НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

НАУЧНЫЙ ДОКЛАД

по результатам подготовленной
научно-квалификационной работы (диссертации)

ФИО: Егурнов Дмитрий Алексеевич

Направление подготовки: 09.06.01 Информатика и вычислительная техника

Профиль (направленность) программы: 05.13.01 Системный анализ, управление и обработка информации

Аспирантская школа по компьютерным наукам

Аспирант _____/Егурнов Д.А. /
подпись

Научный руководитель _____/Игнатов Д.И. /
подпись

Директор Аспирантской школы по компьютерным наукам _____ /Объедков С.А./
подпись

Москва, 2020

Оглавление

Введение

Развитие технологий автоматического сбора и накопления информации привело к генерации большого объёма разнообразных данных, который постоянно увеличивается. Зачастую эти данные содержат скрытые структуры, позволяющие лучше объяснить их природу. Современная прикладная математика и информатика уделяет много внимания проблеме извлечения таких структур и их интерпретации. Кластерный анализ предоставляет широкий набор методов для автоматической классификация объектов и определения их взаимосвязей.

Извлечение мультимодальных структур из многомерных отношений является актуальной задачей Анализа Данных и Машинного Обучения. Естественным расширением идей обычной кластеризации в этом случае будет кластерный анализ мультимодальных данных, в частности бинарных и тернарных (триадических) отношений. Когда входные данные представлены в виде бинарного отношения $I \subseteq G \times M$, без ограничения общности принято называть элементы множества G объектами, а элементы множества M - признаками. Тогда методы бикластеризации используются для одновременного поиска подмножеств объектов и признаков, формирующих однородные структуры во исходном отношении. Термин «бикластеризация» был впервые введён в [Mirkin'96], однако похожий подход предлагался ранее [Hartigan'72]. В триадическом случае методы трикластеризации аналогичным образом работают с данными, к которым добавляется множество условий B.

Приложения методов би- и трикластеризации лежат в таких областях как анализ генной экспрессии [Cheng'00] [Madeira'04] [Zhao'05] [Li'09] [Kaytoue'11] [Eren'12] [Fazendeiro'15], рекомендательные системы [Nanopoulos'10] [Jelassi'13] [Ignatov'14], анализ социальных сетей [Gnatyshak'12] [Ignatov'17], обработка естественных языков [Ustalov'18] и других.

В данной работе рассматриваются методы мультимодальной кластеризации, основанные на Анализе Формальных Понятий (Formal Concept Analysis - FCA)

[Ganter&Wille'99]. Триадический подход рассмотрен в [Lehmann&Wille'95], а обобщения до n-мерного случая предлагались несколькими авторами [Voutsadakis'02] [Cerf'09] [Nataraj'10] [Trabelsi'12]. Аналогичные методы осуществляющее приближенное вычисление понятий предложены в [Cerf'13] [Ignatov'15]. Разложение трехмерных булевых тензоров с применением триадических формальных концептов рассматривается в [Belohlavek'13]. Возможно обобщение формальных понятий до би- и трикластеров [Jaschke'06] [Ignatov'11]. Также существует несколько основанных на FCA подходов для обработки вещественных данных. Например, в [Besson'06] предлагается искать бикластеры (би-сеты) при некоторых заранее определённых условиях, а в [Kaytoue'14] вещественные бикластеры вычисляются с помощью Триадического Анализа Формальных Понятий (3-FCA).

Цели и задачи исследования

Основным объектом исследования в области Триадического Анализа Формальных Понятий является *триадический формальный контекст* $\mathbb{K} = (G, M, B, I)$. Он традиционно определяется в виде трех множеств G, M и B, называемых соответственно *множествами объектов, признаков и условий,* и некоторого подмножества декартова произведения этих множеств $I \in G \times M \times B$, задающего троичное отношение между элементами этих множеств. Допустим $g \in G, m \in M, b \in B$. Если тройка $(g, m, b) \in I$, то говорят, что «объект g обладает признаком m при условии b».

Вместе множества G, M и B называют *измерениями* формального контекста, или координатными множествами, поскольку их элементы являются своего рода координатами троек отношения I. Однако элементы внутри измерений не обязаны быть упорядочены каким-либо образом, что позволяет менять их порядок, для получения удобного представления. Обычно триадические контексты интерпретируются как трёхмерные булевы матрицы. Перестановки элементов

координатных множеств в таком случае соответствуют перестановкам строк, столбцов и слоёв булевой матрицы.

Целью Триадического Анализа Формальных Понятий является поиск в триадических контекстах структур, называемых *триадическими формальными понятиями.* Простейшим определением таких структур будет тройка множеств $T = (X, Y, Z)$, где $X \in G, Y \in M, Z \in B$, такая что $X \times Y \times Z \subseteq I$. В матричном представлении это соответствует кубоиду из положительных значений (единиц или крестов) булевой трёхмерной матрицы, с учётом возможных перестановок строк, столбцов и слоёв. Элементы тройки $T = (X, Y, Z)$ называются соответственно *(формальными) объёмом, содержанием и модусом* данного формального понятия.

Множество всех триадических формальных понятий некоторого фиксированного триадического формального контекста $\mathbb{K}$ образует структуру $\mathfrak{T}(\mathbb{K})$, называемую *решёткой триадических формальных понятий.* В отличие от диадического случая, объёмы, содержания и модусы триадических формальных понятий не составляют *систему замыканий,* так как два разных триадических формальных понятия могут иметь одинаковый объём, но их содержания и модусы могут быть несравнимы по вложению множеств. Этот факт является основным препятствием при попытке расширить методы Диадического Анализа Формальных Понятий на триадический случай.

