Efficient Online Learning via Randomized Rounding

Authored by:

Nicol? Cesa-bianchi
Ohad Shamir

Abstract

Most online algorithms used in machine learning today are based on variants of mirror descent or follow-the-leader. In this paper, we present an online algorithm based on a completely different approach, which combines "random playout" and randomized rounding of loss subgradients. As an application of our approach, we provide the first computationally efficient online algorithm for collaborative filtering with trace-norm constrained matrices. As a second application, we solve an open question linking batch learning and transductive online learning.

1 Paper Body

Online learning algorithms, which have received much attention in recent years, enjoy an attractive combination of computational efficiency, lack of distributional assumptions, and strong theoretical guarantees. However, it is probably fair to say that at their core, most of these algorithms are based on the same small set of fundamental techniques, in particular mirror descent and regularized follow-the-leader (see for instance [14]). In this work we revisit, and significantly extend, an algorithm which uses a completely different approach. This algorithm, known as the Minimax Forecaster, was introduced in [9, 11] for the setting of prediction with static experts. It computes minimax predictions in the case of known horizon, binary outcomes, and absolute loss. Although the original version is computationally expensive, it can easily be made efficient through randomization. We extend the analysis of [9] to the case of non-binary outcomes and arbitrary convex and Lipschitz loss functions. The new algorithm is based on a combination of ?random playout? and randomized rounding, which assigns random binary labels to future unseen instances, in a way depending on the loss subgradients. Our resulting Randomized Rounding (R2) Forecaster has a parameter trading off regret performance and computational complexity, and runs in polynomial time (for T predictions, it requires computing O(T 2) empirical risk minimizers in general, as opposed to O(T) for generic follow-the-leader algorithms). The regret of the R2 Forecaster is determined by the Rademacher complexity of the comparison class. The connection between online learnability and Rademacher complexity has also been explored in [2, 1]. However, these works focus on the information-theoretically achievable regret, as opposed to computationally efficient algorithms. The idea of ?random playout?, in the context of online learning, has also been used in [16, 3], but we apply this idea in a different way. We show that the R2 Forecaster can be used to design the first efficient online learning algorithm for collaborative filtering with trace-norm constrained matrices. While this is a well-known setting, a straightforward application of standard online learning approaches, such as mirror descent, appear to give only trivial performance guarantees. Moreover, our 1

regret bound matches the best currently known sample complexity bound in the batch distribution-free setting [21]. As a different application, we consider the relationship between batch learning and transductive online learning. This relationship was analyzed in [16], in the context of binary prediction with respect to classes of bounded VC dimension. Their main result was that efficient learning in a statistical setting implies efficient learning in the transductive online setting, but at an inferior rate of T 3/4 (where T is the number of rounds). The main open question posed by that paper is whether a better rate can be obtained. Using the R2 Fore? caster, we improve on those results, and provide an efficient algorithm with the optimal T rate, for a wide class of losses. This shows that efficient batch learning not only implies efficient transductive online learning (the main thesis of [16]), but also that the same rates can be obtained, and for possibly non-binary prediction problems as well. We emphasize that the R2 Forecaster requires computing many empirical risk minimizers (ERM?s) at each round, which might be prohibitive in practice. Thus, while it does run in polynomial time whenever an ERM can be efficiently computed, we make no claim that it is a ?fully practical? algorithm. Nevertheless, it seems to be a useful tool in showing that efficient online learnability is possible in various settings, often working in cases where more standard techniques appear to fail. Moreover, we hope the techniques we employ might prove useful in deriving practical online algorithms in other contexts.

