Подходы к агрегации данных и извлечению факторов в задаче поиска мошенничества в банковских транзакциях

© О. И. Травкин

Московский государственный университет имени М. В. Ломоносова,
Москва
travkin.o.i@gmail.com

Аннотация

В данной статье проводится обзор подходов к агрегации данных и извлечению факторов для исследования потребительского поведения клиентов в задаче поиска мошенничества в банковских транзакциях. Для выявления мошеннических операций с банковскими картами очень важно анализировать историческое потребительское поведение клиентов. В данной статье рассмотрено два подхода. Первый подход – на основе трёх профилей клиентов: глобального, локального и частотно-временного. Глобальный профиль строится с помощью кластеризации клиентов исходя из характеристик их транзакций, что позволяет более точно работать с новыми или неактивными клиентами. Локальный профиль – исходя из исторического потребительского поведения каждого клиента. Частотно-временной профиль – с помощью анализа паттернов в операциях клиентов, построенных на основе частот совершения транзакций за определённый промежуток времени. Второй подход – RFM (Recency-Frequency-Monetary). Его суть заключается в расчёте периодичности, частоты и объёма проводимых клиентом операций за определённый промежуток времени. Кроме этого, предлагается модификация алгоритма DBSCAN для частичного обучения, которая может позволить значительно улучшить точность результатов поиска мошенничества на основе выявленных профилей и RFM характеристик.

Работа частично поддержана РФФИ (гранты 14-07-00548, 16-07-01028).

1 Введение

Число пользователей банковских карт в России растёт стремительно. К сожалению, ещё более стремительно растёт количество мошенничества с картами. По данным компании FICO за 2013 год Россия является самой быстрорастущей страной по объёму потерь от мошенничества с банковскими картами [11].

Нетрудно увидеть, что проблема выявления мошенничества в банковских транзакциях стоит достаточно остро. Эффективный инструмент решения проблемы мошенничества – использование алгоритмов машинного обучения. Но для этого, в первую очередь, необходимо определить факторы, позволяющие выявлять мошеннические операции. Особенностью рассматриваемой области является то, что использование сырых данных (поток транзакций) не даёт приемлемого результата [5]. Поэтому данная работа будет сосредоточена на том, чтобы сделать обзор существующих подходов к агрегации и извлечению факторов из транзакционных данных. При таком подходе учитывается важная для выявления мошенничества информация о прошлом поведении клиента и его потребительских привычках.

Существуют различные алгоритмы машинного обучения, а также различные подходы к обучению. Для эффективного и качественного решения задачи выявления мошенничества необходимо выбрать такой подход и алгоритм, который наилучшим образом впишется в рассматриваемую область. Есть основания полагать, что наилучшим вариантом будет подход с частичным обучением. Он учитывают кластерную структуру неразмеченных данных, одновременно учитывая размеченные обучающие примеры. В статье будет приведено обоснование того, почему частичное обучение подходит для решения задачи мошенничества с банковскими картами наилучшим образом. Кроме того, будет предложена модификация алгоритма DBSCAN для выявления мошенничества в банковских транзакциях на основе алгоритмов частичного обучения, тестирование которой будет проведено в рамках следующих работ.

Дальнейшее изложение будет организовано следующим образом: в секции 2 будет дано описание предметной области, в секции 3 – приведён перечень методов, используемых для выявления мошенничества с банковскими картами. В секции 4 будет дан обзор подходов к агрегации данных и извлечению факторов. Далее, в секции 5 будет

Труды XVIII Международной конференции DAMDID/RCDL'2016 «Аналитика и управление данными в областях с интенсивным использованием данных», Ершово, 11-14 октября 2016

рассмотрен алгоритм на основе частичного обучения. В секции 6 приведены результаты эксперимента по применению алгоритма к симуляционным данным.

2 Предметная область

Мошенничество с банковскими картами принято делить на две большие категории: заявочное и поведенческое мошенничество. Заявочное мошенничество может возникнуть при получении новой карты в компании эмитенте [4]. Для предотвращения мошенничества такого типа часто используют кредитный скоринг. Что касается поведенческого мошенничества, то его делят на 4 типа: кража почты (ситуация, когда мошенник получает доступ к конверту с картой до того, как она придёт к законному владельцу), потерянные и украденные карты, подделанные карты (создание физической копии карты) и «без предоставления карты». Мошенничество «без предоставления карты», в отличие от трёх предыдущих не требует наличия самой карты – оно совершается удалённо с помощью украденной информации о реквизитах кары [5]. Это очень удобный способ мошенничества, так как он почти полностью анонимный.

