data. These data are organized as buckets and objects. Client can use third party tools to communicate with walrus. The command line tool s3 curl is used for wrapping Bayesian method with walrus. S3cmd that allows easy command line access to storage that supports the S3 API. S3fs allows users to access S3 buckets as local directories. Eucalyptus source code WalrusManager.java deals with the bucket creation, deletion, listing, putObject and getObject methods. we have developed java code for Bayesian method and it is incorporated into putObject and getObject methods.

5.2 Implementation

We developed a prototype of backup system to evaluate proposed approach and comparative analysis is performed. The Datasets are collected from personal cloud Dropbox, which is an online storage APP supporting automatic synchronization. We conducted the experiments with real-world datasets. The first three Backup Datasets (BDS1, BDS2, BDS3) consists of 89,

128, 272 files, which are word documents, pdf documents, PowerPoint presentations etc, it size varies from 1.4GB to 9.6 GB. Datasets 4 and 5 (BDS4 , BDS5) consists of 289 and 312 files which are disk images, size varies from 1.6GB to 7.9 GB.

During Duplicate detection process as developed in previous section, at each iteration, the pair of chunks with the highest similarity was labelled a duplicate, and the transitive closure of groups of duplicates was updated. After each iteration, Precision(P), Recall(R) and F-measure(F) defined over pairs of duplicates are calculated. The precision is the fraction of identified duplicate pairs that are correct, recall is the fraction of actual duplicate pairs that are identified and F-measure is the harmonic mean of precision and recall. These parameters are defined as follows: Precision = P/N Where P is Number of Correctly Identified Duplicate Pairs and N is Number of Identified Duplicate Pairs Recall = P/T Where T is Number of True Duplicate Pairs Fmeasure = 2 Precision Recall / Precision + Recall The proposed Bayesian method for source local duplicate chunk identification is evaluated using precision, recall and F measure values.

5.3 Experimental Results and Comparative Analysis

The Precision and recall parameters are an significant factor to be measured in solving duplicate detection problem. It decides the deduplication efficiency and deduplication computational overhead. The fig.2 depicts the precision values obtained for various back datasets using Bayesian method. The Proposed Bayesian method is compared with support vector machine (SVM) Model and Genetic programming (GP) approach which is shown in Fig.3. The precision values(P) drawn out in the interval 0 to 1.0.

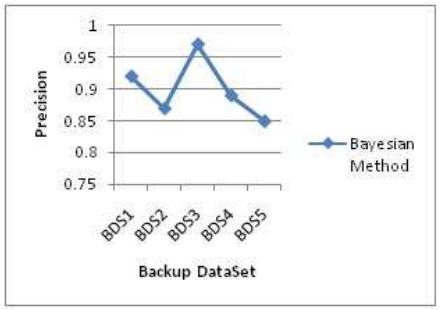

Fig. 2: Precision values of Bayesian Method

Results shows that the system equipped with Bayesian gives precision values minimum (P -0.85) and maximum (P-0.97) for a given backup datasets. Therefore the

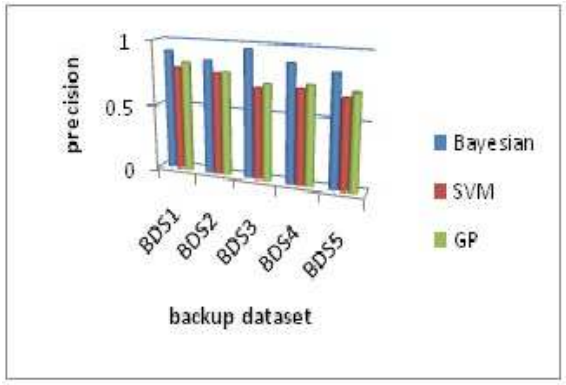

Fig. 3: Precision values comparative analysis

proposed Bayesian source chunk level deduplication method achievs higher precisions than the SVM(P 0.79) and GP (0.84) as shown in Fig 3. This implies that the number of false positives is less in Bayesian approach which proves that it is more reliable

