Online Variance Reduction with Mixtures

Zalán Borsos [1] **Sebastian Curi** [1] **Kfir Y. Levy** [1] **Andreas Krause** [1]

Abstract

Adaptive importance sampling for stochastic optimization is a promising approach that offers improved convergence through variance reduction. In this work, we propose a new framework for variance reduction that enables the use of mixtures over predefined sampling distributions, which can naturally encode prior knowledge about the data. While these sampling distributions are fixed, the mixture weights are adapted during the optimization process. We propose VRM, a novel and efficient adaptive scheme that asymptotically recovers the best mixture weights in hindsight and can also accommodate sampling distributions over sets of points. We empirically demonstrate the versatility of VRM in a range of applications.

1. Introduction

In the framework of Empirical Risk Minimization (ERM), we are provided with a set of samples $\mathcal{D} = \{x_1, \dots, x_n\} \subset \mathcal{X}$ drawn from the underlying data distribution, and our goal is to minimize the average loss over $\mathcal{D}$, known as the empirical risk. Sequential ERM solvers (SGD, SVRG, etc.) proceed in multiple passes over the dataset and usually require an unbiased estimate of the loss in each round of the optimization. Typically, the estimate is generated by sampling uniformly from $\mathcal{D}$, which is oblivious to the fact that different points can affect the optimization differently. This ignorance can hinder the performance of the optimizer due to the high variance of the obtained estimates.

A promising direction that has recently received increased interest is represented by (adaptive) importance sampling techniques. Clever sampling distributions can account for characteristics of datapoints relevant to the optimization in order to improve the performance (Zhao & Zhang, 2015; Namkoong et al., 2017; Katharopoulos & Fleuret, 2018).

[1]Department of Computer Science, ETH Zurich. Correspondence to: Zalán Borsos <zalan.borsos@inf.ethz.ch>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

Why mixtures? The majority of existing works on adaptive sampling distributions are *unable to exploit similarities* between points. Thus, only after *several passes over the dataset* do these methods become effective. This can become a major bottleneck for large datasets.

Fortunately, in many situations, it is possible to exploit the structure present in data in some form of a prior. One way of capturing prior knowledge in the framework of importance sampling for variance reduction is to propose plausible fixed sampling distributions before performing the optimization. This is very natural for problems where *similar objects* can be *grouped together*, e.g., based on the class label or clustering in feature space. In such cases it is sensible to employ standard sampling distributions that draw similar members with the same probability, e.g., as considered by Zhao & Zhang (2014). Another option is to employ sampling distributions that encourage diverse sets of samples, e.g., Determinantal Point Processes (Kulesza et al., 2012).

Suppose that several such proposals for sampling distributions are available. A natural idea is to combine them into a mixture and *adapt the mixture weights* during the optimization process, in order to achieve variance reduction. This setup has the potential of being much more efficient than learning an arbitrary sampling distribution over individual points, provided that one can propose plausible sampling distributions prior to the optimization. Another advantage of this setting is that it enables *efficient* handling of distributions not only on individual points, but also on *sets of points*. While in principle this can still be treated using existing approaches, their computational complexity in this case will increase proportionally to the number of sets, possibly *exponentially* with the number of points.

In this work, we develop an online learning approach to variance reduction and ask the following question: given k fixed sampling distributions, how can we choose the mixture weights in order to achieve the largest reduction in the variance of the estimates? We provide a simple yet efficient algorithm for doing so. As for our main contributions, we:

- formulate the task of adaptive importance sampling for variance reduction with mixtures as an online learning problem,
- propose a novel algorithm for this setting with sublinear regret of $\tilde{\mathcal{O}}(T^{4/5})$,

- substantiate our findings experimentally with accompanying efficient implementation[1].

Related Work: There is a large body of work on employing importance sampling distributions for variance reduction. Prior knowledge of gradient norm bounds on each datapoint has been utilized for fixed importance sampling by Needell et al. (2014) and Zhao & Zhang (2015). Adaptive strategies were presented by Bouchard et al. (2015), who propose parametric importance sampling distributions where the parameters of the distributions are updated during the course of the optimization. Stich et al. (2017) derive a safe adaptive sampling scheme that is guaranteed to outperform any fixed sampling strategy. A significant body of work is concerned with non-uniform sampling of coordinates in coordinate descent (Allen-Zhu et al., 2016; Perekrestenko et al., 2017; Salehi et al., 2018). All the works presented above provide importance sampling schemes over points or coordinates. Sampling over sets of points and exploiting similarities between points in these works remains an open question.

Importance sampling has found applications in optimizing deep neural networks. Johnson & Guestrin (2018) and Katharopoulos & Fleuret (2018) develop methods for choosing the importance sampling distributions over points proportional to their corresponding approximate gradient norm bounds. Johnson & Guestrin (2018) also propose to adapt the learning rate based on the gains in gradient norm reductions. Loshchilov & Hutter (2015) perform sampling based on the latest known loss value with exponentially decaying selection probability on the rank. In the context of reinforcement learning, Schaul et al. (2016) suggest a smoothed importance sampling scheme of experiences present in the replay buffer, based on the last observed TD-error.

Most closely related to our setting, importance sampling for variance reduction has been considered through the lens of online learning in the recent works of Namkoong et al. (2017), Salehi et al. (2017) and Borsos et al. (2018). These works pose the ERM solver as an *adversary* responsible for generating the losses. The goal of the learner (player) is to minimize the cumulative variance by choosing the sampling distributions adaptively based on partial feedback. However, similarly to existing work on adaptive sampling, these methods are not designed for exploiting similarities between points.

