Лабораторная работа №4
КЛАСТЕРНЫЙ АНАЛИЗ

ЦЕЛЬ РАБОТЫ: ознакомление с проблемой кластерного анализа при интеллектуальной обработке данных в информационных системах; изучение алгоритмов кластеризации, использующих построение минимального остовного дерева; приобретение навыков в программной реализации изученных алгоритмов и в компьютерном проведении кластерного анализа.

1. ОБЩИЕ СВЕДЕНИЯ О КЛАСТЕРНОМ АНАЛИЗЕ

Задача кластерного анализа заключается в выявлении естественного локального сгущения объектов, каждый из которых описан набором переменных или признаков. В процессе кластерного анализа осуществляется разбиение исследуемого множества объектов, представленных многомерными данными, на группы похожих в определенном смысле объектов, называемых кластерами.

Кластерный анализ лежит в основе любой интеллектуальной деятельности и является фундаментальным процессом в науке. Любые факты и явления должны быть упорядочены или сгруппированы по их схожести, т.е. классифицированы, прежде чем разрабатываются общие принципы, объясняющие их поведение и взаимную связь.

Кластерный анализ может быть применен к любой предметной области, где необходимо исследовать объекты, заданные экспериментальными или статистическими данными. Применение кластерного анализа не требует предварительных знаний об анализируемых данных, что позволяет его использовать для данных практически произвольной природы. Поэтому задача кластерного анализа обычно решается на начальных этапах исследования, когда о данных мало чего известно. Ее решение помогает лучше понять природу анализируемых объектов.

С точки зрения априорной информации о числе кластеров, на которое требуется разбить исследуемую совокупность объектов, задачи кластерного анализа можно подразделить на следующие основные типы:

– число кластеров априори задано;

– число кластеров неизвестно и подлежит определению;

– число кластеров неизвестно, но его определение не является условием решения задачи, а необходимо построить иерархическое дерево (дендрограмму) разбиения анализируемой совокупности объектов на кластеры. В данном случае требуется осуществить иерархическую кластеризацию, т.е. построить иерархическое дерево разбиения (дендрограмму) анализируемой совокупности объектов на кластеры. Дендрограммой называется такая последовательность разбиений, в которой каждое разбиение вложено в последующее разбиение в последовательности.

Результатом кластерного анализа является как выделение самих кластеров, так и определение принадлежности каждого объекта к одному из них. Часто результаты выполненного кластерного анализа являются отправной точкой для дальнейшего проведения интеллектуального анализа данных. С помощью этого дальнейшего анализа пытаются установить: что означает выявленное разбиение на кластеры и чем оно вызвано; кто является типичным «представителем» каждого кластера; с помощью каких «представителей» кластеров следует решать различные проблемные задачи и др.

2. ФОРМАЛИЗАЦИЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ

В процессе кластеризации осуществляется группировка объектов, к которым можно отнести все, что угодно, включая наблюдения и события.

Состояние исследуемого объекта может быть описано с помощью вектора дескрипторов или многомерного набора зафиксированных на нем признаков:

$$X = \left\{x^1, x^2, ..., x^p\right\}.$$

Тогда X_i – результат измерения этих признаков на i-ом объекте. Часть признаков может носить количественный характер и принимать любые действительные значения. Другая часть носит качественный характер и позволяет упорядочивать объекты по степени проявления какого-либо качества (например, бинарный признак, отображающий присутствие или отсутствие данного свойства).

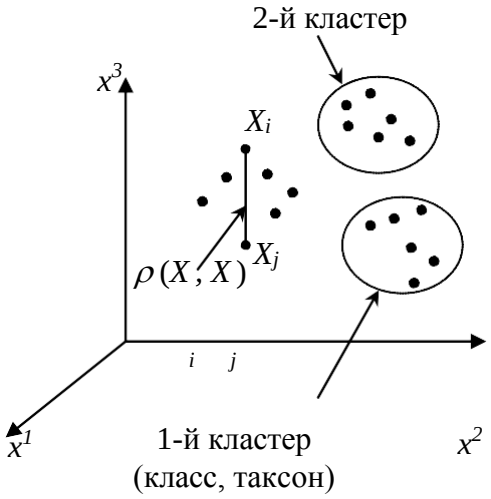


Рис. 1. Геометрическая интерпретация кластеров

Очевидно, что любое многомерное наблюдение может быть геометрически интерпретировано в виде точки в p-мерном пространстве (см. рис. 1). Естественно предположить, что геометрическая близость двух или нескольких точек в этом пространстве означает принадлежность этих точек к одному кластеру.

