PREDICTOR-CORRECTOR ALGORITHMS FOR STOCHASTIC OPTIMIZATION UNDER GRADUAL DISTRIBUTION SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-varying stochastic optimization problems frequently arise in machine learning practice (*e.g.* gradual domain shift, object tracking, strategic classification). Often, the underlying process that drives the distribution shift is continuous in nature. We exploit this underlying continuity by developing predictor-corrector algorithms for time-varying stochastic optimization that *anticipates* changes in the underlying data generating process through a predictor-corrector term in the update rule. The key challenge is the estimation of the predictor-corrector term; a naive approach based on sample-average approximation may lead to non-convergence. We develop a general moving-average based method to estimate the predictor-corrector term and provide error bounds for the iterates, both in presence of pure and noisy access to the queries from the relevant derivatives of the loss function. Furthermore, we show (theoretically and empirically in several examples) that our method outperforms non-predictor corrector methods that do not anticipate changes in the data generating process.

1 INTRODUCTION

Stochastic optimization is a basic problem in modern machine learning (ML) theory and practice. Although there is a voluminous literature on stochastic optimization (Agarwal et al., 2014; Moulines & Bach, 2011; Bottou, 2003; 2012; Bottou & Bousquet, 2007), most prior works consider a *time-invariant* stochastic optimization problem in which the data generating distribution is not changing over time. However, there is an abundance of real examples in which the underlying optimization problem is time-varying which can be broadly divided into two categories: the first kind arises due to exogeneous variation in the data generating process. A concrete example is the object tracking problem in which an observer observes (noisy) signals regarding the position of a moving object, and the goal is inferring the trajectory of the object. The second kind of time-varying optimization problem arises due to endogeneous variation in the data generating process. Examples here include strategic classification (Dong et al., 2018; Hardt et al., 2016) and performative prediction (Perdomo et al., 2020; Mendler-Dünner et al., 2020; Brown et al., 2022). Although there are a few recent papers on time-varying stochastic optimization (e.g., Cutler et al. (2021); Nonhoff & Müller (2020); Dixit et al. (2019; 2018)), they model the temporal drift as discrete, precluding them from exploiting the smoothness in the drift. This leads to worse asymptotic tracking error, depending on the magnitude of the temporal drift of the optimal solution, (*e.g.* see Popkov (2005). Zavlanos et al. (2012), Zhang et al. (2009), Ling & Ribeiro (2013) and references therein).

In this paper, we focus on time-varying stochastic optimization problems in which the temporal drift is driven by a *continuous-time* process. We leverage the smoothness of the temporal drift to develop predictor-corrector (PC) methods for stochastic optimization that *anticipates* future changes in the data generating process to improve the asymptotic tracking error (ATE) (see Definition 2.1). The main benefit of such methods is smaller tracking error (compared to other stochastic optimization algorithms that does not leverage smoothness of the temporal drift). The main challenge in stochastic time-varying optimization is property accounting for the temporal drift through a PC term in the update rule. The noise in the stochastic setting makes naive estimates of the PC term unstable, and may lead to non-convergence. One of our main contributions is developing a general way of estimating the PC term.

We complement the methodological contributions with theoretical results that show PC stochastic optimization algorithms inherit the benefits of their non-stochastic counterparts for time-varying problems. In particular, we show that PC stochastic optimization algorithms have smaller asymptotic tracking error (ATE) compared to their non-PC stochastic optimization algorithm. We also demonstrate the superiority of PC algorithms empirically in a time-varying linear regression and target tracking applications. The rest of the paper is organized as follows: In Section 2, we present the algorithm and highlight the difference between predictor-corrector based algorithm and time non-adaptive algorithms like simple gradient descent. We further present theories regarding the bound on ATE of the algorithms of both kinds. In Section 4, we present three concrete instances of time-varying stochastic optimization problems driven by underlying gradual distributions shifts and derive the details of PC algorithms for these problems.

2 Framework and Algorithm

In a typical learning problem, we have n samples $X_1, \ldots, X_n \sim P$ and based on the data, we estimate some parametric (or non-parametric) functional of the underlying distribution $\theta^\star = \nu(P)$ by minimizing some loss function $\ell(\theta, X)$. A primary assumption necessary for consistent estimation is that the corresponding risk function $\mathcal{R}(\theta) \triangleq \mathbb{E}[\ell(X, \theta)]$ is uniquely minimized at $\theta^\star$. In a non-stationary time varying framework, we assume that the data generating distribution $P \equiv P_t$ changes with time and so does the parameter of interest $\theta_t^\star = \nu(P_t)$ along with the (time-varying) risk function $\mathcal{R}(\theta, t) \triangleq \mathbb{E}_{P_t}[\ell(X, \theta)]$.

As a concrete example, consider the object tracking problem studied in Patra et al. (2020): suppose we have installed n sensors at position $x_1, \ldots, x_n$ (which remain fixed over time) and let $\theta_t^\star$ is the location of the target object at time t. At each time, we get some noisy feedback from the sensors regarding the position of the target, i.e.

$$y_{i,t} = \|x_i - \theta_t^\star\|^2 + \epsilon_{it} \text{ for } 1 \le i \le n,$$

where ϵ_{it}'s are iid (over i and t) with mean zero and variance τ^2. Denote by $\mathbf{y}_t \in \mathbb{R}^n$ to be the vector of observations from n sensors at time t. A natural approach to estimate $\theta_t^\star$ is to minimize squared error loss $\ell(\mathbf{y}_t, \theta) = \sum_{i=1}^n (y_{i,t} - \|x_i - \theta\|^2)^2$, which yields the risk function

$$\mathcal{R}(\theta, t) = \sum_{i=1}^n \mathbb{E}[(y_{i,t} - \|x_i - \theta\|^2)] = n\tau^2 + \sum_{i=1}^n (\|x_i - \theta_t^\star\|^2 - \|x_i - \theta\|^2)^2.$$

From the risk function, it is immediate that under very mild assumptions on $x_1, \ldots, x_n$ (i.e. they are in general position) we have

$$\theta^\star(t) = \arg\min_\theta \mathcal{R}(\theta, t),\ t \ge 0\,. \tag{2.1}$$

We will use the notations $\theta^\star(t)$ and $\theta_t^\star$ interchangeably to denote the same thing. From the above formulation, it is immediate that we are merely observing one sample/incident at each time point. Therefore if $\theta_t^\star$ behaves erratically over time, there is no hope to learn the evolution pattern from the data. Therefore, it is imperative to assume some smoothness on target function $\theta_t^\star$. Our proposed method exploits the smoothness of $\theta_t^\star$ to improve it's estimation over time.

We now formulate the problem: we assume that the underlying distribution function $\{P_t\}$ changes continuously with time t. As statisticians, we query/intervene the model at discrete time steps (i.e. say at time $\{kh\}_{k\in\mathbb{N}}$ where h is the time step which controls the frequency of query) and observe n samples $\{X_{it}\}_{1\le i\le n}$ from the distribution at that time. As a consequence, we have a sequential batch of data using which we aim to estimate $\theta_{kh}^\star$, i.e. the parameters at the time of query. Note that to estimate the parameter at time t one may use all previous data points. We evaluate the quality of our estimator using the asymptotic tracking error (ATE) defined below.

Definition 2.1 (Asymptotic tracking error (ATE))**.** *Let the true dynamic parameter $\{\theta_t^\star,\ t \ge 0\}$ be sequentially estimated as $\{\hat\theta_{kh}, k \in \mathbb{N}\}$ over the time grid $\{kh : k \in \mathbb{N}\}$, where $h > 0$ is the time step. Then the asymptotic tracking error is defined as*

$$\textit{ATE}(\theta) = \limsup_{k\to\infty} \|\hat\theta_{kh} - \theta_{kh}^\star\|_2\,.$$

As mentioned in the Introduction, we here compare performance of a time-adjusted gradient descent method to a time-unadjusted one. As will be evident, both the methods require evaluation of certain derivatives of the risk function.

To motivate the predictor-corrector (PC) algorithm, we start by deriving the prediction term when the optimizer has access to the exact gradients of the (time-varying) cost function. The optimality of $\theta_t^\star$ implies

$$g(t) = \nabla_\theta \mathcal{R}(\theta_t^\star, t) = 0 \text{ for all } t. \tag{2.2}$$

Thus $g'(t) = 0$; *i.e.*

$$\nabla_{\theta\theta}\mathcal{R}(\theta_t^\star, t)\dot{\theta}_t^\star + \nabla_{\theta t}\mathcal{R}(\theta_t^\star, t) = 0, \tag{2.3}$$

where $\dot{\theta}_t^\star$ is the temporal drift of the optimal solution $\theta_t^\star$. We see that $\theta_t^\star$ satisfies the ODE:

$$\dot{\theta}_t^\star = -\nabla_{\theta\theta}^{-1}\mathcal{R}(\theta_t^\star, t)\nabla_{\theta t}\mathcal{R}(\theta_t^\star, t). \tag{2.4}$$

We interpret the right side of this ODE as a *prediction* of the change in $\theta_t^\star$. This suggests modifying the update rule of stochastic optimization algorithms to account for the predicted change in $\theta_t^\star$. This leads to the update rule

$$\hat{\theta}_{(k+1)h} = \hat{\theta}_{kh} - \eta\hat{\nabla}_\theta\mathcal{R}(\hat{\theta}_{kh}, kh) - h\{\hat{\nabla}_{\theta\theta}\mathcal{R}(\hat{\theta}_{kh}, kh)\}^{-1}\hat{\nabla}_{\theta t}\mathcal{R}(\hat{\theta}_{kh}, kh),$$

where $\eta > 0$ is a learning rate, h is a (time) step, and $\hat{\nabla}_\theta\mathcal{R}(\theta, t)$, $\hat{\nabla}_{\theta\theta}\mathcal{R}(\theta, t)$, $\hat{\nabla}_{\theta t}\mathcal{R}(\theta, t)$ are estimates of $\nabla_\theta\mathcal{R}(\theta, t)$, $\nabla_{\theta\theta}\mathcal{R}(\theta, t)$, $\nabla_{\theta t}\mathcal{R}(\theta, t)$ respectively. We summarize the stochastic PC algorithm in Algorithm 1.