Финансовые институты борются с мошенничества на двух уровнях: противодействие мошенничеству и выявление мошенничества [4]. Противодействие мошенничеству включает в себя все действия и меры направленные на то, чтобы мошенничество никогда не случилось. Сюда можно отнести активацию карты перед первым использованием, одноразовые пароли, пин-код и так далее. Что касается выявления мошенничества, то сюда относятся системы и практики по скорейшему выявлению мошенничества, если оно уже произошло [4]. Чем скорее мошенничество будет выявлено, тем меньше будут потери от него, так как карта и все транзакции по ней будут заблокированы.

В данной статье рассматриваются подходы к выявлению поведенческого мошенничества. Прежде всего, необходимо дать определение мошенничеству в рамках выбранной категории: операцию с банковской картой клиента будем называть мошеннической, если она совершается без ведома клиента и против его интересов. Кроме того, необходимо учитывать, что мы можем выявить только те мошеннические операции, которые реализовались достаточное число раз, не похожи по некоторым своим характеристикам на предыдущие операции клиента [15], но похожи на предыдущие выявленные мошеннические транзакции [18].

3 Методы выявления мошенничества

Наибольший интерес представляют статистические методы выявления мошенничества. Они делятся на две большие группы: обучение с учителем и обучение без учителя. Обучение без учителя использует характеристики клиента или его транзакции, чтобы разделить их на небольшие кластеры, максимально непохожие друг на друга. Если новая транзакция не попадает в один из кластеров, считающийся нормальным, то срабатывает триггер для такой транзакции [4]. В тоже время большинство работ рассматривает обучение с учителем, которое использует прошлые мошеннические транзакции, чтобы сделать вывод о подозрительности текущих. Наиболее распространённым инструментом в данной предметной области для обучения с учителем являются искусственные нейронные сети [2,6,9, 12,15,21,23], так как обычно с их помощью достигается более высокое качество. Тем не менее, получаемые модели не интерпретируемы. В последнее время также часто используются ансамбли методов, например, метод случайного леса [3,8,26]. К оставшимся методам, используемым для выявления мошенничества, можно отнести рассуждения на основе прецедентов [25], Байесовские сети [15], деревья решений [20], логистическую регрессию [3,21], скрытые Марковские цепи [22], ассоциативные правила [19], метод опорных векторов [3] и генетические алгоритмы [10].

4 Стратегия агрегации данных

При разработке моделей выявления мошенничества с кредитными картами, обычно, изначально доступен только сырой набор данных, включающий в себя исключительно информацию по индивидуальным транзакциям. В Таблице 1 перечислены атрибуты, которые присутствуют в большинстве выборок данных о транзакциях.

Таблица 1 Типичный состав атрибутов

Имя атрибута	**Описание**
Transaction ID	Уникальный идентификатор транзакции
Time	Дата и время транзакции
Account number	Идентификатор клиента
Card number	Идентификатор карты
Transaction type	Internet, ATM, POS…
Amount	Сумма транзакции
Currency	Валюта транзакции
Merchant code	Код вида торговой точки (MCC Code)
Merchant group	Группа торговой точки
Country	Страна проведения транзакции

Чаще всего сырых данных недостаточно для построения качественной модели [5], так как при выявлении мошенничества необходимо учитывать поведенческие особенности каждого клиента. Для этого создаётся набор новых переменных, включающих в себя информацию о предыдущих транзакциях. Для создания таких переменных применяют так называемую стратегию агрегации [26]. Данный подход наиболее популярен на текущий момент. Смысл агрегации – собрать информацию о транзакциях клиента за последнее время: сумму и

количество транзакций в разрезе карты, страны, валюты, вида торговой точки и так далее. Один из подходов к учёту поведенческих особенностей клиента основан на профилировании, когда для каждого клиента выделяются три составляющие: глобальный профиль, локальный профиль и частотно-временной профиль [18]. Кроме того, существует RFM подход, который позволяет разносторонне исследовать исторический потребительский профиль клиента с помощью расчёта периодичности, частоты и объёма транзакций в различных разрезах [24].

4.3 Глобальный профиль

Глобальное профилирование необходимо для того, чтобы определить глобальные группы клиентов, основанные на характеристиках их транзакций. Это поможет охарактеризовать тех, о ком не достаточно данных в обучающей выборке [18]. Алгоритм профилирования следующий. В первую очередь, для каждого пользователя рассчитываются: общее количество транзакций, средняя сумма транзакций, среднее время между двумя последовательными транзакциями, число транзакций из других стран, типичное (модальное) время (часовой интервал) транзакции. Кластеризация производится с помощью итеративной версии DBSCAN с использованием расстояния Махаланобиса [16]. Итеративная версия используется для того, чтобы избежать некоторых недостатков алгоритма DBSCAN, возникающих при применении на несбалансированной выборке с несколькими большими и множеством маленьких кластеров. Количество итераций и начальные значения для алгоритма подбираются эмпирически [18].

