

머신러닝을 활용한 프로그래밍언어 객관식 문제의 난이도 조정에 대한 연구

(A study on the difficulty adjustment of programming
language multiple-choice problems using machine learning)

김 은 정)*
(EunJung Kim)

요 약 LMS 기반의 온라인 평가를 위해 출제되는 문제들은 교수자가 직접 출제하거나 또는 카테고리별로 나뉘어진 문제은행에서 난이도에 따른 자동 출제 방식을 주로 이용한다. 이중에서 난이도에 따른 자동출제 방식은 평가자들에게 출제되는 문제가 서로 다를수 있기 때문에 무엇보다 객관적이고 효율적인 방법으로 문제의 난이도를 관리하는 것이 중요하다. 본 논문에서는 문제의 정답률뿐만 아니라 해당 문제를 해결하는데 사용된 소요시간을 같이 고려한 난이도 재조정 알고리즘을 제시한다. 이를 위해 머신러닝의 로지스틱 회귀 분류 알고리즘을 이용하였으며, 학습 모델의 예측 확률값을 기반으로 기준 임계값을 설정하여 각 문항별 난이도 재조정에 활용하였다. 그 결과 정답률에만 의존한 문항별 난이도에 많은 변화가 일어남을 확인할 수 있었다. 또한 조정된 난이도의 문제를 이용하여 그룹별 평가를 수행한 결과, 정답률 기반의 난이도 문제에 비해서 대부분의 그룹에서 평균 점수가 향상됨을 확인할 수 있었다.

핵심주제어: 프로그래밍 교육, 온라인 평가, 난이도 조정, 머신러닝

Abstract For the questions asked for LMS-based online evaluation the professor directly set exam questions, or use the automatic question-taking method according to the level of difficulty using the question bank divided by category. Among them, it is important to manage the difficulty of questions in an objective and efficient way, above all, in the automatic question-taking method according to difficulty. Because the questions presented to the evaluators may be different. In this paper, we propose an difficulty re-adjustment algorithm that considers not only the correct rate of a problem but also the time taken to solve the problem. For this, a logistic regression classification algorithm was used of machine learning, and a reference threshold was set based on the predicted probability value of the learning model and used to readjust the difficulty of each item. As a result, it was confirmed that there were many changes in the difficulty of each item that depended only on the existing correct rate. Also, as a result of performing group evaluation using the adjustment difficulty problem, it was confirmed that the average score improved in most groups compared to the difficulty problem based on the percentage of correct answers.

Keywords: programming education, online assessment, difficulty adjustment, machine learning

* Corresponding Author: kimeunjung@pusan.ac.kr
Manuscript received January 25, 2022 / revised March 17,

2022 / accepted April 06, 2022
1) 부산대학교 교양교육원, 제1저자, 교신저자

1. 서론

4차 산업 혁명의 시대에 도래하여 소프트웨어의 중요성과 필요성이 커짐으로 인해 컴퓨팅 사고와 프로그래밍 언어에 대한 교육이 강조되고 있다. 이에 초중고에서 컴퓨팅사고에 대한 교육을 많이 하고 있으며, 대부분의 대학에서도 교양수업에서 프로그래밍 교육을 실시하고 있다. 또한 오늘날 코로나 시국이 장기화되면서 대학 교육에서 교양수업이나 이론 수업은 LMS (Learning Management System) 기반의 온라인 수업이나 실시간 줌(ZOOM) 수업이 많이 이루어지고 있으며, 평가에서도 온라인 시험이 많이 시행되고 있다. 특히 프로그래밍 언어 수업은 평가에서 필기와 실기가 병행되어 실시될 수 있다. 이 중에서 프로그램을 직접 작성하여 파일을 제출하는 실기의 경우에는 시험 감독의 어려움과 같은 현실적인 문제 때문에 평가의 공정성 문제가 발생할 수 있다. 따라서 정해진 시간안에 시행되는 LMS 기반의 필기 시험이 전체 평가 항목에서 많은 비중을 차지하고 있다. 일반적으로 온라인 평가를 위한 문제는 교수자가 직접 출제하거나 카테고리별로 나뉘어진 문제은행에서 난이도에 따른 자동출제 방식을 주로 사용한다. 난이도에 따른 자동출제 방식에서는 출제되는 문제의 난이도가 핵심이기 때문에, 무엇보다 객관적이고 공정한 방법으로 문제의 난이도를 관리하는 것이 중요하다. 또한 교양필수와 같이 많은 분반이 개설되는 과목에서는 분반별로 평가가 이루어지고, 각 분반의 평가 문제가 서로 다르게 출제되기 때문에 문제의 난이도는 특히 더 중요한 변수가 될 수 있다.

온라인 학습 시스템에서 효과적인 평가를 위해 문제를 문제은행에서 자동으로 출제하는 방식에 대한 연구와 문제은행의 문제 난이도를 효율적으로 관리하는 방식에 대한 연구가 많이 있어 왔다(D.E.Choi et al., 2000; K.Kyung-A et al., 2002; L.Hyeon-Joo et al., 2003; R,HeeYeol et al., 2004; L.Min-Kyoung et al., 2006; K.SeongKon et al., 2008; L.ChoongKwon et

al., 2012). 주로 사용하는 방식은 교수자가 문제를 문제은행에 등록할 때 난이도를 같이 등록하고, 평가를 통해 해당 문제의 정답률을 이용하여 난이도를 재조정한다. 이는 학습자들의 학습 능력을 고려할 수 있기 때문에 보다 효율적인 난이도 관리가 가능하며, 이렇게 관리된 문제의 난이도를 기반으로 공정한 문제 출제가 가능하여 많은 온라인 평가에서 이용되고 있다.

본 논문에서는 프로그래밍 언어 과목에서 온라인 평가를 위한 객관식 문제에 초점을 맞추고자 한다. 프로그래밍 언어 과목의 특성상 문제의 유형은 크게 문법을 알고 있는지를 체크하는 문제와 프로그램의 알고리즘을 분석하고 이해하여 수행 결과를 알아낼 수 있는지를 체크하는 문제로 나눌 수 있다. K.EunJung(2019) 연구에서는 프로그래밍 언어의 객관식 문제에서 일반적으로 문법을 체크하는 문제보다 알고리즘을 분석하는 문제가 실제 문제를 해결하는데 더 많은 시간을 소요하며, 학습자들은 소요시간이 더 많이 걸리는 알고리즘 분석 문제를 더 어렵게 느낀다는 것을 실험을 통해 설명하고 있다. 따라서 두 가지 유형에 대해서 교수자가 같은 난이도라고 판단한 문제일지라도 실제로 학습자들이 느끼는 난이도와는 차이가 있을 수 있다. 이를 해결하기 위해서는 문제의 유형과 난이도를 함께 고려하여 출제 문제의 개수를 결정하는 것이 무엇보다 중요하다고 할 수 있다.

