

연구논문/작품 최종보고서

2023 학년도 제 1학기

제목 : 과제 추천 사이트

나재성(2013311058)

(※팀원 이름 모두 기재)

2023 년 4 월 12 일

지도교수: 이은석 서명_____

계획(10)	주제(20)	개념(20)	상세(30)	보고서(20)	총점(100)

■ 요약

실생활에서 자주 사용되는 시스템인 추천시스템은 학업의 영역에서 적용될 수 있다. 특히 프로그래밍 교육 영역에서 개념을 학습하고 연습문제 해결을 시도할 때 개인의 수준에 맞는 문제를 추천해줄 경우 학습자의 학습 효율을 높일 수 있을 것이다. 따라서 우리는 프로그래밍을 학습하는 학생들을 대상으로 개개인의 수준 및 능력에 따라 적절한 맞춤 문제를 추천해주는 시스템을 제작하고자 한다. 해당 시스템의 구현을 위해 필요한 문제 및 학생들의 제출 내역은 Codechef 데이터를 수집한다. 이를 기반으로 Next.js기반의 웹 사이트를 제작하여 학습자가 특정 문제를 틀리거나 맞았을 경우를 가정하여 문제를 추천해주는 사이트를 제작하였다.

■ 서론

가) 제안배경 및 필요성

현재 YouTube, Netflix, Amazon, 등 매우 많은 분야에서 추천시스템이 적용되어 활용되고 있다. 추천시스템이 적용되지 않는 기업을 찾는 게 빠를 정도로 대부분 영역에 추천시스템이 적용되어 개개인을 위한 맞춤 추천이 이루어지고 있다. 소비자가 선택할 수 있는 물품의 종류가 적을 때에는 추천시스템이 큰 의미를 가지지 않을 수 있지만, 소비자가 선택할 수 있는 물품의 폭이 굉장히 커지며 소비자가 모든 물품을 선택, 비교하는 것이 현실적으로 불가능해졌다. 소비자의 입장에서 추천시스템은 효율적으로 자신에게 맞는, 만족스러운 물품을 찾을 수 있게 해준다.

소비의 관점에서 추천시스템은 매우 큰 의미를 가지고, 실제로 많은 분야에 적용되고 있다. 하지만 추천시스템은 학업의 영역에서도 적용될 수 있다. 대부분 사람이 적게는 몇 년에서 몇 십 년까지도 무엇인가를 공부하는 데 사용한다. 그리고 무엇인가를 학습하는 과정에서 배운 이론들을 적용해 연습문제를 푸는 것은 굉장히 중요하고, 대부분 사람이 따르고 있는 공부 방식이다. Kousar.P 에 따르면, 이론적인 수업 방식에 비해 문제풀이식의 수업 방식이 더 높은 학업성취도를 보인다고 하였다. 현재 본인의 수준보다 현저히 낮은 수준의 연습문제를 풀거나, 너무 어려운 문제를 푸는 것은 비효율적이다. 실제로 개인의 수준을 정확하게 평가, 판단할 수 있는 기준은 현실적으로 존재하지 않으며, 연습문제의 수준 또한 정확하게 판단할 수 없다. 여기에 더해, 연습문제의 수는 기하급수적으로 많으며, 그 모두를

분석하여 개인에게 맞는 문제를 푸는 것 또한 불가능하다. 개개인에 따라 한 분야에서도 특정 영역에서는 두각을 나타내어 실력이 빨리 늘 수도 있고, 특정 영역에서는 반대로 더딜 수도 있다. 이러한 경우, 잘 되는 영역에서는 쉬운 문제들을 상대적으로 적게 연습하고 난이도를 빨리 올리고, 잘 안되는 영역에서는 좀 더 기초적인 문제들로 연습량을 늘리는 것이 적합할 것이다.

프로그래밍은 최근 대두되고 있는 교육 영역 중 하나이다. 우리는 프로그래밍을 학습하는 학생들을 대상으로 개개인의 수준 및 능력에 따라 적절한 맞춤 문제를 추천해주어 학업 성취의 속도를 높여주는 시스템을 만들고자 한다.

