# Data Driven Optimization with Unknown Time-Varying Objective Functions

Robert Ravier[1], Jie Ding[2], and Vahid Tarokh[1]

[1]Department of Electrical and Computer Engineering, Duke University
[2]Department of Statistics, University of Minnesota

***Abstract*—Many real-life machine learning challenges can be modeled as a time-varying optimization problem that depends both on personal preferences as well as external factors. Given that these optimization scenarios can heavily depend on the past, and that many widely-used collections of objective functions can be parametrized by a subset of finite dimensional Euclidean space, we propose the novel methodology of simultaneously modeling both objective functions and external data to formulate optimization problems. In particular, we show via theoretical results that modeling objective functions by time series of their parameters is feasible for a wide class of parametrizations and objective functions. We also detail an algorithm that can be used to simultaneously learn appropriate pairs of models on both objective functions and data, and show the wide applicability of our algorithm in both synthetic and real data experiments.**

## I. INTRODUCTION

Recommendation problems represent a wide range of data-driven discoveries in human life. Though the Netflix problem is by far the most famous in the machine learning community, recommendation problems exist when sommeliers recommend beverages to customers, real estate agents recommends houses to prospective homeowners, investment companies recommend portfolios to investors, among many other possibilities. From a research perspective, much attention has been given to the *fixed*-objective scenario: given user preferences, a good recommendation system would yield good recommendations to the user. From an optimization perspective, a user's preferences can be modeled as some (potentially highly nonconvex) optimization function, while a recommendation system functions as a black box that gives a candidate solution, which can be subsequently evaluated by the user.

It is of immense practical interest, however, to also consider the *time-varying*-objective scenario. In practice, a user's interests may depend not only on the total content of their preference history but also on the *time* in which the evaluations are made. For example, a person may prefer heartier foods or warmer beverages in cold months and vice versa in hotter months. Referring back to the Netflix problem, a user who watched one particular combination of genres may be in the process of binge watching this particular combination or may want to switch to something unrelated to movies watched prior, or may get into a particular habit of occasionally trying genres that they historically do not like. From an optimization perspective, the user's objective function is both time-varying and *dependent on the past*. Thus, in order to ensure proper behavior, any such recommendation system/optimizer must be carried out in an online manner.

In the online optimization setting, at a given time $t_k$, an optimizer will compute a candidate optimum $x_{k+1}$ for time $t_{k+1}$ based solely on the information received up until time $t_k$. After computation, the true optimization function $f_{t_{k+1}}$ is revealed, and a loss is suffered. The loss depends on the application in mind. In many branches of online learning, we use losses of the form

$$\sum_{t=1}^{T} f_t(\hat{x}_t) - \left( min_x \sum_{t=1}^{T} f_t(x) \right) \tag{1}$$

where we seek to minimize the cumulative error of all possible choices of actions for each time step with respect to the best fixed cumulative action in hindsight. This form of *regret* is used in the online convex optimization literature, see [21], [24], [27], [28] for general detailed overviews of problems in this scenario and recent work; in particular, it is not hard to prove under general assumptions that the naive generalizations of standard descent algorithms are provably asymptotically optimal in the sense that they achieve optimal regret in the sense of Equation (1). For the course of this paper, we will instead compare our actions at a given time with the best possible action at that time: namely, at a particular time $t$, the *instantaneous regret* of our choice of $x_t$ is given by

$$f_t(\hat{x}_t) - \min_{x_t} f_t(x_t) \tag{2}$$

One can then define the *total regret* to be the sum of all of the instantaneous regrets. This definition is more suitable for the initial setting outlined above, where a fixed, time-invariant recommendation system would not perform optimally compared to one that could adapt in time. Research in this direction focuses on using different methods to, assuming that $x_{t-1}$ is the optimum of $f_{t-1}$, approximate $x_t$ using certain assumptions on both the regularity of the optimal trajectory as well as the objective functions in both space and time (see, e.g., [3], [14], [25] and the references therein).

In the spirit of the recommendation system example outlined earlier, we list a number of assumptions made throughout the remainder of the work.

- Rather than attempting to find the trajectory of a given series of computed optima, we instead assume that we are interested in predicting the trajectory of objective

functions in time. We do not, however, assume that the objective necessarily varies smoothly in time. We will be more precise about this later.
- Related to the previous assumption, we assume that we have access to an oracle that we can call at each time to globally optimize a given function at each time step. This is guaranteed, for example, in the convex setting by using standard descent methods.
- The exact optimization problem of interest at each time ultimately depends on data that is separate from the objective function.

