On the Generalization Ability of Online Learning Algorithms for Pairwise Loss Functions

Purushottam Kar PURUSHOT@CSE.IITK.AC.IN
Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, UP 208 016, INDIA.

Bharath K Sriperumbudur BS493@STATSLAB.CAM.AC.UK
Statistical Laboratory, Centre for Mathematical Sciences, Wilberforce Road, Cambridge, CB3 0WB, ENGLAND.

Prateek Jain PRAJAIN@MICROSOFT.COM
Microsoft Research India, "Vigyan", #9, Lavelle Road, Bangalore, KA 560 001, INDIA.

Harish C Karnick HK@CSE.IITK.AC.IN
Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, UP 208 016, INDIA.

Abstract

In this paper, we study the generalization properties of online learning based stochastic methods for supervised learning problems where the loss function is dependent on more than one training sample (e.g., metric learning, ranking). We present a generic decoupling technique that enables us to provide Rademacher complexity-based generalization error bounds. Our bounds are in general tighter than those obtained by Wang et al. (2012) for the same problem. Using our decoupling technique, we are further able to obtain fast convergence rates for strongly convex pairwise loss functions. We are also able to analyze a class of memory efficient online learning algorithms for pairwise learning problems that use only a bounded subset of past training samples to update the hypothesis at each step. Finally, in order to complement our generalization bounds, we propose a novel memory efficient online learning algorithm for higher order learning problems with bounded regret guarantees.

1. Introduction

Several supervised learning problems involve working with pairwise or higher order loss functions, i.e., loss functions that depend on more than one training sample. Take for example the *metric learning* problem (Jin et al., 2009), where the goal is to learn a metric M that brings points of a similar label together while keeping differently labeled points apart. In this case the loss function used is a pairwise loss function $\ell(M, (\mathbf{x}, y), (\mathbf{x}', y')) = \phi\left(yy'\left(1 - M(\mathbf{x}, \mathbf{x}')\right)\right)$ where ϕ is the hinge loss function. In general, a pairwise loss function is of the form $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ where $\mathcal{H}$ is the hypothesis space and $\mathcal{X}$ is the input domain. Other examples include preference learning (Xing et al., 2002), ranking (Agarwal & Niyogi, 2009), AUC maximization (Zhao et al., 2011) and multiple kernel learning (Kumar et al., 2012).

In practice, algorithms for such problems use intersecting pairs of training samples to learn. Hence the training data pairs are not i.i.d. and consequently, standard generalization error analysis techniques do not apply to these algorithms. Recently, the analysis of *batch* algorithms learning from such coupled samples has received much attention (Cao et al., 2012; Clémençon et al., 2008; Brefeld & Scheffer, 2005) where a dominant idea has been to use an alternate representation of the U-statistic and provide uniform convergence bounds. Another popular approach has been to use algorithmic stability (Agarwal & Niyogi, 2009; Jin et al., 2009) to obtain algorithm-specific results.

While batch algorithms for pairwise (and higher-order) learning problems have been studied well theoretically, online learning based stochastic algorithms are more popular in practice due to their scalability. However, their generalization properties were not studied until recently. Wang et al. (2012) provided the first generalization error analysis of online learning methods

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

applied to pairwise loss functions. In particular, they showed that such higher-order online learning methods also admit online to batch conversion bounds (similar to those for first-order problems (Cesa-Bianchi et al., 2001)) which can be combined with regret bounds to obtain generalization error bounds. However, due to their proof technique and dependence on L_∞ covering numbers of function classes, their bounds are not tight and have a strong dependence on the dimensionality of the input space.

In literature, there are several instances where Rademacher complexity based techniques achieve sharper bounds than those based on covering numbers (Kakade et al., 2008). However, the coupling of different input pairs in our problem does not allow us to use such techniques directly.

In this paper we introduce a generic technique for analyzing online learning algorithms for higher order learning problems. Our technique, that uses an extension of Rademacher complexities to higher order function classes (instead of covering numbers), allows us to give bounds that are tighter than those of (Wang et al., 2012) and that, for several learning scenarios, have no dependence on input dimensionality at all.

Key to our proof is a technique we call *Symmetrization of Expectations* which acts as a decoupling step and allows us to reduce excess risk estimates to Rademacher complexities of function classes. (Wang et al., 2012), on the other hand, perform a symmetrization with probabilities which, apart from being more involved, yields suboptimal bounds. Another advantage of our technique is that it allows us to obtain *fast* convergence rates for learning algorithms that use *strongly convex* loss functions. Our result, that uses a novel two stage proof technique, extends a similar result in the first order setting by Kakade & Tewari (2008) to the pairwise setting.

Wang et al. (2012) (and our results mentioned above) assume an online learning setup in which a stream of points $\mathbf{z}_1, \ldots, \mathbf{z}_n$ is observed and the penalty function used at the t^{th} step is $\hat{\mathcal{L}}_t(h) = \frac{1}{t-1}\sum_{\tau=1}^{t-1} \ell(h, \mathbf{z}_t, \mathbf{z}_\tau)$. Consequently, the results of Wang et al. (2012) expect regret bounds with respect to these *all-pairs* penalties $\hat{\mathcal{L}}_t$. This requires one to use/store all previously seen points which is computationally/storagewise expensive and hence in practice, learning algorithms update their hypotheses using only a bounded subset of the past samples (Zhao et al., 2011).

In the above mentioned setting, we are able to give generalization bounds that only require algorithms to give regret bounds with respect to *finite-buffer* penalty functions such as $\hat{\mathcal{L}}_t^{\text{buf}}(h) = \frac{1}{|B|}\sum_{\mathbf{z}\in B} \ell(h, \mathbf{z}_t, \mathbf{z})$ where B is a *buffer* that is updated at each step. Our proofs hold for any *stream oblivious* buffer update policy including FIFO and the widely used reservoir sampling policy (Vitter, 1985; Zhao et al., 2011)[1].

To complement our online to batch conversion bounds, we also provide a memory efficient online learning algorithm that works with bounded buffers. Although our algorithm is constrained to observe and learn using the *finite-buffer* penalties $\hat{\mathcal{L}}_t^{\text{buf}}$ alone, we are still able to provide high confidence regret bounds with respect to the *all-pairs* penalty functions $\hat{\mathcal{L}}_t$. We note that Zhao et al. (2011) also propose an algorithm that uses finite buffers and claim an *all-pairs* regret bound for the same. However, their regret bound does not hold due to a subtle mistake in their proof.

