מעבדה בחישוב בטוח בענן 203.4850
Laboratory in Secure Computation in the Cloud

Adi Akavia
University of Haifa, Fall 2019

Project Topic:
Privacy-preserving machine learning
SVM inference on FHE encrypted data

Contents

Background

Privacy preserving machine learning (PPML)

- **Machine learning**
  - Huge success, training data required, more data higher accuracy
  - Example: deep learning, support vector machines (**SVM**)
  - Privacy threats
- **Privacy preserving machine learning**
  - Utilize secure computation techniques to
  - execute machine learning inference and training,
  - without revealing information on the input beyond the output.

Fully homomorphic encryption (FHE)

- **FHE** is a conceptually simple secure computation technique:
  - Encryption scheme that allows manipulating ciphertexts (aka, homomorphic evaluation) to produce a ciphertext for a function f on the underlying message $m_1,\ldots,m_n$.
  - FHE is specified by four algorithms **(KeyGen, Enc, Dec, Eval)**.
  - Standard security and correctness are required.
  - Eval is required to be correct in the sense that,

    given f and $c_1 \leftarrow \mathrm{Enc}_{pk}(m_1), \ldots, c_n \leftarrow \mathrm{Enc}_{pk}(m_n)$,

    it holds that $\mathbf{c \leftarrow Eval_{pk}(f,c_1,\ldots,c_n)}$

    is a ciphertext so that

    $$\mathbf{Dec_{sk}(c) = f(c_1,\ldots,c_n)}$$
- Useful for **secure outsourcing** supporting protection of data-in-use:
  - Offline: data owner (client) produces keys (sk,pk), publishes pk, encrypts data and upload to cloud server
  - Online:
    - Client sends function f to the server, plus any additional encrypted or cleartext data as needed
    - server homomorphically evaluate f on ciphertexts to produce the result ciphertext c, and sends c to client
    - client decrypts c to obtain output
  - Advantages:
    - client's online complexity:

            - time to encrypt and decrypt
            - independent of complexity of f
        - server's time polynomial in complexity of f
        - privacy for data owner (client) guaranteed
- **Model of computation:** Computation f is specified as an **arithmetic circuit**
    - Arithmetic circuits are circuits with gate for Addition and Multiplication operations only
    - +,× are in a field F, where F may be:
        - a finite field, eg + mod p, × mod p, or
        - (approximate) real numbers
    - Fact: (+,×) is complete in the sense that it can compute any f with no more than a polynomial overhead.
    - Complexity is governed essentially by:
        - **×-depth** of the circuit, ie, number of multiplication operations along longest path from root to leaf
        - Total **number of multiplication** operations
        - **Depth** = longest path from root to leaf (counting all gates, not just multiplication)
        - **Size** = number of gates in circuit (+ and ×)
- **Challenges** in designing FHE-friendly algorithms:
    - Low level programming (specifying program as a circuit) may be challenging
    - No branching
    - Worst-case input run-time always
- **Complexity** goals: design circuits that are
    - shallow (ie low ×-depth), and
    - use few multiplications
    - have "reasonable" (total) depth and size
- Examples:
    - How to compute *sign* function?
    - How to compute greater than operator?

Project description – SVM inference on FHE encrypted data

Your Tasks

1) **<u>Train SVM model on cleartext data.</u>**
    a. Which **training data** to use?
        E.g.: life sound signals https://research.google.com/audioset
    b. What's the inference goal?
        Single class inference; e.g. identify dog bark
    c. What tools to use for training?
        See resources below for an initial list of available tools
    d. Which kernels to use?
        check accuracy with various common kernels
        choose best accuracy vs. performance tradeoff
        Hint: polynomial kernels of low degree polynomial achieve
        better performance on FHE encrypted data than high degree polynomials
        (can you explain why?)

2) <u>**Design an arithmetic circuit** for computing SVM inference with the SVM model you trained.</u>
    a. Correctness: is your circuit always producing identical results to cleartext inference?
        If not, how do they differ?
        What's the effect on accuracy?
    b. Complexity: What's the complexity of your circuit (×-depth, #mult, depth and size)?

3) **<u>Implement inference with your SVM model on FHE encrypted data</u>**
    a. Use SEAL library for homomorphic encryption
        Note: you must choose whether to work of reals (CKKS) or finite filed (FV)
    b. Use your arithmetic circuit as the inference algorithm.

4) **<u>Evaluate and optimize:</u>** Test your implementation for accuracy and performance.
    Can you improve your solution to have better performance or accuracy?

5) **Write a report** and documentations

a. Report should be in English, written using Latex, and in academic writing style and level

b. Writing tips (search for more online!)

    i. Write top-down, motivate each topic before dwelling into details,

    ii. Each writing unit (report, section, paragraph, etc) should start with a concise sentence explaining its role: what are you going to discuss in this unit? And end with a concise sentence summarizing what your discussed.

    iii. Write with a natural logical flow to keep reader engaged and comprehending.

c. Report should include the following sections:

    i. introduction (concise background, summary and impact of your work),

    ii. background,

    iii. your algorithm design,

    iv. your implementation details,

    v. your empirical evaluation

    vi. conclusions

d. In addition, give documentations and test files to your 1) cleartext training and 2) FHE inference work, with easy to use sample files

6) **Submit**: report + code (source files, test and demo files, benchmarks, documentation).

Further details: architecture for SVM inference on encrypted data

Offline:

- Server trains SVM model on cleartext data
- Client generates (pk,sk), sends pk to server.

Online:

- Client encrypts a data sample c←Enck$_{pk}$(x), and sends the corresponding ciphertexts c to server
- Server homomorphically evaluates the SVM model on the encrypted data sample c to produce an encrypted inference outcome c’, and send c’ to client
- Client decrypts c’ to obtain the inference result.

Further details: evaluation and optimization

Accuracy

- Cleartext model evaluation:
  How good is your model? Use standard model evaluation methods to answer.
  E.g. use cross validation, summarize rates of false positive and false negative, summarized in a Confusion map.
- FHE inference vs. cleartext:
  Is the inference result the same as when evaluating the model on cleartext data?

Performance

- Time of inference on encrypted data: What’s the run time (total and in each part)? What’s the overhead over cleartext data? What are the bottlenecks?
- Communication complexity: bandwidth (how many bytes are transmitted)?

Optimization

- Can performance be improved?
- Can accuracy be improved
- Is there a conflict between accuracy and performance? If so, what’s your recommended “sweet point”?

General guidelines

- **You are responsible** for the management of your time, progress, and meetings with me, to ensure the successful completion of your project. Note: report writing typically takes a long time; do take this under consideration when planning your time.
- **Meeting** weekly / every other week to update and discuss is <u>highly recommended</u>.
    - If you're stuck, this could help show a path to examine
    - If you tend to procrastinate, meeting gives you a deadline
    - Even if you think you're doing great – you might unknowingly be off the right path – a meeting can help examine your current situation and give feedback.
- **Excelling students are invited to discuss with me opportunities in my research group**
    - BSc project, MSc project/thesis, PhD thesis
    - Interesting & challenging research projects
    - Stipend (MILGA) may be possible

Resources (initial list, search for further resources!)

- SEAL FHE library https://www.microsoft.com/en-us/research/project/microsoft-seal/
- SVM tutorial (theory, concise) https://web.mit.edu/zoya/www/SVM.pdf
- SVM tutorial (useful libraries links) https://scikit-learn.org/stable/modules/svm.html
- Training data: life sound signals https://research.google.com/audioset
- Research papers on privacy preserving machine learning with FHE