Collaborative Filtering via Online Mirror Descent

Reza Takapoui
Department of Electrical Engineering
Stanford University
takapoui@stanford.edu

Abstract

In this report, we will study online learning algorithms, and in particular, online mirror descent (OMD) method when applied to the collaborative filtering problem. This is motivated by the problem of real-world large-scale recommendation systems, where the goal is to make relevant recommendations to the users based on their demographic information, their past behavior, and the other users' bahevior. In order to analyze regret bounds of OMD on this problem, we need to equip ourselves with tools from convexity analysis for matrices. We will compare our results to the baseline result and observe a substantial improvement in the regret function.

1 Introduction

The matrix completion paradigm discusses reconstructing a low rank matrix from a few uniformly sampled entries and mainly focuses on minimizing the Frobenius norm error of the matrix reconstruction. This paradigm received much attention as a solution to the collaborative filtering problem in recommendation systems (e.g. Netflix Prize), where the user-by-item preference matrix can be well modeled as a low rank matrix [KOM09, CR08]. A parallel literature on linearly parametrized bandits emphasizes on implicit exploration/exploitation tradeoff between exploring the user's true preferences and exploiting the knowledge already acquired to make better decisions. The main focus here, is to propose adaptive sampling techniques to minimize the incurred regret [RT10, DM12].

In our CS229 project [AAT13], we highlighted the need to reconcile these two frameworks to solve problems in real-world large-scale recommendation systems. We empirically compared the performance of Upper Confidence Bounds (UCB) [ACBF02] and posterior sampling (also known as Thompson sampling) [Tho33] with the simplifying assumption that the latent parameters of items were available. However, implementing posterior sampling in general case can be challenging due to the intractability of the distribution and complexity of the sampling scheme.

The preference matrix of a recommendation system is known to be well-modeled by a low rank matrix. In this report, we study online learning algorithms, and specifically online mirror descent (OMD) and tailor these algorithms to best adapt to our prior knowledge on the rewards. In order to run OMD, we need to discuss different regularization functions and study their behavior in a theoretical framework.

The rest of this paper is organized as the following: in section 2, we will study a general model for the problems and the methods to tackle the problem. We will also explain the motivation to study this problem and the reason better results are expected. In section 3, we formalize the problem into an online learning problem and use different OMD methods to find different regret bounds. Section 4 includes some discussion on the our intuitive understanding of the problem and also future work. Some technical proofs are included in the Appendix.

2 Preliminaries

2.1 A general model

Consider a recommendation system with n users and m items. Define the user-item preference matrix $M \in \mathbf{R}^{n\times m}$ where M_{ij} is the rating that user i would give to item j. We do not know all entries of matrix M, but up to time $t = 0, 1, \cdots$, we have sampled entries in $\Omega(t) \subseteq [n] \times [m]$. At time t, we can sample any pair $(i, j) \in S(t)$, probably with an additive noise, (for example, you can assume $S(t)$ is the associated row to a user who is waiting for a recommendation). The essential idea here is that M can be well modeled by a low rank matrix, and this assumption may be used to impute unobserved entries. More explicitly, if M is rank k, then it can be written as $M = A^T B$ where $A \in \mathbf{R}^{k\times n}$ and $B \in \mathbf{R}^{k\times m}$.

There are different risk metrics that we can define for this problem. One popular risk metric is the Frobenius norm of the error, which has been extensively studied in the matrix completion paradigm. The goal, in this case, is to sample adaptively to minimize

$$\text{minimize} \quad \textstyle\sum_{(ij)\in\Omega(t)}(x_{ij} - r_{ij})^2 + \gamma\,\mathbf{Rank}(X)$$

which can be relaxed to the following convex problem:

$$\text{minimize} \quad \textstyle\sum_{(ij)\in\Omega(t)}(x_{ij} - r_{ij})^2 + \gamma\|X\|_*.$$

Here $\|X\|_*$ denotes the trace norm of X and is the convex envelope of the rank (i.e. the largest lower bounding convex function). This framework focuses on minimizing the Frobenius error via adaptive sampling, however, a better measure for the risk here might be *expected Regret* which is defined as

$$R(T) = \mathbf{E}\left[\sum_{t=1}^{T} M_{i^\star(t)j^\star(t)} - M_{i(t)j(t)}\right]. \tag{1}$$

where $(i(t), j(t)) \in S(t)$, and $(i^\star(t), j^\star(t)) = \operatorname{argmax}_{(ij)\in S(t)} M_{ij}$.