При дальнейшем развитии идеи Триадического Анализа Формальных Понятий допускается некоторое ослабление термина формального понятия, так как требование наличия полного кубоида положительных булевых значений (единиц или крестов) хорошо работает только в случае точных контекстов, но для других случаев может быть слишком строгим. Большие контексты, особенно содержащие данные, собранные из реальных источников, склонны иметь ошибки обработки и пропущенные значения. В таком случае целесообразно опустить условие полноты формального понятия и осуществлять поиск структур, называемых *триадическими*

кластерами (или *трикластерами*). Соответствующие методы называются *методами Трикластеризации.* Обычно определение трикластеров используемое в каждом отдельном случае зависит от конкретной задачи и набора данных, но в наиболее общем виде трикластер определяется просто как тройка множеств $T = (X, Y, Z)$, где $X \in G, Y \in M, Z \in B$.

При работе с вещественными данными используется *вещественный триадический формальный контекст* $\mathbb{K}_V = (G, M, B, I, V)$, аналогичный описанному выше триадическому формальному контексту, но содержащий *функцию значений* $V: I \to \mathbb{R}$, сопоставляющей тройкам из I значения из $\mathbb{R}$. При этом если тройка $(g, m, b) \in I$ и $V(g, m, b) = v$, значит «объект g обладает признаком m со значением v при условии b». В таком случае *трикластером близких значений* будет называться такой трикластер $T = (X, Y, Z)$, в котором значения троек $(g, m, b) \in I \cap X \times Y \times Z$ находятся в некотором небольшом интервале.

В этом контексте целью данной работы является построение эффективного способа поиска трикластеров и трикластеров близких значений в триадических данных. Задача исследования заключается в описании и обосновании применимости предложенного метода к реальным задачам из различных научных областей, а также его сравнение с современными методами решения таких задач, использующимися в соответствующих областях.

Теоретические результаты

За основу был взят метод OAC-трикластеризации, предложенный в [Гнатышак'12], который в свою очередь является триадическим расширением подхода использующегося в [Ignatov'12].

Он заключается в последовательном применении специальных операторов к каждой тройке формального контекста $(g, m, b) \in I$, называемой *генерирующей тройкой* или *генератором*, получении промежуточных множеств объектов,

признаков и условий с помощью различных наборов *триадических операторов*, построении из них трикластеров и добавлении полученных трикластеров в результирующий набор.

Рассмотрим два вида триадических операторов: бокс-операторы и штрих-операторы.

Бокс-операторы

Пусть дан триадический контекст $\mathbb{K} = (G, M, B, I)$ и его фиксированная тройка $(\tilde{g}, \tilde{m}, \tilde{b}) \in I$ (генератор). Нам понадобятся так называемые *штрих операторы от одного множества* $(X \subseteq G, Y \subseteq M, Z \subseteq B)$*:*

$$X' = \{ (m, b) \in M \times B \mid \forall\, \tilde{g}\ in\ X : (\tilde{g}, m, b) \in I \}$$

$$Y' = \{ (g, b) \in G \times B \mid \forall\, \tilde{m} \in Y : (g, \tilde{m}, b) \in I \}$$

$$Z' = \{ (g, m) \in G \times M \mid \forall\, \tilde{b} \in Z : (g, m, \tilde{b}) \in I \}$$

Можно опустить нотацию множеств для операторов в целях упрощения и писать $a^{\square}$ и a' вместо $\{a\}^{\square}$ и $\{a\}'$, но стоит помнить, что в общем случае операторы применяются к множествам. С их помощью задаются бокс-операторы в общем виде:

$$(\tilde{m}, \tilde{b})^{\square} \coloneqq \{ g \mid \exists\, m : (g, m) \in \tilde{b}' \lozenge \exists\, b : (g, b) \in \tilde{m}' \}$$

$$(\tilde{g}, \tilde{b})^{\square} \coloneqq \{ m \mid \exists\, g : (g, m) \in \tilde{b}' \lozenge \exists\, b : (m, b) \in \tilde{g}' \}$$

$$(\tilde{g}, \tilde{m})^{\square} \coloneqq \{ b \mid \exists\, g : (g, b) \in \tilde{m}' \lozenge \exists\, m : (m, b) \in \tilde{g}' \}$$

Здесь символ $\lozenge$ обозначает вариативность между логическими операциями $\wedge$ и $\vee$. Таким образом определяются 2 вида бокс-операторов, получаемые подстановкой вариативных символов. Будем называть их $\vee$-бокс, $\wedge$-бокс операторами. Рассмотрим более подробно их структуру:

1. $\vee$-бокс оператор предлагает наименее строгое условие включения элемента в промежуточное множество. Для это необходимо, чтобы соответствующий

ему срез формального контекста содержал хотя бы одну тройку в серой зоне (см. **Ошибка! Источник ссылки не найден.**). Так как начальное определение затрудняет анализ, можно его упростить, исключив штрих-оператор.

$$(\tilde{m}, \tilde{b})^{\square}_{\vee} \coloneqq \{\, g \mid \exists\, m : \left(g, m, \tilde{b}\right) \in I \vee \exists\, b : (g, \tilde{m}, b) \in I\}$$

$$(\tilde{g}, \tilde{b})^{\square}_{\vee} \coloneqq \{\, m \mid \exists\, g : \left(g, m, \tilde{b}\right) \in I \vee \exists\, b : (\tilde{g}, m, b) \in I\}$$

$$(\tilde{g}, \tilde{m})^{\square}_{\vee} \coloneqq \{\, b \mid \exists\, g : (g, \tilde{m}, b) \in I \vee \exists\, m : (\tilde{g}, m, b) \in I\}$$

