

03, март 2016

УДК 004.55

Нечеткий логический вывод для распознавания поведения пользователя в докликовом процессе человеко-машинного взаимодействия

*Зиняков Г. Н., студент,
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационные системы и телекоммуникации»*

*Научный руководитель: Алфимцев А. Н., к.т.н., доцент,
Лычков И.И., старший преподаватель,
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационные системы и телекоммуникации»
alfim@bmstu.ru*

Введение

Как известно, в крупном онлайн-бизнесе, ориентированным на работу с клиентами через веб-интерфейс, часто встают вопросы анализа эффективности работы интерфейса. В случае с большими системами, состоящими из множества элементов, представляется затруднительным выполнять анализ вручную. Также невозможно выполнить ручной анализ в случае, когда имеется бесконечно большое количество вариантов представления данных в интерфейсе (пример - поисковая выдача в крупном интернет-магазине). В таких случаях необходимо автоматизировать процесс, вычислив необходимые параметры пользовательских сценариев. Доказано, что пользовательские сценарии, процесс чтения, изучения сайта и поведение во многом связаны с перемещением курсора, потому положение областей активности курсора пользователя сопоставимо с направлением взгляда[1]. Исходя из этого, мы можем автоматизировать процесс анализа областей, изучаемых пользователем на сайте и представляющих ему интерес, чтобы отслеживать пользовательские сценарии. В рамках данной статьи мы будем использовать алгоритмы нечеткой логики для данных целей, так как они позволяют дать нечеткую и градуированную оценку, особо ценную при анализе большого количества данных об активности пользователей крупного портала.

Системы, использующие алгоритмы нечеткого логического вывода повсеместно существуют в нашей жизни. Например, для создания графического пользовательского

интерфейса для моделирования лазерных устройств [2]. Такая технология позволяет преодолевать многие проблемы, встречающиеся в массовом производстве, где с каждым новым витком технологии увеличивается число используемых параметров и усложняется возможность полноценной экспертизы человеческими силами. Также, технологии нечеткого логического вывода применяются в системах управления зарядом мобильных устройств [3], в системах машинного обучения для построения экспертной системы[4], оценки риска заболевания раком у человека[5].

Описание реализации метода с указанием использованных библиотек

В качестве средства для создания веб-интерфейса был использован тестовый домен johnpeel.ru, который привязан к частному VPS-серверу, предоставленном компанией FastVPS. Подобный способ обеспечивает доступность данного эксперимента с любого компьютера, имеющего доступ к Интернету. Методом и площадкой для создания интерфейса был использован Wordpress. Wordpress - наиболее популярная и развитая система управления сайтами, работающая на основе PHP и баз данных MySQL. Структура Wordpress состоит из свободно подключаемых модулей, инструментов, плагинов и тем, а также из ядра самой системы. При взаимодействии с сайтом, ядро Wordpress обращается к базе данных и вызывает весь пользовательский контент, который начинает собираться при помощи файлов темы, плагинов и пользовательского содержания, генерируя динамические веб-страницы. Wordpress завоевал свою популярность благодаря легкости подключения модулей, их обилия, а также простоты использования самых современных технологий, таких как мобильная оптимизация, использования препроцессоров CSS и Javascript, интеграции сторонних аналитических сервисов и социальных сетей. Также, Wordpress имеет удобную панель администрирования, которая упрощает разработчикам взаимодействие с сайтом. Wordpress используется в широком спектре сайтов - блогах, интернет-магазинах, корпоративных сайтах и рекламных страницах. Система обладает открытым исходным кодом и активно поддерживается и разрабатывается свободным сообществом WordPress Foundation.