2.2 Methods

One popular method to tackle this problem is *linear bandits*, which takes the item features B as granted. In this method, repeatedly, item features are estimated from the given samples and then a linearly parametrized bandit algorithm is run as though the item features were known. However, we argue that this algorithm can produce regret that grows *linearly* with time T. The reason is that this method is not likely to recommend items without known ratings to users. To see this, consider the case in which the initial sample of entries includes no samples from a certain column j. A naive matrix completion estimate of the entries m_{ij} in column j will then be 0. A linear bandit algorithm will then infer that this column is useless either for exploration or for exploitation. If the highest value entries really lay in this column, we would never discover it.

Algorithm 1 Linear bandits

```
given initial samples $\Omega(0)$
estimate item features $\hat{B}$
for $t = 0, 1, 2, \ldots$ do
    observe arrival of user $i(t)$
    recommend $j(t)$ based on linear bandit algorithm, assuming $\hat{B}$ are true item features
    observe the user's rating $M_{ij}$
    update samples $\Omega(t+1) = \Omega(t) + \{(i, j)\}$
    re-estimate item features $B$
end for
```

Most solution methods for bandit problems compute a *priority* p_{ij} for each user-item pair i, j at each time t, and choose $(i, j) \in S(t)$ to maximize p_{ij}. The differences between them lie in how each priority is constructed. Priorities may be constructed in the following ways:

- *Greedy.* The priority p_{ij} is the MAP estimate of the true reward M_{ij}.

- *UCB.* The priority is an upper confidence bound on the true mean of the distribution [LR85]. Careful tuning is required to identify the best upper confidence percentile for a given application.
- *Thompson sampling.* The priority is computed as a random draw from the posterior distribution [Tho33]. This requires less tuning than UCB, and recent theoretical results [RVR13] show that Bayes risk bounds for Thompson Sampling can be derived from those for UCB algorithms.
- *Information criterion.* The priority is computed using the expected item reward based on a k step look ahead [Git89]. This approach can be very computationally expensive; often a one-step look ahead [RPF12] or an approximation thereof [LB99] is used instead.
- *Online algorithms.* The priorities are updated using online techniques such as online mirror descent. The main focus of this project is on this part.

2.3 Why are we expecting better results?

Since the learner is getting partial feedback after taking an action, *multi-armed bandits* model can serve a suitable model for this problem. A traditional multi-armed bandit framework makes no structured assumptions about the relationships between the rewards of different choices. Thus every user-item pair must be sampled at least once before the multi-armed bandit algorithm can even begin to exploit previous knowledge. This means the regret must include a term proportional to the number of entries in the matrix, mn. Since we know from the matrix completion literature that we can (probably approximately) recover the entire matrix using only $O(nr\log n)$ samples, we expect to be able to do *much* better.

Another argument is the following: we can see this problem as n parallel different multi-armed bandit problems where each row is discussed independently from other rows. treating each row independently will give us a regret bound of $n\sqrt{\text{poly(m)}T}$. But we should capture the relationship between different rows (M being low rank) and hope to get better bounds for the regret. For example, if $k=1$, then different rows are a multiple of each other and by exploring one row, we would find a regret bound of $\sqrt{\text{poly(m)}T}$ for the whole matrix. We will show that the regret bound can be decreased to $n^{3/4}k\sqrt{\text{poly}(m)T}$, where k is the rank of the preference matrix.