나) 연구논문/작품의 목표

본 연구 작품의 목표는 현재 존재하는 추천 시스템 알고리즘들을 파악하여, 프로그래밍 문제 추천 시스템에 적합한 알고리즘을 찾아 개개인들의 수준에 맞춰 가장 효율적인 문제를 추천해주는 것이다.

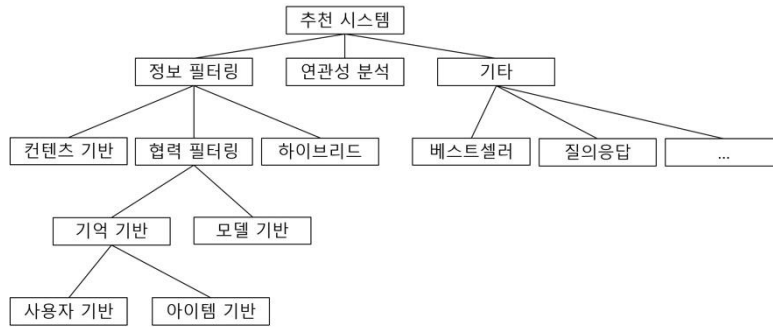
이어지는 관련 연구 장에서는 기존의 추천 시스템에 대해서 논의한다. 제안 작품 소개 장에서는 제안하는 문제 추천시스템의 대략적인 개요와 사용하는 알고리즘에 관해 설명한다. 구현 및 결과분석에서는 지금까지 이루어진 구현 결과에 대한 간략한 논의한 논의를 진행한다. 결론 및 소감 장을 끝으로 본 보고서를 마무리한다.

다) 연구논문/작품의 Overview

Codechef라는 연습문제 사이트에서, 유저들의 문제 별 제출 기록을 추출하여 사용하였다. 약 20만명의 사용자들의 제출 기록 데이터를 수집하였고, 수집한 데이터를 surprise library의 KNN 알고리즘을 사용하여 학습한 결과, 0.4059의 RMSE값을 얻었다.

■ 관련연구

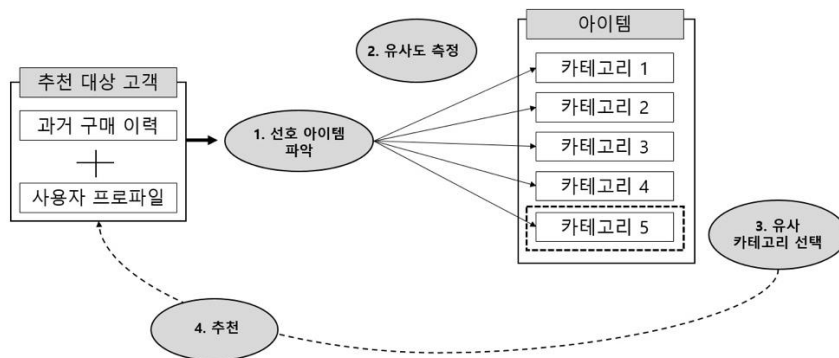
추천 시스템 알고리즘에는 크게 2가지 방향이 있는데, 첫째는 콘텐츠 기반 접근방식이고, 둘째는 협력 필터링이다.



[Figure 1] 추천 시스템의 종류

1. 컨텐츠 기반

컨텐츠기반 접근방식[1,2,3]은 아이템의 컨텐츠를 직접 분석하는 방식이다. 아이템과 아이템, 혹은 아이템과 사용자 선호도 간의 유사성을 분석하여, 이를 토대로 새로운 아이템을 추천하는 방식이다. 먼저 사용자가 아이템에 평가한 점수, 과거 구매내역, 그리고 사용자가 직접 입력한 프로파일 등을 분석하여, 선호 아이템을 파악한다. 그 후에, 아이템들의 카테고리들과 사용자의 선호 아이템의 유사도를 측정하여, 가장 유사도가 높은 카테고리의 아이템을 추천해주는 방식이다.



[Figure 2] 컨텐츠 기반 추천시스템

컨텐츠기반 접근방식의 특징으로는 독립적 정보 활용, 새로운 아이템, 과도한 특수화가 있다.