2

The Minimax Forecaster

We start by introducing the sequential game of prediction with expert advice ?see [10]. The game is played between a forecaster and an adversary, and is specified by an outcome space Y, a prediction space P, a nonnegative loss function ‘ : P ? Y ? R, which measures the discrepancy between the forecaster?s prediction and the outcome, and an expert class F. Here we focus on classes F of static experts, whose prediction at each round t does not depend on the outcome in previous rounds. Therefore, we think of each f ? F simply as a sequence f = (f1 , f2 , . . .) where each ft ? P. At each step t = 1, 2, . . . of the game, the forecaster outputs a prediction pt ? P and simultaneously the adversary reveals an outcome yt ? Y. The forecaster?s goal is to predict the outcome sequence almost as well as the best expert in the class F, irrespective of the outcome sequence y = (y1 , y2 , . . .). The performance of a forecasting strategy A is measured by the worst-case regret ! T T X X VT (A, F) = sup ‘(pt , yt) ? inf ‘(ft , yt) (1) y?Y T

f ?F

t=1

t=1

viewed as a function of the horizon T . To simplify notation, let L(f , y) =

PT

t=1

‘(ft , yt).

Consider now the special case where the horizon T is fixed and known in advance, the outcome space is Y = {?1, +1}, the prediction space is P = [?1, +1], and the loss is the absolute loss ‘(p, y) = —p ? y—. We will denote the regret in this special case as VTabs (A, F). The Minimax Forecaster ?which is based on work presented in [9] and [11], see also [10] for an exposition? is derived by an explicit analysis of the minimax regret inf A VTabs (A, F), where the infimum is over all forecasters A producing at round t a prediction pt as a function of p1 , y1 , . . . pt?1 , yt?1 . For general online learning problems, the analysis of this quantity is intractable. However, for the specific setting we focus on (absolute loss and binary outcomes), one can get both an explicit expression for the minimax regret, as well as an PT explicit algorithm, provided inf f ?F t=1 ‘(ft , yt) can be efficiently computed for any sequence y1 , . . . , yT . This procedure is akin to performing empirical risk minimization (ERM) in statistical learning. A full development of the analysis is out of scope, but is outlined in Appendix A of the supplementary material. In a nutshell, the idea is to begin by calculating the optimal prediction in the last round T , and then work backwards, calculating the optimal prediction at round T ? 1, T ? 2 etc. Remarkably, the value of inf A VTabs (A, F) is exactly the Rademacher complexity RT (F) of the class F, which is known to play a crucial role in understanding the sample complexity in statistical learning [5]. In this paper, we 2

define it as1 : ” RT (F) = E sup

T X

?t ft

(2)

f ?F t=1

where ?1 , . . . , ?T are i.i.d. Rademacher random variables, taking values ?1, +1 with equal probability. When RT (F) = o(T), we get a minimax regret inf A VTabs (A, F) = o(T) which implies a vanishing per-round regret. In terms of an explicit algorithm, the optimal prediction pt at round t is given by a complicated-looking recursive expression, involving exponentially many terms. Indeed, for general online learning problems, this is the most one seems able to hope for. However, an apparently little-known fact is that when one deals with a class F of fixed binary sequences as discussed above, then one can write the optimal prediction pt in a much simpler way. Letting Y1 , . . . , YT be i.i.d. Rademacher random variables, the optimal prediction at round t can be written as pt = E inf L (f , y1 ? ? ? yt?1 (?1) Yt+1 ? ? ? YT) ? inf L (f , y1 ? ? ? yt?1 1 Yt+1 ? ? ? YT) . (3) f ?F

f ?F

In words, the prediction is simply the expected difference between the minimal cumulative loss over F, when the adversary plays ?1 at round t and random values afterwards, and the minimal cumulative loss over F, when the adversary plays +1 at round t, and the same random values afterwards. We refer the reader to Appendix A of the supplementary material for how this is derived. We denote this optimal strategy (for absolute loss and binary outcomes) as the Minimax Forecaster (mf): Algorithm 1 Minimax Forecaster (mf) for t = 1 to T do Predict pt as defined in Eq. (3) Receive outcome yt and suffer loss —pt ? yt — end for The relevant guarantee for mf is summarized in the following theorem. Theorem 1. For any class F ? [?1, +1]T of static experts, the regret of the Minimax Forecaster (Algorithm 1) satisfies VTabs (mf, F) = RT (F). 2.1