In particular, we will take the novel approach of modeling the sequence of objective functions as a time series. From a purely theoretical perspective, this is feasible because a wide class of objective functions of interest can be parametrized as a potentially infinite dimensional vector. Namely, if $f : A \to \mathbb{R}$ is a continuous function, where $A$ is a subset of finite dimensional Euclidean space, then $f$ is uniquely determined by its values on a countable number of points; this is a direct consequence of the density of the rationals on the real line [23]. In particular, if one allows for countably infinite linear combinations of functions, the space of such continuous functions has countable dimension. In practice, one can even go further and assume that spaces of objective functions of interest are inherently finite dimensional. For example, the set of $L^p$ norms for $p > 1$ is parametrizable in one dimension, and many objective functions of interest are parametrizable in two dimensions [4]. Given that, under many circumstances of interest, external data can reasonably be assumed to be finite dimensional, we combine the above to make the reasonable assumption that the both the data and the objective functions can be *simultaneously* modeled via a (potentially high dimensional) vector time series. This allows us to add a wide array of literature of techniques to the study of the time-varying problem. See [7], [16] for non-exhaustive overviews. Note that, from a proper mathematical perspective, objective functions of interest may belong to some general Hilbert or Banach space, and time series of such functions have been studied in, e.g., [6]. For simplicity of presentation, in this paper, we will not utilize the full scope of these techniques and restrict ourselves to finite dimensional spaces, which is sufficient for a wide variety of purposes. In particular, we will see that under reasonable conditions, facts about time series of real-valued functions can be used to obtain analogous facts for time series of objective functions.

Though we will see that forecasting of objective functions can be done via forecasting of their parametrizations, there still remains the notion of accurately modeling the objective functions and data. In practice, the exact process generating both the data and the objective functions are not known; nevertheless, there are wide classes of models, e.g. neural networks, that are used in practice despite them not being the exact generating process. In this mis-specified case, it is of significant interest to learn the models from the collection of interest that best model the data and objective function up to any noise in the process. Numerous problems in online learning, such as the nonstochastic multiarmed bandit problem [2] can be reduced to a variant of the exponential weight algorithm for prediction with expert advice [9]. In particular we will see that minor modifications to this algorithm will allow us to adapt it for simultaneous learning of both the models behind the data and the objective functions.
Given that we allow for random fluctuations in time for both the objective functions and external data, regret is no longer a reasonable quantity of interest to minimize. Instead, under reasonable assumptions on the randomness, expected regret is more reasonable. Related to expected regret, we are particularly interested in accurate modeling and subsequent optimization in the asymptotic case, i.e. that our optimizations are *asymptotically efficient*. We make this more precise in the next section.

The remainder of the paper is organized as follows. In Section II, we discuss relevant preliminaries and formally define asymptotically efficient optimization. In Section III, we theoretically justify why Euclidean time series techniques can extend to those for objective functions. In Section IV, we detail our algorithm to achieve asymptotic efficiency in optimization. In Section V, we illustrate the efficacy of our algorithm by showing results for synthetic and real data experiments. In Section VI we make concluding remarks.

## II. Preliminaries

### A. Relevant Definitions

Recall that a time series on $\mathbb{R}$ is an $\mathbb{R}$-valued stochastic process indexed by time, which for this paper we will assume to be discrete. There are many problems that can be modeled using techniques for time series. Our focus will be restricted to forecasting: given a (possibly infinite) sequence $X_t, X_{t-1}, ...$, we would like to accurately predict $X_{t+1}$ on average up to noise. Note that many results in prediction can be extended to the case that we have an $\mathbb{R}^n$ valued time series for $n > 1$. In this paper we will focus on the real-valued case, though our methods can easily extend for the vector-valued case.

We will also use the notion of a *parametrization*.

**Definition 1.** *A* **parametrization** *of a metric space $\mathcal{X}$ is a continuous bijective function $f : A \to \mathcal{X}$ with continuous inverse (i.e. a homeomorphism) for a subset $A \subseteq \mathbb{R}^n$ .*

Parametrizations frequently show up in manifold theory for locally describing manifolds in terms of coordinates [17]. Similar ideas apply here. Parametrizations allow us to unambiguously refer to objective functions on a space of interest solely in terms of the corresponding vectors in $\mathbb{R}^n$. We will make use of a particular type of parametrization for our theoretical discussions.

**Definition 2.** *Let $(\mathcal{X}, d)$ be a metric space. A parametrization $f : \mathbb{R}^n \to \mathcal{X}$ is* **Lipschitz continuous** *if there exists a constant $C > 0$ such that, for all $x, y \in \mathbb{R}^n$*

$$d(f(x), f(y)) \leq C\|x - y\| \tag{3}$$

This is a direct extension of the definition of a Lipschitz function for target spaces that are not Euclidean. Parametrizations that satisfy Definition 2 thus satisfy a bound on the amount of distortion suffered by mapping from $\mathbb{R}^n$ to $\mathcal{X}$. We will see that this control is sufficient to establish approximation bounds on function predictions.

Finally, we make an abstract definition of an optimization problem that will suit our modeling purposes.