We also provide empirical validation of our proposed online learning algorithm on AUC maximization tasks and show that our algorithm performs competitively with that of (Zhao et al., 2011), in addition to being able to offer theoretical regret bounds.

Our Contributions:

(a) We provide a generic online-to-batch conversion technique for higher-order supervised learning problems offering bounds that are sharper than those of (Wang et al., 2012).

(b) We obtain fast convergence rates when loss functions are *strongly convex*.

(c) We analyze online learning algorithms that are constrained to learn using a finite buffer.

(d) We propose a novel online learning algorithm that works with finite buffers but is able to provide a high confidence regret bound with respect to the *all-pairs* penalty functions.

2. Problem Setup

For ease of exposition, we introduce an online learning model for higher order supervised learning problems in this section; concrete learning instances such as AUC maximization and metric learning are given in Section 6. For sake of simplicity, we restrict ourselves to pairwise problems in this paper; our techniques can be readily extended to higher order problems as well.

For pairwise learning problems, our goal is to learn a

[1] Independently, Wang et al. (2013) also extended their proof to give similar guarantees. However, their bounds hold only for the FIFO update policy and have worse dependence on dimensionality in several cases (see Section 5).

real valued *bivariate* function $h^* : \mathcal{X} \times \mathcal{X} \to \mathcal{Y}$, where $h^* \in \mathcal{H}$, under some loss function $\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}^+$ where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.

The online learning algorithm is given sequential access to a stream of elements $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$ chosen i.i.d. from the domain $\mathcal{Z}$. Let $Z^t := \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$. At each time step $t = 2 \ldots n$, the algorithm posits a hypothesis $h_{t-1} \in \mathcal{H}$ upon which the element $\mathbf{z}_t$ is revealed and the algorithm incurs the following penalty:

$$\hat{\mathcal{L}}_t(h_{t-1}) = \frac{1}{t-1} \sum_{\tau=1}^{t-1} \ell(h_{t-1}, \mathbf{z}_t, \mathbf{z}_\tau). \tag{1}$$

For any $h \in \mathcal{H}$, we define its expected risk as:

$$\mathcal{L}(h) := \mathop{\mathbb{E}}_{\mathbf{z},\mathbf{z}'} \left[\!\left[\ell(h, \mathbf{z}, \mathbf{z}') \right]\!\right]. \tag{2}$$

Our aim is to present an ensemble $h_1, \ldots, h_{n-1}$ such that the expected risk of the ensemble is small. More specifically, we desire that, for some small $\epsilon > 0$,

$$\frac{1}{n-1} \sum_{t=2}^{n} \mathcal{L}(h_{t-1}) \leq \mathcal{L}(h^*) + \epsilon,$$

where $h^* = \mathop{\arg\min}_{h \in \mathcal{H}} \mathcal{L}(h)$ is the population risk minimizer. Note that this allows us to do hypothesis selection in a way that ensures small expected risk. Specifically, if one chooses a hypothesis as $\hat{h} := \frac{1}{(n-1)} \sum_{t=2}^{n} h_{t-1}$ (for convex ℓ) or $\hat{h} := \mathop{\arg\min}_{t=2,\ldots,n} \mathcal{L}(h_t)$ then we have $\mathcal{L}(\hat{h}) \leq \mathcal{L}(h^*) + \epsilon$.

Since the model presented above requires storing all previously seen points, it becomes unusable in large scale learning scenarios. Instead, in practice, a *sketch* of the stream is maintained in a buffer B of capacity s. At each step, the penalty is now incurred only on the pairs $\{(\mathbf{z}_t, \mathbf{z}) : \mathbf{z} \in B_t\}$ where B_t is the state of the buffer at time t. That is,

$$\hat{\mathcal{L}}_t^{\text{buf}}(h_{t-1}) = \frac{1}{|B_t|} \sum_{\mathbf{z} \in B_t} \ell(h_{t-1}, \mathbf{z}_t, \mathbf{z}). \tag{3}$$

We shall assume that the buffer is updated at each step using some *stream oblivious* policy such as FIFO or Reservoir sampling (Vitter, 1985) (see Section 5).

In Section 3, we present online-to-batch conversion bounds for online learning algorithms that give regret bounds w.r.t. penalty functions given by (1). In Section 4, we extend our analysis to algorithms using strongly convex loss functions. In Section 5 we provide generalization error bounds for algorithms that give regret bounds w.r.t. *finite-buffer* penalty functions given by (3). Finally in section 7 we present a novel memory efficient online learning algorithm with regret bounds.

3. Online to Batch Conversion Bounds for Bounded Loss Functions

We now present our generalization bounds for algorithms that provide regret bounds with respect to the *all-pairs* loss functions (see Eq. (1)). Our results give tighter bounds and have a much better dependence on input dimensionality than the bounds given by Wang et al. (2012). See Section 3.1 for a detailed comparison.

As was noted by (Wang et al., 2012), the generalization error analysis of online learning algorithms in this setting does not follow from existing techniques for first-order problems (such as (Cesa-Bianchi et al., 2001; Kakade & Tewari, 2008)). The reason is that the terms $V_t = \hat{\mathcal{L}}_t(h_{t-1})$ do not form a martingale due to the intersection of training samples in V_t and $V_\tau, \tau < t$.

Our technique, that aims to utilize the Rademacher complexities of function classes in order to get tighter bounds, faces yet another challenge at the *symmetrization* step, a precursor to the introduction of Rademacher complexities. It turns out that, due to the coupling between the "head" variable $\mathbf{z}_t$ and the "tail" variables $\mathbf{z}_\tau$ in the loss function $\hat{\mathcal{L}}_t$, a standard symmetrization between true $\mathbf{z}_\tau$ and ghost $\tilde{\mathbf{z}}_\tau$ samples does not succeed in generating Rademacher averages and instead yields complex looking terms.