Making the Minimax Forecaster Efficient

The Minimax Forecaster described above is not computationally efficient, as the computation of pt requires averaging over exponentially many ERM?s. However, by a martingale argument, it is not hard to show that it is in fact sufficient to compute only two ERM?s per round. Algorithm 2 Minimax Forecaster with efficient implementation (mf*) for t = 1 to T do For i = t + 1, . . . , T , let Yi be a Rademacher random variable Let pt := inf f ?F L (f , y1 . . . yt?1 (?1) Yt+1 . . . YT) ? inf f ?F L (f , y1 . . . yt?1 1 Yt+1 . . . YT) Predict pt , receive outcome yt and suffer loss —pt ? yt — end for Theorem 2. For any class F ? [?1, +1]T of static experts, the regret of the randomized forecasting strategy mf* (Algorithm 2) satisfies p VTabs (mf*, F) ? RT (F) + 2T ln(1/?) 1 In the statistical learning literature, it is more common to scale this quantity by 1/T , but the form we use here is more convenient for stating cumulative regret bounds.

3

with probability at least 1 ? ?. Moreover, if the predictions p = (p1 , . . . , pT) are computed reusing the random values Y1 , . . . , YT computed at the first iteration of the algorithm, rather than drawing fresh values at each iteration, then it holds that

E L(p, y) ? inf L(f , y) ? RT (F) for all y ? {?1, +1}T . f ?F

Proof sketch. To prove the second statement, note that E[pt]?yt = E —pt ?yt — for any fixed yt ? {?1, +1} and pt bounded in [?1, +1], and use Thm. 1. To prove the first statement, note that —pt ? yt — ? Ept [pt] ? yt for t = 1, . . . , T is a martingale difference sequence with respect to p1 , . . . , pT , and apply Azuma?s inequality. The second statement in the theorem bounds the regret only in expectation and is thus weaker than the first one. On the other hand, it might have algorithmic benefits. Indeed, if we reuse the same values for Y1 , . . . , YT , then the computation of the infima over f in mf* are with respect to an outcome sequence which changes only at one point in each round. Depending on the specific learning problem, it might be easier to re-compute the infimum after changing a single point in the outcome sequence, as opposed to computing the infimum over a different outcome sequence in each round.

3

The R2 Forecaster

The Minimax Forecaster presented above is very specific to the absolute loss ‘(f, y) = —f ? y— and for binary outcomes Y = {?1, +1}, which limits its applicability. We note that extending the forecaster to other losses or different outcome spaces is not trivial: indeed, the recursive unwinding of the minimax regret term, leading to an explicit expression and an explicit algorithm, does not work as-is for other cases. Nevertheless, we will now show how one can deal with general (convex, Lipschitz) loss functions and outcomes belonging to any real interval [?b, b]. The algorithm we propose essentially uses the Minimax Forecaster as a subroutine, by feeding it with a carefully chosen sequence of binary values zt , and using predictions ft which are scaled to lie in the interval [?1, +1]. The values of zt are based on a randomized rounding of values in [?1, +1], which depend in turn on the loss subgradient. Thus, we denote the algorithm as the Randomized Rounding (R2) Forecaster. To describe the algorithm, we introduce some notation. For any scalar f ? [?b, b], define fe = f /b to be the scaled versions of f into the range [?1, +1]. For vectors f , define e f = (1/b)f . Also, we let ?pt ‘(pt , yt) denote any subgradient of the loss function ‘ with respect to the prediction pt . The pseudocode of the R2 Forecaster is presented as Algorithm 3 below, and its regret guarantee is summarized in Thm. 3. The proof is presented in Appendix B of the supplementary material. Theorem 3. Suppose ‘ is convex and ?-Lipschitz in its first argument. For any F ? [?b, b]T the regret of the R2 Forecaster (Algorithm 3) satisfies r s

