Part 1

The following is an abstraction of a problem. For more context on application, a brief description is attached below.

Description

The setting An adversary is given a finite set of functions \mathcal{F} where every element f_i\in\mathcal{F} is a mapping from a set X to a set L. \mathcal{F}. The adversary is also given a prior distribution, \mathcal{D}, over F.

The game A function f^ *is drawn from \mathcal{D}. The adversary does not know f^*.

Now there are n rounds. In each round i, the adversary selects any x_i\in X and is told f^*(x_i).

After the n'th round, the adversary attempts to guess f^*.

Example X:={(1,1),(0,1),(0,0),(1,0)} L:={0,1} \mathcal{F}:={\text{AND, OR, NAND, XOR}} \mathcal{D} is uniform over F. n=1

A valid policy for the adversary is to query (1,1). If f^*(1,1)=1, choose either OR or AND according to a cointoss. If f^*(1,1)=0 choose NAND or XOR in the same way.

The success rate of this policy if \frac{1}{2} which is better than complete random guessing which yields \frac{1}{4}.

This is a relatively basic example with small sets. In general, |L|,n<<|F|,|X|. One can also reasonably assume \mathcal{D} is power law distributed.

Question

- Is there existing literature which studies this setting? Information theory and computational learning theory seem like the most obvious areas to investigate.
- Given a setting, how much can the adversary learn from the best possible policy? One way of measuring this is the reduction of entropy from the prior distribution to the posterior belief after the last query as a function of n (see part 2 for some more ideas)
- Assuming the adversary is polynomially bounded (runs in probabilistic polynomial time), what is the best policy they can find given the setting?

The actual application

This question comes up when considering doing computation on private information. Feedback from this computation may degrade the privacy and this question aims to quantify to what degree privacy is degraded. More particularly, this problem comes up in a blockchain setting in which a transaction must be ordered in a block along with other transactions. This process requires doing computation on these transactions. In many cases the ordering of transactions within the block has an impact on how a block fares according to some objective function. In order to improve upon brute force search over all orderings, additional computation on private transactions may be required.

A transaction in blockchain can be considered a function which takes as input the state of a VM and outputs the future state (although the input and output can be generalised). In this case \mathcal{F} is the set of all transactions (finite because of gas limits), X is the set of all possible VM states and L are the bits of information leaked (e.g. boolean for success/revert of the transaction). One can simplify the problem by considering \mathcal{F} the set of all swaps, with each swap parameterised by pool, size and slippage limit.

Edit

- The most concrete direction I have in mind at the moment is looking at bounds on concept learning with membership queries in computational learning theory (see this or this) and combining that with results from PAC-Bayes which is PAC learning with priors.

Part 2

Now we layer on some economics.