Proximal Online Gradient is Optimum for Dynamic Regret

Anonymous Authors[1]

Abstract

1. Introduction

For any online algorithm $A \in \mathcal{A}$, the previous dynamic regret $\widetilde{\mathcal{R}}_T^A$ is defined by

$$\widetilde{\mathcal{R}}_T^A = \sum_{i=1}^{n} \sum_{t=1}^{T} \left(g_{i,t}(\mathbf{x}_{i,t}) - g_{i,t}(\mathbf{x}_t^*) \right), \tag{1}$$

2. Related work

Online learning has been studied for decades of years. The static regret of a sequential online convex optimization method can achieve $\mathcal{O}\left(\sqrt{T}\right)$ and $\mathcal{O}\left(\log T\right)$ bounds for convex and strongly convex loss functions, respectively (Hazan, 2016; Shalev-Shwartz, 2012). Recently, both the decentralized online learing and the dynamic regret have drawn much attention due to their wide existence in the practical big data scenarios.

2.1. Decentralized online learning

Online learning in a decentralized network has been studied in (Shahrampour and Jadbabaie, 2018; Kamp et al., 2014; Koppel et al., 2018; Zhang et al., 2018a; 2017b; Xu et al., 2015; Akbari et al., 2017; Lee et al., 2016; Nedi et al., 2015; Lee et al., 2018; Benczúr et al., 2018; Yan et al., 2013). Shahrampour and Jadbabaie (2018) studies decentralized online mirror descent, and provides $\mathcal{O}\left(n\sqrt{nTM}\right)$ dynamic regret. Here, n, T, and M represent the number of nodes in the newtork, the number of iterations, and the budget of dynamics (defined in (3)), respectively. When the Bregman divergence in the decentralized online mirror descent is chosen appropriately, the decentralized online mirror descent becomes identical to the decentralized online gradient descent. Using the same definition of dynamic regret (defined in (1)), our method obtains $\mathcal{O}\left(n\sqrt{TM}\right)$ dynamic regret for a decentralized online gradient descent, which is better than $\mathcal{O}\left(n\sqrt{nTM}\right)$ in Shahrampour and Jadbabaie (2018). The improvement of our bound benefits from a better bounded network error (see Lemma **??**). Kamp et al. (2014) studies decentralized online prediction, and presents $\mathcal{O}\left(\sqrt{nT}\right)$ static regret. It assumes that all data, used to yielded the loss, is generated from an unknown distribution. The strong assumption is not practical in the dynamic environment, and thus limits its novelity for a general online learning task. Additionally, many decentralized online optimization methods are proposed, for example, decentralized online multi-task learning (Zhang et al., 2018a), decentralized online ADMM (Xu et al., 2015), decentralized online sub-gradient descent (Akbari et al., 2017), decentralized continuous-time online saddle-point method (Lee et al., 2016), decentralized online Nesterov's primal-dual method (Nedi et al., 2015; Lee et al., 2018). Those previous methods are proved to yield $\mathcal{O}\left(\sqrt{T}\right)$ static regret, which do not have theoretical guarantee of regrets in the dynamic environment. Besides, Yan et al. (2013) provides necessary and sufficient conditions to preserve privacy for decentralized online learning methods, which is interesting to extend our method to be privacy-preserving in the future work.

2.2. Regret in dynamic environment

Dynamic regret has been widely studied for decades of years (Zinkevich, 2003; Hall and Willett, 2015; 2013; Jadbabaie et al., 2015; Yang et al., 2016; Bedi et al., 2018; Zhang et al., 2017a; Mokhtari et al., 2016; Zhang et al., 2018b; György and Szepesvári, 2016; Wei et al., 2016; Zhao et al., 2018). Zinkevich (2003) first defines the reference points $\{\mathbf{x}_t^*\}_{t=1}^T$ satisfying (3), and then proposes an online gradient descent method. The method yields $\mathcal{O}\left(\sqrt{TM}\right)$ by choosing an appropriate learning rate. The following researches achieve the sublinear dynamic regret, but extend it to different reference points. For example, Hall and Willett (2015; 2013) choose the reference points $\{\mathbf{x}_t^*\}_{t=1}^T$ satisfying $\sum_{t=1}^{T-1} \left\| \mathbf{x}_{t+1}^* - \Phi(\mathbf{x}_t^*) \right\| \leq M$, where $\Phi(\mathbf{x}_t^*)$ is the predictive optimal decision variable. When the func-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

