

Основным модулем системы, который находится в стадии разработки, является модуль оценки кредитоспособности, в котором можно выделить следующие составляющие блоки:

- блок хранения данных – технологический элемент, реализуемый на базе СУБД, предназначен для хранения данных заявок и данных подсистемы анализа;
- блок анализа данных – ядро системы, в котором происходит подготовка исходных данных, построение скоринговых моделей, оценка и сравнение их эффективности, отслеживание их качества и устаревания, актуализация моделей;
- подсистема классификации, в которой происходит применение построенных моделей к новым данным, расчет количественных значений рисков, визуализация и сохранение рассчитанных значений рисков в соответствующих разделах электронных заявок, информирование о возможных формах дальнейшего сотрудничества с потенциальным заемщиком.

Математической основой систем оценки кредитоспособности являются статистические методы, в частности линейная регрессия, логистическая регрессия, деревья решений, деревья классификации, и методы оптимизации и искусственного интеллекта. Наиболее популярными сегодня являются три основных метода:

- на основе логистической регрессии;
- на основе дерева классификации;
- на основе нейронной сети.

Основное различие между этими тремя методами заключается в подходах к способам сегментации прецедентов обучающей выборки.

Сама сегментация имеет цель определить значимые факторы, влияющие на вероятности возможных исходов кредитных сделок, что возможно, если между сконструированными сегментами можно выявить статистически значимое различие в соотношении положительных и отрицательных прецедентов.

В методе логистической регрессии сегментация прецедентов осуществляется на основе разбиения факторного пространства  $n$ -мерной сеткой, где  $n$  – количество значимых факторов. В качестве исходного предположения принимается, что каждая ячейка сетки ( $n$ -мерный прямоугольник) объединяет прецеденты из обучающей выборки, характеризующиеся одинаковой вероятностью исхода. Координаты узлов этой сетки рассчитываются на основании статистических критериев, исходя из принципа максимальности различия между вероятностями исходов кредитных сделок для смежных сегментов прецедентов.

Соотношение положительных и отрицательных прецедентов в каждом сегменте используется для расчета скоринг-баллов в скоринговой карте, а координаты узлов сетки в факторном пространстве как раз и задают интервалы значений признаков в скоринговой карте.

Дерево классификаций (дерево решений) является более общим алгоритмом сегментации обучающей выборки прецедентов, чем логистическая регрессия. В отличие от метода логистической регрессии в методе дерева классификации сегментация прецедентов задается не с помощью  $n$ -мерной сетки, а путем последовательного дробления факторного пространства на вложенные прямоугольные области.

На первом шаге разделение выборки прецедентов на сегменты производится по самому значимому фактору. На втором и последующих шагах в отношении каждого из полученных ранее сегментов процедура повторяется до тех пор, пока никакой вариант последующего дробления не приводит к существенному различию между соотношением положительных и отрицательных прецедентов в новых сегментах. Количество ветвлений на каждом шаге процедуры построения дерева решений выбирается автоматически.

Нейронная сеть позволяет обрабатывать прецеденты обучающей выборки с более сложным видом сегментов. Геометрическая форма сегментов будет существенно зависеть от внутренней структуры нейронной сети, которая может быть настроена с учетом характера взаимосвязей между учитываемыми факторами.

В качестве скоринг-балла может выступать, например, эмпирически рассчитанная доля положительных прецедентов в сегменте. И тогда задача расчета скоринг-балла заемщика равносильна задаче отнесения заемщика к одному из построенных сегментов, что и делается в результате применения построенных скоринговых алгоритмов к новому заемщику.

Решается задача разработки данной системы, которая должна содержать полный набор методов, которые необходимы для хранения информации, расчета, прогнозирования кредитоспособности и выполнения других операций с базой данных.

Список использованных источников:

1. Руководство по кредитному скорингу / Элизабет Мэйз – Минск: ГревцовПаблицер, 2008. – 464 с.
2. Вишняков И.В. Методы и модели оценки кредитоспособности заемщиков. – СПб.: СПбГИЭА, 1998. – 267 с.