Algorithm 1 Stochastic predictor-corrector based method

Require: step size $h > 0$, learning rate $\eta > 0$, estimated gradients $\hat{\nabla}_\theta\mathcal{R}(\theta, kh)$, Hessians $\hat{\nabla}_{\theta\theta}\mathcal{R}(\theta, kh)$ and time derivative of the gradient $\hat{\nabla}_{\theta t}\mathcal{R}(\theta, kh)$ for $k \in \mathbb{N}$

1: Initialize $\hat{\theta}_0$ at some value.
2: **for** $k \geq 0$ **do**
3: Update $\hat{\theta}_{(k+1)h} = \hat{\theta}_{kh} - \eta\hat{\nabla}_\theta\mathcal{R}(\hat{\theta}_{kh}, kh) - h\{\hat{\nabla}_{\theta\theta}\mathcal{R}(\hat{\theta}_{kh}, kh)\}^{-1}\hat{\nabla}_{\theta t}\mathcal{R}(\hat{\theta}_{kh}, kh)$
4: **end for**

To motivate the benefit of accounting for the predicted change in $\theta_t^\star$, we present here a brief comparison of the tracking error of the PC algorithm with simple stochastic gradient descent without the correction term. For simplicity of exposition, we study the tracking error of these two algorithms in the non-stochastic setting. We begin by showing a lower bound on the tracking error of gradient descent. We suspect this result is known to experts, but it does not appear (to the best of our knowledge) in the literature.

Theorem 2.2 (Lower bound)**.** *There exists a $\mathcal{R}(\theta, t)$ that satisfies Assumption 3.1 such that the gradient descent algorithm with time step size $h > 0$ and learning rate $\eta > 0$ satisfies the following: there exists a $c > 0$ such that*

$$\liminf_{k\to\infty} \left\|\hat{\theta}_{kh} - \theta_{kh}^\star\right\|_2 \geq c\frac{h}{\eta}.$$

As we shall see in the subsequent section (see Theorem 3.3), the tracking error of the PC algorithm in the non-stochastic setting is $O(\frac{L''h^2}{M\eta})$. We restate this special case of Theorem 3.3 here for the reader's convenience.

Corollary 2.3. *(Non-stochastic version of Theorem 3.3) Assume that 3.1 and 3.2 holds and that $\mathcal{R}(\theta, t)$ is twice differentiable in both co-ordinates. Then for the special case, when we have access to the true gradients, the sequence of estimates in Algorithm 1 with step size $h > 0$ and learning rate η satisfies:*

$$\limsup_{k\to\infty} \left\|\hat{\theta}_{kh} - \theta_{kh}^*\right\| \leq \frac{L''h^2}{M\eta + hM'},$$

where $M' = \sup_{\theta,t} \|\{\nabla_{\theta\theta}\mathcal{R}(\theta, t)\}^{-1}\nabla_{\theta t}\mathcal{R}(\theta, t)\|_{op}$, $M = \sup_{t\geq 0} \|\nabla_\theta\mathcal{R}(\theta, t)\|$ and $L'' = \max_{t\geq 0} \frac{1}{2}\|\ddot{\theta}(t)\|$ are finite.

To compare the rates for the PC algorithm and gradient descent, we set a learning rate such that $h/\eta \to 0$ as $h \to 0$. We see from Theorem 2.2 that the ATE for gradient descent ATE cannot converge faster than h/η. By comparison, the ATE for PC algorithm in the non-stochastic setting (see Theorem 2.3) converges to zero at rate h^2/η (since it holds $\frac{L''h^2}{M\eta + hM'} \asymp \frac{L''h^2}{M\eta}$ as long as $h/\eta \to 0$),

which a faster rate than h/η. Hence, we conclude that the ATE for predictor-corrector update converges at a faster rate than the ATE for gradient descent update. A high-level reasoning for such a distinction is the following: the gradient descent method does not consider or calibrate for the underlying smoothness in $\theta^\star(t)$ while performing the time updates, whereas predictor-corrector calibrates for the smoothness in the time update by adjusting the term $-h\{\nabla_{\theta\theta}\mathcal{R}(\hat{\theta}_t, t)\}^{-1}\nabla_{\theta t}\mathcal{R}(\hat{\theta}_t, t)$.

One of the main challenge in implementing the stochastic PC algorithm is obtain estimates of $\nabla_\theta\mathcal{R}(\theta, t)$, $\nabla_{\theta\theta}\mathcal{R}(\theta, t)$, $\nabla_{\theta t}\mathcal{R}(\theta, t)$:

1. **Estimation of gradient:** $\nabla_\theta\mathcal{R}(\theta, t)$ also appears in the SGD update rule; it is typically estimated with sample average approximation.
2. **Estimation of PC term:** $\nabla_{\theta\theta}\mathcal{R}(\theta, t)$, $\nabla_{\theta t}\mathcal{R}(\theta, t)$ are quantities that arise due to the presence of the PC term in the stochastic PC update rule. Although it is possible to construct unbiased sample average approximations of them individually, obtaining an unbiased estimate of the overall PC term is generally not possible due to the presence of non-linearity: the PC term is a product of the inverse of the hessian matrix and the cross derivative with respect to the parameter and time. Fortunately, the stochastic PC algorithm is robust against biases in the estimate of the PC term. That said, naively estimating the PC term with sample average approximation can lead to non-convergence of the stochastic PC algorithm (see Remark 3.5 for details). In section 4, we present two ways of estimating/evaluating $\nabla_{\theta\theta}\mathcal{R}(\theta, t)$, $\nabla_{\theta t}\mathcal{R}(\theta, t)$ that ensure the stochastic PC algorithm converges.

In the next section, we elucidate how errors in the estimates of $\nabla_\theta\mathcal{R}(\theta, t)$, $\nabla_{\theta\theta}\mathcal{R}(\theta, t)$, $\nabla_{\theta t}\mathcal{R}(\theta, t)$ affect the asymptotic tracking error of the stochastic PC algorithm.

3 Theoretical properties of the stochastic PC algorithm

We begin by stating our assumptions on the problem. We assume that the risk function is strongly convex with respect to θ. This assumption implies that $\theta^\star(t)$ is uniquely identified at time t and is crucial in studying the convergence of ATE.

Assumption 3.1. *$\mathcal{R}(\theta, t)$ is μ-strongly convex with respect to θ, i.e. for any θ_1, θ_2 and $t \geq 0$ it holds:*

$$\mathcal{R}(\theta_2, t) \geq \mathcal{R}(\theta_1, t) + (\theta_2 - \theta_1)^\top \nabla_\theta\mathcal{R}(\theta_1, t) + \tfrac{\mu}{2}\|\theta_2 - \theta_1\|_2^2 \,.$$

We now assume that as a function of t, $\theta^\star(t)$ is smooth, which is naturally satisfied in numerous examples including dynamic least squares recovery, object tracking, *etc.*.

Assumption 3.2. *The function $\theta^\star \in \mathcal{C}^2_{\mathbb{R}^d}([0, \infty))$, i.e. $\theta^\star$ is twice continuously differentiable with respect to time and its double derivative is uniformly bounded over time.*

As will be seen in the subsequent theorems, the bounds on the ATE of these stochastic algorithms depend on the error in the estimation of the pertinent gradients, which are defined as follows:

$$\xi_t = \hat{\nabla}_\theta R(\theta, t) - \nabla_\theta R(\theta, t)$$

denotes the estimation error of the gradient and

$$\zeta_t = \hat{\nabla}_{\theta\theta} R(\hat{\theta}_{t_k}, t_k)^{-1}\hat{\nabla}_{\theta t} R(\hat{\theta}_{t_k}, t_k) - \nabla_{\theta\theta} R(\theta^*_{t_k}, t_k)^{-1}\nabla_{\theta t} R(\theta^*_{t_k}, t_k)$$

represents the error in the adjustment term for the temporal drift. We use σ_ξ (resp. σ_ζ) to denote an upper bound on $\sup_t \mathbb{E}[\|\xi_t\|]$ (resp. $\sup_t \mathbb{E}[\|\zeta_t\|]$). These bounds may or may not be a function of h, depending on the application. Note that we do not assume the estimates of the gradients are unbiased. Below we present our main theorems regarding the bounds on the ATE of stochastic gradient descent and 1 in terms of σ_ξ, σ_ζ, learning rate η and stepsize h:

Theorem 3.3 (Stochastic predictor-corrector method)**.** *The update sequence of stochastic predictor-corrector method presented in Algorithm 1 yields the follows bound on ATE:*

$$\limsup_{k\to\infty} \mathbb{E}[\|\hat{\theta}_{kh} - \theta^\star_{kh}\|] \leq \frac{L'' h^2}{\eta M + hM'} + \eta\sigma_\xi + h\sigma_\zeta \,.$$

for any small $\eta > 0$. When σ_ξ and σ_ζ are independent of h then choice of $\eta = h$ implies that ATE of stochastic predictor-corrector method is at-most of the order of h.

Note that our analysis does not require the noise to have mean zero; we merely require them to have finite second moment. This is slightly more general that the usual stochastic optimization setting in which the noise is assumed to have mean zero. That said, in most of the applications that we have in mind, the noise comes from approximation of expectations by sample means, so the noise will be mean zero.