**Definition 3.** *Let $\mathcal{X}$ be a function space. An* **optimization problem** *of the form*

$$O(\phi) \text{ subject to } D$$

*is a triple $(\phi, D, O)$, where $\phi \in \mathcal{X}$ is a function from some subdomain of $\mathbb{R}^n \to \mathbb{R}$ in some space $\mathcal{X}$, $D \in \mathbb{R}^m$, and $O : (\mathcal{X}, D) \to \mathbb{R}$.*

In practice, $D$ represents relevant constraints on the optimization problem and $O$ will always be minimization or maximization with respect to the constraints defined by $D$.

### B. Asymptotically Efficient Optimization

The ultimate goal of this work is to define and propose a desirable condition for learning and subsequently solving optimization problems. Specifically, we would like to consider a notion of *asymptotic efficiency* in solving optimization problems. To illustrate what we mean by asymptotic efficiency, consider the following example from statistical learning.

**Example 1.** *Given data points $z_1, ..., z_n \in \mathcal{Z}$ where $\mathcal{Z}$ is the domain of observations. We are frequently interested in finding the data generating model that minimizes the out-of-sample loss for a given loss function $\ell : \mathcal{H}_n[\alpha] \times \mathcal{Z} \to \mathbb{R}$, where $\mathcal{H}_n$ is a collection of statistical model classes for n data-points. Out-of-sample loss is defined as follows. Define $\hat{\theta}_n[\alpha]$ by:*

$$\hat{\theta}_n[\alpha] = \operatorname*{argmin}_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(z_i, \theta[\alpha], \alpha)$$

*where $\theta$ belongs to some compact parameter space and $\alpha \in \mathcal{H}_n$. Then the out-of-sample loss is defined by*

$$L_n[\alpha] := \mathbb{E}\ell(\cdot, \hat{\theta}_n[\alpha], \alpha) = \int \ell(z, \hat{\theta}_n[\alpha], \alpha) p(z) dz$$

*where $p(z)$ is the true data-generating process. For independent data, such model selection is usually performed via cross-validation, where a training set determines the parameters of the model and the out-of-sample loss is approximated by using an dataset independently generated of the training data. There exist methods such as Akaike Information Criterion, Takeuchi Information Criterion and generalizations that under certain assumptions asymptotically yield the true out-of-sample loss [1], [12], [13], [26].*

We now make the following definition. Roughly speaking, we are given a class of *pairs* of models, where one element models the objective function whereas another element models the external data. We do not assume that the model pairs correctly characterize the true process. Assume without loss of generality that we are interested in minimization. We say that an optimization procedure is asymptotically efficient if the value computed by the procedure relative to the true optimization problem converges asymptotically to that of the best possible model pair in a given class. More precisely:

**Definition 4.** *Let $\{\mathbf{P_t}\} := (\phi_t, D_t, O)$ be an time-varying sequence of optimization problems. Without loss of generality assume that $\phi_t > 0$. Let $\mathcal{M}$ be a class of models $m = (m_\phi, m_D)$, where $m_\phi$ and $m_D$ are models for the objective function and data processes respectively. Let $F_m(t)$ for a model $m \in \mathcal{M}$ be a procedure that uses all optimization problems up to time $t$ to output a candidate optimizer $x_{t+1}$ of $\phi_{t+1}$. We say that a procedure $A_{\mathcal{M}}(t)$ for optimization problems is an* **asymptotically efficient minimizer** *if it outputs $\hat{x}_t$ such that*

$$\lim_{t\to\infty} \frac{\phi_{t+1}(\hat{x}_t)}{\min_{m\in\mathcal{M}} \phi_{t+1}(F_m(t))} \to 1.$$

It is natural to ask under what conditions can such an optimization procedure exist. In [12], [13], it was shown under certain assumptions, such asymptotic efficiency can be achieved if the objective function is fixed in time and the number of data points goes to infinity asymptotically. Adapting the result to the case where the objective function is time-varying is a direction of future interest.

## III. Modelling Time Series of Objectives by Time Series of Parameters

In the introduction, we outlined reasons why predicting objective functions via time series is useful. The goal of this section is to theoretically justify our proposal that modeling parameters will model objective functions by proxy. The results of this section are not technically complicated, but are nevertheless important. Our first result concerns bounds relating the accuracy of estimation of a time series of objective functions given the accuracy of estimation of the time series of the parameters.

