Relaxed Clustered Hawkes Process for Procrastination Modeling in MOOCs

Appendix

Algorithm 1

Algorithm Walk-through In the following, we provide some details of Algorithm 1 shown as below. Specifically, the the following subroutine is repeated in the algorithm:
(1) Computation of A_* (lines 8-9): The objective of this part is defined as follow:

$$\min_{A_z} F_A(A_z) := \|A_z - A_s\|_F^2 \text{ s.t. } tr(A_z) \leq c, A_z \geq 0. \tag{1}$$

by following the Accelerated Gradient Method schema, we compute $A_* = \mathcal{M}_{\gamma,\mathcal{S}^{\mathcal{A}}}$ (line 8), where $\mathcal{M}_{\gamma,\mathcal{S}^{\mathcal{A}}} := \frac{1}{\gamma}\|A - \left(S^A - \frac{1}{\gamma}\nabla\mathcal{L}(A)\right)_+\|_F^2 + \rho_3 tr(A)$ (Ji and Ye 2009); where S^A is current search point; γ is the step size; and ρ_3 is the regularization coefficient. Specifically, we use trace norm projection (TrPro) (Cai, Candès, and Shen 2010) to solve the above minimization problem. Finally $(\cdot)_+$ projects negative values to 0 as we constraint A to be nonnegative.
(2) Computation of U_* (line 10): similarly to the compuation of A_*, we compute optimal value of U, $U_* = \mathcal{M}_{S_i^U,\gamma_i}(U)$, where S^U is the current search point of U, and $(\cdot)_+$ is the nonnegative projection. Specifically the objective of this computation is:

$$\min_{U_z} F_U(U_z) := \|U_z - U_s\|_F^2 \text{ s.t. } U_z \geq 0. \tag{2}$$

(3) Computation of Z_* (lines 11-14): as the constraints on Z are more complicated, the proximal operator also has more terms. Specifically, the goal is to solve the following optimization problem:

$$\min_{Z_z} \|Z_z - \hat{Z}_s\|_F^2, \text{ s.t. } tr(Z_z) = k, Z_z \preceq I, Z_z \in S_+^M \tag{3}$$

To solve this problem, we apply eigen decomposition on Z_i such that $Z_i = Q\Sigma Q'$, where $\Sigma = diag(\hat{\sigma}_1, ..., \hat{\sigma}_M)$. It has been shown that $Z_* = Q\Sigma_* Q'$, where $\Sigma_* = diag(\sigma_1^*, ..., \sigma_k^*)$, and σ_i^* is the optimal solution to the problem (Zha et al. 2002):

$$\min_{\Sigma} \|\Sigma_* - \Sigma\|_F^2, \text{ s.t. } \sum_i^M \sigma_i = k, \ 0 \leq \sigma_i \leq 1. \tag{4}$$

To solve Eq. 4 with constraints, we apply the linear algorithm proposed in (Kiwiel 2007).

Remark: we want to quickly show that by solving problem 4, the resulting $Q\Sigma_* Q'$ provides a closed-form solution to Eq. 3. If denote eigen-decomposition of $M_z = P\Lambda P'$, by definition, $P'P = PP' = I$ and $\Lambda = diag(\lambda_1, ..., \lambda_M)$ where λ_i for $i = 1, ..., M$ are eigenvalues of z. Then Eq. 3 can be equivalently written as:

$$\min_{\Lambda,P} \|Q'P\Lambda P'Q - \Sigma\|_F^2 \text{ s.t. } tr(\Lambda) = k \tag{5}$$
$$\lambda = diag(\lambda_1, ..., \lambda_M), 0 \leq \lambda_i \leq 1, P'P = PP' = I.$$

It is easy to see that the constraints of the two equations with respect to Λ and Σ are equivalent. Furthermore, if denote the objectives of Eq. 4 and Eq. 5 as $f(\cdot)$ and $g(\cdot)$ respectively, by definition, the feasible domain of Eq. 4 is a subset of the feasible domain of Eq. 5, therefore $f(\Sigma_*) \geq g(Q'P_*\lambda_*P_*'Q)$. On the other hand, knowing that Σ is a diagonal matrix, $\|Q'P_*\lambda_*P_*'Q - \Sigma\|_F^2 \geq \|(Q'P_*\lambda_*P_*'Q) \circ I - \Sigma\|_F^2$, meaning that the optimal objective value of Eq. 4 is no greater than the optimal objective value of Eq. 5. Therefore, the two problems are equivalent.

