

ми происходит классификация программного средства (для этого каждой характеристике присваивается степень ее присутствия для данного класса программ и производится их расчет). После этого появляется окно с набором характеристик для данного класса программ. Эксперт по испытаниям должен указать наличие или отсутствие данной характеристики в указанной программе. Характеристики должны быть представлены либо в бинарной (для качественных) либо в числовой (для количественных) форме. Далее происходит мультипликативная свертка нормированных значений характеристик с учетом коэффициента важности для получения количественной оценки степени превышения браковочного уровня. Браковочный уровень определяется из ГОСТов или экспертной комиссией. База снабжена подробным описанием каждой характеристики и диапазона ее допустимых значений, имеет возможность пополнения характеристик и механизм их удаления. Так, например, для сертификации «Сетевого компьютерного тренажера по распознаванию и ликвидации газонефтеводопроявления», разработанного на кафедре прикладной математики и компьютерного моделирования РГУ им. И.М. Губкина для обучения буровых инженеров-технологов, можно выделить следующие уникальные функциональные характеристики: возможность выбора модели месторождения для обучения; использование сетевого варианта тренажера; количество дросселей и задвижек; моделирование процесса в реальном временном интервале; наличие реакции на неправильную последовательность действий; возможность задания входных параметров «глушения» скважины (плотности заливаемой жидкости и давление вымывания); звуковая имитация процессов; масштабирование графиков зависимости давления от времени; визуализация действий на дроссели и ряд других. Составленная таким образом полная система характеристик качества утверждается в качестве нормативной базы для данного класса продуктов.

Список использованных источников

1. Липаев В.В. Надежность программных средств, М.: СИНТЕЗ, 1998
2. Костогрызов А.И., Липаев В.В., Сертификация качества функционирования автоматизированных информационных систем, М.: Вооружение. Политика. Конверсия, 1996

УДК 681.3.01

НЕКОТОРЫЕ АСПЕКТЫ ПРИМЕНЕНИЯ ТЕОРИИ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ К ЗАДАЧЕ СЕРТИФИКАЦИИ ПРОГРАММНЫХ СРЕДСТВ

Еснев К.С., Нешта Е.П., Колесников А.О., Ретинская И.В.

Один из способов обеспечения качества программных средств (ПС) является их сертификация. ПС сертифицируются в соответствии с их принадлежностью определенному классу программных средств. Сертифика-

ция это подтверждение соответствия установленным требованиям. В качестве таких требований, установленных на государственном уровне, выступают государственные и международные стандарты, а также отраслевые временные технические условия. Согласно ряду стандартов (например, ИСО/МЭК 9126-93) для проведения сертификации необходимо построение иерархической структуры характеристик качества (ИСХК). Стандартизация таких структур и автоматизация их построения требуют решения следующих задач:

- 1) построения единого рубрикатора программных средств, определяющего типы программ, относящихся к той или иной области сертификации;
- 2) построения для каждой области сертификации системы характеристик качества ПС, принадлежащих данной области;
- 3) построение математической модели и алгоритма классификации сертифицируемого программного продукта в соответствии с его свойствами согласно рубрикатору;
- 4) построение математической модели и алгоритма генерации ИСХК на экспертной информации.

Ниже рассматриваются аспекты решения третьей задачи на основе теории автоматической классификации /1,2/.

Математическая постановка задачи.

Пусть $P = \{p_1, \dots, p_N\}$ - множество известных типов программных продуктов, а N - число типов, S - множество областей сертификации. Обозначим мощность этого множества за M . Поставим в соответствие каждому из элементов множества P вектор принадлежности к областям сертификации размерностью M . Тогда $V_i = \{b_1, \dots, b_M\}$, $b_j = 1$, если i -й программный продукт принадлежит j -ой области, иначе $b_j = 0$. Введенные обозначения разбивают множество P на M классов возможно пересекающихся (иными словами, вводится отношение между элементами множеств P и S). Подмножества множества P соответствующие классам j , будем обозначать через P_j . Число типов программных продуктов, принадлежащих j -ой области X_j , определяется по формуле:

$$X_j = \sum_j b_j$$

Полученное разбиение может быть представлено в виде, изображенном табл. 1. В рассматриваемом случае $M=4$, $N=42$. Количество классов соответствует количеству областей аккредитации испытательной лаборатории, аккредитованной в системе "РОСИНФОСЕРТ". Для каждой области j введем в рассмотрение множество $C_j = \{c_1, \dots, c_{K_j}\}$ всех возможных характеристик всех типов программных продуктов (K_j - число характеристик), принадлежащих данной области. Сопоставим каждому из элементов множества C_j вектор-строку C_{jk} размерности X_j , элементы которой c_{jki} назовем *коэффициентами важности присутствия* k -ой характеристики у j -го