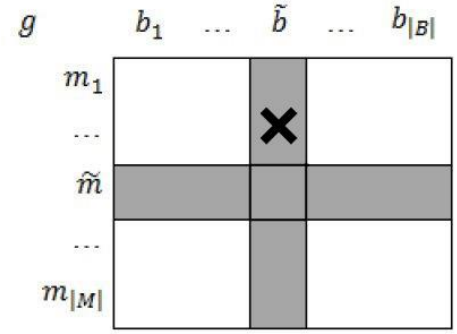

Рисунок 1. Условие ∨-бокс оператора

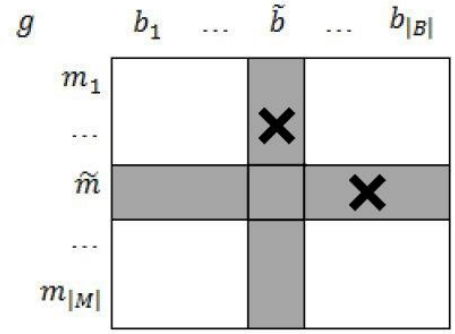

Рисунок 2. Условие ∧-бокс оператора

2. ∧-бокс оператор налагает более строгое условие на включение элемента в промежуточное множество. Соответствующий срез должен содержать тройки в обеих линиях серой зоны (см. Рисунок 2). Упрощенное определение:

$$(\tilde{m}, \tilde{b})^{\square}_{\wedge} \coloneqq \{\, g \mid \exists\, m : \left(g, m, \tilde{b}\right) \in I \wedge \exists\, b : (g, \tilde{m}, b) \in I\}$$

$$(\tilde{g}, \tilde{b})^{\square}_{\wedge} \coloneqq \{\, m \mid \exists\, g : \left(g, m, \tilde{b}\right) \in I \wedge \exists\, b : (\tilde{g}, m, b) \in I\}$$

$$(\tilde{g}, \tilde{m})^{\square}_{\wedge} \coloneqq \{\, b \mid \exists\, g : (g, \tilde{m}, b) \in I \wedge \exists\, m : (\tilde{g}, m, b) \in I\}$$

Тройка множеств $T^{\square}_{\Diamond} = \left(\left(\tilde{m}, \tilde{b}\right)^{\square}_{\Diamond}, \left(\tilde{g}, \tilde{b}\right)^{\square}_{\Diamond}, (\tilde{g}, \tilde{m})^{\square}_{\Diamond}\right)$ называется *трикластером построенным на бокс операторах.*

Штрих-операторы

Метод OAC-трикластеризации, основанный на штрих-операторах [Gnatyshak'13], использует операторы, похожие на указанные выше, но применяемые к паре множеств:

$$(X,Y)' = \{ b \in B | \ (\tilde{g},\tilde{m},b) \in I \ \forall \ \tilde{g} \in X, \tilde{m} \in Y\}$$

$$(X,Z)' = \{ m \in M | \ (\tilde{g},m,\tilde{b}) \in I \ \forall \ \tilde{g} \in X, \tilde{b} \in Z\}$$

$$(Y,Z)' = \{ g \in G | \ (g,\tilde{m},\tilde{b}) \in I \ \ \forall \ \tilde{m} \in Y, \tilde{b} \in Z\}$$

Трикластером, построенным на штрих операторах для некоторого генератора $(g,m,b,) \in I$, будет называться тройка множеств $T' = ((m,b)', (g,b)', (g,m)')$. Здесь указана упрощенная нотация для одноэлементных множеств.

Отношения промежуточных множеств и трикластеров

Рассмотрим, какие отношения связывают промежуточные множества и трикластеры, полученные с помощью различных операторов. Для краткости приведены выкладки только по одной модальности.

Лемма 1. Промежуточное множество, сгенерированное ∨-бокс оператором, содержит в себе промежуточное множество, сгенерированное ∧-бокс оператором от того же генератора, но обратное не всегда верно.

Доказательство: Каждый элемент, удовлетворяющий условию оператора ∧-бокс, также удовлетворяет условию оператора ∨-бокс

$$g \in (\tilde{m},\tilde{b})^{\square}_{\wedge} \Longrightarrow g \in (\tilde{m},\tilde{b})^{\square}_{\vee}$$

Могут быть элементы, удовлетворяющие только условию оператора ∨-бокс.

$$\text{для некоторого } g \ \exists m \neq \tilde{m} : (g,m,\tilde{b}) \in I \text{ и } \forall b \in B : (g,\tilde{m},b) \notin I$$

$$\Longrightarrow g \in (\tilde{m},\tilde{b})^{\square}_{\vee}, g \notin (\tilde{m},\tilde{b})^{\square}_{\wedge}$$

Следовательно операторы не эквивалентны. ■

Лемма 2. Промежуточное множество, сгенерированное Λ-бокс оператором, содержит в себе промежуточное множество, сгенерированное штрих-оператором от того же генератора, но обратное не всегда верно.

Доказательство: воспользуемся определениями операторов.

$$g \in (\tilde{m}, \tilde{b})' \Longrightarrow (g, \tilde{m}, \tilde{b}) \in I \Longrightarrow$$

$$\Longrightarrow \exists m = \tilde{m} : (g, m, \tilde{b}) \in I \; and \; \exists b = \tilde{b} : (g, \tilde{m}, b) \in I \Longrightarrow$$

$$\Longrightarrow g \in (\tilde{m}, \tilde{b})^{\square}_{\wedge}$$

В обратную сторону можно доказать, что может существовать элемент, входящий в промежуточное множество только Λ-бокс оператора:

$$\exists m \neq \tilde{m} : (g, m, \tilde{b}) \in I \; and \; \exists b \neq \tilde{b} : (g, \tilde{m}, b) \in I$$

$$and \; (g, \tilde{m}, \tilde{b}) \notin I \Longrightarrow \; g \in (\tilde{m}, \tilde{b})^{\square}_{\wedge}, g \notin (\tilde{m}, \tilde{b})'$$

Следовательно операторы не эквиваленты. ■

Теорема 1. О порядке вложенности промежуточных множеств. Промежуточные множества, сгенерированные бокс- и штрих-операторами, от одного генератора, упорядочены следующим образом:

$$(\tilde{m}, \tilde{b})' \subseteq (\tilde{m}, \tilde{b})^{\square}_{\wedge} \subseteq (\tilde{m}, \tilde{b})^{\square}_{\vee}$$

Доказательство следуем из Лемм 1 и 2. ■

Следствие 1: Трикластеры, построенные из промежуточных множеств, сгенерированных от одного генератора упорядочены следующим образом в смысле покомпонентной вложенности:

$$T' \sqsubseteq T^{\square}_{\wedge} \sqsubseteq T^{\square}_{\vee}$$

Вложенность триадических понятий

Методы ОАС-трикластеризации могут быть полезны и при поиске триадических понятий. Они не могут найти сами понятия, но в этом пункте будет доказано, что в

трикластеры покрывают все триадические понятия исходного формального контекста.