Complexity Analysis Recall that we consider the setting where there are M students and N assignments. The complexity of the computation of A_* (line $8-9$) is $\mathcal{O}(MN^2)$ where a truncate SVD is used. To solve Eq. 3, we first apply eigen-decomposition on the $M \times M$ matrix S_i^Z (in line 11), which has time complexity of $\mathcal{O}(M^3)$, then we solve Eq. 4 which has shown to be the closed-form solution to Eq. 3 (line 12), a complexity of $\mathcal{O}(M)$ can be achieved (Kiwiel 2007). As we introduce recursive function R in Sec. 4.4, the complexity of computing loss $\mathcal{L}$ (line 15) is $\mathcal{O}(MNK)$ if let K denote the number of activities of the longest student-assignment pair. Each line of the other parts of the algorithm requires $\mathcal{O}(MN)$ as only basic operations are involved. As a result, the time complexity per time step is $\mathcal{O}(\max(M, N)^2 M + MNK)$. In the cases where conventional Hawkes model is used, without the help of recursive function R, computing the loss per time step needs $\mathcal{O}(MNK^2)$. Note that without operations such as truncated SVD, even though a complexity of $\mathcal{O}(MN^2)$ can be avoided for conventional Hawkes models, the parameters of student-assignment pairs that do not have observed activities can not be inferred.

Algorithm 1: Accelerated PGA

Input: $\eta > 1$, step size γ_0, ρ_3, MaxIter

1 initialization: $A_1 = A_0; U_1 = U_0; Z_1 = \frac{k}{M} \times I$; $\alpha_0 = 0; \alpha_1 = 1$;

2 **for** $i = 1$ *to MaxIter* **do**

3 $a_i = \frac{\alpha_{i-1}-1}{\alpha_i}$;

4 $S_i^A = A_i + a_i(A_i - A_{i-1})$;

5 $S_i^B = U_i + a_i(U_i - U_{i-1})$;

6 $S_i^Z = Z_i + a_i(Z_i - Z_{i-1})$;

7 **while** *Ture* **do**

8 Compute $A_* = \mathcal{M}_{S_i^A,\gamma_i}(A)$

9 = $\left(\text{TrPro}(S_i^A - \nabla\mathcal{L}(A)/\gamma_i, \rho_3)\right)_+$;

10 Compute $U_* = \mathcal{M}_{S_i^U,\gamma_i}(U)$;

11 Eigen-decompose $S_i^Z = Q\Sigma Q^{-1}$;

12 Compute $\underset{\sigma_i^*}{\text{argmin}} \sum_i (\sigma_i - \hat{\sigma}_i)^2$ s.t. $\sum_i^M \sigma_i = k,\ 0 \le \sigma_i \le 1$;

13 Compute $\Sigma_* = diag(\sigma_1^*, ..., \sigma_M^*)$;

14 Compute $Z_* = Q\Sigma_* Q^{-1}$;

15 **if** $\mathcal{L}(A_*, U_*, Z_*) \le \mathcal{L}(S_i^A, S_i^U, Z_i) + \sum_{x\in\{A,U,Z\}} \langle S_i^x, \delta\mathcal{L}(S_i^x)\rangle + \alpha_k/2\|S_i^x - x_*\|_F^2$ **then**

16 break;

17 **else**

18 $\gamma_i = \gamma_{i-1} \times \eta$;

19 **end**

20 $A_{i+1} = A_*; U_{i+1} = U_*; Z_{i+1} = Z_*$;

21 **if** *stopping criterion satisfied* **then**

22 break;

23 **else**

24 $\alpha_i = \frac{1+\sqrt{1+4\alpha_{i-1}^2}}{2}$

25 **end**

26 **end**

27 **end**

Output: $A = A_{i+1}, U = U_{i+1}, Z = Z_{i+1}$

When it comes to the number of parameters to be learned, for our model, due to our low rank and cluster structure assumption on $A \in \mathbb{R}^{N\times M}$, the number of parameters it requires to meet these two assumptions is $(M+N)c + 2Mk$ where $c < \min(M, N)$ and $k < M$ is respectively the rank of A and the number of clusters among students, i.e. the rank of $Z \in \mathbb{R}^{M\times M}$. For conventional Hawkes models, each student-assignment pair needs to be learned independently. As a result, the number of parameters need to complete matrix A is $M \times N$.