типа программного продукта. Образованную векторами-строками C_{jk} матрицу M_j размерности $K_j^* \times X_i$ назовем *матрицей характеристик* для области j . Равенство m_{jk} нулю означает, что характеристика не является обязательной для j -го типа программного продукта. Равенство m_{jk} единице, напротив означает, что присутствие k -ой характеристики обязательно для данного типа программ. Значения каждого из элементов матрицы вычисляется путем обработки данных экспертного опроса и определяет, насколько важно присутствие той или иной характеристики для данного типа программных продуктов. Обозначим через m_j^* минимальный уровень важности присутствия характеристики, при котором проверяется ее наличие у сертифицируемого программного продукта. При составлении системы характеристик качества характеристики, имеющие уровень важности меньше m_j^* , исключаются из рассмотрения. Обозначим множество неотсеявшихся характеристик через C_j^* . Характеристику C_{jk} будем называть *базовой* для данной области сертификации. Иными словами, базовой назовем характеристику, коэффициент важности присутствия которой для одного из типов программных продуктов больше аналогичных коэффициентов для всех остальных типов из данной области. Множество базовых характеристик обозначим за C_j^B . Мощность этого множества (количество базовых характеристик данной области) будем обозначать через K_j^B . Под *генеральной совокупностью* задачи классификации в данном случае будем понимать множество P_j^* , полученное путем добавления всех типов программных продуктов из области j P_j еще одного элемента - классифицируемого программного продукта. Аналогично, по множеством характеристик будем понимать множество C_j^B .

Модель решения.

Рассмотрим прямое (декартово) произведение множеств $P_j^* \times C_j^B$. Подмножество $R \subseteq P_j^* \times C_j^B$ определяет бинарное отношение для элементов множеств P_j^* и C_j^B . Для каждой пары данное отношение определяет, обладает ли этот тип базовой характеристикой или нет. Будем считать, что известные типы программ обладают всеми своими базовыми характеристиками и не обладают ни одной из базовых характеристик других типов. Поставим в соответствие подмножеству R множества $P_j^* \times C_j^B$ логическую матрицу L_R , позволяющую выделить пары множества $P_j^* \times C_j^B$, принадлежащие R .

Для измерения однородности введем число, характеризующее степень подобия элементов каждой пары - меру *таксономического подобия* или *таксономическое расстояние*. Под мерой таксономического подобия, или таксономическим расстоянием, будем понимать отображение произ-

ведения множеств $P_j^* \times P_j^*$ на некоторое подмножество вещественных чисел. Это означает, что каждой паре элементов ставится в соответствие вполне определенное вещественное число, которое считается мерой сходства элементов, образующих пару. Отображение в заданную сторону однозначно - каждой паре элементов соответствует лишь одно вещественное число. Обратное утверждение, вообще говоря, неверно: одно и то же таксономическое расстояние может быть между элементами нескольких пар.

Таблица 1 - Распределение ПС по областям сертификации

Прикладные программные средства для научных исследований	Прикладные программные средства для автоматизированных систем управления	Прикладные программные средства для обучения	Программно-информационные продукты
Энциклопедия	Интегрированная информационная система	Обучающие программы	Информационные базы нормативно-правовых документов
Справочники	Интегрированная система управления производственным объектом в реальном времени	Тренажеры	Электронные журналы
Электронные словари	Информационная система оперативного учета и управления	Компьютерные игры	Электронные реферативные издания
Электронные реферативные издания	Информационные системы руководителей	Программы контроля знаний (тестирующие)	Системы электронной документации
Многофункциональные пакеты математического моделирования	Информационные системы дни коллективной работы	Энциклопедии	Глобальные поисковые системы
Пакеты статистической обработки экспериментальных данных	Комплексная система управления предприятием	Справочника	Системы распространения информации о событиях в реальном времени
Языки программирования	Система управления информационными потоками	Электронные словари	Глобальные архивы материалов телеконференций
Коммуникационное ПО	Календарно-сетевое планирование и управления	Электронные учебники	Публичные каталоги библиотек
Программы систематизация и хранения экспериментальных данных	Комплексная система управления предприятием	Многофункциональные пакеты математического моделирования	Электронные библиотеки
Текстовые процессоры	Планирование бюджета предприятия	Пакеты статистической обработки данных	

Рассмотрим теперь на примере двух элементов из множества $P_j^* \times P_j^*$ возможные способы измерения таксономического подобия. Обозначим таксономическое подобие между его элементами p_1 и p_2 через $s(p_1, p_2)$.