**Lemma 1.** *Let $f : \mathbb{R}^n \to \mathcal{X}$ be a Lipschitz map between $\mathbb{R}^n$ and a metric space $(\mathcal{X}, d)$. Let $X_t$ be a time series with elements in $\mathbb{R}^n$, and let $\hat{X}_t$ be an estimate of $X_t$. Assume that $X_t - \hat{X}_t$ has finite p'th moment for $p > 1$. Then we have the following.*

$$\mathbb{E}d(f(X_t), f(\hat{X}_t))^p \leq C\mathbb{E}|X_t - \hat{X}_t|^p \tag{4}$$

The proof follows immediately from properties of expectations and Lipschitz continuity. In general, this bound is tight; an example of this is given in the case that $\mathcal{X}$ is a finite $n$-dimensional subspace of a Hilbert space of loss functions with basis $\{v_i\}$. If $\{e_i\}$ is an orthonormal basis of $R^n$, then the unitary transformation $U : \mathbb{R}^n \to \mathcal{X}$ that takes $e_i$ to $v_i$ is Lipschitz with constant 1 (because unitary operators preserve distances), and the expectations on both sides of the inequality in Theorem 1 are equal because of the Pythagorean theorem for Hilbert spaces. Similar results also extend to cases where the estimate of the time series of the parametrization is misspecified, though the bounds will also depend on the degree of inaccuracy.

Theorem 1 captures a natural statement: if an estimate of the parameters of a loss function is accurate on average, then so is the estimate of the loss function up to noise. It is easy to extend this fact to show that, under certain assumptions on the noise, we can develop Chernoff bounds that give estimates on how close a given estimate of an objective function is to the true objective function in probability. This, however, does not necessarily imply anything about locations or values of optima. If we assume that our objective function $f$ is an element of the Hilbert space $L_2(\mathbb{R})$ of square-integrable losses on $\mathbb{R}$, then it is easy to construct examples of objective functions arbitrarily close in $L_2$ whose optima have locations and values arbitrarily far away from those of $f$: to see this, we only have to pick a tent function whose spike is centered at a specified value with its width depending on its height. We can, however, restrict to a reasonable space of objective functions for which, while the optima can potentially have arbitrary location, the optimal values are reasonably close. Specifically, let $L_\infty(D)$ be the space of bounded functions on a subset $D \subseteq \mathbb{R}^n$ equipped with sup norm $\|f\|_\infty := \sup_{x \in D} |f(x)|$. With respect to the metric $d_\infty$ induced by this norm, any two functions $f, g \in L_\infty(D)$ satisfying $d_\infty(f, g) < \varepsilon$ necessarily have their global minimum (maximum) values differ by at most $\varepsilon$. Note that this space naturally includes Lipschitz functions on compact metric spaces, so the result applies for a wide class of objective functions of interest.

We remark that in general, we cannot extend the results to say that locations of global optima are also reasonably close, even if we restrict ourselves to Lipschitz functions with norm $\|f\| := \|f\|_\infty + L_f$, where $L_f$ is the infimum of all possible Lipschitz constants. In order to do this, we would need to assume lower bounds concerning the differences between the global maxima/minima with the other local maxima/minima[1] That such a result would hold is obvious. We can also get this by having some control over the second derivatives. More precisely:

**Theorem 1.** *Let $f : D \to \mathbb{R}$ be an $m$-strongly convex $C^2$ function defined on a compact subset $D \subseteq \mathbb{R}^n$. Such a function is defined by the inequality*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + m\|y - x\|_2^2$$

*for all $x, y \in D$. [8] Then for every $\varepsilon_1, \varepsilon_2 > 0$, we can find $\delta > 0$ such that $\|f - g\|_\infty < \delta$ implies that $|min_x f(x) - min_x g(x)| < \varepsilon_1$ and $\|argmin_x f(x) - argmin_x g(x)\| < \varepsilon_2$*

*Proof.* It is clear that taking $\delta = \varepsilon_1$ gives the condition on the minimum values. The validity of the argmin condition requires the strong convexity assumption. Assume that $x$ is the argmin of $f$. The assumption that $\|f - g\|_\infty < \delta$ implies that the argmin of $y$ must occur in the set $A := \{y \in D : |f(y) - \min_x f(x)| < \delta\}$. This set is necessarily connected as otherwise the smoothness would imply that $f$ has a local maxima, contradicting the convexity assumption. If we apply the strong convexity assumption around the argmin $x$, we see that

$$f(y) \geq f(x) + m\|y - x\|_2^2$$

as $\nabla f(x) = 0$. Basic manipulations show that, for $y \in A$, we necessarily have $\|y - x\| \leq \sqrt{\frac{\delta}{m}}$. Thus letting $\delta = \min(\varepsilon_1, m\varepsilon_2^2)$ gives the claim. □

To summarize the above discussion: if we have a time series of objective functions in a parametrized space with parametrization with reasonable bounds on distortion, we have guarantees in both high probability and expectation that our estimated objective function is accurate. Under further assumptions on our space of objective functions, we can prove that the optimal value of our objective functions is reasonably accurate, and we have guarantees on accuracy of the location of the optimal values. Thus time series techniques for Euclidean space can be adapted to estimate objective functions.

## IV. Modelling Time-Varying Optimization Problems

Though the previous section justified why a time series approach can successfully model time-varying optimization problems, we now propose a method to accurately model them in practice. As previously mentioned, the exact optimization problems we wish to predict will depend not only on the predicted objective function but our predictions of the data at any given time. In practice we will rarely know the true generating processes governing the data and objectives, so any algorithm we propose must allow for potentially incorrect models.

