The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Cogra: Concept-Drift-Aware Stochastic Gradient Descent for Time-Series Forecasting

Kohei Miyaguchi
The University of Tokyo, Tokyo, Japan
kohei_miyaguchi@mist.i.u-tokyo.ac.jp

Hiroshi Kajino
IBM Research – Tokyo, Tokyo, Japan
kajino@jp.ibm.com

Abstract

We approach the time-series forecasting problem in the presence of concept drift by automatic learning rate tuning of stochastic gradient descent (SGD). The SGD-based approach is preferable to other concept drift algorithms in that it can be applied to any model and it can keep learning efficiently whilst predicting online. Among a number of SGD algorithms, the variance-based SGD (vSGD) can successfully handle concept drift by automatic learning rate tuning, which is reduced to an adaptive mean estimation problem. However, its performance is still limited because of its heuristic mean estimator. In this paper, we present a concept-drift-aware stochastic gradient descent (Cogra), equipped with more theoretically-sound mean estimator called sequential mean tracker (SMT). Our key contribution is that we define a goodness criterion for the mean estimators; SMT is designed to be optimal according to this criterion. As a result of comprehensive experiments, we find that (i) our SMT can estimate the mean better than vSGD's estimator in the presence of concept drift, and (ii) in terms of predictive performance, Cogra reduces the predictive loss by 16–67% for real-world datasets, indicating that SMT improves the prediction accuracy significantly.

1 Introduction

This work is concerned with *online* time-series forecasting in a *concept-drifting* environment, where a probability distribution generating the data may change over time. Concept drift is ubiquitous in real-world time-series (Tsymbal 2004; Gama et al. 2014). For example, the probability distribution of sensor data will drift when there is a change in the environment where the sensors are distributed. For another example, the probability distribution of financial time-series will be also time-variant reflecting the economic climate. These applications motivate us to investigate a method to learn from a concept-drifting environment. In addition, such a learning algorithm needs to be computationally efficient so that it can keep learning while forecasting the future. Otherwise, a trained model will be degraded severely as the environment changes. Therefore, we are interested in a time-series forecasting algorithm that can handle *concept drift* and is *computationally efficient*.

There are two research lines towards achieving our grand goal. The first one is mainly concerned with concept drift adaptation, including instance-based methods (Widmer and Kubat 1996; Klinkenberg and Joachims 2000; Kolter and Maloof 2005; Wang and Abraham 2015) and ensemble methods (Schlimmer and Granger 1986; Street and Kim 2001; Minku and Yao 2012). The instance-based ones judge whether a concept drift occurs, and if it does, they retrain their models using the relevant instances. The ensemble ones keep training a set of multiple predictive models, and when necessary, replace them with new models trained using the recent data. Another research line is based on tracking parameter via stochastic gradient descent algorithms (Almeida et al. 1999; Kushner and Yin 2003; Baydin et al. 2017). While these algorithms are computationally efficient by their nature, one of their technical challenges is the learning rate scheduling, which controls the trade-off between convergence and adaptation. When the environment is stationary, the learning rate should decrease so as to converge to the optimal parameter, and when the environment changes, the learning rate should increase so as to adapt to the changing environment.

When these two lines are compared, the SGD-based ones are preferable in our setting, because it maintains only a *single* predictive model *without retraining*. In particular, we consider that the variance-based stochastic gradient descent (vSGD) (Schaul, Zhang, and LeCun 2013) is promising, because vSGD can balance the convergence-adaptation trade-off without pesky hyperparameter tuning; on the other hand, other SGDs that can control the trade-off typically have hyperparameters. vSGD is designed to minimize the expected loss function $\mathbb{E}_{X_t}[\ell(\theta; X_t)]$ at each time step t. A key observation is that the optimal learning rate is given by $\eta_t^\star = \frac{1}{\mathbb{E}_{X_t}[\nabla^2 \ell(\theta; X_t)]} \frac{\mathbb{E}_{X_t}[\nabla_\theta \ell(\theta; X_t)]^2}{\mathbb{E}_{X_t}[(\nabla_\theta \ell(\theta; X_t))^2]}$. This result indicates that the automatic learning rate scheduling problem is reduced to the adaptive estimation of the first and second order moments of the per-example gradient, $\mathbb{E}[\nabla \ell(\theta; X_t)]$ and $\mathbb{E}[(\nabla \ell(\theta; X_t))^2]$, as well as the curvature of the objective function, $\mathbb{E}[\nabla^2 \ell(\theta; X_t)]$. These statistics are estimated online by moving averages with adaptive weights that control how many past observations should be used for estimation; when the environment is stationary, more observations should be used, and after concept drift, only the recent observations should be used. In this way, in the vSGD paradigm, the automatic learning rate scheduling boils down to the adaptive control of the weights used for the moment estimation.

A downside of vSGD is that its moment estimator is developed without any design principle, and sometimes the estimation fails. As will be shown in Fig. 1, when faced with a concept drift after a long convergence period, a bias persists in the estimator, and it fails to estimate the optimal learning rate $\eta_t^\star$. To this end, we set our research objective to develop a theoretically-sound moment estimator tailored for vSGD.

We present a concept-drift-aware stochastic gradient descent (Cogra), equipped with a theoretically-sound moment estimator called a sequential mean tracker (SMT). Our contribution is that we introduce a design principle for the moment estimator. The principle is derived from the fact that the objective function $\mathbb{E}_{X_t}[\ell(\theta; X_t)]$ is upper-bounded by the estimation error of the statistics defining the optimal learning rate $\eta_t^\star$. This indicates that we can use the estimation error as a goodness of the estimator, *i.e.*, an estimator that minimizes the estimation error is the best for vSGD. SMT is designed based on this idea, and we obtain Cogra by substituting SMT for the existing moment estimator of vSGD.