## **ПРОГРАММНАЯ ПОДДЕРЖКА РЕГРЕССИОННОГО ТЕСТИРОВАНИЯ МЕТОДОМ РЕДУКЦИИ ТЕСТОВОГО НАБОРА**

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

*Мацкевич С. А.*

*Поттосина С.А. – канд. физ.-мат. наук, доц.*

В настоящее время вопросу тестирования программного обеспечения уделяется все больше внимания, как со стороны производителей ПО, так и со стороны научной общественности. Потребность в решении задач тестирования ПО возникает при создании практически каждого программного продукта. При этом подходы к их решению могут существенно отличаться в зависимости от характеристик ПО подлежащего тестированию.

Регрессионное тестирование – это выборочное тестирование, позволяющее убедиться, что изменения не вызвали нежелательных побочных эффектов, или что измененная система по-прежнему соответствует требованиям. После каждой модификации программы необходимо удостовериться, что функциональность программы не оказал влияния модифицированный код. Регрессионное тестирование – дорогостоящий род деятельности: процесс регрессионного тестирования может включать исполнение достаточно большого количества тестов на скорректированной программе, даже если изменений очень мало. Несмотря на то, что усилия, требуемые для внесения небольших изменений, как правило, минимальны, они могут требовать достаточно больших усилий для проверки качества изменённой программы. Тем не менее, проведение регрессионного тестирования необходимо. Надёжная и эффективная разработка и сопровождение программного обеспечения невозможна без регрессионного тестирования. Выполнить полное регрессивное тестирование вряд ли возможно. По мере роста и расширения становится все сложнее тестировать отдельные части системы программного обеспечения. Эта проблема усложняется из-за частоты создания сборок программ. Необходимо тестировать предыдущую функциональность, чтобы обеспечить возможность тестирования новых исправлений и новых функций [1].

В настоящее время выделяют следующие методы отбора тестов для регрессионного тестирования: случайные методы; безопасные методы; методы минимизации; методы, основанные на покрытии кода. Каждый из представленных методов имеет свои достоинства и недостатки [1].

Целью работы является создание методики отбора тестов для проведения регрессионного тестирования, основанной на безопасном выборочном методе, позволяющем сократить временные затраты на проведение регрессионного тестирования без потери качества продукта.

В отличие от задач традиционной математики, где решение получается с помощью целенаправленной вычислительной процедуры, однозначно ведущей к цели, решение комбинаторной задачи сводится зачастую к полному перебору различных вариантов. Перебираются и испытываются конструкции определённого вида, среди которых должно находиться решение задачи. Как только выясняется, что очередная конструкция является решением, процесс поиска решения можно считать завершённым. Комбинаторные задачи характерны ещё тем, что множество, среди элементов которого отыскивается решение, всегда конечно. Реализовав полный перебор, либо найдём решение, либо убедимся в том, что решения нет. Таким образом, всякая подобная задача может быть решена за конечное время [2].

Рассмотрим два множества  $A = \{a_1, a_2, \dots, a_n\}$  и  $B = \{b_1, b_2, \dots, b_n\}$  некоторых явлений (событий), связанных отношением причинности: будем говорить, что явление  $a_i$  влечёт (имплицирует) явление  $b_j$ , если из существования  $a_i$  неизбежно следует существование  $b_j$  и явление  $a_i$  отрицает явление  $b_j$ , если из существования явления  $a_i$  неизбежно следует отсутствие явления  $b_j$ . Можно считать, что  $A$  является множеством причин, а  $B$  – множеством следствий. Можно также рассматривать  $A$  как множество некоторый явлений, а  $B$  – как множество сопутствующих им признаков.

В частности, множество  $A$  можно интерпретировать как множество всех функциональных модулей программы, которые должны подвергнуться проверке, а  $B$  – соответствующие элементам множества – тесты. Положим также, что вся информация, на основе которой можно делать свои предположения или, как говорят, ставить диагноз, содержится в булевой матрице  $D$ , называемой диагностической матрицей. Строки этой матрицы соответствуют элементам множества  $A$ , столбцы элементам множества  $B$ . Ограничимся рассмотрением случая полной связи между множествами явлений  $A$  и  $B$ . В первом случае элемент матрицы  $D$  на пересечении  $i$ -ой строки и  $j$ -ого столбца имеет значение 1, во втором – значение 0 [2].