2. Problem Setup

In the framework of ERM, the goal is to minimize the empirical risk,

$$\min_{\theta\in\Theta}\mathcal{L}(\theta)=\min_{\theta\in\Theta}\frac{1}{n}\sum_{i=1}^{n}\ell(x_i,\theta),$$

where $\ell:\mathcal{X}\times\Theta\mapsto\mathbb{R}_{\geq 0}$ is the loss function, and $\Theta\subseteq\mathbb{R}^d$ is a compact domain. A typical sequential ERM solver will run over T rounds and update the parameters based on an *unbiased estimate* $\tilde{\mathcal{L}}_t$ of the empirical loss in each round $t\in[T]$. A common approach for producing $\tilde{\mathcal{L}}_t$ is to sample a point $i_t\in\{1,...,n\}$ uniformly at random, thus ignoring the underlying structure of the data. However, using importance sampling, we can produce these estimates by sampling with *any* distribution $q\in\Delta_n$, where Δ_n is the n-dimensional probability simplex, provided that we compensate for the bias through importance weights.

Suppose we are provided with k sampling distributions $p_1,...,p_k\in\Delta_n$. We combine these distributions into a *mixture*, in which the probability of sampling x_i is given by $w^\intercal p(i)$, where $w\in\Delta_k$ is the mixture weight vector and $p(i):=[p_1(i),...,p_k(i)]$. Using the mixture, we obtain the loss estimate

$$\tilde{\mathcal{L}}_t(\theta)=r_i\cdot\ell(x_i,\theta),$$

where $r_i=\frac{1}{n\cdot w^\intercal p(i)}$ is the importance weight of point i.

The performance of solvers such as SGD, SAGA (Defazio et al., 2014) and SVRG (Johnson & Zhang, 2013) is known to improve when the variance of $\tilde{\mathcal{L}}_t$ is smaller. Thus, a natural performance measure for our mixture sampling distribution is the *cumulative variance* of $\tilde{L}_t$ through the T rounds of optimization,

$$\sum_{t=1}^{T}\mathrm{Var}(\tilde{\mathcal{L}}(\theta_t))=\frac{1}{n^2}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{\ell^2(x_i,\theta_t)}{w^\intercal p(i)}-\sum_{t=1}^{T}\mathcal{L}^2(\theta_t).$$

Since only the cumulative second moments depend on w, we define our cost function at time t as $\frac{1}{n^2}f_t(w)$, where

$$f_t(w)=\sum_{i=1}^{n}\frac{\ell_t^2(i)}{w^\intercal p(i)},$$

and where we have introduced the shorthand $\ell_t(i):=\ell(x_i,\theta_t)$. Through the lens of online learning, it is natural to regard the sequential solver as an *adversary* responsible for generating the losses $\{\ell_t\}_{t\in[T]}$ and to measure the performance using the notion of the *cumulative regret*,

$$\mathrm{Regret}_T=\frac{1}{n^2}\left(\sum_{t=1}^{T}f_t(w_t)-\min_{w\in\Delta_k}\sum_{t=1}^{T}f_t(w)\right).$$

By devising a no-regret algorithm, we are guaranteed to compete asymptotically with the best mixture weights in hindsight. The protocol for online variance reduction with mixtures (OVRM) is presented in Figure 1.

Motivated by empirical insights, we impose a natural mild restriction on our setting, which is easily verified in practice:

[1]The code is available at https://github.com/zalanborsos/variance-reduction-mixtures

Assumption 1 Throughout the work, we assume that the losses are bounded, $\ell_t^2(i) \leq L$ for all $t \in [T]$, $i \in [n]$ and that all mixture components place a probability mass at most $p_{\max} = \frac{c}{n}$ on any specific point, where $c \in [1, n]$. That is, $p_j(i) \leq \frac{c}{n}$, for all $i \in [n], j \in [k]$.

The choice of c is in the hands of the designer who determines the fixed sampling distributions in the mixture. Although c can be as large as n, in our experiments, we show how we can obtain a large speedup in the optimization due to the reduced variance by using mixtures with small values of c (the maximal value of c is less than 50 in the experiments). We note that our setup also allows choosing $k = n$, as many mixture components as points, where a mixture puts all its probability mass on a specific point, i.e., $p_j(i) = \delta_{ij}$ for $i \in [n], j \in [k]$ and consequently $c = n$. Under this perspective, our setting is a *strict generalization* of adaptively choosing sampling distributions over the points, which is the main objective of Salehi et al. (2017), Namkoong et al. (2017) and Borsos et al. (2018). However, in practical scenarios, k is usually small due to the limited number of available proposal distributions.

OVRM Protocol

Input: Dataset $\mathcal{D} = \{x_1, ..., x_n\}$,
sampling distributions $p = [p_1, ..., p_k]$.
for $t = 1, \ldots, T$ **do**
  player chooses $w_t \in \Delta_k$
  adversary chooses $\ell_t \in \mathbb{R}^n$
  player draws $I_t \sim w_t^\intercal p$
  player incurs cost $f_t(w_t)/n^2$ and receives $\ell_t(I_t)$ as partial feedback
end for

Figure 1. Online variance reduction protocol with mixtures and partial feedback.

We have formulated our objective as minimizing the cumulative second moment of the loss estimates. If we choose to substitute $\ell_t(i)$ with $\|\nabla \ell(x_i, \theta_t)\|$, the norm of the loss gradients, the corresponding cumulative second moment has a stark relationship to the quality of optimization — for example, this quantity directly appears in the regret bounds of AdaGrad (Duchi et al., 2011). For a more detailed discussion see Borsos et al. (2018).

Let us discuss some properties of our setting. Since $f_1, ..., f_T$ are convex functions on Δ_k, the problem is an instance of online convex optimization (OCO). While the OCO framework offers a wide range of well-understood tools, our biggest challenge here is posed by the fact that the cost functions are *unbounded*, together with the fact that we have *partial feedback*. The majority of existing regret analyses assume boundedness of the cost functions.

