УДК 519.876.5

Моделирование кластеризации многомерных объектов в Visual C++

© З.Н. Русакова, А.В. Орел

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Представлен гибридный алгоритм кластеризации, не требующей априорной информации ни о числе кластеров, ни о форме выборки. Алгоритм основан на объединении итеративного метода поиска локальных сгущений и методов определения связных компонент графа. Описан программный модуль моделирования задач кластерного анализа, использующий для реализации нелинейные динамические структуры.

***Ключевые слова**: кластеризация, моделирование, программа, алгоритм, графы, метрика, динамические нелинейные структуры.*

Введение. Одна из актуальных задач интеллектуального анализа данных — кластерный анализ, используемый в различных областях: в информационных системах при решении задач распознавания, классификации закономерностей, обработке изображений [1–4].

Под методами кластеризации понимается множество вычислительных процедур, решающих задачу классификации объектов на однородные группы при отсутствии априорной информации о характере распределения [5, 6].

Постановка задачи. Выборка многомерных перемешанных векторов наблюдений представляется в виде прямоугольной таблицы, где строка — вектор измерения признаков объекта. Задача кластеризации состоит в разбиении выборки на кластеры (на подмножества) так, чтобы обеспечить экстремум некоторого критерия функционала качества — максимально правильное количество классифицированных объектов [3, 4, 6].

К наиболее используемым критериям относятся сумма квадратов расстояний до центров кластеров, сумма внутрикластерных расстояний между объектами, сумма попарных внутриклассовых расстояний между элементами кластеров. Критерием управления процессом кластеризации является выбор меры сходства между объектами, называемой также метрикой, или функцией расстояний. Выбор метрики определяет результат кластеризации и, в свою очередь, определяется формой выборки и типами признаков объекта [4, 6].

На практике применяют следующие метрики: евклидово расстояния; манхэттенское расстояние (расстояние городских кварталов); расстояние Чебышева, расстояние Махаланобиса, вычисляющее расстояние между векторами с помощью матрицы ковариаций.

Задачи кластерного анализа решаются на основе таких подходов, как вероятностный, иерархический, теория графов, нечеткие алгоритмы [4–7]. В современном подходе принятие решение осуществляется группировкой комплекса алгоритмов.

Гибридный алгоритм кластеризации. В работе проводятся исследование и разработка структурной модификации алгоритма кластеризации на основе объединения двух подходов: метода поиска локальных сгущений и методов определения связных компонент графа, построения и анализа минимального покрывающего дерева, объединяющего точки данных. Предложенный алгоритм гибридной кластеризации использует идею итеративного метода поиска сгущений и методов поиска покрытий в графах [5–7].

Разработанный алгоритм обеспечивает кластеризацию таких классов, как класс типа слабого сгущения, класс типа изолированного облака и обычных классов типа сильного сгущения. В случае пересекающихся классов алгоритм кластеризации выполняется в предположении слабого их пересечения. Для случая пересекающихся кластеров, например, данные выборки пересекаются в крестообразной форме, а результат является начальным приближением для разбиения другими методами.

Двухэтапная процедура кластеризации. В предлагаемом алгоритме реализуется двухэтапная процедура кластеризации, не требующая априорной информации о центрах предполагаемых кластеров и форме выборки данных. На первом этапе реализуется модифицированный метод поиска локальных сгущений: для каждого элемента выборки определяется локальное сгущение, центром которого является сам элемент. Локальное сгущение определяется как список ближайших соседей. На втором этапе на основе алгоритмов построения связных компонент графа осуществляется кластеризация путем слияния отдельных сгущений в кластеры.

Суть итеративного метода поиска сгущений заключается в применении гиперсферы заданного радиуса и заданного центра для поиска ближайших соседей, формирующих локальные сгущения объектов, т. е. совокупности точек, попавших внутрь данной сферы. Для этого требуется вычисление матрицы расстояний между объектами и выбор первоначального центра сферы. Алгоритм выбора радиуса сферы играет определяющую роль при решении задачи. Адаптация алгоритма к форме кластера реализуется путем настройки параметра, учитывающего специфику типов кластеров.

