Исследование подходов к построению самообучающихся экспертных систем с нечеткой логикой

В.А. Бакаев[1]**, А.В. Благов**[1]

[1]Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

Аннотация. Существует множество подходов к построению экспертных систем в зависимости от решаемой задачи и области применения. Данная статья включает в себя обзор традиционного подхода к построению самообучающихся экспертных систем с нечеткой логикой, описание альтернативного подхода, предлагаемого авторами, а также описание генерации синтетических тестов и итоговую таблицу сравнения двух подходов.

1. Введение

В данной статье предлагается альтернативный подход к построению экспертной системы с нечеткой логикой для игры, похожей на 20q (или Акинатор), а также приводится сравнение с классическим подходом, основанном на теореме Байеса.

Практическая ценность подобных экспертных систем заключается в помощи пользователю с поиском «того, не знаю чего» (например, когда мы ищем подарок для близкого человека, мы располагаем только знаниями об этом человеке и ничем более). Такие игры, как 20q и Акинатор выступают в качестве тестирующей системы, позволяя разработчикам тестировать модель не на синтетических данных, а на реальных пользователях. Однако, поскольку авторы статьи не располагают доступом к широкой аудитории, сравнение будет происходить на основании данных, полученных при помощи синтетических тестов.

Суть игры «Акинатор» заключается в следующем:

1. Пользователь загадывает любого персонажа и начинает игру.
2. Система задает пользователю вопросы о персонаже из имеющейся базы вопросов.
3. Пользователь выбирает один из вариантов ответа: «да», «нет» или «не знаю».
4. После нескольких итераций система делает предположение о том, какого же именно персонажа загадал пользователь. Если предположение оказалось верным - алгоритм завершается, иначе - возвращается к шагу 1 с накопленными знаниями.

Можно провести аналогию с помощью в поиске подарков, когда вместо загаданного персонажа мы имеем некий абстрактный подарок, который должен быть релевантным, а вопросы задаются относительно виновника торжества. Также, мы делаем предположение, что если пользователь приобрел товар, то он является релевантным.

Систему можно разбить на несколько логических составляющих:

- выбор наиболее вероятного варианта ответа;
- выбор следующего вопроса;
- выбор момента, когда система готова предложить вариант ответа.

2. Вероятностный подход, основанный на теореме Байеса

Традиционно, для построения экспертных систем с нечеткой логикой, применяют вероятностный подход, основанный на теореме Байеса [1]. К его недостаткам можно отнести высокую вычислительную сложность при выборе вопроса, а также незначительное увеличение эффективности при использовании хитрых модификаций [2].

2.1. Выбор наиболее вероятного варианта ответа

Предположим, что пользователь уже дал ответы некоторое количество вопросов и теперь система должна выдвинуть предположение, какой именно объект C подойдет пользователю. Для этого предлагается воспользоваться теоремой Байеса:

$$P(c \mid \langle Q_1, A_1\rangle, \dots, \langle Q_n, A_n\rangle) = \frac{P(\langle Q_1, A_1\rangle, \dots, \langle Q_n, A_n\rangle \mid c)P(c)}{\sum_{c'} P(\langle Q_1, A_1\rangle, \dots, \langle Q_n, A_n\rangle \mid c')P(c')} \quad (1)$$

где c – элемент из множества объектов, которые могут быть предложены пользователям в качестве искомого ответа; P(c) – априорная вероятность того, что пользователю подойдет объект c (может быть вычислена как доля сессий, в которых был выбран этот объект, среди всех сессий); $\langle Q_i, A_i\rangle$ – пара, состоящая из вопроса, который был задан пользователю, и соответствующего ему ответа.

Для упрощения вычислений авторы предполагают условную независимость ответов на вопросы при заданном объекте:

$$P(\langle Q_1, A_1\rangle, \dots, \langle Q_n, A_n\rangle \mid c) = \prod_{i=1}^{n} P(\langle Q_i, A_i\rangle \mid c) \quad (2)$$

Тогда получим следующую формулу для вычисления вероятности выбора объекта C при известных ответах на заданные вопросы:

$$P(c \mid \langle Q_1, A_1\rangle, \dots, \langle Q_n, A_n\rangle) = \frac{P(c)\prod_{i=1}^{n} P(\langle Q_i, A_i\rangle \mid c)}{\sum_{c'} P(c')\prod_{i=1}^{n} P(\langle Q_i, A_i\rangle \mid c')} \quad (3)$$

2.2. Выбор следующего вопроса

Основываясь на истории игр и ответах, которые пользователь уже дал, можно оценить вероятность каждого варианта ответа при фиксированном вопросе по следующей формуле:

$$P_A = \sum_C P(\langle Q, A\rangle \mid c)P(c \mid \langle Q_1, A_1\rangle, \dots, \langle Q_n, A_n\rangle) \quad (4)$$

где Q – фиксированный вопрос, который выбрала система, чтобы задать его пользователю; A – ответ на вопрос, вероятность которого мы собираемся оценить.