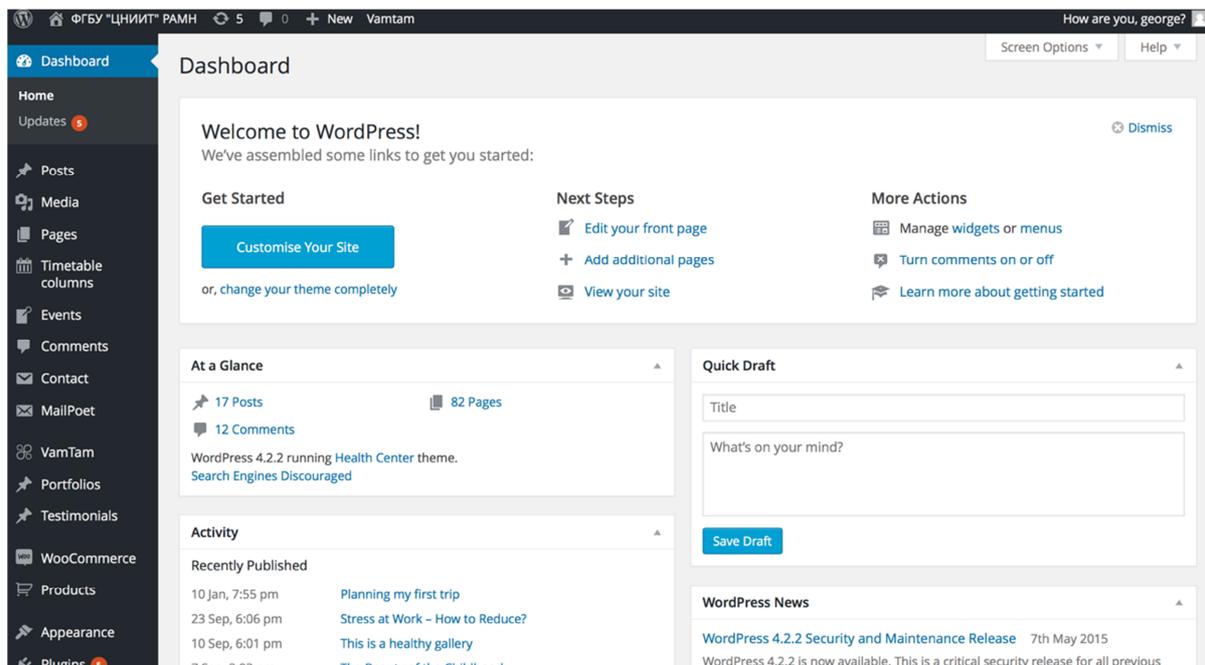


Рис. 1. Общий вид панели управления Wordpress

Инструментом для сбора данных о поведении курсора пользователя выступает плагин Heat Map Tracker, который интегрируется со средой Wordpress и записывает перемещение курсора в БД MySQL. Плагин также позволяет генерировать тепловые карты для движений мыши и кликов, что позволяет наглядно убедиться о наличии пользовательской активности.

Вторым инструментом для реализации алгоритма нечеткой логики использован плагин Shortcode Exec PHP, который, будучи интегрированным в Wordpress, дает возможность оптимально подключаться ко всем ресурсам CMS и писать скрипты на PHP, используя уже существующее подключение к базе данных и глобальным переменным Wordpress (что существенно безопасней нового подключения «с нуля»), а также выводить результат выполнения скриптов в пользовательский интерфейс, что актуально для данного эксперимента.