More specifically, suppose we have *true* variables $\mathbf{z}_t$ and *ghost* variables $\tilde{\mathbf{z}}_t$ and are in the process of bounding the expected excess risk by analyzing expressions of the form

$$E_{\text{orig}} = \ell(h_{t-1}, \mathbf{z}_t, \mathbf{z}_\tau) - \ell(h_{t-1}, \tilde{\mathbf{z}}_t, \tilde{\mathbf{z}}_\tau).$$

Performing a traditional symmetrization of the variables $\mathbf{z}_\tau$ with $\tilde{\mathbf{z}}_\tau$ would give us expressions of the form

$$E_{\text{symm}} = \ell(h_{t-1}, \mathbf{z}_t, \tilde{\mathbf{z}}_\tau) - \ell(h_{t-1}, \tilde{\mathbf{z}}_t, \mathbf{z}_\tau).$$

At this point the analysis hits a barrier since unlike first order situations, we cannot relate E_{symm} to E_{orig} by means of introducing Rademacher variables.

We circumvent this problem by using a technique that we call *Symmetrization of Expectations*. The technique allows us to use standard symmetrization to obtain Rademacher complexities. More specifically, we analyze expressions of the form

$$E'_{\text{orig}} = \mathop{\mathbb{E}}_{\mathbf{z}} \left[\!\left[\ell(h_{t-1}, \mathbf{z}, \mathbf{z}_\tau) \right]\!\right] - \mathop{\mathbb{E}}_{\mathbf{z}} \left[\!\left[\ell(h_{t-1}, \mathbf{z}, \tilde{\mathbf{z}}_\tau) \right]\!\right]$$

which upon symmetrization yield expressions such as

$$E'_{\text{symm}} = \mathop{\mathbb{E}}_{\mathbf{z}} \left[\!\left[\ell(h_{t-1}, \mathbf{z}, \tilde{\mathbf{z}}_\tau) \right]\!\right] - \mathop{\mathbb{E}}_{\mathbf{z}} \left[\!\left[\ell(h_{t-1}, \mathbf{z}, \mathbf{z}_\tau) \right]\!\right]$$

which allow us to introduce Rademacher variables since $E'_{\text{symm}} = -E'_{\text{orig}}$. This idea is exploited by the

lemma given below that relates the expected risk of the ensemble to the penalties incurred during the online learning process. In the following we use the following extension of Rademacher averages (Kakade et al., 2008) to bivariate function classes:

$$\mathcal{R}_n(\mathcal{H}) = \mathbb{E}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{\tau=1}^{n} \epsilon_\tau h(\mathbf{z}, \mathbf{z}_\tau)\right]$$

where the expectation is over ϵ_τ, $\mathbf{z}$ and $\mathbf{z}_\tau$. We shall denote composite function classes as follows : $\ell \circ \mathcal{H} := \{(\mathbf{z}, \mathbf{z}') \mapsto \ell(h, \mathbf{z}, \mathbf{z}'), h \in \mathcal{H}\}$.

Lemma 1. *Let $h_1, \ldots, h_{n-1}$ be an ensemble of hypotheses generated by an online learning algorithm working with a bounded loss function $\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to [0, B]$. Then for any $\delta > 0$, we have with probability at least $1 - \delta$,*

$$\frac{1}{n-1}\sum_{t=2}^{n} \mathcal{L}(h_{t-1}) \leq \frac{1}{n-1}\sum_{t=2}^{n} \hat{\mathcal{L}}_t(h_{t-1}) + \frac{2}{n-1}\sum_{t=2}^{n} \mathcal{R}_{t-1}(\ell \circ \mathcal{H}) + 3B\sqrt{\frac{\log\frac{n}{\delta}}{n-1}}.$$

The proof of the lemma involves decomposing the excess risk term into a martingale difference sequence and a residual term in a manner similar to (Wang et al., 2012). The martingale sequence, being a bounded one, is shown to converge using the Azuma-Hoeffding inequality. The residual term is handled using uniform convergence techniques involving Rademacher averages. The complete proof of the lemma is given in the Appendix A.

Similar to Lemma 1, the following converse relation between the population and empirical risk of the population risk minimizer h^* can also be shown.

Lemma 2. *For any $\delta > 0$, we have with probability at least $1 - \delta$,*

$$\frac{1}{n-1}\sum_{t=2}^{n} \hat{\mathcal{L}}_t(h^*) \leq \mathcal{L}(h^*) + \frac{2}{n-1}\sum_{t=2}^{n} \mathcal{R}_{t-1}(\ell \circ \mathcal{H}) + 3B\sqrt{\frac{\log\frac{1}{\delta}}{n-1}}.$$

An online learning algorithm will be said to have an *all-pairs* regret bound $\mathfrak{R}_n$ if it presents an ensemble $h_1, \ldots, h_{n-1}$ such that

$$\sum_{t=2}^{n} \hat{\mathcal{L}}_t(h_{t-1}) \leq \inf_{h\in\mathcal{H}} \sum_{t=2}^{n} \hat{\mathcal{L}}_t(h) + \mathfrak{R}_n.$$

Suppose we have an online learning algorithm with a regret bound $\mathfrak{R}_n$. Then combining Lemmata 1 and 2 gives us the following online to batch conversion bound:

Theorem 3. *Let $h_1, \ldots, h_{n-1}$ be an ensemble of hypotheses generated by an online learning algorithm working with a B-bounded loss function ℓ that guarantees a regret bound of $\mathfrak{R}_n$. Then for any $\delta > 0$, we have with probability at least $1 - \delta$,*

$$\frac{1}{n-1}\sum_{t=2}^{n} \mathcal{L}(h_{t-1}) \leq \mathcal{L}(h^*) + \frac{4}{n-1}\sum_{t=2}^{n} \mathcal{R}_{t-1}(\ell \circ \mathcal{H}) + \frac{\mathfrak{R}_n}{n-1} + 6B\sqrt{\frac{\log\frac{n}{\delta}}{n-1}}.$$

As we shall see in Section 6, for several learning problems, the Rademacher complexities behave as $\mathcal{R}_{t-1}(\ell \circ \mathcal{H}) \leq C_d \cdot \mathcal{O}\left(\frac{1}{\sqrt{t-1}}\right)$ where C_d is a constant dependent only on the dimension d of the input space and the $\mathcal{O}(\cdot)$ notation hides constants dependent on the domain size and the loss function. This allows us to bound the excess risk as follows:

$$\frac{\sum_{t=2}^{n} \mathcal{L}(h_{t-1})}{n-1} \leq \mathcal{L}(h^*) + \frac{\mathfrak{R}_n}{n-1} + \mathcal{O}\left(\frac{C_d + \sqrt{\log(n/\delta)}}{\sqrt{n-1}}\right).$$

Here, the error decreases with n at a standard $1/\sqrt{n}$ rate (up to a $\sqrt{\log n}$ factor), similar to that obtained by Wang et al. (2012). However, for several problems the above bound can be significantly tighter than those offered by covering number based arguments. We provide below a detailed comparison of our results with those of Wang et al. (2012).