For simplicity, we focus on choosing *datapoints*, nevertheless, our method applies to choosing coordinates or blocks of *coordinates* in coordinate descent and can work on top of any sequential solver that builds on unbiased loss estimates. As we will see in Section 4, the complexity and the performance guarantee of our algorithm is *independent of* n, which broadens its applicability significantly. For example, instead of learning mixtures of distributions over points, we can learn a mixture for variance-reduced sampling of *minibatches*, where each mixture component is a fixed k-Determinantal Point Process (Kulesza et al., 2012).

3. Full Information Setting

Let us assume for the moment that in each round of Protocol 1, the player receives full information feedback, i.e., sees the losses associated to *all* points $[\ell_t(1), ..., \ell_t(n)]$ instead of observing only the loss $\ell_t(I_t)$ associated with the chosen point. This setup, referred to as full information setting, is unrealistic, yet it serves as the main tool for the analysis of the partial information setting, which we discuss in Section 4. Here, we first show an efficient algorithm for the full information setting (Alg. 1), ensuring a regret bound of $\tilde{\mathcal{O}}(k^{1/2}T^{2/3})$.

Unfortunately, even under Assumption 1, our cost function is unbounded. In order to tackle this, we consider that the last mixture component (the k-th one) is always the uniform distribution. If this is not the case in practice, we can simply attach the uniform distribution to the given distributions. This is w.l.o.g., since the optimal w in hindsight is allowed to assign 0 weight on any component. Thus, we have that $p(i) = [p_1(i), ..., p_{k-1}(i), 1/n]$ for all $i \in [n]$. For the analysis, we consider the restricted simplex $\Delta_k' = \{w \in \Delta_k | \, w(k) \geq \gamma\}$, where the last weight corresponding to the uniform component is larger than some $\gamma \in (0, 1]$. This allows for decomposing the regret as follows:

$$\text{Regret}_T = \underbrace{\frac{1}{n^2}\left(\sum_{t=1}^T f_t(w_t) - \min_{w \in \Delta_k'} \sum_{t=1}^T f_t(w)\right)}_{(A)} + \underbrace{\frac{1}{n^2}\left(\min_{w \in \Delta_k'} \sum_{t=1}^T f_t(w) - \min_{w \in \Delta_k} \sum_{t=1}^T f_t(w)\right)}_{(B)}. \quad (1)$$

This decomposition introduces a trade-off. By choosing larger γ, we pay more in term (B) for potentially missing the optimal w. Nevertheless, larger γ makes the cost function "nicer": not only does it reduce the upper bounds on the costs, but it also turns f_t into an *exp-concave* function, as we will later show.

First, let us focus on bounding (B), which captures the excess regret of working in Δ_k' instead of Δ_k.

Lemma 1. *The reduction to the restricted simplex Δ'_k incurs the excess regret of*

$$(B) \leq \gamma LT.$$

Proof. Let $w_* = \arg\min_{w\in\Delta_k} \sum_{t=1}^T f_t(w)$. Let $w'_* = (1-\gamma)w_* + \gamma e_k$, where $e_k = [0, ..., 0, 1]$. Now clearly $w'_* \in \Delta'_k$. We can observe that for all $i \in [n]$,

$$\frac{1}{w_*'^\intercal p(i)} - \frac{1}{w_*^\intercal p(i)} = \frac{\gamma(w_* - e_k)^\intercal p(i)}{w_*'^\intercal p(i)\cdot w_*^\intercal p(i)} = \frac{\gamma\left(w_*^\intercal p(i) - \frac{1}{n}\right)}{w_*'^\intercal p(i)\cdot w_*^\intercal p(i)}.$$

If for some i we have $w_*^\intercal p(i) - 1/n < 0$, or, equivalently $w_*^\intercal p(i) < 1/n$, we can ignore this specific term. Otherwise, if $w_*^\intercal p(i) \geq 1/n$, then also evidently $w_*'^\intercal p(i) \geq 1/n$. Denote $\mathbb{I}_+$ the set of i's for which $w_*^\intercal p(i) \geq 1/n$. Using the previous observations, we can now bound (B):

$$\begin{aligned} n^2\cdot(B) &\leq \sum_{t=1}^T \sum_{i\in\mathbb{I}_+} \frac{\gamma\ell_t^2(i)\left(w_*^\intercal p(i) - \frac{1}{n}\right)}{w_*'^\intercal p(i)\cdot w_*^\intercal p(i)} \\ &\leq \gamma L \sum_{t=1}^T \sum_{i\in\mathbb{I}_+} \frac{w_*^\intercal p(i)}{w_*'^\intercal p(i)\cdot w_*^\intercal p(i)} \leq n^2\gamma LT, \end{aligned}$$

where the last inequality uses the fact that $w_*'^\intercal p(i) \geq 1/n$ for all $i \in \mathbb{I}_+$ and that $|\mathbb{I}_+| \leq n$. This proves the claim. □

By constraining our convex set to the restricted simplex Δ'_k, we achieve desirable properties of our cost function: f_t and its gradient norm are bounded. The first natural option for solving the problem is Online Gradient Descent (OGD). However, OGD can only guarantee a $\mathcal{O}(\sqrt{T})$ bound on (A); we elaborate on this in the supplementary material. We can obtain better regret bounds by noticing that restricting the domain to Δ'_k has another advantage: it allows for exploiting curvature information as f_t is *exp-concave* on this domain.

A convex function $g : \mathcal{K} \mapsto \mathbb{R}$, where $\mathcal{K}$ is a convex set, is called α-exp-concave, if $e^{-\alpha g(x)}$ is concave. Exp-concavity is a weaker property than strong convexity, but it can still be exploited to achieve logarithmic regret bounds (Hazan et al., 2006). In the following result, we establish the exp-concavity of our cost function on the restricted simplex Δ'_k.