Теорема 2. Для каждого триадического формального понятия $T = (X, Y, Z)$ данного формального контекста существует трикластер T' такой что формальное понятие T вкладывается в него поэлементно.

Доказательство: выберем генератор $(\tilde{g}, \tilde{m}, \tilde{b})$, входящий в формальное понятие T. ($\tilde{g} \in X, \tilde{m} \in Y, \tilde{b} \in Z$). Теперь докажем, что любой элемент X, будет входить в промежуточное множество $(\tilde{m}, \tilde{b})'$.

$$\forall x \in X, \forall y \in Y, \forall z \in Z : (x, y, z) \in I \Longrightarrow$$

$$\Longrightarrow (x, \tilde{m}, \tilde{b}) \in I \Longrightarrow x \in (\tilde{m}, \tilde{b})'$$

Аналогично $\forall y \in Y, y \in (\tilde{g}, \tilde{b})'$ и $\forall z \in Z, z \in (\tilde{g}, \tilde{m})'$. Следовательно, триадическое формальное понятие T поэлементно вкладывается в трикластер $T' = ((m, b)', (g, b)', (g, m)')$, построенный от генератора $(\tilde{g}, \tilde{m}, \tilde{b})$. ■

Кластеризация вещественных данных

В предыдущих пунктах доказано, что наиболее компактные трикластеры порождаются методом, использующим штрих-операторы. На их основе разработаны дельта-операторы для поиска трикластеров близких значений с параметром δ в вещественных триадических формальных контекстах.

$$(\tilde{g}, \tilde{m}, \tilde{b})_g^{\delta} = \{ g \mid (g, \tilde{m}, \tilde{b}) \in I \wedge \left| V(g, \tilde{m}, \tilde{b}) - V(\tilde{g}, \tilde{m}, \tilde{b}) \right| < \delta \}$$

$$(\tilde{g}, \tilde{m}, \tilde{b})_m^{\delta} = \{ m \mid (\tilde{g}, m, \tilde{b}) \in I \wedge \left| V(\tilde{g}, m, \tilde{b}) - V(\tilde{g}, \tilde{m}, \tilde{b}) \right| < \delta \}$$

$$(\tilde{g}, \tilde{m}, \tilde{b})_b^{\delta} = \{ b \mid (\tilde{g}, \tilde{m}, b) \in I \wedge \left| V(\tilde{g}, \tilde{m}, b) - V(\tilde{g}, \tilde{m}, \tilde{b}) \right| < \delta \}$$

В этом случае трикластер близких значений, построенный от генератора $(\tilde{g}, \tilde{m}, \tilde{b})$ выглядит следующим образом $T^{\delta} = \left((\tilde{g}, \tilde{m}, \tilde{b})_g^{\delta}, (\tilde{g}, \tilde{m}, \tilde{b})_m^{\delta}, (\tilde{g}, \tilde{m}, \tilde{b})_b^{\delta} \right)$

Альтернативой этому подходу может служить применение классического алгоритма K-средних с адаптированной мерой расстояния между триплетами $t_1 = (g_1, m_1, b_1) \in I$ и $t_2 = (g_2, m_2, b_2) \in I$:

$$\rho(t_1, t_2) = |V(t_1) - V(t_2)| + \gamma * ([g_1 \neq g_2] + [m_1 \neq m_2] + [b_1 \neq b_2])$$

где γ является неким вещественным параметром, определяющим приоритетность близости значений внутри трикластера относительно расширения его по измерениям. Выражение $[a_1 \neq a_2]$ принимает значение 0, если координаты триплета в соответствующем измерении эквивалентны, и 1 иначе.

В результате работы алгоритма мы получаем k обычных кластеров, состоящих из триплетов. Для сравнения с NOAC нам требуется перевести их в трикластеры. Пусть $H \subseteq I$ – множество триплетов, входящих в кластер. Тогда трикластером близких значений будем называть тройку множеств $T = (\{g|\exists(g, m, b) \in H\}, \{m|\exists(g, m, b) \in H\}, \{b|\exists(g, m, b) \in H\})$. Таким образом, в соответствующие измерения получившегося трикластера войдут все значения, содержащиеся в триплетах исходного кластера.

Алгоритмы

На основании предложенных методов мультимодальной кластеризации вещественных данных были построены два алгоритма:

Numerical OAC

NOAC (Numerical OAC) реализует метод вещественной OAC-трикластеризации с помощью дельта-операторов. Помимо данных формального контекста, он принимает на вход параметр дельта. Перебор всех триплетов в худшем случае займёт $O(|G||M||B|)$. Построение трикластера производится за $O(|G| + |M| + |B|)$. Следовательно, сложность алгоритма можно оценить как $O(|G||M||B| \cdot \max(|G|, |M|, |B|))$. Так как применение дельта-операторов зависит от конкретного генератора, то предподсчитать результаты их вычисления невозможно.

Алгоритм 1. NOAC

Ввод: триадический контекст $\mathbb{K} = (G, M, B, I)$, параметр δ

Вывод: множество трикластеров близких значений $\mathfrak{T}$

1: $\mathfrak{T} \coloneqq \emptyset$

2: **for all** $(g, m, b) \in I$ **do**

3: $T \coloneqq (DeltaOpG(g, m, b), DeltaOpM(g, m, b), DeltaOpB(g, m, b)$

4: $\mathfrak{T}.add(T)$

5: **end for**

Tri-K-Means

Альтернативный алгоритм Tri-K-Means реализует вариацию алгоритма K-Means (K-Medoid) с оригинальной мерой расстояния. Для работы ему требуются два параметра: количество кластеров k и параметр γ, участвующий в мере расстояния.