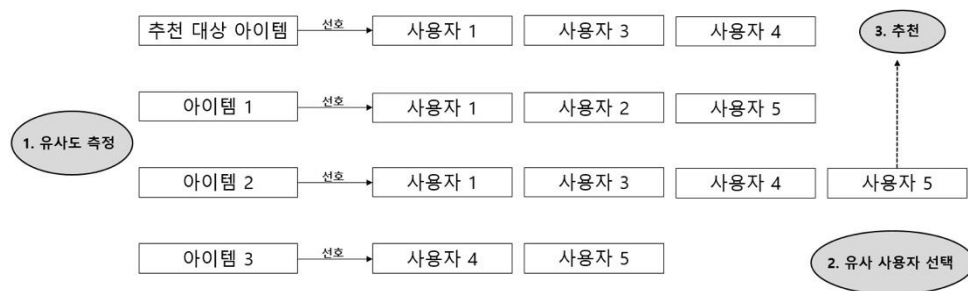
컨텐츠기반 접근방식의 경우, 자신만의 구매내역이나, 프로파일 정보가 이용되므로, 다른 사용자의 정보가 부족한 경우에도 사용될 수 있다. 하지만, 선호도를 파악할 수 있을 만큼의, 과거 구매 이력이 부족한 경우, 성능이 저하된다. 만약 과거 구매 이력과, 프로파일 정보가 모두 존재하지 않는다면 추천 시스템을 구현할 수 없다.

새로운 아이템이 추가된 경우, 콘텐츠기반 접근방식을 사용하면, 해당 아이템의 점수가 아직 누구에게도 매겨지지 않았다고 하더라도, 아이템 간의 속성을 파악하여, 카테고리에 할당이 가능하다. 그 이후에, 해당 카테고리과 유사한 선호도를 가진 고객에게 추천이 가능하다. 그러므로, 협력 필터링 방식에서 생기는 누군가가 점수를 주기 전까지는 추천 리스트에 포함할 수 없는 “first rater”의 문제가 발생하지 않는다.

하지만, 콘텐츠기반 접근방식의 경우, 해당 사용자만의 정보를 분석하여 추천해주기 때문에, 다른 사용자들의 취향이나 선호도를 반영하지 못한다. 그로 인해, 과거에 추천된 아이템들과 유사한 아이템들만 지속해서 추천되는 문제가 있다. 이 같은 문제를 해결하기 위해, 추천 과정에서 유전자 알고리즘을 적용하여 무작위 요소를 추가하는 방향으로 추천 아이템의 다양성을 보강할 수 있다.

2. 협력필터링

협력필터링[1,4]은, ‘특정 아이템에 대해 선호도가 유사한 고객들은 다른 아이템에 대해서도 비슷한 선호도를 보일 것’이라는 가정을 바탕으로, 사용자 혹은 아이템 간 유사도를 기반으로 선호를 예측한다. 추천 대상이 되는 고객과 비슷한 사용자를 선정하여, 그들이 선호하는 아이템을 추천 대상에게 추천하기 때문에 추천 아이템의 다양성이 보장된다.



[Figure 3] 사용자 기반 추천시스템



[Figure 4] 아이템 기반 추천시스템

위와 같이, 추천 대상이 정해지면, 다른 사용자 혹은 아이템 간의 유사도를 측정하여, 가장 유사한 대상이 선호한 아이템 중, 추천 대상이 아직 구매하지 않은 아이템을 추천해주는 방식이다. 이런 식으로, 사용자 간 유사도를 계산하여, 유사도가 높은 사용자가 선택한 아이템을 추천해주는 방식을 기억 기반 협력필터링이라고 한다. 군집화, 분류, 예측의 단계에서 기계학습 또는 데이터마이닝 기법을 활용하는 방식은 모델 기반 협력필터링이다.

협력필터링의 특징으로는 데이터 희소성, 확장성, Grey Sheep, 그리고 Shilling Attack이 있다.