1 2T 2 VT (R , F) ? ? RT (F) + ? b +2 2T ln (4) ? ? with probability at least 1 ? ?. The prediction pt which the algorithm computes is an empirical approximation to

e e b EYt+1 ,...,YT inf L f , z1 . . . zt?1 0 Yt+1 . . . YT ? inf L f , z1 ? ? ? zt?1 1 Yt+1 ? ? ? YT f ?F

f ?F

by repeatedly drawing independent values to Yt+1 , . . . , YT and averaging. The accuracy of the approximation is reflected in the precision parameter ?. A larger value of ? improves the regret bound, but also increases the runtime of the algorithm. Thus, ? provides a trade-off between the computational complexity of the algorithm and its regret guarantee. We note 4

Algorithm 3 The R2 Forecaster Input: Upper bound b on —ft —, —yt — for all t = 1, . . . , T and f ? F; upper bound ? on supp,y?[?b,b] ?p ‘(p, y); precision parameter ? ? T1 . for t = 1 to T do pt := 0 for j = 1 to ? T do For i = t, . . . , T , let Yi be a Rademacher random variable

e Draw ? := inf L f , z1 . . . zt?1 (?1) Yt+1 . . . YT ? inf L e f , z1 . . . zt?1 1 Yt+1 . . . YT f ?F

f ?F

Let pt := pt + ?bT ? end for Predict pt Receive outcome yt and suffer loss ‘(pt , yt) Let rt := 21 1 ? ?1 ?pt ‘(pt , yt) ? [0, 1] Let zt := 1 with probability rt , and zt := ?1 with probability 1 ? rt end for

that even when ? is taken to be a constant fraction, the resulting algorithm still runs in polynomial time O(T 2 c), where c is the time to compute a single

ERM. In subsequent results pertaining to this Forecaster, we will assume that ? is taken to be a constant fraction. We end this section with a remark that plays an important role in what follows. Remark 1. The predictions of our forecasting strategies do not depend on the ordering of the predictions of the experts in F. In other words, all the results proven so far also hold in a setting where the elements of F are functions f : {1, . . . , T } ? P, and the adversary has control on the permutation ?1 , . . . , ?T of {1, . . . , T } that is used to define the prediction f (?t) of expert f at time t.2 Also, Thm. 1 implies that the value of VTabs (F) remains unchanged irrespective of the permutation chosen by the adversary.

4

Application 1: Transductive Online Learning

The first application we consider is a rather straightforward one, in the context of transductive online learning [6]. In this model, we have an arbitrary sequence of labeled examples (x1 , y1), . . . , (xT , yT), where only the set {x1 , . . . , xT } of unlabeled instances is known to the learner in advance. At each round t, the learner must provide a prediction pt for the label of yt . The true label yt is then revealed, and the learner incurs a loss ‘(pt , yt). The learner?s

PT goal is to minimize the transductive online regret t=1 ‘(pt , yt) ? inf f ?F ‘(f (xt), yt) with respect to a fixed class of predictors F of the form {x 7? f (x)}. The work [16] considers the binary classification case with zero-one loss. Their main result is that if a class F of binary functions has bounded VC dimension d, and there exists an efficient algorithm to perform empirical risk minimization, then one can construct an efficientprandomized algorithm for transductive online learning, whose regret is at most O(T 3/4 d ln(T)) in expectation. The significance of this result is that efficient batch learning (via empirical risk minimization) implies efficient learning in the transductive online setting. This is an important result, as online learning can be computationally harder than batch learning ?see, e.g., [8] for an example in the context of Boolean learning. ? A major open question posed by [16] was whether one can achieve the optimal rate O(dT), matching the rate of a batch learning algorithm in the statistical setting. Using the R2 Forecaster, we can easily achieve the above result, as well as similar results in a strictly more general setting. This shows that efficient batch learning not only implies efficient transductive online learning (the main thesis of [16]), but also that the same rates can be obtained, and for possibly non-binary prediction problems as well. 2 Formally, at each step t: (1) the adversary chooses and reveals the next element ?t of the permutation; (2) the forecaster chooses pt ? P and simultaneously the adversary chooses yt ? Y.