tion Φ predicts accurately, a small M is enough to bound the dynamics. The dynamic regret is thus effectively decreased. Jadbabaie et al. (2015); Yang et al. (2016); Bedi et al. (2018); Zhang et al. (2017a); Mokhtari et al. (2016); Zhang et al. (2018b) chooses the reference points $\{\mathbf{y}_t^*\}_{t=1}^T$ with $\mathbf{y}_t^* = \operatorname{argmin}_{\mathbf{z}\in\mathcal{X}} f_t(\mathbf{z})$, where f_t is the loss function at the t-th iteration. György and Szepesvári (2016) provides a new analysis framework, which achieves $\mathcal{O}\left(\sqrt{TM}\right)$ dynamic regret for any given reference points. Besides, Zhao et al. (2018) presents that the lower bound of the dynamic regret is $\mathcal{O}\left(\sqrt{TM}\right)$. Those previous methods define the regret as (1), which is a special case of our definition. When setting $\beta = 1$, we achieve the state-of-the-art regret, that is, $\mathcal{O}\left(\sqrt{TM}\right)$.

In some literatures, the regret in a dynamic environment is measured by the number of changes of a reference point over time. It is usually denoted by shifting regret or tracking regret. (Herbster and Warmuth, 1998; György et al., 2005; Gyorgy et al., 2012; György and Szepesvári, 2016; Mourtada and Maillard, 2017; Adamskiy et al., 2016; Wei et al., 2016; Cesa-Bianchi et al., 2012; Mohri and Yang, 2018; Jun et al., 2017). Both the shifting regret and the tracking regret can be considered as a variation of the dynamic regret, and is usually studied in the setting of "learning with expert advice". But, the dynamic regret is usually studied in a general setting of online setting.

3. Notations

For any $i \in [n]$ and $t \in [T]$, the random variable $\xi_{i,t}$ is subject to a distribution D_t, that is, $\xi_{i,t} \sim D_t$. Besides, a set of random variables $\Xi_{n,T}$ and the corresponding set of distributions are defined by

$$\Xi_{n,T} = \{\xi_{i,t}\}_{1\le i\le n, 1\le t\le T}, \text{ and } \mathcal{D}_T = \{D_t\}_{1\le t\le T},$$

respectively. For math brevity, we use the notation $\Xi_{n,T} \sim \mathcal{D}_T$ to represent that $\xi_{i,t} \sim D_t$ holds for any $i \in [n]$ and $t \in [T]$. $\mathbb{E}$ represents mathematical expectation. ∂ and ∇ represent sub-gradient and gradient operators, respectively. $\|\cdot\|$ represents the ℓ_2 norm in default.

4. Problem formulation

4.1. Setup

For any online algorithm $A \in \mathcal{A}$, define its dynamic regret as

$$\mathcal{R}_T^A = \mathop{\mathbb{E}}_{\Xi_{n,T}\sim\mathcal{D}_T} \left(\sum_{i=1}^n \sum_{t=1}^T f_{i,t}(\mathbf{x}_{i,t};\xi_{i,t}) - f_{i,t}(\mathbf{x}_t^*;\xi_{i,t}) \right), \tag{2}$$

where n is the number of nodes in the decentralized network. The local loss function $f_{i,t}(\mathbf{x};\xi_{i,t})$ is defined by

$$f_{i,t}(\mathbf{x};\xi_{i,t}) := \beta g_{i,t}(\mathbf{x}) + (1-\beta) h_t(\mathbf{x};\xi_{i,t})$$

with $0 < \beta < 1$, and $\xi_{i,t}$ is a random variable drawn from an unknown distribution D_t. Note that $g_{i,t}$ is an adversary loss function, which is yielded by the learning model. $h_t(\cdot;\xi_{i,t})$ is a known loss function, which depends on the random variable $\xi_{i,t}$. The expectation of $h_t(\cdot;\xi_{i,t})$ is a global model, and does not depend on the i-th node.