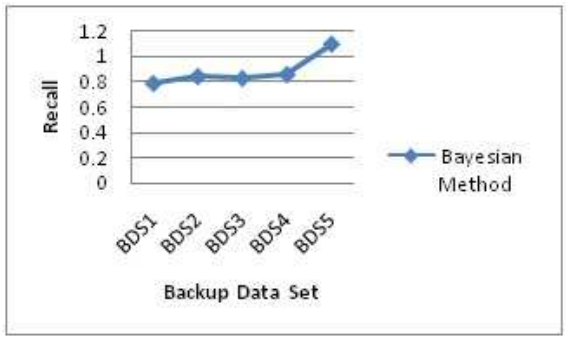

Fig. 4: Recall values of Bayesian Method

Bayesian method produces Recall (R) values in the range 0 to 1.2 and minimum (R-0.79) and maximum (1.1) as shown in Fig. 4. Correspondingly it is also proved that from Fig.5, Bayesian has the highest recall value of 1.1 on comparing with SVM (R 0.72) and GP(0.79) The high recall values imply that the number of false negatives is very less in Bayesian approach

Fig.6 depicts the F-Measure(F) of proposed Bayesian method and it is compared with SVM and GP approaches. F-Measure is a metric used to weigh precision and recall uniformly. The F values draw out in the interval 0 to 1.2 It is apparent that Bayesian has the highest F value of 1.09 with respect to SVM approach (F 0.78) and GP (0.791) implying that Bayesian approach is much more efficient when compared with SVM and GP.

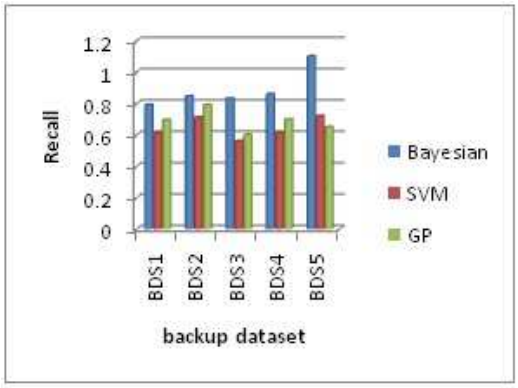

Fig. 5: Recall comparative analysis

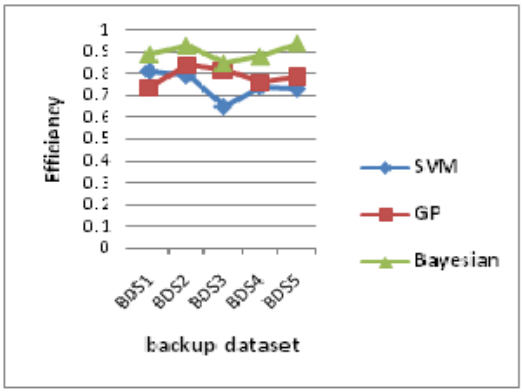

Fig. 7: Deduplication efficiency analysis

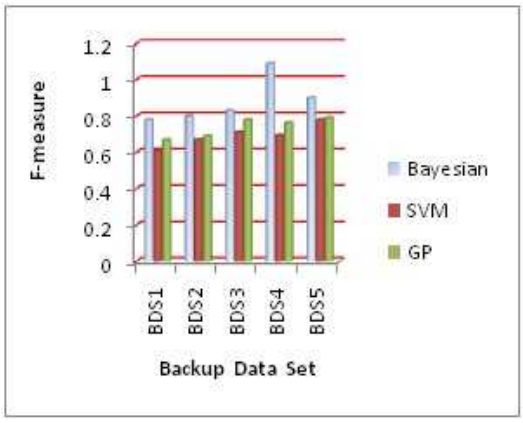

Fig. 6: F-measure comparative analysis

5.4 Deduplication Efficiency

We define deduplication efficiency as the ratio between the amount of the identified duplicate chunk and the total amount of the duplicate chunk present in deduplication method. Therefore deduplication efficiency is derived from the obtained precision and recall values. Perceptibly, the maximum de-duplication efficiency is 1(100%) and the minimum is 0 (0%).