To see the benefits of the PC algorithm in the stochastic setting, we compare the ATE of the PC algorithm with that of stochastic approximation. Recall the stochastic gradient descent update rule is:

$$\hat{\theta}_{(k+1)h} = \hat{\theta}_{kh} - \eta \hat{\nabla}_\theta \mathcal{R}(\hat{\theta}_{kh}, kh). \tag{3.1}$$

for some estimate of gradient. Its ATE is known (Cutler et al., 2021), but we restate it here to facilitate comparison:

Theorem 3.4 (Stochastic gradient descent, Cutler et al. (2021))**.** *The update sequence of stochastic gradient method satisfies:*

$$\limsup_{k\to\infty} \mathbb{E}\left[\|\hat{\theta}_{t_k} - \theta^*_{t_k}\|\right] \leq \frac{Lh}{\mu\eta} + \eta\sigma_\xi$$

for any small $\eta > 0$. Minimizing the right hand side with respect to η yields:

$$\limsup_{k\to\infty} \mathbb{E}\left[\|\hat{\theta}_{t_k} - \theta^*_{t_k}\|\right] \leq \sqrt{\frac{Lh\sigma_\xi}{\mu}}\,.$$

Therefore, when σ_ξ does not depend on h, the rate is $O(\sqrt{h})$.

If we assume the error variances are independent of h, then simple stochastic gradient descent yields a bound of the order $\sqrt{h}$ on the ATE, whereas, the time-adjusted predictor-corrector based method yields a bound of the order of h, implying the superiority of the later for small stepsize. The superiority continues to hold even when the variances depend on h, as will be evident in the applications in the subsequent section.

Remark 3.5 (Naive estimation of PC term fails)**.** *In practice, it is straightforward to obtain estimates of $\nabla_\theta \mathcal{R}(\theta, t)$ (e.g. sample average approximation), but it is less straightforward to estimate the cross-derivative term $\nabla_{\theta t}\mathcal{R}(\theta, t)$. A naive application of first-order finite differences to estimating the cross-derivative term leads to poor tracking performance because this estimate of the cross-derivative term leads to a σ_ζ term that is $O(h^{-1})$. Indeed, we have*

$$\begin{aligned}\hat{\nabla}_{\theta t}\mathcal{R}(\theta, kh) &= \frac{\hat{\nabla}_\theta \mathcal{R}(\theta, kh) - \hat{\nabla}_\theta \mathcal{R}(\theta, (k-1)h)}{h} \\ &= \frac{\nabla_\theta \mathcal{R}(\theta, kh) - \nabla_\theta \mathcal{R}(\theta, (k-1)h)}{h} + \frac{\xi_{kh} - \xi_{(k-1)h}}{h}.\end{aligned}$$

As long as $\mathcal{R}(\theta, t)$ is smooth with respect to t, the the first term is $\nabla_{\theta t}\mathcal{R}(\theta, t) + O(h)$, but the second term is generally $O(h^{-1})$, e.g., if we assume that the errors over time are independent. Plugging this into the bound on the tracking error from Theorem 3.3, we see that the term that includes σ_ζ no longer depends on h, leading to a vacuous $O(1)$ tracking error bound. In Section 4, we use moving average schemes to obtain more accurate estimates of $\nabla_{\theta t}\mathcal{R}(\theta, t)$ to avoid this pitfall (e.g., for least square recovery problem in §4.1 see Equation (4.4) *and Lemma 4.3 for it's estimation and error analysis).*

4 Applications

In this section, we present three concrete time-varying optimization problem; all are characterized a gradual underlying distribution shift. This allows us to leverage the predictor-corrector (PC) method to improve tracking of the optimal trajectory. In some applications (e.g. the strategic classification example), there is a model for the distribution shift, so it is possible to evaluate the PC term exactly. In other applications, there is no such model, so it is necessary to approximate the PC term. We use a generic finite-difference approach in §4.1 and §4.3 to approximate the PC term. This approach is generally applicable, but it may not be optimal in applications in which the underlying distribution shift exhibits higher orders of smoothness.

4.1 LEAST SQUARES RECOVERY WITH FIXED DESIGN MATRIX

We first demonstrate the performance of the predictor-corrector method and compare it with gradient descent method in a linear regression model. We observe $\mathbf{y}_k \triangleq \{Y_{k_j}\}_{j=1}^n$ at time kh for $k \in \mathbb{N}$ and some fixed stepsize h, where the observations are modeled as: $\mathbf{y}_t = \mathbf{X}\theta_t^\star + \boldsymbol{\epsilon}_t$. Here $\mathbf{X} \in \mathbb{R}^{n\times d}$ is a fixed time-invariant design matrix and the co-ordinates of $\boldsymbol{\epsilon}_t$ are i.i.d with mean 0 and variance τ^2. The parameter of interest here is the function $\theta^\star : [0,\infty) \to \mathbb{R}^d$. We consider here the low dimensional scenario (i.e. $d < n$) and assume that the columns of $\mathbf{X}$ are in general position, i.e. $\mathbf{X}^\top\mathbf{X}$ is invertible. This immediately implies the following:

Lemma 4.1. *For any $t \in [0,\infty)$, $\theta_t^\star$ is the unique solution of the least square problem:*

$$\theta_t^\star = \arg\min_{\theta\in\mathbb{R}^d} \mathcal{R}(\theta,t) = \arg\min_{\theta\in\mathbb{R}^d} \tfrac{1}{2n}\mathbb{E}\left[\|\mathbf{y}_t - \mathbf{X}\theta\|^2\right]$$

where $\mathcal{R}$ is the risk function and the expectation is taken with respect to the distribution of $\boldsymbol{\epsilon}$.

The proof of Lemma can be found in Appendix A. We now compare the performance of stochastic gradient descent method (3.1) and PC method (Algorithm 1). The gradient of the risk function (with respect to) θ is: $\nabla_\theta\mathcal{R}(\theta,t) = \frac{1}{n}\mathbf{X}^\top\left(\mathbf{X}\theta - \mathbf{X}\theta_t^\star\right)$. We propose a *moving average based technique* to estimate the gradient of the risk function, *i.e.* for any time t we define:

$$\hat{\nabla}_\theta\mathcal{R}(\theta,t) = \tfrac{1}{n}\mathbf{X}^\top\left(\mathbf{X}\theta - \textstyle\sum_{i=0}^{m-1}\alpha_i\mathbf{y}_{t-ih}\right), \;\; \xi_t = \hat{\nabla}_\theta\mathcal{R}(\theta,t) - \nabla_\theta\mathcal{R}(\theta,t)\,. \tag{4.1}$$

The optimal choice of the moving window length m and $\{\alpha_i\}_{i=0}^{m-1}$ depends on a careful analysis of bias-variance trade-off of the estimation error ξ_t. First, note that ξ_t can be decomposed into two terms as follows:

$$\xi_t = \tfrac{1}{n}\mathbf{X}^\top\mathbf{X}\left\{\theta_t^\star - \textstyle\sum_{i=0}^{m-1}\alpha_i\theta_{t-ih}^\star\right\} - \tfrac{1}{n}\textstyle\sum_{i=0}^{m-1}\alpha_i\mathbf{X}^\top\boldsymbol{\epsilon}_{t-ih} \triangleq A + B\,.$$

We can expand the bias term A via a two step Taylor expansion:

$$\theta_t^\star - \textstyle\sum_{i=0}^{m-1}\alpha_i\theta_{t-ih}^\star = \theta_t^\star\left\{1 - \textstyle\sum_{i=0}^{m-1}\alpha_i\right\} - h\dot{\theta}^\star(t)\textstyle\sum_{i=0}^{m-1} i\alpha_i + \frac{h^2}{2}\textstyle\sum_{i=0}^{m-1} i^2\alpha_i\ddot{\theta}^\star(\tilde{t}_i)$$

for some $\tilde{t}_i \in [t-ih, t]$. The following lemma presents an optimal scheme for choosing $\{m;\alpha_i, i = 0,\ldots,m-1\}$:

Lemma 4.2 (Gradient estimate)**.** *For any fixed m, choosing the weights $\{\alpha_i\}$ as:*

$$(\alpha_0,\ldots,\alpha_{m-1}) \in \arg\min\left\{\textstyle\sum_{i=0}^{m-1} a_i^2 : \textstyle\sum_{i=0}^{m-1} a_i = 1, \text{ and } \textstyle\sum_{i=0}^{m-1} ia_i = 0, a_i \in \mathbb{R}\right\} \tag{4.2}$$

we obtain the following:

1. $\|A\|_2^2 = a_0^2 \times \mathcal{O}(m^4h^4)$, *where* $a_0^2 = \max_{t\geq 0}\left\|\left(\frac{\mathbf{X}^\top\mathbf{X}}{n}\right)\ddot{\theta}^\star(t)\right\|_2^2$,
2. $\mathbb{E}[\|B\|_2^2] = b_0^2 \times \mathcal{O}(m^{-1})$, *where* $b_0^2 = \tau^2\left\{\mathrm{TR}\left(\frac{\mathbf{X}^\top\mathbf{X}}{n^2}\right)\right\}$.

Therefore, taking $m = \mathcal{O}(h^{-4/5}\{a_0/b_0\}^{2/5})$ we have:

$$\mathbb{E}[\|\xi_t\|_2] \leq \|A\|_2 + \{\mathbb{E}[\|B\|_2^2]\}^{1/2} = \mathcal{O}(a_0m^2h^2) + \mathcal{O}(b_0m^{-1/2}) = \mathcal{O}(a_0^{1/5}b_0^{4/5}h^{2/5})\,. \tag{4.3}$$

The proof of this Lemma is deferred to the Appendix A. It is established in the proof that the optimal weights are: $\alpha_i = 2(2m-1-3i)/\{m(m+1)\}$. Using the error bound of Lemma 4.2 in Theorem 3.4 yields for small η, ATE is bounded by $\mathcal{O}(h/\eta) + \mathcal{O}(\eta h^{2/5})$. The right hand is minimized by taking $\eta = h^{3/10}$, which yields the order for asymptotic tracking error $\mathcal{O}(h^{7/10})$.