На первом этапе алгоритма формируются списки ближайших соседей, в качестве критерия объединения которых выбирается минимальное расстояние от фиксированной точки до остальных. Процедура определения локального сгущения состоит в следующем: последова-

тельно перебирается выборка объектов, которые рассматриваются как точки в многомерном пространстве. Важный частный случай — точка на плоскости, которую можно рассматривать или как проекцию многомерного вектора, или как результат редукции. Для каждой точки создается список ближайших точек в смысле некоторой метрики. С этой целью для каждой рассматриваемой точки (текущая точка) вычисляются расстояния в евклидовой метрике между ней и всеми остальными и расчитывается минимальное расстояние. Это расстояние определяет радиус сферы для поиска локального сгущения.

Далее применяется модифицированный метод поиска сгущений для ближайших соседей с адаптивным радиусом сферы: радиус масштабируется коэффициентом k, который выбирается из соображения, что разброс значений лежит в интервале 3σ, где σ — среднеквадратическое отклонение, а для минимума это условие выполняется строже. Из анализа процесса кластеризации в качестве начального используется эвристическое значение $k = 1{,}5$. Оно оказывается допустимым с учетом таких факторов, как скорость и качество кластеризации. Для адаптации алгоритма к форме выборки значение k модифицируется и принятие решения осуществляется на основе комплекса алгоритмов с разными значениями k, определяющего радиус сферы поиска локального сгущения. В ближайшие соседи включаются все точки, попадающие в рассматриваемую сферу.

На втором этапе кластеризации объединяются полученные цепочки. Принцип объединения вытекает из инцидентности вершин и дерева минимального покрытия в представлении графов. Если элемент входит в список ближайших родительского элемента, к этому списку подключается список ближайших дочернего элемента (его самого), что обеспечивает разделение на классы для многих видов кластеров.

Нелинейные динамические структуры гибридного алгоритма. Исходные данные представляются в виде прямоугольной таблицы, где строка — вектор измерения признаков, описывающих объекты. Объекты во входной выборке перемешаны.

Для моделирования кластеризации разработана нелинейная динамическая структура, описываемая двунаправленным динамическим списком [8–10], информационным полем которого является другой список. Эту структуру можно представить как горизонтальный двунаправленный динамический список, к каждому звену которого подвешен вертикальный список. Горизонтальный список включает элементы всей выборки. В вертикальный список для каждой точки, записанной в звене горизонтального списка, включаются ближайшие соседи, определяющие локальное сгущение. Информационные поля звеньев содержат номера точек, заданных при вводе, и вектор их координат. Описанная структура представлена на рис. 1.