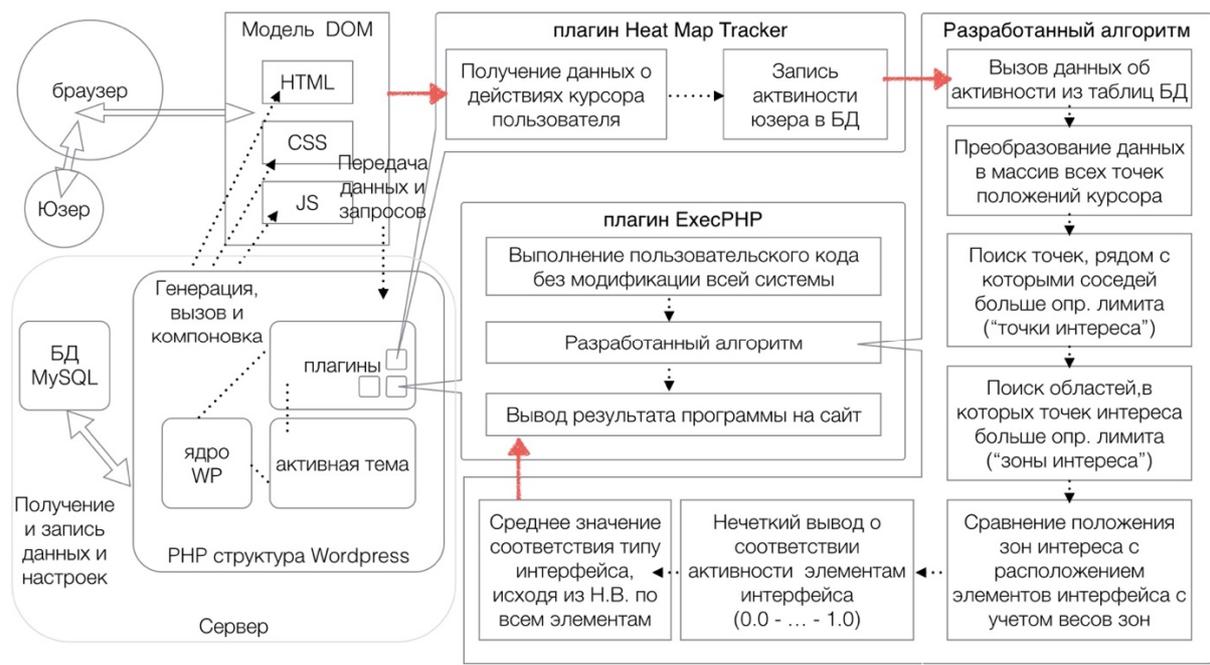


Рис. 3. Структурная схема рабочей системы и разработанного алгоритма внутри нее. Красные стрелки обозначают события в работе всей системы, которые связывают разные ее компоненты между собой. Блоки с носовым выступом показывают вхождение данного блока в другой. Точки интереса – это центры областей интереса, где есть активность пользователей. Лимит – определяющее значение количества точек, после которого срабатывает алгоритм поиска точек и областей интереса (см. описание). Н.В. – нечеткий вывод

Реализованный PHP-скрипт, подключаясь к таблице в БД, в которой собраны все передвижения курсора пользователя, делает MySQL-выборку по строкам, соответствующим заданной странице на сайте (ее URL), дате начала и окончания выборки. Выбранные значения имеют координаты точки положения курсора по X, Y, а также текущее разрешение экрана пользователя. В нашем эксперименте мы берем полученные данные и создаем массив всех точек, содержащий исключительно их координаты. После этого мы проходим в цикле по всем точкам из этого массива и смотрим, какое количество «соседей» имеет каждая - сколько других точек попадет в радиус 40 пикселей от координаты исследуемой точки (40 пикселей выбраны из соображения минимального размера элемента интерфейса в рамках созданного интерфейса). Количество «соседей» записывается отдельным параметром для каждой точки в массиве.

В коде программы, для реализации этого алгоритма мы обращаемся к таблице базы данных, где хранятся данные о передвижении курсора – «hmaptracker_mmove». SQL-запрос задает URL исследуемой страницы, а также временной период выборки. Результат запроса парсится в цикле foreach, записывая извлеченные значения в массив точек. После этого, мы выполняем двойной цикл, проходясь по каждой точке и вычисляя, сколько рядом с ней соседей. Данная операция осуществлена внутренним циклом, который для текущей точки проводит сравнение с каждой другой точкой на вопрос ее близости в 40-пиксельном радиусе. Если условие выполняется, исследуемая точка увеличивает свое число соседей, то есть создается массив соседей, в котором порядковый номер – номер нашей точки, а значение – количество соседей у этой точки.

Чтобы начать анализ активности среди полученных точек, мы должны определить пороговый, высокий уровень интересующей нас пользовательской активности. Ведь, как известно, активность пользователя курсором на экране может быть как случайной, так и связанной с заинтересованностью - случайные движения мышью, как правило, хаотично разбросаны по экрану и слабо насыщены по количеству точек на область. Отсюда, мы должны отсечь все области, не достигшие высокого уровня активности пользователя. Понятие «высокого уровня активности»мы берем исходя из отношения общего количества проанализированных точек к площади страницы в пикселях, умножив все на k - коэффициент усиления. Данный коэффициент в дальнейшем может быть привязан к количеству пользователей, посетивших сайт, особенностям бизнес-модели и требуемым пользовательским сценариям, но в рамках нашего эксперимента мы задали коэффициент равным 8 на основании практических наблюдений и n-го количества точек, сгенерированных в рамках эксперимента. Также, зная радиус каждой из исследуемых областей (40 пикселей), мы можем сразу узнать количества точек, необходимых для того, чтобы мы могли считать данную область «зоной интереса». Это пороговое значение представляет собой отношение количества точек к площади S экрана (в пиксельном выражении), умноженное на произведение коэффициента усиления и площади одной минимальной зоны интереса.

$$I = \frac{n_{\text{соседей}}}{S} * (k * \pi * r^2),$$

где n – количество точек, S – площадь экрана, k – коэфф. усиления, r – радиус одной области.