	아이템 1	아이템 2	아이템 3	아이템 4
사용자 1	4	?	5	
사용자 1	4	4	1	
사용자 1	3		2	
사용자 1	4	4		
사용자 1				

[Figure 5] 협력 필터링의 동작 예시

위와 같은 사용자-아이템 행렬에서, 평가값이 상품의 인기도에 따라 편중되거나, 아직 평가값을 입력하지 않은 새로운 사용자가 존재할 수 있다. 이런 경우, 유사도를 측정할 만한 데이터가 충분하지 않기 때문에, 유사도를 측정하고, 선호도를 예측하는 것이 불가능하며 이를 Cold Start라고 한다. 또한 아이템 4와 같이 아직 구매가 이루어지지 않아 평가점수가 없는 아이템의 경우 누군가가 점수를 주기 전까지는 추천이 이루어질 수 없는 First Rater의 문제가 있다.

위와 같은 데이터 희소성으로 생기는 문제들을 해결하기 위해 여러 가지 연구가 진행되고 있다. 논문 추천시스템[1]에서는 논문 이용 정보 대신 논문의 키워드를 활용하는 방식을 사용하였다. 다른 예로는, 신규고객의 과거 구매 이력이 없을

경우, 프로필 정보를 기반으로 선호도가 비슷한 사용자를 결정하는 알고리즘도 개발되었다.

최근 상품의 종류가 다양해지고, 사용자는 늘어남에 따라 사용자-아이템의 데이터 셋이 점점 커져, 시간과 비용이 많이 드는 문제가 생기고 있습니다. 이러한 문제를 해결하기 위해, 유사도가 큰 순서대로 N개의 사용자/아이템만을 선정하여 선호도를 예측하는 방법이 제안되었다.

협력필터링의 특성상, 사람들의 취향에 경향, 혹은 패턴이 존재한다는 가정이 들어간다. 하지만, 일관성 없는 의견을 가진 사용자들의 데이터가 데이터 셋에 포함될 경우, 예측의 정확도를 떨어뜨리는 문제가 있다.

■ 제안 작품 소개

본 제안 작품은 문제 추천 시스템을 적용하여 실제로 문제를 추천 받을 수 있는 웹사이트를 구현하고자 한다. 이를 위해 구현하여야 할 것들은 (1) 데이터베이스 구축, (2) 추천 시스템 알고리즘, (3) 웹사이트 구현 이다.

1. 데이터베이스

추천시스템의 경우 사용자들의 이용내역을 수집한 큰 규모의 데이터들을 사용하여 각각 사용자들의 관계성을 계산한 뒤 예측 값을 도출한다. 문제 추천 시스템의 경우, 여러가지 문제들의 데이터와, 어떤 사용자가 어떤 문제를 풀었고, 정답을 맞추었는지 아닌지에 대한 데이터가 필요하다. 이를 위해서 코딩 문제 모음 사이트인 “Codechef.com”을 사용하였다. Selenium 라이브러리를 사용하여, 해당 웹사이트에 올라와 있는 코딩 연습 문제들과 사용자들, 그리고 해당 사용자들의 제출 기록을 수집한다.

협력필터링 방식을 통한 추천 외에도, 난이도 별, 태그 별 추천을 위하여 해당 문제들의 난이도와 태그도 같이 수집한다. 이를 통해 약 4천가지의 문제들과 약 20만명의 사용자들의 데이터를 수집할 수 있고, 각각의 사용자가 제출 기록 또한 수집할 수 있다. 수집한 데이터의 저장을 위해서는 MySQL DB를 사용한다.

사용자들의 제출 기록의 경우, 같은 문제에 대해 여러 번의 제출 기록이 있는 경우가 대다수인데, 이는 해당 문제를 틀려 여러 번 시도를 하는 이유 때문이고, 같은 문제라 하여도 여러 프로그래밍 언어를 사용하여 제출하는 경우도 있다. 이러한 경우, 여러 번의 제출 기록 중, 한번이라도 정답을 제출한 경우가 있는 경우에만 해당 사용자가 해당 문제를 맞게 제출했다고 수집한다. 반대로 여러 번의 제출 기록에도 불구하고 정답을 제출 한 적이 없는 경우에는 해당 문제를 틀렸다고 수집한

다.