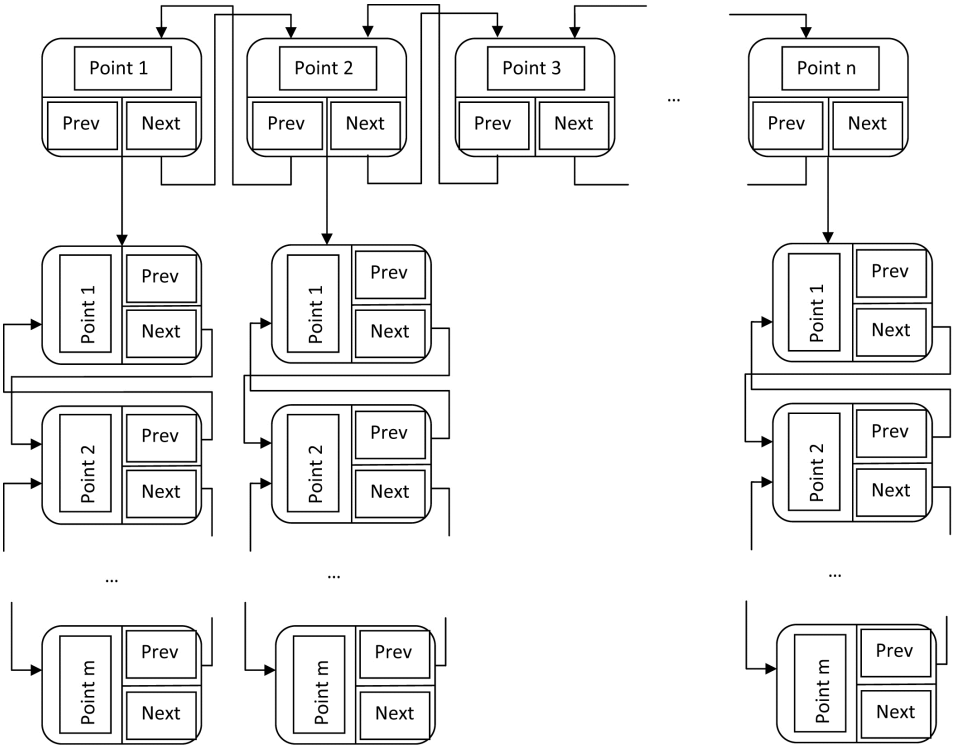

Рис. 1. Представление структуры данных алгоритма кластеризации

Для программной реализации описанной нелинейной динамической структуры данных используются средства Visual C++ [8–10]. Решение задачи осуществляется на основе объектно-ориентированного подхода с разработкой классов, методами которых решается задача. Основные спроектированные классы рассмотрены далее. Звено горизонтального списка — элемент кластеризации — включает поля и методы, приводимые ниже.

Информационное поле Pointz *p — указатель на вектор, описывающий точку в многомерном пространстве, в прототипе системы — точка на плоскости. Класс Pointz включает также поле, задающее номер точки при вводе (или имя точки), и поле, куда в результате кластеризации записывается номер кластера, которому назначается точка в процессе кластеризации.

Element *next; Element *previous; — указатели на следующее и предыдущее звенья в горизонтальном списке, включающем все объекты выборки.

ListOfNearElementz *lnp — указатель на вертикальный список, содержащий точки локального сгущения или ближайшие точки для элемента звена (текущей точки) горизонтального списка. Определение класса Element для описания элемента кластеризации:

```
class Element{
public:
   Pointz *p; // указатель на класс элемента
   Element *next; // указатель на следующее звено
   Element *previous; // указатель на предыдущее звено
   ListOfNearElementz *lnp; //указатель на список ближайших соседей
   Element(){p=new Pointz;lnp=new
                        ListOfNearElementz;
            next=0;previous=0; } //конструктор класса
   ~Element(){}
// метод вычисления расстояния на плоскости от текущей точки до точки по указателю p1
double findDistance(Pointz *p1){ return System::Math::Sqrt((p → x – – p1 → x) · (p → x – p1 → x) + (p → y – p1 → y)*(p → y – p1 → y));}
// метод формирования кластеров, описанный вне класса
void clusteringElement(int k){}
};
```

Звено списка ближайших соседей (список локального сгущения текущего элемента), описываемого классом class NearElement, включает следующие поля: указатели на следующий и предыдущий элементы — NearElement *next, NearElement * previous и указатель на элемент выборки (точку в пространстве) Pointz *p:

```
class NearElement{
public:
   Pointz *p;
   NearElement *next;
   NearElement *previous;
   NearElement(){p=0; next=0; previous=0;}
   ~NearElement(){}
};
```

Звено списка ближайших элементов NearElement включает ссылки на следующий и предыдущий элементы, но не содержит ссылку на вертикальный список элементов.

Класс, описывающий список ближайших элементов, включает поля ссылок на начало и конец списка first, last, указатели просмотра списка cur, sel, методы, формирующие список: добавления элемента в голову и хвост списка. Параметр, ограничивающий радиус локальной сферы включения, описывается параметром double distance. Описание класса ListOfNearElementz, методами которого создается список ближайших соседей, — локальное сгущение:

```
class ListOfNearElementz{
public:
  NearElement *first, *last; // указатели на начало и
                                конец списка
  NearElement *cur, *sel;  ; // текущие указатели
  double distance; //радиус сферы включения
  ListOfNearElementz(){  // конструктор first=0;
    last=0;cur=0;   sel=0;   distance=0;  }
  ~ListOfNearElementz(){} //деструктор
// методы класса – добавить элемент в хвост или голову списка
  void copyElementToEnd(Pointz *v);
  void copyElementToHead(Pointz *v){
  };
```

Основная структура моделирования — класс ListOfElementz, описывающий двунаправленный горизонтальный список элементов, который объединяет все описанные и представленные на рис. 1 классы. Каждый элемент горизонтального списка включает ссылку на двунаправленный список локального окружения, формирующего список ближайших соседей. В методах этого класса осуществляется кластеризация: формируются списки ближайших соседей, из списка ближайших удаляются те, у которых расстояние превышает среднее по всем классам ближайших, а также метод, в котором реализуется классификация, т. е. кластерам назначаются точки, вычисляются центры кластеров и объединяются цепочки ближайших элементов. Класс можно описать следующим образом:

```
class ListOfElementz{
public:
Element *first, *last; //    первый, последний
Element *cur, *sel; //    текущий, выбранный
ListOfElementz(){first=0;last=0;cur=0;sel=0;}
// методы класса, реализация методов вне класса
void nearElements(); // метод формирования списков
                        локальных сгущений
void deleteLongDistance(); // метод удаления связей
                        между кластерами
// метод слияния списков локальных сгущений в кластер — кластеризация
void clustering();
};
```

В методы класса включены методы вычисления ближайших соседей, метод удаления связей между кластерами, метод кластеризации, в котором точкам назначаются номера кластеров и объединяются цепочки ближайших элементов.