Иными словами, если точка имеет количество «соседей», большее или равное пороговому значению, то она означает область высокой насыщенности пользовательской активности. Подобные точки мы записываем в отдельный массив «интересных точек», в котором каждая точка, помимо координат, имеет значение своего веса P , то есть отношения количества соседей к пороговому значению интереса.

$$P = \frac{n_{\text{соседей}}}{I},$$

где n – число соседей, I – пороговое значение.

Для реализации алгоритмов нечеткой логики, значение веса каждой точки интереса колеблется от 0 до 1, выражая функцию принадлежности. В случае значения веса, большего единицы, оно приравнивается к единице. Также, чтобы не допустить дублирования областей интереса, мы проходимся циклом по всем полученным областям и удаляем все те, которые расположены друг от друга на расстоянии менее заданного минимума размера элемента.

Следующим шагом является выделение элементов интерфейса. Элементы, интересующие нас для анализа пользовательских сценариев, задаются экспертом, исходя из представлений о бизнес-процессах и потребностях в его оптимизации путем данного эксперимента. Элемент также представляет собой точку на странице сайта, соответствующую координатами фактическому центру элемента. Мы создаем массив элементов интерфейса, имеющий их координаты, а также радиус области изучения элемента, также заданный экспертом - область возле элемента, в рамках которой необходимо исследовать активность пользователя.

В первую очередь, мы определяем количество точек (областей) интереса, вошедших в данный радиус. При определении радиуса необходимо учитывать, что некоторые области могут оказаться на его границы и потому могут оказаться случайными, не имеющими отношения к исследуемому элементу, потому радиус задается с 40-пиксельным вычетом.

Чтобы вычислить вес целого элемента E , мы вычисляем среднее арифметическое весов всех точек, находящихся внутри него, определенных на предыдущем этапе. Помимо этого, мы определяем среднее арифметическое отклонения всех точек интереса, входящих в элемент. Это значение дает нам представление о том, насколько положение центра пользовательской активности отличается от фактического центра элемента интерфейса.

для $\forall E_i$ при $r_i = X_{\text{экспертное}} - 40$

и $\forall r_{n_{\text{интереса}}} \in r_i$

$$P_i = \sum_0^{n_{\text{интереса}}} P_{n_{\text{интереса}}}$$

$$D_{xi} = \sum_0^{n_{\text{интереса}}} D_{x_{n_{\text{интереса}}}}$$

$$D_{yi} = \sum_0^{n_{\text{интереса}}} D_{y_{n_{\text{интереса}}}}$$

$$D_i = \sqrt{D_x^2 + D_y^2},$$

где E – i -й элемент, r – радиус i -й области интереса, P – вес i -го элемента интерфейса, D – среднее арифметическое отклонения i -й точки интереса.

Согласно теории, нечеткий логический вывод Мамдани основан на степенях принадлежности, например, для входного вектора $X = (x_1, x_2, \dots, x_n)$ нечетким термам d_j из базы знаний рассчитывается следующим образом:

$$\mu_{d_j}(X) = \bigvee_{p=\overline{1,k_j}} w_{jp} * \bigwedge_{i=\overline{1,n}} [\mu_{jp}(x_i)], j = \overline{1,m},$$

где $\bigvee (\bigwedge)$ – операция из s -нормы (t -нормы), т.е. из множества реализаций логической операций ИЛИ (И). Наиболее часто используются следующие реализации: для операции ИЛИ – нахождение максимума и для операции И – нахождение минимума.[8]

Ключевым моментом в реализации механизма нечеткой логики является анализ того, насколько близка была активности пользователя к элементу того или иного типа [9]. Для этого, мы выбираем повторяющиеся элементы одного типа (например, кнопка «купить товар»), и, учитывая вес и среднее отклонение активности для каждого элемента, строим нечеткий логический вывод, с учетом веса каждого элемента и его отклонения. Иначе говоря, значение отклонения для более весомого элемента играет большую роль в полученном значении среднего общего отклонения. Полученное значение M среднего общего отклонения колеблется от 0 до 1, где 1 – полное совпадение центра активности с фактическим центром, а 0 – наименьшее.

$$P_{\text{среднее}} = \sum_0^i P_i$$

$$M = \frac{\sum_0^i P_i * D_i}{i * P_{\text{среднее}}}$$

где M – среднее общее отклонение

Такая технология позволяет выяснить, насколько желаемые пользовательские сценарии совпали с фактическими. Например, если предполагается, что пользователь будет одинаково заинтересован в покупке всех товаров, а по факту его заинтересовали только первые три, то гипотеза считается неверной и ситуация переходит в сферу анализа проблем usability и бизнес-процессов.

