УДК 004.622

А. В. Молдавская[1], В. М. Рувинская
Одесский национальный политехнический институт
Проспект Шевченко, 1, 65044 Одесса, Украина
[1]e-mail: am.poly@ya.ru

Методика предобработки данных в задаче секвенциального анализа

Статья посвящена разработке методики предобработки данных при проведении секвенциального анализа. Выделены и экспериментально подтверждены проблемы предобработки в данной области: зашумленность данных, избыточность алфавита, возможное наличие циклических повторов. Предложены способы их решения. Эффективность и результативность предложенных подходов показана на экспериментах.

Ключевые слова: *интеллектуальный анализ данных, предобработка данных, секвенциальный анализ, паттерны, последовательности.*

Введение

Перед интеллектуальным анализом данных (data mining) постоянно возникают новые задачи, прежде всего, в связи с тем, что постоянно увеличивается объем обрабатываемой информации, ее разнотипность, поэтому данная область пополняется новыми направлениями исследований и новыми методами. Их практическое использование и применимость зависят от выполнения некоторых предварительных этапов. Данная статья касается особенностей секвенциального анализа — разновидности интеллектуального анализа, в которой объектом являются последовательные данные.

Целью исследования является повышение эффективности секвенциального анализа и увеличение его результативности за счет применения способов и методики предобработки исходных данных. В связи с этим поставлены следующие *задачи*:

— проанализировать этапы применения методов интеллектуального анализа в целом, выделить и описать этап предобработки данных;

— изучить особенности работы методов секвенциального анализа с данными, выявить слабые стороны этих методов, связанные с недостатками исходной выборки.

— разработать способы и методику предобработки данных для секвенциального анализа, позволяющую убрать такие недостатки.

Предобработка данных и ее место в интеллектуальном анализе

Использование методов интеллектуального анализа подразумевает прохождение следующих шагов [1].

1. *Постановка задачи и выбор подходящего метода интеллектуального анализа.* На этом шаге проводится анализ задачи и особенностей области знаний. Выбирается набор входных и выходных (целевых) параметров. Также происходит выбор метода интеллектуального анализа, сравнивается эффективность различных методов применительно к данной задаче.

2. *Организация сбора и хранения данных.* Проектирование, создание или выбор существующего хранилища данных. Важно учитывать требования выбранного на первом шаге метода интеллектуального анализа, а также способ сбора и обновления данных [2].

3. *Предобработка (препроцессинг) данных.* Преобразование данных перед тем, как они будут обработаны алгоритмом анализа. Предобработка обеспечивает более эффективную работу методов анализа.

4. *Работа метода интеллектуального анализа*. На этом этапе выбранный ранее метод интеллектуального анализа применяют на обработанных данных. Возможно многократное применение с уточнением параметров метода. В случае, если целью анализа является построение модели, то на этом шаге происходит обучение модели.

5. *Постобработка результатов*. Применение дополнительных методов, позволяющих сократить результирующую выборку, отобрать наиболее значимые результаты, а также привести выборку в какую-либо иную форму (например, в форму графа).

6. *Анализ и интерпретация полученных знаний.* Проводится экспертная оценка полученных зависимостей и шаблонов.

7. *Работа с моделью.* Если интеллектуальный анализ был применен для построения модели, то теперь происходит использование этой модели для классификации или регрессии.

Рассмотрим шаг предобработки, или препроцессинга данных. Задачей предобработки является повышение качества данных — совокупности свойств и характеристик данных, влияющих на их пригодность для последующего анализа [4]. Предобработкой данных является также приведение исходных данных в ту форму, которая может быть обработана выбранным методом интеллектуального анализа [3], и многие методы имеют те или иные готовые практические реализации, которые налагают свои требования на формат входных данных. В этом случае на шаге предобработки требуется привести данные в этот формат.

Задачи предобработки данных

Существуют работы, детально описывающие методы предобработки данных для data mining в целом [3, 5, 6].