Чтобы решить задачу кластеризации алгоритмически, необходимо количественно определить понятие сходства и разнородности объектов. Тогда объекты X_i и X_j будем относить к одному кластеру, когда расстояние между этими объектами будет достаточно малым, и к разным – если будет достаточно большим.

Таким образом, для определения «похожести» объектов необходимо ввести меру близости или расстояния между объектами.

Неотрицательная, вещественнозначная функция $\rho(X_i, X_j)$ называется *функцией расстояния (метрикой)*, если:

1. $\rho(X_i, X_j) \geq 0$ для всех X_i и X_j;
2. $\rho(X_i, X_j) = 0$ тогда и только тогда, когда $X_i = X_j$;
3. $\rho(X_i, X_j) = \rho(X_j, X_i)$;

4. выполняется неравенство треугольника

где X_i, X_j, X_k – любые 3 объекта.

$$\rho(X_i, X_j) \le \rho(X_i, X_k) + \rho(X_k, X_j),$$

Существуют различные способы вычисления расстояний. Наиболее употребительна *евклидова метрика*, которая связана с интуитивным представлением о расстоянии.

$$\rho_E(X_i, X_j) = \sqrt{\sum_{k=1}^{p} (x_i^k - x^k)^2}_{j} .$$

Хеммингово расстояние используется как мера различия объектов, задаваемых дихотомическими (бинарными) признаками. Данная мера равна числу несовпадений значений соответствующих признаков в рассматриваемых *i*-ом и *j*-ом объектах:

$$\rho_X(X_i, X_j) = \sum_{k=1}^{p} \left| x_i^k - x_j^k \right|.$$

Существуют и другие более абстрактные меры близости. Если исследуемые признаки смешанные (количественные и качественные), то необходима нормировка по всем значениям x_i^k количественных признаков x^k:

$$\frac{x_i^k}{\max x_i^k}, \quad i = 1, \dots n$$

которая приводит к общей евклидовой мере близости. При разработке моделей и методов кластеризации обычно исходят из того, что объекты внутри одного кластера должны быть близки друг к другу и далеки от объектов, вошедших в другие кластеры. Точность кластеризации определяется тем, насколько близки объекты одного кластера и насколько удалены объекты, принадлежащие разным кластерам.

3. АЛГОРИТМ КЛАСТЕРИЗАЦИИ

Пусть результаты измерений *n* объектов представлены в виде матрицы

данных размером $p \times n$, в которой множество строк представляет объекты, а множество столбцов – признаки.

Тогда близость между парами объектов можно представить в виде симметричной матрицы расстояний:

Исследуемый в данной работе алгоритм кластеризации основан на понятии минимального остовного дерева, построенного с использованием матрицы расстояний R.

Общий алгоритм кластерного анализа, использующий подалгоритм построения минимального остовного дерева, содержит следующие основные шаги:

Шаг 0. [Инициализация] Построение матрицы расстояний (близости) R по результатам измерений n объектов, представленным матрицей данных размером $p \times n$.

Шаг 1. [Построение минимального остовного дерева] С использованием матрицы R осуществляется построение минимального остовного дерева T. Для построения минимального остовного дерева предлагается воспользоваться алгоритмами Крускала и Прима, описанными в разделе 4.

Пусть $\{d_1, d_2, ..., d_{n-1}\}$ – множество весов (длин) рѐбер минимального остовного дерева. На рис. 2 представлен пример минимального остовного дерева, построенного для 7 объектов.