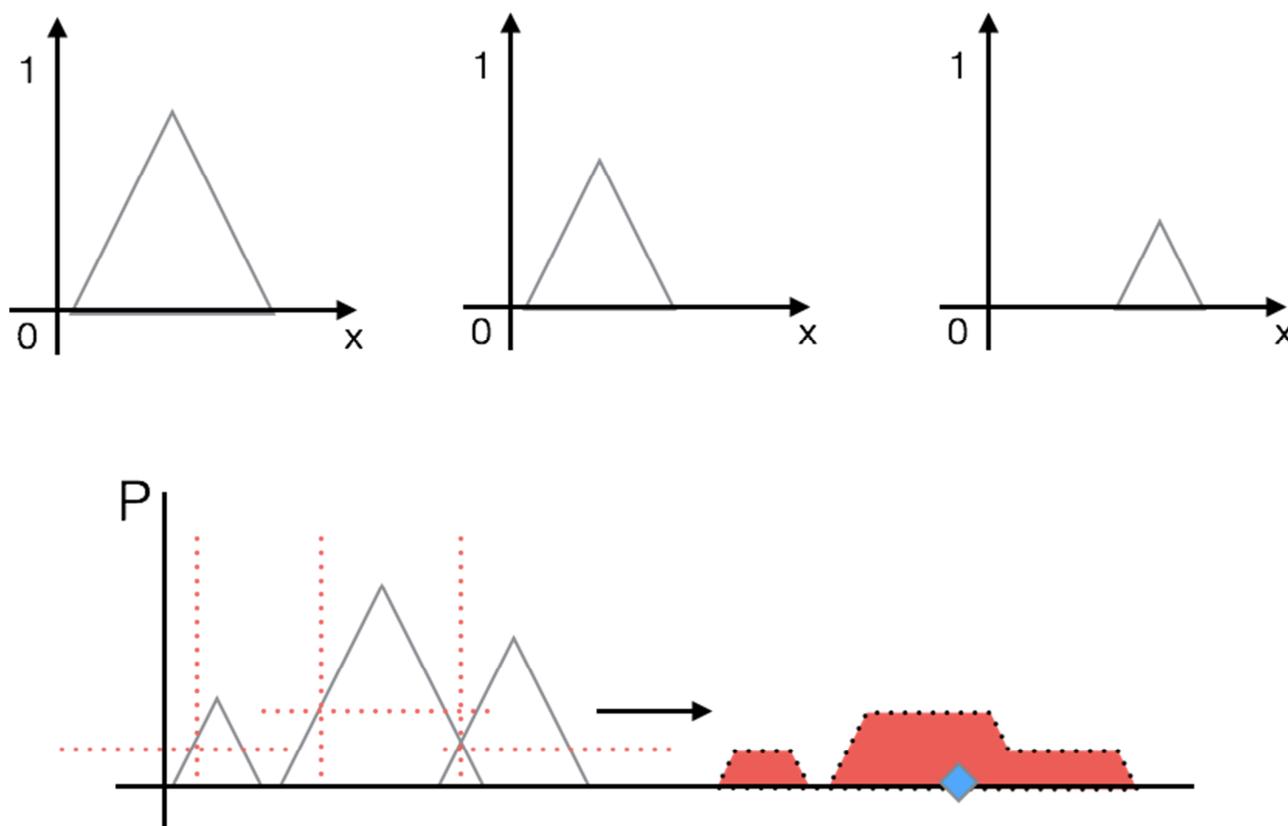


Рис. 3. Вычисление нечеткого логического вывода для повторяющихся элементов, обладающих разным весом и значением смещения. По оси P представлен вес, а по горизонтальной оси – расположение элемента. Положения областей интереса соответствуют определенному отдалению от заданного центра элемента интерфейса и отображается пунктиром, следовательно, центр массы полученной фигуры есть среднее значение отклонения активности пользователя от положения элементов для всего интерфейса