The effectiveness of our method is empirically validated by extensive simulations. In specific, we design two experiments to answer the following questions: **(Q1)** Does SMT estimate the moments better than the estimator used in vSGD? **(Q2)** Does SMT improve the predictive performance? **(Q3)** When does Cogra outperform the other SGDs? The first experiment, answering **(Q1)**, evaluates the estimation error of a moment on synthetic data. The result shows that SMT decreases the error by 60% in total, measured by squared loss, as compared to vSGD, answering **(Q1)** in the affirmative. The second one, answering **(Q2)** and **(Q3)**, evaluates the predictive performances on both synthetic and real-world data. From the results, we conclude that Cogra performs comparably or better than other methods consistently across a wide range of datasets, and specifically, outperforms the others on real-world datasets with drastic changes.

2 Problem Setting

We first introduce our problem setting. Let $\{X_t\}_{t=1}^{\infty}$ be a stochastic process ($X_t \in \mathcal{X} \subseteq \mathbb{R}^D$). Suppose that at each time step $t = 1, 2, \dots$, we observe $x_t \in \mathcal{X}$, a realization of X_t. For each t, we wish to learn a probabilistic model online that well predicts X_t using the past observations, namely, $p_\theta(X_t \mid x^{t-1})$, where $\theta \in \Theta$ is a model parameter, and $x^{t-1} \overset{\text{def}}{=} x_1, x_2, \dots, x_{t-1}$ is a set of the past observations.

Our key assumption is that the probability distribution generating the observation may drift over time, and the optimal parameter at time step t may not be optimal at the next time step. Therefore, our problem setting is to learn a series of model parameters $\theta_1^\star, \theta_2^\star, \dots$ such that,

$$\theta_t^\star = \operatorname*{argmin}_{\theta \in \Theta} \mathcal{L}_t(\theta) \tag{1}$$

$$\text{where} \quad \mathcal{L}_t(\theta) \overset{\text{def}}{=} \mathbb{E}_{X_t \mid x^{t-1}} \left[\ell(\theta; X_t) \right],$$

$$\ell(\theta; X_t) \overset{\text{def}}{=} -\log p_\theta(X_t \mid x^{t-1}),$$

in an online manner. $\ell(\theta; X_t)$ measures the loss the model with parameter θ suffers when X_t is observed. In the following, we omit X_t from the loss function, and simply denote it by $\ell_t(\theta)$ (the subscript t indicates that the loss is time-dependent because the distribution of $X_t \mid x^{t-1}$ may drift).

3 Existing Method: vSGD

This section briefly reviews the existing method approaching to our problem setting (1), vSGD (Schaul, Zhang, and LeCun 2013). Their assumption is that the optimal parameter $\theta_t^\star$ does not change so often. This motivates us to define a series of model parameters based on SGD with an automatic learning-rate scheduler, because it converges to the optimal parameter when the objective function is stationary, and otherwise it can adapt to the changing environment.

In specific, vSGD updates the model parameter as follows:

$$\theta_t = \theta_{t-1} - \eta_t^\star \nabla \ell_t(\theta_{t-1}),$$

where $\eta_t^\star$ is chosen to minimize $\mathcal{L}_t(\theta)$ with respect to the learning rate for each time step, *i.e.*, $\eta_t^\star = \operatorname{argmin}_\eta \mathcal{L}_t^{\mathrm{SGD}}(\eta)$, where $\mathcal{L}_t^{\mathrm{SGD}}(\eta) \overset{\text{def}}{=} \mathbb{E}_{X_t \mid x^{t-1}} \left[\mathcal{L}_t(\theta_{t-1} - \eta \nabla \ell_t(\theta_{t-1})) \right]$. This approach is equivalent to restricting the feasible set of Problem (1), Θ, to its one-dimensional affine subset $\{\theta_{t-1} - \eta \nabla \ell_t(\theta_{t-1}) \in \Theta \mid \eta \in \mathbb{R}\}$.

They first derive an analytic expression of $\eta_t^\star$ via quadratic approximation of the objective function, and then, construct its estimator that can be computed online.

3.1 Analytic Expression of $\eta_t^\star$ via Approximation

Their key observation is that if the loss function is approximated with a quadratic function, $\eta_t^\star$ is characterized by the moments of the stochastic gradient.

Assume that the objective ℓ_t is Lipschitz smooth and hence it can be majorized by quadratic functions $\tilde{\ell}_t$. Then, by considering Problem (1) with $\tilde{\ell}_t$ in place of ℓ_t, we can safely assume that ℓ_t is quadratic. Also, to simplify the following analysis, assume Θ is one-dimensional[1]. Thus, the expected objective is also quadratic and one-dimensional, $\mathcal{L}_t(\theta) = \frac{h_t}{2}\left[(\theta - \theta^\star)^2 + \sigma^2\right]$. Substituting θ with $\theta_{t-1} - \eta \nabla \ell_t(\theta_{t-1})$ and completing the square with respect to η, we obtain

$$\mathcal{L}_t^{\mathrm{SGD}}(\eta) = \frac{h_t v_t}{2} \left(\eta - \frac{g_t^2}{h_t v_t} \right)^2 + C, \tag{2}$$

where $g_t \overset{\text{def}}{=} \mathbb{E}[\nabla \ell_t(\theta_{t-1})]$, $v_t \overset{\text{def}}{=} \mathbb{E}\left[\nabla \ell_t(\theta_{t-1})^2\right]$, $h_t \overset{\text{def}}{=} \mathbb{E}\left[\nabla^2 \ell_t(\theta_{t-1})\right]$, and C is a constant. Expression (2) immediately yields the optimal learning rate at time t:

$$\eta_t^\star = \frac{g_t^2}{h_t v_t}. \tag{3}$$

Since the first and second moments of the stochastic gradient are not available in general, we need to estimate them using the past gradients; Problem (1) is now reduced to estimation of g_t, v_t and h_t in concept-drifting environments.

[1] This is generalized to multidimensional cases in Section 4.4.

