

Группа G называется σ -примарной, если G является σ_i -группой для некоторого $\sigma_i \in \sigma$; σ -нильпотентной, если $G = G_1 \times \dots \times G_n$ для некоторых σ -примарных групп G_1, \dots, G_n .

Пусть $\Pi \subseteq \sigma$. Тогда группу G называют Π -группой, если $\sigma(G) \subseteq \Pi$.

Пусть \mathfrak{N}_Π – класс всех σ -нильпотентных Π -групп. Группу G назовем Π -скованной, если $C_G(G_{\mathfrak{N}_\Pi}) \leq G_{\mathfrak{N}_\Pi}$.

Задачу существования и сопряженности \mathfrak{H} -инъекторов для σ -класса Хартли в Π -скованных группах решает следующая

Теорема. Пусть \mathfrak{X} – непустой класс Фиттинга, h – H_σ -функция такая, что $h(\sigma_i) = \mathfrak{X}$ для любого $\sigma_i \in \Pi = \text{Supp}(h)$ и G – группа. Если $\mathfrak{H} = LH_\sigma(h)$ и фактор $G/G_{\mathfrak{X}}$ Π -скован, то справедливы следующие утверждения:

1) подгруппа V группы G является \mathfrak{H} -инъектором G в точности тогда, когда $V/G_{\mathfrak{X}} - \mathfrak{N}_\Pi$ -инъектор $G/G_{\mathfrak{X}}$;

2) в G существуют \mathfrak{H} -инъекторы и любые два из них сопряжены.

Заключение. В настоящей работе доказано существование и сопряженность \mathfrak{H} -инъекторов для σ -класса Хартли \mathfrak{H} , определяемого постоянной H_σ -функцией.

1. Doerk, K. Finite Soluble Groups / K. Doerk, T. Hawkes. – Berlin; New York: Walter de Gruyter, 1992. – 891 p.
2. Fischer, B. Injektoren endlicher auflösbarer Gruppen / B. Fischer, W. Gaschütz, B. Hartley // Math. Z. – 1967. – Vol. 102. – P. 337–339.
3. Shemetkov, L.A. Injectors in finite groups / L.A. Shemetkov // Izvestija Gomel'skogo gos. un-ta im. F. Skoriny. Voprosy algebrы. – 2000. – № 3 (16). – P. 186–187.
4. Шеметков, Л.А. О подгруппах π -разрешимых групп / Л.А. Шеметков // в кн.: Конечные группы. – Минск: Наука и техника. – 1975. – С. 207–212.
5. Fischer, B. Klassen konjugierter Untergruppen in endlichen auflösbaren Gruppen / B. Fischer. – Habilitationsschreft, Universität Frankfurt am Mainz, 1966.
6. Hartley, B. On Fischer's dualization of formation theory / B. Hartley // Proc. London Math. Soc. – 1969. – Vol. 3, № 2. – P. 193–207.
7. Blessenohl, D. Fittingklassen endlicher Gruppen in denen gewisse Hauptfaktoren einfach sind / D. Blessenohl, H. Laue // J. Algebra. – 1979. – Vol. 56, № 3. – P. 516–532.
8. Liu, Y.F. Description of \mathfrak{F} -injectors of finite soluble groups / Y.F. Liu, W. Guo, N. T. Vorob'ev // Math. Sci. Res. J. – 2008. – Vol. 12, № 1. – P. 17–22.
9. Guo, W. On injectors of finite soluble groups / W. Guo, N.T. Vorob'ev // Comm. Algebra. – 2008. – Vol. 36. – P. 3200–3208.
10. Yang, N. On \mathfrak{F} -injectors of Fitting set of a finite group / N. Yang, W. Guo, N. T. Vorob'ev // Comm. Algebra. – 2018. – Vol. 46, № 1. – P. 217–229.
11. Skiba, A.N. On σ -properties of Finite groups I / A.N. Skiba // Problems of Physics, Mathematics and Technics. – 2014. – № 4(21). – P. 89–96.
12. Skiba, A.N. On σ -properties of Finite groups II / A.N. Skiba // Problems of Physics, Mathematics and Technics. – 2015. – № 3(24). – P. 70–83.
13. Guo, W. On σ -local Fitting classes / W. Guo, L. Zhang, N.T. Vorob'ev // J. Algebra. – 2020. – Vol. 542, № 15. – P. 116–129.