Вычислительная сложность одной итерации алгоритма K-Means линейно зависит от параметра k и общего количества трикластеров, поэтому может быть оценена как $O(k|G||M||B|)$. Будем считать, что метод сходится за достаточно малое количество итераций, не влияющее на порядок сложности. Тогда время работы всего алгоритма можно оценить $O(k|G||M||B|)$.

Оценка качества трикластеров

Для оценки качества обнаруживаемых трикластеров были использованы показатели плотности и дисперсии.

Пусть $T = (X, Y, Z)$ трикластер в многозначном формальном контексте $\mathbb{K} = (G, M, B, I, V)$. Определим плотность трикластера T как отношение входящих в него триплетов к его геометрическому объёму: $\rho = \frac{|I \cap (X \times Y \times Z)|}{|X||Y||Z|}$. Можно заметить, что такая оценка не учитывает значения входящих в трикластер триплетов. Поэтому будем оценивать трикластеры ещё и по дисперсии значений, содержащихся в нём

триплетов. Пусть $S = \{V(g,m,b)|(g,m,b) \in Q \cap |X \times Y \times Z|\}$ – выборка, состоящая из этих значений. Тогда несмещённая оценка дисперсии будет выглядеть так:

$$\sigma = \frac{\sum_{i=1}^{|S|} S_i^2 - \frac{\left(\sum_{i=1}^{|S|} S_i\right)^2}{|S|}}{|S| - 1}$$

Значения трикластера будем считать близкими при некотором параметре δ, если среднеквадратичное отклонение будет меньше или равно параметру. То есть $\sqrt{\sigma} \leq \delta$.

Программный комплекс

Основной программой для проведения экспериментов в данной работе являлась Triclustering Toolbox. В ней содержатся реализации всех предложенных алгоритмов, интегрированы средства контроля времени их исполнения и вычисления оценок результатов кластеризации.

Изначально она была написана на языке C# на платформе .Net Framework 4 в среде Microsoft Visual Studio 2010 [Гнатышак'12]. Доработка велась на том же языке, но в среде Microsoft Visual Studio 2012.

Все измерения производились на компьютере с процессором Intel Core i5-M430 2.27 ГГц и 3 ГБ оперативной памяти под управлением 64-разрядной версии Microsoft Windows 7 SP1.

В ходе предварительного тестирования было выявлено, что для сравнения результатов работы предложенных методов сообразно брать параметр γ метода Tri-k-means равным параметру δ метода NOAC, используемому в аналогичном случае.

Эксперименты

Синтетические данные

Был сгенерирован набор вещественных триадических формальных контекстов со следующими характеристиками:

- Эталонный контекст, состоящий из двух кубоидов размером 10x10x10 и 5x6x4 со значениями, соответственно, 3 и 7. Всего 1120 триплетов
- Контексты для экспериментов на пропущенные значения с уровнями потерь от 10% до 90% с шагом 10%.
- Размытые контексты с амплитудой размытия 0.1, 0.5, 0.9, 1, 1.5 и 2.

• % потерь	Количество трикластеров Tri-k-means	Обнаружение исходных кубоидов Tri-k-means	Количество трикластеров NOAC	Обнаружение исходных кубоидов NOAC
0	2	+	2	+
10	2	+	603	+
20	2	+	862	-
30	2	+	771	-
40	2	+	664	-
50	2	+	554	-
60	2	+	469	-
70	2	+	329	-
80	2	+	239	-
90	2	+	105	-

Таблица 1. Эксперимент с отсутствием данных

В Таблица 1 находятся результаты испытаний на отсутствие данных. Они демонстрируют прекрасную устойчивость метода Tri-k-means к неполноте информации. Метод NOAC напротив оказался довольно требовательным к

плотности входных контекстов. Уже на уровне потерь 20% исходные кубоиды не были обнаружены методом, хотя многие трикластеры, количество которых было соизмеримо с количеством оставшихся в контексте триплетов, были близки к ним. Вероятно, это обусловлено способом построения трикластеров.

В этих экспериментах параметры Tri-k-means k = 2, γ = 0; параметр NOAC δ = 0.

Амплитуда	Количество трикластеров Tri-k-means	Средняя дисперсия Tri-k-means	Количество трикластеров NOAC	Средняя дисперсия NOAC
0	2	0	2	0
0.1	2	0,0031	2	0,0031
0.5	2	0,0775	2	0,0775
0.9	2	0,2767	562	0,2622
1	2	0,3424	967	0,3221
1.5	2	1,5556	1115	0,7573
2	2	2,9508	77	1,2769

Таблица 2. Эксперименты на шумоустойчивость

Шумоустойчивость методов проверялась на эталонном контексте с различными амплитудами отклонений в обе стороны. В этих экспериментах использовались Tri-k-means с параметрами k = 2, γ = 1; NOAC параметр δ = 1.

Таблица 2 содержит результаты экспериментов на шумоустойчивость. По приведённым оценкам видно, что при малых отклонениях значений оба метода хорошо справляются с поиском трикластеров близких значений. С ростом амплитуды размытия Tri-k-means начинает быстро терять в оценке дисперсии обнаруживаемых трикластеров. Начиная с отметки 1.5, он выдаёт трикластеры, не удовлетворяющие условию близости значений. Метод NOAC сохраняет близость значений трикластеров с увеличением размытия, но помимо искомых, обнаруживает множество схожих трикластеров, в количестве соизмеримом с

общим количеством триплетов. Возможным решением этой проблемы будет фильтрация и объединение похожих трикластеров. Оба метода перестали удовлетворять условию близости значений трикластеров при амплитуде равной 2, т.е. когда диапазоны размытых значений двух эталонных трикластеров стали пересекаться.

