Abstract

Easy access to web and expansive computing has reshaped the developments in education specially towards Massive Open Online Courses (MOOCs). Despite their popularity, online platforms have been criticized on grounds of lacking a personalized educational experience and interaction with other students. This cause low engagement in courses and high drop out rate. A crucial task in both class- room environment and online platforms is how to group students such that they have efficient interaction and can benefit equally from learning.

Here we consider the problem of team formation and scheduling educational materials for students. Given a time frame d, a set of students S with their required need to learn different activities T and given k as the number of desired groups, we study the problem of finding k group of students. The goal is to teach students within time frame d such that their potential for learning is maximized and find the best schedule for each group.

Group Schedule Problem

Let P ⊆ S be a group of students and T be a set of topics. For every s ∈ S and t ∈ T let req(s, t) be the requirement of s on t that is given for every student-topic pair. Find a partition P of students into K groups, such that

\[ B(P,d) = \sum B(P, A_p) \]

is maximized, where \( A_p = \text{Schedule}(P, d) \) for every group.

Cohort Selection Problem is NP-hard.

Catalog Segmentation Problem < Cohort Selection Problem

Algorithm 3 is a heuristic to solve Cohort Selection problem.

Algorithm 2 Benefit algorithm for computing the benefit of a single student S from a schedule R.

Input: requirements req(s, t) for a student s ∈ P and every topic t ∈ T and a single schedule R

Output: b(s, R) Benefit of s from schedule R.

1: \( b = 0 \)
2: for all topics \( t \) do
3: \( b = b + \min_{\text{occurrences of } t} \text{b}(s, c_j) \)
4: end for

Algorithm 1 Schedule algorithm for computing an optimal schedule \( A_P \) for a group P.

Input: requirements req(s, t) for every s ∈ P and every topic t ∈ T, deadline d.

Output: schedule \( A_P \).

1: \( A_P = [] \)
2: \( B ← \{\text{m}(P,\{t,1\})\} \) for \( t \in T \)
3: \( R ← \{0\} \) for all \( t \in T \)
4: while |\( A_P \)| < d do
5: Find topic \( u_t \) with maximum marginal benefit in \( B \)
6: \( A_P ← u_t, R[u_t] \)
7: \( R[u_t]++ \)
8: Update \( B[u_t] \) to \( \text{m}(P,\{t, R[u_t]\}) \)
9: end while

Cohort Selection Problem

Let S be a set of students and T be a set of topics. For every s ∈ S and t ∈ T let req(s, t) be the requirement of s on t that is given for every student-topic pair. Find a partition P of students into K groups, such that

\[ B(P,d) = \sum B(P, A_p) \]

is maximized, where \( A_p = \text{Schedule}(P, d) \) for every group.

Algorithm 3 CohPart for computing the partition \( P \) based on the benefit of students from schedules.

Input: requirement req(s, t) for every s ∈ S and t ∈ T, number of timeslots d, number of groups K.

Output: partition \( P \).

1: \( C = \)
2: \( P = \{P_1, P_2, \ldots, P_K\} \)
3: Run Random Partitioning on the students and obtain \( P_i \)
4: for \( i = 1, \ldots, K \) do
5: \( c_i = \text{Schedule}(P_i, d) \)
6: end for
7: while convergence is achieved do
8: for all students \( s \in S \) do
9: \( P_i ← s \) such that \( i = \arg\max_{j=1,\ldots,K} b(s, c_j) \)
10: end for
11: for \( i = 1, \ldots, K \) do
12: \( c_i = \text{Schedule}(P_i, d) \)
13: end for
14: end while

Experiments

Run Time

Reference