Для матричной формулировки данной задачи надо ввести следующие понятия. Будем говорить, что некоторый столбец диагностической матрицы  $D$  различает две её строки, если одна из них содержит в данном столбце единицу, а другая – нуль. Некоторое подмножество  $D'$  столбцов матрицы  $D$  различает заданное подмножество  $R$  её строк, если для каждой пары строк и подмножества  $R$  найдётся в подмножестве  $D'$  различающий их столбец. Подмножество  $D'$  в этом случае представляет безусловный диагностический тест.

Практический интерес представляет задача нахождения минимального диагностического теста, которую можно свести к задаче о кратчайшем покрытии. Для этого сначала следует получить матрицу различий, строки которой соответствуют парам строк диагностической матрицы  $D$  и показывают, какими компонентами отличаются строки в этих парах. При этом используется покомпонентная операция сложения по модулю два, выполняемая над всеми над всеми парами строк матрицы  $D$ . Затем надо найти для полученной матрицы различий кратчайшее покрытие, только не строчное, а столбцовое, т.е. состоящее из некоторых столбцов, покрывающих в совокупности все строки данной матрицы. Множество признаков, соответствующих столбцам из найденного покрытия, будет искомым [2].

Рассмотрим полученную матрицу с точки зрения тестового набора регрессионного тестирования. Положим, строки этой матрицы, соответствующие элементам множества  $A$  – это множество всех функциональных модулей программы, которые должна подвергнуться проверке, а столбцы – соответствующие элементам множества – тесты. Тогда полученная матрица – и есть минимальный тестовый набор для регрессионного тестирования.

Рассмотрим на небольшом примере работу данного алгоритма. Пусть дана матрица  $Z$ .

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | a |
| 2 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | b |
| 3 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | c |
| 4 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | d |
| 5 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | e |
| 6 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | f |

Рис.1 – матрица  $Z$ 

Составим для матрицы  $Z$  матрицу различий, строки которой соответствуют парам строк диагностической матрицы и показывают, какими компонентами отличаются строки в этих парах. При этом будем использовать покомпонентную операцию сложения по модулю два, выполняя её над всеми парами строк диагностической матрицы.

|     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |    |
|-----|---|---|---|---|---|---|---|---|----|
| 1 2 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | ab |
| 1 3 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | ac |
| 1 4 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | ad |
| 1 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ae |
| 1 6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | af |
| 2 3 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | bc |
| 2 4 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | bd |
| 2 5 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | be |
| 2 6 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | bf |
| 3 4 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | cd |
| 3 5 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | ce |
| 3 6 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | cf |
| 4 5 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | de |
| 4 6 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | df |
| 5 6 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | ef |

Рис. 2 – Матрица различий для матрицы  $Z$ 

К данной матрице можно применить правила редукции, т.е. удалить из неё поглощающие строки и поглощаемые столбцы. В результате получаем, что одним из кратчайших столбцовых покрытий данной матрицы является совокупность столбцов 1, 3, 5 и 6. Соответствующая совокупность признаков является минимальной для заданной диагностической матрицы.

В рамках работы был предложен метод минимизации тестового набора на основе задачи о поиске минимального безусловного диагностического теста. В предложенном методе использовался подход, который обеспечивает безопасность и целесообразность уменьшения тестового набора. Анализ результатов проведённого на практике регрессионного тестирования подтверждает работоспособность и эффективность предложенного метода и средства поддержки регрессионного тестирования программного обеспечения: как было показано, уменьшение первоначального тестового набора достигает 25%.

Список использованных источников:

1. Котляров, В.П. Основы тестирования программного обеспечения: Учебное пособие. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. – 285 с.
2. Закревский А. Д. Основы логического проектирования. В 3 кн. Кн. 1. Комбинаторные алгоритмы дискретной математики / А. Д. Закревский, Ю. В. Поттосин, Л.Д. Черемисинова – Мн.: ОИПИ НАН Беларуси, 2004. – 226 с.

## МОДЕЛИРОВАНИЕ ОПЦИОНОВ И РАЗРАБОТКА ТОРГОВОЙ СИСТЕМЫ ДЛЯ БИРЖЕВОГО ТРЕЙДЕРА

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

Солоненко И.В.

Поттосина С.А., канд. физ.-мат. наук, доц.