Обновление кластеров происходит с некоторой периодичностью, например, раз в месяц. Более частое переобучение почти не имеет смысла в виду того, что будет относительное небольшое количество данных и характеристики клиентов меняются не так быстро.

После того, как выборка разбита на кластеры, нам необходимо уметь быстро определять принадлежность новой заявки к тому или иному кластеру. Для этих целей можно использовать дерево решений, которое будет обучено на полученных кластерах на основе тех же факторов.

4.2 Локальный профиль

Локальный профиль необходим для того, чтобы охарактеризовать индивидуальные поведенческие закономерности в транзакциях клиента. Процесс локального профилирования состоит в аппроксимации эмпирического распределения атрибутов транзакций гистограммой [18]. Выбран именно этот способ ввиду его простоты, наглядности и эффективности.

В процессе работы алгоритма для каждой новой транзакции используется метод HBOS [13]. С его помощью рассчитывается вероятность операции в контексте предыдущих. Для этого строятся одномерные гистограммы по значениям каждого из атрибутов. Гистограммы нормируются так, чтобы максимальная высота столбца была равна единице. Чтобы оценить аномальность транзакции рассчитывается следующая величина для каждой транзакции в контексте оцененного распределения:

$$log \frac{1}{hist_i(t_i)}$$

где t_i – это i-ый атрибут транзакции t, $hist_i(t_i)$ – высота столбца гистограммы, в который попало новое значение. Если пришло значение фактора, которое не встречалось ранее для этого клиента, то для него в гистограмме используется частота, рассчитанная по его кластеру из глобального профиля. Если это не возможно, то используется частота, оценённая на всей выборке. Таким же образом действуем и с клиентами, данных по которым очень мало.

Выбор временного периода для расчёта эмпирического распределения очень важен [1]. Мы остановимся на 2-х временных интервалах: 7 дней – для учёта наиболее актуальных потребительских трендов и 4 недели – для выявления более долгосрочных потребительских особенностей.

Обновление частот гистограмм должно осуществляться с периодичностью, не большей чем минимальное временное окно. В нашем случае – это 7 дней. Идеальный вариант - ежедневно. Обновление происходит с использованием экспоненциального сглаживания. Это позволяет не делать расчёты со старыми данными, снижая нагрузку на вычислительную систему, и, в тоже время, позволяет учесть прошлую информацию о сделках в распределении факторов:

$$hist_i(t_i) = a * hist_i(t_i)_t + (1 - a) * hist_i(t_i)_{t-1},$$

где параметр a выбирается эмпирически.

Предлагается строить гистограммы для следующих атрибутов: номинальные – Merchant code, Merchant group, Country, Currency, Transaction type; интервальные – Amount, Time. Для номинальных переменных мы просто подсчитываем частоту появления каждой группы. Для интервальных – проводится процедура разбиения на интервалы. Для Time – оно переводится в часы, округляясь в большую сторону если минут больше чем 30 и в меньшую в противоположном случае, например: 21.03.2011 21:31 станет 22. Далее считается частота появления каждого часа. Для Amount – разбивается на 10 равномерных интервалов между максимальным и минимальным значением для клиента за период и подсчитывается число попаданий в каждый из интервалов [13].

4.3 Частотно-временной профиль

С помощью данного профиля выявляются мошеннические транзакции, которые маскируются под не мошеннические. Примером могут служить

небольшие по сумме, но очень частые транзакции, которые не будут пойманы с помощью локального или глобального профилей.

Таким образом, для каждого клиента рассчитывается абсолютная сумма транзакций в этот день, количество транзакций в день и максимальное количество транзакций в день за последнее время. Для каждого из этих факторов рассчитывается среднее значение и стандартное отклонение за выбранный период. Аномальной будет считаться та транзакция, которая выходит за интервал: среднее значение +/- стандартное отклонение. Для редко расплачивающихся картами клиентов данный профиль не создаётся. Обновление среднего значения также происходит с помощью экспоненциального сглаживания.

4.4 RFM

RFM подход основывается на расчёте периодичности, частоты и объёма транзакций клиента в различных разрезах и за различные периоды времени [24]. В первую очередь определим временные периоды. Возьмём такие же, как для построения локального профиля. Теперь определимся с разрезами. Наиболее удобным форматом для представления изучаемых разрезов и полученных факторов будет таблица, которая изображена далее.