3.2 Online Estimation of $\eta_t^\star$

In vSGD, the moments are estimated by exponential moving averages with adaptive time constants. Let $\overline{g}_t, \overline{v}_t$ and $\overline{h}_t$ be the current estimations of g_t, v_t, and h_t, and let us introduce the current time constant τ_t. Then, they are updated as follows:

$$\begin{aligned}
\overline{g}_t &= (1-\tau_{t-1}^{-1})\overline{g}_{t-1} + \tau_{t-1}^{-1}\nabla\ell_t(\theta_{t-1}),\\
\overline{v}_t &= (1-\tau_{t-1}^{-1})\overline{v}_{t-1} + \tau_{t-1}^{-1}(\nabla\ell_t(\theta_{t-1}))^2,\\
\overline{h}_t &= (1-\tau_{t-1}^{-1})\overline{h}_{t-1} + \tau_{t-1}^{-1}\nabla^2\ell_t(\theta_{t-1}),\\
\tau_t &= \left(1-\overline{g}_t^2/\overline{v}_t\right)\tau_{t-1} + 1.
\end{aligned}$$

3.3 Discussion

vSGD balances the trade-off by adaptively tuning the time constant τ_t. When the optimal model parameter is stationary over time, *i.e.*, $\theta_t^\star \approx \theta^\star$, the model parameter gathers around $\theta^\star$, $\overline{g}_t$ becomes close to 0, and the time constant τ_t increases one by one. This makes the moment estimates more accurate and leads to convergence. When $\theta_t^\star$ moves away from $\theta^\star$ after the convergence, g_t also moves away from 0, which sooner or later moves $\overline{g}_t$ from 0 even if the time constant is large. This decreases the time constant and leads to adaptation.

Although this time constant update rule intuitively seems to be reasonable, we find that it sometimes fails to adapt due to its rather heuristic update rule. We observe that the longer the convergence period is, the slower the adaptation becomes. When the convergence period is long, the moment ratio $\overline{g}_t/\overline{v}_t$ becomes significantly small and τ_t keeps growing. This makes $\overline{g}_t$ to be insensitive to the change in the gradient distribution $\nabla\ell_t$. As a result, $\overline{g}_t$ becomes biased for a while. This phenomenon will be showcased in Figure 1.

4 Proposed Method: Cogra

According to the framework of vSGD, the problem of optimal learning rate is reduced to the problem of estimating the moments of the gradients. Thus, our very remaining concern is how to construct *good* estimators of the moments.

A technical difficulty in designing a good estimator is that the underlying distribution may change in our problem setting. We need to decide how many past observations to be used for estimation; after the underlying distribution changes, only the recent observations should be used, and when the distribution is (temporarily) not changing, we should take more observations into account for more accurate estimation. As we have shown in Section 3.3, vSGD employs a natural, but rather heuristic estimator, which sometimes fails to estimate its target statistic, resulting in a poor performance.

Our idea to deal with this difficulty is to define a goodness of the estimator. We derive a goodness criterion using an upper-bound of the loss we suffer at each time step t, which coincides with the estimation error of $\eta_t^\star$ (Sec. 4.1). Then, we derive an oracle mean estimator based on adaptive moving average that sequentially minimizes the estimation error (Sec. 4.2). Then, we present the sequential mean tracker (SMT), which approximately computes the oracle estimator. Finally, combining this with vSGD leads to the concept-drift-aware stochastic gradient descent (Cogra).

4.1 Goodness of the Moment Estimators

Once we substitute $\overline{\eta}_t$ for $\eta_t^\star$ in Eq. (2), the expected loss is characterized by the estimation error of $\eta_t^\star$,

$$\mathbb{E}_{\overline{\eta}_t}\left[\mathcal{L}_t^{\mathrm{SGD}}(\overline{\eta}_t)\right] = \frac{h_t v_t}{2} r^2(\overline{\eta}_t;\eta_t^\star) + C,$$

where $r(\overline{\eta}_t;\eta_t^\star) = \sqrt{\mathbb{E}_{\overline{\eta}_t}(\overline{\eta}_t - \eta_t^\star)^2}$ is RMSE, or *risk*, of the estimator $\overline{\eta}_t$. Assuming boundedness of the moments, $|\overline{g}_t|, |g_t| \leq G$, $v_t, \overline{v}_t \in [V_{\min}, V_{\max}]$, $h_t, \overline{h}_t \in [\lambda_{\min}, \lambda_{\max}]$, the risk of $\overline{\eta}_t$ is bounded by individual risks of the moments,

$$\begin{aligned}
&r(\overline{\eta}_t;\eta_t^\star) &(4)\\
&= \sqrt{\mathbb{E}_{\overline{\eta}_t}\left[\frac{(\overline{g}_t+g_t)}{\overline{h}_t\overline{v}_t}(\overline{g}_t-g_t) - \frac{g_t^2}{\overline{h}_t\overline{v}_t}\frac{\overline{v}_t-v_t}{v_t} - \frac{g_t^2}{\overline{h}_t v_t}\frac{\overline{h}_t-h_t}{h_t}\right]^2}\\
&\leq \frac{2G\cdot r(\overline{g}_t;g_t)}{V_{\min}\lambda_{\min}} + \eta_t^\star\frac{\kappa^2\cdot r(\overline{v}_t;v_t)}{V_{\min}} + \eta_t^\star\frac{\kappa\cdot r(\overline{h}_t;h_t)}{\lambda_{\min}}, &(5)
\end{aligned}$$

where $\kappa = \max(V_{\max}/V_{\min}, \lambda_{\max}/\lambda_{\min})$. This implies that estimators that minimize the individual risks are favorable in our case. In particular, among three individual risks, we focus on minimizing $r(\overline{g}_t;g_t)$ because the contribution of the others can be relatively negligible when SGD is converging.[2]

4.2 Tracking Mean with Greedy Strategy

We develop a mean estimator that sequentially observes $\nabla_t = \nabla\ell_t(\theta_{t-1})$, and estimates $g_t = \mathbb{E}[\nabla_t]$ with small risks. We consider the following adaptive moving averaging estimator, $\overline{g}_0 = \mathbf{0}$, $\overline{g}_t = a_t\overline{g}_{t-1} + (1-a_t)\nabla_t$, where $a_t \in \mathbb{R}$ denotes the forgetting rate of the estimate.

This method has two desirable properties when applied to our problem setting. First, since $\overline{g}_t$ is defined by recursion, it can be updated with a finite memory. Second, the forgetting rate a_t can adaptively control how many observations are used for estimation; when the mean g_t shifts, by setting a_t to be small, the estimator can forget the past and uses the recent observations for estimation. Thus, the remaining design space is how to choose a_t at each time t.