이렇게 문제를 해결하는데 사용하는 소요시간의 차이는 학습자 개인이 느끼는 문제의 어렵고 쉬움의 정도에 많은 영향을 미칠 수 있기에 평가 문제의 난이도를 결정하는데 중요한 변수가 될 수 있다. 본 논문에서는 이러한 문제 해결의 소요시간을 각 개별 문제의 난이도 조정에 직접 사용하고자 한다. 단순히 문제의 유형만으로 소요시간을 구분하는 것은 각 개별 문제의 소요시간에 대한 객관적인 근거가 될 수 없다. 실제로 전체 문제 해결의 총 소요시간안에서 각각의 개별문제의 소요시간은 같은 유형과 난이도에서도 다양하게 나타날 수 있다. 따라서 보다 신뢰할 수 있는

개별 문제의 난이도를 위해서는 정답률뿐만 아니라 소요시간도 같이 고려된 난이도 조정 알고리즘이 필요하다. 이에 본 논문에서는 프로그래밍 언어 객관식 문제의 난이도 조정을 위하여 정답률뿐만 아니라 머신러닝의 로지스틱 회귀 알고리즘을 활용한 소요시간에 따른 난이도의 예측 확률값을 같이 이용한 새로운 난이도 재조정 알고리즘을 제시한다.

논문의 구성은 다음과 같다. 먼저 2장에서는 온라인 평가에서 프로그래밍 언어의 객관식 문제 출제에 대한 관련 연구를 살펴보고, 본 논문에서 제시하는 난이도 재조정 알고리즘의 개략적인 과정을 설명한다. 3장에서는 머신러닝의 로지스틱 회귀 알고리즘을 활용한 그룹별 평가를 위한 시험 문제의 난이도를 재조정하는 과정을 설명한다. 4장에서는 기존의 정답률 기반 난이도와 새로 조정된 난이도를 기반으로 그룹별 평가를 시행하고, 그 결과를 분석한다. 마지막으로 5장에서 결론으로 매듭을 짓는다.

2. 관련 연구

온라인상에서 문제해결의 역량을 강화할 수 있는 플립러닝(Flipped learning) 방식의 프로그래밍 언어 교육 방법에 대한 연구와 온라인 평가에서의 효율적인 문제 출제에 대한 연구가 많이 있어 왔다(Y.Chen et al., 2014; W.Yeomyeong et al., 2014; C.Sook Young, 2017; K.EunJung, 2019; S.Chuandong et al., 2020). 이 중에서 K.EunJung(2019) 연구에서는 프로그래밍 언어 과목의 객관식 문제 유형을 크게 문법과 알고리즘으로 구분하여 문제 유형과 난이도를 함께 고려하여 출제 문제수를 결정하는 알고리즘을 제시하였다. 여기에서 각 문제의 난이도는 출제하는 교수자가 직접 입력하여 구분하였다. 그리고 K.Kyung-A et al.(2002) 연구에 근거하여 예상 평균점수에 따른 정답률별 문제수 비율에 따라 문법 문제와 알고리즘 문제의 개수를 정하여 출제하였다. 여기서 예상되는 평균점수란 출제된 전체 문제

들의 정답률의 평균이 예상평균점수가 되도록 출제한다는 의미이다. 이때 각 유형별 문제의 난이도가 교수자의 주관적인 판단 기준에 따른 것이므로 객관적인 근거가 될수 없고, 문제의 유형별로 실시한 그룹 평가에서 전체 문제 해결의 총 소요시간의 차이를 기반으로 문법 유형의 문제를 알고리즘 분석 유형의 문제보다 더 쉬운 문제로 간주하고 출제 문제수를 결정하기 때문에 각각의 개별 문제의 소요시간에 대한 신뢰성이 떨어진다.

본 논문에서는 프로그래밍 언어의 객관식 문제에 대한 보다 객관적이고 신뢰할 수 있는 난이도를 위하여 새로운 난이도 재조정 알고리즘을 제시한다. Fig. 1에서 알고리즘의 개략적인 과정을 설명한다. 첫째, 구축한 문제은행 데이터베이스를 각 단원별 퀴즈 문제를 통하여 계속해서 업데이트한다. 문제은행에 있는 각 문제의 난이도는 전체 응시자중에서 정답을 맞춘 인원수 비율인 정답률을 기반으로 상/중/하로 관리한다. 그리고 각 문제의 소요시간은 전체 응시자들이 해당 문제를 해결하는데 사용한 소요시간의 평균을 관리한다.

둘째, 축적된 문제은행 데이터베이스에서 소요시간과 난이도의 상관관계를 분석한다. 이를 위해 머신러닝의 로지스틱 회귀 알고리즘을 이용하여 독립변수인 소요시간과 종속변수인 난이도를 학습시켜 분류 모델을 구축하고 각 난이도별 예측 확률값을 구한다.

셋째, 문제 은행에서 그룹별 평가를 위한 문제를 추출하여 새로운 문제 테이블을 생성한다. 새로 생성한 문제 테이블에 대하여 각 문제에서 사용하는 데이터 수정 및 질문 변경과 같은 전체 문제의 점검 과정을 거친다.

넷째 각 문제의 정답률과 소요시간 그리고 난이도별 예측 확률 값을 이용하여 문제 테이블의 개별 문제에 대한 난이도를 재조정한다. 마지막으로 출제 문제의 조정 난이도를 이용하여 그룹 평가를 위한 난이도별 카테고리를 설정하고 문제를 등록한다.

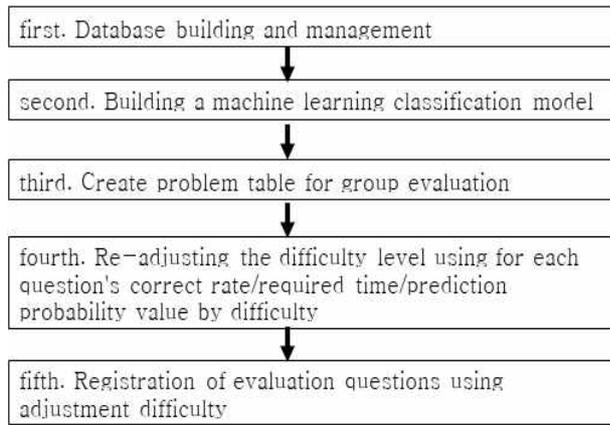


Fig. 1 Algorithm process

3. 머신러닝을 활용한 그룹별 평가 문제의 난이도 조정

3.1 문제은행의 데이터베이스 구조

C 언어의 객관식 문제 유형을 크게 2가지 유형으로 분류하여 관리한다. ‘카테고리-A’ 문제는 문법을 체크하는 문제이고 ‘카테고리-B’ 문제는 알고리즘을 분석하는 문제이다. 2개의 데이터베이스 구조는 동일하며, ‘카테고리-A’ 구조를 살펴보면 Table 1과 같다.