Таким образом, мы видим, что RFM подход является аналогом частотно-временного профиля, но более детализированным. Кроме того, в рамках данного подхода выделяются факторы, характеризующие первичность транзакции в рассматриваемом временном интервале. Таким образом, оба подхода могут быть гармонично объединены в один, что должно повысить ранжирующие способности моделей, построенных на извлечённых факторах.

5 Частичное обучение

Одной из особенностей мошенничества с банковскими картами является то, что возможности банка по разметке обучающих данных сильно ограничены. Лишь небольшая доля сделок попадает в расследования специалистов противодействия мошенничеству, а клиенты не всегда сообщают о фактах мошенничества с их картами. Это приводит к тому, что в данных очень мало размеченных транзакций, а те что размечены имеют тенденцию быть мошенническими. Всё это снижает эффективность методов обучения с учителем. Кроме того, мошенники часто меняют свои стратегии вывода средств, чтобы оставаться непойманными, что снижает эффективность методов обучения с учителем ещё сильнее, так как алгоритмы этого типа наиболее чувствительны к тем мошенническим схемам, которые встречаются в выборке для обучения наиболее часто. Одним из решений описанных сложностей могло бы стать использование алгоритмов машинного обучения без учителя, для выявления аномальных транзакций или обнаружения кластеров в пространстве рассматриваемых признаков, но это приводит к другим, возможно более серьёзным проблемам. Без использования размеченных обучающих примеров, полученные результаты могут быть непредсказуемыми и тяжело трактуемыми. Именно поэтому алгоритмы обучения без учителя не получили широкого распространения в решении проблемы выявления мошенничества в банковских транзакциях. Наилучшим компромиссом в рассматриваемой ситуации будут алгоритмы с частичным обучением, которые находятся посередине между двумя упомянутыми ранее типами.

Таблица 2 Разрезы для расчёта факторов в рамках RFM

Recency	Время, прошедшее с предыдущей транзакции
MC	для данного вида торговой точки
MC Category	для данной категории торговой точки
Global	для всех транзакций клиента
Country	для всех транзакций в той же стране
Currency	для всех транзакций в той же валюте
Transaction type	для всех транзакций того же типа
Frequency	Общее количество транзакций
MC	для данного вида торговой точки
MC Category	для данной категории торговой точки
Global	для всех транзакций клиента
Country	для всех транзакций в той же стране
Currency	для всех транзакций в той же валюте
Transaction type	для всех транзакций того же типа
Monetary Value	Средняя сумма транзакции
MC	для данного вида торговой точки
MC Category	для данной категории торговой точки
Global	для всех транзакций клиента
Country	для всех транзакций в той же стране
Currency	для всех транзакций в той же валюте
Transaction type	для всех транзакций того же типа
Event occurrence	Первая покупка?
MC	для данного вида торговой точки
MC Category	для данной категории торговой точки
Global	для всех транзакций клиента
Country	для всех транзакций в той же стране
Currency	для всех транзакций в той же валюте
Transaction type	для всех транзакций того же типа

5.1 Основные принципы

Задача частичного обучения ставится следующим образом. Есть множество объектов X и множество классов Y. $X^l = \{x_1, \dots, x_l\}, \{y_1, \dots, y_l\}$ – размеченная выборка, $X^k = \{x_{l+1}, \dots, x_{l+k}\}$ – неразмеченная

выборка. Необходимо построить алгоритм классификации a: $X \to Y$.

Существует несколько подходов к решению данной задачи. Первый и наиболее простой подход – это эвристические методы, такие как self-training и co-learning. Такие подходы требуют многократного обучения, поэтому вычислительно неэффективны. Второй – модификации методов кластеризации. Он достаточно прост в реализации (необходимо внести лишь некоторые ограничения), но, как правило, трудоёмкий в вычислениях. Наконец, модификации методов классификации. Данный подход реализуется сложнее, но даёт более вычислительно-эффективные методы.

5.2 Применение в области выявления мошенничества с банковскими картами

Применение методов частичного обучения в задачах поиска мошеннических транзакций весьма перспективна ввиду причин, перечисленных выше. Но, к сожалению, существуют ограничения, которые должны быть наложены на алгоритмы и в рамках частичного обучения. В ситуации, когда мы имеем практически один размеченный класс, выбор алгоритмов существенно ограничен: модификации алгоритмов классификации будут заведомо хуже или вовсе не применимы, так как для них необходима выборка хотя бы с двумя размеченными классами.

Прежде чем окончательно определиться с используемым алгоритмом, необходимо сделать предположение о структуре данных. Вероятнее всего, что мошеннические операции представляют собой небольшие скопления в пространстве признаков. Это обосновано тем, что мошенники часто действуют очень схожим образом: отрабатывают определённую работающую схему до тех пор, пока её не закроют, либо мошеннические действия совершает вредоносная программа на электронном устройстве клиента, которая действует по чёткому алгоритму. В тоже время поведение мошенников изменчиво, так как они находятся в постоянном поиске новых лазеек в системах противодействия и выявления мошенничества.