Lemma 2. *f_t is $\frac{2\gamma}{n^2L}$-exp-concave on Δ'_k for all $t \in [T]$.*

Proof sketch. In order to prove exp-concavity, we rely on the following result (Hazan et al., 2016): a twice differentiable function g is α-exp-concave at x, iff

$$\nabla^2 g(x) \succeq \alpha \nabla g(x) \nabla^\intercal g(x). \tag{2}$$

In our case, $\mathcal{K} = \Delta'_k$ and $\nabla f_t(w) = -\sum_{i=1}^n \frac{\ell_t^2(i)p(i)}{(w^\intercal p(i))^2}$ and $\nabla^2 f_t(w) = 2\sum_{i=1}^n \frac{\ell_t^2(i)p(i)p(i)^\intercal}{(w^\intercal p(i))^3}$. We can prove the property of exp-concavity using the following observation: for $x_1, ..., x_n \in \mathbb{R}^d$ we have

$$\left(\sum_{i=1}^n x_i\right)\left(\sum_{i=1}^n x_i\right)^\intercal \preceq n\sum_{i=1}^n x_i x_i^\intercal, \tag{3}$$

which is a result of the definition of positive semi-definiteness and Jensen's inequality. If we instantiate $x_i := -\frac{\ell_t^2(i)p(i)}{(w^\intercal p(i))^2}$ and plug f_t into Equation 2, we can identify $\alpha = 2\gamma/(n^2L)$ after an additional step of lower bounding $w^\intercal p(i)$ by γ/n. From the last step, we can see that working in the restricted simplex Δ'_k is crucial for achieving exp-concavity. □

Algorithm 1 ONS

input Dataset $\mathcal{D} = \{x_1, ..., x_n\}$, sampling distributions $p = [p_1, ..., p_{k-1}, 1/n]$, parameters $\gamma, \beta, \varepsilon \geq 0$.
$w_1 = [1/k, ..., 1/k]$
$H_0 = \varepsilon\mathbb{I}$
for t in 1 to T **do**
  play w_t, observe $f_t(w_t)$
  update: $H_t = H_{t-1} + \nabla f_t(w_t)\nabla^\intercal f_t(w_t)$
  Newton step: $w' = w_t - \frac{1}{\beta}H_t^{-1}\nabla f_t(w_t)$
  project: $w_{t+1} = \arg\min_{w\in\Delta'_k}(w-w')^\intercal H_t(w-w')$
end for

Since the f_t's are α-exp-concave functions in the restricted simplex, we can bound (A) by employing Algorithm 1, known as Online Newton Step (ONS), which provides the following guarantee for appropriately chosen β and ε (Hazan et al., 2006):

$$\sum_{t=1}^T f_t(w_t) - \min_{w\in\Delta'_k}\sum_{t=1}^T f_t(w) \leq 5\left(\frac{1}{\alpha} + GD\right)k\log T, \tag{4}$$

where $D = \sqrt{2}$ is the diameter of Δ'_k and $G \geq \sup_{w\in\Delta'_k, t\in[T]} \|\nabla f_t(w)\|_2$ is an upper bound on the gradient norm:

$$\begin{aligned} \|\nabla f_t(w)\|_2 &= \left\|\sum_{i=1}^n \frac{\ell_t^2(i)p(i)}{(w^\intercal p(i))^2}\right\|_2 \leq \frac{n^2L}{\gamma^2}\left\|\sum_{i=1}^n p(i)\right\|_2 \\ &= \frac{n^2L}{\gamma^2}\|(1, ..., 1)\|_2 = \frac{n^2L\sqrt{k}}{\gamma^2} =: G, \end{aligned}$$

where the inequality uses that in Δ'_k we have $w^\intercal p(i) \geq \gamma/n$. Using these bounds together with Lemma 2 in Equation 4, we can finally bound (A):

Lemma 3. *Algorithm 1 ensures*

$$(A) \leq \frac{10Lk^{3/2}\log T}{\gamma^2}.$$

Finally, we can combine the results from Lemma 3 and 1 and optimize the parameter γ that controls the trade-off, to arrive at the following regret bound with respect to the full simplex:

Theorem 4. *The regret of Algorithm 1 is*

$$\mathit{Regret}_T \leq 5Lk^{1/2}T^{2/3}\log^{1/3} T.$$

4. The Partial Information Setting

In a practice, the player only receives partial feedback from the environment corresponding to the loss of the chosen point, as presented in Figure 1. Even under partial feedback, the unbiasedness of the loss estimates must be ensured. For this, we propose our main algorithm, *Variance Reduction with Mixtures (VRM)*, presented in Algorithm 2. VRM is inspired by the seminal work of Auer et al. (2002), in its approach to obtaining unbiased estimates under partial information. The algorithm in line 4 samples $I_t \sim w_t^\intercal p$ and receives only $\ell_t(I_t)$ as feedback in round t. We obtain an estimate by

$$\tilde{\ell}_t^2(i) = \frac{\ell_t^2(i)}{w_t^\intercal p(i)} \cdot \mathbb{1}_{I_t=i}, \tag{5}$$

which is clearly unbiased due to $\mathbb{E}\left[\tilde{\ell}_t^2(i)|\ell_t, w_t\right] = \ell_t^2(i)$. We can analogously define $\tilde{f}_t(w) = \sum_{i=1}^n \frac{\tilde{\ell}_t^2(i)}{w^\intercal p(i)}$. With this choice, the estimates can be readily used, similar to the full information setting, in Algorithm 2.