Convergence Analysis As mentioned earlier in this section, we have shown that Algorithm 1 repeatedly solves the subroutines respectively defined in Eq. 1, 2 and 3, where solving Eq. 3 is mathematically equivalent to solving Eq. 4. As it is known that accelerated gradient descent can achieve the optimal convergence rate of $\mathcal{O}(1/k^2)$ when the objective function is smooth, and only the subrountine of solving Eq. 1 involves non-smooth trace norm, the focus of the following section is to provide a convergence analysis on this subroutine. Specifically, by following the outline of proof provided in Ji and Ye's work (Ji and Ye 2009), we show that a rate of $\mathcal{O}(1/\epsilon^2)$ can be achieved in solving Eq. 1, even with the presence of trace norm in the objective. Specifically, if let A_* denotes the optimal solution, by applying Lemma 3.1 from Ji and Ye's work, we can obtain the following:

$$F_A(A_*) - F_A(A_1) \ge \frac{\gamma_1}{2}\|A_1 - S_1^A\|^2 + \gamma_1\langle S_1^A - A_*, A_1 - S_1^A\rangle = \frac{\gamma_1}{2}\|A_1 - A_*\|^2 - \frac{\gamma_1}{2}\|S_1^A - A_*\|^2, \tag{6}$$

which is equivalent to:

$$\frac{2}{\gamma_1}(F_A(A_1) - F_A(A_*)) \le \|S_1^A - A_*\|^2 - \|A_1 - A_*\|^2. \tag{7}$$

Then by following the proof of Theorem 4 in Ji and Ye's work, we can obtain the following inequality, using the equality $\alpha_i^2 = \alpha_{i+1}^2 - \alpha_{i+1}$ derived from the equation in line 24 of our algorithm and the definition of S_i^A in line 4:

$$\begin{aligned}&\frac{2}{\gamma_{i+1}}\left[\alpha_i^2(F_A(A_i) - F_A(A_*)) - \alpha_{i+1}^2\left((F_A(A_{i+1}) - F_A(A_*))\right)\right] \\ &\ge \|\alpha_{i+1}A_{i+1} - (\alpha_{i+1})A_i - A_*\|^2 - \|\alpha_i A_i \\ &-(\alpha_i - 1)A_{i-1} - A_*\|^2.\end{aligned} \tag{8}$$

As $\eta \ge 1$ and we update γ_{i+1} by multiplying η with γ_i, we know that $\gamma_{i+1} \ge \gamma_i$. By plugging in this inequality to Eq. 8, we can obtain the following:

$$\begin{aligned}&\frac{2}{\gamma_i}\alpha_i^2(F_A(A_i) - F_A(A_*)) - \frac{2}{\gamma_{i+1}}\alpha_{i+1}^2(F_A(A_{i+1}) - F_A(A_*)) \\ &\ge \|\alpha_{i+1}A_{i+1} - (\alpha_{i+1} - 1)A_i - A_*\|^2 - \|\alpha_i A_i \\ &-(\alpha_i - 1)A_{i-1} - A_*\|^2.\end{aligned} \tag{9}$$

By summing up each side of Eq. 9 from $i = 1$ to $i = k$, then combining with Eq. 7, we can obtain the following:

$$\begin{aligned}&\frac{2}{\gamma_i}\alpha_i^2\left(F_A(A_i) - F_A(A_*)\right) \le \|A_1 - A_*\|^2 \\ &-\|\alpha_i A_i - (\alpha_i - 1)A_{i-1} - A_*\|^2 + \frac{2}{\gamma_1}\left(F_A(A_1) - F_A(A_*)\right) \\ &\le \|A_i - A_*\|^2 - \|\alpha_i A_i - (\alpha_i - 1)A_{i-1} - A_*\|^2 \\ &+\|A_0 - A_*\|^2 - \|A_i - A_*\|^2 \\ &\le \|A_0 - A_*\|^2\end{aligned} \tag{10}$$

Using the fact that $\alpha_i \ge \frac{i+1}{2}$ (can be shown using induction from line 24 of the algorithm), we can obtain:

$$F_A(A_i) - F_A(A_*) \le \frac{2\gamma_i\|A_* - A_0\|^2}{(i+1)^2}. \tag{11}$$

Intuition Explained

Since our goal is to study sequences of student activities and their inter-arrival times, point processes are of the best choices for our application. Poisson process assumes that the past and future activities are completely independent. Unlike the memory less nature of the Poisson process, the Hawkes process expects that activities to be exciting both externally

(similar to the Poisson process) and internally, that is, activities are self-exciting.