Данные проекта GroupLens

Для опытов с реальными данными использовался набор 100k проекта GroupLens, содержащий информацию о 100 000 оцениваниях по пятибальной шкале 1000 пользователями 1700 фильмов в 19 жанрах на сайте MovieLens. В наборе также указано, к каким жанрам относятся фильмы. Каждый фильм мог иметь несколько жанров. В качестве множества объектов были взяты пользователи, множества признаков – фильмы, множества условий – жанры. Триплеты составлялись из идентификатора пользователя, фильма, оценённого пользователем и одного из жанров, приписанных фильму. Общее количество троек в контексте – 212595, плотность контекста - 0,00658. Так как оценка пользователя не зависела от жанра, то все триплеты, соответствующие паре пользователь-фильм, означивались одинаково. Эта особенность получившегося контекста является основным его недостатком.

В силу длительного времени исполнения программ на больших объёмах данных, было решено не обрабатывать более 100000 триплетов за раз. В Таблица 3 и Таблица 4 содержится информация об экспериментах над реальными данными. Количество трикластеров в методе Tri-k-means задавалось вручную и бралось равным количеству трикластеров, обнаруженному NOAC в соответствующем эксперименте. Параметры γ и δ равнялись 1. Метод NOAC использовался с отсечением трикластеров, плотность которых меньше 0.5 или поддержка по объёму и признакам меньше 4, с целью сократить время работы и повысить качество выходных трикластеров.

Количество триплетов	Количество трикластеров	Средняя дисперсия	Средняя плотность
10 000	13	0,4831	0,5269
20 000	47	0,6840	0,5208
30 000	101	0,7947	0,5399
40 000	153	0,7667	0,5436
50 000	259	0,8186	0,5509
60 000	372	0,8240	0,5471
70 000	618	0,8200	0,5434
80 000	864	0,8265	0,5471
90 000	1135	0,8545	0,5508
100 000	1421	0,8672	0,5527

Таблица 3. NOAC на данных GroupLens

Количество триплетов	Количество трикластеров	Средняя дисперсия	Средняя плотность
10 000	13	1,0447	0,0126
20 000	47	0,8264	0,0717
30 000	101	0,8152	0,1057
40 000	153	0,7006	0,1253
50 000	259	0,6286	0,2041
60 000	372	0,5657	0,2300
70 000	618	0,4854	0,3190
80 000	864	0,4339	0,3777
90 000	1135	0,4303	0,3929
100 000	1421	0,4087	0,4422

Таблица 4. Tri-K-Means на данных GroupLens

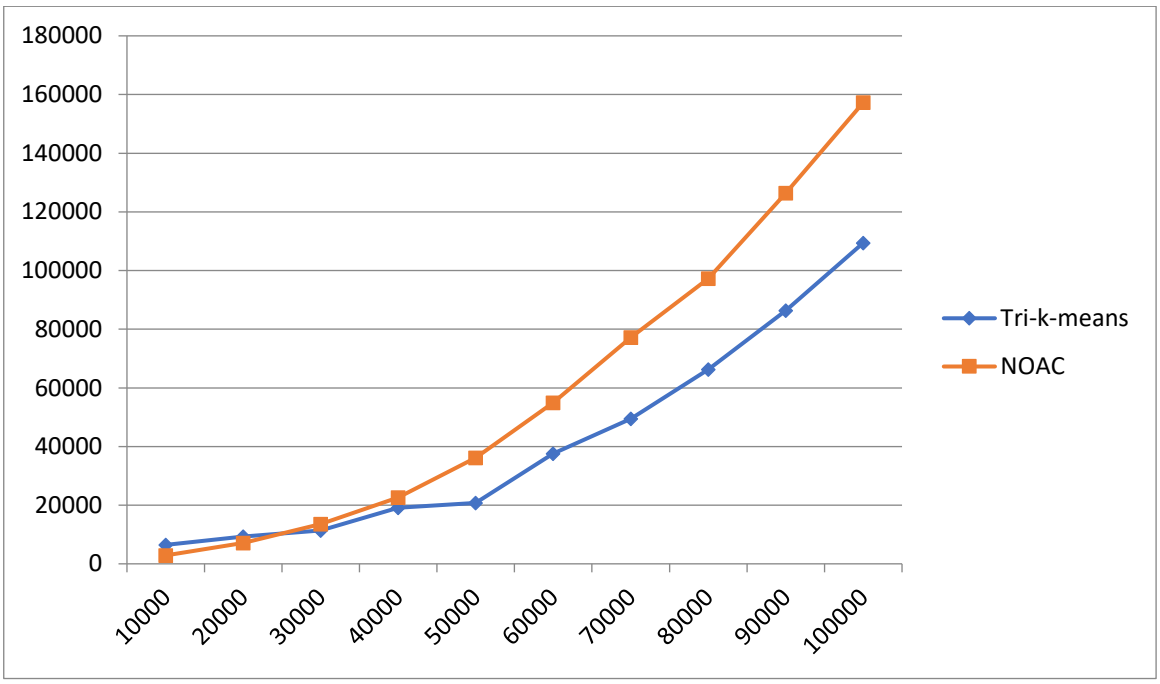

Диаграмма 1. Зависимость времени работы от объема выборки

Диаграмма 1 демонстрирует, что метод Tri-k-means имеет меньший порядок роста при увеличении объёма входных данных, чем NOAC.

Полученные в результате обработки данного реального контекста трикластеры можно интерпретировать, как клубы по интересам, состоящие из пользователей, одинаково оценивших схожие по жанрам фильмы. Отсутствующие значения в таком случае можно применить в рекомендательной системе, предполагая, что новые оценки будут дополнять существующие трикластеры, не сильно отклоняясь от среднего значения.

