Multiplicação de matrizes distribuída segura através do uso de Raízes da Unidade

Roberto Assis Machado [13]

Resumo: O uso de dados tem ficado cada vez mais comum para basear a tomada de decisões tais como campanha de marketing, estratégia de crescimento, criação de máquinas autônomas, etc. O aumento da quantidade dados para as mais diversas aplicações dificulta a computação e, na maioria das vezes, são infactíveis localmente. Isso faz com que as pessoas comecem a computar os modelos nos poderosos servidores na nuvem. Nesse contexto, os usuários podem requerer tarefas de alto custo computacional em grandes quantidades de dados que estão distribuídos em servidores alocados em diferentes regiões do mundo. Por um lado, tal flexibilidade de computação reduz o custo individual para as empresas e profissionais que não precisam manter um equipamento poderoso apenas com o objetivo de fazer os cálculos massivos em um imenso banco de dados. Mas em contrapartida, aumenta a preocupação com a privacidade dos dados para evitar a exposição de dados sensíveis de usuários. Dessa forma, os algoritmos devem beneficiar os usuários que não possuem equipamentos poderosos enquanto mantêm a privacidade dos dados. Existem pelo menos três problemas de otimização a serem analisados: 1) Minimizar o custo de upload; 2) Minimizar o custo de download; 3) Minimizar o custo total de comunicação. Considere que tenhamos L fontes de dados $A1, A2, ..., AL$ queremos computar uma função $GA1, A2, ..., AL$ com a ajuda de N servidores. Cada uma das fontes está conectada a cada servidor, através de links confiáveis e sem erros. Todos os servidores estão conectados entre si. Impomos a condição de privacidade teoricamente perfeita com o cruzamento de T servidores, i.e., a informação contida em quaisquer T servidores não permite o vazamento de qualquer informação tanto dos valores de entrada $A1, A2, ..., AL$ quanto do resultado da função $GA1, A2, ..., AL$. Assumimos, também, que os servidores são honestos, responsivos e curiosos o que significa que eles seguem os passos designados honestamente, mas eles podem tentar cruzar T servidores para tentar deduzir alguma informação a respeito dos valores de entrada e de saída.

Nesse trabalho, focaremos na operação segura de multiplicação distribuída de duas matrizes $A = A1, A2, ..., AL$ e $B = B1\vdots BL$ utilizando N servidores e mantendo a privacidade de A, B e $AB = A1B1, A2B2 + ... + ALBL$ sob o cruzamento de T servidores. Todas as operações serão realizadas em um corpo finito Fq, com q elementos. Um elemento $x \in Fq$ é uma k-ésima raiz da unidade se $xk = 1$ mas $xj \neq 1$ para todo $0 < j < k$. Sejam $L = 1$, $T = 1$ e $F7 = \{0, 1, 2, 3, 4, 5, 6\}$. Considere A, B, R e S matrizes quadradas de ordem $(m \times m)$ sendo R e S matrizes geradas aleatoriamente. Estamos interessados em computar AB de modo privado. Considere os polinômios $A(x) = A + Rx$ e $B(x) = B + Sx$. Note que $ABx = AxBx = AB + AS + RSx + RSx2$ e que $AB = AB(0) = AB(1) + AB(2) + AB(4)3$.

[13]Rutgers University,
robertoassismachado@gmail.com

Enviando os valores $A(i)$, $B(i)$ para o i-ésimo servidor é possível computar o produto AB sem que cada servidor consiga deduzir algo de A, B ou AB.

Teorema: Dadas as condições acima, sempre é possível fazer a multiplicação de matrizes com no máximo $L + 2T$ servidores.

Referências

[1] I. S. Reed and G. Solomon, Polynomial codes over certain finite fields, Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 2, pp. 300–304, 1960.

[2] R. D'Oliveira, S. El Rouayheb, and D. Karpuk, GASP codes for secure distributed matrix multiplication, arXiv e-prints, Dec. 2018.

[3] N. Mital, C. Ling and D. Gunduz, Secure Distributed Matrix Computation with Discrete Fourier Transform, arXiv e-prints, Jul. 2020.