3.1. Discussion on the nature of our bounds

As mentioned above, our proof enables us to use Rademacher complexities which are typically easier to analyze and provide tighter bounds (Kakade et al., 2008). In particular, as shown in Section 6, for L_2 regularized learning formulations, the Rademacher complexities are dimension independent i.e. $C_d = 1$. Consequently, unlike the bounds of (Wang et al., 2012) that have a linear dependence on d, our bound becomes independent of the input space dimension. For sparse learning formulations with L_1 or trace norm regularization, we have $C_d = \sqrt{\log d}$ giving us a mild dependence on the input dimensionality.

Our bounds are also tighter that those of (Wang et al., 2012) in general. Whereas we provide a confidence bound of $\delta < \exp\left(-n\epsilon^2 + \log n\right)$, (Wang et al., 2012) offer a weaker bound $\delta < (1/\epsilon)^d \exp\left(-n\epsilon^2 + \log n\right)$.

An artifact of the proof technique of (Wang et al., 2012) is that their proof is required to exclude a constant fraction of the ensemble $(h_1, \ldots, h_{cn})$ from the

analysis, failing which their bounds turn vacuous. Our proof on the other hand is able to give guarantees for the *entire* ensemble.

In addition to this, as the following sections show, our proof technique enjoys the flexibility of being extendable to give fast convergence guarantees for strongly convex loss functions as well as being able to accommodate learning algorithms that use finite buffers.

4. Fast Convergence Rates for Strongly Convex Loss Functions

In this section we extend results of the previous section to give *fast* convergence guarantees for online learning algorithms that use strongly convex loss functions of the following form: $\ell(h, \mathbf{z}, \mathbf{z}') = g(\langle h, \phi(\mathbf{z}, \mathbf{z}')\rangle) + r(h)$, where g is a convex function and $r(h)$ is a σ-strongly convex regularizer (see Section 6 for examples) i.e. $\forall h_1, h_2 \in \mathcal{H}$ and $\alpha \in [0, 1]$, we have

$$r(\alpha h_1 + (1-\alpha)h_2) \leq \alpha r(h_1) + (1-\alpha) r(h_2) - \frac{\sigma}{2}\alpha(1-\alpha)\left\|h_1 - h_2\right\|^2.$$

For any norm $\|\cdot\|$, let $\|\cdot\|_*$ denote its dual norm. Our analysis reduces the pairwise problem to a first order problem and a martingale convergence problem. We require the following *fast* convergence bound in the standard first order *batch* learning setting:

Theorem 4. *Let $\mathcal{F}$ be a closed and convex set of functions over $\mathcal{X}$. Let $\wp(f, \mathbf{x}) = p(\langle f, \phi(\mathbf{x})\rangle) + r(f)$, for a σ-strongly convex function r, be a loss function with $\mathcal{P}$ and $\hat{\mathcal{P}}$ as the associated population and empirical risk functionals and f^* as the population risk minimizer. Suppose $\wp$ is L-Lipschitz and $\|\phi(\mathbf{x})\|_* \leq R, \forall \mathbf{x} \in \mathcal{X}$. Then w.p. $1-\delta$, for any $\epsilon > 0$, we have for all $f \in \mathcal{F}$,*

$$\mathcal{P}(f) - \mathcal{P}(f^*) \leq (1+\epsilon)\left(\hat{\mathcal{P}}(f) - \hat{\mathcal{P}}(f^*)\right) + \frac{C_\delta}{\epsilon\sigma n}$$

where $C_\delta = C_d^2 \cdot (4(1+\epsilon)LR)^2\,(32 + \log(1/\delta))$ and C_d is the dependence of the Rademacher complexity of the class $\mathcal{F}$ on the input dimensionality d.

The above theorem is a minor modification of a similar result by Sridharan et al. (2008) and the proof (given in Appendix B) closely follows their proof as well. We can now state our online to batch conversion result for strongly convex loss functions.

Theorem 5. *Let $h_1, \ldots, h_{n-1}$ be an ensemble of hypotheses generated by an online learning algorithm working with a B-bounded, L-Lipschitz and σ-strongly convex loss function ℓ. Further suppose the learning algorithm guarantees a regret bound of $\mathfrak{R}_n$. Let $\mathfrak{V}_n = \max\left\{\mathfrak{R}_n, 2C_d^2 \log n \log(n/\delta)\right\}$ Then for any $\delta > 0$, we have with probability at least $1-\delta$,*

$$\frac{1}{n-1}\sum_{t=2}^{n}\mathcal{L}(h_{t-1}) \leq \mathcal{L}(h^*) + \frac{\mathfrak{R}_n}{n-1} + C_d \cdot \mathcal{O}\left(\frac{\sqrt{\mathfrak{V}_n \log n \log(n/\delta)}}{n-1}\right),$$

where the $\mathcal{O}(\cdot)$ notation hides constants dependent on domain size and the loss function such as L, B and σ.

The decomposition of the excess risk in this case is not made explicitly but rather emerges as a side-effect of the proof progression. The proof starts off by applying Theorem 4 to the hypothesis in each round with the following loss function $\wp(h, \mathbf{z}') := \mathop{\mathbb{E}}_{\mathbf{z}}[\ell(h, \mathbf{z}, \mathbf{z}')]$. Applying the regret bound to the resulting expression gives us a martingale difference sequence which we then bound using Bernstein-style inequalities and a proof technique from (Kakade & Tewari, 2008). The complete proof is given in Appendix C.