The main claim in this section is that there exists an optimal forgetting rate, named *oracle rate*, which greedily minimizes the risk of $\overline{g}_t$. The oracle rate is defined as

$$a_t^\star \overset{\mathrm{def}}{=} \operatorname*{argmin}_{a_t\in\mathbb{R}} r^2(\overline{g}_t;g_t).$$

Since the squared risk is convex with respect to a_t, there exists the unique minimizer $a_t^\star$:

$$a_t^\star = \frac{\sigma_t^2 - \epsilon_t}{\sigma_t^2 + \gamma_{t-1}^2 + \delta_t^2 - 2\epsilon_t}, \tag{6}$$

where $\sigma_t^2 = \operatorname{tr}\operatorname{Var}(\nabla_t)$, $\gamma_t^2 = \operatorname{tr}\operatorname{Var}(\overline{g}_t)$, $\delta_t = \left\|\mathbb{E}[\overline{g}_{t-1}] - g_t\right\|$ and $\epsilon_t = \operatorname{tr}\operatorname{Cov}(\overline{g}_{t-1}, \nabla_t)$.

This oracle rate is motivated by its convergence property shown in Theorem 1.[2]

Theorem 1. *Let* $\overline{C}_\tau \overset{\mathrm{def}}{=} \sup_{|s-t|\geq\tau}\frac{|\operatorname{tr}\operatorname{Cov}(\nabla_s,\nabla_t)|}{\sigma_s\sigma_t}$. *Assume* $\overline{C}_1 < 1$, $\lim_{\tau\to\infty}\overline{C}_\tau = 0$ *and* $\sup_t \sigma_t^2 < +\infty$. *Then, if the total variation* $\sum_{i=1}^t \|g_{i+1} - g_i\|$ *is bounded, we have* $\lim_{t\to\infty} r(\overline{g}_t;g_t) = 0$.

[2] Both proofs appear in the extended version.

Algorithm 1 Sequential Mean Tracker, SMT

```
**Input:** Streaming stochastic gradient $\{\nabla_t\}_{t=1}^{\infty}$.
**Output:** Estimates of mean and oracle rate $\{(g_t, \overline{a}_t)\}_{t=1}^{\infty}$.
**Initialization:** $\overline{a}_1 \leftarrow 1/2,\ \overline{g}_1, \overline{\overline{g}}_1 \leftarrow \nabla_1,\ \overline{g^2}_1, \overline{\overline{g}^2}_1 \leftarrow \nabla_1^2$.
**for** each observation $\nabla_t$ $(t = 2, 3, \ldots)$ **do**
    Update adaptive moving averages as Eqs. (9).
    Compute second-order estimates as Eqs. (8).
    Compute forgetting rate $\overline{a}_t$ as Eq. (7).
```

Theorem 1 says that, if the target is moving slowly in terms of the total variation, the short-sighted strategy of choosing $a_t^\star$ makes the risk converge to zero even with potentially strong interdependence in $\{\nabla_t\}_{t=1}^{\infty}$. It also guarantees the converging tendency even with non-converging target $\{g_t\}_{t=1}^{\infty}$. The risk constantly approaches zero over any time period $[t_1, t_2]$ in which $\sum_{t=t_1}^{t_2} \|g_{t+1} - g_t\|$ is small. This can be easily seen by that the proof holds for any time shift $t \mapsto t + h$.

In the next section, we will present a sequential mean tracker (SMT), which uses an *estimate* of the oracle rate. Although Theorem 1 does not directly guarantee the convergence of SMT, it is expected that SMT inherits such a property from the oracle algorithm. The convergence property of SMT as well as its effectiveness when combined with vSGD will be demonstrated in the experiments.

4.3 Sequential Mean Tracker (SMT)

We demonstrate how to track the means g_t sequentially based on the above greedy strategy. Since the oracle rate $a_t^\star$ is unknown, we approximate it by plugging empirical moments in the right-hand side of Eq. (6). Noticing that in our case $\epsilon_t = \mathrm{Cov}(\overline{g}_{t-1}, \nabla_t) \approx 0$ if we assume mixing of ∇_t, we plug zero into ϵ_t, yielding the following estimator,

$$\overline{a}_t \stackrel{\text{def}}{=} \frac{\overline{\sigma}_t^2}{\overline{\sigma}_t^2 + \overline{\gamma}_t^2 + \overline{\delta}_t^2}. \tag{7}$$

All of the empirical moments appearing in the above expression can be estimated as follows:

$$\overline{\sigma}_t^2 = \overline{g^2}_t - \|\overline{g}_t\|^2,\ \overline{\gamma}_t^2 = \overline{\overline{g}^2}_t - \left\|\overline{\overline{g}}_t\right\|^2,\ \overline{\delta}_t = \left\|\overline{\overline{g}}_t - \overline{g}_t\right\|, \tag{8}$$

where we use four recursive moment estimators,

$$\begin{aligned} \overline{g}_t &= \overline{a}_t \overline{g}_{t-1} + (1 - \overline{a}_t)\nabla_t, \\ \overline{\overline{g}}_t &= \overline{a}_{t-1}\overline{\overline{g}}_{t-1} + (1 - \overline{a}_{t-1})\overline{g}_t, \\ \overline{g^2}_t &= \overline{a}_{t-1}\overline{g^2}_{t-1} + (1 - \overline{a}_{t-1})\left\|\nabla_t\right\|^2, \\ \overline{\overline{g}^2}_t &= \overline{a}_{t-1}\overline{\overline{g}^2}_{t-1} + (1 - \overline{a}_{t-1})\left\|\overline{g}_t\right\|^2. \end{aligned} \tag{9}$$

By sequentially updating them, we obtain a series of forgetting rate $\{\overline{a}_t\}_{t=1}^{\infty}$ online. We call this the sequential mean tracker (Algorithm 1). We initialize $\overline{a}_1 = 1/2$, which is optimal if ∇_1 and ∇_2 are i.i.d.