Необработанные исходные данные могут содержать следующие недостатки.

1. *Разнотипность данных*. Встречается в данных, собранных из различных источников. Например, в медицине данные в выборке могут быть получены из электрокардиограмм, рентгенограмм, трехмерных томограмм и тому подобное

[3]. Возможно разделение данных на несколько разных выборок по типу или же приведение данных к единому типу, если предметная область позволяет это.

2. *Незавершенность*: нехватка параметров или значений параметров. Причиной может являться ошибка хранилища данных, ошибка при заполнении или сборе значений параметров. Пример: возраст = « ».

3. *Зашумленность*: содержит ошибки или значения параметров, выходящие за пределы допустимого диапазона. Чаще всего является результатом ошибок при передаче данных либо человеческой ошибки. Пример: возраст = «0». Также данные могут содержать так называемые выбросы — реальные, но излишне нетипичные данные.

4. *Несовместимость*: значения некоторых параметров противоречат друг другу. Такое бывает в случаях, когда данные получены из разных источников, либо произошла ошибка на уровне хранилища данных при операции объединения полей. Пример: возраст = «30», дата рождения = «01/01/2016».

5. *Дублирующиеся записи* [7].

Соответственно, на этапе предобработки могут потребоваться:

— перевод данных в единый формат, выделение признаков с помощью эксперта либо методов автоматического выделения (feature extraction) [8];

— очистка данных от ошибочных данных или слишком выбивающихся единичных элементов (выбросов);

— дополнение недостающих данных;

— очистка от дублирующихся записей;

— избавление от зашумленности (фильтрация) либо незначительных признаков;

— нормализация выборки средствами статистического анализа;

— разделение выборки на обучающую и контрольную группы [9].

Предобработка в секвенциальном анализе

Опишем методику предобработки, предлагаемую нами к использованию при проведении секвенциального анализа.

Секвенциальным анализом называется процесс добычи закономерных, часто встречающихся подпоследовательностей, называемых *паттернами*, из выборки последовательностей. Секвенциальный анализ — одно из направлений добычи знаний.

Последовательность — это кортеж из наборов элементов (itemsets) — непустых множеств одновременно встречающихся элементов [10]. Пример последовательности: $S = <\{a\},\{a, b, c\},\{b\},\{b, c\},\{a, d\}>$. Строки текста также являются последовательностями, состоящими из единичных наборов элементов: $S = <\{s\},\{t\},\{r\},\{o\},\{k\},\{a\}>$. Итак, последовательность содержит не только наборы элементов, но и *порядок* их следования. Примерами последовательных данных могут служить любые динамические данные, логи посещений различных страниц вебсайта пользователями, данные о покупках, совершенных в разное время одним и тем же потребителем, данные о поведении программ в системе, данные о цепочках ДНК. Последовательность обычно не содержит временных меток.

При определении того, является ли подпоследовательность паттерном, используется величина *поддержки* — частота появления паттерна в исходной вы-

борке [10]. При проведении анализа обычно задается минимальная допустимая поддержка, а алгоритм анализа выдает все паттерны, которые имеют поддержку, равную минимальной, или большую. Также уточненное определение паттерна может налагать дополнительные ограничения на те частые подпоследовательности, которые попадают в результирующую выборку. Такие уточненные разновидности паттернов достаточно подробно освещены в [11]. В частности, при поиске *закрытых паттернов* не учитывают те подпоследовательности, которые полностью входят в более длинные частые подпоследовательности и при этом имеют ту же поддержку.

Существует ряд алгоритмов секвенциального анализа (Bide+, CloSpan, SPADE, PrefixSpan и другие) с различным подходом к поиску паттернов, практические реализации большинства из них доступны в открытом фреймворке SPMF [12]. Все эти алгоритмы не включают в себя этап предобработки данных, однако требуют его проведения для корректной работы.