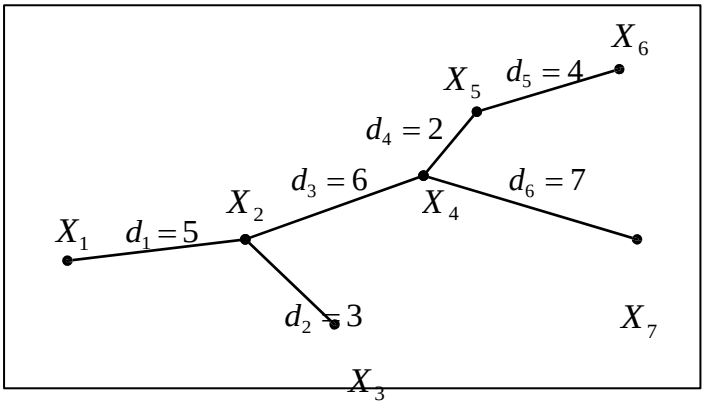


Рис. 2. Пример минимального остовного дерева T

Шаг 2. [Группировка объектов в кластеры] Вершины – объекты минимального остовного дерева группируются в кластеры.

Выбираются два объекта, которым соответствует минимальное ребро $\min_j d_j$, где $j=\overline{1,n-1}$. Далее эти объекты стягиваются в один кластер (класс, таксон, страту) и процедура шага 2 повторяется до тех пор, пока на *n-1* этапе группирования не будет сформирован один кластер, объединяющий все объекты. STOP.

На рис. 3 представлена последовательность группировки объектов в кластеры для заданного на рис. 2 примера минимального остовного дерева. Порядок объединения объектов в кластеры отображен на ребрах, которые связывают объединяемые объекты (см. рис. 3.1-3.7). Таким образом, первыми объединяются объекты X_4 и X_5, которые в T связывает минимальное ребро d_4 с весом 2 (см. рис. 2 и 3.1). Вторыми объединяются объекты X_2 и X_3, связанные ребром d_2 с весом 3 (см. рис. 2 и 3.2), и так далее, пока на шестом этапе группирования ранее связанные объекты (X_1, X_2, X_3, X_4, X_5, X_6) не будут объединены с объектом X_7 ребром с весом 7 (см. рис. 2 и 3.6).