From the branching process point of view of the Hawkes process, activities are assumed to have latent or unobserved branching structures, where the offspring activities (i.e. future activities) are triggered by parent activities (i.e. past activities) while the immigrant activities arrive independently. Therefore, the offspring are also said to be structured into clusters. In the online learning setting, smaller activity chunks towards a goal or deadline can be examples of offspring: students divide the big tasks (the whole process) into small sub-tasks (offspring clusters). The deadline (external stimuli) of a big task (such as a task) triggers the follow-up activities related to small tasks, which come one after another in a so-called burst mode (self-excitement).

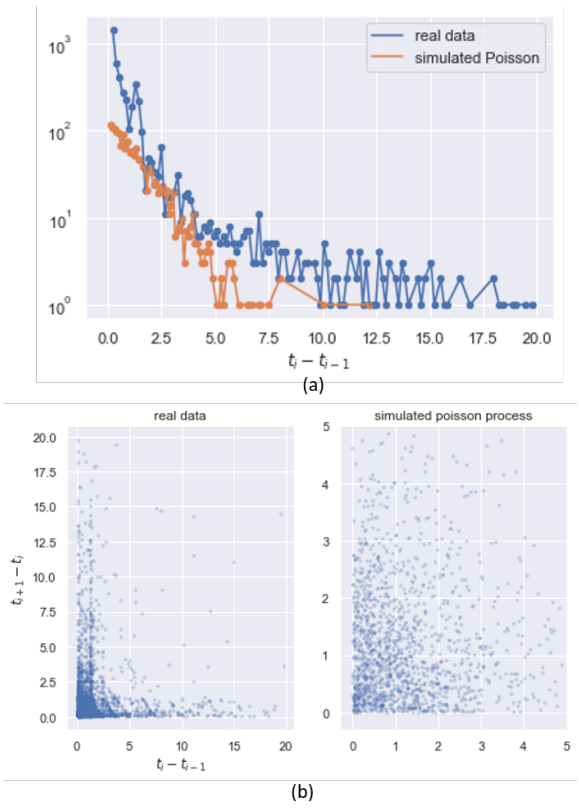

Figure 1: Two tests show the differences between a simulated Poisson process vs. a true process extracted from CANVAS dataset in terms of (a) inter-arrival times distributions and (b) inter-arrival times autocorrelation.

To empirically demonstrate that self-excitement or burstiness is observed in the online course setting, we conducted two tests to show that Poissonian properties are not present in true student activity sequences. The first test is to check the distribution of the inter-arrival times, which is defined as the difference between two consecutive activity occurrences' arrival times. In Figure. 1 (a), we show the inter-arrival times versus simulated Poisson process in a real student's sequence of activities for an assignment. The simulated Poisson process is generated with the same average rate, as the real student's sequence, on a log-log scale. We see that the Poisson process almost forms a straight line, indicating the exponential distribution of inter-arrival times, whereas the real data is "nonpoissonian", i.e. includes short pauses followed by long ones. The second test is to check the 1-lag autocorrelation of inter-arrival times. As we can see in Figure. 1 (b), no autocorrelation is spotted in the Poisson process, whereas the real data exhibits some pattern: dense activities followed by long pauses.

References

Cai, J.-F.; Candès, E. J.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization* 20(4): 1956–1982.

Ji, S.; and Ye, J. 2009. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th annual international conference on machine learning*, 457–464.

Kiwiel, K. C. 2007. On linear-time algorithms for the continuous quadratic knapsack problem. *Journal of Optimization Theory and Applications* 134(3): 549–554.

Zha, H.; He, X.; Ding, C.; Gu, M.; and Simon, H. D. 2002. Spectral relaxation for k-means clustering. In *Advances in neural information processing systems*, 1057–1064.