Algorithm 2 VRM

input Dataset $\mathcal{D} = \{x_1, ..., x_n\}$, sampling distributions $p = [p_1, ..., p_{k-1}, 1/n]$, parameters $\gamma, \beta, \varepsilon \geq 0$.
1: $w_1 = [1/k, ..., 1/k]$
2: $H_0 = \varepsilon\mathbb{I}$
3: **for** t in 1 to T **do**
4: sample $I_t \sim w_t^\intercal p$, receive $\ell_t(I_t)$, set $\tilde{f}_t(w) = \frac{\tilde{\ell}_t^2(I_t)}{w^\intercal p(I_t)}$
5: update: $H_t = H_{t-1} + \nabla\tilde{f}_t(w_t)\nabla^\intercal\tilde{f}_t(w_t)$
6: Newton step: $w' = w_t - \frac{1}{\beta}H_t^{-1}\nabla\tilde{f}_t(w_t)$
7: project: $w_{t+1} = \arg\min_{w\in\Delta'_k}(w-w')^\intercal H_t(w-w')$
8: **end for**

In the partial information setting, the natural performance measure of the player is the *expected regret* $\mathbb{E}\left[\mathrm{Regret}_T\right]$, where the expectation is taken with respect to the randomized choices of the player and actions of the adversary. Crucially, we allow the adversary to adapt to the player's past behavior. This non-oblivious setting naturally arises in stochastic optimization, where ℓ_t depends on w_{t-1}. For analyzing the expected regret incurred by the VRM under partial information, we can reuse the full information analysis. However, the exp-concavity constant and the gradient norm bounds change, and the non-oblivious behavior requires further analysis, resulting in the no-regret guarantee of Theorem 5, which is *independent* of n.

Theorem 5. *VRM achieves the expected regret*

$$\mathbb{E}\left[\mathit{Regret}_T\right] = \tilde{\mathcal{O}}\left(k^{3/8}c^{1/5}LT^{4/5}\right).$$

Proof sketch. We first start by bounding the pseudo-regret, which compares the cost incurred by VRM to the cost incurred by the optimal mixture weights in expectation. It can be shown that $\tilde{f}_t(w)$ is $\frac{2\gamma^2}{n^2L}$-exp concave on Δ'_k and has the gradient bound

$$\left\|\nabla\tilde{f}_t(w)\right\|_2 = \frac{\tilde{\ell}_t^2(I_t)\left\|p(I_t)\right\|_2}{(w^\intercal p(I_t))^2} \leq \frac{Ln^2c\sqrt{k}}{\gamma^3},$$

where the inequality uses the fact that $w^\intercal p(I_t) \geq \gamma/n$ and Assumption 1, which implies $\|p(i)\|_2 \leq c\sqrt{k}/n$ for all $i \in [n]$. Combined with the guarantee in Equation 4, this gives the bound on the expectation of (A) from the regret decomposition. The upper bound on (B) from Lemma 1 does not change under expectation and the modified losses. For bounding the expected regret, we rely on Freedman's lemma (Freedman, 1975) for the martingale difference sequence $\{Z_t := \sum_{i=1}^n \tilde{\ell}_t^2(i) - \sum_{i=1}^n \ell_t^2(i)\}_{t\in[T]}$ in order to account for the non-oblivious nature of the adversary. □

5. Efficient Implementation

We now address practical aspects of VRM. Naively implemented, each iteration of the algorithm has a complexity of $\mathcal{O}(k^3)$. One might argue that this can become a bottleneck when performed in each round of stochastic optimization. In practice, however, one usually has a limited number of available proposal distributions, limiting k to the small regime. Moreover, in the following, we present several tricks that improve on the complexity of the iteration.

The online Newton update and step in lines 5 and 6 of Algorithm 2 can be implemented in $\mathcal{O}(k^2)$ due to the Sherman-Morrison formula (Hazan et al., 2006):

$$H_t^{-1} = H_{t-1}^{-1} - \frac{H_{t-1}^{-1}\nabla\tilde{f}_t(w_t)\nabla\tilde{f}_t(w_t)^\intercal H_{t-1}^{-1}}{1+\nabla\tilde{f}_t(w_t)^\intercal H_{t-1}^{-1}\nabla\tilde{f}_t(w_t)}.$$

Thus, the most costly operation of the algorithm is the projection step that requires solving a quadratic program, having a complexity of $\mathcal{O}(k^3)$. In practice, we can trade off accuracy for efficiency in solving the quadratic program approximately by employing only a few steps of a projection-based iterative solver (e.g., projected gradient descent, etc.). The key to the success of such a proposal is an efficient projection step onto the restricted simplex Δ'_k, which captures the constraints of the quadratic program. Our proposed method, Algorithm 3, is a two-stage projection procedure

that is inspired by the efficient projection onto the simplex (Gafni & Bertsekas, 1984; Duchi et al., 2008) and has $\mathcal{O}(k \log k)$ time complexity due to the sorting.

Algorithm 3 Projection

input w, γ
function proj_simplex $(w \in \mathbb{R}^d, z \in (0, 1])$
  sort w decreasingly into u
  $\rho = \max \left\{ j \in [d] : u_j - \left(\sum_{\tau=1}^{j} u_\tau - z \right) / j > 0 \right\}$
  $\lambda = \left(\sum_{\tau=1}^{\rho} u_\tau - z \right) / \rho$
  return $\max\{w - \lambda, 0\}$
end function

$w =$ proj_simplex $(w, 1)$
if $w(k) < \gamma$ **then**
  $w(k) = \gamma$
  $w(1 : k-1) =$ proj_simplex $(w(1 : k-1), 1 - \gamma)$
end if
return w

The idea behind projecting to Δ'_k is the following: if the projection step with respect to the full simplex results in a point in the restricted simplex, we are done. Otherwise, we set the last coordinate of w to γ, and project the first $k-1$ coordinates to have mass $1 - \gamma$.