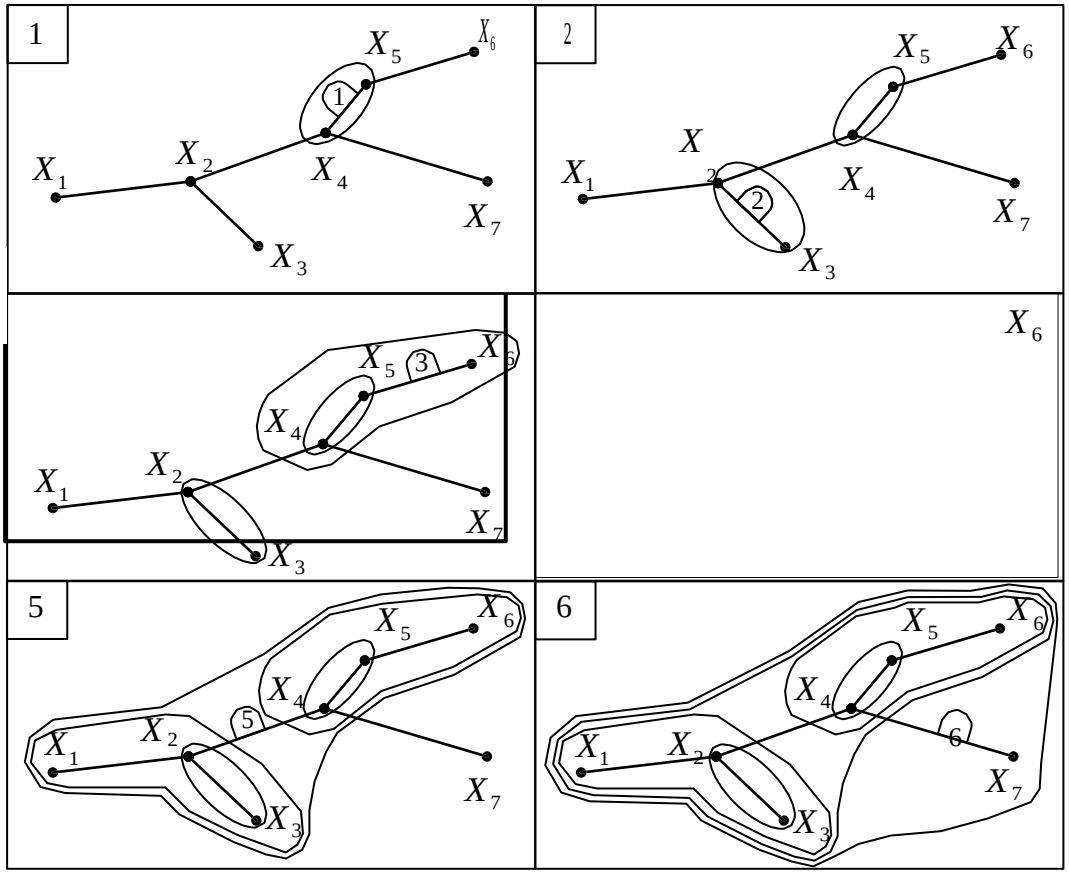


Рис. 3. Последовательность группировки объектов в кластеры

Порядок объединения объектов в кластеры может быть задан с помощью скобочного описания. Для рассматриваемого примера такая скобочная запись имеет следующий вид:

$$\Big(\big(\big((X_4, X_5), X_6\big), \big((X_2, X_3), X_1\big)\big), X_7\Big).$$

Наиболее удобным и распространенным способом описания результатов иерархической кластеризации является дендрограмма, изображенная для рассматриваемого примера на рис. 4.

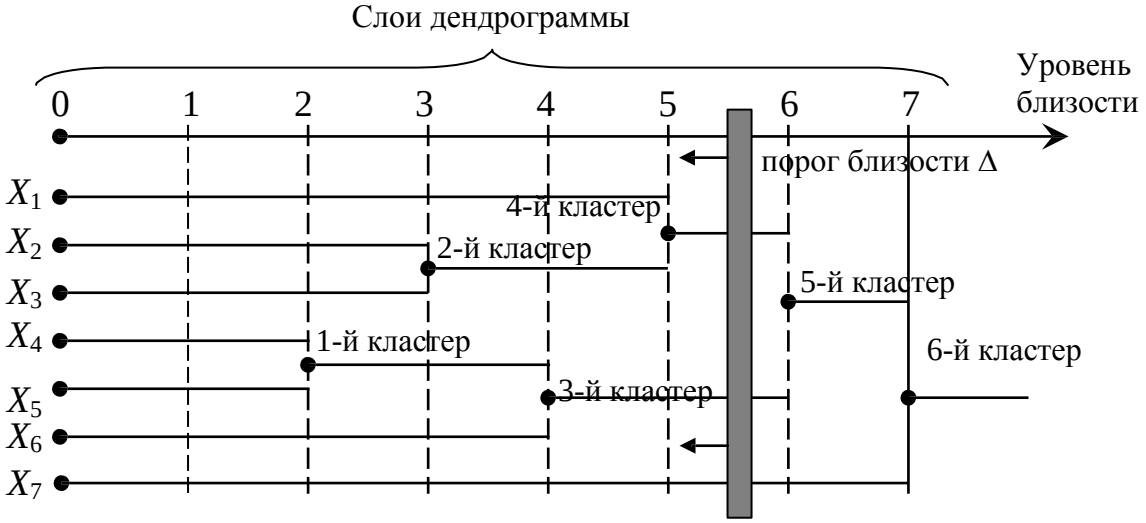


Рис. 4. Дендрограмма результатов иерархической кластеризации

Дендрограмма имеет специальную структуру дерева, состоящего из слоев вершин, любая из которых представляет один кластер. Каждый слой вершин характеризуется своим уровнем близости. Расположение произвольной вершины – кластера относительно слоев дендрограммы определяется ее уровнем близости, который измеряется весом последнего стягиваемого ребра при образовании данного кластера.

Формирование дендрограммы начинается со слоя нулевого уровня близости, в котором каждый из исходных объектов помещается в отдельный кластер. Линии, соединяющие вершины, формируют кластеры, которые вложены один в другой. В целом, дендрограмма отражает порядок вложенности кластеров, в котором число кластеров последовательно уменьшается, пока не будет сформирован один кластер, объединяющий все исходные объекты.

Срез дендрограммы, определяемый ее порогом близости Δ, используется для проведения кластерного анализа на заданное число кластеров. С этой целью порог близости Δ последовательно уменьшается от максимально возможного значения до нуля. При таком уменьшении Δ дендрограмма последовательно распадается сначала на два кластера, затем на три и т.д., пока не будут выполнены требования к необходимому числу кластеров.

