ENABLING BETTER MEDICAL IMAGE CLASSIFICATION THROUGH SECURE COLLABORATION

Jaideep Vaidya

Rutgers University
MSIS Department
Newark, NJ 07102
jsvaidya@rbs.rutgers.edu

Bhakti Tulpule

Rutgers University
BME Department
Piscataway, NJ 08854
btulpule@eden.rutgers.edu

ABSTRACT

Privacy is of growing concern in today's day and age. Protecting the privacy of health data is of paramount importance. With the rapid advancement in imaging technology, analysis of medical images is now one of the most dynamic fields of study today. Image analysis is performed for a variety of purposes, ranging from image enhancement to image segmentation. It can easily be seen that having access to more information makes the analysis results more accurate. For example, supervised classification based image segmentation requires good and plentiful training data. We wish to utilize the training data at different locations to obtain more accurate image segmentation while still protecting the privacy of individual patients. Work in the field of secure multi-party computation (SMC) in cryptography shows how to compute functions securely and quantifies what it means to be secure. Applying SMC protocols in image processing is a challenging problem. This paper looks at how some of this work can be leveraged to perform privacy-preserving image analysis and classification.

Index Terms— Communication system security, Image Analysis, Distributed Algorithms, Cryptography

1. INTRODUCTION

In the new information age, while data is increasingly ubiquitous, access to it needs to be significantly more restricted. Privacy/security concerns severely restrict the sharing of data. This is especially true in the case of medical data. HIPAA rules [1] do not allow sharing of medical data without appropriate anonymization. The marriage of computers and medicine has lead to large dividends. However, with many medical studies, the foremost problem is the lack of real data. Real data is scarce – getting the appropriate permissions, suitably anonymizing it, etc. is a formidable task.

Consider the case of the development of a computer-aided diagnostic (CAD) system. Such a system typically uses supervised classification based image segmentation. An example of this is a CAD system that automatically segments out the cancerous regions from an image. Such classifiers first need to be trained using labeled data (i.e., data for which the classes are already known). In the case of CAD systems such data is called the ground truth. Clearly, increased ground truth improves the accuracy of the classifier. However, such data is not easy to access. Privacy/security concerns restrict the availability of the ground truth data. The paucity of real data often forces the researcher to develop his classifier based on the limited data available. Such a classifier is often not very accurate. Interestingly, the kinds of information extracted from the ground truth is mostly summary in nature and does not automatically breach privacy in and of itself. It is mainly the access to the original data that is the problem.

Thus, the problem that we wish to address is the following: A biomedical informatics researcher wants to develop a CAD system but has very limited ground truth available. Due to this limitation the system is likely to be inaccurate at the best. The researcher is in touch with several doctors at medical schools, who have the ground truth data available but are restricted from sharing it due to privacy/security concerns. Is there some way in which the researcher can develop the system without looking at the data or knowing what it is. Furthermore, can this be done in a quantifiable way (exactly what information is revealed), so that regulations can be met.

1.1. Model

Formally, we consider the following model: There are k parties (i.e., doctors / labs) $P_1, \ldots, P_k$ who each own or have access to several medical images. Assume that party P_i has n_i images. Thus, the total number of images is $n = \sum_{i=1}^{n} n_i$. A researcher R (who could be one of the doctors/labs earlier) wishes to use all of the images to formulate an accurate classification model. While the researcher can be trusted with the model, he/she does not have access to the original image data.

2. RELATED WORK

Secure computation has a very rich history. Yao first postulated the two-party comparison problem (Yao's Millionaire

Protocol) and developed a provably secure solution[2]. Goldreich et al.[3] generalized this to multi-party computation and proved that there exists a secure solution for any functionality. The approach used is as follows: the function F to be computed is first represented as a combinatorial circuit, and then the parties run a short protocol for every gate in the circuit. Every participant gets random shares of the input and output wires for every gate. This approach, though appealing in its generality and simplicity, means that the number of rounds of the protocol grows with the size of the circuit. This grows with the size of the input. This is highly inefficient for large inputs or complicated circuits. The sheer size of images (number of pixels) makes the general method completely unfeasible for image processing. However this does prove that secure solutions exist.

