ФИЗИКА И МАТЕМАТИКА

ПРОБЛЕМА ВЫБОРА ПРИ НЕЧЕТКОМ ОТНОШЕНИИ ПРЕДПОЧТЕНИЯ НА МНОЖЕСТВЕ АЛЬТЕРНАТИВ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Базылев Д. А.

Калугина М. А. – канд. физ.-мат. наук, доцент

В теории принятия решений центральное место занимают задачи многокритериального выбора. В докладе описывается принцип работы процедуры для определения выбора при нечетком отношении предпочтения.

Суть проблемы выбора при нечетком отношении предпочтения в том, что все данные, которыми мы располагаем и используем для принятия решения, носят вероятностный характер, т.е. верны только с некоторой вероятностью. Задача принятия решений — одна из самых распространенных в любой предметной области. Например, в экономике при планировании стратегий, подборе персонала, а также в технических областях, когда необходимо разработать проектное или конструкторское решение.

Для изучения проблемы выбора на практике было выбрано конкретное направление: досуг (развлечения). Программное средство должно предложить пользователю фильмы, концерты, события, которые проходят в данный момент и, скорее всего, заинтересуют его. С этой целью была изучена предметная область и разработаны алгоритмы для решения поставленной задачи.

Для описания модели данных используем ее представление в виде графа (рисунок 1).

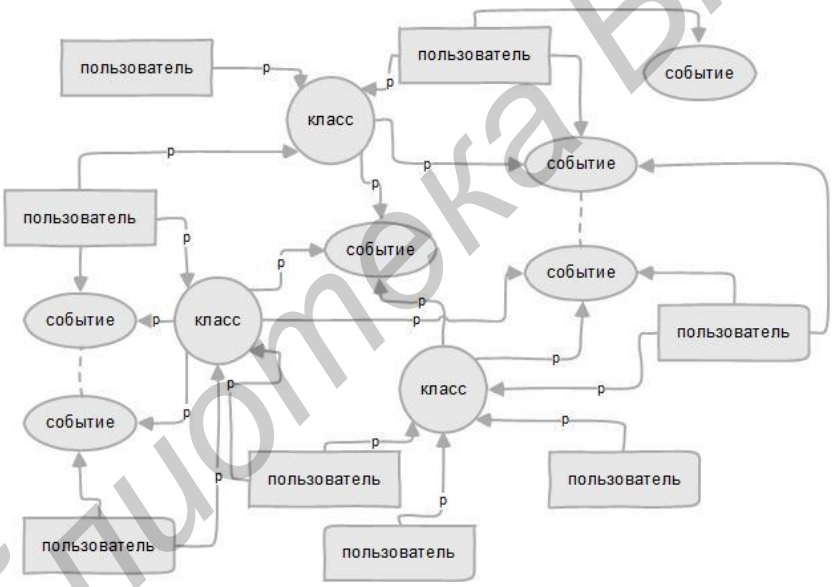

Рис. 1 – Представление данных в виде графа

Как видно из схемы, определены 3 основные сущности.

− *Пользователь*. Представляет собой лицо, которому будут предложены события, и ему они, скорее всего, понравятся (т. е. с большой вероятностью).

− *Событие*. Представляет собой некоторое мероприятие, например, кинофильм, концерт, или сеанс катания на катке.

− *Класс пользователей*. Группа пользователей, которая обладает общими признаками, и, с некоторой вероятностью, имеет одинаковые предпочтения. Начальные классы выделяются при помощи проведения опроса среди потенциальных пользователей ПС. Впоследствии они могут видоизменяться, а также могут выделяться новые.

Между введенными сущностями определены 4 типа связей.

− *Пользователь -> событие*. Данная связь означает, что пользователь побывал на данном мероприятии. Эта связь может нести как позитивный смысл (оценка – «понравилось»), так и негативный оттенок (оценка – «не понравилось»).

− *Пользователь -> класс пользователей*. Эта связь носит вероятностный характер и означает, что с некоторой вероятностью *p* данный пользователь принадлежит к указанному классу.

− *Класс пользователей -> событие*. Вероятностный характер этой связи означает, что с некоторой вероятностью *p* связанное событие понравится пользователям, которые относятся к данному классу.

− *Событие* <-> *событие*. Данная связь объединяет наиболее похожие события между собой.

При помощи представленной модели данных можно полностью описать взаимоотношение сущностей в ПС. Предлагая пользователю события, которые им понравятся, и, собирая у них обратную связь, мы выстраиваем граф отношений, в котором пользователи с помощью обратной связи корректируют свои вероятности отношения к определенным классам, а также вероятности связи “класс – событие”, тем самым делая предугадывание системы более точным. Благодаря связям “событие – событие” мы можем более быстро внедрять новые событие в граф данных на основе уже имеющихся. Пользователю предлагаются события, у которых максимальное произведение вероятностей между связями пользователь->класс и класс - >событие. После обратной связи данные вероятности будут подкорректированы в ту или иную сторону.