4.4 Cogra

Finally, we present concept-drift-aware stochastic gradient descent (Cogra), which substitutes SMT for the moment estimators of vSGD. Cogra employs SMT to estimate g_t, v_t, and h_t. The estimates are plugged into Eq. (3) to yield the adaptive learning rate $\overline{\eta}_t$, with which the parameters are updated. The detailed algorithm is presented in Algorithm 2.

Algorithm 2 Cogra algorithm

```
**Input:** Losses $\{\ell_t\}_{t=1}^{\infty}$, initial parameter $\theta_0$, SMT $\mathcal{A}$.
**Output:** Parameters $\{\theta_t\}_{t=1}^{\infty}$.
**for** each loss function $\ell_t$ $(t = 1, 2, \ldots)$ **do**
    Incur loss $\ell_t(\theta_{t-1})$
    Update SMT: $(\overline{g}_t, \overline{a}_t) \leftarrow \mathtt{push}(\mathcal{A}, \nabla\ell_t(\theta_{t-1}))$
    Update the rest of the estimates:
    $\overline{v}_t \leftarrow \overline{a}_t\overline{v}_{t-1} + (1 - \overline{a}_t)\nabla\ell_t^2(\theta_{t-1})$,
    $\overline{h}_t \leftarrow \overline{a}_t\overline{h}_{t-1} + (1 - \overline{a}_t)\nabla^2\ell_t(\theta_{t-1})$
    Compute learning rate: $\overline{\eta}_t \leftarrow \overline{g}_t^2\overline{h}_t\overline{v}_t$
    Update parameter: $\theta_t \leftarrow \theta_{t-1} - \overline{\eta}_t\nabla\ell_t(\theta_{t-1})$
```

We use the same forgetting rate $\overline{a}_t$ for $\overline{g}_t, \overline{v}_t, \overline{h}_t$ so that the estimates be statistics of the same function $\hat{\mathcal{L}}_t(\theta) = \sum_{s \le t}\prod_{s<i\le t}(1 - a_i)a_s\ell_s(\theta)$, which is an empirical approximation of the true objective $\mathcal{L}_t$. This makes $\overline{\eta}_t$ the optimal learning rate for $\hat{\mathcal{L}}_t$, and therefore, reduces its estimation error compared to ad-hoc plug-in estimates.

4.5 Multi-dimensional Case

Cogra can be extended to multi-dimensional parameter spaces in the same way as vSGD, resulting in three variants. Cogra-local (Cogra-l) runs Algorithm 2 for each scalar of θ separately. Cogra-block (Cogra-b) ties up the variables into arbitrary blocks and runs Cogra for each of the blocks. In each block, the optimal learning rate is estimated by $\overline{\eta}_t = \|\overline{g}_t\|^2/\overline{h}_t^+\overline{v}_t$, where $\overline{g}_t$, $\overline{v}_t$ and $\overline{h}_t^+$ are the SMT estimates of $\mathbb{E}\nabla\ell_t$, $\mathbb{E}\left\|\nabla\ell_t\right\|^2$, and the largest eigenvalue of $\mathbb{E}\nabla^2\ell_t$, respectively. We compute $\overline{h}_t^+$ using BBProp (LeCun et al. 2012). As a special case of Cogra-b, Cogra-global (Cogra-g) ties up all variables into a single block.

The performance of Cogra depends on the grouping strategy due to the trade-off between the adaptivity and estimation errors. As the block becomes smaller, we can assign different learning rates to the parameters with different properties, but at the same time, the estimation errors increase due to the learge degree of freedom, and as the block becomes larger, the opposite happens. In general, we recommend to use Cogra-b, regarding each vectoral parameter as a block.

4.6 Discussion

Cogra has no hyperparameter other than the grouping strategy as with vSGD. The major difference is that the derivation of $\overline{a}_t$ in Cogra is more theoretically-grounded than τ_t in vSGD. We empirically validate in the experiments that this difference is critical in moment estimation and prediction performances. Still, we should note that $\overline{a}_t$ is based on some heuristics (9), where the forgetting rate $\overline{a}_t$ is reused to estimate the statistics necessary for estimating $\overline{a}_t$ itself. However, it is reasonable since all of the statistics $\overline{\overline{g}}_t, \overline{g^2}_t$ and $\overline{\overline{g}^2}_t$ are functions of ∇_t, and they are likely to share the oracle rate $a_t^\star$.

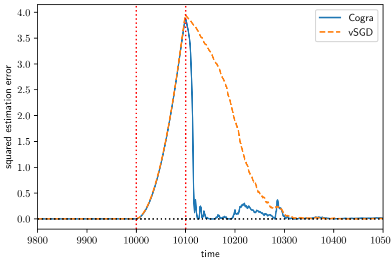


Figure 1: Estimation errors of g_t. The dotted vertical lines show the start and end of the change. Cogra (blue solid) achieves smaller errors than vSGD (orange dashed).

5 Related Work

This section reviews the automatic learning rate scheduling methods and clarifies our contribution to the literature.

A decent amount of studies have leveraged SGD for learning time-varying environments. Most of them deal with non-stationarity by continuously changing the learning rate using meta SGD (AdaptiveSA (Kushner and Yang 1995), Almeida (Almeida et al. 1999), and Hypergradient-Descent (Baydin et al. 2017)). The main drawback is that the learning rate of meta SGD is highly sensitive to data and models, and we still need hyper-hyperparameter tuning.

AdaGrad (Duchi, Hazan, and Singer 2011), RMSProp (Tieleman and Hinton 2012), and ADAM (Kingma and Ba 2014) are ones of the most widely adopted methods for controlling the learning rates adaptively. While these methods are designed mainly to accelerate convergence in a stationary environment, we empirically observe that some of them can adapt to a changing environment. Their main drawback is that one needs to carefully optimize their hyperparameters depending on data and models. vSGD (Schaul, Zhang, and LeCun 2013), on the other hand, is designed to be free of hyperparameters. While vSGD is also designed for stationary environments, its capability with concept drift has been investigated using a toy example in the original paper.

Our contributions are that (i) we develop Cogra by refining vSGD so as to handle the non-stationarity more accurately, and that (ii) we conduct extensive experiments to study whether Cogra as well as the methods described above can handle the non-stationarity.