We now note some properties of this result. The effective dependence of the above bound on the input dimensionality is C_d^2 since the expression $\sqrt{\mathfrak{V}_n}$ hides a C_d term. We have $C_d^2 = 1$ for non sparse learning formulations and $C_d^2 = \log d$ for sparse learning formulations. We note that our bound matches that of Kakade & Tewari (2008) (for *first*-order learning problems) up to a logarithmic factor.

5. Analyzing Online Learning Algorithms that use Finite Buffers

In this section, we present our online to batch conversion bounds for algorithms that work with *finite-buffer* loss functions $\hat{\mathcal{L}}_t^{\text{buf}}$. Recall that an online learning algorithm working with finite buffers incurs a loss $\hat{\mathcal{L}}_t^{\text{buf}}(h) = \frac{1}{|B_t|}\sum_{\mathbf{z}\in B_t}\ell(h_{t-1}, \mathbf{z}_t, \mathbf{z})$ at each step where B_t is the state of the buffer at time t.

An online learning algorithm will be said to have a *finite-buffer* regret bound $\mathfrak{R}_n^{\text{buf}}$ if it presents an ensemble $h_1, \ldots, h_{n-1}$ such that

$$\sum_{t=2}^{n}\hat{\mathcal{L}}_t^{\text{buf}}(h_{t-1}) - \inf_{h\in\mathcal{H}}\sum_{t=2}^{n}\hat{\mathcal{L}}_t^{\text{buf}}(h) \leq \mathfrak{R}_n^{\text{buf}}.$$

For our guarantees to hold, we require the buffer update policy used by the learning algorithm to be *stream oblivious*. More specifically, we require the buffer update rule to decide upon the inclusion of a particular point $\mathbf{z}_i$ in the buffer based only on its stream index $i \in [n]$. Popular examples of stream oblivious policies include Reservoir sampling (Vitter, 1985) (referred to

as **RS** henceforth) and FIFO. Stream oblivious policies allow us to decouple buffer construction randomness from training sample randomness which makes analysis easier; we leave the analysis of *stream aware* buffer update policies as a topic of future research.

In the above mentioned setting, we can prove the following online to batch conversion bounds:

Theorem 6. *Let $h_1, \ldots, h_{n-1}$ be an ensemble of hypotheses generated by an online learning algorithm working with a finite buffer of capacity s and a B-bounded loss function ℓ. Moreover, suppose that the algorithm guarantees a regret bound of $\mathfrak{R}_n^{buf}$. Then for any $\delta > 0$, we have with probability at least $1-\delta$,*

$$\frac{\sum_{t=2}^{n} \mathcal{L}(h_{t-1})}{n-1} \leq \mathcal{L}(h^*) + \frac{\mathfrak{R}_n^{buf}}{n-1} + \mathcal{O}\left(\frac{C_d}{\sqrt{s}} + B\sqrt{\frac{\log \frac{n}{\delta}}{s}}\right)$$

If the loss function is Lipschitz and strongly convex as well, then with the same confidence, we have

$$\frac{\sum_{t=2}^{n} \mathcal{L}(h_{t-1})}{n-1} \leq \mathcal{L}(h^*) + \frac{\mathfrak{R}_n^{buf}}{n-1} + C_d \cdot \mathcal{O}\left(\sqrt{\frac{\mathfrak{W}_n \log \frac{n}{\delta}}{sn}}\right)$$

where $\mathfrak{W}_n = \max\left\{\mathfrak{R}_n^{buf}, \frac{2C_d^2 n \log(n/\delta)}{s}\right\}$ and C_d is the dependence of $\mathcal{R}_n(\mathcal{H})$ on the input dimensionality d.

The above bound guarantees an excess error of $\tilde{\mathcal{O}}(1/s)$ for algorithms (such as Follow-the-leader (Hazan et al., 2006)) that offer logarithmic regret $\mathfrak{R}_n^{\text{buf}} = \mathcal{O}(\log n)$. We stress that this theorem is not a direct corollary of our results for the *infinite buffer* case (Theorems 3 and 5). Instead, our proofs require a more careful analysis of the excess risk in order to accommodate the finiteness of the buffer and the randomness (possibly) used in constructing it.

More specifically, care needs to be taken to handle randomized buffer update policies such as **RS** which introduce additional randomness into the analysis. A naive application of techniques used to prove results for the unbounded buffer case would result in bounds that give non trivial generalization guarantees only for large buffer sizes such as $s = \omega(\sqrt{n})$. Our bounds, on the other hand, only require $s = \tilde{\omega}(1)$.

Key to our proofs is a conditioning step where we first analyze the conditional excess risk by conditioning upon randomness used by the buffer update policy. Such conditioning is made possible by the stream-oblivious nature of the update policy and thus, stream-obliviousness is required by our analysis. Subsequently, we analyze the excess risk by taking expectations over randomness used by the buffer update policy. The complete proofs of both parts of Theorem 6 are given in Appendix D.

Note that the above results only require an online learning algorithm to provide regret bounds w.r.t. the *finite-buffer* penalties $\hat{\mathcal{L}}_t^{\text{buf}}$ and do not require any regret bounds w.r.t the *all-pairs* penalties $\hat{\mathcal{L}}_t$.

For instance, the finite buffer based online learning algorithms OAM_{seq} and OAM_{gra} proposed in (Zhao et al., 2011) are able to provide a regret bound w.r.t. $\hat{\mathcal{L}}_t^{\text{buf}}$ (Zhao et al., 2011, Lemma 2) but are not able to do so w.r.t the *all-pairs* loss function (see Section 7 for a discussion). Using Theorem 6, we are able to give a generalization bound for OAM_{seq} and OAM_{gra} and hence explain the good empirical performance of these algorithms as reported in (Zhao et al., 2011). Note that Wang et al. (2013) are not able to analyze OAM_{seq} and OAM_{gra} since their analysis is restricted to algorithms that use the (deterministic) FIFO update policy whereas OAM_{seq} and OAM_{gra} use the (randomized) **RS** policy of Vitter (1985).