For **prediction-corrector based method** (Algorithm 1) we additionally need to estimate the Hessian and the time derivative of the gradient. The Hessian is constant $(\mathbf{X}^\top\mathbf{X}/n)$ and known (as we assume to know $\mathbf{X}$). To estimate $\nabla_{\theta t}\mathcal{R}(\theta,t) = -\{\mathbf{X}^\top\mathbf{X}/n\}\dot{\theta}_t^\star$, we again resort to a moving average based method, *i.e.* we set:

$$\widehat{\nabla}_{\theta t}\mathcal{R}(\theta,t) = -\tfrac{1}{n}\mathbf{X}^\top\left\{\textstyle\sum_{i=0}^{p-1}\beta_i\mathbf{y}_{t-ih}\right\} \tag{4.4}$$

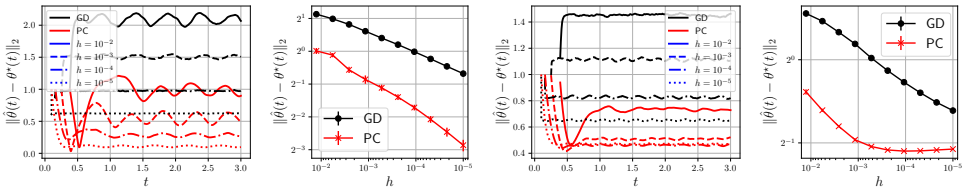

Figure 1: The two left-most figures show the performance of gradient descent and the PC method for time-varying linear regression. The left-most figure presents tracking error of GD and the PC method for various choices of h on the time interval $t \in [0, 3]$. The second figure from the left one compares the performance for a fixed $t = 3$ and different h to illustrate how the performance of the methods vary with h. The two right-most figures show the performance of gradient descent and the PC method for object tracking. The second figure from the right shows the tracking error of the object tracking model for GD and the PC method for different choices of h. The Y-axis represents the tracking error and X-axis represents the time interval $t \in (0, 3)$.

for some choice of p and the weights $\{\beta_j\}_{j=0}^{p-1}$, where the weights and the size of the window is obtained from a bias-variance trade-off:

$$\zeta_t = \left(\frac{\mathbf{X}^\top \mathbf{X}}{n}\right)^{-1} \left[\widehat{\nabla}_{\theta t}\mathcal{R}(\theta, t) - \nabla_{\theta t}\mathcal{R}(\theta, t)\right]$$
$$= -\sum_{i=0}^{p-1} \beta_i \theta^\star_{t-ih} + \dot{\theta}^\star_t - \left(\frac{\mathbf{X}^\top \mathbf{X}}{n}\right)^{-1} \frac{\mathbf{X}^\top}{n} \left(\sum_{i=0}^{p-1} \beta_i \boldsymbol{\epsilon}_{t-ih}\right) \triangleq C + D\,.$$

The following lemma presents the optimal choice of the weights, window length and consequently an error bound:

Lemma 4.3. *(Estimate of time derivative) For any p, if we choose the weights β_i's as:*

$$(\beta_0, \ldots, \beta_{p-1}) \in \arg\min \left\{\sum_{i=0}^{p-1} b_i^2 : \sum_{i=0}^{p-1} b_i = 0, \text{ and } \sum_{i=0}^{p-1} i b_i = -\tfrac{1}{h}, b_i \in \mathbb{R}\right\} \tag{4.5}$$

we have the following bounds on the error:

1. $\|C\|_2^2 = \max_{t \geq 0} \|\ddot{\theta}^\star(t)\|_2^2 \times \mathcal{O}(h^2 p^2) \triangleq c_0^2 \times \mathcal{O}(h^2 p^2)$,
2. $\mathbb{E}[\|D\|_2^2] = \tau^2 \{\text{TR}[(X^\top X/n)^{-1}]/n\} \times \mathcal{O}(h^{-2} p^{-3}) \triangleq d_0^2 \times \mathcal{O}(h^{-2} p^{-3})$.

Setting $p = (d_0/c_0)^{1/4} h^{-3/4}$, we have:

$$\mathbb{E}[\|\zeta_t\|_2] \leq \|C\|_2 + \{\mathbb{E}[\|D\|_2^2]\}^{1/2} = \mathcal{O}(c_0 h p) + \mathcal{O}(d_0 h^{-2} p^{-3}) = \mathcal{O}(c_0^{3/4} d_0^{1/4} h^{1/4})\,. \tag{4.6}$$

The β_j in the above lemma takes the value $\frac{6(p-1-2j)}{p(p^2-1)h}$ as elaborated in the proof of the Lemma (see Appendix A). From the above lemma we have $\sigma_\zeta = \max_k \mathbb{E}[\|\zeta_{kh}\|_2] = \mathcal{O}(h^{1/4})$. Furthermore, we have established in the analysis of gradient descent method that the order of the error of the gradient estimation is $\sigma_\xi = \mathcal{O}(h^{2/5})$ (Lemma 4.2). Using these rates in Theorem 3.3 we conclude that predictor-corrector method with step size η ATE is of the order $\mathcal{O}(h^2/\{\eta + h\}) + \mathcal{O}(\eta h^{2/5}) + \mathcal{O}(h^{5/4})$. The bound is minimized at $\eta = h^{4/5}$ and consequently, the order of the tracking error is $\mathcal{O}(h^{6/5})$. Therefore, predictor-corrector method yields faster rate (in terms of the step-size h) in comparison to the gradient based method.

Simulation studies: We compare the tracking performance of the gradient descent based and the predictor-corrector based method for the regression model via simulation. For simulation purpose, we set $d = 2, n = 40$. The values of $\{X_i\}_{i=1}^n$ are generated independently from $\mathcal{N}(0, I_2)$ and remain fixed over time. The true parameter is taken to be $\theta_t^* = (\sin(2\pi t), \cos(2\pi t))$ and at time t, the observation $\mathbf{y}_t \in \mathbb{R}^n$ is generated as $\mathbf{y}_t = \mathbf{X}\theta_t^* + \boldsymbol{\epsilon}_t$ where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, 0.5 I_n)$. Runtime analysis for the gradient descent based and the predictor-corrector based methods can be found in Figure 2 in Appendix C.

The tracking performance of the methods over a finite time interval ($t \in (0, 3)$) and the effect of h on the limiting error (i.e. tracking error for some large t) are presented in Figure 1. In the left plot

of Figure 1, we observe that the predictor-corrector based method (denoted by PC) outperforms the gradient-descent based method (denoted by GD) in terms of the tracking error for all moderately large t (here, $t \geq 1$) for all choices of $h \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. As illustrated in our theory, the rate of convergence of limiting error of predictor-corrector based method decreases faster in h compared to the gradient descent based method. The right side of Figure 1 establishes this phenomena through comparing the performances at $t = 3$ for several choices of h. As $\|\hat{\theta}_t - \theta_t^\star\|$ is a random quantity, error bars (over 10 Monte-Carlo iterations) are provided to quantify the variability of the tracking error, which turns out to be relatively small compared to the mean difference of the tracking error of the methods.

4.2 Strategic classification

In strategic classification, samples correspond to agents who change their features strategically to affect the output of the ML model (e.g. scammers modifying their scam to skirt a scam detector). To keep things simple, we assume the ML model is a binary classifier, and the positive output is advantageous for the agents. Following Hardt et al. (2016), we assume the agents are maximizing utility, and they have full information regarding the classifier. Thus an agent with features x changes their features by solving the (expected) utility maximization problem

$$x_+(x) \triangleq \arg\max_{x' \in \mathcal{X}} u_+ f(x') - c(x, x'), \tag{4.7}$$

where $u_+ > 0$ is the utility from a positive output, $f(x)$ is the predicted probability of having positive output for x from an ML model, $u_+ f(x)$ is the expected utility at x, and $c(x, x') > 0$ encodes the cost of changing features from x to x'.

Let p_1 be the probability density function (pdf) of the positive class conditional at time t and f_t be the ML model deployed at time t. The agents respond strategically to f_t; *i.e.* an agent with feature x changes their features to $x_+(x)$ (their label remains unchanged). The resulting change in the class conditional satisfies the continuity equation:

$$\partial_t p_1 + \nabla \cdot (v p_1) = 0,$$

where $\nabla \triangleq [\partial_{x_1} \quad \dots \quad \partial_{x_d}]$ is the spatial gradient operator and v is the vector field $v(x) \triangleq x_+(x) - x$. Similarly, the pdf of the negative class conditional also satisfies the continuity equation.

This change in the distribution of agent features leads to a time-varying optimization problem:

$$\theta(t) \in \arg\min_\theta f(\theta, t) \triangleq \pi \int_{\mathcal{X}} \ell(f_\theta(x), 1) dP_1(x) + (1 - \pi) \int_{\mathcal{X}} \ell(f_\theta(x), 0) dP_0(x),$$

where $\pi \triangleq \mathbb{P}\{Y = 1\}$ is the faction of the positive class and ℓ is a loss function for the classification task. Interchanging limits freely, we see that it is possible to estimate $\nabla_\theta f(\theta, t)$ and $\nabla_{\theta\theta} f(\theta, t)$ empirically:

$$\begin{aligned}
\nabla_\theta f(\theta, t) &= \pi \int_{\mathcal{X}} \nabla_\theta \ell(f_\theta(x), 1) dP_1(x) + (1 - \pi) \int_{\mathcal{X}} \nabla_\theta \ell(f_\theta(x), 0) dP_0(x), \\
\nabla_{\theta\theta} f(\theta, t) &= \pi \int_{\mathcal{X}} \nabla_\theta^2 \ell(f_\theta(x), 1) dP_1(x) + (1 - \pi) \int_{\mathcal{X}} \nabla_\theta^2 \ell(f_\theta(x), 0) dP_0(x)
\end{aligned}$$

Similarly, it is possible to estimate $\nabla_{\theta t} f(\theta, t)$ empirically:

$$\begin{aligned}
\nabla_{\theta t} f(\theta, t) &= \pi \int_{\mathcal{X}} \ell(f_\theta(x), 1) \partial_t p_1(x, t) dx + (1 - \pi) \int_{\mathcal{X}} \ell(f_\theta(x), 0) \partial_t p_0(x, t) dx \\
&= -\pi \int_{\mathcal{X}} \ell(f_\theta(x), 1) \nabla \cdot (v(x) p_1(x, t)) dx - (1 - \pi) \int_{\mathcal{X}} \ell(f_\theta(x), 0) \nabla \cdot (v(x) p_0(x, t)) dx \\
&= \pi \int_{\mathcal{X}} \nabla_x \ell(f_\theta(x), 1)^\top v(x) p_1(x, t) dx + (1 - \pi) \int_{\mathcal{X}} \nabla_x \ell(f_\theta(x), 0)^\top v(x) p_0(x, t) dx.
\end{aligned}$$

where we appealed to the continuity equation in the second step and Green's identities in the third step. Unlike the other two applications in this section, it is possible to compute the PC term exactly (without resorting to finite-difference approximation) here.

