How to Learn with an Untrusted Intermediary

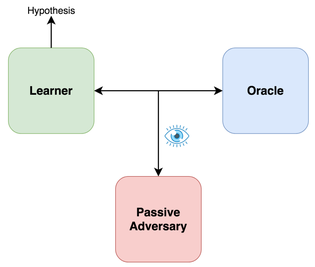
Ari Karchmer (on joint work with Ran Canetti)

Boston University

Covert Learning (CL)

Covert Learning twists the usual learning with membership queries setting: every example obtained by the learner is also obtained by a computationally bounded adversary. The high level goal is for the learner to construct queries that are useful to herself, but are unintelligible to any adversary—at least without knowing the "line of reasoning" that led to their construction.

The basic setting

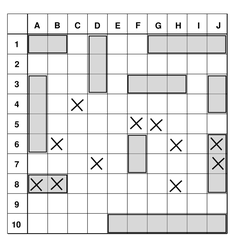


Intuition: battleship with an eavesdropper

▷ Can you play battleship against Alice while Eve is in the room listening (you want to prevent Eve from gaining an understanding of where Alice's ships are)?

▷ Bonus: if you have special information about the locations of Alice's ships, how do you prevent Alice from realizing this and moving her ships?

▷ What is your usual battleship strategy? Does employing that strategy hide the location of the ships from Eve?



▷ You can play battleship by running the KM algorithm...that doesn't work either.

▷ Hint: Can you do it with random missles? Pseudorandom missles?

Our goals

▷ **Learning**: Given access to a membership oracle, the learner is guaranteed to acquire some satisfactory hypothesis for the concept.

▷ **Hypothesis-hiding**: No passive adversary learns anything about the learner's output hypothesis or even the given hypothesis *class*.

▷ **Concept-hiding**: No passive adversary learns *anything* about the concept (aside some random examples).

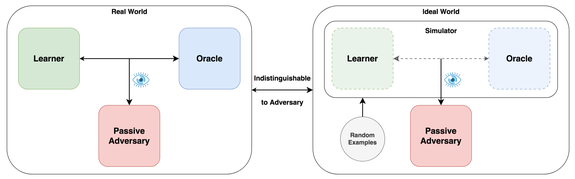
Modelling choices

▷ We define learning for a *collection* of hypothesis classes (rather than the single fixed class, as is customary in learning theory).

▷ This naturally models intent and prior knowledge used to select hypothesis class: We require learning guarantees for every hypothesis class in the collection: the learner gets to select a hypothesis class in the collection that must be learned.

Simulation of the interaction

To model the hiding, we employ the simulation paradigm:



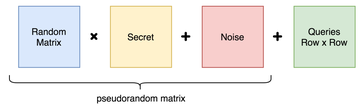
▷ There exists a simulator that generates an (ideal) transcript of the (real) interaction between the learner and the concept oracle, with access to random examples to the concept, but not further access to the concept oracle. Furthermore, the simulator should operate without knowledge of the learner's target hypothesis class.

▷ The simulated transcript should be indistinguishable from a real transcript, to a (polynomial time) adversary that has access to auxiliary information on the concept, and may even fully choose the concept and hypothesis class.

Covert Learning of Fourier coeffs.

How do we do it?

▷ We use a new technique called "query masking." Assuming hardness of the LPN assumption, we produce one-time-pads for each desired query. The resulting sum is then a "masked query." The joint distribution over many queries is then pseudorandom.



▷ We run "Goldreich-Levin" queries under the masking technique.

▷ Knowing the LPN secrets used for the masks, we can "unmask" the responses to the masked GL queries.

▷ The analysis involves demonstrating that the unmasked responses are noticeably correlated with $O(log\ n)$-degree parities whenever the original concept has a "heavy" Fourier coefficient on that parity.

▷ We can thus collect the heavy $O(log\ n)$-degree Fourier coefficients.

A "real world" application: model extraction attacks

▷ In a model extraction attack, an adversary interacts with a query interface to a ML model, attempting to obtain enough information to reverse engineer the underlying model.

▷ In one main type of defense that has been proposed, MLaaS providers monitor the queries submitted by a user, to decide when a client is benign (i.e. using the query interface in an honest way), or malicious (is attempting to reverse engineer the model).

▷ The Covert Learning model provides a framework for studying the viability of query monitoring defenses. Membership query learning algorithms under the Covert Learning model can be seen to circumvent such defenses.

▷ The Covert Learning hiding guarantees prevent any efficient eavesdropper (in this case, any efficient extraction monitors) from using a learner's (extractor's) queries to gather information about the concept (the MLaaS model), or the learner's resulting hypothesis (the extractor's reverse engineering). This raises concerns about the efficacy of query monitor defenses.

Our full paper and related works

We are inspired by and therefore highlight two recent works that explore related settings to ours. Check out their work!

▷ Cryptographic Sensing by Ishai, Kushilevitz, Ostrovsky and Sahai (Crypto 2019)

▷ Interactive Proofs for Verifying Machine Learning by Goldwasser, Rothblum, Shafer and Yehudayoff (ITCS 2021)

See our full paper at ia.cr/2021/764

Summary of results

We formally define the new learning models and construct algorithms for salient learning tasks within the models. Assuming hardness of the Learning Parity with Noise (LPN) assumption, we show:

▷ Covert learning for a noisy parity problem.

▷ Covert learning for $O(log\ n)$-degree "heavy" Fourier coefficients of any function $f : \{0,1\}^n \to \{-1,1\}$.

▷ Covert learning of $poly(n)$ size decision trees.

▷ Covert and *verifiable* learning for low-degree Fourier coefficients and decision trees (this algorithm works for battleship!).

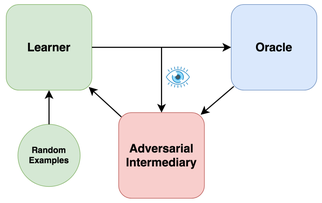
▷ The first *public* verifiable PAC learning protocol.

▷ A transformation of our covert learning algorithm for noisy parities to a cryptographic key exchange protocol from LPN.

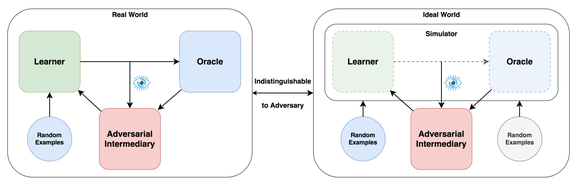
Augmenting CL with verifiability

▷ Stronger adversary: can eavesdrop *and* tamper with oracle responses.

▷ Either to learn more about the hypothesis/concept, or to deceive the learner.



▷ **Verifiability**: Even if the intermediary behaves maliciously, the learner is guaranteed to acquire a satisfactory model, *as long as she does not decide to abort altogether.*



▷ For any intermediary, there exists a simulator (given random examples) such that no external adversary (who may even choose the concept and hypothesis class), can distinguish whether the intermediary is interacting with a real learner or the simulator.

▷ **How we do it:** We wrap the CL algorithms with a loop that randomly executes a"test" phase or a "learning" phase. Crucially, the test and learning phases are comp. indistinguishable. It follows that the intermediary cannot reliably lie on the learning phase (without breaking LPN).

Acknowledgments

The authors would like to thank Shafi Goldwasser and Ronitt Rubinfeld for very helpful discussions on the model and its motivation.