Since precision and recall values are high, the results in Fig. 7 shows that Bayesian identifies maximum 94% of the duplicate chunk while SVM about 81% and GP method about 84% of duplicate chunk data. The minimum deduplication efficiency of 89% in proposed Bayesian method, SVM about minimum of 65% and in GP method minimum of 74%. Therefore the proposed Bayesian method increases deduplication efficiency.

5.5 Deduplication Overhead

Reduced throughput defines deduplication overhead which is significant parameter in source local chunk deduplication. The Performance analysis indicates that the deduplication time for detecting duplicate chunk. During each backup session chunk retrieval time of various datasets in deduplication process is used as a metric to evaluate the deduplication overhead. When a request arrives to retrieve a chunk, the corresponding chunk id is obtained. It is forwarded to the storage node where the index for chunks are maintained. The hierarchical index with linear hash table is implemented to hold reasonably large number of chunkID entries. The Fig. 8 shows the time taken to retrieve a chunks of various sizes for the datasets given.

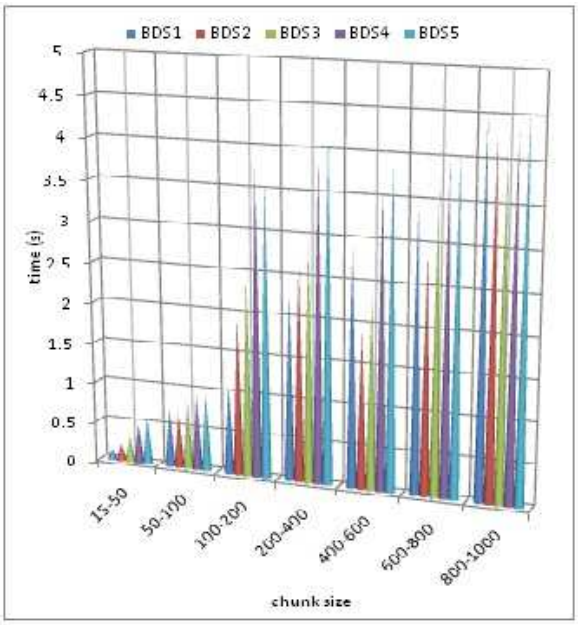

Fig. 8: Deduplication Overhead of Bayesian method

5.6 Deduplication Overhead Comparative Analysis

In Bayesian method, the time taken for deduplication is in the order of only a few seconds which is much less than that of the SVM and GP technique. The results in Fig.9 shows that in the proposed Bayesian method deduplication overhead is 26.5% on average. But in case of SVM Deduplication overhead is 48.8% and in GP 36.7%. This implies that Bayesian does not consume much deduplication time. The proposed Bayesian source local chunk model for deduplication process at the client site takes less time by singling out duplicated chunks. Thus it is proved that the proposed system gives much less deduplication overhead.

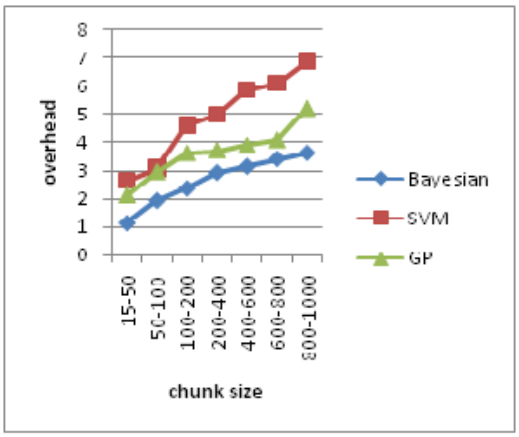

Fig. 9: Deduplication Overhead Comparative Analysis

6 Conclusion

We have proposed a Bayesian method o detect duplicate chunks at source level in deduplication process in cloud back up service. Learnable similarity metrics are utilized to check the string similarities and training phase and duplicate detection phase present in Bayesian layer generates duplicate chunks. The performance evaluation shows that precision, recall and F-measure values are high compared to SVM and GP. Due to these high values, deduplication efficiency is increased and deduplication overhead is reduced. The main application of the proposed Bayesian method is useful for optimize the storage space in any cloud backup data centers. As a future work we have planned to develop a expanded Bayesian model for global deduplication system to detect duplicate chunks across multiple clients in cloud backup services.