Предобработка для секвенциального анализа проводится на последовательных данных либо включает в себя приведение исходных данных к последовательным. Цель предобработки заключается в приведении данных в необходимый формат, а также в уменьшении объема малозначимых и ошибочных данных в соответствии с принятыми на первом шаге постановкой задачи и методом анализа. Направления предобработки выделим следующие:

— уменьшение зашумленности выборки за счет удаления коротких последовательностей;

— сокращение объема алфавита, то есть сокращение количества различных элементов выборки;

— выявление и удаление из выборки циклически повторяющихся подпоследовательностей (тандемных повторов).

Далее рассмотрим проблемы исходных данных, в связи с которыми требуются перечисленные направления предобработки.

Уменьшение зашумленности выборки: проблема и способы решения

Очевидно, что если в исходной выборке встречаются последовательности, слабо характеризующие предметную область и потому не содержащие ни одного паттерна, то они влияют на достоверность результатов. В частности, наличие таких последовательностей делает невозможным получение каких-либо результатов при поддержке в 100 %, т.е. не будет существовать ни одного паттерна, который встречался бы во всех последовательностях выборки. Такими могут быть последовательности, полученные ошибочно, либо же слишком короткие — состоящие всего из нескольких элементов. Все они должны быть выявлены и убраны на этапе предобработки. Достаточно применить экспертную оценку и примерно определить длину, при которой последовательность никак не может характеризовать предметную область. К примеру, для большинства вредоносных программ класса «троян» известно, что для своего внедрения в систему они должны выполнить не менее 4-х различных действий в системе: поиск по реестру, открытие/создание записи реестра, изменение значения этой записи, закрытие записи. Поэтому слишком короткими в этом случае будут считаться последовательности длины, меньшей четырех.

Для выборок, содержащих данные о явлениях фиксированной длины — к примеру, помесячная статистика за несколько десятков лет — этап предобработки должен включать в себя очистку от всех последовательностей, длина которых не равна зафиксированной.

Сокращение объема алфавита

Помимо уменьшения объема выборки, может также потребоваться уменьшение объема алфавита. *Алфавитом* называется непустое конечное множество всех символов, которые встречаются в выборке. Секвенциальный анализ обычно работает с последовательностями, элементы которых закодированы. Кодирование происходит на этапе предобработки как часть приведения к необходимому формату. При этом даже схожие по смыслу элементы (например, текстовые — слова-синонимы человеческой речи, названия похожих системных функций и тому подобное) получают различные коды. Объем алфавита в этом случае приближается к количеству элементов в выборке. Это делает затруднительным анализ алгоритмами, не учитывающими нечеткий вывод и не умеющими работать с приближениями. Вместо того чтобы создавать новые алгоритмы и повышать их вычислительную сложность, можно на этапе предобработки выделить эквивалентные (близкие) элементы и придать им один и тот же код.

В частности, при анализе числовых последовательностей, не прошедших предобработку, примерно равные числа кодируются как различные элементы. Рассмотрим пример с ежемесячной статистикой, собранной в течение нескольких лет. В табл. 1 представлена статистика по безработице в Одесской области за 2012–2015 годы.

Таблица 1. Выборка S статистики по безработице

Год	Месяцы											
	01	02	03	04	05	06	07	08	09	10	11	12
2012	19404	20278	18324	15800	13938	12585	12132	12097	12013	12318	15893	18545
2013	20360	21349	19314	16800	14284	14284	11034	10225	9758	9025	11338	14241
2014	16449	16963	14535	12401	11418	10475	10376	10274	10395	11158	14026	16998
2015	17790	18082	16965	14761	13504	12113	11246	10526	10472	10627	13296	16267

Легко заметить, что наблюдается закономерность: рост безработицы в зимние месяцы и падение — в летние. Однако так как все числовые значения являются разными, то кодирование этих значений как отдельных элементов приведет к тому, что секвенциальный анализ не выделит ни одного паттерна. Закодируем теперь эти данные в алфавите длины 3. Возьмем все значения алфавита выборки S и разделим их на три группы. Результирующая выборка S' представлена в табл. 2. Проанализируем S' алгоритмами Bide+ и Clospan. Эти алгоритмы позволяют получить закрытые паттерны, но обладают различными принципами поиска. Результаты секвенциального анализа представлены в табл. 3, где графа «поддержка» означает количество последовательностей в исходной выборке, содержащих соответствующие паттерны.