Идея способа выбора вопроса состоит в следующем: если бы мы играли в игру с вариантами ответа «да» и «нет», оптимальной стратегией был бы выбор того вопроса, который отсекает половину вариантов ответов. Автор предлагает обобщение этого метода, при котором система будет каждый раз выбирать тот вопрос, который минимизирует условную энтропию при известном ответе:

$$H(Q, A) = H[P(c \mid \langle Q_1, A_1\rangle, \dots, \langle Q_n, A_n\rangle, \langle Q, A\rangle)]$$
$$Q^* = arg\min_Q \sum_A H(Q, A)P_A \quad (5)$$

Таким образом можно выбирать следующий вопрос за время O(|C| + |Q| + |A|).

2.3. Выбор момента, когда система готова предложить вариант ответа

В качестве момента, при котором система должна выдвинуть свое предположение, авторы предлагают воспользоваться следующим условием:

$$\exists\, c \epsilon C:\ P(c \mid \langle Q_1, A_1\rangle, \dots, \langle Q_n, A_n\rangle) \geq 0.8 \quad (6)$$

3. Матричный подход

В предложенном авторами данной статьи подходе мы имеем дело с матрицей размера |Q| x |C|:

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1m} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nm} \end{pmatrix} \quad (7)$$

где p_{ij} – доля сессий, в которых на i-й вопрос для j-го объекта был дан ответ «да», среди всех сессий, нормированная к отрезку [-1; 1].

При разработке данного подхода были сделаны следующие предположения:

- $p_{ij} \xrightarrow{K\to\infty} 0$, если i-й вопрос никак не влияет на выбор j-го объекта;
- $p_{ij} \xrightarrow{K\to\infty} 1$, если j-й объект обладает свойством, которое характеризуется положительным ответом на i-й вопрос;
- $p_{ij} \xrightarrow{K\to\infty} -1$, если j-й объект обладает свойством, которое характеризуется отрицательным ответом на i-й вопрос;
- ответы пользователя представляются в виде вектора-строки $A = (a_1, a_2, \dots, a_n)$, где $a_i \in \{-1, 0, 1\}$ и характеризует ответ пользователя на i-й вопрос;
- ответ «не знаю» не используется в дальнейших вычислениях и приравнивается к ситуации, при которой вопрос вообще не задавался.

К преимуществам данного подхода можно отнести его простоту, а также возможность добавлять модификации в любой шаг алгоритма.

3.1. Выбор наиболее вероятного варианта ответа

Поскольку каждый объект характеризуется своим вектором-столбцом $P_j = (p_{1j}, p_{2j}, \dots, p_{nj})^T$ из матрицы P, мы можем, используя косинусную меру, определить правило выбора наиболее вероятного варианта ответа следующим образом:

$$C^* = arg\max_j \langle A, P_j \rangle \tag{8}$$

Таким образом, мы выбираем объект, для которого средний вектор ответов по всему набору сессий наиболее близок к вектору ответов пользователя.

3.2. Выбор следующего вопроса

При выборе следующего вопроса мы, по аналогии с вероятностным подходом, основанном на теореме Байеса, будем стараться выбирать тот вопрос, который отсекает половину вариантов ответов. При этом те вопросы, которые не влияют на выбор большинства объектов, должны иметь наименьший приоритет. Исходя из описанных требований авторами была предложена следующая метрика для выбора вопроса:

$$Q^* = arg\max_i \sum_{j=1}^{m} \sum_{\substack{k=j+1 \\ a_{ij}\cdot a_{ik}<0}}^{m} (|a_{ij}| + |a_{ik}|) \tag{9}$$

Очевидно, что при таком подходе максимум достигается в том случае, когда количество положительных элементов в строке примерно соответствует количеству отрицательных элементов и абсолютные значения всех элементов строки близки к 1. Вычислительная сложность данного метода равна O(|Q|·|C|), что значительно больше, чем у вероятностного подхода.

Эффективность данного метода можно улучшить, если выбирать следующий вопрос по некоторому подмножеству объектов (например, по наиболее вероятным).

3.3. Выбор момента, когда система готова предложить вариант ответа

В наивной реализации матричного подхода система выдвигает предположение о наиболее вероятном объекте спустя 5 заданных вопросов. Данное число было выбрано эксперементальным путем.