6. Applications

In this section we make explicit our online to batch conversion bounds for several learning scenarios and also demonstrate their dependence on input dimensionality by calculating their respective Rademacher complexities. Recall that our definition of Rademacher complexity for a pairwise function class is given by,

$$\mathcal{R}_n(\mathcal{H}) = \mathbb{E}\left[\!\!\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{\tau=1}^{n} \epsilon_\tau h(\mathbf{z}, \mathbf{z}_\tau)\right]\!\!\right].$$

For our purposes, we would be interested in the Rademacher complexities of *composition classes* of the form $\ell \circ \mathcal{H} := \{(\mathbf{z}, \mathbf{z}') \mapsto \ell(h, \mathbf{z}, \mathbf{z}'), h \in \mathcal{H}\}$ where ℓ is some Lipschitz loss function. Frequently we have $\ell(h, \mathbf{z}, \mathbf{z}') = \phi\left(h(\mathbf{x}, \mathbf{x}')Y(y, y')\right)$ where $Y(y, y') = y - y'$ or $Y(y, y') = yy'$ and $\phi : \mathbb{R} \to \mathbb{R}$ is some margin loss function (Steinwart & Christmann, 2008). Suppose ϕ is L-Lipschitz and $Y = \sup_{y,y'\in\mathcal{Y}} |Y(y, y')|$. Then we have

Theorem 7. $\mathcal{R}_n(\ell \circ \mathcal{H}) \leq LY\mathcal{R}_n(\mathcal{H})$.

The proof uses standard contraction inequalities and is given in Appendix E. This reduces our task to computing the values of $\mathcal{R}_n(\mathcal{H})$ which we do using a two stage proof technique (see Appendix F). For any subset X of a Banach space and any norm $\|\cdot\|_p$, we define $\|X\|_p := \sup_{\mathbf{x}\in X} \|\mathbf{x}\|_p$. Let the domain $\mathcal{X} \subset \mathbb{R}^d$.

AUC maximization (Zhao et al., 2011): the goal here is to maximize the area under the ROC curve for a linear classification problem where the hypothesis space $\mathcal{W} \subset \mathbb{R}^d$. We have $h_{\mathbf{w}}(\mathbf{x}, \mathbf{x}') = \mathbf{w}^\top\mathbf{x} - \mathbf{w}^\top\mathbf{x}'$ and $\ell(h_{\mathbf{w}}, \mathbf{z}, \mathbf{z}') = \phi\left((y - y')h_{\mathbf{w}}(\mathbf{x}, \mathbf{x}')\right)$ where ϕ is the

hinge loss. In case our classifiers are L_p regularized for $p > 1$, we can show that $\mathcal{R}_n(\mathcal{W}) \leq 2 \|\mathcal{X}\|_q \|\mathcal{W}\|_p \sqrt{\frac{q-1}{n}}$ where $q = p/(p-1)$. Using the sparsity promoting L_1 regularizer gives us $\mathcal{R}_n(\mathcal{W}) \leq 2 \|\mathcal{X}\|_\infty \|\mathcal{W}\|_1 \sqrt{\frac{e \log d}{n}}$. Note that we obtain dimension independence, for example when the classifiers are L_2 regularized which allows us to bound the Rademacher complexities of kernelized function classes for bounded kernels as well.

Metric learning (Jin et al., 2009): the goal here is to learn a Mahalanobis metric $M_{\mathbf{W}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^\top \mathbf{W}(\mathbf{x} - \mathbf{x}')$ using the loss function $\ell(\mathbf{W}, \mathbf{z}, \mathbf{z}') = \phi\left(yy'\left(1 - M^2_{\mathbf{W}}(\mathbf{x}, \mathbf{x}')\right)\right)$ for a hypothesis class $\mathcal{W} \subset \mathbb{R}^{d \times d}$. In this case it is possible to use a variety of mixed norm $\|\cdot\|_{p,q}$ and Schatten norm $\|\cdot\|_{S(p)}$ regularizations on matrices in the hypothesis class. In case we use trace norm regularization on the matrix class, we get $\mathcal{R}_n(\mathcal{W}) \leq \|\mathcal{X}\|_2^2 \|\mathcal{W}\|_{S(1)} \sqrt{\frac{e \log d}{n}}$. The $(2,2)$-norm regularization offers a dimension independent bound $\mathcal{R}_n(\mathcal{W}) \leq \|\mathcal{X}\|_2^2 \|\mathcal{W}\|_{2,2} \sqrt{\frac{1}{n}}$. The mixed $(2,1)$-norm regularization offers $\mathcal{R}_n(\mathcal{W}) \leq \|\mathcal{X}\|_2 \|\mathcal{X}\|_\infty \|\mathcal{W}\|_{2,1} \sqrt{\frac{e \log d}{n}}$.

Multiple kernel learning (Kumar et al., 2012): the goal here is to improve the SVM classification algorithm by learning a *good* kernel K that is a positive combination of *base* kernels $K_1, \ldots, K_p$ i.e. $K_{\boldsymbol{\mu}}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^p \boldsymbol{\mu}_i K_i(\mathbf{x}, \mathbf{x}')$ for some $\boldsymbol{\mu} \in \mathbb{R}^p, \boldsymbol{\mu} \geq 0$. The base kernels are bounded, i.e. for all i, $|K_i(\mathbf{x}, \mathbf{x}')| \leq \kappa^2$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ The notion of goodness used here is the one proposed by Balcan & Blum (2006) and involves using the loss function $\ell(\boldsymbol{\mu}, \mathbf{z}, \mathbf{z}') = \phi\left(yy' K_{\boldsymbol{\mu}}(\mathbf{x}, \mathbf{x}')\right)$ where $\phi(\cdot)$ is a margin loss function meant to encode some notion of alignment. The two hypothesis classes for the combination vector $\boldsymbol{\mu}$ that we study are the L_1 regularized unit simplex $\Delta(1) = \{\boldsymbol{\mu} : \|\boldsymbol{\mu}\|_1 = 1, \boldsymbol{\mu} \geq 0\}$ and the L_2 regularized unit sphere $\mathcal{S}_2(1) = \{\boldsymbol{\mu} : \|\boldsymbol{\mu}\|_2 = 1, \boldsymbol{\mu} \geq 0\}$. We are able to show the following Rademacher complexity bounds for these classes: $\mathcal{R}_n(\mathcal{S}_2(1)) \leq \kappa^2 \sqrt{\frac{p}{n}}$ and $\mathcal{R}_n(\Delta(1)) \leq \kappa^2 \sqrt{\frac{e \log p}{n}}$.