Lemma 6. *Algorithm 3 returns*

$$x = \arg\min_{x'} \|x' - w\|_2^2 \quad \text{s.t. } x \in \Delta'_k.$$

Proof. As shown by Duchi et al. (2008), the *proj_simplex* function solves the following minimization problem:

$$\min_x \|x - w\|_2^2 \quad \text{s.t. } \sum_{i=1}^{d} x_i = z,\ x_i \geq 0.$$

Denoting $x_* = \arg\min_{x \in \Delta_k} \|x - w\|_2$ and $x'_* = \arg\min_{x \in \Delta'_k} \|x - w\|_2$, we only need to inspect the case when $x_* \neq x'_*$. In this case, we have $x'_*(k) = \gamma$. To see this by proof of contradiction, assume $x'_*(k) > \gamma$. Now we have[2] $\|x_* - w\| < \|x'_* - w\|$, and there also exists a small ϵ such that $y := (1 - \epsilon)x'_* + \epsilon x_* \in \Delta'_k$ and $y(k) = \gamma$. The contradicts with the optimality of x'_* since,

$$\|y - w\|_2 \leq (1-\epsilon) \|x'_* - w\|_2 + \epsilon \|x_* - w\|_2 < \|x'_* - w\|_2 .$$

As a consequence, if $x_* \neq x'_*$ we can set $w(k) = \gamma$ and call the *proj_simplex* function for the first $k-1$ coordinates and with the $1 - \gamma$ leftover mass. □

Thus we have reduced the cost of one iteration in VRM to $\mathcal{O}(k^2)$, and we further investigate its efficiency in the experiments.

[2] This is since the projection objective $\|x - w\|^2$ is strongly-convex, and hence the optimum must be unique.

6. Experiments

In this section, we evaluate our method experimentally. The experiments are designed to illustrate the underlying principles of the algorithm as well as the beneficial effects of variance reduction in various real-world domains. We emphasize that it is crucial to design good sampling distributions for the mixture, and that this is an application-specific task. The following experiments provide guidance to this design process, but deriving better sampling distributions is an open question for future work.

6.1. SVM on blobs

Consider the toy dataset consisting of $n = 10\,000$ datapoints arranged in 6 well-separated, balanced, Gaussian blobs illustrated in the left of Figure 2. Points belonging to the leftmost three blobs are assigned negative class labels, and points in the rightmost three are labelled as positive.

In this setting it is natural to propose $k = 6$ sampling distributions, one corresponding to each blob. A specific component assigns uniformly large probability to its associated points and uniformly small probability everywhere else. Notice that in this case $c = k$. We run 5 epochs of online gradient descent for SVM with step size $0.01/\sqrt{t}$ at iteration t. At each iteration, the sampler gets as feedback the norm of the gradient of the hinge loss. This way, VRM is expected to propose critical points (producing high norm loss gradients) more frequently, i.e., to sample the two middle blobs often, since they contain the support vectors. This intuition is confirmed in the middle plot of Figure 2, where the points' color intensities represent their corresponding blob's mixture weights obtained by VRM at the end of the training. This also results in the fact that VRM achieves a certain level of accuracy faster than uniform sampling, due to discovering the support vectors earlier.

6.2. k-DPPs

The following experiment illustrates that our method can handle distributions over sets of points. k-Determinantal point processes (k-DPP) (Kulesza et al., 2012) over a discrete set is a distribution over all subsets of size k. Being a member of the family of repulsive point processes, their diversity-inducing effect has recently been used in Zhang et al. (2017) for sampling minibatches in stochastic optimization. In this experiment, we take a similar path and investigate variance reduction in linear regression with sampling batches from a mixture of k-DPP kernels. This is rendered possible by our theoretical results, which show that the regret is independent of the number of points (which is $\binom{n}{k}$ in this case).

We solve linear regression on a synthetic dataset of size $n = 1\,000$ and dimension $d = 10$ generated as follows: the

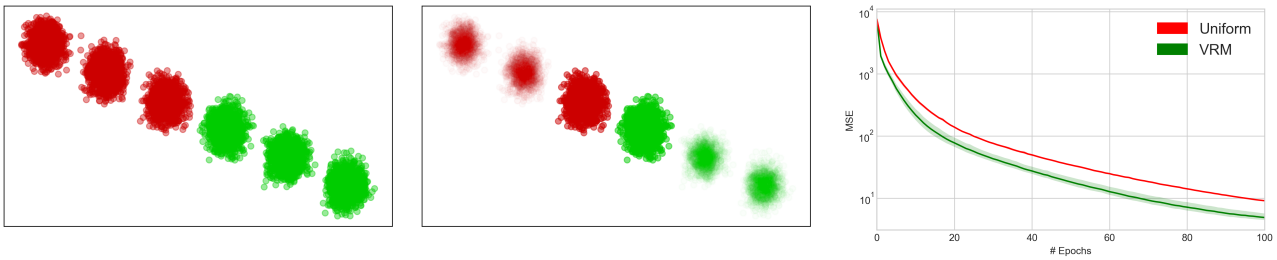

Figure 2. Left: toy dataset consisting of 6 blobs, green indicates positive labels. Middle: illustration of mixture weights after 10 000 iterations, where high transparency corresponds to low weight. Due to large mixture weights, points from the two middle blobs are sampled more often, leading to faster discovery of support vectors. Right: Mean squared error achieved by the samplers on the regression task. VRM with k-DPPs provides $1.4\times$ speedup over uniform sampling in terms of iterations.

features are drawn from a multivariate normal distribution with random means and variances for each dimension. In order to change the relative importance of the samples, the features of 10 randomly selected points are scaled up by a factor of 10. The dependent variables Y are generated by $Y = Xw_0 + \epsilon$, where X is the feature matrix, w_0 is a vector drawn from a normal distribution with mean 0 and variance 25 and ϵ is the standard normal noise. The optimization is performed with minibatch SGD with step size $10^{-4}/\sqrt{t}$ in round t over 100 epochs and batch size of 5. The feedback to the samplers is norm of the gradient of the mean squared error.