$\{\mathbf{x}_t^*\}_{t=1}^T$ is the sequence of reference points, and

$$\{\mathbf{x}_t^*\}_{t=1}^T \in \left\{ \{\mathbf{z}_t\}_{t=1}^T : \sum_{t=1}^{T-1} \|\mathbf{z}_t - \mathbf{z}_{t+1}\| \le M \right\}.$$

Here, M is the budget of the dynamics, that is,

$$\sum_{t=1}^{T-1} \left\|\mathbf{x}_{t+1}^* - \mathbf{x}_t^*\right\| \le M. \tag{3}$$

When $M = 0$, all $\mathbf{x}_t^*$s are same, and it degenerates to the static online learning problem. When the dynamic environment changes significantly, M becomes large to model the dynamics. Besides, we denote

$$H_t(\cdot) = \mathop{\mathbb{E}}_{\xi_{i,t}\sim D_t} h_t(\cdot;\xi_{i,t}) \qquad \text{for } \forall i \in [n],$$

and

$$F_{i,t}(\cdot) = \mathop{\mathbb{E}}_{\xi_{i,t}\sim D_t} f_{i,t}(\cdot;\xi_{i,t}).$$

Recall that the previous definition of the dynamic regret is (1). Using (1), the classic online learning in a decentralized network only considers the loss function, i.e., $g_{i,t}$, incurred by the learning model on every node. Comparing with it, our definition of the dynamic regret, i.e., (2), still considers the loss function, i.e., H_t. It is incurred by a global model, which is used to let the decision variables, e.g., $\mathbf{x}_{i,t}$, have some good property in practical scenarios. We present some application scenarios to explain it in Section 4.2.

4.2. Application scenarios

To protect privacy, users prefer to placing their data in the local node, instead of providing it to a centralized server. Decentralized computing provides an alternative method to solve the problem. There is a user named Bob, who subscribes the online music recommendation service.

Online music recommendation with profiling features. In the task, we want to decide whether to recommend some a music to Bob's mobile phone by using two kinds of features.

1. The first kind of features is about Bob's preference to music on the Youtube, which is obtained by using Bob's historical browsing data. For example, these features includes *the latest music ever listened and its player*, *the player whose music is listened most frequently*, and *types of music listened for the past month* etc. Note that Bob's perference to music changes over time, which is usually impacted by the time-varying trends of hot topics in the Internet. The dynamic nature of perference implies that the optimal learning model should change over time. Thus, it is necessary to use dynamic regret to measure the quality of the learning model. In the case, $g_{i,t}(\mathbf{x}_{i,t})$ represents the loss incurred by this kind of features.

2. The second kind of features is about users, who give comments about the music on the Youtube. For example, these features are "gender" and "age", and an instance is *male, 20 years old*, or *female, 18 years old*. Note that user's gender and age do not usually change. All values corresponding to these features can be modeled by an unknown distribution D_t at time t. Since there may be more and more users giving their comments to the music, D_t may change over time. In the case, $h_t(\cdot; \xi_{i,t})$ represents the loss incurred by this kind of features, and $H_t(\cdot) = \mathbb{E}_{\xi_{i,t} \sim D_t} h_t(\cdot; \xi_{i,t})$.

In a nutshell, an instance consists of those two kinds of features, and the label of an instance is *whether Bob listened the music*. A small β means significant attention on the second kind of features.

Suppose we use logistic regression to decide whether to recommend some a music to Bob. Without loss of generality, features corresponding to the preference to music are denoted by the beginning s features. Given a user's browsing record $\mathbf{a}_{i,t}$ and its label $\mathbf{y}_{i,t} \in \{1, -1\}$. In the case, $g_{i,t}(\mathbf{x}) = \log\left(1 + \exp\left(-\mathbf{y}_{i,t}\mathbf{a}_{i,t}^{\mathrm{T}}\hat{\mathbf{I}}\mathbf{x}\right)\right)$, where $\hat{\mathbf{I}}$ is yielded by letting the first s diagonal elements of an identity matrix be 0s. $\xi_{i,t} = \check{\mathbf{I}}\mathbf{a}_{i,t}\mathbf{y}_{i,t}^{\mathrm{T}}$, and $\tilde{h}(\mathbf{x}; \xi_{i,t}) = \log\left(1 + \exp\left(-\xi_{i,t}^{\mathrm{T}}\mathbf{x}\right)\right)$, where $\check{\mathbf{I}}$ is yielded by letting the last $(d-s)$ diagonal elements of an identity matrix be 0s. Here, $\xi_{i,t}$ is drawn form an unknown distribution, that is, $\xi_{i,t} \sim D_t$. In the case, $H_t(\mathbf{x})$ allows the decision variable $\mathbf{x}$ to represent different models to treat those two kinds of features.