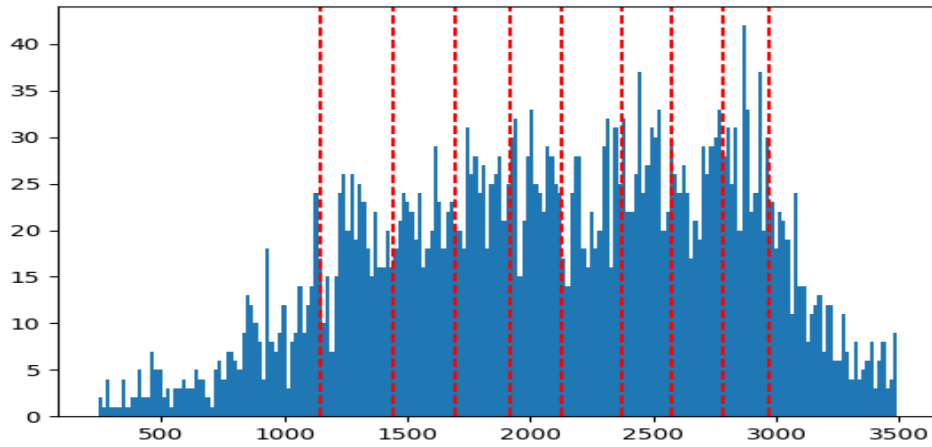
2. 추천 시스템 알고리즘

위에서 수집한 데이터를 토대로, 어떤 문제를 추천해줄 것이지를 결정하는 알고리즘을 구현한다. 위에서 소개한 바와 같이 추천시스템 알고리즘에는 협력필터링, 콘텐츠 기반 등등 여러가지의 알고리즘이 있다. 목표가 해당 사용자에게 맞는 문제를 추천해주는 것이므로, 해당 사용자와 제출 이력이 비슷한 사람의 문제를 추천해주는 것이 적합하므로, 협력필터링 기법을 사용한다.

협력필터링 기법을 구현하기 위해, Surprise 라이브러리를 사용한다. Surprise 라이브러리는 KNN, SVD, Co-clustering 등 많은 예측 알고리즘을 제공해준다. 이 중에서 기본적인 협력필터링을 구현해주는 KNN 알고리즘을 사용한다.

Surprise 라이브러리에서 제공하는 KNN 알고리즘은, 최대 k값, 최소 k값, 아이템 기반인지 혹은 사용자 기반인지, 사용자간의 유사도 측정 방법을 정할 수 있다. 사용자 기반의 경우, 데이터셋의 크기가 커짐에 따라 모든 사용자 간의 유사도를 측정해야 하므로 시간이 오래 걸리고, 심지어 성능 또한 대부분의 경우 아이템 기반에 비해 낮게 나오기 때문에, 아이템 기반의 협력필터링을 사용한다. k값의 경우, Surprise의 KNN 알고리즘의 기본값이 40 과 1을 각각 최대값과 최소값으로 사용한다. 유사도 측정 방법의 경우, Surprise 라이브러리가 제공하는 Cosine, MSD, Pearson 중에서 MSD 방법을 선택한다.

다른 영화나, 물품 구매 추천시스템 같은 경우, 특정 사용자가 실제 구매 이력이 없는 모든 아이템들에 대한 예측 평점 중 가장 높은 점수를 갖는 아이템을 추천해준다. 사용자들의 평점 이력과, 사용자들의 문제 제출 이력은 다르기 때문에 적합한 방법을 고안하였다. 그 결과, 사용자가 문제를 틀렸을 경우 0, 맞았을 경우 1로 데이터를 수집하였다. 이러한 방법으로 수집한 데이터를 사용하면, 사용자의 문제에 대한 예측 평점이 아닌 예측 점수라고 판단 할 수 있다. 즉, 예측 점수가 1에 가까울 경우, 실제로 풀었을 때 맞추기 쉬운 문제라고 할 수 있고, 0에 가까울 수록 틀리기 쉬운 문제라고 볼 수 있다. 그래서 위의 KNN 알고리즘을 통하여, 데이터셋을 학습시킨 후, 특정 사용자의 예측 점수를 모든 문제들에 대해 계산한 뒤, 1에 가까운 3문제, 0.5에 가까운 3문제, 0에 가까운 3문제를 추천해주도록 구현하였다. 1에 가까운 문제의 경우, 좀더 기본기를 연습할 수 있는 문제들 일 것이고, 0.5에 가까운 문제의 경우, 가장 현재의 수준에 맞는 문제, 그리고 0에 가까운 문제들은 어렵지만 도전해볼 만한 문제이다.