5

Theorem 4. Suppose we have a computationally efficient algorithm for empirical risk minimization (with respect to the zero-one loss) over a class F of {0, 1}-valued functions with VC dimension d. Then, in the transductive online model, the efficient randomized forecaster ? mf* achieves an expected regret of O(dT) with respect to the zero-one loss. Moreover, for an arbitrary class F of

[?b, b]-valued functions with Rademacher complexity RT (F), and any convex ?-Lipschitz loss function, if there exists a computationally efficient algorithm for empirical risk minimization, then the R2 Forecaster is computationally effip cient and achieves, in the transductive online model, a regret of ?RT (F)+O(?b T ln(T /?)) with probability at least 1 ? ?. Proof. Since the set {x1 , . . . , xT } of unlabeled examples is known, we reduce the online transductive model to prediction with expert advice in the setting of Remark 1. This is done by mapping each function f ? F to a function f : {1, . . . , T } ? P by t 7? f (xt), which is equivalent to an expert in the setting of Remarks 1. When F maps to {0, 1}, and we care about the zero-one loss, we can use the forecaster mf* to compute randomized predictions and apply Thm. 2 to bound the expected ? transductive online regret with RT (F). For a class with VC dimension d, RT (F) ? O(dT) for some constant c ¿ 0, using Dudley?s chaining method [12], and this concludes the proof of the first part of the theorem. The second part is an immediate corollary of Thm. 3. We close this section by contrasting our results for online transductive learning with those of [7] about standard online learning. If F contains {0, 1}-valued functions, then the optimal ? regret bound for online learning is order of d0 T , where d0 is the Littlestone dimension of F. Since the Littlestone dimension of a class is never smaller than its VC dimension, we conclude that online learning is a harder setting than online transductive learning.

5

Application 2: Online Collaborative Filtering

We now turn to discuss the application of our results in the context of collaborative filtering with trace-norm constrained matrices, presenting what is (to the best of our knowledge) the first computationally efficient online algorithms for this problem. In collaborative filtering, the learning problem is to predict entries of an unknown m ? n matrix based on a subset of its observed entries. A common approach is norm regularization, where we seek a low-norm matrix which matches the observed entries as best as possible. The norm is often taken to be the trace-norm [22, 19, 4], although other norms have also been considered, such as the max-norm [18] and the weighted trace-norm [20, 13]. Previous theoretical treatments of this problem assumed a stochastic setting, where the observed entries are picked according to some underlying distribution (e.g., [23, 21]). However, even when the guarantees are distribution-free, assuming a fixed distribution fails to capture important aspects of collaborative filtering in practice, such as non-stationarity [17]. Thus, an online adversarial setting, where no distributional assumptions whatsoever are required, seems to be particularly well-suited to this problem domain. In an online setting, at each round t the adversary reveals an index pair (it , jt) and secretely chooses a value yt for the corresponding matrix entry. After that, the learner selects a prediction pt for that entry. Then yt is revealed and the learner suffers a loss ‘(pt , yt). Hence, the goal of a learner is to minimize the regret with respect to a fixed class W