Online music recommendation with user-specified privacy protection. In the task, we want to conduct online music recommendation with the user-specified privacy protection for Bob, because he wants to protect his data in the way he likes. We provide several choices for users to make a tradeoff between the accuracy of recommendation and the privacy protection. For example, when we use ϵ-differential privacy, these choices may include *strong privacy, weak accuracy* ($\epsilon = 0.01$), *medium privacy, medium accuracy* ($\epsilon = 0.05$), and *weak privacy, strong accuracy* ($\epsilon = 0.1$). Note that Bob's choice may change over time. For example, he may tolerate weak privacy protection to receive the newest song produced by his favorite player timely, but may want strong privacy protection when seeing a privacy-leaking news from a newspaper. In the case, $g_{i,t}(\mathbf{x}_{i,t})$ represents the loss incurred by the learning model. $h_t(\mathbf{x}_{i,t}; \xi_{i,t})$ represents the loss incurred by some a randomization encryption method, e.g., objective perturbation (Chaudhuri et al., 2011; Wang et al., 2017), to protect the privacy. Since Bob's preference to music may change over time, the optimal recommendation model should change over time. Thus, the dynamic regret is necessary to measure the quality of the model.

Similarly, suppose we want to learn a logistic regression model with the user-specified privacy protection. Given an instance $\mathbf{a}_{i,t} \in \mathbb{R}^d$ and its label $\mathbf{y}_{i,t} \in \{1, -1\}$. In the case, $g_{i,t}(\mathbf{x}) = \log\left(1 + \exp(-\mathbf{y}_{i,t}\mathbf{a}_{i,t}^{\mathrm{T}}\mathbf{x})\right)$. We use the objective perturbation strategy (Chaudhuri et al., 2011; Wang et al., 2017) to protect the privacy. Specifically, we let $h_t(\mathbf{x}; \xi_{i,t}) = \mathbf{x}^{\mathrm{T}}\xi_{i,t}$, where $\xi_{i,t}$ is random variable which is drawn from D_t. The density of $\xi_{i,t}$ is

$$v(\mathbf{x}) = \frac{1}{\lambda}\exp(-\delta_{i,t}\|\mathbf{x}\|).$$

Here, λ is a normalizing constant, $\delta_{i,t}$ is a known function of $\epsilon_{i,t}$ for $\epsilon_{i,t}$-differential privacy (Dwork and Roth, 2014). For example, when $\delta_{i,t} = \epsilon_{i,t}$, $\lambda = ?$.

5. Algorithm

Algorithm 1 DOG: Decentralized Online Gradient method.

Require: The learning rate η, number of iterations T, and the confusion matrix $\mathbf{W}$. $\mathbf{x}_{i,1} = \mathbf{0}$ for any $i \in [n]$.

for $t = 1, 2, ..., T$ **do**
  For the i-th node with $i \in [n]$:
  Predict $\mathbf{x}_{i,t}$.
  Observe the loss function $f_{i,t}$, and suffer loss $f_{i,t}(\mathbf{x}_{i,t}; \xi_{i,t})$.
  Update:
  Query a sub-gradient $\partial f_{i,t}(\mathbf{x}_{i,t}; \xi_{i,t})$.
  $\mathbf{x}_{i,t+1} = \sum_{j=1}^{n}\mathbf{W}_{i,j}\mathbf{x}_{j,t} - \eta\partial f_{i,t}(\mathbf{x}_{i,t}; \xi_{i,t})$.
end for