Исходя из представленных выше предположений, наилучшим будет тот алгоритм, который умеет выделять небольшие плотные скопления и ему не нужно задавать их количество. Одним из возможных вариантов являются алгоритмы на основе плотности, например, DBSCAN. В данной работе будет рассмотрена модификация данного алгоритма под поставленную задачу частичного обучения с одним размеченным классом.

Стоит отметить, что на текущий момент существуют реализации алгоритма DBSCAN с частичным обучением, например [14], но они не пригодны для применения в рамках описанной ранее предметной области, где преобладают наблюдения с одним размеченным классом.

5.3 DBSCAN

DBSCAN (Density Based Spatial Clustering of Applications with Noise) [17] – это алгоритм кластеризации, основанный на плотности и работающий следующим образом: пусть имеются точки в некотором пространстве, алгоритм объединяет вместе точки, находящиеся близко друг к другу (которые имеют много близкорасположенных соседей), а те точки, что лежат в областях с низкой плотностью (ближайшие соседи расположены далеко) оставляет в качестве шума.

Перейдём к модификации данного алгоритма для возможности использования частичного обучения с учётом ограничений, накладываемых предметной областью. Основные функции:

1. **expandCluster** (*D, D_l, P, NeighborPts, C, eps, MinPts*) – функция расширения кластера за счёт соседей, обладающих необходимыми параметрами *eps* и *MinPts*.
2. **trueEps** (*D, D_l, P*) – функция рассчитывающая *eps* и *MinPts* для оптимальной кластеризации.

Здесь D – все не размеченные точки, D_l – все размеченные точки, P – точка вокруг которой ищются соседи, *NeighborPts* – соседи рассматриваемой точки с необходимыми параметрами *eps* и *MinPts*.

В данной реализации поиск кластеров производится вокруг уже известных (размеченных) мошеннических наблюдений, что позволяет выявить другие потенциально мошеннические транзакции, а одна точка может принадлежать сразу нескольким кластерам. Смысл этих изменений в том, чтобы позволить алгоритму разбивать всё пространство на области, каждая из которых характеризуется некоторой мерой, отражающей шанс того, что в ней содержатся мошеннические транзакции. Причём некоторые из областей, вероятно, будут содержать в себе другие полученные области. Такой подход позволяет более тонко управлять процессом выявления мошенничества в зависимости от соотношения цены ошибки первого и второго рода.

Алгоритм выполняет следующие действия: для каждой размеченной точки он находит другую ближайшую размеченную точку и строит гиперсферу, диаметром которой является прямая, соединяющая две эти точки (eps'). С помощью гиперсферы мы оцениваем плотность точек в пространстве между двумя выбранными (подразумевается, что они принадлежат одному кластеру) для того, чтобы подобрать параметры DBSCAN, максимально отражающие структуру данных кластера. Параметру алгоритма *eps* присваивается значение равное радиусу сферы, умноженное на долю объёма гиперсферы, не заполненную точками:

$$0.5 * eps' * (1 - \frac{(V\ of\ n-cube) * len\left(distinct\left(scale(trueNeighbors)\right)\right)}{(V\ of\ n-sphere)})$$

Рассмотрим выражение:

$$\frac{(V\ of\ n{-}cube)*len\left(distinct\left(scale(trueNeighbors)\right)\right)}{(V\ of\ n{-}sphere)},$$

где $trueNeighbors$ – это число точек внутри гиперсферы, $(V\ of\ n-cube)$ – объём гиперкуба, построенного вокруг точки, $len\left(distinct\left(scale(trueNeighbors)\right)\right)$ – число уникальных точек, приведённых к необходимой шкале, $(V\ of\ n-sphere)$ – объём гиперсферы. Фактически, это отношение выражает долю объёма гиперсферы, заполненную точками. Процесс шкалирования – это преобразование коррдинат точек к такому виду, чтобы можно было заполнить гиперсферу неперсекающимися гиперкубами определённого размера, построенными вокруг точек. Например, заполним гиперсферу гиперкубами со стороной $k = eps'/100$, тогда, если мы преобразуем каждую координату как $round(x_i/k, 0) * k$, то сможем заполнить всё пространство внутри гиперсферы не персекающимися гиперкубами со стороной k. Таким образом, каждую уникальную точку (некоторые из них могли сойтись в одну из-за округления) мы заменяем гиперкубом и складываем их площади, получая оценку площади, занятой точками. Поделив это на объём гиперсферы, который можно приближённо вычислить (для n>3) по формуле: $V_n(R) \sim \frac{1}{\sqrt{\pi * n}} * \left(\frac{2\pi e}{n}\right)^{0.5*n} * R^n$, и вычитая из единицы, получаем долю объёма сферы, не заполненную точками. Умножая это на eps' получаем величину eps, которая тем больше, чем меньше заполнена гиперсфера. Этот же коэффициент применяем для определения оптимального *MinPts*: чем меньше заполнена сфера, тем меньше надо соседних точек для включения в кластер. Далее алгоритм, используя полученные параметры *MinPts* и *eps*, находит соседей двух рассматриваемых точек, объединяет их и действует по схеме функции **expandCluster**.