PT PT of prediction matrices, t=1 ‘(pt , yt) ? inf W ?W t=1 ‘ Wit ,jt , yt . Following reality, we will assume that the adversary picks a different entry in each round. When the learner?s performance is measured by the regret after all T = mn entries have been predicted, the online collaborative filtering setting reduces to prediction with expert advice as discussed in Remark 1. As mentioned previously, W is often taken to be a convex class of matrices with bounded trace-norm. Many convex learning problems, such as linear and kernel-based predictors, as well as matrix-based predictors, can be learned efficiently both in a stochastic and an online setting, using mirror descent or regularized follow-the-leader methods. However, 6

for reasonable choices of W, a straightforward application of these techniques can lead to algorithms with trivial bounds. In particular, in the case of W consisting of m?? n matrices with trace-norm at most r, standard online regret bounds would scale like O r T . ? Since for this norm one typically has r = O mn , we get a per-round regret guarantee p of O(mn/T). This is a trivial bound, since it becomes ?meaningful? (smaller than a constant) only after all T = mn entries have been predicted. On the other hand, based on general techniques developed in [15] and greatly extended in [1], it can be shown that online learnability is information-theoretically possible for such W. However, these techniques do not provide a computationally efficient algorithm. Thus, to the best of our knowledge, there is currently no efficient (polynomial time) online algorithm, which attain non-trivial regret. In this section, we show how to obtain such an algorithm using the R2 Forecaster. Consider first the transductive online setting, where the set of indices to be predicted is known in advance, and the adversary may only choose the order and values of the entries. It is readily seen that the R2 Forecaster can be applied in this setting, using any convex class W of fixed matrices with bounded entries to compete against, and any convex Lipschitz loss function. To do so, we let {ik , jk }Tk=1 be the set of entries, and run the R2 Forecaster with respect to F = {t 7? Wit ,jt : W ? W}, which corresponds to a class of experts as discussed in Remark 1. What is perhaps more surprising is that the R2 Forecaster can also be applied in a nontransductive setting, where the indices to be predicted are not known in advance. Moreover, the Forecaster doesn?t even need to know the horizon T in advance. The key idea to achieve this is to utilize the non-asymptotic nature of the learning problem ?namely, that the game is played over a finite m ? n matrix, so the time horizon is necessarily bounded. The algorithm we propose is very simple: we apply the R2 Forecaster as if we are in a setting with time horizon T = mn, which is played over all entries of the m ? n matrix. By Remark 1, the R2 Forecaster does not need to know the order in which these m ? n entries are going to be revealed. Whenever W is convex and ‘ is a convex function, we can find an ERM in polynomial time by solving a convex problem. Hence, we can implement the R2 Forecaster efficiently. To show that this is indeed a viable strategy, we need the following lemma, whose proof is presented in Appendix C of the supplementary material. Lemma 1. Consider a (possibly randomized) forecaster A for a class F whose regret after T steps satisfies VT (A, F) ? G with probability at least 1 ? ? ¿ 21 . Furthermore, suppose the loss function is such that inf sup inf ‘(p, y) ? ‘(p0 , y) ? 0. Then 0 p ?P y?Y p?P

max Vt (A, F) ? G

t=1,...,T

with probability at least 1 ? ?.

Note that a simple sufficient condition for the assumption on the loss function to hold, is that P = Y and ‘(p, y) ? ‘(y, y) for all p, y ? P. Using this lemma, the following theorem exemplifies how we can obtain a regret guarantee for our algorithm, in the case of W consisting of the convex set of matrices with bounded trace-norm and bounded entries. For the sake of clarity, we will consider n ? n matrices. Theorem 5. Let ‘ be a loss function which satisfies the conditions of Lemma 1. Also, let W consist of n ? n matrices with trace-norm at most r = O(n) and entries at most b = O(1), suppose we apply the R2 Forecaster over time horizon n2 and all entries of the matrix. Then with probability at least 1 ? ?, after T rounds, the algorithm achieves an average per-round regret of at most ! p n3/2 + n ln(n/?) O uniformly over T = 1, . . . , n2 . T Proof. In our setting, where the adversary chooses a different entry at each round, [21, Theorem 6] implies that for the class W 0 of all matrices with trace-norm at most r = O(n), 7