[Figure 6] 문제의 난이도별 분포도

또한, 사용자들간에 유사도를 측정하여 추천해주는 방식 외에도, 난이도와 태그를 기반으로 한 추천 방식도 구현한다. Figure 5의 경우 수집한 문제들의 난이도 분포이다. 해당 난이도 분포의 평균은 2101, 표준 편차는 703 이고, 위의 난이도 분포를 10% 단위로 1~10단계로 분류하여, 유저가 푼 문제의 단계에서 1~2단계 윗 단계의 문제와, 아랫단계 문제를 추천해주도록 구현하였다. 특정 문제를 틀렸을 경우, 해당 문제보다 조금 낮은 난이도를 갖는 문제들을 추천하고, 맞았을 경우는 조금 높은 난이도의 문제를 추천해주는 것이다.

	Tag 1	Tag 2	Tag 3		Tag n
Item 2	1	0	1	...	0
Item 3	1	1	1	...	1
Item n	0	1	1		0

[Figure 7] 문제-태그 행렬 예제

또한, 수집한 문제들의 tag를 기반으로 약 4천가지의 문제들 간의 유사도를 측정하였다. 유사도 측정에는 일반적으로 MSD, Cosine, Pearson이 가장 많이 사용되는데, 이중 MSD와 Pearson 유사도 측정의 경우 아이템에 대한 유저들의 평가점수를 기반으로 계산되고, Cosine 유사도 측정의 경우 대부분 TF-IDF에 사용되어 문장들의 유사도를 측정하는데 사용된다. Figure 7과 같이 각각의 문제들이 얼마나 유


사한 태그들을 가지는 지를 측정해야하기 때문에, 평가점수들의 유사도를 측정하는 MSD나 Pearson보다는 문장의 유사도를 측정하는 Cosine 유사도 측정이 적절하다고 판단하였다. Cosine 유사도를 측정한 것을 기반으로, 사용자가 특정 문제를 제출하였을 때, 특정 문제와 가장 유사도가 높은 3문제, 가장 유사도가 낮은 3문제를 추천하도록 구현하였다. 유사도가 가장 높은 문제는 해당 문제와 tag가 유사하여, 비슷한 방식으로 해결할 수 있는 문제이고, 유사도가 가장 낮은 문제는 해당 문제와 tag가 가장 달라, 완전히 다른 방식으로 해결 해야 하는 문제라고 볼 수 있다.

3. 웹사이트 구현

실제로 위의 추천되는 문제들을 볼 수 있도록 웹사이트를 구현한다. Next.js 프레임워크와 Prisma, Bootstrap을 사용하여 사이트를 구현한다.

사이트의 구성은, 로그인 페이지, 회원가입 페이지, 문제들을 볼 수 있는 문제리스트 페이지, 문제를 풀고 추천 문제를 볼 수 있는 제출 페이지 이다. 로그인, 회원가입 등 DB에 있는 데이터의 접근을 위해 Prisma를 사용하였다. 문제 제출 페이지의 경우, 다른 코딩 연습문제 사이트들과 같이 제출코드란을 구현해 놓았지만, 실제로 컴파일 하여 코드를 실행시켜 해당 문제를 맞추었는지 아닌지를 구분 하는 기능을 추가하는 것은 본 프로젝트의 범위를 벗어나므로, 정답 버튼과 오답 버튼을 임시로 구현한다. 정답 버튼을 누를 경우, 해당 사용자의 해당 문제를 맞춘 것으로 간주하여 제출 이력이 저장되고, 오답 버튼의 경우 틀린 것으로 간주하여 저장한다.

■ 구현 및 결과분석



```
SVD rmse
RMSE: 0.4097
```

[Figure 8] RMSE value

위에 기술한 바와 같이, 약 20만명의 사용자가 약 4천개의 문제에 대한 제출 이력을 토대로 아이템 기반의 협력필터링 추천시스템을 구현하였다. 이 제출 이력 데이터셋을 KNN 알고리즘을 사용하여 학습한 결과, RMSE값이 0.4059로 나왔다.