The decentralized online gradient method, namely DOG, is presented in Algorithm 1. At every iteration, every node needs to collect the decision variable, e.g., $\mathbf{x}_{i,t}$, from its neighbours, and then update its decision variable. Here, $\mathbf{W} \in \mathbb{R}^{n\times n}$ is the confusion matrix. It is a doublely stochastic matrix, which implies that every element of $\mathbf{W}$ is non-negative, $\mathbf{W}\mathbf{1} = \mathbf{1}$, and $\mathbf{1}^{\mathrm{T}}\mathbf{W} = \mathbf{1}^{\mathrm{T}}$. Denote $\bar{\mathbf{x}}_t = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{i,t}$. We can verify that $\bar{\mathbf{x}}_{t+1} =$

$\bar{\mathbf{x}}_t - \frac{\eta}{n}\sum_{i=1}^n \partial f_{i,t}(\mathbf{x}_{i,t};\xi_{i,t})$ (see Lemma **??**).

6. Theoretical analysis

Assumption 1. *We make the following assumptions.*

- *For any $i \in [n]$, $t \in [T]$, and $\mathbf{x}$, there exists a constant G such that*
$$\max\left\{\mathop{\mathbb{E}}_{\xi_{i,t}\sim D_t}\|\nabla h_t(\mathbf{x};\xi_{i,t})\|^2, \|\partial g_{i,t}(\mathbf{x})\|^2\right\} \le G,$$
and
$$\mathop{\mathbb{E}}_{\xi_{i,t}\sim D_t}\|\nabla h_t(\mathbf{x};\xi_{i,t}) - \nabla H_t(\mathbf{x})\|^2 \le \sigma^2.$$
- *For any $\mathbf{x}$ and $\mathbf{y}$, we assume $\|\mathbf{x}-\mathbf{y}\|^2 \le R$.*
- *For any $i \in [n]$ and $t \in [T]$, we assume the function $f_{i,t}$ is convex, but may be non-smooth. Furthermore, we assume the function H_t has L-Lipschitz gradients. In brief, $g_{i,t}$ may be non-convex, non-smooth. H_t is smooth, but may be non-convex. $f_{i,t}$ is convex, but may be non-smooth.*

6.1. Main results

Theorem 1. *Denote $\bar{\mathbf{x}}_t = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_{i,t}$, and constants C_0 and C_1 by*

$$C_0 := \frac{1}{\sqrt{\beta^2+\eta}} + 4;$$
$$C_1 := \frac{\beta}{2\eta} + L + \frac{\sqrt{\beta^2+\eta}}{2\eta} + 2\eta L^2 + C_0(1-\beta)^2 L^2\eta.$$

Using Assumption 1, and choosing $\eta > 0$ in Algorithm 1, we have

$$\mathop{\mathbb{E}}_{\Xi_{n,T}\sim\mathcal{D}_T}\sum_{t=1}^T\sum_{i=1}^n f_{i,t}(\mathbf{x}_{i,t};\xi_{i,t}) - f_{i,t}(\mathbf{x}_t^*;\xi_{i,t})$$
$$\le \eta T\left(n\beta G + (1-\beta)\sigma^2\right) + n(1-\beta)C_0\left(\mathop{\mathbb{E}}_{\Xi_{n,T}\sim\mathcal{D}_T}\sum_{t=1}^T\left(H_t(\bar{\mathbf{x}}_t) - H_t(\bar{\mathbf{x}}_{t+1})\right)\right)$$
$$+ (1-\beta)\frac{nT\eta^2 GC_1}{(1-\rho)^2} + n(1-\beta)C_0\left(4T\beta^2\eta G + \frac{TGL\eta^2}{2}\right) + \frac{n}{2\eta}\left(4\sqrt{RM}+R\right).$$

Corollary 1. *Recall that*
$$C_0 = \frac{1}{\sqrt{\beta^2+\eta}} + 4.$$
Using Assumption 1, and choosing
$$\eta = \sqrt{\frac{nM}{T\left(n\beta G + (1-\beta)\sigma^2\right)}}$$
in Algorithm 1, we have

$$\mathcal{R}_T^{\mathrm{DOG}} \lesssim \sqrt{nMT\left(\beta nG + (1-\beta)\sigma^2\right)} + n(1-\beta)C_0\mathop{\mathbb{E}}_{\Xi_{n,T}\sim\mathcal{D}_T}\sum_{t=1}^T\left(H_t(\bar{\mathbf{x}}_t) - H_t(\bar{\mathbf{x}}_{t+1})\right).$$