Table 1 Category-A (grammar)

field	datatype
num	int
chapter	int
question	string
example1	string
example2	string
example3	string
example4	string
right-answer	string
headcount	int
corr_answers	float
difficulty	char
turnaround-time	char

번호(num)는 문제를 식별하는 번호이다. 장절(chapter)은 문제가 출제된 교재의 장-절 정보이다. 문제(question)와 보기(example) 4개를 입력하고 정답(right-answer)을 입력한다. 응시자수(head count)는 해당 문제를 응시한 전체 응시자수를 누적 기록한다. 정답률(corr_answers)은 전체 응시자중에서 정답을 맞춘 인원수 비율을 저장한다. 난이도(difficulty)는 정답률에 기반하여 문제의 난이도를 상/중/하로 입력한다. 정답률이 70% 이상이면 ‘하’, 30% 이상 ~ 70% 미만이면 ‘중’, 30% 미만이면 ‘상’으로 분류한다. 소요시간(turnaround-time)은 해당 문제를 해결하는데 사용한 각 응시자들의 소요시간을 평균으로 계산하여 저장한다.

문제은행 데이터베이스는 각 장의 수업이 끝난 후에 퀴즈 형식의 문제를 출제하여, 모든 학습자들의 응시 결과를 바탕으로 구축하여 계속 관리한다. 퀴즈 문제는 한 문제씩 개별 출제하여 모든 응시자가 해당 문제를 해결하는데 사용한 소요시간을 문제별로 체크한다. Fig. 2는 각 개별 출제된 퀴즈문제 예시 그림이다.

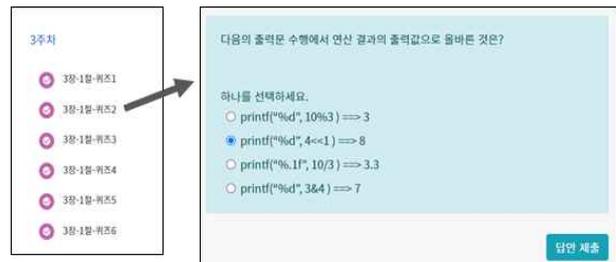


Fig. 2 Quiz test screen

모든 분반의 학생들이 퀴즈시험에 응시해야 하며, 각 분반별로 해당 퀴즈 응시 결과에 대한 보고서를 분석한다. Fig. 3은 특정 분반에서 응시한 1개의 퀴즈에 대한 응시 결과 보고서 그림이다. 엑셀 파일로 다운받아서, 전체 응시자수와 각 문제의 정답자수 그리고 각 문제의 소요시간을 추출하여 데이터베이스에 업데이트한다.

for 문의 알고리즘과 함께 증감연산자(++)의 전위형과 후위형을 같이 파악해야 하므로 응시자별로 소요시간의 차이가 많이 나는 문제 예이다. (6)의 경우에는 할당/비교/논리 연산자와 단축 연산에 대한 문법을 정확히 이해하고 있어야 하는 문제로 정답률이 높지 않으면서 동시에 소요시간도 많이 걸리는 문제 예이다. (7)의 문제는 간단한 나누기 연산의 문제로 소요시간은 매우 적은 문제이나 변수의 자료형과 연산식에서의 자동 자료형 변환에 대한 정확한 이해가 필요한 문제로 정답률은 매우 낮은 문제 예이다.

위의 예시에서 보이듯이 문제의 유형이나 정답률과 상관없이 각 개별 문제를 해결하는데 사용하는 소요시간은 다양하게 나타남을 알 수 있다. 따라서 정해진 시간안에 모든 문제를 해결해야 하는 그룹별 평가에서 각 문제의 소요시간도 문제의 난이도 결정에 중요한 변수로 처리되어야 한다.

3.3 머신러닝 모델 구축과 예측값 분석

머신러닝의 지도학습 알고리즘 중에서 회귀(Regression)는 연속적인(Continuous) 값의 훈련 데이터에 대한 특성과 상관관계를 파악한 후에, 새로운 데이터에 대하여 연속적인 숫자값을 예측하는 알고리즘이다. 분류(Classification)는 이산적인(Discrete) 값의 훈련 데이터에 대한 특성과 상관관계를 파악하고, 미지의 데이터가 어떤 종류의 값으로 분류될 수 있는지를 예측하는 알고리즘이다(Fig. 4).

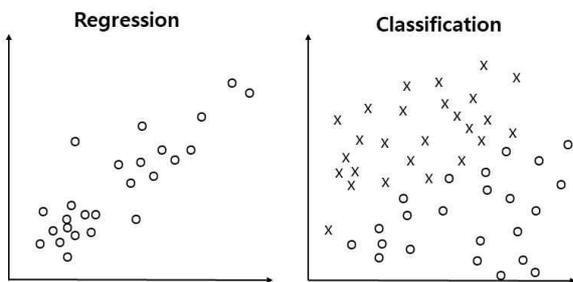


Fig. 4 supervised learning algorithm

본 논문에서는 소요시간과 난이도의 상관관계를 분석하고자 한다. 문제는행의 난이도는 정답률에 기반하여 구분되어져 있다. 소요시간과 정답률 데이터의 분포를 설명하는 Fig. 5에서 소요시간에 따른 정답률 데이터가 연속적인 값보다는 이산적인 값의 형태를 보이고 있음을 알 수 있다. 이에 본 논문에서는 분류 알고리즘을 이용하여 소요시간과 정답률 난이도의 상관 관계를 분석하고자 한다.

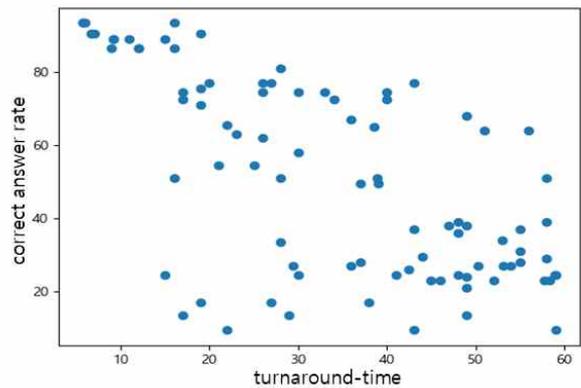


Fig. 5 Distribution of turnaround-time/correct answer rate

머신러닝의 분류 알고리즘중에서 로지스틱 회귀(Logistic Regression) 알고리즘은 독립변수의 선형 결합을 이용하여 사건의 발생 가능성을 예측하는데 사용되는 통계 기법으로 독립 변수와 종속 변수간의 관계를 구체적인 함수로 나타내어 향후 예측 모델에 사용하는 지도학습 알고리즘이다. Fig. 6는 로지스틱 회귀 분류 알고리즘을 설명하고 있다. 먼저 훈련 데이터의 특성과 분포를 나타내는 최적의 직선을 찾고 (Linear Regression), 다음으로 직선을 기준으로 새로운 입력 데이터가 어떤 범주에 해당하는지를 예측할 수 있는 확률값을 계산하여 특정 범주로 분류(Classification)하는 알고리즘이다.