Assume that we have some finite collection $\mathcal{M}$ of models for both the objective function and data generating processes. At a given time $t = n$, we do not know the future data or objective, however a given pair of models (one for data, one for objective) will give the best performance with respect to the collection of estimating the future. We would like to asymptotically predict the best model as time goes to infinity in the sense of Definition 4. From a practical perspective, we desire such a method that achieves good performance in practice even for short time periods and classes of models that are not sufficiently rich.

We summarize the main algorithm of our paper in Algorithm 1. Our algorithm is a modification of the standard algorithm for expert learning with exponential weights. At each time, we use the previously observed data and objective functions as inputs to prediction algorithms $F_i(t - 1)$ that fit models to both the data generating and objective function generating processes using all data obtained up to time $t - 1$, predict the next optimization problem, and output the optimizer of the predicted problem. For each predicted optimizer, we obtain a loss $\phi_{i_0+t}(F_i(t - 1))$ by evaluating our prediction at the true objective function, and update the predicative weights on our models accordingly. We add an additional optional step for introducing new models: there are frequent scenarios where one would like to fit a model to given data but cannot do so until enough arrives. This step allows for gradual introduction of models by readjusting the weights of the predicative distribution at a given time of introduction using a *change parameter m*. Without this step, the algorithm reduces to the classical exponential weighting algorithm. The specific criteria for expansion is problem dependent and thus

[1] This is of little concern from a practical perspective given that many functions of interest can be highly nonconvex for which local optima can be more than sufficient.

**Algorithm 1** Expert Learning Algorithm for Modeling Data and Objectives

```
Input: Parameters $\eta > 0$, $0 < m < 1$ initial data $D_1, ..., D_{i_0}$, initial objectives $\phi_1, ..., \phi_{i_0}$,
Output: $p_t = [p_{t,1}, ..., p_{t,N}]$ (predicative distribution over the active objective/data model pairs
Initialize $w_{1,0} = ... = w_{k_0,0} = 1$, $w_{k,0} = 0$ for $k > k_0$
$M = k_0$
for $t = 1 \to T$ do
    if Received sufficient amount of objectives and data to fit model $M + 1$ then
        $w_{1,0} = (1 - m)w_{1,t-1}, ..., w_{M,t-1} = (1 - m)w_{M,t-1}$
        $w_{M+1,t-1} = m$
        $M = M + 1$
    end if
    Calculate the predicative distribution $p_{i,t} = w_{i,t-1} / \sum_{j=1}^{N} w_{j,t-1}$ for each $1 \le i \le N$
    Compute $F_i(t - 1)$ for each $1 \le i \le M$
    Observe optimization problem $(\phi_{i_0+t}, D_{i_0+t})$
    Compute $l_i = \exp(-\eta\phi_{i_0+t}(F_i(t - 1))$ for each $1 \le i \le M$
    Compute $w_{i,t} = p_{i,t-1} l_i$
end for
```

open-ended. The $\eta$ parameter is standard in the exponential weighting algorithm and is called the *learning rate*.

## V. Experiments

We now evaluate the behavior of our proposed algorithm on problems using both synthetic and real data. Given that our framework depends on black box global optimizers, we restrict our experiments to convex problems, though the algorithm can readily be applied to nonconvex problems. For completeness, we state necessary model definitions.

In this paper we will make use of two.

**Definition 5.** *An* **autoregressive** *process of lag $k$ with mean zero, denoted $AR(k)$, is a process for which there exist constants $a_1, ..., a_k$ satisfying some conditions such that*

$$X_t = \sum_{i=1}^{k} a_i X_{t-i} + \varepsilon_t,$$

*where $\varepsilon_t$ are i.i.d. Gaussian random variables with mean zero and variance 1.*

Specific conditions on the constants can be found in standard textbooks on time series.

**Definition 6.** *A* **moving average** *process of lag 1 with mean zero, denoted $MA(1)$, is a time series for which there exists a constant $|a| < 1$ such that*

$$X_t = \varepsilon_t + a\varepsilon_{t-1}$$

*where the $\varepsilon_t$ are i.i.d. Gaussian random variables with mean zero and variance 1.*

### A. Synthetic Data Experiment

Convex combinations of $L^1$ and squared $L^2$ losses are of immense importance in machine learning. We consider such a collection of loss functions: at time $t$, our loss function is of the form $\lambda_t \| \cdot \|_2^2 + (1 - \lambda_t) \| \cdot \|_1$, where $\lambda_t$ is generated by a reflected Brownian motion starting at $\lambda_1 = 0.5$, where the variance of each time step is 0.01. We specify that the $\lambda$ are always at least 0.1 and always at most 0.9. These losses are always convex, hence an oracle can optimize all of these losses exactly.