6 Experiments

So far, we have enhanced vSGD by replacing its mean estimator with SMT. In this section, we aim to empirically confirm the questions **(Q1)**, **(Q2)**, and **(Q3)** raised in Sec. 1. For this purpose, we design two experiments. First, we compares the errors of estimated means of SMT with those of vSGD to answer **(Q1)**. Second, we compare the predictive performance of Cogra against the existing methods listed in Sec. 5 to answer **(Q2)** and **(Q3)**. These experiments will clarify when and why Cogra performs better than the other methods.

Predictive Model. We employ vector autoregression, VAR(p) (Lütkepohl 2005), whose conditional density is given by $p(x_t|x^{t-1}; \mu, W, \Sigma) = \mathcal{N}[\mu + W \operatorname{vec}(z_t), \Sigma], x_t \in \mathbb{R}^d$, where $z_t = (x_{t-1}, x_{t-2}, \ldots x_{t-p}) \in \mathbb{R}^{d \times p}$ is lagged observations, $\mu \in \mathbb{R}^d$ is the bias, $W \in \mathbb{R}^{d \times dp}$ is the autoregressive coefficients, and $\Sigma \in \mathbb{S}^d_+$ is the noise covariance. Model parameters to be learned are W and μ (thus, $\theta = \{W, \mu\}$ in Problem (1)), and Σ is fixed to the identity matrix. We refer to VAR as AR if $d = 1$.

Note that the loss function of VAR is quadratic, and hence, it satisfies all of the assumptions made by the SGDs we compare in our experiments; if some of the assumptions were violated, it would make it harder to study the experimental results, because the performance difference can be attributed to the violation, not to the SGDs.

6.1 SMT Performance

To see that the ability of SMT to track the mean of gradients, we compute the difference between the estimated and true first moments, namely, $\|\overline{g}_t - \mathbb{E}[\nabla \ell_t]_{\theta=\theta_{t-1}}\|_2^2$ using a synthetic dataset. We generate a one-dimensional time-series of length 20,000 using AR(0) with time-varying mean μ_t. Initially, μ_t is set to -1, and is linearly varied from -1 to 1 from time 10,000 to 10,100, and is set to 1 afterward. As for the predictive model, we also employ AR(0).

Figure 1 shows that, while the estimation error of vSGD is persistent even after hundreds steps from the end point of the change, that of Cogra rapidly decreases immediately after the change ends. This result indicates that SMT works correctly and achieves lower risk than vSGD, as expected from our theoretical analysis, answering **(Q1)** in the affirmative.

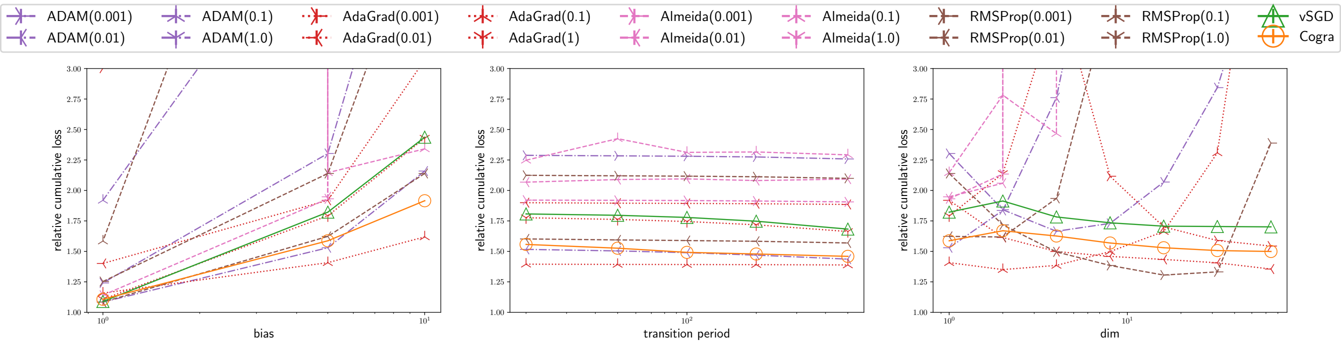
6.2 Predictive Performance

To see whether SMT contributes to the performance of SGDs in terms of time-series forecasting, we compare Cogra with existing SGDs including vSGD in terms of their predictive performance. We conduct two lines of experiments: one with synthetic datasets, in which we inspect the performance of SGDs on a wide variety of datasets, and the other with real-world datasets, in which we verify if our theoretical idea is useful in real environments.

Methods Compared. We compare Cogra with AdaGrad, ADAM, Almeida, RMSProp, and vSGD. For AdaGrad, ADAM, and RMSProp, we employ multiple initial learning rates, fixing the other hyperparameters to be as recommended as in the original papers. The initial learning rates are searched over $\{10^{-x}\}_{x=0}^{3}$. For RMSProp, we add 10^{-4} in the real-world experiments so as to show that the best rate resides inside the search space, not on its boundary.

Almeida requires careful tuning of the initial learning rate and the hyper learning rate. We fix the hyper-learning rate as 10^{-3}, 10^{-2}, and 10^{-1}, and we search the initial learning rate so that the model parameters do not diverge.

For vSGD, we slightly modify the algorithm so that memory size τ_t does not fall below threshold $\tau_{\min} = 2$. This modification is necessary to avoid from τ_t being stuck to 1, which happens after the data distribution radically changes.



(a) Vary bias μ ($d = 1, T = 0$). (b) Vary transition period T ($d = 1, \mu = 5$). (c) Vary dimension d ($T = 0, \mu = 5$).

Figure 2: Performance dependency on dataset parameters. The vertical axes show relative cumulative loss.

Table 1: Predictive performance with real-world datasets. For each table, the first and second row describe methods and hyperparameters to be tuned, and the third to the seventh rows show relative cumulative predictive losses.