Our mixture consists of three k-DPPs with regularized linear kernel $L = XX^\intercal + \lambda\mathbb{I}$, where $\lambda \in \{1, 10, 100\}$. We introduce a small bias by applying soft truncation to the importance weights: $r' = 0.8r + 0.2$. The result of the 10 runs of the optimization process with different random seeds shown in right of Figure 2, where VRM significantly outperforms the uniform sampling in terms of number of iterations needed for a certain error level. However, since we use exact k-DPP sampling, the computational overhead outweighs the practical benefits of our method in this setting[3].

6.3. Prioritized Experience Replay

In this experiment, our goal is to identify good hyperparameters for prioritized experience replay (Schaul et al., 2016) with Deep Q-Learning (DQN) (Mnih et al., 2015) on the Cartpole environment of the Gym (Brockman et al., 2016). Prioritized experience replay is an importance sampling scheme that samples observations from the replay buffer approximately proportional to their last associated temporal difference (TD) error. The sampling distribution over a point j in the buffer is $p(j) \propto (|\delta_j| + \epsilon)^\alpha$, where δ_j is the last observed TD-error associated to experience j, whereas ϵ and α are hyperparameters for smoothing the probabilities. With the appropriately chosen hyperparameters, prioritized experience replay can significantly improve the performance of DQN learning.

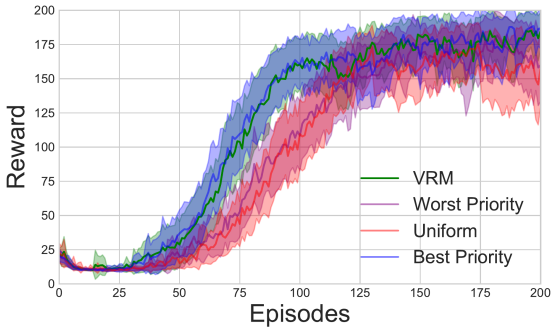

Figure 4. Evolution of rewards over 200 episodes of the different experience replay samplers on Cartpole. 50 runs with different random seeds. VRM identifies the mixture component corresponding to the best hyperparameter setting in early stages and assigns a large mixture weight to it.

In this experiment, we show how VRM allows for automatic hyperparameter selection in a single run without performance loss. We generate 9 mixture components of prioritized experience replays with all possible parameter combinations of $\epsilon = \{0.01, 0.1, 1\}$ and $\alpha = \{0.1, 0.5, 0.9\}$. The feedback to VRM is the TD-error incurred by the sampled experiences. During the optimization process, the prioritized replay buffers are also updated as new observations are inserted and the TD-errors are refreshed. This is a deviation from our presentation, where we relied on fixed sampling distributions. However, our framework easily extends to sampling distributions that *change* over time, i.e., sampling point i in round t is $i \sim w_t^\intercal p_t(i)$ and we allow p_t to depend on t. The result of 50 runs with different random seeds over 200 episodes is presented in Figure 4. The shaded areas represent 68% confidence intervals. VRM successfully identifies the mixture component corresponding to the best hyperparameter setting in early stages and assigns the largest mixture weight to it. As a consequence, VRM performs hyperparameter selection in a single run without loss of performance compared to the best setting.

[3]Efficient k-DPP samplers are available, e.g. (Li et al., 2016); we leave the investigation of time-performance trade-offs with these samplers for future work.

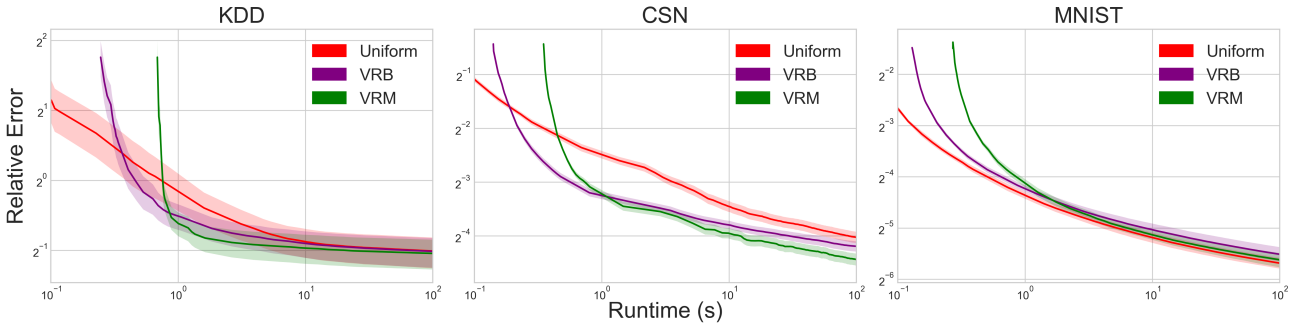

Figure 3. k-means loss evolution on the test set. VRM suffers from a larger setup time due the cost of initializing the mixture components, but eventually outperforms the other methods in terms of relative error, where the reference is the batch k-means.

6.4. k-means

Next we investigate the gains of our sampler for minibatch k-means (Sculley, 2010). We reproduce the experimental setup of Borsos et al. (2018) as well as their proposed algorithm, Variance Reducer Bandit (VRB). We compare VRM to uniform sampling and to VRB. The parameters of VRB where chosen as indicated by the authors. For both VRM and VRB, the points in the batch are sampled *independently* according to the samplers and the feedback is given in a delayed fashion, once per batch. The feedback corresponds to the norm of the minibatch k-means loss gradient.