일반적으로 2개의 클래스로 분류하는 이항 분류에서는 기본 임계값 0.5를 기준으로 특정 클래스로 분류하며, 3개 이상의 클래스로 분류하는 다항 분류에서는 가장 높은 확률값을

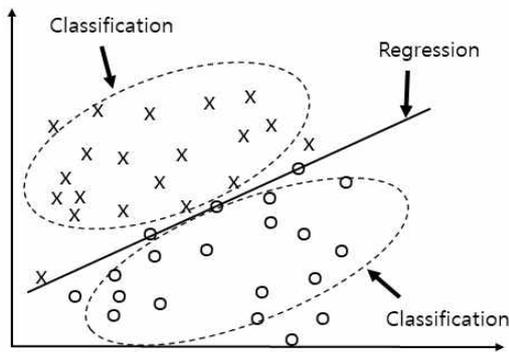


Fig. 6 Logistic Regression

Table 2 Difficulty Prediction Probability Value(example)

independent variable		dependent variable		
turnaround-time (seconds)	correct answer rate-difficulty	Prediction Difficulty		
		low(1)	middle(2)	upper(3)
		Prediction Probability Value		
9.2	1	0.9433066	0.04830809	0.0083853
5.7	1	0.9654015	0.02999257	0.00460593
21.6	1	0.5746423	0.32408445	0.10127324
43.0	2	0.09081527	0.58038558	0.32879914
37.6	2	0.25720099	0.45331377	0.28948524
45.0	3	0.0376284	0.56647224	0.39589933
58.3	3	0.00886629	0.48743953	0.5036941
:		:		:

가지는 클래스로 분류한다.

문제은행 데이터베이스에는 정답률에 기반한 각 문제의 난이도와 각 문제마다 평균 소요시간이 저장되어 있다. 여기서 개별 문제의 평균 소요시간과 정답률에 기반한 난이도의 상관관계를 분석하기 위하여 로지스틱 회귀

알고리즘의 분류 모델을 구축한다. 그리고 데이터베이스에서 독립변수인 소요시간과 종속 변수인 난이도를 추출하여 구축한 모델을 학습시킨다. 학습된 모델의 평가를 위하여 데이터베이스에서 임의의 문제 10개를 추출하여 소요시간에 대한 예측 난이도와 확률값을 분석한다. Table 2는 특정 문제의 소요시간에 따른 예측 난이도의 확률값 예시를 설명하고 있다. 먼저, 첫 번째 예시에서는 평균 소요시간(turnaround-time)이 9.2이고 정답률 난이도(correct answer rate-difficulty)가 ‘하(1)’인 문제로 94%의 매우 높은 확률값으로 예측 난이도 ‘하’로 분류되어진다. 세 번째 예시에서는 소요시간이 21.6인 정답률 난이도 ‘하(1)’인 경우로 57%의 예측 확률값으로 ‘하’로 분류되어진다. 이는 21.6의 소요시간이 난이도 ‘중’에도 가깝기 때문으로 예측된다. 네 번째 이후의 예시에서는 정답률 난이도가 ‘중(2)’이거나 ‘상(3)’인 경우로 40~50%대의 예측 확률값으로 난이도가 분류되어짐을 보이고 있다. 이는 난이도가 ‘중’이거나 ‘상’인 문제의 소요시간이 뚜렷한 차이를 보이지 않기 때문으로 예측된다.

3.4 분반별 평가를 위한 시험문제 테이블 구성과 난이도 재조정 알고리즘

분반별로 시행되는 중간고사 또는 기말고사와 같은 그룹별 평가를 위한 문제 출제는 문제은행에서 카테고리별로 선택하여 시험문제 테이블을 재구성한다. Table 3은 그룹평가를

Table 3 Exam question table for group exam

category	question	...	correct answer rate (%)	correct answer rate-difficulty	turnaround-time (seconds)	Prediction Probability Value			prediction difficulty	adjustment difficulty
						low(1)	middle(2)	upper(3)		
A	q1	...	89.0	1	9.2	0.94330662	0.04830809	0.0083853	1	1
A	q2		71.4	1	33.4	0.16700763	0.5613263	0.27166606	2	2
A	q3		51.0	2	43.0	0.09081527	0.58038558	0.32879914	2	2
A	q4		38.5	2	47.6	0.18720099	0.55331377	0.25948524	2	3
A	q5		13.5	3	39.0	0.03762842	0.56647224	0.39589933	2	3
A	q6		29.6	3	19.7	0.61464231	0.29408445	0.10127324	1	2
:	:				:	:	:	:		

위해 구성된 [시험문제] 테이블 예시이다. 먼저 문제은행에서 카테고리, 문제, 보기 4개, 정답, 정답률, 정답률 난이도, 소요시간에 대한 데이터를 추출하여 테이블을 구성한다. 새로 구성된 [시험문제] 테이블에 새로운 항목 3개를 추가한다. 첫 번째, 예측 확률값(Prediction Probability Value) 항목은 로지스틱 회귀 모델에서 소요시간에 대하여 분류될 난이도의 예측 확률값을 구하여 추가한다. 두 번째 예측 난이도(prediction difficulty)는 3개의 예측 확률값중에서 가장 높은 확률값으로 분류되는 난이도를 추가한다. 그리고 세 번째 추가되는 항목은 조정 난이도(adjustment difficulty)이다. 본 논문에서는 정답률과 머신러닝 모델의 소요시간에 대한 예측 확률값을 같이 고려하여 정답률 난이도를 재조정하여 추가한다.

난이도 재조정을 위하여 문제은행에서 3개의 난이도(상/중/하)로 구분되어 있는 정답률을 5개 구간으로 다시 구분한다. Fig. 7에서 5개 구간의 정답률을 설명하고 있다. 새로 구성된 [시험문제] 테이블에서는 정답률(%)이 0~20이면 '상', 41~60이면 '중', 81~100이면 '하'로 정한다. 이는 문제를 해결하는데 사용하는 소요시간과 상관없이 정답률만으로 문제의 어렵고 쉬움의 정도를 결정하는 것이 가능하다고 판단하기 때문이다. 반면 정답률이 21~40인 경우와 61~80인 경우에는 문제의 어렵고 쉬움의 정도를 정답률과 함께 문제해결의 소요시간도 같이 고려하여 결정하고자 한다.

