WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Recursive Regularization Based Feature Selection Framework for Hierarchical Classification

***K.Mubeena*[#1], *F.Asmathunnissa*[#2]**

[#1] M.Sc, Department of Computer Science, Kamban College of Arts and Science for Women, Tiruvannamalai-606603.

[#2] Assistant professor, Department of Computer Science, Kamban College of Arts and Science for Women, Tiruvannamalai-606603

ABSTRACT

Cloud data owners prefer to outsource documents in an encrypted form for the purpose of privacy preserving. Therefore it is essential to develop efficient and reliable ciphertext search techniques. One challenge is that the relationship between documents will be normally concealed in the process of encryption, which will lead to significant search accuracy performance degradation. Also the volume of data in data centers has experienced a dramatic growth. This will make it even more challenging to design ciphertext search schemes that can provide efficient and reliable online information retrieval on large volume of encrypted data. In this paper, a hierarchical clustering method is proposed to support more search semantics and also to meet the demand for fast ciphertext search within a big data environment. The proposed hierarchical approach clusters the documents based on the minimum relevance threshold, and then partitions the resulting clusters into sub-clusters until the constraint on the maximum size of cluster is reached. In the search phase, this approach can reach a linear computational complexity against an exponential size increase of document collection. In order to verify the authenticity of search results, a structure called minimum hash sub-tree is designed in this paper. Experiments have been conducted using the collection set built from the IEEE Xplore. The results show that with a sharp increase of documents in the dataset the search time of the proposed method increases linearly whereas the search time of the traditional method increases exponentially. Furthermore, the proposed method has an advantage over the traditional method in the rank privacy and relevance of retrieved document

KEYWORDS: Clustering , Encryption Schemes ,Aes

1.INTRODUCTION:

The hierarchical class structure is obviously important auxiliary information for classification learning. This information helps to divide a large and complex task into a set of relatively small and easy subtasks. Growing attention has been paid to this topic in recent years To combat the challenge of feature selection for large-scale classification, hierarchical structures can also be considered. It is not feasible to assume that all of the classes share the same set of relevant features for hierarchical feature selection [16]. The useful features for distinguishing some classes may be useless for others They did not consider the dependence among different classes in the hierarchical tree, and independently selected features for each node. However, classes in a hierarchical structure have both parent-child and sibling relationships. Classes with a parent-child relationship are similar to each other and may share In this paper, we design a hierarchical feature selection framework with recursive regularization for hierarchical classification. This framework considers the hierarchical information of the class structure. First, we model the loss term for each node with the hierarchical class structure. We then model the hierarchical information as a hierarchical regularization with parent-child relationship, sibling relationship, and family relationship, respectively. We use square loss function to measure the dependencies of hierarchical structure among parent and children. We then use the HilbertSchmidt Independence Criterion (HSIC) to measure the independence of the sibling classes, and penalize the dependence among the features selected at sibling nodes. Thus the final subsets are similar if the nodes have a parent-child relationship, while they are different if there is a sibling relationship. The contribution

of this paper are summarized as follows. • We design a hierarchical feature selection framework using the hierarchical class structure and select

different feature subsets for different class nodes. • We attempt to conduct hierarchical feature selection by considering the hierarchical class structure of parent-child relationship, sibling relationship, and family relationship, respectively. These relationships are modeled by hierarchical recursive regularization, which is more reasonable for representing the relationships between nodes than flat approaches. • In contrast to existing algorithms, we model hierarchical feature selection as a convex objective function and explore an alternation minimization strategy to solve the optimization problem with guaranteed convergence. • We use six metrics to evaluate the performance of the proposed feature selection algorithms. Extensive experiments on six hierarchical datasets demonstrate the effectiveness of our algorithms in terms of efficiency and accuracy. The original idea in this work appeared in [21]. We extend it here in the following aspects. • Based on different

hierarchical relationships, we propose Hier-FS, HiRRpar-FS and HiRRsib-FS models respectively. We discuss the relationships among different models. • We add several hierarchical feature selection methods to compare with our methods. • Instead of considering local classification accuracy on each node,we include experiments using hierarchical classification to compare.

2. LITERATURE SURVEY

[1]. **K. Ren** Cloud computing represents today's most exciting computing paradigm shift in information technology. However, security and privacy are perceived as primary obstacles to its wide adoption. Here, the authors outline several critical security challenges and motivate further investigation of security solutions for a trustworthy public cloud environment.

[2]. **C. Gentry** We propose the first fully homomorphic encryption scheme, solving an old open problem. Such a scheme allows one to compute arbitrary functions over encrypted data without the decryption key—i.e., given encryptions E(m1), ..., E(mt) of m1, ..., m t, one can efficiently compute a compact ciphertext that encrypts f(m1, ..., m t) for any efficiently computable function f.

Fully homomorphic encryption has numerous applications. For example, it enables encrypted search engine queries—i.e., a search engine can give you a succinct encrypted answer to your (boolean) query without even knowing what your query was. It also enables searching on encrypted data; you can store your encrypted data on a remote server, and later have the server retrieve only files that (when decrypted) satisfy some boolean constraint, even though the server cannot decrypt the files on its own. More broadly, it improves the efficiency of secure multiparty computation.