Stationary SGD	AdaGrad (initial LR)				ADAM (initial LR)				RMSProp (initial LR)				
	0.001	0.01	0.1	1.0	0.001	0.01	0.1	1.0	0.0001	0.001	0.01	0.1	1.0
A(1)	324	4.58	1.26	6.77	12.1	2.47	7.68	174	101	9.82	4.19	253	2.53e+4
A(2)	169	3.00	1.17	4.90	6.55	2.06	8.37	146	54.3	5.38	2.72	129	1.29e+4
A(3)	312	5.38	1.32	6.92	13.0	2.73	7.09	91.2	98.5	10.4	3.37	158	1.59e+4
EEG	56.1	2.25	3.32	258	9.18	3.41	32.3	3.14e+3	69.1	7.09	105	1.04e+4	1.04e+6
GAS	10.4	3.68	1.19	57.0	7.30	2.21	4.72	341	10.9	6.40	9.39	824	8.24e+4

Non-stationary SGD	Almeida (initial LR, hyper-LR)			vSGD (LR sharing strategy)			Cogra (LR sharing strategy)		
	(1e-9,1e-3)	(1e-9,1e-2)	(1e-9,1e-1)	block	global	local	block	global	local
A(1)	4.24	3.82	2.53	1.00	2.32e+5	5.76	**0.797**	5.96e+6	0.903
A(2)	2.93	2.66	1.90	1.00	2.20e+5	6.34	**0.839**	3.31e+6	1.14
A(3)	4.35	3.96	2.72	1.00	4.12e+5	9.87	**0.800**	6.04e+6	0.938
EEG	1.61	1.61	1.60	1.00	6.50e+6	9.44e+17	**0.612**	1.47e+9	4.32e+4
GAS	1.26e+85	1.16e+66	2.05e+20	1.00	5.81e+7	3.66e+18	**0.328**	2.48e+10	7.87e+6

Experimental Procedure. We repeat the following procedure for each time step of a time-series; the model predicts the next data point, receives the observation, computes the loss, and updates the model parameters using SGD.

We measure the predictive performance by the cumulative loss relative to the *benchmark* cumulative loss. For synthetic data, we use the theoretical lower-bound of the cumulative loss as the benchmark one. For real-world data, we use the cumulative loss of vSGD-b as the benchmark one.

Performance on Synthetic Datasets

In order to investigate when Cogra performs better than the other SGDs, we vary three parameters of a time-series and compare the performance of each of SGDs. Throughout this experiment, we employ VAR(4) as a predictive model.

Datasets. We design a synthetic dataset containing one change point. It has three parameters: dimension d, transition period $T \in \mathbb{Z}_+$, and bias $\mu \in \mathbb{R}^d$. T controls the type of changes; if $T = 0$, the data distribution changes abruptly, which we call an *abrupt change*, and if T is large, the data distribution changes gradually, which we call a *gradual change*. μ controls the magnitude of the change, and d controls the number of parameters to be estimated.

We synthesize d-dimensional VAR(4) by concatenating d AR(4), $x_t = \mu_t + \sum_{i=1}^{4} a_{i,t} x_{t-i} + \epsilon_t$ ($\epsilon_t \sim \mathcal{N}(0, 1)$). We prepare two AR(4) parameters:

$$\mu_t = -\mu,\ \mathbf{a}_t = (0.85, -0.26, 0.0335, -0.0015), \tag{10}$$
$$\mu_t = \mu,\ \mathbf{a}_t = (0.25, 0.07, -0.0115, -0.0015), \tag{11}$$

where we denote $\mathbf{a}_t \overset{\text{def}}{=} (a_{1,t}, a_{2,t}, a_{3,t}, a_{4,t})$. For each AR(4), we first generate a time-series of length 10,000 with parameter (10). Then, we generate a time-series of length T by linearly varying parameters from (10) to (11), and generate a time-series of length 10,000 with parameter (11).

We define the reference parameter set as $d = 1$, $T = 0$, $\mu = 5.0$. We vary each of the parameters while fixing the other two to see the performance dependency on each parameter. The parameters are varied as $d \in \{2^0, 2^1, \ldots, 2^6\}$, $T \in \{0, 20, 50, 100, 200, 500\}$, and $\mu \in \{1.0, 5.0, 10.0\}$. For each setting, we run the experiment ten times and report its mean.

Results. Figure 2 shows the performance dependency on the dataset parameters. We obtain the following three insights on the performance of each SGD. First of all, we observe Cogra performs consistently better than vSGD in almost all settings. This validates our idea to substitute SMT for the moment estimator of vSGD. Second, while AdaGrad(0.1) performs the best in Figures 2(a) and 2(b), its performance severely degrades as the dimension increases as Figure 2(c) shows; on the other hand, Cogra performs more robustly than AdaGrad(0.1). This property is essential in our problem setting, because often we are not able to foresee the data properties before we encounter it. Third, from Figures 2(a), 2(b), and 2(c), we observe that Cogra has less dependency on the magnitude of a change and the dimension, and its performance slightly improves when the change is gradual.

In summary, we conclude that, in most cases, Cogra performs better than vSGD (thus, answering **(Q2)** in the affirmative), the performance of Cogra is much less sensitive to the dataset parameters than the other SGDs including vSGD, and thanks to the insensitivity, Cogra performs better than the others when the magnitude of a change is large and/or the transition period is large, which answers **(Q3)**.

Performance on Real-world Datasets

Finally, we verify the performance of Cogra with real-world datasets. We employ $\mathrm{VAR}(3)$ as a predictive model and use three datasets from the UCI repository (Lichman 2013).

Datasets. The *activity recognition dataset* records three dimensional acceleration data (Casale, Pujol, and Radeva 2012). Each participant is engaged in seven activities, and therefore, this dataset is considered to contain abrupt changes. We downsample the original data from 52 Hz to 5.2 Hz, and we employ the first three participants' data (A(1)–A(3)) because the other participants' data result in similar results. The *EEG eye state dataset* records 14 dimensional EEG measurements. Since the eye state (whether open or close) is correlated with the EEG distribution, this dataset is considered to contain abrupt changes. The *gas sensor array dataset* (Fonollosa et al. 2015) collects the recordings of 18 chemical sensors exposed to the gas mixture with dynamically-varying concentrations. The gas concentrations are increased, decreased, or set to zero at random times, and therefore, this dataset is considered to contain gradual changes. We use the first 50,000 data points.