There has been some recent effort on applying secure computation concepts to image analysis. [4] propose a technique for secure image filtering based on securely computing the scalar product. [5] propose a technique for blind vision – securely using a face detection algorithm without revealing either the images or the algorithm. There has also been significant work in the privacy-preserving data mining area that looks at how to perform data analysis securely over distributed data. This work should be relevant since many of the concerns and challenges (including the size of the data) are quite the same. [6] provides a comprehensive survey.

3. ALGORITHM

To develop the classification model, the researcher needs to first identify a set of features that would give a good classification. We assume that the researcher has identified such features. The classifier then needs to be trained using these features and the ground truth (i.e., the labeled data). After training, the classifier can now be used to classify new data.

3.1. Training

Let us consider a few common statistical features that are typically extracted from the set of images. First is the mean, following which the standard deviation is computed. These can actually be computed using the secure sum algorithm in Section 4.

First, all parties standardize their data. This ensures that across all the images, similar regions fall whithin the same range of intensity values. Each party, locally computes the mean intensity level around each pixel for every image that it owns. The mean around each pixel is computed using the pre-selected window size of $c \times c$. Next, all of these means are added together to generate a local sum, sum_i. This can be done locally. Similarly the variable n_i is initialized to the total number of pixels of ground truth in all of the images. R then invokes the secure sum algorithm (Section 4) to sum up all the n_i for all the i parties. This gives the total number of pixels of ground truth n, with all of the parties. Similarly, R invokes the secure sum algorithm to sum up the local sum, sum_i for all of the parties P_i. Finally, the global mean is computed by dividing global sum (sum) by the total number of pixels (n). The secure sum algorithm ensures that the researcher gets all the information he wants without knowing the local values at each party. The steps for computing the standard deviation are quite similar. Algorithm 1 provides the complete details.

While here we only describe how to compute the mean and the variance, these are crucial steps to computing any standard image processing functionality. The computation of many statistical features typically require sliding window operations with computation of mean and variance. Securely computing these is an indispensible part of image processing. In the future we will explore how to compose specific techniques using these and other secure sub-blocks.

4. SECURE SUM

What are some of the cryptographic tools that may help? One building block frequently required is a way to securely calculate the sum of values from individual sites. Assuming three or more parties and no collusion, the following method ([7]) securely computes such a sum.

Assume that the value $v = \sum_{i=1}^{k} v_i$ to be computed is known to lie in the range $[0..n-1]$ where v_i denotes the share of party P_i.

The parties also randomly order themselves into a ring. The ordering can be selected by one of the parties, or by a third party. Without loss of generality, we assume that this order is the canonical order $P_1, \ldots, P_k$. In general, any order can be decided on. The protocol proceeds as follows:

P_1 is designated as the *master* site. P_1 generates a random number r, uniformly chosen from $[0..n-1]$. P_1 adds this to its local value v_1, and sends the sum $r + v_1 \bmod n$ to P_2. Since the value r is chosen uniformly from $[0..n-1]$, the number $r + v_1 \bmod n$ is also distributed uniformly across this region, so P_2 learns nothing about the actual value of v_1. For the remaining sites $i = 2..k-1$, the algorithm is as follows. P_i receives

$$V = r + \sum_{j=1}^{i-1} v_j \bmod n.$$

Since this value is uniformly distributed across $[0..n-1]$, P_i learns nothing. P_i then computes

$$r + \sum_{j=1}^{i} v_i \bmod n = (v_i + V) \bmod n$$

and passes it to P_{i+1}.