The details of the Rademacher complexity derivations for these problems and other examples such as similarity learning can be found in Appendix F.

7. OLP : Online Learning with Pairwise Loss Functions

In this section, we present an online learning algorithm for learning with pairwise loss functions in a finite buffer setting. The key contribution in this section is a buffer update policy that when combined with a variant of the GIGA algorithm (Zinkevich, 2003) allows us to give high probability regret bounds.

Algorithm 1 RS-x : Stream Subsampling with Replacement

Input: Buffer B, new point $\mathbf{z}_t$, buffer size s, timestep t.

```
if |B| < s then                                   //There is space
    B ← B ∪ {z_t}
else                                              //Overflow situation
    if t = s + 1 then                             //Repopulation step
        TMP ← B ∪ {z_t}
        Repopulate B with s points sampled uniformly with replacement from TMP.
    else                                          //Normal update step
        Independently, replace each point of B with z_t with probability 1/t.
    end if
end if
```

Algorithm 2 OLP : Online Learning with Pairwise Loss Functions

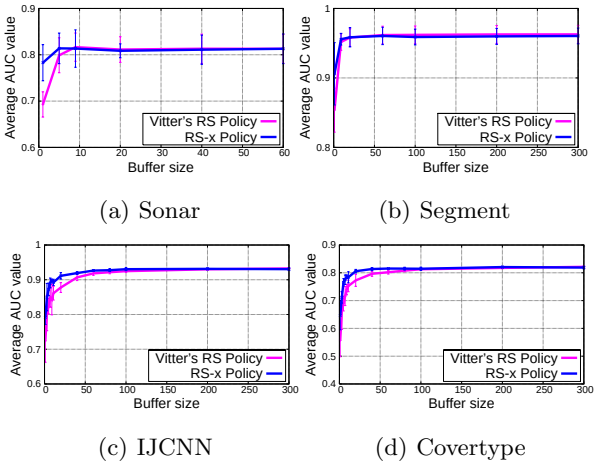
Input: Step length scale η, Buffer size s

Output: An ensemble $\mathbf{w}_2, \ldots, \mathbf{w}_n \in \mathcal{W}$ with low regret

```
w_0 ← 0, B ← φ
for t = 1 to n do
    Obtain a training point z_t
    Set step length η_t ← η/√t
    w_t ← Π_W [ w_{t-1} + (η_t/|B|) Σ_{z∈B} ∇_w ℓ(w_{t-1}, z_t, z) ]
                                                  //Π_W projects onto the set W
    B ← Update-buffer(B, z_t, s, t)               //using RS-x
end for
return w_2, ..., w_n
```

In previous work, Zhao et al. (2011) presented an online learning algorithm that uses finite buffers with the **RS** policy and proposed an *all-pairs* regret bound. The **RS** policy ensures, over the randomness used in buffer updates, that at any given time, the buffer contains a uniform sample from the preceding stream. Using this property, (Zhao et al., 2011, Lemma 2) claimed that $\mathbb{E}\left[\!\left[\hat{\mathcal{L}}_t^{\text{buf}}(h_{t-1})\right]\!\right] = \hat{\mathcal{L}}_t(h_{t-1})$ where the expectation is taken over the randomness used in buffer construction. However, a property such as $\mathbb{E}\left[\!\left[\hat{\mathcal{L}}_t^{\text{buf}}(h)\right]\!\right] = \hat{\mathcal{L}}_t(h)$ holds only for functions h that are either fixed or obtained independently of the random variables used in buffer updates (over which the expectation is taken). Since h_{t-1} is learned from points in the buffer itself, the above property, and consequently the regret bound, does not hold.

We remedy this issue by showing a relatively weaker claim; we show that with high probability we have $\hat{\mathcal{L}}_t(h_{t-1}) \leq \hat{\mathcal{L}}_t^{\text{buf}}(h_{t-1}) + \epsilon$. At a high level, this claim is similar to showing uniform convergence bounds for $\hat{\mathcal{L}}_t^{\text{buf}}$. However, the reservoir sampling algorithm is not particularly well suited to prove such uniform conver-



(a) Sonar (b) Segment

(c) IJCNN (d) Covertype

Figure 1. Performance of **OLP** (using **RS-x**) and OAM_{gra} (using **RS**) by (Zhao et al., 2011) on AUC maximization tasks with varying buffer sizes.

gence bounds as it essentially performs sampling without replacement (see Appendix G for a discussion). We overcome this hurdle by proposing a new buffer update policy **RS-x** (see Algorithm 1) that, at each time step, guarantees s i.i.d. samples from the preceding stream (see Appendix H for a proof).

Our algorithm uses this buffer update policy in conjunction with an online learning algorithm **OLP** (see Algorithm 2) that is a variant of the well-known GIGA algorithm (Zinkevich, 2003). We provide the following *all-pairs* regret guarantee for our algorithm:

Theorem 8. *Suppose the **OLP** algorithm working with an s-sized buffer generates an ensemble $\mathbf{w}_1, \ldots, \mathbf{w}_{n-1}$. Then with probability at least $1-\delta$,*

$$\frac{\mathfrak{R}_n}{n-1} \leq \mathcal{O}\left(C_d\sqrt{\frac{\log\frac{n}{\delta}}{s}} + \sqrt{\frac{1}{n-1}}\right)$$

See Appendix I for the proof. A drawback of our bound is that it offers sublinear regret only for buffer sizes $s = \omega(\log n)$. A better regret bound for constant s or a lower-bound on the regret is an open problem.

8. Experimental Evaluation

In this section we present experimental evaluation of our proposed **OLP** algorithm. We stress that the aim of this evaluation is to show that our algorithm, that enjoys high confidence regret bounds, also performs competitively in practice with respect to the OAM_{gra} algorithm proposed by Zhao et al. (2011) since our results in Section 5 show that OAM_{gra} does enjoy good generalization guarantees despite the lack of an *all-pairs* regret bound.

In our experiments, we adapted the **OLP** algorithm to the AUC maximization problem and compared it with OAM_{gra} on 18 different benchmark datasets. We used 60% of the available data points up to a maximum of 20000 points to train both algorithms. We refer the reader to Appendix J for a discussion on the implementation of the **RS-x** algorithm. Figure 1 presents the results of our experiments on 4 datasets across 5 random training/test splits. Results on other datasets can be found in Appendix K. The results demonstrate that **OLP** performs competitively to OAM_{gra} while in some cases having slightly better performance for small buffer sizes.