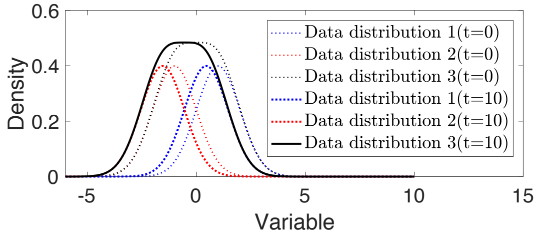


Figure 1. The illustration of the dynmaics caused by the time-varying distributions of data. Data distribution 1 and 2 are Normal distrbutions with 1 variance and $1+\sin(t)$ mean, and 1 variance and $-1+\sin(t)$ mean, respectively. Data distribution 3 is the sum of them, which changes over time.

6.2. Connections with the previous results

7. Empirical studies

We conduct online logistic regression in the empirical studies. We let $f_{i,t}(\mathbf{x};\xi_{i,t}) = \log\left(1+\exp(-\mathbf{y}_{i,t}\mathbf{A}_{i,t}^{\mathrm{T}}\mathbf{x})\right) + \frac{\gamma}{2}\|\mathbf{x}\|^2$, where $\gamma = 10^{-3}$ is the given hyper-parameter. The learning rate η is set to be $\sqrt{\frac{M}{100T}}$.

7.1. Datasets

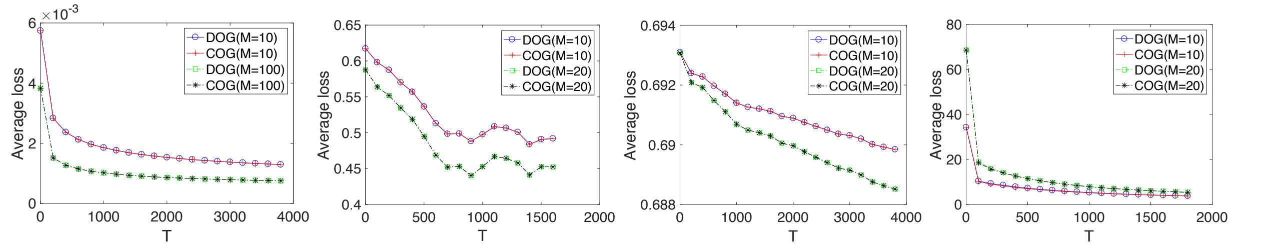
Synthetic data We generate a data matrix $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2 + \cdots + \mathbf{A}_n$, where $\mathbf{A}_i$ is placed on the i-th node, and $\mathbf{A}_i = 0.1\tilde{\mathbf{A}}_i + 0.9\hat{\mathbf{A}}_i$, where $\tilde{\mathbf{A}}_i$ represents the adversary part of data, and $\hat{\mathbf{A}}_i$ represents the stochastic part of data. $\mathbf{y}_i \in \{1,-1\}$ is the label of an instance $\hat{\mathbf{A}}_{i,t}$. The dimension of every instance is $d = 10$. Specifically, elements of $\tilde{\mathbf{A}}_i$ is sampled from the interval $[-0.5+\sin(i), 0.5+\sin(i)]$ randomly. Note that $\tilde{\mathbf{A}}_i$ and $\tilde{\mathbf{A}}_j$ with $i \neq j$ are different distributions. Besides, $\mathbf{y}_{i,t} \in \{1,-1\}$ is generated randomly. When $\mathbf{y}_{i,t} = 1$, $\hat{\mathbf{A}}_{i,t}$ is generated by sampling from a multi-variate Normal distribution $\hat{\mathbf{A}}_{i,t} \sim N((1+0.5\sin(t))\cdot\mathbf{1},\mathbf{I})$. When $\mathbf{y}_{i,t} = -1$, $\hat{\mathbf{A}}_{i,t}$ is generated by sampling from another multi-variate Normal distribution $\hat{\mathbf{A}}_{i,t} \sim N((-1+0.5\sin(t))\cdot\mathbf{1},\mathbf{I})$. As illustrated in Figure 1, the data distribution changes over time, and the optimal learning model thus change over time, where the dynamic regret is practical and necessary to measure the goodness of a learning model.