ВОЗМОЖНОСТИ ИЕРАРХИЧЕСКОЙ КЛАСТЕРИЗАЦИИ ДЛЯ АНАЛИЗА СТРУКТУРЫ УЧЕБНОЙ НАГРУЗКИ КАФЕДРЫ

*С.А. Ермоченко, Д.Д. Гончарова
Витебск, ВГУ имени П.М. Машерова*

Формирование учебной нагрузки кафедры и распределение на её основе учебных поручений между преподавателями кафедры в некоторых случаях представляет собой достаточно сложную и трудоёмкую задачу. Нагрузка состоит из большого количества дисциплин, каждая из которых может дробиться на более мелкие элементы, распределяемые между разными преподавателями, например, отдельно лекционная нагрузка и нагрузка на проведение практических занятий или лабораторных работ по подгруппам. Кроме того, количество ограничений при распределении таких элементов между преподавателями может быть достаточно большим. Например, невозможность проведения некоторых видов занятий преподавателями, не имеющими учёной степени; минималь-

ный и максимальный объём нагрузки на год для каждого преподавателя и т.д. Одним из самых сложных ограничений является учёт специфики каждой дисциплины и квалификация или опыт работы каждого преподавателя. Так могут быть предметы, по которым практические занятия или лабораторные работы могут проводить не все преподаватели кафедры, а лишь некоторая часть, имеющих для этого достаточную квалификацию и опыт. При большом количестве элементов нагрузки задача распределение учебных поручений становится очень трудоёмким и долговременным процессом, требующем от заведующего кафедрой рассматривания различных вариантов и комбинаций элементов, экспертной оценки трудоёмкости каждого элемента нагрузки и её учёт при распределении. Так, по кафедре прикладного и системного программирования на 2023/24 учебный год нагрузка состоит из более чем 1300 элементов, которые необходимо было распределить между 33 штатными единицами.

Применение иерархического кластерного анализа может облегчить анализ структуры нагрузки, объединяя некоторые элементы в группы схожих элементов.

Цель данной работы – выработать концепцию применения алгоритмов иерархической кластеризации при реализации программного обеспечения распределения учебных поручений.

Актуальность работы продиктована её практической ориентированностью на решение насущных проблем, стоящих перед заведующими кафедрами, за счёт автоматизации отдельных процессов.

На основе поставленной цели сформулируем задачи:

– формализовать требования для автоматизированной системы распределения учебных поручений;

– описать и проанализировать возможности методов иерархической кластеризации по анализу структуры нагрузки кафедры.

Материал и методы. Данное исследование базируется на классических методах кластерного анализа [1]. Рассматривается иерархический кластерный анализ и, в частности, агломеративные методы. Отличительной особенностью иерархических методов является пошаговое построение дерева кластеров с постепенным объединением отдельных анализируемых объектов в более крупные группы. Такие методы, в отличие от методов, где заранее задаётся количество кластеров, позволяют формировать дерево кластеров без начального предположения о полученном результате. Такие методы как раз удобно применять на первых этапах анализа данных.

В качестве анализируемых данных рассматривается нагрузка кафедры прикладного и системного программирования за последние 5 лет. Использование данных за несколько лет позволит в будущем проследить, какие изменения в структуре кластеров произошли, а какие характеристики остаются более-менее постоянными.

Результаты и их обсуждение. Так как задачей исследования является не просто проведение разовой кластеризации данных, а ежегодное выполнение данной процедуры в процессе формирования и распределения нагрузки, то важным требованием является разработка программного модуля или библиотеки, которая позволит осуществлять указанную процедуру в составе программного обеспечения проектируемой системы. Для этого на концептуальном уровне необходимо спроектировать программные интерфейсы, цель которых – обобщение задачи иерархической кластеризации.

Так как проектируемая система будет иметь сложную логику функционирования и может потребовать постоянной доработки и модификации, то важным является выбор языка программирования и технологий реализации, обладающие максимальной гибкостью. Кроме того, язык программирования должен иметь строгую статическую типизацию, так как поддержкой системы с течением времени может заниматься не один человек, и необходимо, во-первых, обеспечить преемственность владения программным кодом, во-вторых, создать предпосылки для эффективной командной работы.

На основе описанных особенностей было принято решение использовать язык программирования и технологии Java [2]. Также плюсами выбранных средств являются:

- относительно высокое быстродействие;
- кроссплатформенность, что даёт некоторую независимость в выборе аппаратных средств или относительную свободу при необходимости миграции на новую аппаратную платформу;
- распространение на условиях бесплатной лицензии.

При проектировании интерфейсов применялись принципы и подходы объектно-ориентированного программирования и проектирования [3].

Рассмотрим кратко основные полученные результаты. Так, основой библиотеки кластеризации является класс `Cluster`, представляющий собой некоторый кластер. Исходный код класса представлен ниже (тривиальные методы опущены для краткости):

```
public class Cluster<T> {
    private T object;
    private List<Cluster<T>> subclusters;
    private double distance;
    // конструкторы, get-теры, set-теры
}
```

Данный класс является обобщённым, что позволяет применять его для кластеризации объектов любой природы.