4.3 Object tracking

Our third application is object tracking problem proposed and analyzed in Patra et al. (2020). Assume we have n sensors placed the position $\{X_i\}_{i=1}^n$ in $\mathbb{R}^3$ and $\theta^\star(t)$ denotes position of the object (that we aim to track) at t. At any given point, we observe a noisy version of some monotone function of distance of the object from the sensors, i.e. we observe:

$$Y_{i,t} = f(\|X_i - \theta_t^\star\|^2) + \epsilon_{i,t} \;\; \forall\, 1 \leq i \leq n\,.,$$

In this example, we assume we know f. The case of unknown f is more complicated and beyond the scope of the paper. The risk function for estimating $\theta_t^\star$ under quadratic loss function is:

$$\mathcal{R}(\theta, t) = \mathbb{E}\left[\frac{1}{2n}\sum_{i=1}^n \left(Y_{i,t} - f(\|X_i - \theta\|^2)\right)^2\right] = \frac{\sigma^2}{2} + \frac{1}{2n}\sum_{i=1}^n \left(f(\|X_i - \theta\|^2) - f(\|X_i - \theta_t^\star\|^2)\right)^2$$

Note that the risk function here is not strongly convex (so it does not satisfy the assumptions of Theorem 3.3), but as we shall see, the PC algorithm nevertheless outperforms standard first-order methods. To apply the gradient descent method and predictor-corrector method we need to estimate the gradient, Hessian and the time derivative of the gradient of the risk function. The gradient and the Hessian can be easily estimated as using sample averages (exact expressions are presented in the Appendix B). The time derivative of the gradient is:

$$\nabla_{\theta,t}\mathcal{R}(\theta, t) = -\frac{2}{n}\sum_{i=1}^n f'(\|X_i - \theta\|^2)(\theta - X_i)\frac{d}{dt}f(\|X_i - \theta_t^\star\|^2)$$

To estimate $\frac{d}{dt}f(\|X_i - \theta_t^\star\|^2)$ we again resort to the moving average procedure:

$$\widehat{\partial}_t f(\|X_i - \theta_t^\star\|^2) = \sum_{j=0}^{p-1}\beta_j Y_{i,t-jh}\,,$$

where $p, \{\beta_j\}_{j=1}^p$ are chosen carefully to balance the bias variance trade-off. For notational simplicity we drop the index i and define $g(t) = f(\|X_i - \theta_t^\star\|^2)$. From the relation $Y_{i,t} = g(t) + \epsilon_{i,t}$ and Assumption 3.2, we have:

$$\sum_{j=0}^{p-1}\beta_j Y_{i,t-jh} = \sum_{j=0}^{p-1}\beta_j\{g(t) - jhg'(t) + \frac{j^2h^2}{2}g''(\tilde{t}_j)\} + \sum_{j=0}^{p-1}\beta_j\epsilon_{i,t-jh}$$

Now if the sequence $\{\beta_j\}_0^{p-1}$ satisfies $\sum_j \beta_j = 0$ and $\sum_j j\beta_j = -(1/h)$ then we have:

$$\sum_{j=0}^{p-1}\beta_j Y_{i,t-jh} - g'(t) = (\sum_j j^2\beta_j)O(h^2) + \sum_{j=0}^{p-1}\beta_j\epsilon_{i,t-jh}\,.$$

The variance of the error term is $\sigma_\epsilon^2\sum_j\beta_j^2$. Therefore we will choose $\{\beta_j\}$ and p by minimizing $\sum_j\beta_j^2$ subject to the above constraints. Similar calculation as of Example 1 (Lemma 4.3) yields $p = \mathcal{O}(h^{-3/4})$ and $\{\beta_j\}_{j=0}^{p-1}$ is $\beta_j = \frac{6(p-1-2j)}{p(p^2-1)h}$.

Simulation study We now compare the performance of the methods on some simulation for this object tracking problem. We take $n = 11^2$ many sensors for simulation purpose, which are generated as follows: we first take 11 equi-spaced points between $[-1, 1]$ both on X-axis and Y-axis. The sensors are then positioned at the Cartesian product of these generated points. The path of the moving object is taken to be $\theta_t^\star = (\sin(2\pi t), \cos(2\pi t))$ and the non-parametric monotone function is set as $f(x) = x$. We generate noisy observations of the object as $y_{it} = f(\|X_i - \theta_t^\star\|^2) + \epsilon_{it}$ where $\epsilon_{it} \sim \mathcal{N}(0, 0.25)$. Run-time analysis for the two methods can be found in Figure 2 in Appendix C. Figure 1 (third and forth plots from left) presents the tracking error of both the methods (gradient descent based method (resp. predictor-corrector based method) is represented as GD (resp. PC)) for three choices of $h \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. The superiority of the performance of the predictor-corrector based method for all $t \geq 1$ and for various choices of h is evident from the third picture from the left-side of Figure 1. In the last picture, we compare the limiting performance of both the methods (here we take the error at $t = 3$ to be the limiting error) for several choices of h. The superiority of the performance of PC corroborates our theoretical finding that the tracking error for PC converges faster in terms of h compared to the GD. We also show error bars (over 10 Monte-Carlo iterations) of the random tracking error $\|\hat{\theta}_t - \theta_t^\star\|$ at $t = 3$ on the right picture, which indicates that the variability is relatively small compared to the difference of the mean tracking error of the methods.

5 Conclusion

We developed predictor corrector algorithms for stochastic time-varying optimization. These algorithms leverage smoothness in the temporal drift to anticipate changes to the optimal solution. We showed that these algorithms have smaller asymptotic tracking errors than their non-predictor corrector counterparts and demonstrated their efficacy in three applications. Although we focused on first-order algorithms in this paper, the predictor corrector term in the update rules of our PC algorithms can be incorporated into the update rules of other algorithms (*e.g.* Newton-type methods). We hope that the benefits of first-order PC algorithms motivates others to study PC versions of other algorithms.

REFERENCES

Alekh Agarwal, Olivier Chapelle, Miroslav Dudík, and John Langford. A reliable effective terascale linear learning system. *The Journal of Machine Learning Research*, 15(1):1111–1133, 2014.

Léon Bottou. Stochastic learning. In *Summer School on Machine Learning*, pp. 146–168. Springer, 2003.

Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pp. 421–436. Springer, 2012.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. *Advances in neural information processing systems*, 20, 2007.

Gavin Brown, Shlomi Hod, and Iden Kalemaj. Performative prediction in a stateful world. In *International Conference on Artificial Intelligence and Statistics*, pp. 6045–6061. PMLR, 2022.

Joshua Cutler, Dmitriy Drusvyatskiy, and Zaid Harchaoui. Stochastic optimization under time drift: Iterate averaging, step-decay schedules, and high probability guarantees. In *Thirty-Fifth Conference on Neural Information Processing Systems*, May 2021.

Rishabh Dixit, Amrit Singh Bedi, Ruchi Tripathi, and Ketan Rajawat. Time varying optimization via inexact proximal online gradient descent. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 759–763. IEEE, 2018.

Rishabh Dixit, Amrit Singh Bedi, Ruchi Tripathi, and Ketan Rajawat. Online learning with inexact proximal online gradient descent algorithms. *IEEE Transactions on Signal Processing*, 67(5): 1338–1352, 2019.

Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 55–70, 2018.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pp. 111–122, 2016.

Qing Ling and Alejandro Ribeiro. Decentralized dynamic optimization through the alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 62(5):1185–1197, 2013.

Celestine Mendler-Dünner, Juan Perdomo, Tijana Zrnic, and Moritz Hardt. Stochastic optimization for performative prediction. *Advances in Neural Information Processing Systems*, 33:4929–4939, 2020.

Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in neural information processing systems*, 24, 2011.

Marko Nonhoff and Matthias A Müller. Online gradient descent for linear dynamical systems. *IFAC-PapersOnLine*, 53(2):945–952, 2020.

Rohit K Patra, Moulinath Banerjee, and George Michailidis. A semi-parametric model for target localization in distributed systems. *arXiv preprint arXiv:2012.02025*, 2020.

Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative Prediction. *arXiv:2002.06673 [cs, stat]*, June 2020.

A Yu Popkov. Gradient methods for nonstationary unconstrained optimization problems. *Automation and Remote Control*, 66(6):883–891, 2005.

Michael M Zavlanos, Alejandro Ribeiro, and George J Pappas. Network integrity in mobile robotic networks. *IEEE Transactions on Automatic Control*, 58(1):3–18, 2012.

Yunong Zhang, Ke Chen, and Hong-Zhou Tan. Performance analysis of gradient neural network exploited for online time-varying matrix inversion. *IEEE Transactions on Automatic Control*, 54 (8):1940–1945, 2009.