Эксперименты с параллелизацией

Одним из способов ускорения работы алгоритма является распараллеливание вычислительных потоков. Алгоритмы, обладающие такой возможностью лучше приспособлены к обработке больших объемов данных. Для экспериментов с параллелизацией был использован набор данных задачи извлечения семантических фреймов из [Ustalov'18], состоящий из 100000 троек. Алгоритм NOAC обработал две серии экспериментов с разными параметрами: $(\delta = 100, \rho_{min} = 0.8, minsup = 2)$ и $(\delta = 100, \rho_{min} = 0.5, minsup = 0)$. Измерялось

время исполнения относительно количества входных троек. Диаграмма 2 содержит результаты. Видно, что параллельная версия алгоритма работает в среднем на 35% быстрее.

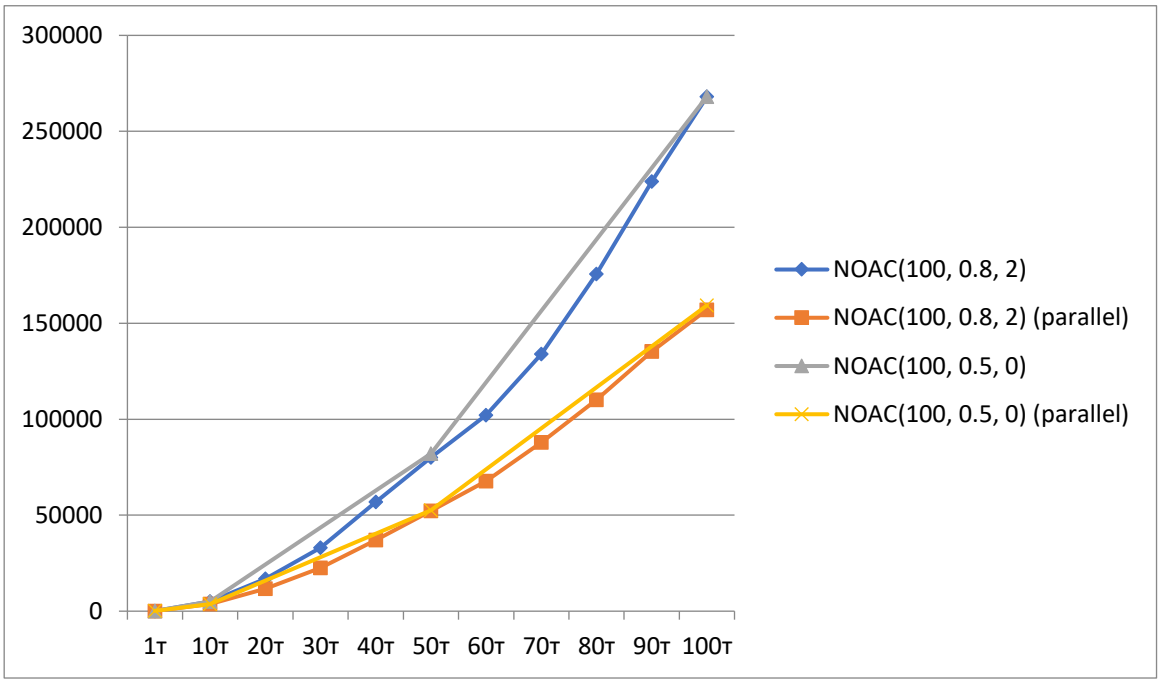

Диаграмма 2. Зависимость времени исполнения, мс от объема входных данных

Заключение

В этой работе рассмотрены методы мультимодальной кластеризации реляционных данных. В теоретической части описаны два вида триадических операторов, используемых для построения трикластеров, описаны их свойства и выведены отношения. На основе одного из них предложен метод для поиска трикластеров близких значений в вещественных триадических формальных контекстах. Практическая часть содержит описание двух алгоритмов, один из которых основан на указанном методе, а другой адаптирует к задаче алгоритм классической кластеризации K-Means. Проведены эксперименты как на искусственных, так и реальных данных. Также изучены возможности параллелизации первого алгоритма.

Публикации

В ходе работы были опубликованы следующие статьи:

1. Egurnov D., Ignatov D. I., MEPHU NGUIFO E. On Containment of Triclusters Collections Generated by Quantified Box Operators, 23rdInternational Symposium on Methodologies for Intelligent Systems “Proceedings - LNAI. Birkhauser/Springer, 2017. P. 573-579.
2. Egurnov D., Ignatov D. I., MEPHU NGUIFO E. Mining Triclusters of Similar Values in Triadic Real-Valued Contexts, in: 14th International Conference on Formal Concept Analysis - Supplementary Proceedings. University Rennes 1,2017. P. 31-47
3. Dmitrii Egurnov, Dmitry I. Ignatov, Triclustering Toolbox in Supplementary Proceedings ICFCA 2019 Conference and Workshops - CEUR Workshop Proceedings, vol. 2378, pp 65-69
4. Dmitry I. Ignatov, Dmitry Tochilkin, Dmitry Egurnov, Multimodal Clustering of Boolean Tensors on MapReduce: Experiments Revisited in Supplementary Proceedings ICFCA 2019 Conference and Workshops - CEUR Workshop Proceedings, vol. 2378, pp 137-151

Список литературы

[Гнатышак'12] Гнатышак Д. В. Сравнительный анализ методов трикластеризации и их приложения. – Москва, 2012

[Belohlavek'13] Belohlavek, R., Glodeanu, C.V., Vychodil, V.: Optimal factorization of three-way binary data using triadic concepts. Order30(2) (2013) 437–454

[Besson'06] Besson, J., Robardet, C., Raedt, L.D., Boulicaut, J.: Mining bi-sets in numerical data. In: Knowledge Discovery in Inductive Databases, 5th International Workshop, KDID 2006, Berlin, Germany, September 18, 2006, Revised Selected and Invited Papers. (2006) 11–23

[Cerf'09] Cerf, L., Besson, J., Robardet, C., Boulicaut, J.: Closed patterns meet n-ary relations. TKDD3(1) (2009) 3:1–3:36

[Cerf'13] Cerf, L., Besson, J., Nguyen, K., Boulicaut, J.: Closed and noise-tolerant patterns in n-ary relations. Data Min. Knowl. Discov.26(3) (2013) 574–619

[Cheng'00] Cheng, Y., Church, G. M., Biclustering of Expression Data. In Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology: 93-103, 2000

[Eren'12] K. Eren, M. Deveci, O. Kucuktunc, and U. V. Catalyurek. A comparative analysis of biclustering algorithms for gene expression data. Briefings in Bioinform., 2012.