Таким образом, мы получили алгоритм, принимающий на вход две выборки (размеченные и не размеченные) и подбирающий все необходимые параметры для выявления кластеров автоматически. Это позволяет исключить практически все общеизвестные недостатки оригинального алгоритма DBSCAN: нет проблемы с граничными точками, так как фактически есть только один тип для кластера; значения параметров *MinPts* и *eps* подбираются автоматически; нет проблемы с различной плотностью кластеров, так как параметры подбираются индивидуально для каждого потенциального кластера. К сожалению, все эти изменения делают алгоритм более трудоёмким с точки зрения вычислений. Для того, чтобы облегчить вычисления, можно ограничивать возможные значения *eps*, так как в расмотрении очень больших областей смысла нет. Кроме того, есть потенциал в распараллеливании данного алгоритма и применении технологий Apache Spark [20].

5.4 Отбор факторов

Одним из важнейших этапов в процессе кластеризации является отбор факторов. Включение незначимых или сильно коррелирующих факторов может привести к тому, что адекватные кластеры не будут найдены. Существует два очевидных подхода по отбору факторов для кластеризации: использование априорных соображений и анализ важности факторов на специально подготовленной размеченной выборке. В нашем случае априорные соображения учтены полностью ввиду специфики подготовки факторов (подробнее об этом в секции 4). Что касается анализа на размеченной выборке, то этот способ кажется очень удобным в рамках поставленной задачи. Таким образом для отбора факторов предлагается использовать деревья решений. Это связано с необходимостью учитывать возможную нелинейность факторов относительно мошенничества, а деревья решений являются наиболее простым и понятным подходом для решения таких задач. Один из возможных вариантов реализации подхода к отбору факторов на основе деревьев решений заключается в следующем: для каждого фактора строится своё дерево решений, которое предсказывает известные факты мошенничества против всего остального. Предварительно отбираем обучающую выборку следующим образом: берём все размеченные данные, а потом дополняем их неразмеченными так, чтобы соотношение мошеннических транзакций к остальным было 1:1. Дальнейший алгоритм построен следующим образом: фактор разбивается на 40 равномерных интервалов (для интервальных факторов), каждому интервалу присваивается номер, и на этих номерах строится дерево решений, которое в итоге должно выдать не более чем 7 итоговых интервалов. Для номинальных переменных в дереве используются их фактические значения. Разбиение на 40 интервалов необходимо для того, чтобы обезопаситься от переобучения и получения очень маленьких итоговых интервалов. Для номинальных (категориальных) переменных также используется дерево решений, но без предварительного разбиения на интервалы. После этого рассчитывается коэффициент Gini для каждой переменной, отражающий способность фактора к ранжированию. Порог отсечения по Gini для факторов подбирается эмпирически. Полученный список значимых факторов проверяем на корреляцию и исключаем один из тех, по кому корреляция составила более 70%. Предпочтение отдаётся фактору, имеющему больший Gini. Таким образом, получаем финальный список факторов для кластеризации.

5.5 Обработка результатов

В результате применения описанного алгоритма, на выходе получаем набор кластеров, каждый из

которых характеризуется некоторой мерой, отражающей шанс того, в нём содержатся мошеннические транзакции, например, плотность кластера: отношение количества элементов в кластере к оценке объёма кластера. Кроме того, можно учитывать расстояние от точки до центра кластера, корректируя вероятность быть мошеннической для конкретной точки внутри кластера.

Теперь мы можем каждой новой транзакции поставить в соответствие выбранную меру, чтобы определить её шансы быть мошеннической (предварительно отнеся её к одному из кластеров). Далее, в зависимости от политики банка, применяются различные меры по борьбе с мошенничеством.