3 Online Mirror Descent Algorithm

3.1 The setup

Let the set of actions S, be the set of real matrices with positive entries, such that the sum of entries in each row is less than, equal to 1. More formally, define:

$$S \triangleq \{W \in \mathbf{R}^{n\times m} : W \geq 0, W\mathbf{1} \preceq \mathbf{1}\}.$$

At iteration $t = 1, \cdots, T$, the learner chooses $W_t \in S$ and then she samples an action in each row $a_{it} \sim (W_t)_{i,:}$, and observes the rewards $(Z_t)_{i,a_{it}}$ for $1 \leq i \leq n$ and enjoys the reward $\sum_{i=1}^{n}(Z_t)_{i,a_{it}}$. Hence, the expected regret will be:

$$\text{Regret} = \max_{U\in S}\sum_{t=1}^{T}[\mathbf{Tr}(W_t^T Z_t) - \mathbf{Tr}(U^T Z_t)].$$

Let $l = \min\{n,m\}$ and the rank of the reward matrices Z_t be equal to $k \ll l$.are chosen from specific distributions. For now, we work on the genral case Assume that $\Psi : \mathbf{R}^{n\times m} \to \mathbf{R}$ is $\frac{1}{\eta}$-strongly convex on the set of actions (and experts) $S \subset \mathbf{R}^{n\times m}$ with respect to norm $\|\cdot\|$. Remember that running Online Mirror Descent algorithm is simply using the following update rule: $W_{t+1} = \Pi_s(\overline{W}_{t+1})$ where $\overline{W}_{t+1} = \nabla\Psi^*\left(\nabla\Psi\left(\overline{W}_t\right) - \eta Z_t\right)$. We know that running OMD will give us the following regret bound:

$$\text{Regret} \leq \max_S \Psi - \min_S \Psi + \eta T \max\{\|Z_t\|_*\}^2.$$

In this section, we will study two different regularization functions Ψ, one of which captures the low rank assumption, and will see that the bound from this regularization function is significantly better than the generic bounds.

For the rest of this section, let $\psi : \mathbf{R}^l \to \mathbf{R}$ be the entopic function: $\psi(w) = \sum_{i=1}^{l} w_i \mathrm{log} w_i$. Also let $\sigma : \mathbf{R}^{n \times m} \to \mathbf{R}^l$ have singular values of matrix W, such that $\sigma_1(W) \geq \cdots \sigma_l(W)$. Also let $\sigma_{max} = \max\{\|W\|_2 : W \in S\}$ and $u = \max\{\|Z\|\}$.

3.2 A naive bound on the regret

We can get a bound on the regret, by simply running online mirror decsent on each row independently. However, we do not take advantage of the prior knowledge on the eank of the reward matrices. So we expect suboptimal bounds in this case. Running OMD on each row will give us $O(\sqrt{Tm})$ bound (notice that here the action space is not the simplex anymore, and that is the reason m appears instead of $\log(m)$, in fact the system can decide not to make recommendations and get zero reward). Hence, the upper bound for the total regret will be $O(n\sqrt{Tm})$.

3.3 A bound on the regret using OMD on matrices

Define $\Psi(W) = \frac{2\sqrt{ln}}{\eta}(\psi \circ \sigma)(W) = \frac{2\sqrt{ln}}{\eta} \sum_{i=1}^{l} \sigma_i(W) \mathrm{log} \sigma_i(W)$ which is $\frac{1}{\eta}$-strongly convex on S with respect to the trace norm. Then we will have the following regret bound:

$$\text{Regret} \leq O\left(\frac{m\sqrt{ln}}{\eta}\right) + \eta T u^2.$$

By choosing an appropriate value for η, we get the upper bound: $O(\sqrt{u^2 Tm\sqrt{ln}})$. Noticing that $u^2 = O(n)$, and $l \leq n$, we get the regret bound $O(n\sqrt{mT})$, which is no better than treating the different rows independently.