It remains to specify how to construct our mixture sampler. We use a mixture with 10 components. Inspired by VRB, we choose each mixture's sampling distribution proportional to the square root of the distances to a randomly chosen center with small uniform smoothing. More formally, for each component j, we define its sampling distribution as

$$p_j(i) = \frac{0.9 \cdot \sqrt{d^2(x_i, \mu_j)}}{\sqrt{\sum_{k=1}^{n} d^2(x_k, \mu_j)}} + \frac{0.1}{n},$$

where μ_j is the randomly chosen center for component j. We note that this design of sampling distributions leads to low values of c, as presented in Table 1.

We use batch size $b = 100$ and number of clusters $k = 100$, and initialize the centers via k-means++ (Arthur & Vassilvitskii, 2007), with shared initialization across all methods. We generate 10 different set of initial centers and run each method 10 times on each set of initial centers. We train the algorithms on 80% of the data. For the mixture sampler, we perform an additional 80%-20% split the training data, in order to choose the hyperparameters β and γ. We report the loss on the test sets of the datasets presented in Table 1 (KDD Cup 2004; Faulkner et al., 2011; LeCun et al., 1998) with more details in the supplementary material.

We are ultimately interested in the performance versus computational time trade-off. Thus, for the samplers, we include in the time measurement the setup and the sampling time. The results are shown in Figure 3, where we measure the relative error of minibatch k-means combined with different samplers compared to batch k-means. The shaded areas represent 95% confidence intervals. VRM suffers initially from a high setup time due to calculation of the proposed sampling distributions of the mixture, but eventually outperforms the other methods. Similarly to Borsos et al. (2018), we observe no advantage on MNIST, where the best-in-hindsight mixture weights are uniform.

Table 1. Dataset details

	KDD	CSN	MNIST
nr. of points	145 751	80 000	70 000
nr. of features	74	17	10
c	48.18	42.42	3.09

7. Conclusion

We proposed a novel framework for online variance reduction with mixtures, in which structures in the data can be easily captured by formulating fixed sampling distributions as mixture components. We devised VRM, a novel importance sampling method for this setting that relies on the Online Newton Step algorithm and we showed that it asymptotically recovers the optimal mixture weight in hindsight. After several considerations for improving efficiency, including a novel projection step on the restricted simplex, we empirically demonstrate the versatility of VRM in a range of applications.

Acknowledgements

This research was supported by the SNSF grant 407540_167212 through the NRP 75 Big Data program and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement No 815943. K.Y.L. is supported by the ETH Zurich Postdoctoral Fellowship and Marie Curie Actions for People COFUND program.

References

Allen-Zhu, Z., Qu, Z., Richtárik, P., and Yuan, Y. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pp. 1110–1119, 2016.

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics, 2007.

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

Borsos, Z., Krause, A., and Levy, K. Y. Online variance reduction for stochastic optimization. In Bubeck, S., Perchet, V., and Rigollet, P. (eds.), *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pp. 324–357. PMLR, 06–09 Jul 2018.

Bouchard, G., Trouillon, T., Perez, J., and Gaidon, A. Online learning to sample. *arXiv preprint arXiv:1506.09016*, 2015.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 1646–1654, 2014.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pp. 272–279. ACM, 2008.

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

Faulkner, M., Olson, M., Chandy, R., Krause, J., Chandy, K. M., and Krause, A. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pp. 13–24. IEEE, 2011.

Freedman, D. A. On tail probabilities for martingales. *the Annals of Probability*, pp. 100–118, 1975.

Gafni, E. M. and Bertsekas, D. P. Two-metric projection methods for constrained optimization. *SIAM Journal on Control and Optimization*, 22(6):936–964, 1984.

Hazan, E., Kalai, A., Kale, S., and Agarwal, A. Logarithmic regret algorithms for online convex optimization. In *Lecture Notes in Computer Science*, volume 4005, pp. 499–513. Springer-Verlag Berlin Heidelberg, June 2006.

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

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.

Johnson, T. B. and Guestrin, C. Training deep models faster with robust, approximate importance sampling. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 7276–7286. Curran Associates, Inc., 2018.

Katharopoulos, A. and Fleuret, F. Not all samples are created equal: Deep learning with importance sampling. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

KDD Cup 2004. KDD Cup 2004. Protein Homology Dataset. http://osmot.cs.cornell.edu/kddcup/, 2004. Accessed: 10.11.2016.

Kulesza, A., Taskar, B., et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, C., Jegelka, S., and Sra, S. Efficient sampling for k-determinantal point processes. In *Artificial Intelligence and Statistics*, pp. 1328–1337, 2016.

Loshchilov, I. and Hutter, F. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Namkoong, H., Sinha, A., Yadlowsky, S., and Duchi, J. C. Adaptive sampling probabilities for non-smooth optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2574–2583, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Needell, D., Ward, R., and Srebro, N. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in Neural Information Processing Systems*, pp. 1017–1025, 2014.

Perekrestenko, D., Cevher, V., and Jaggi, M. Faster coordinate descent via adaptive importance sampling. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54. PMLR, 2017.

Salehi, F., Celis, L. E., and Thiran, P. Stochastic Optimization with Bandit Sampling. *ArXiv e-prints*, August 2017.

Salehi, F., Thiran, P., and Celis, E. Coordinate descent with bandit sampling. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 9267–9277. Curran Associates, Inc., 2018.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. In *International Conference on Learning Representations*, Puerto Rico, 2016.

Sculley, D. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pp. 1177–1178. ACM, 2010.

Stich, S. U., Raj, A., and Jaggi, M. Safe adaptive importance sampling. In *Advances in Neural Information Processing Systems 30*, pp. 4384–4394. Curran Associates, Inc., 2017.

Zhang, C., Kjellstrom, H., and Mandt, S. Determinantal point processes for mini-batch diversification. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Zhao, P. and Zhang, T. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv preprint arXiv:1405.3080*, 2014.

Zhao, P. and Zhang, T. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 1–9, 2015.