4. АЛГОРИТМЫ ПОСТРОЕНИЯ МИНИМАЛЬНОГО ОСТОВНОГО ДЕРЕВА (МОД)

4.1. Общие сведения о задаче построения МОД

Действия алгоритмов построения минимального остовного дерева T рассмотрим на конкретных примерах матрицы расстояний R.

Пусть задана симметричная матрица расстояний R:

$$R = \begin{vmatrix} 0 & 11 & 9 & 7 & 8 \\ 11 & 0 & 15 & 14 & 13 \\ 9 & 15 & 0 & 12 & 14 \\ 7 & 14 & 12 & 0 & 6 \\ 8 & 13 & 14 & 6 & 0 \end{vmatrix},$$

которой можно поставить в соответствие взвешенную полносвязную сеть G с n=5 вершинами и m=10 ребрами, представленную на рис. 5.

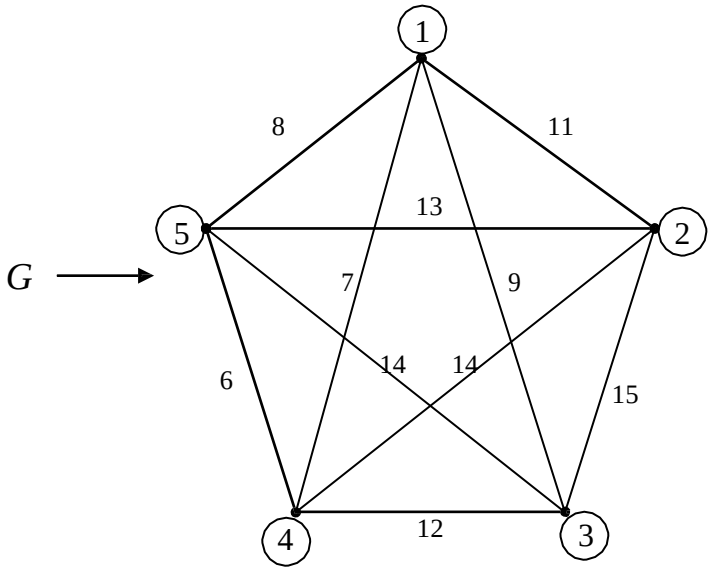


Рис. 5. Исходная сеть G для построения МОД

Тогда минимальным остовным деревом T сети G является самая дешевая подсеть, т.е. подсеть минимального веса, которая покрывает все вершины сети G и не содержит циклов. Очевидно, что такая подсеть является деревом.

Для построения минимального остовного дерева T во взвешенной, связной и полной сети G с n вершинами и m ребрами можно использовать ряд алгоритмов, среди которых наиболее известными являются алгоритмы Крускала и Прима.

4.2. Алгоритм Крускала

Шаг 0. [Инициализация] Создаем сеть T с n вершинами, но без ребер. Создаем сеть H идентичную сети G.

Шаг 1. [Цикл] До тех пор, пока сеть T не является связной сетью выполнять шаг 2, в противном случае STOP.

Шаг 2. [Отыскание ребра с наименьшим весом] Пусть (u,v) – ребро с наименьшим весом в сети H. Если при добавлении ребра (u,v) к сети T в последней не образуется циклов, то это ребро добавляется к T.

Шаг 3. [Удаление (u,v) из *H*] Удаляем ребро (u,v) из сети H.

На рис. 6 представлен пример построения с помощью алгоритма Крускала минимального остовного дерева T для исходной сети G, изображенной на рис. 5.

