Privacy Preserving Reinforcement Learning over Distributed Datasets

Regis Loeb[2], Timothy Verstraeten[1], Ann Nowe[1], and Ann Dooms[1]

[1] Vrije Universiteit Brussel, Belgium
[2] Universite Libre de Bruxelles, Belgium

As the IoT is becoming increasingly popular, the possibility to exchange data between a group, or *fleet*, of similar devices arises. This allows institutions to share their data about a specific control task to boost learning processes. However, such data sets are often confidential and cannot be shared in their raw form. We propose a privacy-preserving reinforcement learning technique that allows knowledge transfer among similar agents.

Our starting point is the setting of fleet reinforcement learning (RL), in which similar agents (the fleet) need to solve a similar task. The objective of an agent is to learn a control policy that maximizes the expected cumulative reward. Before the learning process, agents are allowed to share data. However, as agents in reality have small discrepancies (e.g., degradation or production errors), not all data samples are representative for a particular agent. Therefore, knowledge needs to be transferred only when it is relevant.

To construct a privacy-preserving (PP) fleet RL method, we enhance the fleet Gaussian process reinforcement learning (FGPRL) method exposed in [1] with a secure multi-party computation (SMPC) protocol. In FGPRL, the goal is to estimate the transition function, such that the optimal policy can be inferred. This is done using a Gaussian process (GP). A GP is a collection of annotated random variables, any finite number of which have a joint Gaussian distribution. In a regression context, these random variables are the outputs of an unknown function, and their annotations are the inputs to that function. The most important parameter of a GP is its covariance kernel, which describes how these outputs are correlated, i.e., how knowledge about one output gives information about another. Coregionalization extends this idea to the outputs over multiple agents. A coregionalized GP is used in FGPRL as the joint transition model, such that the transitions between fleet members can be correlated in order to effectively share data based on the similarities between the members. In order to perform prediction on new input points X_*, a GP is conditioned on the fleet's training data. The following formulas can be used to obtain the posterior statistics of $f_* := f(X_*)$

$$\mathbb{E}[f_*] = K(X_*, X)K(X, X)^{-1}y \tag{1}$$

$$\mathbb{V}[f_*] = K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*) \tag{2}$$

($K(X, Y)$ is the covariance matrix between all pairs of X and Y coordinates).

In order to make the FGPRL method PP, we have to compute (1) and (2) in such a way that each agent in the fleet leaks no information about its own samples. To this end, we use a SMPC, which is an m-party cryptographic protocol. It consists of specifying a random process describing a desired functionality mapping m inputs (one local input per party) to m outputs (one local output per party). If the m parties could trust some external party, then they would each send their local input to that trusted party, who would compute the outcome of the process and send to each party the corresponding output. *This imaginary trusted party can be emulated by the mutually distrustful parties themselves.* We expose two models of SMPC with different assumptions in which such an emulation is possible. Those assumptions are about communication channels, adversarial behavior and the desired level of emulation of the trusted party (which is equivalent to the level of security).

We use secret sharings and perform computations modulo q on those. A value x is secretly shared over $\mathbb{Z}_q$ by picking $x_1, ..., x_m$ randomly and uniformly in $\mathbb{Z}_q$ (the ring of integers modulo q represented by $\{0, 1, ..., q-1\}$) under the constraint that $x = \sum_{i=1}^{m} x_i \; mod \; q$ and then distributing each share x_i to party i. Additions, subtractions or multiplications by a public constant of secret sharings can be performed *locally by the parties without any interaction. This is not the case for other operations*, [2] provides us with protocols to securely compute those building block operations that we can compose: *secure matrix multiplication*, *truncation* in order to deal with real numbers and not only integers and *secure covariance matrix inversion*. Choosing the right covariance kernel and using the secure protocols mentioned above, we manage to compute the calculations of (1) and (2) on secret sharings.

We implemented our method on the same experimental setting as described in [1], which consists of three mountain cars with different masses. We managed to share data between the cars successfully. Although there is no loss in accuracy, the computation is heavy as the bottleneck is the matrix inversion appearing in equations (1) and (2). This being said the covariance matrix inversion is performed *only once* before any prediction on a new input is made. The prediction itself is actually relatively light in terms of computation but the amount of communication required is significant.

We investigated one potential solution to lighten the communication load: the introduction of homomorphic encryptions (HE) in our protocols. We leave the implementation for further work.

References

1. Timothy Verstraeten, Ann Nowe. Reinforcement learning for fleet applications using coregionalized Gaussian processes (2018).
2. M De Cock, R Dowsley, A Nascimento, S Newman. Fast, privacy preserving linear regression over distributed datasets based on pre-distributed data (2015).