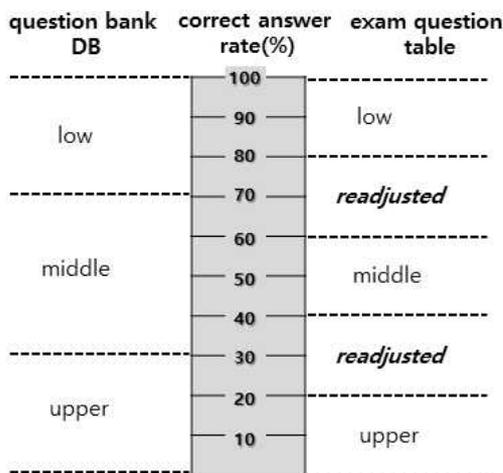


Fig. 7 difficulty ratio

Fig. 7에서 정답률(%)이 21~40인 경우와 61~80인 경우가 난이도를 재조정하는 구역이다. 정답률 21~30인 경우에는 정답률 난이도 '상'에 해당하는 범위이기 때문에 난이도 '상'으로 분류되는 예측 확률값의 임계시간을 기준으로 분류하고, 정답률 31~40인 경우에는 정답률 난이도 '중'에 해당하는 범위이므로 난이도 '중'으로 분류되는 예측 확률값의 임계시간을 기준으로 분류한다. 그리고 정답률 61~70인 경우에는 난이도 '중'으로 분류되는 예측 확률값의 임계시간을 기준으로 분류하고, 정답률 71~80인 경우에는 난이도 '하'로 분류되는 예측 확률값의 임계시간을 기준으로 분류한다. 여기서 임계시간은 예측확률값의 기준 임계값에 해당하는 시간으로 정한다. Table 2의 예시처럼 3개 이상의 클래스로 분류하는 다항 분류에서는 높은 확률값을 가지는 특정 클래스로 분류한다. 이 경우 3개의 클래스중에서 1개의 확률값이 뚜렷한 차이를 보이면서 특정 클래스로 분류되는 경우도 있고, 3개의 확률값이 거의 비슷하면서 근사한 차이로 특정 클래스로 분류되는 경우도 발생한다. 이런 경우에는 모든 확률값이 기본 임계값 0.5에 미치지 못하는 경우도 발생할수 있기 때문에, 본 논문에서는 각 난이도의 기준 임계값을 (최고 확률값-최저 확률값)/2 로 계산하여 정한다.

Table 3에서 난이도가 '하(1)'인 q1 문제는 소요시간이 9.2초이므로 분류모델에서 94%의 높은 확률값으로 '하(1)'로 구분하였다. 또한 정답률이 89%이기 때문에 조정없이 그대로 '하(1)'로 정한다. q2 문제는 정답률 71.4%인 '중' 문제로 소요시간 33.4에 의해 분류 모델에서 '중(2)'으로 분류하였다. 그리고 재조정 구역에 해당하는 '하' 문제이므로 '하'의 예측 확률값 임계시간을 기준으로 소요시간 33.4는 '중(2)'으로 조정된다. q4 문제는 정답률 38.5%인 '중' 문제이다. 소요시간 47.6으로 인해 분류 모델에서는 난이도 '중(2)'으로 구분하였지만, '중'의 예측 확률값 임계시간을 기준으로 재조정했을때 '상(3)'으로 조정되어진다. 마지막 q6 문제는 정답률이 29.6%인

난이도 '상'인 문제로, 소요시간 19.7에 따라 분류 모델에서는 '하(1)'로 분류되었다. 그리고 재조정 구역에 해당하는 '상' 문제이므로 '상'의 예측 확률값 임계시간을 기준으로 재조정했을 때, '중(2)'으로 조정되어진다.

Fig. 8에서 난이도 재조정 알고리즘을 대략적으로 설명하고 있다.

```

① load the question bank database.
② independent variable = turnaround-time
   dependent variable = difficulty
③ model = Logistic Regression Classification Algorithm
④ model.learning(independent variable, dependent variable)
⑤ Upper/Middle/Low Threshold
   = (max(Prediction Probability Value)+min(Prediction Probability Value))/2
   Upper/Middle/Low threshold time = Time required for each threshold
⑥ [Exam Questions] Create table.
⑦ Prediction Probability Value = model. Prediction Probability Value(turnaround-time)
   Prediction Difficulty = model. Prediction Difficulty (turnaround-time)
⑧ Repeat the following for the entire problem.
   if correct answer rate <= 20, Adjustment Difficulty='upper'
   if correct answer rate > 20 and correct answer rate <= 30,
       if turnaround-time > upper_threshold time, Adjustment Difficulty='upper'
       else Adjustment Difficulty='middle'
   if correct answer rate > 30 and correct answer rate <= 40,
       if turnaround-time > middle_threshold time, Adjustment Difficulty='upper'
       else Adjustment Difficulty='middle'
   if correct answer rate > 40 and correct answer rate <= 60, Adjustment Difficulty='middle'
   if correct answer rate > 60 and correct answer rate <= 70,
       if turnaround-time > middle_threshold time, Adjustment Difficulty='middle'
       else Adjustment Difficulty='low'
   if correct answer rate > 70 and correct answer rate <= 80,
       if turnaround-time > low_threshold time, Adjustment Difficulty='middle'
       else Adjustment Difficulty='low'
   if correct answer rate > 80, Adjustment Difficulty='low'
    
```

Fig. 8 Difficulty Adjustment Algorithm

① 문제 은행에서 모든 데이터를 읽어들이는다. ② 독립변수인 소요시간과 종속변수인 난이도를 추출한다. ③ 로지스틱 회귀 분류 모델을 구축한다. ④ 독립변수와 종속변수를 이용하여 모델을 학습시킨다. ⑤ 학습된 모델에서 소요시간 0초부터 60초사이에서 0.5초 간격으로 각 난이도별 예측 확률값을 구한다. 이를 이용하여 난이도 상/중/하의 기준 임계값을 구하고, 각각의 기준 임계값에 해당하는 기준 임계시간을 정한다. ⑥ 문제은행 데이터베이스에서 문제를 추출하여 그룹 평가를 위한 [시험문제] 테이블을 생성한다. ⑦ [시험문제] 테이블의 소요시간을 이용하여 머신러닝 모델에서 예측 난이도와 분류를 위한 예측 확률값을 구하여 새로운 항목으로 추가한다. ⑧ 전체 문제에 대해서 정답률과 소요시간 그리고 상/중/하

난이도의 기준 임계시간을 기준으로 개별 문제의 난이도를 재조정한다.

4. 실험 결과 및 분석

4.1 로지스틱 회귀 모델 구축과 학습

본 논문에서 제시한 알고리즘을 적용했을 때, 정답률 난이도가 어떻게 조정되는지를 분석하기 위해서 파이썬 환경의 머신러닝에서 유용하게 사용할 수 있는 공개 모듈인 사이킷런(scikit-learn) 라이브러리를 이용하였다. 사이킷런 패키지의 선형모델에서 제공하는 분류 알고리즘을 위한 클래스중에서 LogisticRegression 클래스를 사용하여 머신러닝 모델을 구축하였다. 그리고 LogisticRegression 클래스의 멤버 메소드 .predict()을 이용하여 예측되는 분류값을 구하였고, .predict_proba()을 이용하여 해당 레이블로 분류된 확률값을 구하였다.