Results. Table 1 shows the relative cumulative losses. For all of the datasets, Cogra-b outperforms the other methods. This confirms Cogra's capability of adjusting learning rate in response to real-world concept drifts. In the following, we discuss why the block estimation (Cogra-b and vSGD-b) performs better than the local and global estimations, and why Cogra-b outperforms the other SGDs.

The global estimation (Cogra-g and vSGD-g) fails because of the significant difference of the scales of the Hessians of W and μ. For example, the Hessian of W is larger than that of μ by more than a factor of 10^6 in the activity dataset. Hence, estimating them with one global parameter results in the poor estimation of $\eta_t^\star$. The local estimation (Cogra-l and vSGD-l) fails specifically on the EEG and gas datasets. Since the dimensionality of these datasets are relatively high, 14 and 18, there are huge degrees of freedom in the local Hessian estimation, $D = 602$ and 990 respectively, while that of the activity dataset is only 30. This makes the estimation of $\eta_t^\star$ unstable, affecting the predictive performance negatively. The block estimation (Cogra-b and vSGD-b) estimates the Hessians of μ and W separately, but using one parameter for each regardless of the dimensionality of data. Therefore, it does not much suffer either from the difference of the scale of Hessians or from the high-dimensional time-series.

We notice that Cogra-b outperforms the others more clearly on the EEG and gas datasets. Based on the study on the synthetic dataset, we consider that this is because both datasets contain larger changes than the activity dataset. The mean of the EEG dataset is about twice as large as that of the activity dataset, requiring us to adapt to the mean at the beginning. The values of the gas dataset change much more drastically than the others, which requires us to adapt to the drastic changes repeatedly. We have expected that Almeida should be able to handle such drastic changes, but it does not, especially on the gas dataset. This is mainly because its initial learning rate is highly sensitive.

7 Conclusion and Future Work

We have considered online forecasting of streaming data in concept-drifting environments. As suggested by vSGD, this problem can be reduced to estimating moments of stochastic gradients. Our finding is that moment estimators of vSGD are heuristically designed and can yield a bias, which also exerts a negative impact on the predictive performance in our problem setting. To this end, we have developed a moment estimator called sequential mean tracker (SMT). SMT is designed to minimize the estimation error greedily and is guaranteed to converge. By substituting SMT for the moment estimator of vSGD, we have obtained an adaptive SGD without any hyperparameters, namely Cogra. Through the comprehensive experiments, we have obtained three results. First, the estimation error of SMT converges to zero much faster than that of vSGD. Second, Cogra is robust to the properties of the dataset. Third, Cogra outperforms a number of existing SGDs with a wide range of real-world datasets, specifically on radically changing data. These three results strongly support the effectiveness of Cogra in concept-drifting environments.

One future direction is to investigate the applicability of Cogra to more complex objective functions. Although we have validated the effectiveness of Cogra using VAR, it is still open whether the same holds for more complex models such as RNNs. Thus, it is valuable to clarify the applicability by extensive experiments and/or to extend our framework by relaxing the assumptions on the objective function.

Another direction is to apply Cogra to reinforcement learning. For example, when learning the action-value functions in SARSA, the objective function tends to be highly non-stationary because the policy is updated during learning. Cogra is expected to handle such non-stationarity well and achieve faster and stabler learning of policies.

Acknowledgments This work was supported by JST CREST Grant Number JPMJCR1304, Japan.

References

Almeida, L. B.; Langlois, T.; Amaral, J. D.; and Plakhov, A. 1999. Parameter adaptation in stochastic optimization. In *On-Line Learning in Neural Networks*. Cambridge University Press. 111–134.

Baydin, A. G.; Cornish, R.; Rubio, D. M.; Schmidt, M.; and Wood, F. 2017. Online learning rate adaptation with hypergradient descent. *arXiv preprint arXiv:1703.04782*.

Casale, P.; Pujol, O.; and Radeva, P. 2012. Personalization and user verification in wearable systems using biometric walking patterns. *Personal and Ubiquitous Computing* 16(5):563–580.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12:2121–2159.

Fonollosa, J.; Sheik, S.; Huerta, R.; and Marco, S. 2015. Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring. *Sensors and Actuators B: Chemical* 215:618–629.

Gama, J. A.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; and Bouchachia, A. 2014. A survey on concept drift adaptation. *ACM Comput. Surv.* 46(4):44:1–44:37.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klinkenberg, R., and Joachims, T. 2000. Detecting concept drift with support vector machines. In *ICML*, 487–494.

Kolter, J. Z., and Maloof, M. A. 2005. Using additive expert ensembles to cope with concept drift. In *Proceedings of the 22nd international conference on Machine learning*, 449–456. ACM.

Kushner, H. J., and Yang, J. 1995. Analysis of adaptive stepsize sa algorithms for parameter tracking. *IEEE Transactions on Automatic Control* 40(8):1403–1410.

Kushner, H., and Yin, G. G. 2003. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media.

LeCun, Y. A.; Bottou, L.; Orr, G. B.; and Müller, K.-R. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*. Springer. 9–48.

Lichman, M. 2013. UCI machine learning repository.

Lütkepohl, H. 2005. *New introduction to multiple time series analysis*. Springer Berlin Heidelberg.

Minku, L. L., and Yao, X. 2012. Ddd: A new ensemble approach for dealing with concept drift. *IEEE transactions on knowledge and data engineering* 24(4):619–633.

Schaul, T.; Zhang, S.; and LeCun, Y. 2013. No more pesky learning rates. In *Proceedings of the 30th International Conference on Machine Learning*.

Schlimmer, J. C., and Granger, R. H. 1986. Incremental learning from noisy data. *Machine learning* 1(3):317–354.

Street, W. N., and Kim, Y. 2001. A streaming ensemble algorithm (sea) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 377–382. ACM.

Tieleman, T., and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4(2):26–31.

Tsymbal, A. 2004. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin* 4(C):2004–15.

Wang, H., and Abraham, Z. 2015. Concept drift detection for streaming data. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, 1–9. IEEE.

Widmer, G., and Kubat, M. 1996. Learning in the presence of concept drift and hidden contexts. *Machine Learning* 23(1):69–101.