3.4 A better bound on the regret using OMD on matrices

Now we try to design a regularizer that can capture the low rank property of reward matrices. This regularizer will penalize us more on the second term of the regret, but we get a discount on the first term and we get a better bound over all. Define $\Psi(W) = \frac{2\sqrt{n}}{\eta}(\psi \circ \sigma)(W) = \frac{2\sqrt{n}}{\eta} \sum_{i=1}^{l} \sigma_i(W) \mathrm{log} \sigma_i(W)$ which is $\frac{1}{\eta}$-strongly convex on S with respect to the *operator norm*. Notice that the regularizer defined in previous subsection was convex with respect to the operator norm too (strong convexity with respect to the trace norm is stronger indeed), but this regularizer is different within a factor. We will discuss this difference more in the next section. Then we will have the following regret bound:

$$\text{Regret} \leq O\left(\frac{m\sqrt{n}}{\eta}\right) + \eta T k^2 u^2.$$

By choosing an appropriate value for η, we get the upper bound: $O(\sqrt{k^2 u^2 Tm\sqrt{n}})$. similar to previous subsection, we get the regret bound $O(n^{3/4} k\sqrt{mT})$, which is substantially better than the generic bound.

4 Discussion

4.1 Intution about the choice of norm

In the previous section, our two choices of the regularizer function were the same up to a scaler, and the choice of norm was very critical in developing tighter bounds. In order to justify this phenomenon, let's consider the entropic regularizer function $f(w) = \sum_{i=1}^{n} w_i \log(w_i)$ on $S \subset \mathbf{R}^n$. The Hessian of this function evaluated at a point w is equal to $(\mathrm{diag}(w))^{-1}$. So, for an arbitrary point $x \in \mathbf{R}^n$, $x^T \nabla^2 f(w) x = \sum_{i=1}^{n} x_i^2 / w_i$. We notice that $\sum_{i=1}^{n} x_i^2 / w_i \geq \|x\|_1^2 / \sum_i w_i$, which shows that f is $\frac{1}{\max_{w \in S} \sum w_i}$-strongly convex with respect to l_1 norm on S. On the other hand, we notice that $\sum_{i=1}^{n} x_i^2/w_i \geq \|x\|_\infty^2/\max_{w\in S} w_i$, which means f is $\frac{1}{\max_{w\in S} w_i}$-strongly convex with respect to l_∞ norm on S. So if we, for example, know that $S = [0,1]^n$, we will be ensured to have $\frac{1}{n}$-strong convexity of f with respect to l_1 norm, and 1-strong convexity of f with respect to l_∞ norm. Therefore, although strong convexity with respect to l_∞ might seem weaker than convexity with respect to l_1, we should take into account the set we are working in.

There are two crucial steps to tackle the problem:

1) Finding unbiased estimates of Z_t at each iteration, and reducing the problem to an online learning problem with expert advice. It looks that the following estimate might do the trick:

$$(\hat{Z}_t)_{ia} = \begin{cases} \frac{(Z_t)_{i,a}}{(W_t)_{i,a}} & \text{if } a = a_{it} \\ 0 & \text{otherwise.} \end{cases}$$

References

[AAT13] M. Afkhamizadeh, A. Avakov, and R. Takapoui. Automated recommendation systems, collaborative filtering through reinforcement learning. *CS229 Project*, December 2013.

[ACBF02] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[CR08] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *CoRR*, abs/0805.4471, 2008.

[DM12] Yash Deshpande and Andrea Montanari. Linear Bandits in High Dimension and recommendation Systems. *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1750 – 1754, 2012.

[Git89] J.C. Gittins. *Bandit Processes and Dynamic Allocation Indices*. John Wiley, 1989.

[KOM09] Raghunandan H. Keshavan, Sewoong Oh, and Andrea Montanari. Matrix completion from a few entries. *CoRR*, abs/0901.3150, 2009.

[LB99] M.S. Lobo and S. Boyd. Policies for simultaneous estimation and optimization. In *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, volume 2, pages 958–964. IEEE, 1999.

[LR85] T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[RPF12] Ilya O Ryzhov, Warren B Powell, and Peter I Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195, 2012.

[RT10] P. Rusmevichientong and J. N. Tsitsiklis. Linearly Parameterized Bandits. *Mathematics of Operations Research*, 35(2):395–411, April 2010.

[RVR13] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *arXiv preprint arXiv:1301.2609*, 2013.