6 Результаты эксперимента

Результаты работы алгоритма были оценены на 10 симуляционных двумерных выборках. Двумерная выборка была выбрана в виду наибольшей наглядности результатов. Выборки были сформированы следующим образом. Вокруг трёх последовательно расположенных опорных точек были сформированы кластеры различной плотности. Кластеры формировались таким образом, чтобы точки внутри каждого кластера были распределены нормально по обеим координатам со средним значением равным соответствующей координате опорной точки. По такому же принципу в каждый из кластеров в небольшом количестве, пропорциональном его размеру, были добавлены размеченные (мошеннические) точки. После этого, равномерно по всему рассматриваемому пространству были добавлены точки шума, а также размеченные точки, не попадающие в кластеры. Пример на рисунке ниже. Укрупнённые точки – это размеченные (мошеннические) точки.

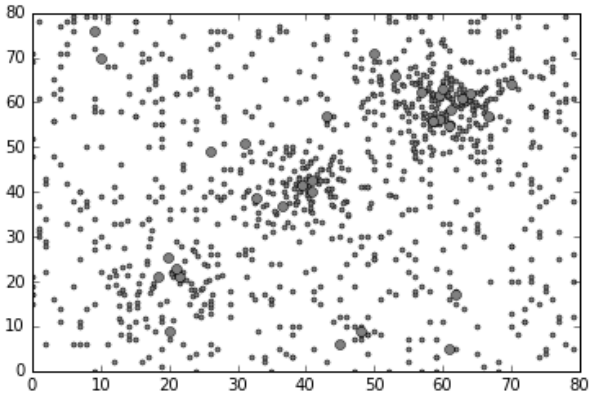

Рисунок 1 Пример симуляционной выборки

В рамках рассматриваемой задачи данный алгоритм будет применяться для классификации транзакций, поэтому в качестве метрики качества было решено использовать индекс Джини. Чтобы рассчитать данный индекс, предполагалось, что точки, находящиеся внутри кластера также являются мошенническими (данное предположение не использовалось для бучения, а только для расчёта качества). В результате оценивалось то, насколько точно и полно выделенные кластеры соответствуют реальным мошенническим кластерам.

Для того чтобы оценить качество разработанного алгоритма, результаты его работы были сравнены с результатами алгоритма HDBSCAN [7]. Его основные преимущества в том, что он применяет алгоритм DBSCAN к данным, используя различные значения параметра eps, подбирая лучшую кластеризацию, на основе стабильности данного параметра. Для работы алгоритм требует только один параметр – минимальный размер кластера. После применения алгоритма HDBSCAN так же оценивалось то, насколько точно и полно выделенные кластеры соответствуют реальным кластерам. Единственным исключением было то, что кластеры, которые не содержали изначально размеченные данные, исключались из анализа для повышения точности.

В результате была получена оценка среднего индекса Джини для алгоритма с частичным обучением: 85%, и для алгоритма HDBSCAN: 71,7%. Таким образом, мы видим, что предложенный подход оказывается в среднем лучше на 13 процентных пунктов.

Заключение

В данной статье произведён обзор подходов к агрегации данных и извлечению факторов в задаче поиска мошенничества с банковскими картами. Кроме того, обсуждалась предобработка данных с использованием деревьев решений и отбор факторов для оптимальной кластеризации, а также представлен новый алгоритм, являющийся модификацией DBSCAN для применения в задаче частичного обучения. Как показали эксперименты на симуляционных данных, данный алгоритм получает устойчивые положительные результаты на различных выборках без подбора параметров, необходимых классическому DBSCAN, кроме того, алгоритм показал себя лучше, чем алгоритм HDBSCAN.

В следующих работах будет произведено тестирование всех описанных подходов на реальных данных банковских транзакций.

Литература

[1] Alejandro Correa Bahnsen, Djamila Aouada, Aleksandar Stojanovic, Björn Ottersten. Feature engineering strategies for credit card fraud detection. Expert Systems With Applications 51 pp. 134–142, 2016.

[2] Aleskerov, E., Freisleben, B., Rao, B. Cardwatch: A neural network based database mining system for credit card fraud detection. In: Computational Intelligence for Financial Engineering (CIFEr),

1997., Proceedings of the IEEE/IAFE 1997. IEEE, p. 220–226, 1997.

[3] Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J. C. Data mining for credit card fraud: A comparative study. Decision Support Systems 50 (3), p. 602–613, 2011.

[4] Bolton, R. J., & Hand, D. J. Statistical fraud detection: A review. Statistical Science, 17(3), p. 235–249, 2002.

[5] Bolton, R. J., & Hand, D. J. Unsupervised profiling methods for fraud detection. In Conference on credit scoring and credit control, Edinburgh, 2001.

[6] Brause, R., Langsdorf, T., Hepp, M. Neural data mining for credit card fraud detection. In: Proceedings. 11th IEEE International Conference on Tools with Artificial Intelligence. IEEE, p. 103–106, 1999.