Real data We use three real time-series datasets: *room-occupancy*[1], *online-retail*[2], and *spam*[3]. The data distribution of those datasets changes over time in those practial scenarios, leading to the change of the optimal learning model

[1] https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+

[2] https://archive.ics.uci.edu/ml/datasets/Online+Retail

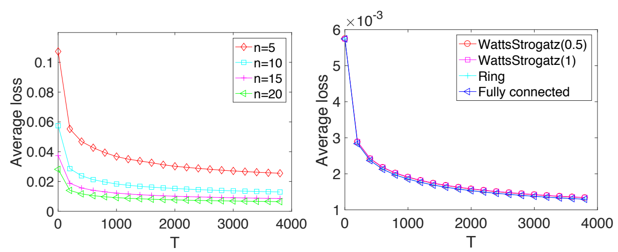
[3] http://mlkd.csd.auth.gr/concept_drift.html



(a) *synthetic data*, 100 nodes (b) *room-occupancy*, 5 nodes (c) *online-retail*, 5 nodes (d) *spam*, 5 nodes

Figure 2. The performance of DOG is comparable to that of COG.

during online learing. In those dynamic environment, the dynamic regret is practial and necessary.



(a) *Network size* (b) *Network topology*

Figure 3. The performance of DOG is comparable to that of COG.

7.2. Evaluation tasks

The average loss $\frac{1}{nT}\sum_{i=1}^{n}\sum_{t=1}^{T} f_{i,t;\xi_{i,t}}(\mathbf{x}_{i,t})$ is used as a metric. We use it to varify:

1. whether Decentralized Online Gradient method (DOG), i.e., Algorithm 1, has comparable performance with the Centralized Online Gradient method (COG).
2. whether DOG has a good performance with the increase of the network size.
3. whether DOG has a good performance with the change of the network topology.

7.3. Results

First, we simulate a decentralized network consisting of 100 nodes to handle the synthetic data, and a network consisting 5 nodes to handle the real data. Those nodes are connected by using a ring topology. As shown in Figure 3, both DOG and COG are effective to optimize the decision variable, and they have very similar performance. Comparing with COG, DOG is good enough to conduct online learning.

Second, we evaluate the performance of DOG by varying the network size. As shown in Figure 3(a), the average loss decreases effectively for a large network. After that, we change the topology of the network, and still evaluate the performance of DOG. The topology *Fully connected* means all nodes are connected, where DOG de-generates to be COG. The topology *WattsStrogatz* represents a Watts-Strogatz small-world graph. There is a parameter can be tuned, e.g., 0.5 or 1 in the legend of Figure 3(b), to control the number of random edges in the Watts-Strogatz small-world graph. As illustrated in Figures 3(b), the topology *Fully connected* has the best performance. But, there is no significant difference between all topologies.

References

D. Adamskiy, W. M. Koolen, A. Chernov, and V. Vovk. A closer look at adaptive regret. *Journal of Machine Learning Research*, 17(23):1–21, 2016.

M. Akbari, B. Gharesifard, and T. Linder. Distributed online convex optimization on time-varying directed graphs. *IEEE Transactions on Control of Network Systems*, 4(3): 417–428, Sep. 2017.

A. S. Bedi, P. Sarma, and K. Rajawat. Tracking moving agents via inexact online gradient descent algorithm. *IEEE Journal of Selected Topics in Signal Processing*, 12 (1):202–217, Feb 2018.

A. A. Benczúr, L. Kocsis, and R. Pálovics. Online Machine Learning in Big Data Streams. *CoRR*, 2018.

N. Cesa-Bianchi, P. Gaillard, G. Lugosi, and G. Stoltz. Mirror Descent Meets Fixed Share (and feels no regret). In *NIPS 2012*, page Paper 471, 2012.

K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially Private Empirical Risk Minimization. *Journal of Machine Learning Research*, 2011.

C. Dwork and A. Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

A. György and C. Szepesvári. Shifting regret, mirror descent, and matrices. In *Proceedings of the 33rd Interna-*

tional Conference on International Conference on Machine Learning - Volume 48, ICML'16, pages 2943–2951. JMLR.org, 2016.

A. György, T. Linder, and G. Lugosi. Tracking the Best of Many Experts. *Proceedings of Conference on Learning Theory (COLT)*, 2005.