At each time $t$, we would like to select an $AR(k)$ model, for $1 \le k \le 20$ that best fits the above process. For each of these models of $\lambda$,, we then fit another $AR(k')$ model, $1 \le k' \le 20$ that best fits an $MA(1)$ model of data with zero mean and lag coefficient 0.8. At each time, we use each process to predict the value of $\lambda$, and then subsequently for each value of predicted $\lambda$ compute the predicted lags of the desired $AR$ process for the data. We start at time $t = 11$ with 11 data points, and go to time $t = 80$, adding a new data point at each time. After fitting each model, we observe the expected loss with respect to the true convex combination (note that in this case it is not hard to derive an analytical formula for the expected loss). Higher model lags are gradually added as enough observations become available to ensure accurate fitting. The optimization for this experiment was performed using Gurobi and YALMIP in Matlab [15], [19]. Keeping with notation, the objective function at $\phi_t$ is the true convex combination of expected $L_1$ and squared $L_2$ losses.

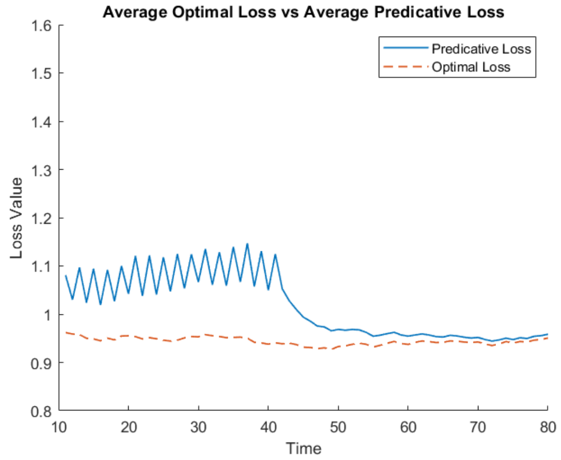

Fig. 1: Predicative vs Optimal Model Loss of the weighted $L^1$-$L^2$ experiment.

Figure 1 show the results of the average of the experiment repeated 50 times. In this figure, the change multiplier $m = .2$ and the tuning parameter $\eta = 2$. We that as more data is added, the average of the predicative losses converges to the average optimal loss for the pairs of model classes.

We also use this example to empirically analyze the effect of the choice of parameters of our algorithm affect convergence to optimum values. We first investigate the effect of choosing different tuning parameters. We use the data from the previous experiment with $\eta = 2$ but $m$ ranging between 0.05 and 0.5

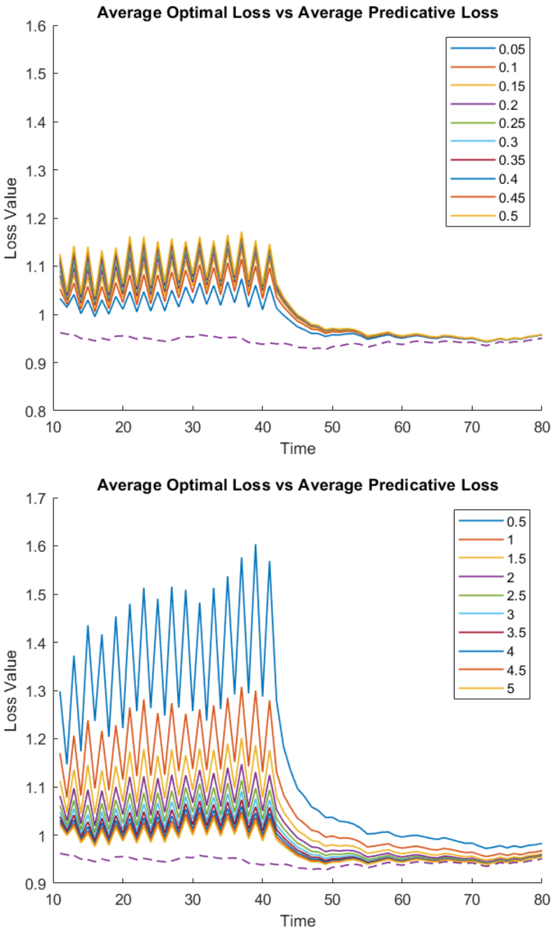

Fig. 2: Top: Convergence results for different change parameters. Bottom: Convergence results for different learning rates.

in increments of 0.05. The results can be found in the top of Figure 2. We see that, asymptotically, the precise value of the change parameter has negligible effect on convergence of the average predicative loss to the average optimal loss given by the dashed line. The figure directly below illustrates the same result but with $m = 0.2$ fixed and $\eta$ ranging between 0.5 and 5 in increments of 0.5. We again observe asymptotic convergence on average to the optimal loss, with stronger convergence for higher learning rates. These results are not surprising: lower learning rates make the predicative distribution closer to uniform, requiring more time to converge. Nevertheless, convergence is observed, albeit more weakly for the lowest learning rate.