A SUPPLEMENTARY PROOFS

Proof of Corollary 2.3. This is a special case of Theorem 3.3 with $\sigma_\xi = 0$ and $\sigma_\zeta = 0$. □

Proof of Theorem 2.2. We prove Theorem 2.2 by a simple counterexample, where we show that the error is lower bounded by h. Pick some $0 < \eta < 1$. Consider the risk function $R(\theta, t) = \frac{\mu}{2}(\theta - t\theta^*)^2$ for some $\theta^* \in \mathbb{R}$. Therefore the gradient is:

$$\nabla_\theta f(\theta, t) = \mu(\theta - t\theta^*)$$

and the strong convexity assumption 3.1 is satisfied since it holds:

$$\begin{aligned}
&R(\theta_2, t) - R(\theta_1, t) - (\theta_2 - \theta_1)\nabla_\theta R(\theta_1, t) \\
&= \frac{\mu}{2}(\theta_2 - t\theta^*)^2 - \frac{\mu}{2}(\theta_1 - t\theta^*)^2 - (\theta_2 - \theta_1)\mu(\theta_1 - \theta^\star t) \\
&= \frac{\mu}{2}\{\theta_2^2 - 2\theta_2\theta^\star t + t^2\theta^{\star 2} - \theta_1^2 + 2\theta_1\theta^\star t - t^2\theta^{\star 2} - 2\theta_1\theta_2 + 2\theta_1^2 + 2\theta_2\theta^\star t - 2\theta_1\theta^\star t\} \\
&= \frac{\mu}{2}(\theta_2 - \theta_1)^2
\end{aligned}$$

which implies the gradient update is:

$$\begin{aligned}
\hat{\theta}_{t_{k+1}} &= \hat{\theta}_{t_k} - \eta\nabla_\theta f(\hat{\theta}_{t_k}, t_k) \\
&= \hat{\theta}_{t_k} - \eta\mu(\hat{\theta}_{t_k} - t_k\theta^*) \\
&= (1 - \eta\mu)\hat{\theta}_{t_k} + \eta\mu t_k\theta^*
\end{aligned}$$

This implies that the error:

$$\begin{aligned}
e_{k+1} &= \hat{\theta}_{t_{k+1}} - t_{k+1}\theta^* \\
&= (1 - \eta)\mu(\hat{\theta}_{t_k} - t_k\theta^*) + t_k\theta^* - t_{k+1}\theta^* \\
&= (1 - \eta\mu)(\hat{\theta}_{t_k} - t_k\theta^*) - h\theta^* \\
&= (1 - \eta\mu)e_{t_k} - h\theta^*
\end{aligned}$$

Unravelling the recursion we have:

$$e_{k+1} = (1 - \eta\mu)^{k+1}e_0 - h\sum_{j=0}^{k}(1 - \eta\mu)^j$$

By triangle inequality:

$$|e_{k+1}| \geq h\sum_{j=0}^{k}(1 - \eta\mu)^j - (1 - \eta\mu)^{k+1}|e_0|$$

which further implies:

$$\liminf_{k\to\infty} |e_{k+1}| \geq \frac{h}{\eta\mu}\,.$$

This completes the proof. □

Proof of Theorem 3.3. The proof of Theorem 3.3 is similar to that of Theorem 2.3. The true function $\theta^*(t)$ satisfies the following differential equation:

$$\dot{\theta}^*(t) = -\nabla_{\theta\theta}R(\theta^*(t), t)^{-1}\nabla_{\theta t}R(\theta^*(t), t)$$

This implies (via a two-step Taylor expansion):

$$\theta^*_{t_{k+1}} = \theta^*_{t_k} - h\nabla_{\theta\theta}R(\theta^*_{t_k}, t_k)^{-1}\nabla_{\theta t}R(\theta^*_{t_k}, t_k) + \underbrace{\frac{h^2}{2}\ddot{\theta}^*(\tilde{t}_k)}_{\leq L'' h^2}\,. \tag{A.1}$$

Subtracting (A.1) from the update rule in Algorithm 1 and using the first order condition $\nabla_\theta R(\theta^*_{t_k}, t_k) = 0$ we have:

$$\begin{aligned}
e_{t_{k+1}} &= e_{t_k} - \eta\left(\hat{\nabla}_\theta R(\hat{\theta}_{t_k}, t_k) - \nabla_\theta R(\theta^*_{t_k}, t_k)\right) \\
&\quad - h\left(\hat{\nabla}_{\theta\theta} R(\hat{\theta}_{t_k}, t_k)^{-1}\hat{\nabla}_{\theta t} R(\hat{\theta}_{t_k}, t_k) - \nabla_{\theta\theta} R(\theta^*_{t_k}, t_k)^{-1}\nabla_{\theta t} R(\theta^*_{t_k}, t_k)\right) - \frac{h^2}{2}\ddot{\theta}^*(\tilde{t}_k) \\
&= e_{t_k} - \eta\xi_{t_k} - \eta\left(\nabla_\theta R(\hat{\theta}_{t_k}, t_k) - \nabla_\theta R(\theta^*_{t_k}, t_k)\right) - h\zeta_{t_k} \\
&\quad - h\left(\nabla_{\theta\theta} R(\hat{\theta}_{t_k}, t_k)^{-1}\nabla_{\theta t} R(\hat{\theta}_{t_k}, t_k) - \nabla_{\theta\theta} R(\theta^*_{t_k}, t_k)^{-1}\nabla_{\theta t} R(\theta^*_{t_k}, t_k)\right) - \frac{h^2}{2}\ddot{\theta}^*(\tilde{t}_k)
\end{aligned}$$

Now define a function $g(x)$ as:

$$g(x) = x - \eta\nabla_\theta R(x, t) - h\nabla_{\theta\theta} R(x, t)^{-1}\nabla_{\theta t} R(x, t)$$

which implies:

$$\|\nabla_x g(x)\|_{op} \leq (1 - \eta M - hM')$$

The above bound yields:

$$\begin{aligned}
\|e_{t_{k+1}}\| &\leq \Big\| e_{t_k} - \eta\left(\nabla_\theta R(\hat{\theta}_{t_k}, t_k) - \nabla_\theta R(\theta^*_{t_k}, t_k)\right) - \eta\xi_{t_k} - h\zeta_{t_k} \\
&\qquad - h\left(\nabla_{\theta\theta} R(\hat{\theta}_{t_k}, t_k)^{-1}\nabla_{\theta t} R(\hat{\theta}_{t_k}, t_k) - \nabla_{\theta\theta} R(\theta^*_{t_k}, t_k)^{-1}\nabla_{\theta t} R(\theta^*_{t_k}, t_k)\right)\Big\| + L''h^2 \\
&= \left\| g(\hat{\theta}_{t_k}) - g(\theta^*_{t_k})\right\| + \eta\|\xi_{t_k}\| + h\|\zeta_{t_k}\| + L''h^2 \\
&\leq (1 - \eta M - hM'')\|e_{t_k}\| + \eta\|\xi_{t_k}\| + h\|\zeta_{t_k}\| + L''h^2\,.
\end{aligned}$$

Unravelling the recursion yields:

$$\begin{aligned}
\|e_{t_{k+1}}\| &\leq (1 - \eta M - hM'')^{t+1}\|e_0\| + L''h^2\sum_{j=0}^{t}(1 - \eta M - hM'')^j \\
&\qquad + \eta\sum_{j=0}^{t}(1 - \eta M - hM'')^j\|\xi_{t_{k-j}}\| + h\sum_{j=0}^{t}(1 - \eta M - hM'')^j\|\zeta_{t_{k-j}}\| \\
&\leq (1 - \eta M - hM'')^{t+1}\|e_0\| + \frac{L''h^2}{\eta M + hM'} \\
&\qquad + \eta\sum_{j=0}^{t}(1 - \eta M - hM'')^j\|\xi_{t_{k-j}}\| + h\sum_{j=0}^{t}(1 - \eta M - hM'')^j\|\zeta_{t_{k-j}}\|
\end{aligned}$$

The statement of the theorem now follows by taking expectation on the both sides and letting $k \to \infty$ on the both sides of the above equation. □

Proof of Theorem 3.4. We note that

$$\begin{aligned}
e_{t_{k+1}} &= \hat{\theta}_{t_{k+1}} - \theta^*_{t_{k+1}} \\
&= \hat{\theta}_{t_k} - \theta^*_{t_k} - \eta\left(\nabla_\theta R(\hat{\theta}_{t_k}, t_k) - \nabla_\theta R(\theta^*_{t_k}, t_k)\right) + (\theta^*_{t_k} - \theta^*_{t_{k+1}}) - \eta\xi_{t_k} \qquad \text{(A.2)} \\
&\triangleq e_{t_k} - \eta\left(\nabla_\theta R(\hat{\theta}_{t_k}, t_k) - \nabla_\theta R(\theta^*_{t_k}, t_k)\right) + (\theta^*_{t_k} - \theta^*_{t_{k+1}}) - \eta\xi_k
\end{aligned}$$

For the rest of the analysis, define a function $g(x)$ as:

$$g(x) = x - \eta\nabla_\theta R(x, t)$$

Therefore, the gradient of g with respect to x is:

$$\nabla g(x) = I - \eta\nabla_{\theta\theta} R(x, t)$$

Therefore we have, $\|\nabla g(x)\|_{op} \leq 1 - M\eta < 1$ as soon as $\eta < 1/M$. Hence from (A.2) we have:

$$\begin{aligned}\|e_{t_{k+1}}\| &\leq \left\| e_{t_k} - \eta \left(\nabla_\theta R(\hat{\theta}_{t_k}, t_k) - \nabla_\theta R(\theta^*_{t_k}, t_k) \right) \right\| + Lh + \eta \|\xi_t\| \\ &= \|g(\hat{\theta}_{t_k}) - g(\theta^*_{t_k})\| + Lh + \eta\|\xi_t\| \\ &\leq (1 - \eta M)\|e_{t_k}\| + Lh + \eta\|\xi_t\| \,.\end{aligned}$$

Unravelling the recursion yields:

$$\begin{aligned}\mathbb{E}[\|e_{t_{k+1}}\|] &\leq (1-\eta M)^{t+1}\mathbb{E}[\|e_{t_k}\|] + Lh\sum_{j=0}^{t}(1-\eta M)^j + \eta\sum_{j=0}^{t}(1-\eta M)^j\mathbb{E}[\|\xi_{t-j}\|] \\ &\leq (1-\eta M)^{t+1}\mathbb{E}[\|e_{t_k}\|] + \frac{Lh}{\eta M} + \eta\sigma_\xi \,,\end{aligned}$$

from which the statement of the theorem follows immediately by taking $k \to \infty$. □

Proof of Lemma 4.1. Defining $L(\theta) = \frac{1}{2n}\mathbb{E}\left[\|\mathbf{y}_t - \mathbf{X}\theta\|^2\right]$ we note that the gradient

$$\begin{aligned}\nabla_\theta L(\theta) &= \frac{1}{n}\mathbb{E}\left[\mathbf{X}^\top(\mathbf{y}_t - \mathbf{X}\theta)\right] \\ &= \frac{\mathbf{X}^\top\mathbf{X}}{n}(\theta^\star_t - \theta)\end{aligned}$$

is zero only when $\theta = \theta^\star_t$. Thus, the risk is minimized at $\theta^\star_t$.