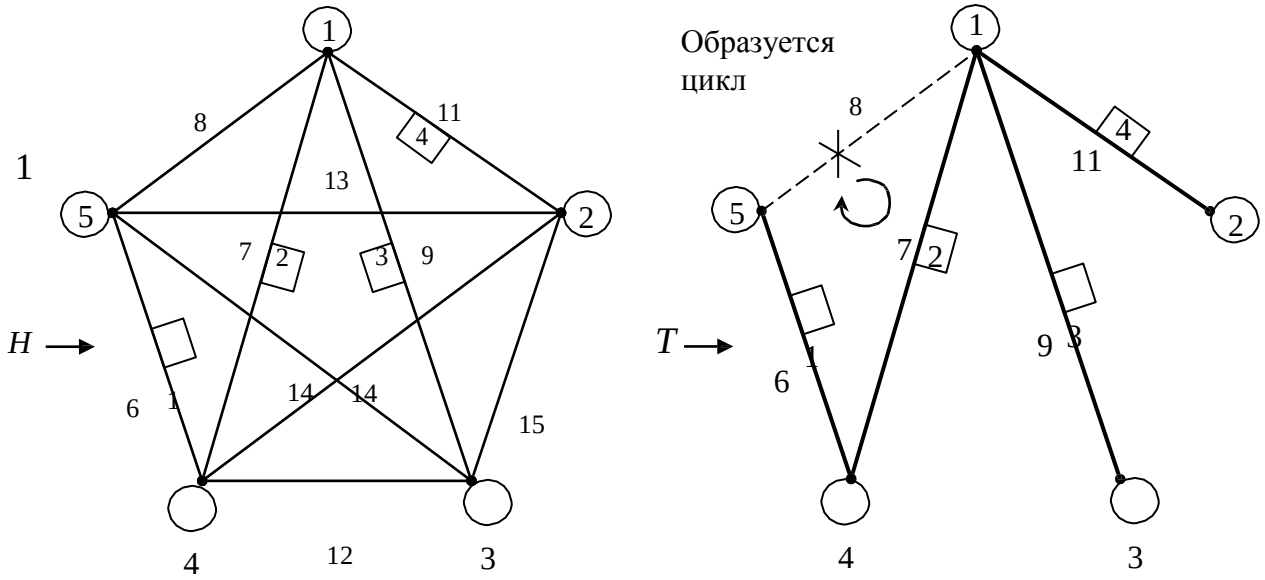


Рис.6. Пример построения минимального остовного дерева по алгоритму Крускала

Порядок присоединения ребер (u,v) к сети *T* и удаления этих ребер из сети *H* указан цифрами в квадратах на соответствующих ребрах этих сетей. Из рис. 6 видно, что на третьем этапе претендентом на присоединение к *T* является ребро (5,1), как имеющее на данном этапе в сети *H* наименьший вес 8. Однако его присоединение к *T* не происходит, так как оно приводит к появлению цикла (1,4,5,1), и на третьем этапе присоединяется следующее по значению ребро (1,3) с весом 9.

Необходимость решения вопросов о связности сети и наличии в ней циклов делают алгоритм Крускала недостаточно эффективным. Следующий алгоритм, разработанный Примом, гарантирует построение минимального остовного дерева без проведения проверок создаваемой сети на связность и наличие в ней циклов.

4.3. Алгоритм Прима

С помощью алгоритма Прима минимальное остовное дерево порождается посредством разрастания одного поддерева от выбранной вершины. Алгоритм реализуется путем прибавления ребер, причем добавляемое ребро должно иметь наименьший вес. Процедура выполняется на сети *G* с *n* вершинами и *m* ребрами до тех пор, пока число ребер в сети *T* не станет равным *n*-1.