In our solution, we begin by designing a somewhat homomorphic "boostrappable" encryption scheme that works when the function f is the scheme's own decryption function. We then show how, through recursive self-embedding, bootstrappable encryption gives fully homomorphic encryption.

[3]. **D. Boneh** We study the problem of searching on data that is encrypted using a public key system. Consider user Bob who sends email to user Alice encrypted under Alice's public key. An email gateway wants to test whether the email contains the keyword "urgent" so that it could route the email accordingly. Alice, on the other hand does not wish to give the gateway the ability to decrypt all her messages. We define and construct a mechanism that enables Alice to provide a key to the gateway that enables the gateway to test whether the word "urgent" is akeyword in the email without learning anything else about the email. We refer to this mechanism as Public Key Encryption with keyword Search. As another example, consider a mail server that stores various messages publicly encrypted for Alice by others. Using our mechanism Alice can send the mail server a key that will enable the server to identify all messages containing some specific keyword, but learn nothing else. We define the concept of public key encryption with keyword search and give several constructions.

[4]. **D. X. Song** It is desirable to store data on data storage servers such as mail servers and file servers inencrypted form to reduce security and privacy risks. But this usually implies that one has to sacrifice functionality for security. For example, if a client wishes to retrieve only documents containingcertain words, it was not previously known how to let the data storage server perform the search and answer the query, without loss of data confidentiality. We describe our cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for the resulting crypto systems. Our techniques have a number of crucial advantages. They are provably secure: they provide provable secrecy for encryption, in the sense that the untrusted server cannot learn anything about the plaintext when only given the ciphertext; they provide query isolation for searches, meaning that the untrusted server cannot learn anything more about the plaintext than the search result; they provide controlled searching, so that the untrusted server cannot search for an arbitrary word without the user's authorization; they also support hidden queries, so that the user may ask the untrusted server to search for a secret word without revealing the word to the server. The algorithms presented are simple, fast (for a document of length n, the encryption and search algorithms only need O(n) stream cipher and block cipher operations), and introduce almost no space and communication overhead, and hence are practical to use today .

[5]. **Y.-C. Chang and M. Mitzenmacher**We consider the following problem: a user U wants to store his files in an encrypted form on a remote file server S. Later the user U wants to efficiently retrieve some of the encrypted files containing (or indexed by) specific keywords, keeping the keywords themselves secret and not jeopardizing the security of the remotely stored files. For example, a user may want to store old e-mail messages encrypted on a server managed by Yahoo or another large vendor, and later retrieve certain messages while travelling with a mobile device.

In this paper, we offer solutions for this problem under well-defined security requirements. Our schemes are efficient in the sense that no public-key cryptosystem is involved. Indeed, our approach is independent of the encryption method chosen for the remote files. They are also incremental, in that U can submit new files which are secure against previous queries but still searchable against future queries.

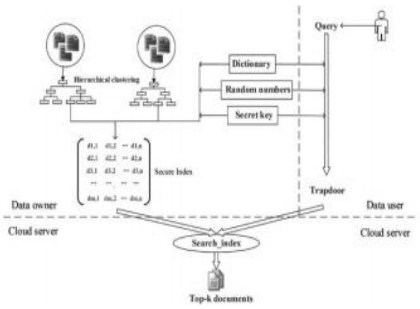
3.PROPOSED SYSTEM:

In this paper, a vector space model is used and every document is represented by a vector, which means every document can be seen as a point in a high dimensional space. Due to the relationship between different documents, all the documents can be divided into several categories.

Instead of using the traditional sequence search method, a backtracking algorithm is produced to search the target documents. Cloud server will first search the categories and get the minimum desired sub-category. Then the cloud server will select the desired k documents from the minimum desired sub-category. The value of k is previously decided by the user and sent to the cloud server. If current sub-category can not satisfy the k documents, cloud server will trace back to its parent and select the desired documents from its brother categories. This process will be executed recursively until the desired k documents are satisfied or the root is reached.

To verify the integrity of the search result, a verifiable structure based on hash function is constructed

SYSTEM ARCHITECTURE:



4. METHODOLOGY:

Data Owner Module

This module helps the owner to register those details and also include login details. This module helps the owner to upload his file with encryption using RSA algorithm. This ensures the files to be protected from unauthorized user. Data owner has a collection of documents *F ={f1; f2; :::; fn}* that he wants to outsource to the cloud server in encrypted form while still keeping the capability to search on them for effective utilization. In our scheme, the data owner firstly builds a secure searchable tree index *I* from document collection *F*, and then generates an encrypted document collection *C* for *F*. Afterwards, the data owner outsources the encrypted collection *C* and the secure index *I* to the cloud server, and securely distributes the key information of trapdoor generation and document decryption to the authorized data users. Besides, the data owner is responsible for the update operation of his documents stored in the cloud server. While updating, the data owner generates the update information locally and sends it to the server.

