We propose a task-based approach for learning probabilistic ML models in the loop of stochastic optimization.

Introduction

Predictive algorithms often operate within some larger process, but are trained on criteria unrelated to this process. Standard image classification treats all mistakes as equal (via 0.1 loss), but the wrong kind of mistake could lead to undesirable driving behavior. We train a model not (solely) for predictive accuracy, but to minimize the task-based objective we ultimately care about.

Related Work

Bengio [1] uses task-based learning in a deterministic setting by tuning a financial price prediction model based on returns from a hedging strategy that employs it. We extend this work to a stochastic setting by optimizing objectives governed by a random process [2].

Setting: Stochastic Optimization

Stochastic optimization makes decisions under uncertainty by optimizing objectives governed by a random process [2].

Standard ML Approaches

Standard approaches to ML in stochastic optimization are:

1) Traditional model learning. Model conditional distribution \( p(y|x) \) by learning distribution parameters \( \theta \).

2) Model-free policy optimization: Map directly from inputs \( x \) to actions \( z \). Forgo learning model of \( y \).

We offer an intermediate approach where we both learn a model of \( y \) and adjust model parameters with respect to \( z \).

Our Method

Our model-based approach incorporates knowledge of the final task. We provide a general framework for adjusting model parameters in stochastic optimization to optimize closed-loop performance of the resulting system.

Our method chooses parameters \( \theta \) for \( y \) to minimize task loss:

\[
\min_x L(\theta) = E_{y \sim p(y|x)} \left[ f(x, y, z^*(x; \theta)) \right]
\]

where \( z^*(x; \theta) \) are the optimal actions w.r.t. our predictions, i.e.,

\[
z^*(x; \theta) = \arg\min_{z} E_{y \sim p(y|x)} \left[ f(x, y, z | \theta) \right]
\]

(with constraints omitted above for simplicity of illustration).

Algorithm

input: \( D \) // ability to sample from true, unknown distribution
initialize: \( \theta \) // initial distribution parameters
for \( t = 1, \ldots, T \) do
sample \( x, y \) \( \sim D \)
compute \( z^*(x; \theta) \) via Equation (*) (with constraints)
// stop in violated constraint or objective
if \( \exists i: g_i(y, z^*(x; \theta)) \geq 0 \) then
update \( \theta \) with \( \frac{\partial}{\partial \theta} L(\theta, x, y) \)
else
update \( \theta \) with \( \frac{\partial}{\partial \theta} L(\theta, x, y) \)
end if
end for

Technical Challenge: Argmin Differentiation

The gradient of the objective depends on the argmin result \( z^*(x; \theta) \):

\[
\Delta L = \nabla L \Delta z^* = \frac{\partial}{\partial \theta} \arg\min_z \left[ f(x, y, z | \theta) \right]
\]

To obtain the gradient, we write the KKT optimality conditions of (*). Assuming convexity allows us to replace the general equality constraints \( h(x) = 0 \) with the linear constraints \( Ax = b \).

A point \( (x, z, \lambda) \) is a primal-dual optimal point if it satisfies

\[
E_{y \sim p(y|x)} \left[ f(x, y, z | \theta) \right] \leq 0 \quad A x = b \quad \lambda \geq 0 \quad \lambda \cdot E_{y \sim p(y|x)} [g(y)] = 0
\]

where expectations are over \( y \sim p(y|x) \), \( g \) is the vector of all inequality constraints, and the dependence on \( x \) and \( y \) is via \( f \) and \( g \).

Differentially these equations and applying the implicit function theorem yield linear equations we can solve to get the necessary Jacobians.

In practice, we use SQP to solve (*), finding \( z^*(x; \theta) \) via a solution for fast argmin differentiation in QP [3] and then taking derivatives through the quadratic approximation at this optimum.

Experiments

We outperform both traditional model learning and model-free policy optimization in terms of cost, the objective of actual interest in the closed-loop system.

Inventory stock problem: Order quantity \( z \) of a product to minimize costs over stochastic demand \( y \).

Generator scheduling: Schedule electricity generation \( z \) to minimize costs over stochastic demand \( y \).

Battery arbitrage: Schedule battery charge/discharge \( z \) to minimize costs over energy prices \( y \).

Our task-based model outperforms:

- Traditional model-based MLE in all but the realizable case, correcting for effects of model misspecification.
- Model-free policy optimizer, due to increased data efficiency.

Conclusions

We propose an end-to-end approach for learning machine learning models used within stochastic optimization. Our experiments indicate that our task-based model learning method outperforms both traditional MLE and “black-box” policy-optimizing methods with respect to task cost.

Future work includes an extension of this method to stochastic learning models with multiple rounds, and further to model predictive control and full reinforcement learning settings.

References