Таблица 2. Выборка S' статистики по безработице

Год	Месяцы											
	01	02	03	04	05	06	07	08	09	10	11	12
2012	3	3	3	3	2	2	1	1	1	1	2	2
2013	3	3	3	2	2	1	1	1	1	1	2	2
2014	3	3	3	2	2	1	1	1	1	2	2	3
2015	3	3	3	2	2	2	1	1	1	1	2	3

Таблица 3. Паттерны, полученные при секвенциальном анализе выборки S'

Поддержка	Паттерны	
	Bide+	CloSpan
2	3 3 3 2 2 1 1 1 1 2 2 3 3 3 2 2 1 1 1 1 2 3 3 3 3 3 2 2 1 1 1 1 3 3 3 2 2 2 2 3	3 3 3 2 2 1 1 1 1 2 2 3 3 3 2 2 1 1 1 1 2 3 3 3 3 3 2 2 1 1 1 1 3 3 3 2 2 2 2 3
3	3 3 3 2 2 1 1 1 1 2 3 3 3 2 2 1 1 1 1 3 3 3 3 2 2 2 2	3 3 3 2 2 1 1 1 1 2 3 3 3 2 2 1 1 1 1 3 3 3 3 2 2 2 2
4	3 3 3 2 2 1 1 1 1 3 3 3 3	3 3 3 2 2 1 1 1 1 3 3 3 3

Полученные паттерны показывают, в частности, что во все годы безработица имела тенденцию к спаду к середине года, а вот возрастала ближе к концу года уже по-разному.

В данном примере объем алфавита был задан вручную. Для данных, требующих более тщательной статистической обработки, можно использовать разбиение на группы (или интервалы), например, с помощью эмпирического правила Стерджесса или других формул, описанных в [13]. Они позволяют определить число групп, оптимальное для текущей выборки. После этого можно кодировать значения каждой группы одним символом алфавита. Объем алфавита, таким образом, будет равен числу групп.

В ситуации, когда обработке подлежат не статистические данные, а, к примеру, последовательности действий, для уменьшения объема алфавита необходимо учитывать смысл действий. Действия, близкие по смыслу, могут быть сгруппированы и обозначены одним символом алфавита. Это может осуществить эксперт в предметной области, исходя из поставленной задачи. К примеру, действия «покупка тетради» и «покупка карандаша» могут быть объединены как «покупка канцтоваров», если анализ проводится для определения сезонного интереса потребителей к канцтоварам, а в наличии только подробная база покупок.

Рассмотрим теперь влияние изменения объема алфавита на эффективность работы методов секвенциального анализа. Использованы те же алгоритмы, что и в предыдущем эксперименте. В качестве исходной выборки применена сгенерированная выборка объемом 80 последовательностей длиной в 80 элементов, где в

каждую последовательность была заложена одна и та же подпоследовательность. Результаты данного эксперимента показаны в табл. 4. Можно сделать вывод, что затраты памяти больше на малых объемах алфавита.

Таблица 4. Уменьшение затрат памяти при росте объема алфавита в обучающей выборке

Объем алфавита	Алгоритм		
	ClaSP	Bide+	CloSpan
20	27	193	32
80	12	49	19
140	9	98	16
200	11	51	17
260	10	50	15

Все вышесказанное свидетельствует о том, что сокращение объема алфавита является важным этапом предобработки для тех выборок, где имеется много различных элементов, часть из которых схожи по смыслу. Данный способ предобработки позволяет увеличить результативность анализа, однако, как показывают эксперименты, в ущерб эффективности. Затраты памяти увеличиваются по мере уменьшения объема алфавита. Очевидно, это связано с большей результативностью методов анализа и соответственно с увеличением ресурсов для его выполнения.