Обзор альтернативных подходов к решению задачи:

Альтернативные методы решения задачи можно разделить на два существенных отличия: архитектурное и делегированное. Решения, имеющие другое архитектурное решение, в первую очередь, могут иметь меньшую степень интеграции с Wordpress и сторонними плагинами. Это позволит сделать работу программы более независимой от сторонних разработчиков, ответственных за ядро системы и функционал, безопасность плагинов. Однако, в таком случае, нам придется вручную создавать весь алгоритм записи положений курсора в базу данных, с учетом данных о текущей дате и просматриваемой странице. Немного помочь в этом сможет бесплатная библиотека `heatmap.js`, которая позволяет генерировать тепловые карты по входным данным при помощи `javascript`. Дальнейшие способы реализации в плане архитектуры могут быть любыми и зависимыми исключительно от предпочтений разработчика и поставленных целей. Например, если разработчик считает язык PHP ненадежным, та же система может быть реализована на любом другом серверном языке программирования, таком как Java, .NET или Ruby On Rails. В отличие от прочих, наша реализация достаточно проста в установке и не требует серьезных навыков для освоения.

В области делегирования можно рассмотреть вариант переноса задачи на сторонних разработчиков, которые смогут своими собственными средствами анализировать сайт и предоставлять все необходимые выводы. Подобные услуги предоставляют такие компании, как Crazy Egg, VWO и им подобные. Преимуществом в данном случае есть то, что ответственность за верность анализа целиком лежит на другой компании, которая также может предоставить дополнительные услуги, вроде А/Б тестирования. Однако, такой способ не является бесплатным.

Описание проведенного эксперимента

Мы провели эксперимент на трех разных типах интерфейса - магазин, блог, и смешанный тип с баннерами записей. В ходе эксперимента проверялось отклонение фактической активности пользователя от действительного положения элементов. В случае интерфейса магазина анализировались кнопки «купить товар», в случае с блогом ссылка «оставить комментарий», в смешанном типе проведен анализ как кнопок «купить товар», так и других элементов, вроде ссылки на корзину пользователя, описание товара.

Рассмотрим каждые три случая и результаты работы программы:

1. Интерфейс «Магазин» - много однотипных кнопок «купить товар»

Для начала сделаем гипотезу, которая будет проверяться. Предположим, что пользователю будут интересны кнопки покупки товара больше всего остального на сайте. Судя по тепловой карте, сгенерированной плагином, мы видим, что пользователь был заинтересован довольно сильно в нескольких конкретных кнопках покупки, так как вокруг них заметны большие очаги активности (рисунок 4)

Чтобы узнать исходя из данного наблюдения, насколько проверилась наша гипотеза, запустим программу.

На выходе программа дает информацию о том, что очаги пользовательской активности для кнопки купить товар близки к фактическому расположению элемента со степенью 0,597 и количество элементов подобного типа - 12

Мы можем сделать вывод, что наша гипотеза проверилась, так как степень близости достаточно велика и ее нельзя игнорировать или считать случайными действиями, не имеющими отношения к реальной заинтересованности пользователя

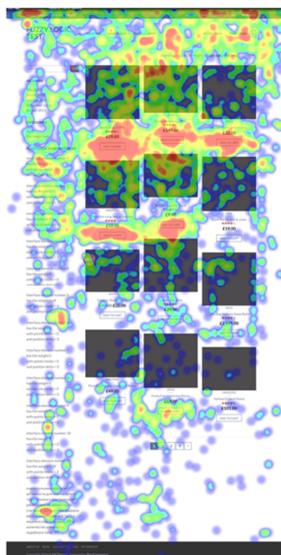


Рис. 4. Интерфейс типа «Магазин»

2. Интерфейс «Блог» - повторяющиеся элементы «оставить комментарий»

Точно также установим гипотезу, что пользователю будут интересны наши элементы «оставить комментарий», чтобы узнать насколько часто он задумывается оставить комментарий к блогу.

Как мы можем видеть по тепловой карте (рисунок 5), активность пользователя равномерно распределена по странице и особо не затрагивает кнопки, проверим это программой.

На выходе программа дает информацию о том, что очаги пользовательской активности для кнопки купить товар близки к фактическому расположению элемента со степенью 0,089 и количество элементов подобного типа – 7.

Как мы можем видеть, степень соответствия крайне мала, из чего мы можем вывести, что наша гипотеза не верна.