A. Gyorgy, T. Linder, and G. Lugosi. Efficient tracking of large classes of experts. *IEEE Transactions on Information Theory*, 58(11):6709–6725, Nov 2012.

E. C. Hall and R. Willett. Dynamical Models and tracking regret in online convex programming. In *Proceedings of International Conference on International Conference on Machine Learning (ICML)*, 2013.

E. C. Hall and R. M. Willett. Online Convex Optimization in Dynamic Environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.

E. Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, Aug 1998.

A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan. Online Optimization : Competing with Dynamic Comparators. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 398–406, 2015.

K.-S. Jun, F. Orabona, S. Wright, and R. Willett. Improved strongly adaptive online learning using coin betting. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 943–951, 20–22 Apr 2017.

M. Kamp, M. Boley, D. Keren, A. Schuster, and I. Sharfman. Communication-efficient distributed online prediction by dynamic model synchronization. In *Proceedings of the 2014th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECMLPKDD'14, pages 623–639, Berlin, Heidelberg, 2014. Springer-Verlag.

A. Koppel, S. Paternain, C. Richard, and A. Ribeiro. Decentralized online learning with kernels. *IEEE Transactions on Signal Processing*, 66(12):3240–3255, June 2018.

S. Lee, A. Ribeiro, and M. M. Zavlanos. Distributed continuous-time online optimization using saddle-point methods. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4314–4319, Dec 2016.

S. Lee, A. Nedi, and M. Raginsky. Coordinate dual averaging for decentralized online optimization with nonseparable global objectives. *IEEE Transactions on Control of Network Systems*, 5(1):34–44, March 2018.

M. Mohri and S. Yang. Competing with automata-based expert sequences. In A. Storkey and F. Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pages 1732–1740, 09–11 Apr 2018.

A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Proceedings of IEEE Conference on Decision and Control (CDC)*, pages 7195–7201. IEEE, 2016.

J. Mourtada and O.-A. Maillard. Efficient tracking of a growing number of experts. *arXiv.org*, Aug. 2017.

A. Nedi, S. Lee, and M. Raginsky. Decentralized online optimization with global objectives and local communication. In *2015 American Control Conference (ACC)*, pages 4497–4503, July 2015.

S. Shahrampour and A. Jadbabaie. Distributed online optimization in dynamic environments using mirror descent. *IEEE Transactions on Automatic Control*, 63(3):714–725, March 2018.

S. Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

D. Wang, M. Ye, and J. Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems 30*, pages 2722–2731. 2017.

C.-Y. Wei, Y.-T. Hong, and C.-J. Lu. Tracking the best expert in non-stationary stochastic environments. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Proceedings of Advances in Neural Information Processing Systems*, pages 3972–3980, 2016.

H.-F. Xu, Q. Ling, and A. Ribeiro. Online learning over a decentralized network through admm. *Journal of the Operations Research Society of China*, 3(4):537–562, Dec 2015.

F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi. Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2483–2493, Nov 2013.

T. Yang, L. Zhang, R. Jin, and J. Yi. Tracking Slowly Moving Clairvoyant - Optimal Dynamic Regret of Online

Learning with True and Noisy Gradient. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2016.

C. Zhang, P. Zhao, S. Hao, Y. C. Soh, B. S. Lee, C. Miao, and S. C. H. Hoi. Distributed multi-task classification: a decentralized online learning approach. *Machine Learning*, 107(4):727–747, Apr 2018a.

L. Zhang, T. Yang, J. Yi, R. Jin, and Z.-H. Zhou. Improved Dynamic Regret for Non-degenerate Functions. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2017a.

L. Zhang, T. Yang, rong jin, and Z.-H. Zhou. Dynamic regret of strongly adaptive methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 5882–5891, 10–15 Jul 2018b.

W. Zhang, P. Zhao, W. Zhu, S. C. H. Hoi, and T. Zhang. Projection-free distributed online learning in networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, pages 4054–4062, International Convention Centre, Sydney, Australia, 06–11 Aug 2017b.

Y. Zhao, S. Qiu, and J. Liu. Proximal Online Gradient is Optimum for Dynamic Regret. *CoRR*, cs.LG, 2018.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 928–935, 2003.