실제 문제를 해결한 소요시간과 정답률에 기반한 난이도의 상관관계를 분석하기 위하여 Table 1의 문제은행 데이터베이스에서 독립변수인 '소요시간'과 종속변수인 '난이도'를 추출하여 구축한 인공지능 모델을 학습하였다.

4.2 정답률 난이도와 예측 난이도 분석

문제은행에서 그룹평가를 위한 문제를 90개 추출하여 [시험문제] 테이블을 구성하였다. 이때 정답률에 기반한 정답률 난이도의 '하'-문제 30개, '중'-문제 30개, '상'-문제 30개를 랜덤하게 추출하였다. 구성한 [시험문제] 테이블의 소요시간에 대하여 머신러닝 모델의 예측 난이도를 구하여 정답률 난이도와 비교 분석하였다. 머신러닝 모델에서 예측한 난이도의 개수는 Table 4와 같다. 표에서 보듯이 소요시간에 대한 난이도를 학습한 머신러닝 모델에서는 예측 난이도를 '하' 문제의 난이도와 '상' 문제의 난이도로 많이 분류하였다. 이는 난이도 '중'에 해당하는 많은 문제들의 소요시간이 난이도 '하' 또는 '상'

문제의 소요시간과 뚜렷이 구별될 만큼의 차이를 보이지 않음을 짐작할 수 있다.

Table 4 Number of Prediction Difficulties

	correct answer rate difficulty	prediction difficulty
low (1)	30	38
middle (2)	30	18
upper (3)	30	34

4.3 난이도별 예측 확률값 분석

머신러닝 모델에서 소요시간에 따른 난이도를 Table 4와 같이 분류한 이유를 알아보기 위하여 소요시간 0초부터 60초사이에서 0.5초 간격으로 각 난이도별 예측 확률값을 구하여 그래프로 분석하였다. Fig. 9는 난이도-’하’로 분류한 예측 확률값이다. 그림에서 보이듯이 최고 확률값이 거의 1에 가까우며, 소요시간이 대략 15초 이전일때에는 높은 확률값을 가진다는 것을 알 수 있다. 이는 난이도가 ’하’이면서 소요시간이 15초이전인 문제가 많이 학습되어 높은 확률값으로 난이도 ’하’로 분류됨을 의미한다. Fig. 10과 Fig. 11은 난이도-’중’과 난이도-’상’으로 분류한 예측 확률값이다. 2개의 그림에서는 최고 확률값이 0.5~0.6 사이에 있고, 대략 30초 이후에서는 소요시간이 난이도-’중’과 ’상’의 문제를 뚜렷하게 구별할 만큼의 차이를 보이지 않음을 알 수 있다. Fig. 12에서는 3개의 예측 확률값을 같이 분석한 그래프이다. 그림에서 ①의 위치를 살펴보면, 소요시간이 대략 28초일때까지 3개의 레이블 중에서 난이도-’하’의 예측 확률값이 가장 높음을 알 수 있다. 그리고 이후부터 ②의 위치인 대략 40초까지는 난이도-’중’의 예측 확률값이 가장 높음을 알 수 있다. 하지만 이 구간에서도 난이도-’중’과 ’상’의 소요시간이 뚜렷한 차이를 보이지는 않는다. 그 결과 Table 4에서 보이듯이 예측 확률값으로 분류한 난이도의 개수가 ’중’ 보다는 ’하’ 와 ’상’이 더 많이 분류되었음을 알 수 있다.

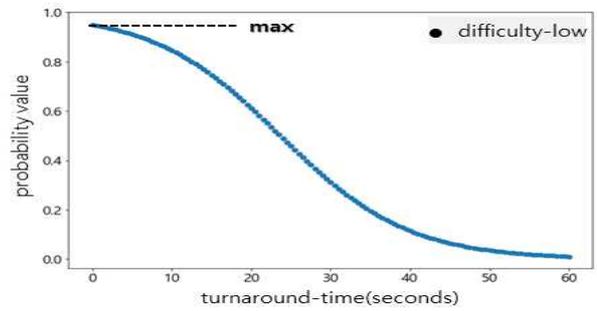


Fig. 9 'low' predicted probability value

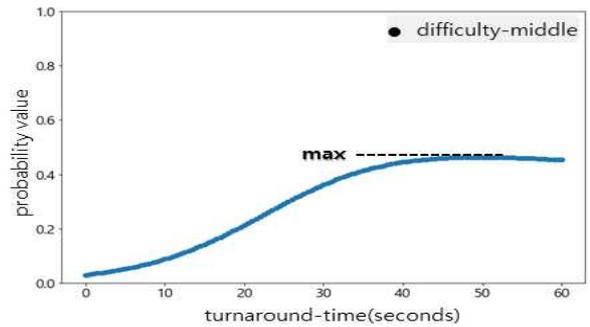


Fig. 10 'middle' predicted probability value

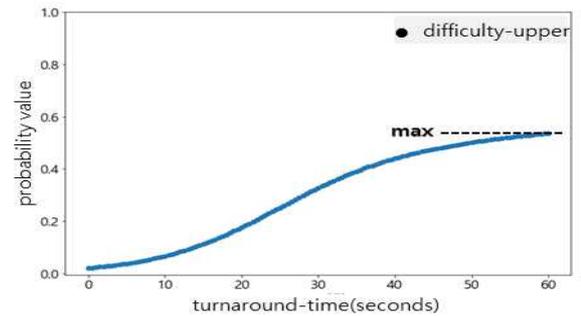


Fig. 11 'upper' predicted probability value

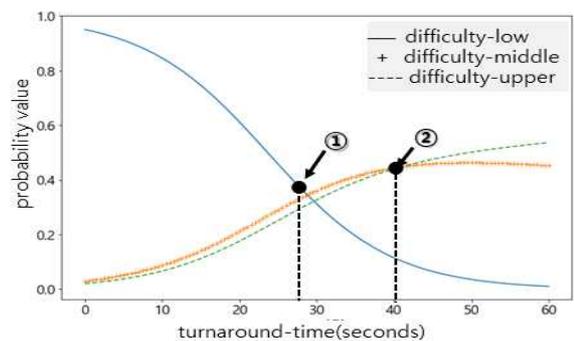


Fig. 12 predicted probability value

4.4 기준 임계값에 따른 조정 난이도 분석

Fig. 7에서 설명했듯이 정답률이 21~40, 61~80인 경우에는 머신러닝의 예측 확률값을 이용하여 난이도를 재조정한다. 이를 위해 난이도 상/중/하의 예측 확률값에 대한 기준 임계값을 (최고 확률값-최저 확률값)/2 로 계산하여 정한다. Fig. 13에서 기준 임계값을 설명하고 있다. ①은 난이도-’하‘의 기준 임계값 위치이다. 이 위치에서의 기준 임계 확률값은 0.47297, 기준 임계시간은 24.5 이다. 즉 소요시간 24.5초를 기준으로 난이도를 ’중‘과 ’하‘로 분류한다. ②는 난이도-’중‘의 기준 임계값 위치이다. 기준 임계 확률값은 0.25178, 기준 임계시간은 22.5이다. 즉 소요시간 22.5초를 기준으로 난이도를 ’상‘, ’중‘, ’하‘로 분류한다. ③는 난이도-’상‘의 기준 임계값 위치이다. 기준 임계 확률값은 0.28262, 기준 임계시간은 27.0이다. 즉 소요시간 27.0초를 기준으로 난이도를 ’상‘ 또는 ’중‘으로 분류한다.