Today's Problems Challenging Problems • ANDORJUN • SUBPRNL • MARBLES • ARCOT • COOBIN Practice Problems • BOUNCHER • GAMENUM • SUBRIGHT • LUCKYCOUR • CHALLENGE	WATERREQ	Water Requirement	351
	LTIME	Lunchtime	352
	APPROXING	Apples and Oranges	355
	CE7W	OneTwo	2149
	PRATY2	Chef gives Party	363
	JABO	Jam Board	2151
	EMAILREM	Email Reminders	379
	SIBRENDIS	Sis Friends	382
	CARS	The Cheaper Cab	399
	ADJUVANT	Rating Improvement	400
	COCHCOLURES	Chef and Chocolates	405
	VOLCONTROL	Volume Control	409
	HQBTEST	ATM	410
	FOOTCUP	Football Cup	412
	FRC	Fill the Bucket	419
	FUDWGO2	Food Remandu	421
	MANIPULATS	Edo and Guards	427
	SHMARKET	Stable market	2151
	EMRY	Expiring Bread	440
	TYRE	Tyre problems	452
	COBNCHE	Statements Checking	2151
	MOULM	Positive Mex	2153
	SCOWING	Sum of Digits	455
	BUDGET	Monthly Budget	456
	SUBMEDS	Chef and Subtime MEDs	2158
	CHFRAN	Chiffons and Ranges	2159

[Figure 9] 문제목록 화면

구현된 웹페이지 중 핵심 화면 중 하나인 문제목록 화면이다. 해당 화면에는 제공하는 문제와 태그를 보여준다. 해당 페이지에는 “오늘의 문제”라는 섹션에 학생이 오늘 풀 가능성이 높거나 혹은 실력향상에 필요한 조금 더 어려운 문제들을 추천해준다.

welcome **JK**

Sign off

Home Profile

Search

How Many Maximums

Problem Code: HOWMANYMAX

Difficulty Rating: 1332

Tags: [Implementation](#) [Algorithms](#)

Problem

From an array A containing N integers, you construct a binary string SSS of length (N-1)(N-1) as follows. For all 1≤i<N let i & N(i≤N). If A[i]<A[i+1] & A[i+1]<A[i+1], then S[i+1] = '0'. If A[i]<A[i+1] & A[i+1]<A[i+1], then S[i+1] = '1'. Given the string SSS, determine the count of indices i (1≤i≤N) such that it is possible for A[i] to be the maximum element of the array A.

Write your code here

Correct

Incorrect

Opposite Problems	Similar Problems	Similar Difficulty Problems	Challenging Problems
1. COOK	1. BAKE	1. CHEFRAV	1. BIFREND
2. SQUATZ	2. CTIME	2. CARVANS	2. RESQ
3. CHSPPL	3. SORRELI	3. CBANG	3. COVPRD
4. RUBBER	4. GENALZAD		
5. THEATRE	5. JOKEBETRN		

[Figure 10] 코딩 페이지 및 추천 결과 확인

코딩 페이지에서는 간단한 코드 에디터와 학생이 맞거나 틀렸을 경우 각 상황에 맞는 문제들을 추천해준다. 현재 문제를 맞췄을 경우, 조금 더 난이도가 있는 문제를 추천해주고, 틀렸을 경우 조금 더 쉬운 문제를 추천해준다. 추천된 문제들의 제목을 클릭하면 해당 문제 페이지로 이동 가능하다.

하지만, 추천된 문제들이 정말로 해당 사용자에게 맞는 문제인지를 알 수 없다. 추천 알고리즘이 많이 사용되는 Netflix, YouTube, Amazon 등의 시스템에서는,

해당 추천 시스템이 적용된 후, 특정 사용자의 영상 조회수 증가, 영화 조회수 증가, 추천 받은 영화에 대한 높은 평점, 쇼핑 물품 추천이라면 실제 매출의 증가처럼 추천 시스템이 제대로 동작하는지를 판단 할 수 있는 방법이 있다. 본 연구 작품의 경우, 추천 시스템이 제대로 동작하는 지 판단할 수 있는 방법이 RMSE값 외에는 없다. 만약 실제로 배포하여, 실제 사용자들이 많이 사용한다고 가정하였을 경우, 사용자의 시간대비 문제 난이도 상승률을 관찰하면 성공적인 작동여부를 판단할 수 있을 것이다. 혹은, 사용자 그룹을 모집하여, 레벨 테스트를 실시해 각각의 레벨로 분류 후, 문제 추천 시스템을 사용하여 공부하는 그룹과, 그렇지 않은 그룹으로 나누어 일정 시간 뒤의 학업성취도를 비교하는 방법도 가능할 것이다.