[Fazendeiro'15] Fazendeiro, P., Oliveira, A. L., Observer Biased Analysis of Gene Expression Profiles. In Big Data Analytics in Bioinformatics and Healthcare: 117-137, 2015

[Ganter&Wille'99] Ganter B., Wille R., Formal Concept Analysis: Mathematical Foundations. – Springer, 1999

[Gnatyshak'12] D. V. Gnatyshak, D. I. Ignatov, A. V. Semenov, and J. Poelmans. Gaining insight in social networks with biclustering and triclustering. In BIR, volume 128 of Lecture Notes in Business Information Processing, pages 162–171. Springer, 2012.

[Gnatyshak'13] Gnatyshak, D., Ignatov, D.I., Kuznetsov, S.O.: From triadic FCA to triclustering: Experimental comparison of some triclustering algorithms. In: Proceedings of the Tenth International Conference on Concept Lattices and Their Applications, LaRochelle, France. (2013) 249–260

[Hartigan'72] Hartigan, J.A.: Direct Clustering of a Data Matrix. Journal of the American Statistical Association 67(337) (1972) pp. 123-129

[Ignatov'11] Ignatov, D.I., Kuznetsov, S.O., Magizov, R.A., Zhukov, L.E.: From triconcepts to triclusters. In: Rough Sets, Fuzzy Sets, Data Mining and Granular Computing - 13th International Conference, RSFDGrC 2011, Moscow, Russia, June 25-27, 2011. Proceedings. (2011) 257–264

[Ignatov'12] Ignatov, D.I., Kuznetsov, S.O., Poelmans, J.: Concept-based biclustering for internet advertisement. In: 12th IEEE International Conference on Data Mining Workshops, ICDM Workshops, Brussels, Belgium, December 10, 2012. (2012) 123–130

[Ignatov'14] D. I. Ignatov, E. Nenova, N. Konstantinova, and A. V. Konstantinov. Boolean Matrix Factorisation for Collaborative Filtering: An FCA-Based Approach. In AIMSA 2014, Varna, Bulgaria, Proceedings, volume LNCS 8722, pages 47–58, 2014.

[Ignatov'15] Ignatov, D.I., Gnatyshak, D.V., Kuznetsov, S.O., Mirkin, B.G.: Triadic formal concept analysis and triclustering: searching for optimal patterns. Machine Learning 101(1-3) (2015) 271–302

[Ignatov'17] Ignatov, D.I., Semenov, A., Komissarova, D., Gnatyshak, D.V.: Multimodal clustering for community detection. In: Formal Concept Analysis of Social Networks, pp. 59–96. Lecture Notes in Social Networks, Springer (2017).

[Jaschke'06] Jaschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: TRIAS - an algorithm for mining iceberg tri-lattices. In: Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China (2006) 907–911

[Jelassi'13] M. N. Jelassi, S. B. Yahia, and E. Mephu Nguifo. A personalized recommender system based on users' information in folksonomies. In WWW (Companion Volume), pp. 1215–1224. ACM, 2013.

[Kaytoue'11] M. Kaytoue, S. O. Kuznetsov, A. Napoli, and S. Duplessis. Mining gene expression data with pattern structures in formal concept analysis. Inf. Sci., 181(10):1989–2001, 2011.

[Kaytoue'14] Kaytoue, M., Kuznetsov, S.O., Macko, J., Napoli, A.: Biclustering meets triadic concept analysis. Ann. Math. Artif. Intell.70(1-2) (2014) 55–79

[Lehmann&Wille'95] Lehmann F., Wille R., A triadic approach to formal concept analysis. Conceptual Structures: Applications, Implementation and Theory. Springer, 1995. – pp. 32-43

[Li'09] A. Li and D. Tuck. An effective tri-clustering algorithm combining expression data with gene regulation information. Gene regul. and syst. biol., 3:49–64, 2009.

[Madeira'04] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biology Bioinform., 1(1):24–45, 2004.

[Mirkin'96] Mirkin, B.: Mathematical Classification and Clustering. Kluwer, Dordrecht (1996)

[Nanopoulos'10] A. Nanopoulos, D. Rafailidis, P. Symeonidis, and Y. Manolopoulos. Musicbox: Personalized music recommendation based on cubic analysis of social tags. IEEE Transactions on Audio, Speech & Language Processing, 18(2):407–412, 2010.

[Nataraj'10] Nataraj, R.V., Selvan, S.: Closed pattern mining from n-ary relations. International Journal of Computer Applications1(9) (February 2010) 9–13 Published by Foundation of Computer Science.

[Trabelsi'12] Trabelsi, C., Jelassi, N., Yahia, S.B.: Scalable mining of frequent tri-concepts from folksonomies. In: Advances in Knowledge Discovery and Data Mining - 16th Pacific-

Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29 - June1, 2012, Proceedings, Part II. (2012) 231–242

[Ustalov'18] D. Ustalov, A. Panchenko, A. Kutuzov, C. Biemann, and S. P. Ponzetto. Unsupervised semantic frame induction using triclustering. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers, pages 55–62. Association for Computational Linguistics, 2018

[Voutsadakis'02] Voutsadakis G., Polyadic Concept Analysis / Order, T. 19, #3, 2002. – pp. 295-304

[Zhao'05] L. Zhao and M. J. Zaki. Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. In SIGMOD 2005 Conference, pages 694–705, 2005