Программное средство состоит из модулей (независимых частей), которые взаимодействуют между собой только посредством открытых интерфейсов. Все внутренние алгоритмы и данные модулей инкапсулированы внутри них. На рисунке 2 изображена концептуальная архитектура ПС. Данная архитектура позволяет легко заменять и расширять отдельные модули, не затрагивая всю систему, масштабировать систему горизонтально и вертикально.

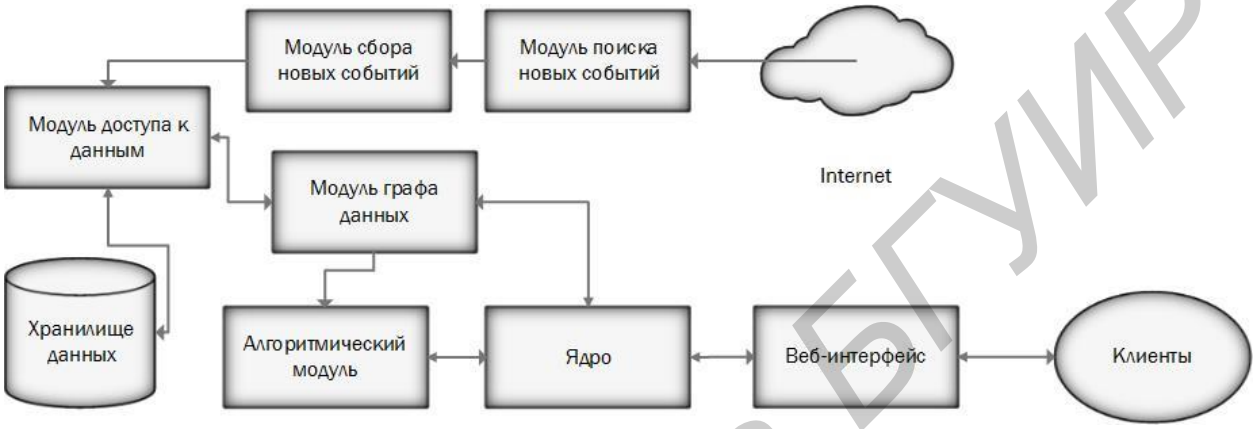

Рис. 2 – Концептуальная архитектура ПС

Краткое описание модулей:

- *Модуль поиска новых событий* содержит логику поиска новых событий в сети интернет. Он посещает электронные афиши и иные сайты, на которых можно найти актуальные события. Все найденные события он передает в *модуль сбора новых событий*.

- *Модуль сбора новых событий* содержит логику по фильтрации событий, которые пришли из *модуля поиска новых событий*. Все новые события проверяются на дубликат (на случай если данное событие уже есть в системе). Здесь же могут быть отсеяны события по каким-то другим правилам, затем оставшиеся события приводятся к общему виду и передаются в *модуль доступа к данным* для сохранения в системе.

- *Модуль доступа к данным* определяет всю логику работы с конкретным хранилищем данных. Это позволяет остальной системе не зависеть от особенностей конкретного хранилище данных, а также довольно быстро менять хранилище данных на более подходящее в случае необходимости.

- *Модуль графа данных* строит вышеприведенный (рисунок 1) граф данных в оперативной памяти на основе данных из *модуля доступа к данным*.

- *Алгоритмический модуль* содержит алгоритмы для определения выбора.

- *Ядро* содержит основную логику работы: вызовы алгоритмов, обработка пользовательских действий, логирование.

- *Веб-интерфейс* контролирует взаимодействие пользователя с программой.

Данное программное средство позволяет решать задачи по принятию решений не только в пределах конкретного направления – досуга, но также является готовым шаблоном для построения решений в других направлениях. Из оставшихся нерешенных проблем стоит выделить проблему начальной классификации новых пользователей. Т.к. мы классифицируем пользователя и предлагаем новые события на основе оценок данного пользователя других событий, то при малом количестве таких оценок вероятность того, что предлагаемые события будут оценены пользователем положительно, меньше, чем у пользователя с большим количеством таких оценок. Одно из возможных решений данной проблемы заключается во введении обязательного опроса всех новых пользователей, результаты которого помогут более точно классифицировать нового пользователя.

Список использованных источников

1. Martin Fowler. Patterns of Enterprise Application Architecture.− Addison Wesley, 2002. – 560 с.

2. Разработка моделей многокритериального выбора альтернатив на основе нечетких множеств второго порядка для решения экономических задач [Электронный ресурс]. – Режим доступа: http://www.scienceforum.ru/2013/235/3388. – Дата доступа: 01.03.2014.