### B. Real Data Experiment

We now consider our method on the real world problem of **portfolio recommendation**. Though there are a number of different ways to construct portfolios in both online and offline settings (see, e.g., [10], [20], [22], we restrict our attention to Markowitz portfolio theory. Given a collection of assets $z_1, ..., z_n$, the Markowitz optimal portfolio allocation is given by:

$$\underset{w \in \mathbb{R}^n}{\operatorname{argmin}}\, w^T \Sigma w - \lambda w^T \mu \tag{5}$$

$$w_1, ..., w_n \geq 0, \sum_{i=1}^{n} w_i = 1$$

where $\Sigma$ represents the covariance matrix for the returns of the assets, $\mu$ is the average return of each asset, and $\lambda > 0$ is a parameter encoding the tradeoff between expected returns and risk; optimizing with $\lambda = 0$ finds the portfolio with the least amount of risk. The above function is quadratic, hence there exists a global optimizer. For our purposes, $\Sigma, \mu$, and $\lambda$ will be time-varying.

Using our framework, we consider the following recommendation problem. In this scenario, a portfolio manager reaches out to his client every month (30 days) with a series of new portfolios, each of which is constructed built by estimation of their client's risk tolerance and subsequent optimizing of Equation (5) given different lookback periods on a given collection of assets to be between 15 and 90 days in increments of 15 for computing relevant means and covariances. The risk tolerance is estimated by a series of autoregressive processes on the risk tolerance with lags between 30 and 180 days in increments of 30[2].

Unbeknownst to the manager, the client evaluates the portfolio also via Equation (5), but with a 150 day lookback period and a risk generated by a recursion dependent on the past 240 days. The recursion is generated in this simulation by letting the risk parameter be .5 for the first 240 days followed by a recursion generated by multiplying a uniformly randomly generated point on the eight dimensional unit simplex by 0.75. To simulate uncertainty in risk at any particular time, each computed risk parameter is perturbed by independent exponentially distributed noise with parameter $\lambda = 10$. The client will only choose one portfolio at each 30 day period.

As data, we make use of the NYSE dataset used frequently in the portfolio optimization literature, a collection of 36 stock returns taken over a period of 22 years [5], [18]. We also add a risk-free asset that gives constant, low returns of 1% every 360 days compounded daily. The goal, as with the synthetic experiment, is to predict the best portfolio of those offered that optimizes the client's objective function without knowing that objective function in the future. We use MATLAB's built-in quadprog function using a slightly regularized covariance matrix to ensure numerical well-posedness of the quadratic problem. Due to numerical issues, computed optima may (very slightly) lie outside of the feasible set; we project these optima onto the feasible set by removing any (slightly) negative values followed by normalizing optima to sum to 1.

The results of this experiment are detailed in Figure 3 after running 50 experiments (each experiment corresponds to a different risk tolerance recursion) with learning parameter $\eta = 100$[3]. Our analysis starts 300 points in on the available return data, meaning that all models for the optimization

[2] Risk is only observed every 30 days

[3] This is needed as the values of the objectives had small magnitude, hence the distributions would otherwise be close to uniform

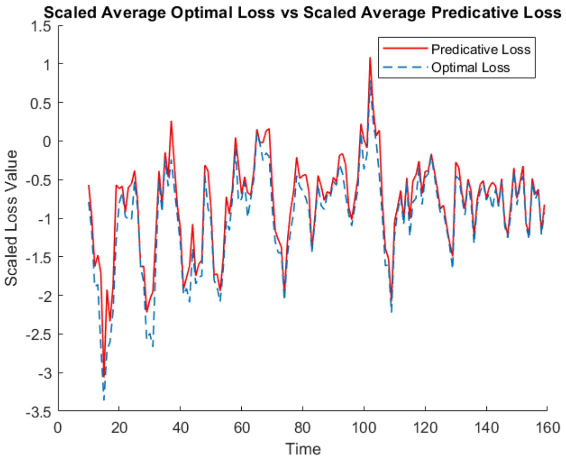

Fig. 3: Scaled Predicative vs Scaled Optimal Model Loss of the Portfolio Optimization Prediction Experiment.

problems are well-specified so no change multiplier is needed. The result is the same as in the synthetic case: with enough observations, the predicative loss asymptotically approaches that of the optimal loss. The loss in this experiment is scaled for visualization purposes.

## VI. Conclusion

In this paper we have proposed a novel methodology for time-varying optimization by simultaneously modeling objective functions and data relevant to the objective function. We showed that a large class of objective functions can be reasonably modeled in this way. We also detailed an algorithm for simultaneous prediction of data and objective function and showed experimentally in scenarios involving both synthetic and real data that predicative losses will converge asymptotically to optimal losses for given classes of objective functions.