9. Conclusion

In this paper we studied the generalization capabilities of online learning algorithms for pairwise loss functions from several different perspectives. Using the method of *Symmetrization of Expectations*, we first provided sharp online to batch conversion bounds for algorithms that offer *all-pairs* regret bounds. Our results for bounded and strongly convex loss functions closely match their first order counterparts. We also extended our analysis to algorithms that are only able to provide *finite-buffer* regret bounds using which we were able to explain the good empirical performance of some existing algorithms. Finally we presented a new memory-efficient online learning algorithm that is able to provide *all-pairs* regret bounds in addition to performing well empirically.

Several interesting directions can be pursued for future work, foremost being the development of online learning algorithms that can guarantee sub-linear regret at constant buffer sizes or else a regret lower bound for finite buffer algorithms. Secondly, the idea of a *stream-aware* buffer update policy is especially interesting both from an empirical as well as theoretical point of view and would possibly require novel proof techniques for its analysis. Lastly, scalability issues that arise when working with higher order loss functions also pose an interesting challenge.

Acknowledgment

The authors thank the anonymous referees for comments that improved the presentation of the paper. PK is supported by the Microsoft Corporation and Microsoft Research India under a Microsoft Research India Ph.D. fellowship award.

References

Agarwal, Shivani and Niyogi, Partha. Generalization Bounds for Ranking Algorithms via Algorithmic Stability. *JMLR*, 10:441–474, 2009.

Balcan, Maria-Florina and Blum, Avrim. On a Theory of Learning with Similarity Functions. In *ICML*, pp. 73–80, 2006.

Bellet, Aurélien, Habrard, Amaury, and Sebban, Marc. Similarity Learning for Provably Accurate Sparse Linear Classification. In *ICML*, 2012.

Brefeld, Ulf and Scheffer, Tobias. AUC Maximizing Support Vector Learning. In *ICML workshop on ROC Analysis in Machine Learning*, 2005.

Cao, Qiong, Guo, Zheng-Chu, and Ying, Yiming. Generalization Bounds for Metric and Similarity Learning, 2012. arXiv:1207.5437.

Cesa-Bianchi, Nicoló and Gentile, Claudio. Improved Risk Tail Bounds for On-Line Algorithms. *IEEE Trans. on Inf. Theory*, 54(1):286–390, 2008.

Cesa-Bianchi, Nicoló, Conconi, Alex, and Gentile, Claudio. On the Generalization Ability of On-Line Learning Algorithms. In *NIPS*, pp. 359–366, 2001.

Clémençon, Stéphan, Lugosi, Gábor, and Vayatis, Nicolas. Ranking and empirical minimization of U-statistics. *Annals of Statistics*, 36:844–874, 2008.

Cortes, Corinna, Mohri, Mehryar, and Rostamizadeh, Afshin. Generalization Bounds for Learning Kernels. In *ICML*, pp. 247–254, 2010a.

Cortes, Corinna, Mohri, Mehryar, and Rostamizadeh, Afshin. Two-Stage Learning Kernel Algorithms. In *ICML*, pp. 239–246, 2010b.

Cristianini, Nello, Shawe-Taylor, John, Elisseeff, André, and Kandola, Jaz S. On Kernel-Target Alignment. In *NIPS*, pp. 367–373, 2001.

Freedman, David A. On Tail Probabilities for Martingales. *Annals of Probability*, 3(1):100–118, 1975.

Hazan, Elad, Kalai, Adam, Kale, Satyen, and Agarwal, Amit. Logarithmic Regret Algorithms for Online Convex Optimization. In *COLT*, pp. 499–513, 2006.

Jin, Rong, Wang, Shijun, and Zhou, Yang. Regularized Distance Metric Learning: Theory and Algorithm. In *NIPS*, pp. 862–870, 2009.

Kakade, Sham M. and Tewari, Ambuj. On the Generalization Ability of Online Strongly Convex Programming Algorithms. In *NIPS*, pp. 801–808, 2008.

Kakade, Sham M., Sridharan, Karthik, and Tewari, Ambuj. On the Complexity of Linear Prediction: Risk Bounds, Margin Bounds, and Regularization. In *NIPS*, 2008.

Kakade, Sham M., Shalev-Shwartz, Shai, and Tewari, Ambuj. Regularization Techniques for Learning with Matrices. *JMLR*, 13:1865–1890, 2012.

Kumar, Abhishek, Niculescu-Mizil, Alexandru, Kavukcuoglu, Koray, and III, Hal Daumé. A Binary Classification Framework for Two-Stage Multiple Kernel Learning. In *ICML*, 2012.

Ledoux, Michel and Talagrand, Michel. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, 2002.

Sridharan, Karthik, Shalev-Shwartz, Shai, and Srebro, Nathan. Fast Rates for Regularized Objectives. In *NIPS*, pp. 1545–1552, 2008.

Steinwart, Ingo and Christmann, Andreas. *Support Vector Machines*. Information Science and Statistics. Springer, 2008.

Vitter, Jeffrey Scott. Random Sampling with a Reservoir. *ACM Trans. on Math. Soft.*, 11(1):37–57, 1985.

Wang, Yuyang, Khardon, Roni, Pechyony, Dmitry, and Jones, Rosie. Generalization Bounds for Online Learning Algorithms with Pairwise Loss Functions. *JMLR - Proceedings Track*, 23:13.1–13.22, 2012.

Wang, Yuyang, Khardon, Roni, Pechyony, Dmitry, and Jones, Rosie. Online Learning with Pairwise Loss Functions, 2013. arXiv:1301.5332.

Xing, Eric P., Ng, Andrew Y., Jordan, Michael I., and Russell, Stuart J. Distance Metric Learning with Application to Clustering with Side-Information. In *NIPS*, pp. 505–512, 2002.

Zhao, Peilin, Hoi, Steven C. H., Jin, Rong, and Yang, Tianbao. Online AUC Maximization. In *ICML*, pp. 233–240, 2011.

Zinkevich, Martin. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *ICML*, pp. 928–936, 2003.