P_k performs the above step, and sends the result to P_1. P_1, knowing r, can subtract r to get the actual result. Note that P_1 can also determine $\sum_{i=2}^{k} v_i$ by subtracting v_1. This

Algorithm 1 Secure computation of mean and standard deviation

Require: k parties, $P_1, \ldots, P_k$ owning images with ground truth
Require: A researcher R who wishes to use the ground truth for training classification models
Require: The parties/researcher agree on a window size, $c \times c$

All parties standardize all of the data
{Compute the mean}
for each party P_i {Parallel Operations} **do**
  for each owned image I_j **do**
    $n_{ij} \leftarrow$ number of pixels/voxels in the ground truth
    for each pixel/voxel q in the ground truth **do**
      Locally compute the mean μ_{ijq} using the window size, c
    end for
    Compute $sm_{ij} = \sum_q \mu_{ijq}$
  end for
  Compute $n_i = \sum_j n_{ij}$
  Compute $sum_i = \sum_j sm_{ij}$
end for
At R: Invoke the secure sum algorithm (Section 4) to compute $n = \sum_i n_i$
At R: Invoke the secure sum algorithm (Section 4) to compute $sum = \sum_i sum_i$
At R: Compute the global mean, $\mu = sum/n$
At R: Send the global mean μ to all the other parties
{Compute the standard deviation}
for each party P_i {Parallel Operations} **do**
  for each owned image I_j **do**
    for each pixel/voxel q in the ground truth, compute $sv_q = (v_q - \mu)^2$ (v_q represents the intensity of pixel/voxel q)
    Compute $sdsum_{ij} = \sum_q sv_q$
  end for
  Compute $sdsum_i = \sum_j sdsum_{ij}$
end for
At R: Invoke the secure sum algorithm (Section 4) to compute $sdsum = \sum_i sdsum_i$
At R: Compute the global standard deviation, $\sigma = \sqrt{\frac{1}{n-1} * sdsum}$

is possible from the global result *regardless of how it is computed*, so P_1 has not learned anything from the computation.

This method faces an obvious problem if sites collude. Sites P_{i-1} and P_{i+1} can compare the values they send/receive to determine the exact value for v_i. The method can be extended to work for an honest majority. Each site divides v_i into shares. The sum for each share is computed individually. However, the path used is permuted for each share, such that no site has the same neighbor twice. To compute v_i, the neighbors of P_i from each iteration would have to collude. Varying the number of shares varies the number of dishonest (colluding) parties required to violate security. Detailed analysis of this method can be found in [8].

Another inherent problem due to the modular addition is the fact that overflow leads to completely wrong results. While we assume that the global sum v is within the range $[0 \ldots n-1]$, if this assumption is broken, then the final result is wrong. However, it is quite easy to choose n to be large enough such that the global sum is definitely within it.

5. SECURITY

To prove the security of the "integer sum" method in Section 4, we need the framework of secure multiparty computation, which provides a solid theoretical underpinning for privacy. The key notion is to show that a protocol reveals nothing except the results. This is done by showing how everything seen during the protocol can be simulated from knowing the input and the output of the protocol.

We assume the semi-honest model for security. A semi-honest party follows the rules of the protocol using its correct input, but is free to later use what it sees during execution of the protocol to compromise security. This model fits our problem domain very well. The individual doctors or researcher does not necessarily want to actually attack the system. What they care more about is ensuring confidentiality and meeting federal regulations and standards for privacy. The formal definition of private two-party computation in the semi-honest model can be found in [9].

Intuitively, a protocol is secure in the semi-honest model if a group of parties which follow the protocol are unable to learn anything that they could not in the ideal model (where they simply give their inputs to a trusted third party which computes the functionality). This leads to the notion of privacy by simulation. Thus, a computation is secure if given only a particular party's input and output, it is possible to *simulate* his view of the entire protocol without knowing any other party's input or output. Since he cannot distinguish his view of simulation (which does not depend on any data not known to him) from his view of the protocol (which does, generally depend on the inputs of the other parties), we claim that he does not learn anything beyond what he should by participating in the protocol. Thus, in a proof of security, we only need to show the existence of a simulator for each party that satisfies the above definitions.

This does not quite guarantee that private information is protected. For example, if there are only two parties, the sum of the private values automatically reveals the private value of the other party. Similarly, even with more than two parties, a party can always subtract its own value from the final result to get the sum of the other values. This is inherent from the function computed, and would be possible even if we have a completely secure protocol using a trusted third party. Thus, the key to the definition of privacy is that nothing is learned *beyond* what is inherent in the result, i.e., the process

of computing the function does not reveal anything. We now intuitively present the security of the secure sum protocol. A complete formal proof of security can be found in [10].