Рис. 5. Интерфейс типа «Блог»

3. Галерея, большое количество картинок

Теперь проверим то, насколько пользователю будут интересны графические баннеры, картинки в интерфейсе типа «Галерея». В качестве гипотезы предположим, что пользователь будет проводить много времени внутри картинок, то есть на всех элементах, расположенных на их территории. Посмотрим на собранную тепловую карту (рисунок 6)

На выходе программа дает информацию о том, что очаги пользовательской активности для кнопки купить товар близки к фактическому расположению элемента со степенью 0,771 и количество элементов подобного типа – 7.

Значение близости сравнительно велико, но в то же время находится посередине, из чего мы не можем сделать однозначного вывода об активности пользователя, но можем констатировать, что область изображений так или иначе сыграла немаловажную роль в данном пользовательском сценарии.

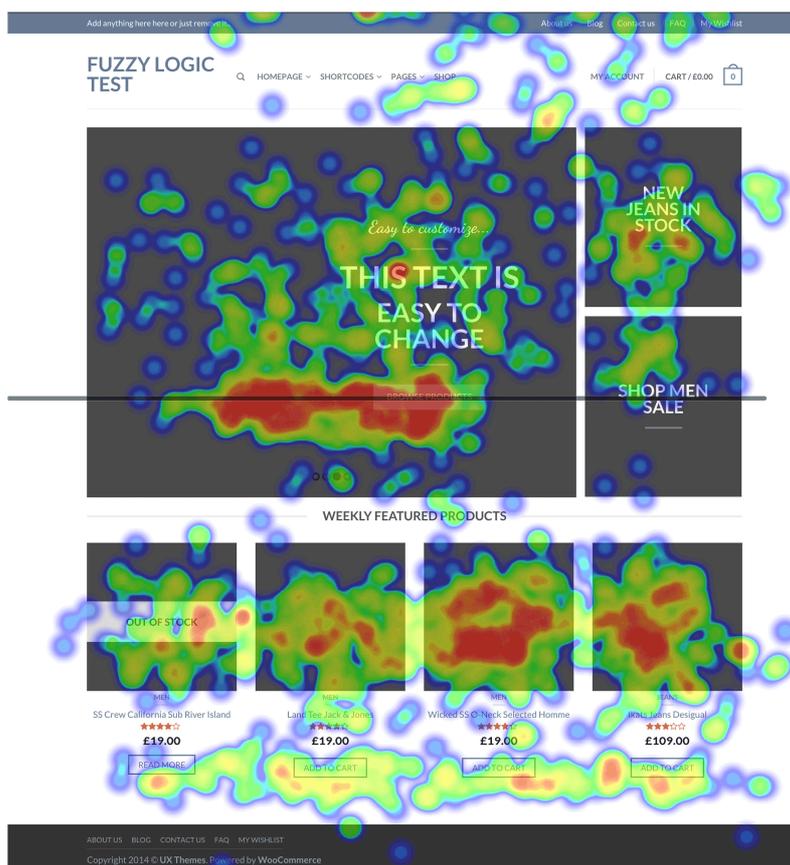


Рис. 6. Интерфейс типа «Галерея»

Составим сводные таблицы (табл. 1, табл. 2, табл. 3) для этих и других результатов экспериментов, чтобы пронаблюдать динамику значений.

Таблица 1

Тип интерфейса «Магазин»

№ Эксперимента	Процент распознавания сценария	Кол-во иссл. элементов
1 эксперимент	59,7%	12
2 эксперимент	79,3%	16
3 эксперимент	60,1%	10
4 эксперимент	52,1%	18
Среднее значение	62,8%	14

Таблица 2

Тип интерфейса «Блог»

№ Эксперимента	Процент распознавания сценария	Кол-во иссл. элементов
1 эксперимент	89,4%	15
2 эксперимент	61,1%	10
3 эксперимент	55,3%	12
4 эксперимент	75,8%	14
Среднее значение	70,4%	13

Таблица 3

Тип интерфейса «Галерея»

№ Эксперимента	Процент распознавания сценария	Кол-во иссл. элементов
1 эксперимент	77,1%	17
2 эксперимент	73,9%	10
3 эксперимент	67,3%	12
4 эксперимент	51,3%	14

№ Эксперимента	Процент распознавания сценария	Кол-во иссл. элементов
5 эксперимент	50,3%	16
Среднее значение	59,9%	14