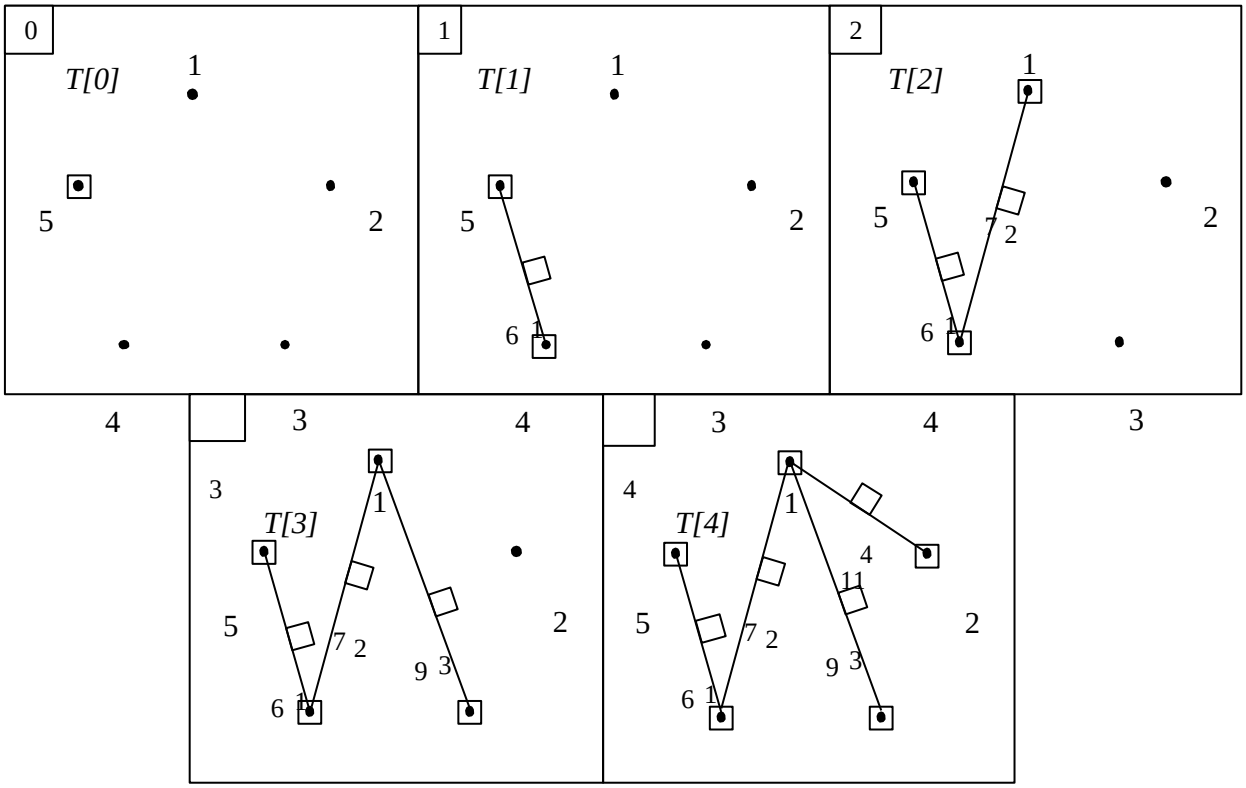
Алгоритм реализует следующие основные шаги:

Шаг 0. [Инициализация] Помечаем все вершины «невыбранными». Создаем сеть *T* с *n* вершинами, но без ребер. Выбираем произвольную вершину и помечаем ее «выбранной».

Шаг 1. [Цикл] До тех пор, пока существуют «невыбранные» вершины, выполнять шаг 2, в противном случае – STOP.

Шаг 2. [Отыскание ребра с наименьшим весом] Пусть (u,v) – ребро с наименьшим весом между произвольно выбранной вершиной *u* и произвольной невыбранной вершиной v. Помечаем v как «выбранную» и добавляем ребро (u,v) в сеть *T*.

На рис. 7 представлен пример построения с помощью алгоритма Прима минимального остовного дерева *T* для исходной сети *G*, изображенной на рис. 5. Порядок добавления ребер указан цифрами в квадратах на соответствующих ребрах порождаемой сети *T*. Выбранные вершины также помечены квадратами. Заметим, что исходные сети *G* для работы алгоритмов Крускала и Прима не обязательно ограничивать только классом полных сетей. Отсутствующим ребрам следует приписать бесконечный вес. Рис. 7. Построение минимального остовного дерева с помощью алгоритма Прима



5. СОДЕРЖАНИЕ РАБОТЫ И ЗАДАНИЕ

В ходе работы необходимо:

1. Изучить различные виды алгоритмов кластерного анализа, отличающиеся подалгоритмами построения минимального остовного дерева.
2. Самостоятельно выбрать предметную область кластеризации и найти набор данных для выполнения кластерного анализа.
3. Программно реализовать один из алгоритмов кластерного анализа на основе построения минимального остовного дерева.
4. Программно реализовать алгоритм k-средних.
5. Получить и проанализировать результаты проведения кластерного анализа при разных значениях исходных данных.

Отчет должен содержать:

1. Название, цель работы, вариант задания.
2. Краткое описание алгоритмов кластеризации.
3. Описание набора исходных данных.
4. Описание процедуры предварительной обработки исходных данных.
5. Листинг программного кода.
6. Результаты проведения кластерного анализа и их анализ и интерпретация.