[Tho33] W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

A Proof of strong convexity of Ψ

Theorem. Assume that f is a closed and convex function and let f^* be the Fenchel conjugate of f. Then f is β-strongly convex w.r.t. a norm $\|\cdot\|$ if and only if f^* is $\frac{1}{\beta}$-strongly smooth w.r.t. the dual norm $\|\cdot\|_*$.

Definition. A function $g : \mathbf{R}^n \to \mathbf{R}^*$ is *symmetric* if $g(x)$ is invariant under arbitrary permutations of x. We say g is *absolutely symmetric* if $g(x)$ is invariant under arbitrary permutations and sign changes of the components of x.

Theorem *(Lewis[1995]).* Let $f : \mathbf{R}^l \to \mathbf{R}^*$ be an absolutely symmetric function. Then, $(f \circ \sigma)^* = f^* \circ \sigma$.

Lemma *Juditsky and Nemirovski [2008].* Let Δ be an open interval. Suppose $\phi : \Delta \to \mathbf{R}^*$ is a twice differentiable convex function such that ϕ'' is monotonically non-decreasing. Let $\mathbf{S}^n(\Delta)$ be the set of all symmetric $n \times n$ matrices wth eigenvalues in Δ. Define the function $F : \mathbf{S}^n(\Delta) \to \mathbf{R}^*$

$$F(X) = \sum_{i=1}^{n} \phi(\lambda_i(x))$$

and let $f(t) = F(X + tH)$ for some $X \in \mathbf{S}^n(\Delta)$, $H \in \mathbf{S}^n$. Then we have,

$$f''(0) \leq 2 \sum_{i=1}^{n} \phi''(\lambda_i(X))\lambda_i(H)^2.$$

Theorem. Define $F(X) = \sum_i \sigma_i(X)\text{log}(\sigma_i(X))$ on its domain $\{X \in \mathbf{R}^{n \times m} : \sum_i \sigma_i(X) \leq 1\}$, i.e. the unit norm ball of the trace norm, and $F(X) = \infty$ elsewhere. Then $F(X)$ is $1/2$-strongly convex w.r.t. the trace norm.

Proof. We prove that the function $g \circ \sigma(X)$ is 2-smooth w.r.t. the operator norm where

$$g(\mathbf{x}) = \log\left(\sum_{i=1}^{n} \exp(\mathbf{x}_i)\right).$$

Since g is symmetric, we have $(g \circ \sigma)^* = g^* \circ \sigma$, where g^* can be shown to be the function

$$g^*(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{x}_i \text{log} \mathbf{x}_i$$

with domain $\{\mathbf{x} \geq \mathbf{0} : \sum_i \mathbf{x}_i \leq 1\}$. Notice that 2-smoothness of $g \circ \sigma$ implies $1/2$ strong convexity of $(g \circ \lambda)^*$. Fix arbitrary X, H and define

$$f(t) = \sum_{i=1}^{n} \exp(\sigma_i(X + tH)) = \sum_{i=1}^{n} \exp(\lambda_i((X + tH)^T(X + tH))$$

and let $h(t) = \text{log(f(t)}$. Note that $h(t) = (g \circ \sigma)(X + tH)$. To prove 2-smoothness of $g \circ \sigma$, it suffices to prove $h''(0) \leq 2\|\sigma(H)\|_\infty^2$. By the chain rule,

$$h''(t) = -\frac{(f'(t))^2}{f(t)^2} + \frac{f''(t)}{f(t)}.$$

The first term is non-positive and therefore $h''(0) \leq f''(0)/f(0)$. By the lemma,

$$\begin{aligned} f''(0) &\leq 2\sum_{i=1}^{n} exp(\sigma_i(X))\sigma_i(H)^2 \\ &\leq 2\|\sigma(H)\|_\infty^2 \sum_{i=1}^{n} \exp(\sigma_i(X)) \\ &= 2\|\sigma(H)\|_\infty^2 f(0), \end{aligned}$$

whence $h''(0) \leq f''(0)/f(0) \leq 2\|\sigma(H)\|_\infty^2$.