Исходя из таблиц, можем сделать вывод, что, хотя значения степени близости растут вместе с ростом числа элементов для этого типа интерфейса, они падают для другого и при этом дают противоречивую картину в смешанном типе. Это свидетельствует о том, что степень близости по большей части определяется типом и предназначением интерфейса

Согласно определению коэффициента Стьюдента, Пусть Y_0, Y_0, \dots, Y_n

- независимые стандартные нормальные случайные величины, такие что $Y_i \sim N(0,1), i = 0$. Тогда распределение случайной величины t , где $t = \frac{Y_0}{\sqrt{\frac{1}{n} \sum_{i=1}^n Y_i^2}}$,

Вычислим распределение для нашего случая:

$$t = \frac{0,897}{\sqrt{\frac{2,049}{13}}} = 1,802.$$

Исходя из полученного распределения, вычислим коэффициенты распределения Стьюдента. По определению,

$F(n)$ — функция распределения Стьюдента $t(n)$ с n степенями свободы, и $\alpha \in [0,1]$.

Тогда квантилью этого распределения называется число $t_{\alpha,n}$ такое, что $F_{\alpha,n}(t_{\alpha,n}) = 1 - \alpha$

Подставляя наши значения, $F_{\alpha,n}(t_{\alpha,n}) = 1 - 1,802 = 0,802$.

Вывод

Исходя из проведенных экспериментов, мы получаем градуированное значение нечеткого логического вывода для каждой из гипотез. Подобная точность позволяет лаконично оценивать эффективность пользовательских интерфейсов в виде числового параметра, переводя условный язык «удобства» и «неудобства» в конкретные величины, обоснованные анализом по собранным данным. Такой подход в дальнейшем может быть использован для создания системы автоматического программного преобразования интерфейса и любой другой системы, где важно положение элементов и активность вокруг них, под требования, необходимые для улучшения качества взаимодействия и улучшения бизнес-процессов.

Список литературы

- [1]. Huang J., Xin Fu, White R.W., Dumais S. No Clicks, No Problem: Using Cursor Movements to Understand and Improve Search. Available at: <http://research.microsoft.com/en-us/um/people/sdumais/huangchi2011.pdf>, accessed 25.09.2015.
- [2]. Sivarao, Brevern P., El-Tayeb N.S.M., Vengatesh V.C. GUI Based Mamdani Fuzzy Inference System Modeling To Predict Surface Roughness in Laser Machining. Available at: <http://www.ijens.org/1957091%20ijecs.pdf>, accessed 25.09.2015.
- [3]. Moghimi M., Venkatesh J., Zappi P., Rosing T. Context-Aware Mobile Power Management Using Fuzzy Inference as a Service. Available at: http://seelab.ucsd.edu/papers/Moghimi_MobiCASE12.pdf, accessed 25.09.2015.
- [4]. Tzung-Pei Hong, Chai-Ying Lee. Induction of fuzzy rules and membership functions from training examples. Available at: <http://comp.eng.ankara.edu.tr/files/2013/03/BLM436Lecture8.pdf>, accessed 25.09.2015.
- [5]. Shleeg A.A., Ellabib I.M. Comparison of Mamdani and Sugeno Fuzzy Interference Systems for the Breast Cancer Risk. Available at: <http://waset.org/publications/17193/comparison-of-mamdani-and-sugeno-fuzzy-interference-systems-for-the-breast-cancer-risk>, accessed 25.09.2015.
- [6]. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику. Режим доступа: http://matlab.exponenta.ru/fuzzylogic/book1/1_7_5_1.php, (дата обращения 25.09.2015).
- [7]. Зиняков Г.Н. Особенности интерфейса посадочных веб-страниц и их преимущества для бизнеса // Молодежный научно-технический вестник. Электрон. журн. 2015. №3. Режим доступа: <http://sntbul.bmstu.ru/doc/771207.html>, (дата обращения 25.09.2015).
- [8]. Девятков В.В., Алфимцев А.Н. Нечёткая конечно-автоматная модель интеллектуального мультимодального интерфейса // Проблемы управления. 2011. № 2. С. 69-77.
- [9]. Алфимцев А.Н. Нечеткое агрегирование мультимодальной информации в интеллектуальном интерфейсе // Программные продукты и системы. 2011. № 3. С. 44-48.