

РЕАЛИЗАЦИЯ МЕХАНИЗМОВ ПОДДЕРЖКИ ИНДИВИДУАЛИЗИРОВАННОГО ОБУЧЕНИЯ В ИНТЕЛЛЕКТУАЛЬНЫХ ОБУЧАЮЩИХ СИСТЕМАХ

Каяшев М.В.

*Каяшев Михаил Владиславович - студент,
кафедра программной инженерии,
Высшая школа ИТИС,
Казанский (Приволжский) федеральный университет, г. Казань*

Аннотация: в данной статье рассматриваются инструменты для повышения индивидуализации процесса обучения с целью обеспечения высокого качества подготовки студентов в высших учебных заведениях.

Ключевые слова: интеллектуальная обучающая система, рекомендательная система, коллаборативный, индивидуализация, образование, качество, компетенции, линейная регрессия, регуляризация, функция-хендлер.

Преподавание информационных технологий в высших учебных заведениях сталкивается с целым рядом проблем. Преподавателю сложно уделить достаточно внимания каждому студенту, глубоко оценить уровень знаний, проявить индивидуальный подход к подаче материала, предоставить качественную и своевременную обратную связь. Современная IT-индустрия нуждается в высококвалифицированных, компетентных специалистах. Вследствие этого, необходимо повышать качество подготовки студентов, учитывая ограниченность в ресурсах.

Одни из самых трудоемких и требующих особого внимания задач в работе педагога - это отследить пробелы в знаниях студента и помочь устранить их. Некоторые такие функции преподавателя, отвечающие за индивидуализацию процесса обучения, можно возложить на интеллектуальную автоматизированную систему. Автоматизация решения этих педагогических задач может эффективно решить проблему качества подготовки студентов.

При решении поставленной задачи и определении требований было решено разработать рекомендательную систему гибридного типа, которая включает в себя как контентную фильтрацию, так и коллаборативную. Такой подход позволяет решить ряд проблем, имеющих место при применении этих методов по отдельности [1]. Также были сформированы требования к заданиям. Каждое задание имеет следующие характеристики: тема, условие, сложность, навыки, оценка.

Для реализации поставленной задачи была выбрана методология, основанная на регрессионной модели. Данная модель является легко расширяемой (чтобы добавить новый алгоритм рекомендации, достаточно лишь написать еще одну функцию) и гибкой (например, можно самостоятельно настраивать вес каждой функции-рекомендации).

Основой методологии являются *функции-хендлеры*, отвечающие за оценку заданий для рекомендаций. Каждый хендлер принимает в качестве аргумента модель студента со всеми его характеристиками (выполненные задания, набранные баллы) и предлагает на выходе набор заданий по заранее определенному алгоритму. Каждому заданию присваивается числовое значение (вес) от 0 до 1 в зависимости от того, насколько данное задание соответствует критерию хендлера. Чем больше вес, тем выше вероятность того, что это задание будет показываться для студента. На каждую предложенную рекомендацию отслеживается реакция студента: задание, предложенное каким хендлером, он решил, или посмотрел, или просто проигнорировал рекомендацию. В зависимости от реакции студента на рекомендации пересчитываются веса хендлеров. После того, как каждый хендлер определит набор заданий для рекомендации, происходит их комбинирование и подготовка к показу для студента с помощью линейной регрессии [2].

Со временем вес одного из хендлеров может постепенно возрастать, а других – уменьшаться. Это может привести к ситуации, когда рекомендуются задания только из одного хендлера. Чтобы избежать данной проблемы, нужно производить переобучение с помощью регуляризации. Для данной задачи была выбрана лассо-регуляризация, которая уменьшает все веса β_i , а те, что и так были относительно небольшими, становятся равны нулю [3].

При определении типов рекомендаций были разработаны следующие функции-хендлеры, которые рекомендуют задания:

1. По интересу к теме (Приоритет в выборе отдается заданиям из тем, в которых студент решил больше всего заданий).
2. С повышенной сложностью (Рекомендуются задания повышенной сложности из тем, в которых

студент решил больше всего заданий)

3. От похожих студентов (Хендлер коллаборации. Приоритет в выборе отдается заданиям, которые решали студенты, имеющие похожий набор характеристик).

4. С неосвоенными навыками (Приоритет в выборе отдается заданиям, покрывающим наибольшее число неосвоенных студентом навыков в теме).

5. Ближайшие к дате сдачи (Рекомендуются задания, которые нужно выполнить в ближайший срок).

Для реализации задачи был выбран язык программирования Python с использованием библиотек SciPy и ScLearn, которые облегчают работу с линейной регрессией. Программное решение было разработано и выложено в открытый доступ на сайте: <https://github.com/mishauni13/recommender-system>.

Данное решение в виде рекомендательной системы поможет преподавателю в процессе обучения студентов. Педагог в любое время может отследить успехи студента, определить темы, в которых студент разбирается лучше всего, а также выявить проблемные задания, в решении которых студент испытывает затруднения. Также студент постоянно получает рекомендации, которые помогают ему улучшить знания по малоизученным темам.

Список литературы

1. Рекомендательные системы. [Электронный ресурс]. Режим доступа: <https://www.ibm.com/developerworks/ru/library/os-recommender1/index.html/> (дата обращения: 15.06.2017).
2. Рекомендательные системы в онлайн-образовании. [Электронный ресурс]. Режим доступа: <https://habrahabr.ru/company/stepic/blog/307670/> (дата обращения: 20.06.2017).
3. L1-регуляризация линейной регрессии. [Электронный ресурс]. Режим доступа: [http://www.machinelearning.ru/wiki/images/7/7e/VetrovSem11_LARS.pdf/](http://www.machinelearning.ru/wiki/images/7/7e/VetrovSem11_LARS.pdf) (дата обращения: 20.06.2017).