Для вычисления $s(p_1, p_2)$ нам понадобятся две строки матрицы начальных данных. Их можно записать в виде таблицы 2×2 :

p_1	P_2	
	1	0
1	g_{11}	g_{10}
0	g_{01}	g_{00}

В этой таблице g_{11} означает число общих (принадлежащих одновременно элементам P_1 и P_2) свойств, g_{00} число свойств, отсутствующих у элементов P_1 и P_2 , а смысл g_{10} и g_{01} ясен непосредственно из таблицы. Сумму частот обозначим через $g = g_{01} + g_{10} + g_{01} + g_{00}$.

Для вычисления величины $s(p_1, p_2)$ можно воспользоваться информацией, содержащейся в представленной выше таблице.

Все показатели подобия обладают следующими свойствами:

1) величина $s(p_1, p_2)$ симметрична относительно p_1 и p_2 , т. е.

$$s(p_2, p_1) = s(p_1, p_2)$$

2) $s(p_1, p_2)$ - монотонно возрастающая функция от g_{01} .

Добавим также, что, обычно, показатель подобия принимает значения, лежащие в интервале $0 < s \ll 1$, и

$$s(p_1, p_1) = 1. \quad (1)$$

Наиболее часто применяемые на практике формулы для вычисления показателя подобия и их названия представлены в табл. 2.

Свойство симметрии позволяет сократить объем вычислений. Заметим также, что за исключением s_1 , для всех остальных показателей выполняется соотношение (1). Следовательно, вместо n^2 пар меру подобия необходимо вычислять лишь для $n(n-1)/2$ пар. Результаты в этом случае можно представить в виде треугольной матрицы. Каждая из формул для вычисления s_1 - s_5 позволяет найти численную меру близости пары элементов p_1 и p_2 .

Однако, следует заметить, что показатели s_2 , s_4 и s_5 плохо подходят для рассматриваемого случая, так как они симметричны по g_{01} и g_{10} . Очевидно, что в случае, когда речь идет о программных продуктах, одновременное отсутствие у них тех или иных свойств не может служить свидетельством сходства. В качестве примера можно привести следующую пару программ: электронный словарь и многофункциональный пакет математического моделирования. Обе программы не обладают возможностью создания автоматического контроля знаний, но проводить на этом основании какие-либо аналогии между ними не представляется возможным. Таким образом, для вычисления показателя подобия корректно будет вос-

пользоваться лишь формулами Рассела-Рао и Жаккарда, так как лишь они являются монотонными функциями от g_{11} .

Таблица 2 – Показатели подобия

Обозначение	Авторы	Формула
S_1	Рассел и Рао	$\frac{g_{11}}{g}$
S_2	Сокал и Миченер	$\frac{g_{11} + g_{00}}{g}$
S_3	Жаккард	$\frac{g_{11}}{g_{11} + g_{10} + g_{01}}$
S_4	Юл	$\frac{g_{11}g_{00} - g_{10}g_{01}}{g_{11}g_{00} + g_{10}g_{01}}$
S_5	Чупров	$\frac{g_{11}g_{00} - g_{10}g_{01}}{(g_{11} + g_{10})(g_{11} + g_{01})(g_{10} + g_{00})(g_{01} + g_{00})}$

Дальнейшее представляется очевидным. Построим для каждого из элемента P_i множества P_j^* вектор T_i размерности K_j^B по следующему правилу: для элементов множества, принадлежащих пересечению P_j^* и C_j^* (т.е. элементов множества P_j^* - известных типов программ) ненулевыми (равными 1) будут лишь компоненты вектора T_i , соответствующие базовым характеристикам, порожденным i -ым элементом. Для классифицируемого программного продукта соответствующий вектор T^* будем заполнять на основе сведений о наличии или отсутствии у него базовых характеристик, полученных от разработчика. Далее, сравнивая поочередно вектор T^* с каждым из векторов T_i , и, вычисляя для соответствующей пары элементов показатель подобия, выделим из множества P_j^* элемент r^{**} , наиболее схожий с классифицируемым продуктом. Будем считать, что рассматриваемая программа принадлежит типу, соответствующему r^{**} . В случае обнаружения нескольких элементов, близких к рассматриваемому, для которых показатели подобия будут совпадать, необходимо либо повторить процедуру классификации, скорректировав вектор T^* , либо принять решение об одновременной сертификации рассматриваемого продукта в нескольких областях сертификации.

Список использованных источников

1. Классификация и кластер. Под ред. Дрон Ван Найдена/ М.: Мир, 1980, 383 с.
2. Г. Парницкий. Основы статистической информатики. М.: Финансы и статистика. 1981, 194с.
3. Распознавание, классификация, прогноз. Ежегодник/Под ред. Ю.И. Журавлева. М.: Наука, 1989, 298 с.