Data User Module

This module includes the user registration login details. This module is used to help the client to search the file using the multiple key words concept and get the accurate result list based on the user query. The user is going to select the required file and register the user details and get activation code in mail email before enter the activation code. After user can download the Zip file and extract that file. Data users are authorized ones to access the documents of data owner. With *t* query keywords, the authorized user can generate a trapdoor *TD* according to search control mechanisms to fetch *k* encrypted documents from cloud server. Then, the data user can decrypt the documents with the shared secret key.

Cloud Server :

This module is used to help the server to encrypt the document using AES Algorithm and to convert the encrypted document to the Zip file with activation code and then activation code send to the user for download. Cloud server stores the encrypted document collection *C* and the encrypted searchable tree index *I* for data owner. Upon receiving the trapdoor *TD* from the data user, the cloud server executes search over the index tree *I*, and finally returns the corresponding collection of top- *k* ranked encrypted documents. Besides, upon receiving the update information from the data owner, the server needs to update the index *I* and document collection *C* according to the received information. The cloud server in the proposed scheme is considered as “honest-but-curious”, which is employed by lots of works on secure cloud data search

Rank Search Module

These modules ensure the user to search the files that are searched frequently using rank search.This module allows the user to download the file using his secret key to decrypt the downloaded data. This module allows the Owner to view the uploaded files and downloaded files. The proposed scheme is designed to provide not only multi-keyword query and accurate result ranking, but also dynamic update on document collections. The scheme is designed to prevent the cloud server from learning additional information about the document collection, the index tree, and the query.

Hierarchical clustering Search Module

These modules ensure the user to search the files that are searched frequently using hierarchical clustering search.This module allows the user to download the file using his secret key to decrypt the downloaded data. This module allows the Owner to view the uploaded files and downloaded files. The proposed scheme is designed to provide not only multi-keyword query and accurate result clustering, but also dynamic update on document collections. The scheme is designed to prevent the cloud server from learning additional information about the document collection, the index tree, and the query.

5 CONCLUSION

In this paper, we investigated ciphertext search in the scenario of cloud storage. We explore the problem of maintaining the semantic relationship between different plain documents over the related encrypted documents and give the design method to enhance the performance of the semantic search. We also propose the MRSE-HCI architecture to adapt to the requirements of data explosion, online information retrieval and semantic search. At the same time, a verifiable mechanism is also proposed to guarantee the correctness and completeness of search results. In addition, we analyze the

search efficiency and security under two popular threat models. An experimental platform is built to evaluate the search efficiency, accuracy, and rank security. The experiment result proves that the proposed architecture not only properly solves the multi-keyword ranked search problem, but also brings an improvement in search efficiency, rank security, and the relevance between retrieved documents.

6. RESULTS

Fig:1 Hirerachy Recursive Clustering

Fig:2 Comparison of different clustering

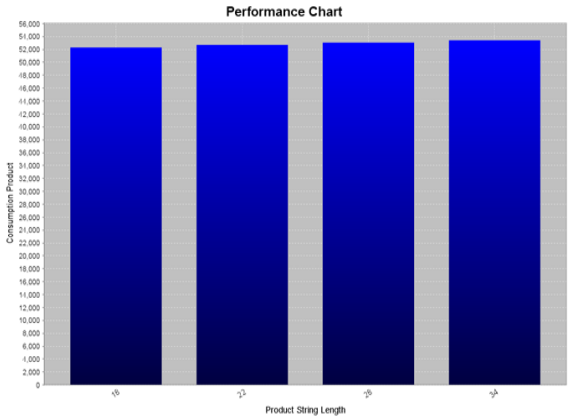
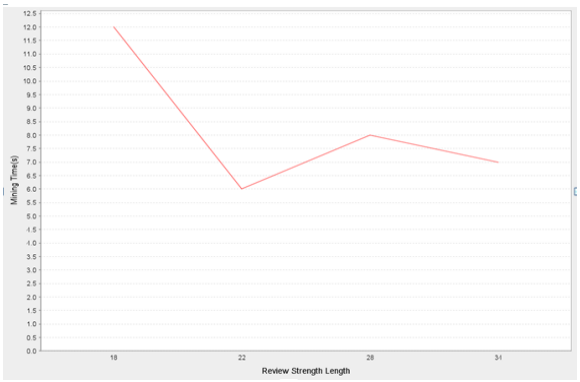


Fig:3 Performance Graph



REFERENCES:

[1] K. Ren, C.Wang, Q.Wang*et al.*, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[2] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Financial Cryptography and Data Security*. Springer, 2010, pp. 136–149.

[3] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[4] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," *Journal of the ACM (JACM)*, vol. 43, no. 3, pp. 431–473, 1996.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-Eurocrypt 2004*. Springer, 2004, pp. 506–522.

[6] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III, "Public key encryption that allows pir queries," in *Advances in Cryptology-CRYPTO 2007*. Springer, 2007, pp. 50–67.

[7] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*. IEEE, 2000, pp. 44–55.

[8] E.-J. Goh*et al.*, "Secure indexes." *IACR Cryptology ePrint Archive*, vol. 2003, p. 216, 2003.

[9] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Proceedings of the Third international conference on Applied Cryptography and Network Security*. Springer-Verlag, 2005, pp. 442–455.

[10] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 79–88.