□

Proof of lemma 4.2. The gradient decomposition in (4.2) easily realized from the following:

$$\begin{aligned}\xi_t &= \hat{\nabla}_\theta \mathcal{R}^\star(\theta, t) - \nabla_\theta \mathcal{R}^\star(\theta, t) \\ &= X^\top(\cancel{X\theta} - \sum_{i=0}^{m-1}\alpha_i y_{t-ih})/n - X^\top(\cancel{X\theta} - X\theta^\star(t))/n \\ &= \{X^\top X/n\}\theta^\star(t) - \{X^\top X/n\}\sum_{i=0}^{m-1}\alpha_i\theta^\star(t-ih) - \sum_{i=0}^{m-1}\alpha_i X^\top\epsilon(t-ih)/n \,.\end{aligned}$$

Next we compute the weights α_i's in the scheme (4.2), which shall be needed to establish the bounds For A and B. For Lagrangian constants λ_1 and λ_2 we define the Lagrangian problem:

$$\mathbf{L} = \sum_{i=0}^{m-1}\left\{\frac{\alpha_i^2}{2} - \lambda_1\alpha_i - \lambda_2 i\alpha_i\right\} + \lambda_1 \tag{A.3}$$

and notice that at optimum it holds:

$$\partial_i \mathbf{L} = \alpha_i - \lambda_1 - \lambda_2 i = 0 \text{ or } \alpha_i = \lambda_1 + \lambda_2 i \,. \tag{A.4}$$

Defining $S_j = \sum_{i=0}^{m-1} i^j,\ j = 0, 1, 2$, we notice from (A.4) that the following hold:

$$1 = \sum_{i=0}^{m-1}\alpha_i = \lambda_1 S_0 + \lambda_2 S_1 \text{ and } 0 = \sum_{i=0}^{m-1} i\alpha_i = \lambda_1 S_1 + \lambda_2 S_2 \,. \tag{A.5}$$

The above system of equations has the solution $\lambda_1 = S_2/(S_0S_2 - S_1^2)$ and $\lambda_2 = -S_1/(S_0S_2 - S_1^2)$, where noticing that $S_0S_2 - S_1^2 = m^2(m^2-1)/12$, $S_1 = m(m-1)/2$ and $S_2 = m(m-1)(2m-1)/6$ we have $\lambda_1 = 2(2m-1)/\{m(m+1)\}$, $\lambda_2 = -6/\{m(m+1)\}$, and the optimal weights are

$$\alpha_i = \tfrac{2(2m-1-3i)}{m(m+1)} \,. \tag{A.6}$$

We now establish the bounds for A and B.

Bounds for A and B We start with the decomposition of $\|B\|_2^2$

$$\begin{aligned}\|B\|_2^2 &= \Big\| \sum_{i=0}^{m-1} \alpha_i X^\top \epsilon(t-ih)/n \Big\|_2^2 \\ &= \sum_{i=0}^{m-1} \alpha_i^2 \|X^\top \epsilon(t-ih)/n\|_2^2 + \sum_{i\neq j=0}^{m-1} \alpha_i \alpha_j \epsilon(t-ih)^\top \{X^\top X/n^2\} \epsilon(t-jh)\,,\end{aligned}$$

where we notice that $\epsilon(t-ih)^\top\{X^\top X/n^2\}\epsilon(t-jh),\ i\neq j$ has zero mean (concluded from: $\epsilon(t-ih)$ and $\epsilon(t-jh)$ for $i\neq j$ are independent random vectors has zero mean). This means we have

$$\begin{aligned}\mathbb{E}[\|B\|_2^2] &= \sum_{i=0}^{m-1} \alpha_i^2 \mathbb{E}[\|X^\top \epsilon(t-ih)/n\|_2^2] \\ &= \sum_{i=0}^{m-1} \alpha_i^2 \mathbb{E}[\epsilon(t-ih)^\top \{X^\top X/n^2\} \epsilon(t-ih)] \\ &= \Big\{ \sum_{i=0}^{m-1} \alpha_i^2 \Big\} \tau^2 \mathrm{TR}\{X^\top X/n^2\}\end{aligned}$$

where we notice that

$$\begin{aligned}\sum_{i=0}^{m-1} \alpha_i^2 &= \sum_{i=0}^{m-1} \frac{4(2m-1-3i)^2}{m^2(m+1)^2} \\ &\asymp \sum_{i=0}^{m-1} \frac{4(2m-1-3i)^2}{m^4} \\ &= \sum_{i=0}^{m-1} \frac{4(2-1/m-3i/m)^2}{m^2} \\ &\asymp \frac{1}{m^2} \sum_{i=0}^{m-1} 4\Big(2-3\frac{i}{m}\Big)^2 \\ &\asymp \frac{1}{m} \int_0^1 4(2-3x)^2 dx = \mathcal{O}(m^{-1})\,,\end{aligned}$$

where, we see that the last approximation holds from the limit of Riemannian sum, and the bound on $\mathbb{E}[\|B\|_2^2]$ holds. Now, to establish the bound for $\|A\|_2$ we notice that

$$\begin{aligned}\theta^\star(t) - \sum_{i=0}^{m-1} \alpha_i \theta^\star(t-ih) &= \theta^\star(t) - \sum_{i=0}^{m-1} \alpha_i \{\theta^\star(t) - ih\dot\theta^\star(t) + (ih)^2/2\ddot\theta^\star(\tilde t_i)\} \\ &= \theta^\star(t)\cancelto{0}{\Big\{1 - \sum_{i=0}^{m-1} \alpha_i\Big\}} - h\dot\theta^\star(t) \cancelto{0}{\sum_{i=0}^{m-1} i\alpha_i} + \frac{h^2}{2} \sum_{i=0}^{m-1} i^2 \alpha_i \ddot\theta^\star(\tilde t_i)\,,\end{aligned}$$

where $\tilde t_i$ is some number in the interval $[t-ih, t]$. Hence, we have

$$\begin{aligned}\|A\|_2 &= \Big\| \{X^\top X/n\} \Big\{ \theta^\star(t) - \sum_{i=0}^{m-1} \alpha_i \theta^\star(t-ih) \Big\} \Big\|_2 \\ &= \frac{h^2}{2} \Big\| \{X^\top X/n\} \sum_{i=0}^{m-1} i^2 \alpha_i \ddot\theta^\star(\tilde t_i) \Big\|_2 \\ &\leq \frac{h^2}{2} \sum_{i=0}^{m-1} i^2 |\alpha_i| \|\{X^\top X/n\} \ddot\theta^\star(\tilde t_i)\|_2 \\ &\leq \big\{ \max_{t\geq 0} \|\{X^\top X/n\} \ddot\theta^\star(t)\|_2 \big\} \frac{h^2}{2} \sum_{i=0}^{m-1} i^2 |\alpha_i|\,,\end{aligned}$$

where

$$\begin{aligned}\sum_{i=0}^{m-1} i^2|\alpha_i| &= \sum_{i=0}^{m-1} \frac{2i^2|2m-1-3i|}{m(m+1)} \\ &\asymp \sum_{i=0}^{m-1} \frac{2i^2|2m-3i|}{m^2} \\ &= m^2 \frac{1}{m} \sum_{i=0}^{m-1} 2\Big(\frac{i}{m}\Big)^2 \Big|2-3\frac{i}{m}\Big| \\ &\asymp m^2 \int_0^1 2x^2|2-3x|dx\end{aligned}$$

and we get the bound for $\|A\|_2$.