[7] Campello R., Moulavi D., and Sander J. Density-Based Clustering Based on Hierarchical Density Estimates. In Advances in Knowledge Discovery and Data Mining, Springer. P. 160-172, 2013.

[8] Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., Bontempi, G. Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications 41 (10), p. 4915–4928, 2014.

[9] Dorronsoro, J. R., Ginel, F., Sgnchez, C., Cruz, C. Neural fraud detection in credit card operations. Neural Networks, IEEE Transactions on 8 (4), p. 827–834, 1997.

[10] Duman, E., Elikucuk, I. Solving credit card fraud detection problem by the new metaheuristics migrating birds optimization. In: Rojas, I., Joya, G., Cabestany, J. (Eds.), Advances in Computational Intelligence. Vol. 7903 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 62–71, 2013..

[11] FICO. Evolution of card fraud in Europe. Russia http://www.fico.com/landing/fraudeurope2013/country.php?countrycode=RUS

[12] Ghosh, S., Reilly, D. L. Credit card fraud detection with a neural-network. In: Proceedings of the Twenty-Seventh International Conference on System Sciences. Vol. 3. IEEE, p. 621–630, 1994.

[13] Goldstein, M., Dengel, A. Histogram-Based Outlier Score (HBOS): A Fast Unsupervised Anomaly Detection Algorithm, 2012.

[14] Levi Lelis, Jörg Sander. Semi-Supervised Density-Based Clustering. ICDM '09 Proceedings of the 2009 Ninth IEEE International Conference on Data Mining. p. 842-847. 2009

[15] Maes, S., Tuyls, K., Vanschoenwinkel, B., Manderick, B. Credit card fraud detection using Bayesian and neural networks. In: Proceedings of the 1st international naiso congress on neuro fuzzy technologies, 2002.

[16] Mahalanobis, Prasanta Chandra. On the generalised distance in statistics. Proceedings of the National Institute of Sciences of India 2 (1). p. 49–55, 1936.

[17] Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. p. 226–231, 1996.

[18] Michele Carminati, Roberto Caron, Federico Maggi, Ilenia Epifani, Stefano Zanero. BankSealer: An Online Banking Fraud Analysis and Decision Support System. ICT Systems Security and Privacy Protection, Volume 428 of the series IFIP Advances in Information and Communication Technology, p. 380-394, 2014.

[19] S´anchez, D., Vila, M., Cerda, L., Serrano, J.-M. Association rules applied to credit card fraud detection. Expert Systems with Applications 36 (2), p. 3630–3640, 2009.

[20] Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills. Advanced Analytics with Spark, Patterns for Learning from Data at Scale. O'Reilly, Pages: 276, 2015.

[21] Shen, A., Tong, R., Deng, Y. Application of classification models on credit card fraud detection. In: Service Systems and Service Management, 2007 International Conference on. IEEE, p. 1–4, 2007.

[22] Srivastava, A., Kundu, A., Sural, S., Majumdar, A. K. Credit card fraud detection using hidden markov model. Dependable and Secure Computing, IEEE Transactions on 5 (1), p. 37–48, 2008.

[23] Syeda, M., Zhang, Y.-Q., Pan, Y. Parallel granular neural networks for fast credit card fraud detection. In: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems. Vol. 1. IEEE, p. 572–577, 2002.

[24] Veronique Van Vlasselaer, Cristian Bravo, Olivier Caelen, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, Bart Baesens, APATE: A Novel Approach for Automated Credit Card Transaction Fraud Detection using Network-Based Extensions, Decision Support Systems, 2015.

[25] Wheeler, R., Aitken, S. Multiple algorithms for fraud detection. Knowledge-Based Systems 13 (2), p. 93–99, 2000.

[26] Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., Adams, N. M. Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery 18 (1), p. 30–55, 2009.

Data aggregation and feature extraction strategies for credit card fraud detection

Oleg Travkin

This paper provides an overview of approaches to data aggregation and feature extraction strategies for credit

card fraud detection. In order to identify credit card fraud, it is very important to analyze historical spending behavior of customers. This paper discusses two approaches. The first approach is based on global, local and temporal customer profiles. Global profile is built via clustering customers based on characteristics of transactions, that allows analyze new or inactive customers more accurate. Local profile is based on historical consumer behavior of each client. Temporal profile is based on analyzing patterns in customer transactions that are based on its frequency for a certain period of time. The second approach is called RFM (Recency-Frequency-Monetary). Within this approach the recency, the frequency and monetary volume of customer transactions are calculated for a certain period of time. In addition, we proposed semi-supervised modification of DBSCAN algorithm, which may allow to significantly improve the accuracy of modeling of fraud based on identified profiles and RFM characteristics.