it holds that RT (W 0)/T ? O(n3/2 /T). Therefore, Rn2 (W 0) ? O(n3/2). Since W ? W 0 , 3/2 we get by definition of the Rademacher complexity p that Rn2 (W) = O(n) as well. By 2 3/2 Thm. 3, the regret after n rounds is O(n + n ln(n/?)) with probability at least 1 ? ?. Applying Lemma 1, wepget that the cumulative regret at the end of any round T = 1, . . . , n2 is at most O(n3/2 + n ln(n/?)), as required. This bound becomes non-trivial after n3/2 entries are revealed, which is still a vanishing proportion of all n2 entries. While the regret ? might seem unusual compared to standard regret bounds (which usually have rates of 1/ T for general losses), it is a natural outcome of the non-asymptotic nature of our setting, where T can never be larger than n2 . In fact, this is the same rate one would obtain in a batch setting, where the entries are drawn from an arbitrary distribution. Moreover, an assumption such as boundedness of the entries is required for currently-known guarantees even in a batch setting ?see [21] for details. Acknowledgments The first author acknowledges partial support by the PASCAL2 NoE under EC grant FP7216886.

2 References

[1] K. Sridharan A. Rakhlin and A. Tewari. Online learning: Random averages, combinatorial parameters, and learnability. In NIPS, 2010. [2] J. Abernethy, P. Bartlett, A. Rakhlin, and A. Tewari. Optimal strategies and minimax lower bounds for online convex games. In COLT, 2009. [3] J. Abernethy and M. Warmuth. Repeated games against budgeted adversaries. In NIPS, 2010. [4] F. Bach. Consistency of trace-norm minimization. Journal of Machine Learning Research, 9:1019?1048, 2008. [5] P. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. In COLT, 2001. [6] S. Ben-David, E. Kushilevitz, and Y. Mansour. Online learning versus offline learning. Machine Learning, 29(1):45?63, 1997. [7] S. Ben-David, D. P? al, and S. Shalev-Shwartz. Agnostic online learning. In COLT, 2009. [8] A. Blum. Separating distribution-free and mistake-bound learning models over

the boolean domain. SIAM J. Comput., 23(5):990?1000, 1994. [9] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. Helmbold, R. Schapire, and M. Warmuth. How to use expert advice. Journal of the ACM, 44(3):427?485, May 1997. [10] N. Cesa-Bianchi and G. Lugosi. Prediction, learning, and games. Cambridge University Press, 2006. [11] T. Chung. Approximate methods for sequential decision making using expert advice. In COLT, 1994. ? [12] R. M. Dudley. A Course on Empirical Processes, Ecole de Probabilit?es de St. Flour, 1982, volume 1097 of Lecture Notes in Mathematics. Springer Verlag, 1984. [13] R. Foygel, R. Salakhutdinov, O. Shamir, and N. Srebro. Learning with the weighted trace-norm under arbitrary sampling distributions. In NIPS, 2011. [14] E. Hazan. The convex optimization approach to regret minimization. In S. Nowozin S. Sra and S. Wright, editors, Optimization for Machine Learning. MIT Press, To Appear. [15] P. Bartlett J. Abernethy, A. Agarwal and A. Rakhlin. A stochastic view of optimal regret through minimax duality. In COLT, 2009. [16] S. Kakade and A. Kalai. From batch to transductive online learning. In NIPS, 2005. 8

[17] Y. Koren. Collaborative filtering with temporal dynamics. In KDD, 2009. [18] J. Lee, B. Recht, R. Salakhutdinov, N. Srebro, and J. Tropp. Practical large-scale optimization for max-norm regularization. In NIPS, 2010. [19] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In NIPS, 2007. [20] R. Salakhutdinov and N. Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In NIPS, 2010. [21] O. Shamir and S. Shalev-Shwartz. Collaborative filtering with the trace norm: Learning, bounding, and transducing. In COLT, 2011. [22] N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In NIPS, 2004. [23] N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In COLT, 2005.

9