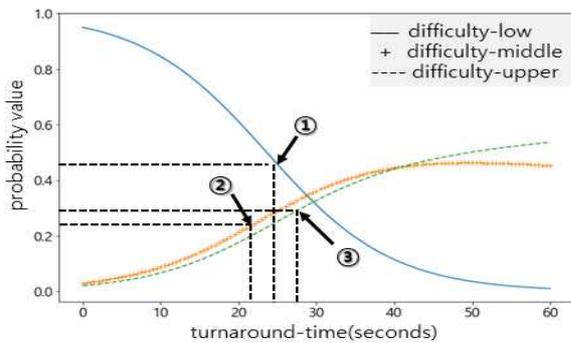


Fig. 13 Difficulty Prediction Probability Value

위의 임계시간과 정답률을 기반으로 난이도를 재조정된 결과를 Fig. 14에서 설명하고 있다. 최초 문제은행에서 추출한 90개의 시험문제는 난이도가 ’하‘-30개, ’중‘-30개, ’상‘-30개였다. 여기에 머신러닝의 소요시간에 따른 예측 확률값과 정답률을 같이 고려하여 난이도를 재조정된 결과 ’하‘-24개, ’중‘-25개, ’상‘-41개로 조정되었음을 확인할 수 있다.

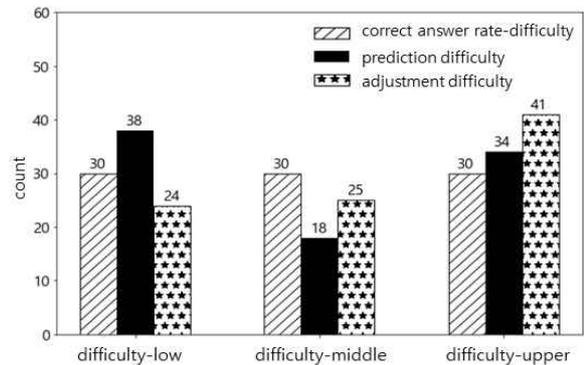


Fig. 14 Adjusted number of difficulties

4.5 분반별 그룹평가 결과 분석

정답률에 기반한 난이도의 문제와 본 논문에서 제시하는 알고리즘으로 새로 조정된 난이도의 문제를 이용하여 그룹별 평가를 수행한 결과를 비교 분석하였다. 실험은 현재 A대학에서 운영하는 교양필수 과목인 컴퓨터 프로그래밍 수업 6개의 분반에서 퀴즈시험 형식으로 2회 실시하였다. 퀴즈시험 결과는 5%의 비율로 계산하여 전체 총점에 추가점으로 포함하였다. 첫 번째 퀴즈시험에서는 정답률 난이도를 기반으로 문제를 출제하였고, 두 번째 퀴즈시험에서는 조정된 난이도를 기반으로 문제를 출제하였다. 출제 문제수는 K.Kyung-A et al.(2002) 연구에 근거하여 예상 평균점수 60점에 따른 문제수 비율에 따라, 전체 20문제 중에서 상-3문제, 중-7문제, 하-10문제를 출제하였다. 이때 Fig. 14의 결과에서 난이도 조정이 일어난 문제를 두 번의 퀴즈시험에 각각 포함하여 출제하였다. 퀴즈 응시 시간은 20분으로 제한하였으며, 한 문항당 5점으로 전체 100점으로 처리하였다. Table 5와 Fig. 15에서 두 번의 퀴즈시험에 대한 각 분반별 평점과 그 차이를 설명하고 있다.

Table 5 Average of each class

class	head count (people)	quiz exam average			total average
		Correct answer rate Difficulty ①	Adjustment Difficulty ②	②-①	
1	38	69.57	76.04	6.48	68.34
2	37	73.66	80.81	7.15	75.79
3	38	66.88	71.36	4.48	64.12
4	33	70.45	74.74	4.29	62.87
5	36	61.87	62.79	0.92	59.91
6	38	67.39	69.07	1.68	61.56

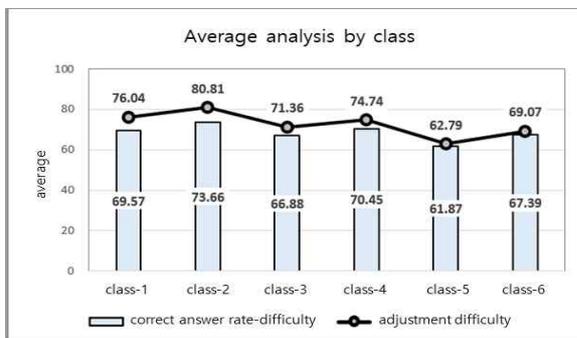


Fig. 15 Average analysis of each class

Fig. 15에서 보이듯이, 실험한 모든 분반에서 정답률 난이도의 문제로 실시한 퀴즈의 평점보다 조정된 난이도의 문제로 실시한 퀴즈의 평점이 조금 더 높게 나오는 것을 확인할 수 있었다. 이는 정답률 난이도가 '하' 또는 '중'이었던 문제중에서 조정 난이도 '중' 또는 '상'으로 바뀐 문제들이 전체 평점 상승에 영향을 준 것으로 분석할 수 있다. 특히 학습 능력이 낮은 분반보다 학습 능력이 높은 분반에서 더 많은 평점의 차이를 보이는 것으로 분석되었다. Table 5에서 중간고사와 기말고사를 모두 합친 전체 평점을 살펴보면, 분반-2는 75.79이고 분반-5는 59.91이다. 전체적으로 학습 능력이 높은 그룹인 분반-2의 경우에는 소요시간이 고려된 조정 난이도의 문제들이 평점 상승에 많은 영향을 준 반면에 학습 능력이 낮은 그룹인 분반-5의 경우에는 그 차이가 크지 않음을 보이고 있다. 따라서 정해진 시간안에 모든 문제를 해결해야 하는 평가시험에서 문제를 해결하는데 소요되는

시간은 해당 문제의 어렵고 쉬움의 정도를 결정하는데 중요한 변수로 작용할수 있으며, 학습 능력을 측정하는 평점에 영향을 미친다는 것을 확인할 수 있었다.