There are many directions in which this line of research can evolve. From a theoretical perspective, it would be interesting to look at the problem from the context of functional time series, where we assume that the objective function is in some Hilbert or Banach space with countable dimension rather than a finite-dimensional space, as is done in [6]. These cases cover a wide variety of loss functions of interest, such as those in $L^2$ or Lipschitz losses. It is also of great theoretical and practical interest to investigate the case where we have multiple objective functions, which could be modeled as a vector time series [11]. This situation creates a notion of tradeoff, where we may have to favor one objective function over another after employing some scalarization procedure. This could also lead to the study of modeling manifold-valued time series for objective functions, for which geometric techniques could be employed.

From a practical perspective, we considered only limited classes of time series in the experiments, and it would be extremely interesting to see how other classes adapt to our outlined framework, particularly autoregressive integrated moving average (ARIMA) and generalized autoregressive conditional heteroskedasticity (GARCH) models for time series that are used in many applications. We will also investigate the behavior of neural networks in the time-varying objective scenario, for which many interesting questions could be posed.

## VII. Acknowledgements

This research was funded by the Defense Advanced Research Projects Agency (DARPA) under grant number FA8650-18-1-7837.

## Appendix

In this section we detail the expected error of fitting an $MA(1)$ model by an $AR(k)$ model. We consider the expected squared error, the expected absolute error can be dealt with similarly.

For an $MA_1$ process with lag parameters $a$ we have that $X_{t+1} = \varepsilon_{t+1} + a\varepsilon_t$. If we estimate the $MA(1)$ process via an $AR(k)$ process, we see that the predicted $\hat{X}_{t+1}$ satisfies

$$\hat{X}_{t+1} = \sum_{i=1}^{k} b_i(\varepsilon_{t-i+1} + a\varepsilon_{t-i}),$$

which, after regrouping terms, we have

$$X_{t+1} - \hat{X}_{t+1} = (a - b_1)\varepsilon_t - \left(\sum_{i=1}^{k}(b_i a + b_{i+1})\varepsilon_{t-i}\right) - b_k a\varepsilon_k,$$

Since the $\varepsilon$ terms are independent and mean zero, standard algebraic manipulations yield

$$\mathbb{E}(X_{t+1} - \hat{X}_{t+1})^2 = (a - b_1)^2 + \sum_{i=1}^{k}(ab_i + b_{i+1})^2 + a^2 b_k^2$$

## References

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, pages 199–213. Springer, 1998.

[2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[3] B. Bank. *Non-linear parametric optimization*, volume 58. Birkhauser, 1982.

[4] J. T. Barron. A more general robust loss function. *arXiv preprint arXiv:1701.03077*, 2017.

[5] A. Borodin, R. El-Yaniv, and V. Gogan. Can we learn to beat the best stock. In *Advances in Neural Information Processing Systems*, pages 345–352, 2004.

[6] D. Bosq. *Linear processes in function spaces: theory and applications*, volume 149. Springer Science & Business Media, 2012.

[7] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[8] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[9] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[10] T. M. Cover. Universal portfolios. In *The Kelly Capital Growth Investment Criterion: Theory and Practice*, pages 181–209. World Scientific, 2011.

[11] K. Deb. Multi-objective optimization. In *Search methodologies*, pages 403–449. Springer, 2014.

[12] J. Ding, E. Diao, J. Zhou, and V. Tarokh. Approaching the predicative limit of learning, 2016.

[13] J. Ding, E. Diao, J. Zhou, and V. Tarokh. A penalized method for the predictive limit of learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4414–4418. IEEE, 2018.

[14] J. Guddat, F. G. Vazquez, and H. T. Jongen. *Parametric optimization: singularities, pathfollowing and jumps*. Springer, 1990.

[15] L. Gurobi Optimization. Gurobi optimizer reference manual, 2018.

[16] J. D. Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, NJ, 1994.

[17] J. M. Lee. Smooth manifolds. In *Introduction to Smooth Manifolds*, pages 1–29. Springer, 2003.

[18] B. Li and S. C. Hoi. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):35, 2014.

[19] J. Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.

[20] H. Markowitz. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952.

[21] M. J. Neely and H. Yu. Online convex optimization with time-varying constraints. *arXiv preprint arXiv:1702.04783*, 2017.

[22] A. Ponsich, A. L. Jaimes, and C. A. C. Coello. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation*, 17(3):321–344, 2013.

[23] W. Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1976.

[24] S. Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[25] A. Simonetto and E. Dall'Anese. Prediction-correction algorithms for time-varying constrained optimization. *IEEE Transactions on Signal Processing*, 65(20):5481–5494, 2017.

[26] K. Takeuchi. The distribution of information statistics and the criterion of goodness of fit of models. *Mathematical Science*, 153:12–18, 1976.

[27] H. Yu, M. Neely, and X. Wei. Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems*, pages 1428–1438, 2017.

[28] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.