При проведении процедуры объединения нескольких кластеров в более крупные кластеры на каждом шаге метода агломеративной иерархической кластеризации выбираются кластеры, наиболее близкие друг к другу в смысле используемой метрики. Таким образом ключевым вопросом применения методов кластеризации является описание способа измерения меры близости двух кластеризуемых объектов между собой, которую в дальнейшем будем называть расстоянием между объектами [1]. Для обобщения этой подзадачи опишем отдельный интерфейс:

```
public interface ObjectDistanceCalculator<T> {
    double calcDistance(T a, T b);
}
```

Помимо вычисления расстояния между элементарными (исходными) объектами, необходимо будет вычислять расстояние между целыми кластерами. Для этого также могут применяться различные способы или алгоритмы. Чтобы обобщить их, воспользуемся возможностями принципа наследования в объектно-ориентированном программировании [3] и опишем абстрактный класс для вычисления такого расстояния, подклассы которого могут реализовывать различные метрики, например, метод ближайших соседей, метод дальних соседей, центроидный метод и т.д. [1].

Исходный код данного класса здесь приводить не будем ввиду его громоздкости. Кроме того, основная идея этого класса практически не отличается от интерфейса `ObjectDistanceCalculator`. Добавляется лишь зависимость от указанного интерфейса.

В дальнейшем для кластеризации объектов, связанных с нагрузкой, достаточно будет лишь описать подклассы, реализующие требуемые интерфейсы.

Опишем общую концепцию работы таких подклассов.

Основной задачей, решение которой непосредственно будет влиять на результат кластеризации и адекватность этого результата поставленной цели, является моделирование меры сравнения двух элементов нагрузки. Каждый элемент нагрузки имеет большое количество числовых параметров, позволяющих вычислять расстояние с использованием различных метрик. Но эти параметры не характеризуют элементы нагрузки с точки зрения тех ограничений, которые влияют на способ распределения нагрузки. Какое-то влияние будет оказывать лишь суммарная нагрузка в часах.

При кластеризации, безусловно, стоит учитывать, какие виды занятий присутствуют в одном элементе: есть ли часы лекций, практических занятий, лабораторных

работ, экзамены, зачёты и т.д. Можно было бы для каждой возможной комбинации видов занятий в рамках одного элемента нагрузки сопоставлять некоторое число и использовать его как характеристику, но, поскольку абсолютное значение этого числа никакого смысла в предметной области не имело бы, то вычисление модуля разности таких характеристик для некоторых двух элементов нагрузки не давало бы действительной информации о мере близости объектов в рамках данного фактора. Поэтому при вычислении расстояния воспользуемся следующим подходом: при сравнении двух объектов если некоторый вид занятия присутствует только у одного из двух объектов, то будем увеличивать на единицу счётчик различающихся видов занятий. А затем в качестве меры расстояния по рассматриваемой характеристике будем принимать отношение значения счётчика к общему количеству видов занятий.

Ещё одной характеристикой, важной с точки зрения кластеризации, является условная трудоёмкость дисциплины или вида работы. Пока такая характеристика будет определяться экспертным методом. В дальнейшем будут рассмотрены методы оценки этой характеристики другими методами.

Заключение. Таким образом, в результате проделанной работы выработана концепция применения методов иерархической кластеризации для реализации системы распределения учебных поручений.

Работа выполнена в рамках НИР «Методы искусственного интеллекта для оптимизации образовательного процесса, №ГР 20210790» задания «Информационные технологии повышения качества образовательного процесса» ГПНИ «Цифровые и космические технологии, безопасность человека, общества и государства».

1. Жамбю, М. Иерархический кластер-анализ и соответствия. – Москва: Финансы и статистика, 1988. – 345 с.
2. Gosling, J. The Java® Language Specification. Java SE 17 Edition [Electronic Resource] / J. Gosling, B. Joy, G. Steele etc. – Oracle Inc., 2021. – Mode access: <https://docs.oracle.com/javase/specs/jls/se17/html/index.html>. – Date access: 23.01.2022
3. Гамма, Э. Приёмы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. – Санкт-Петербург: Питер, 2020. – 368 с.

ВОЗМОЖНОСТЬ ИСПОЛЬЗОВАНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ВЫЯВЛЕНИЯ СКРЫТЫХ ЗАБОЛЕВАНИЙ ГАЙМОРОВЫХ ПАЗУХ

*Т.Д. Жук, П.В. Травничева, А.В. Каторжевский, И.А. Гаврученко
Витебск, ВГУ имени П.М. Машерова*

Актуальной задачей нейросетевого моделирования является использование нейронных сетей для решения прикладных задач медицинского характера. Одним из наиболее распространенных вопросов является диагностика заболеваний с помощью механизмов компьютерного зрения.

Гайморит, как правило, является следствием ОРВИ, но и может быть вызван неправильным лечением зубов верхней челюсти. По причине близкого расположения корней верхних зубов и гайморовых пазух, общераспространенные заболевания зубов (кариес или хронический периодонтит, например) могут спровоцировать развитие одонтогенного гайморита. Так, при возникновении патологии пародонта зубов верхней челюсти, инфекция может распространиться из зубной полости в гайморовы пазухи.

Частым случаем в медицинской практике является хронический гайморит, который возникает в следствии стоматологических проблем пациента. Один из способов выявления подобных случаев – анализ панорамного снимка челюсти.

Целью настоящей работы является исследование возможности диагностики заболеваний гайморовых пазух с помощью методов искусственного интеллекта.