Использование различных метрик, для которых возможно подобрать оптимальное пороговое значение, в данной работе не приводится.

4. Описание тестирующей системы

Для тестирования подходов, описанных в данной работе, был разработан генератор синтетических тестов и система для тестирования моделей. В основе тестирующей системы лежит идея, предложенная научным сотрудником Университета Дьюка [3].

Алгоритм тестирующей системы состоит в следующем:

1. В пространстве $\{(x, y) : x \in [0; 1] \wedge y \in [0; 1]\}$ случайным образом генерируется m точек (объектов) и n прямых линий (вопросов). Если j-я точка лежит правее i-й прямой, то ответом на i-й вопрос для j-го объекта является «да», иначе – «нет».
2. Генерируется вектор-столбец $mistakes$ размера n с нормальным распределением. Для i-го вопроса соответствующий элемент вектора $mistakes$ задает вероятность, с которой система должна дать ложный ответ.
3. На i-й вопрос с вероятностью $nkp = 0.2$ система отвечает «не знаю», а с вероятностью $mistakes_i$ дает ложный ответ.
4. Если модель вернула предполагаемый ответ, то вычисляем евклидово расстояние от предлагаемого объекта до загаданного и, если оно меньше порогового, считаем, что модель дала верный ответ, иначе – неверный.

Такой подход позволяет эмулировать ошибки пользователей, незнание ответа на заданные вопросы, а также специфические вопросы, на который дать корректный ответ достаточно трудно. Кроме того, объекты изначально сгенерированы таким образом, что их можно разбить на кластеры, чтобы одному вектору ответов соответствовало несколько релевантных объектов (а не только тот, который изначально загадала система).

5. Итоговая таблица сравнения подходов

Данные, приведенные в таблице 1, были вычислены на основании 200 последних сессий. Из нее следует, что на предложенных синтетических тестах матричная модель делает в 2.5 раза меньше попыток и задает в 3 раза меньше вопросов, чем традиционная вероятностная модель, основанная на теореме Байеса.

Таблица 1. Итоговая таблица сравнения подходов.

	Вероятностная модель		Матричная модель	
	Кол-во попыток	Кол-во заданных вопросов	Кол-во попыток	Кол-во заданных вопросов
$runs = 1000$ $items = 100$ $questions = 20$	6.58 ± 4.98	46.35 ± 35.59	2.42 ± 1.39	12.07 ± 6.73
$runs = 1000$ $items = 200$ $questions = 50$	5.30 ± 5.34	36.77 ± 37.66	2.67 ± 1.87	13.38 ± 9.34
$runs = 1000$ $items = 500$ $questions = 100$	6.56 ± 3.67	46.48 ± 27.50	2.58 ± 1.63	12.93 ± 8.14

6. Заключение

Самообучающиеся экспертные системы с нечеткой логикой вполне могут стать достойной альтернативой продавцам-консультантам как в онлайн, так и в оффлайн магазинах. Обслуживая десятки тысяч клиентов они накапливают знания, чтобы делать наиболее релевантные предложения конкретному покупателю. Человек, к сожалению, лишен такой возможности масштабирования.

Дальнейшее исследование может быть продолжено в части реализации эвристик, направленных на улучшение эффективности матричного подхода. Например, можно выбирать следующий вопрос только по подмножеству наиболее вероятных объектов или реализовать механизм прогнозирования ответов пользователя на еще не заданные вопросы.

7. Литература

[1] Yangel, B. Bayesian approach and Akinator [Electronic resource]. – Access mode: http://www.machinelearning.ru/wiki/images/7/78/BayesML-2010-Yangel-Akinator.pdf (2.11.2018).

[2] Jedynak, B. Twenty questions with noise: Bayes optimal policies for entropy loss Network / B. Jedynak, P. Frazier, R. Sznitman // Journal of Applied Probability. – 2012. – Vol. 49. – P. 114-136.

[3] Suresh, S.R. A Bayesian Strategy to the 20 Question Game with Applications to Recommender Systems [Electronic resource]. – Access mode: https://dukespace.lib.duke.edu/dspace/bitstream/handle/10161/16414/Suresh_duke_0066N_14270.pdf (2.11.2018).

Благодарности

Работа была поддержана Министерством образования и науки Российской Федерации в рамках реализации Программы повышения конкурентоспособности СГАУ среди ведущих мировых научных и образовательных центров на 2013-2020 годы.

The research of approaches to create self-learning expert systems with fuzzy logic

V. Bakaev[1], A. Blagov[1]

[1]Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

Abstract. The article is devoted to the study of existing methods for creating self-learning expert systems. The authors analyze existing approaches and methods and offer their own approach, which has several advantages.