In order to prove the security of the protocol, it is sufficient to show how to construct a simulator for each party. The simulator for each party proceeds simply by executing the actual protocol. In order to show that the view of each party can be simulated, we only need to simulate the messages received by each party using only the local input and the final output. The messages sent can be simulated by actually computing them from the local input and the messages received during the protocol.

P_1's view: At the end of the protocol, P_0 receives $r + \sum_{i=1}^{k} v_i \bmod n$ from P_k. Since, P_1 knows r, and the final result $\sum_{i=1}^{k} v_i \bmod n$, it can easily compute the message it receives. $P_2, \ldots, P_k$'s view: Site P_i $(i = 2 \ldots k)$ receives $V = r + \sum_{j=1}^{i-1} v_j \bmod n$. To simulate this, P_i simply chooses a random number r_i from a uniform distribution over $0..n-1$. Now,

$$\begin{aligned} Pr\left[VIEW_i^{Protocol} = x\right] &= Pr\left[r + \sum_{j=1}^{i-1} v_j \bmod n = x\right] \\ &= Pr\left[r = x - \sum_{j=1}^{i-1} v_j \bmod n\right] \\ &= \frac{1}{n} \end{aligned}$$

Since the simulator chooses each number in the range 0 to $n-1$ equiprobably, this matches the probability of seeing that number. Therefore, what each site sees is indistinguishable from that simulated with a random number generator.

Note that P_1 can also determine the sum of the other values ($\sum_{i=1}^{k} v_i$) by subtracting v_1. However, this is possible from the global result *regardless of how it is computed*, so P_1 does not learn anything from the computation.

6. COMPUTATION/COMMUNICATION ANALYSIS

The secure sum protocol is remarkably efficient. The only extra computation required by the first party is the generation of the random number, and the final subtraction. In terms of communications, k rounds are required, 1 for each party. For a collusion resistant protocol (if at least q parties have to be malicious), the communication and computation cost is further multiplied by q. Overall, most of the computation is local. The secure sum algorithm is only invoked thrice. Thus, the overhead due to privacy/security is minimal.

7. CONCLUSION

In this paper we have explored the possibility of using tools from the field of secure computation to enable secure collaboration – thus allowing us to create more accurate image analysis and classification. While the specific features considered (mean and standard deviation) are quite basic, they demonstrate the fact that secure collaboration is possible and can significantly help in expanding the applicabilities of medical studies. The basic algorithm can easily be applied to all additive features (i.e., all features where the global value can be expressed as a sum of local values). In the future, we plan to significantly expand this suite of tools to enable computation of many more features, and create a collaboration environment for the biomedical researcher.

8. REFERENCES

[1] "Standard for privacy of individually identifiable health information," *Federal Register*, vol. 66, no. 40, Feb. 28 2001.

[2] A. Yao, "How to generate and exchange secrets," in *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*. IEEE, 1986, pp. 162–167.

[3] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game - a completeness theorem for protocols with honest majority," in *19th ACM Symposium on the Theory of Computing*, 1987, pp. 218–229.

[4] N. Hu, S. Cheung, and T. Nguyen, "Secure image filtering," in *Proceedings of the 13th IEEE International Conference on Image Processing (ICIP)*, 2006.

[5] S. Avidan and M. Butman, "Blind vision.," in *ECCV (3)*, Ales Leonardis, Horst Bischof, and Axel Pinz, Eds. 2006, vol. 3953 of *Lecture Notes in Computer Science*, pp. 1–13, Springer.

[6] J. Vaidya, C. Clifton, and M. Zhu, *Privacy-Preserving Data Mining*, vol. 19 of *Advances in Information Security*, Springer-Verlag, 1st edition, 2005.

[7] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," in *The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02)*, Madison, Wisconsin, June 2 2002, pp. 24–31.

[8] B. Chor and E. Kushilevitz, "A communication-privacy tradeoff for modular addition," *Information Processing Letters*, vol. 45, no. 4, pp. 205–210, 1993.

[9] O. Goldreich, *The Foundations of Cryptography*, vol. 2, chapter General Cryptographic Protocols, Cambridge University Press, 2004.

[10] M. Kantarcıoğlu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1026–1037, Sept. 2004.