□

Proof of lemma 4.3. The proof of lemma 4.3 is very similar to the proof of lemma 4.2. The decomposition of ζ_t is easily realized after noticing that

$$\begin{aligned}\zeta_t &= \{\hat{\nabla}_{\theta\theta}\mathcal{R}^\star(\theta,t)\}^{-1}\hat{\nabla}_{\theta t}\mathcal{R}^\star(\theta,t) - \{\nabla_{\theta\theta}\mathcal{R}^\star(\theta,t)\}^{-1}\nabla_{\theta t}\mathcal{R}^\star(\theta,t) \\ &= \{X^\top X/n\}^{-1}X^\top\Big\{\sum_{i=0}^{p-1}\beta_i y_{t-ih}\Big\}/n - \dot{\theta}^\star(t) \\ &= \{X^\top X/n\}^{-1}X^\top\Big\{\sum_{i=0}^{p-1}\beta_i X\theta^\star(t-ih) + \sum_{i=0}^{p-1}\beta_i X\epsilon(t-ih)\Big\}/n - \dot{\theta}^\star(t) \\ &= \sum_{i=0}^{p-1}\beta_i\theta^\star(t-ih) - \dot{\theta}^\star(t) + \{X^\top X/n\}^{-1}\sum_{i=0}^{p-1}\beta_i X\epsilon(t-ih)\end{aligned}$$

We now compute the optimal weights $\{\beta_j\}_{j=0}^{p-1}$ in the weighting scheme (4.5) which we require to establish the bounds on C and D. We define the Lagrangian problem:

$$\mathbf{L} = \sum_{j=0}^{p-1}\Big\{\frac{\beta_j^2}{2} - \lambda_1\beta_j - \lambda_2 j\beta_j\Big\} - \frac{\lambda_2}{h} \tag{A.7}$$

where λ_1 and λ_2 are Lagrangian constants, and notice that at optimum it holds:

$$\partial_j L = \beta_j - \lambda_1 - \lambda_2 j = 0, \text{ or } \beta_j = \lambda_1 + \lambda_2 j\,. \tag{A.8}$$

From the above equations we obtain

$$0 = \sum_{j=0}^{p-1}\beta_j = p\lambda_1 + \frac{p(p-1)\lambda_2}{2}\,, \tag{A.9}$$

and

$$-\frac{1}{h} = \sum_{j=0}^{p-1} j\beta_j = \frac{p(p-1)\lambda_1}{2} + \frac{p(p-1)(2p-1)\lambda_2}{6}\,. \tag{A.10}$$

The above two equations has solutions $\lambda_1 = \frac{6}{p(p+1)h}$ and $\lambda_2 = -\frac{12}{p(p^2-1)h}$ which finally implies the optimal weights are

$$\beta_j = \frac{6(p-1-2j)}{p(p^2-1)h}\,. \tag{A.11}$$

Bounds on C and D As in the proof of lemma 4.2 we notice that

$$\mathbb{E}[\|D\|_2^2] = \sum_{j=0}^{p-1}\beta_j^2\mathbb{E}[\|X^\top\epsilon(t-ih)/n\|_2^2] = \Big\{\sum_{j=0}^{p-1}\beta_j^2\Big\}\tau^2\mathrm{TR}\{X^\top X/n^2\}$$

where

$$\begin{aligned}
\sum_{j=0}^{p-1} \beta_j^2 &= \sum_{j=0}^{p-1} \frac{36(p-1-2j)^2}{p^2(p^2-1)^2h^2} \\
&\asymp \sum_{j=0}^{p-1} \frac{36(p-2j)^2}{p^6h^2} \\
&= \sum_{j=0}^{p-1} \frac{36(1-2j/p)^2}{p^4h^2} \\
&\asymp \frac{1}{p^3h^2} \int_0^1 36(1-2x)^2 dx = \mathcal{O}(p^{-3}h^{-2})\,.
\end{aligned}$$

Drawing our attention to C we notice that

$$\begin{aligned}
C &= \sum_{i=0}^{p-1} \beta_i \theta^\star(t-ih) - \dot{\theta}^\star(t) \\
&= \Big\{\cancelto{0}{\sum_{i=0}^{p-1} \beta_i}\Big\}\theta^\star(t) - h\Big\{\cancelto{-1/h}{\sum_{i=0}^{p-1} i\beta_i}\Big\}\dot{\theta}^\star(t) + \frac{h^2}{2}\Big\{\sum_{i=0}^{p-1} i^2\beta_i\Big\}\ddot{\theta}^\star(\tilde{t}_i) - \dot{\theta}^\star(t) \\
&= \frac{h^2}{2}\Big\{\sum_{i=0}^{p-1} i^2\beta_i\Big\}\ddot{\theta}^\star(\tilde{t}_i)
\end{aligned}$$

where $\tilde{t}_i$ is a number in between t and $t-ih$. Hence,

$$\begin{aligned}
\|C\|_2 &\le \frac{h^2}{2}\Big\{\sum_{i=0}^{p-1} i^2|\beta_i|\Big\} \max_{t\ge 0} \|\ddot{\theta}^\star(t)\|_2 \\
&= \frac{1}{2} \max_{t\ge 0} \|\ddot{\theta}^\star(t)\|_2 \sum_{i=0}^{p-1} h^2 i^2 |\beta_i| \\
&= \max_{t\ge 0} \|\ddot{\theta}^\star(t)\|_2 \sum_{i=0}^{p-1} h^2 i^2 \frac{3|p-1-2i|}{p(p^2-1)h} \\
&= 3 \max_{t\ge 0} \|\ddot{\theta}^\star(t)\|_2 hp \sum_{i=0}^{p-1} (i/p)^2 \frac{3|1-2(i/p)|}{p} \\
&= 3 \max_{t\ge 0} \|\ddot{\theta}^\star(t)\|_2 hp \int_0^1 x^2|1-2x|dx = \max_{t\ge 0} \|\ddot{\theta}^\star(t)\|_2 \mathcal{O}(hp)
\end{aligned}$$

□

B ESTIMATES OF DERIVATIVES FOR OBJECT TRACKING PROBLEM: SUBSECTION 4.3

Recall that in the object tracking problem, we have n-sensors placed at $X_1, \ldots, X_n$. At time t, we observe a noisy version of the distance of the true object $\theta^\star(t)$ from the sensors:

$$Y_{i,t} = f(\|X_i - \theta_t^\star\|^2) + \epsilon_{i,t}\,.$$

where $\epsilon_{i,t}$'s are independent centered errors with variance σ^2. The risk function we seek to minimize to track the object is a quadratic risk function:

$$\begin{aligned}\mathcal{R}(\theta, t) &= \frac{1}{2n}\sum_{j=1}^{n}\mathbb{E}\left[(Y_{i,t} - f(\|X_i - \theta\|^2))^2\right] \\ &= \frac{\sigma^2}{2} + \frac{1}{2n}\sum_{i=1}^{n}\left(f(\|X_i - \theta\|^2) - f(\|X_i - \theta_t^\star\|^2)\right)^2\end{aligned}$$

as presented in Subsection 4.3. The gradient of the risk function is:

$$\nabla_\theta \mathcal{R}(\theta, t) = \frac{2}{n}\sum_{j=1}^{n}\left\{\left[f(\|X_i - \theta\|^2) - f(\|X_i - \theta_t^\star\|^2)\right] f'(\|X_i - \theta\|^2)(\theta - X_i)\right\}$$

We estimate the gradient from the observed responses using a moving average based method akin to Subsection 4.1:

$$\hat{\nabla}_\theta \mathcal{R}(\theta, t) = \tfrac{2}{n}\textstyle\sum_{j=1}^{n}\left\{\left[f(\|X_i - \theta\|^2) - \sum_{j=0}^{m-1}\alpha_j Y_{i,t-jh}\right] f'(\|X_i - \theta\|^2)(\theta - X_i)\right\}$$

The crux in the gradient estimation is that the unknown $f(\|X_i - \theta_t^\star\|^2)$ is estimated as

$$f(\|X_i - \theta_t^\star\|^2) \approx \textstyle\sum_{j=0}^{m-1}\alpha_j Y_{i,t-jh} \tag{B.1}$$

where denoting $g(t) = f(\|X_i - \theta_t^\star\|^2)$ we notice that

$$\sum_{j=0}^{m-1}\alpha_j Y_{i,t-jh} = \sum_{j=0}^{m-1}\alpha_j\{g(t) - jhg'(t) + \frac{j^2h^2}{2}g''(\tilde{t}_j) + \sum_{j=0}^{m-1}\alpha_j\epsilon_{i,t-jh}$$

for some $\tilde{t}_j \in [t - jh, t]$. Since $\sum\alpha_j = 1$ and $\sum j\alpha_j = 0$ we understand that $\sum_{j=0}^{m-1}\alpha_j Y_{i,t-jh}$ approximates $g(t) = f(\|X_i - \theta_t^\star\|^2)$. The Hessian of the risk function is technically more involved:

$$\begin{aligned}\nabla_{\theta\theta}\mathcal{R}(\theta, t) = &\frac{2}{n}\sum_{i=1}^{n}\left\{f(\|X_i - \theta\|^2) - f(\|X_i - \theta_t^\star\|^2)\right\} f'(\|X_i - \theta\|^2) I_3 \\ &+ \frac{4}{n}\sum_{i=1}^{n}\left\{f(\|X_i - \theta\|^2) - f(\|X_i - \theta_t^\star\|^2)\right\} f''(\|X_i - \theta\|^2)(\theta - X_i)(\theta - X_i)^\top \\ &+ \frac{4}{n}\sum_{i=1}^{n}\left(f'(\|X_i - \theta\|^2)\right)^2(\theta - X_i)(\theta - X_i)^\top.\end{aligned}$$

We also estimate it using moving average based method as in the case of time varying linear regression model in Subsection 4.1 by replacing $f(\|X_i - \theta_t^\star\|^2)$ with $\sum_{j=0}^{m-1}\alpha_j Y_{i,t-jh}$ as seen in (B.1). The choice of $\{\alpha_i\}$ remains same as mentioned in Subsection 4.1.

C Extra details for application

In Figure 2 we include a plot for the run-time (in second) and their error-bars over 10 repetitions for gradient descent and predictor-corrector methods in the regression model §4.1 and object tracking §4.3 applications.

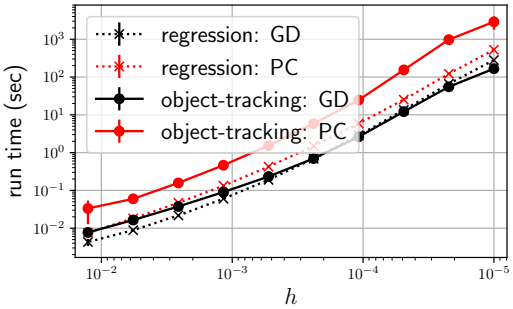

Figure 2: Run time (in sec) for gradient descent and predictor-corrector algorithm in regression model §4.1 and object tracking §4.3.