5. 결론

온라인 학습 시스템에서 평가를 위한 문제 출제는 카테고리별로 저장된 문제은행에서 자동으로 출제하는 방식을 주로 많이 사용한다. 특히 교양필수와 같이 전교생을 대상으로 개설되는 과목의 경우에는 분반별로 평가가 이루어지고, 각 분반별 평가 문제가 서로 다르게 출제되는 경우가 많다. 이런 경우에는 무엇보다 문제의 난이도가 핵심이기에 보다 객관적이고 신뢰할수 있는 방법으로 난이도를 관리하는 것이 중요하다. 본 논문에서는 프로그래밍 언어 과목의 객관식 문제에 대한 난이도 조정에 초점을 맞추어 설명하였다. 프로그래밍 언어 과목의 특성상 문제에 따라서 정답률과 상관없이 해당 문제를 해결하는데 소요되는 시간은 다양하게 나타날 수 있다. 이러한 문제 해결의 소요시간은 제한된 시간 안에서 이루어지는 평가 시스템에서 평가자 개개인마다 느끼는 문제의 어렵고 쉬움의 정도에 많은 영향을 미칠수 있기에 문제의 난이도 결정에 중요한 변수가 될 수 있다. 이에 본 논문에서는 개별 문제의 난이도를 결정함에 있어서 응시자의 정답률과 각 문제를 해결하는데 소요되는 시간도 함께 고려한 새로운 난이도 재조정 알고리즘을 제시하였다.

알고리즘의 과정으로 먼저 문제은행 데이터 베이스에 퀴즈 형식으로 실시하는 학습자들의 응시 결과를 바탕으로 각 문제의 정답률과 정답률에 기반한 난이도 그리고 문제 해결의 소요시간에 대한 정보를 구축하고 계속 업데이트하며 관리하였다. 다음으로 데이터베이스에서 독립변수인 소요시간과 종속변수인 난이도를 추출하여 머신러닝의 분류 모델을 구축하고, 각 난이도별 예측 확률값을 구하여 분류를 위한 기준 임계 시간을 설정하였다.

다음으로 그룹평가를 위한 시험문제 테이블을 생성하고, 시험 문제의 질문 변경 및 데이터 수정 등의 점검과정을 거친 다음에 각 문제의 정답률과 소요시간을 이용하여 난이도별 분류 임계 시간을 기준으로 난이도를 재조정하였다.

마지막으로 기존 정답률 기반의 난이도 문제와 조정된 난이도 문제를 각각 이용하여 분반별 평가를 실시하여 그 결과를 분석해 본 결과, 조정된 난이도의 문제가 전체 분반에서 평점 상승에 많은 영향을 미치는 것을 확인할 수 있었다. 이에 프로그래밍 언어 과목의 특성상 각 개별 문제의 난이도를 조정함에 있어 정답률뿐만 아니라 문제 해결의 소요시간도 같이 고려하는 알고리즘이 기존의 정답률만 고려하는 알고리즘에 비해 보다 객관적이고 신뢰할 수 있는 방법임을 확인할 수 있었다.

References

- C.SookYoung (2017) , Design and Application of an Instructional Model for Flipped learning of Programming Class, *The Journal of Korean association of computer education*, Vol.20 No.4, 27-36.
- D.E.Choi, H.J.Seo, K.S.Park, J.Y.Lee (2000), A Design and Implementation of Dynamic Test Generating System , *Korean Institute of Information Scientists and Engineers, Proceedings of academic presentations*, Vol.27 No.1B, 690-692.
- R,HeeYeol, K,EunJung (2004), Degree of Difficulty Adjustment Algorithms of Selection Question using Education Ability in WBI, *Journal of the Korea Industrial Information Systems Research*, Vol.9 No.4, 47-55
- K.KyungA, C.EunMan (2002) , Automated Selection System of Examination Questions in Web-Based Instruction, *KIPS Transactions on Computer and Communication Systems*, Vol.9 No.3, 301-310.
- K.EunJung (2004) , Examination Questions Selection Algorithm in Web-based Engineer Test Education System, *Journal of the Korea Industrial Information Systems Research*, v.9, no.3, 11-18
- K.SeongKon, L.SangKwan, K.EunJung (2008), Dynamic Adjustment Policy of degrees of difficulty for E-learning Databank Based Selection System, *Korean Institute Of Maritime information & Communication Science*, Vol.12 No.12, 2232-2238.
- K.EunJung (2019), A Study on Difficulty Equalization Algorithm for Multiple Choice Problem in Programming Language Learning System , *The Journal of Korean association of computer education*, Vol.22 No.3, 55-65.
- L.HyeonJoo, L,MiSook, H.SeungMi, L.ChanHee,Jung, Soon-Ho, (2003), Web-based Automatic Question-Issuing System Using Level Estimation for Learners , *KIPS Transactions on Computer and Communication Systems*, Vol.10 No.5, 579-588.
- L.MinKyoung, K.SooYong (2006), a Web-Based Item pool system for the level-learning, *Korean Institute of Information Scientists and Engineers, Proceedings of academic presentations*, Vol.33 No.2A, 103-107.
- L.ChoongKwon, Y.Sangjin, J,Sangmin (2012), A Study of the Measurement of the Perceived Distances among Programming Languages, *Journal of the Korea Industrial Information Systems Research*, v.17, no.1, 95-104
- S.Chuangong, W.Haifeng, Y.Bin, Z.Wei (2020), Online and Offline Teaching Mode of C Language Programming, *Proceedings of the 2020 The 2nd World Symposium on Software Engineering*, 207-210

W.Yeomyeong, B.Jiwoong, S.Jaemin, Y.Jinyeong, L.Sangjun (2014), Design and Implementation of the Web-based Learning System for C Programming Language, *KIISE Transactions on Computing Practices (KTCP)*, Vol.20 No.12, 640-645.

Y.Chen, Y.Wang, Kinshuk, & Chen, N.S. Ch (2014). Is FLIP enough? Or should we use the FLIPPED model instead?. *Computers & Education*, Vol 79, 16-27.



김 은 정 (EunJung Kim)

- 정회원
- 국립 경상대학교 전자계산학과 공학석사
- 국립 경상대학교 전자계산학과 공학박사
- (주)LG전자 멀티미디어연구소 연구원
- 부산외국어대학교 컴퓨터공학과 비정년 전임강사
- 부산가톨릭대학교 컴퓨터공학과 비정년 전임강사
- (현재) 부산대학교 교양교육원 강사
- 관심분야: 프로그래밍교육, 원격교육, 난이도 조정