References

[1] Zhen Huang Yuan Yuan, Yuxing Peng Storage Allocation for Redundancy Scheme in Reliability-Aware Cloud Systems, IEEE Conference on Cloud Storage for Cloud Computing (2011).
[2] Yujuan Tan, Hong Jiang, Edwin Hsing-Mean Sha, Zhichao Yan, Dan Feng. SAFE: A Source Deduplication Framework for Efficient Cloud Backup Services, Journal of Sign Process Systems, Springer Science, 72, 209−228 (2013).
[3] Yinjin Fu, Hong Jiang, Nong Xiao, Lei Tian, Fang Liu, and Lei Xu, Application-Aware Local-Global Source Deduplication for Cloud Backup Services of Personal Storage, IEEE Transactions On Parallel And Distributed Systems,25 (2014).
[4] Benjamin Z. L. Kai, and P. Hugo, Avoiding the disk bottleneck in the data domain deduplication file system, in Proceedings of the Conference on File and Storage Technologies (2008).
[5] Junbeom Hur, Dongyoung Koo, Youngjoo Shin, Kyungtae KangSecure, Data Deduplication with Dynamic Ownership Management in Cloud Storage, IEEE Transactions on Knowledge and Data Engineering (2016).
[6] Harnik D., B. Pinkas, and A. Shulman-Peleg, Side channels in cloud services: Deduplication in cloud storage, IEEE Security and Privacy, 8 (2010).
[7] Meyer D. T. and Bolosky W. J, A study of practical deduplication, in FAST'11: Proceedings of the 9th Conference on File and Storage Technologies (2011).
[8] McCallum A. and Wellner B, Conditional models of identity uncertainty with application in Advances in Neural Information Processing Systems (2004).
[9] Chaudhuri, S. Ganti V, and Motwani. R, Robust identification of fuzzy duplicates. In Proc. 21st IEEE Intl Conf. Data Eng. (2005).
[10] Winkler, W. E, Methods for record linkage and bayesian networks. In Technical Report Statistical Research Report Series RRS, US Bureau of the Census,Washington, D.C., (2002).
[11] Osama Helmi Akel, A Comparative Study of Duplicate Record Detection Techniques, Middle East University, Jordan (2012).
[12] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios, Duplicate record detection: A survey. IEEE Transactions on Knowledge and Data Engineering, 19, 1−16 (2007).
[13] Joachims, T, Making large-scale SVM learning practical. MIT Press(2009).
[14] Moises G. de Carvalho, Alberto H.F. Laender, Marcos Andre Goncalves and Altigran S. da Silva, A Genetic Programming Approach to Record Deduplication IEEE transactions on knowledge and data engineering,24 (2012).
[15] Aline MAdalina Lonea, Private Cloud set up using Eucalyptus open Source, Proceedings of Soft Computing Applications,381-389 (2013).
[16] AWS Storage Services Overview (2015).
[17] Yohan Wadia The Eucalyptus Open-Source Private Cloud, Cloudbook Journal,3 (2012).
[18] Gluster File System 3.3.0 Administration Guide (2012).

P. Neelaveni is a Research Scholar at the Department of Information Science and Technology, Anna University, Chennai. She received her B.E degree in Computer Science and Engineering from Government College of Technology, Coimbatore and M.E degree in Computer Science and Engineering from PSG College of Technology, Coimbatore. Her research interest includes Cloud Storage, Compiler Design, Data Base Systems.

M. Vijayalakshmi is a Assistant Professor at the Department of Information Science and Technology, Anna University, Chennai. She received her B.E Degree from Regional Engineering College, Trichy. M.E and Ph.D from Anna University, Chennai. Her research interest includes Mobile Databases, Mobile Cloud Computing, Web Technology, Programming languages and Algorithms.