또한, 본 연구작품에서 사용자들의 문제 제출 기록을 수집할 때, 특정 문제에 대한 제출 기록이 여러 번 있는 경우가 많았다.

■ 결론 및 소감

프로그래밍 실력 향상에 있어, 코딩 연습 문제를 풀어보는 것은 굉장히 중요하다 생각한다. 이론적으로 배운 알고리즘, 자료구조와 문제해결방법등을 실제로 구현해보고, 문제들에 적용시켜 보며 보다 깊은 이해를 할 수 있기 때문이다. 하지만, 웹에 존재하는 연습문제 사이트들의 경우, 보통 수천 개의 문제가 존재하고, 이들은 태그와 난이도 별로 정리 되어있는 것이 보통 전부이다. 그러므로, 현재 다른 분야에서 많이 발전되고 있는 추천 알고리즘들을 연습문제 추천의 영역에도 적용시킨다면, 본인과 같은 많은 학생들과 프로그래밍을 처음 시작하여 공부하는 사람들에게 큰 도움이 될 수 있을 것이라 생각한다.

수업을 듣고, 과제를 할 때는, 제가 일상생활에서 경험하는 것들을 프로그램 하고, 구현해본 적이 없어 굉장히 어렵지만 흥미로웠다. 항상 한 문제를 푸는 소스코드 하나를 작성하는 것이 많아, 이처럼 여러가지 코드들이 모여 하나의 프로그램으로 작동하도록 무엇을 만들어 본건 처음이었다. 내가 일상에서 사용하는 프로그램들, 보는 웹사이트들, 앱들이 실질적으로 어떻게 동작을 하는지, 이론수업을 들었을 때는 막연하게 이렇겠구나 생각했던 구조가 훨씬 자세하게 이해되었다. 물론, 구조 하나하나가 실무에서 사용하는 방식처럼 매끄럽게 구현된 것은 아니겠지만, 어떻게든 동작이 되게 구현해본 것이 굉장히 흥미로웠고 좋은 개발 경험이었다.

■ 참고문헌

[1] Jieun Son, Seoung Bum Kim, Hyunjoong Kim and Sungzoon Choo, "Review and

- Analysis of Recommender Systems", April 2015.
- [2] Lops, Pasquale, et al. "Trends in content-based recommendation." *User Modeling and User-Adapted Interaction* 29.2 (2019): 239-249.
 - [3] Geetha, G., et al. "A hybrid approach using collaborative filtering and content based filtering for recommender system." *Journal of Physics: Conference Series*. Vol. 1000. No. 1. IOP Publishing, 2018.
 - [4] Kluver, Daniel, Michael D. Ekstrand, and Joseph A. Konstan. "Rating-based collaborative filtering: algorithms and evaluation." *Social Information Access* (2018): 344-390.
 - [5] Surprise library, <https://surpriselib.com/>.
 - [6] Kousar Perveen, "Effect of the Problem-Solving Approach on Academic Achievement of Students in Mathematics", March 2010.
 - [7] Debashis Das, Laxman Sahoo, Sujoy Datta, "A Survey on Recommendation System", February 2017.
 - [8] Miha Grčar, Blaž Fortuna, Dunja Mladenič, Marko Grobelnik, "kNN Versus SVM in the Collaborative Filtering Framework", August 2005.
 - [9] Fethi Fkih, "Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison", *Journal of King Saud University – Computer and Information Sciences*, October 2022.
 - [10] 김동현, 박찬영, 오진오, 유환조, "추천시스템에서의 Cold-Start 문제에 관한 연구 동향", June 2016.