Обработка циклически повторяющихся подпоследовательностей

В выборках может встречаться ситуация, когда частые подпоследовательности повторяются многократно подряд в пределах одной последовательности. Такое явление называется *тандемным повтором* [14]. По смыслу тандемные повторы обозначают циклические события в выборке, поэтому далее будем называть их в контексте секвенциального анализа просто *циклами*. Обработка циклов затратна для алгоритмов секвенциального анализа по памяти и особенно по времени. Кроме того, как будет показано ниже, присутствие циклов в последовательностях зачатую приводит к неверным результатам при проведении секвенциального анализа существующими методами.

Продемонстрируем на экспериментах. Пусть дана выборка объемом 200 последовательностей, длина каждой последовательности — 100 элементов, объем алфавита — 20. В каждую последовательность заложена одна и та же подпоследовательность, повторяющаяся заданное количество раз. Параметр «количество циклов» означает количество идущих подряд повторов этой подпоследовательности. Для эксперимента была использована программная система SPMF, содержащая реализации ряда популярных алгоритмов секвенциального анализа [12]. В табл. 5 показаны затраты памяти и времени для разного количества циклов и разных алгоритмов. Наблюдается значительный рост затрат по мере увеличения количества циклов.

Рассмотрим шире проблемы, с которыми столкнулись опробованные алгоритмы при обработке циклов. Алгоритм PrefixSpan не смог определить, где начался и закончился цикл, поэтому перечислил в качестве паттернов множество

вариантов подпоследовательности-цикла. Объем результатов при этом возрос от 344 при единократном появлении цикла в выборке — до 38513 в том случае, когда цикл повторился 4 раза подряд. Аналогичные результаты показал алгоритм SPADE. Алгоритмы анализа закрытых паттернов Bide+ и CloSpan выдали при любом количестве циклов единственный паттерн в качестве результата. Этот паттерн содержал все повторы заложенной в выборку подпоследовательности-цикла.

Таблица 5. Зависимости времени и объема памяти от количества циклов

Количество циклов	Алгоритм							
	Bide+		SPADE		ClaSP		PrefixSpan	
	время, мс	память, мб	время, мс	память, мб	время, мс	память, мб	время, мс	память, мб
1	1467	50	282	22,5	687	31	344	27
2	6727	193	2889	52	5264	242	1777	98
3	14709	194	81281	62	52014	836	21962	494
4	38513	195	1272619	212	687	31	38513	527

Результаты эксперимента позволяют сделать вывод, что циклы значительно усложняют анализ выборки, вплоть до невозможности анализа, а также приносят избыточные результаты. Следовательно, сокращение объемов циклов является необходимым этапом предобработки для тех данных, в которых они встречаются.

Предлагается на этапе предобработки производить сворачивание тандемных повторов методом, предложенным в [14]. Этот метод позволяет выявить тандемные повторы и сократить их до одного вхождения повторяющейся подпоследовательности. Выявленные в результате повторы можно также сохранить в качестве закономерных подпоследовательностей, могущих дополнить результаты секвенциального анализа.

Помимо снижения затрат на обработку самого цикла, при использованни данного метода снижаются длины последовательностей, что также приводит к дополнительной экономии по времени и памяти. В табл. 6 приведены результаты экспериментов над выборкой, в которой были свернуты циклы. Использована выборка из предыдущего эксперимента.

Таблица 6. Результаты анализа выборки, подвергшейся сворачиванию циклов

Показатель	Алгоритм			
	Bide+	SPADE	ClaSP	PrefixSpan
Память, мб	49	22	29	25
Время, мс	1633	253	687	295

Как видно из полученных результатов, обработка выборки со свернутыми циклами является значительно менее затратной, чем обработка выборки с несколькими повторами циклической подпоследовательности.

Выводы

Рассмотрено понятие предобработки данных в интеллектуальном анализе и раскрыты особенности предобработки данных для проведения секвенциального анализа.

Экспериментально показано, как различные случаи необработанных данных негативно влияют на эффективность работы методов анализа и их результат. Предложена методика предобработки для решения этих проблем, содержащая три этапа: уменьшение зашумленности выборки, сокращение объема алфавита и обработку циклов. Благодаря сокращению объема алфавита удалось получить ненулевые результаты из выборки с объемом алфавита, равным количеству элементов.

Впервые была освещена проблема циклических повторов частых подпоследовательностей в исходной выборке секвенциального анализа. Предложен способ решения: применение сворачивания тандемных повторов. Это позволило получить следующие показатели роста эффективности: для алгоритма Bide+ на 75 % по памяти и 96 % по времени; для алгоритма ClaSP на 96 % по памяти и 98,7 % по времени; для алгоритма SPADE на 64,5 % по памяти и 99,7 % по времени; для алгоритма PrefixSpan на 95 % по памяти и 98,6 % по времени. Таким образом, удалось показать, что сворачивание тандемных повторов способно как снизить затратность анализа, так и существенно улучшить качество результирующей выборки, т.е. полученных при секвенциальном анализе паттернов.

1. Арсеньев С. Извлечение знаний из медицинских баз данных. 1999. URL: http://neural.narod.ru/Arsen.htm

2. Кунгурцев А.Б. Формирование представления данных распределенных информационных систем в терминах предметной области. *Нові технології*. 2003. № 2(3). С. 74–77.

3. Афанасьева С.М. Применение компьютерных технологий для автоматизации анализа медицинской информации. *ВНМТ*. 2005. № 3–4. С. 104–106.

4. Любицын В.Н. Повышение качества данных в контексте современных аналитических технологий. *Вестник Южно-Уральского государственного университета*. *Серия: Компьютерные технологии, управление, радиоэлектроника*. 2012. № 23.

5. Давыдов А.А. Knowledge Discovery and Data Mining в системной социологии. Москва: ИС РАН, 2009. URL: http://www.isras.ru/Davydov_Knowledge.html

6. Markov Z., Larose D.T. Data-mining the Web: uncovering patterns in Web content, structure, and usage. John Wiley & Sons Inc., 2007. 218 p.

7. Луньков А.Д., Харламов А.В. Интеллектуальный анализ данных: учебно-методическое пособие данных. Саратовский государственный университет им. Н.Г. Чернышевского. URL: http://elibrary.sgu.ru/uch_lit/1141.pdf

8. Guyon I., Elisseeff A. An introduction to feature extraction. *Feature extraction*. Springer Berlin Heidelberg, 2006. С. 1–25.

9. Дюк В.А., Жвалевский О.В., Рудницкий С.Б., Толстоногов Д.А. Предварительные результаты обработки разнотипных биометрических данных методами data mining. Труды СПИИРАН. 2009. Вып. 9. С. 197–210.

10. Agrawal R. and Srikant R. Mining Sequential. *Journal Intelligent Systems*. 1997. Vol. 9. N 1. P. 33–56.

11. Молдавская А.В. Метод формирования многоуровневых последовательных паттернов. *Проблеми програмування*. 2016. № 2/3. С. 158–163.

12. Fournier-Viger P., Lin C.W., Gomariz A., Gueniche T., Soltani A., Deng Z., Lam H.T. The SPMF Open-Source Data Mining Library Version 2. Proc. 19-th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2016). Part III. Springer LNCS 9853, 2016. P. 36–40.

13. Орлов Ю.Н. Оптимальное разбиение гистограммы для оценивания выборочной плотности функции распределения нестационарного временного ряда. Москва: Институт прикладной математики им. М.В. Келдыша, 2013. № 14. 26 с. (Препринт. Ин-т прикладной математики им. М.В. Келдыша).

14. Смит Б. Методы и алгоритмы вычислений на строках. Издательский дом Вильямс, 2006. 496 с.

Поступила в редакцию 13.03.2017