Основные шаги алгоритма кластеризации. ***Создание структуры нелинейного списка.*** Под нелинейным разветвленным списком понимается список, элементами которого могут быть тоже списки. Из входного потока объектов выборки формируются звенья горизонтального нелинейного двунаправленного списка элементов: в информационные поля записываются многомерный вектор и его основные параметры (в двумерном случае — точка и ее параметры) и номер точки, заданный при вводе. Поле — номер кластера — является вычисляемым и определяется в процессе кластеризации.

Звено списка описано классом class Element. Класс описывает многомерный вектор и его основные параметры, в двумерном случае — точку на плоскости и ее параметры: номер точки, заданный при вводе, вычисляемый параметр — номер кластера, которому назначается точка.

Формирование списков локальных сгущений. На этом этапе точки не назначаются кластерам, а формируются локальные сгущения, которые необходимо объединить в кластеры на следующем шаге. Для каждого элемента звена горизонтального списка, т. е. для каждой точки, создается список ближайших соседей в смысле некоторой метрики. Алгоритм формирования локального сгущения основан на применении сферы с центром в рассматриваемой точке и радиусом, вычисляемым в процессе формирования матрицы близости (в данном случае — расстояний). Радиус сферы модифицируется адаптивным коэффициентом со значениями в интервале 1…3, масштабирующим минимальное расстояние среди расстояний от текущей точки до всех остальных. Этот параметр управляет процессом кластеризации. Один из вариантов его выбора эвристически обосновывается правилом трех сигм. В ближайшие точки текущего элемента включаются те, которые попадают в определяемую алгоритмом сферу. Эти элементы записываются в вертикальные списки для каждого элемента горизонтального списка.

Для каждого списка локального сгущения определяется среднее значение и сохраняется в соответствующем поле класса. По значениям средних каждого списка локального сгущения вычисляется среднее значение для всех списков ближайших. Если среднее списка ближайших больше среднего всех ближайших, то список ближайших удаляется, что позволяет удалить возможные связи между точками соседних кластеров.

Кластеризация элементов. В поле номер кластера для всех элементов горизонтального и вертикальных списков записывается номер кластера, в который элемент попадает в результате кластеризации. Алгоритм разнесения точек по кластерам можно реализовать как рекурсивно, так и итеративно. Цель алгоритма — объединить цепочки ближайших элементов в одну цепочку, соответствующую агрегированному кластеру.

Описание итерационного алгоритма:

просматривается горизонтальный список элементов. Текущему элементу (назовем его родительским) назначается номер кластера. Для рассматриваемого родительского элемента просматривается вертикальный список ближайших элементов, которые назовем дочерними;

на каждом шаге цикла просмотра из вертикального списка локального сгущения выбирается текущий элемент — дочерний для родительского, и для него формируется кластер. Этому выбранному элементу назначается заданный номер родительского кластера, т. е. элемент помещается в кластер родительского элемента. Этот же номер кластера должен быть назначен элементу с таким же номером элемента (или именем точки) в горизонтальном списке и его ближайшим элементам, записанным в его вертикальном списке.

В результате осуществляется агрегирование данных: слияние локальных сгущений дочерних элементов с родительскими, для чего ближайшие элементы дочернего подключаются (переносятся) в список ближайших соседей родительского элемента верхнего уровня.

При подключении списков необходимо определить пересечение элементов рассматриваемого подключаемого дочернего списка ближайших и списка элементов родительского формируемого кластера и исключить элементы пересечения: т. е. элементы, уже входящие в список ближайших, не добавляются в формируемый кластер. После подключения новых элементов списка к кластеру рассматриваемый дочерний элемент горизонтального списка и его вертикальный список ближайших удаляются. Если элементы списка локального сгущения уже включены в список родительского списка, он сам удаляется из горизонтального списка без подключения.

Такая процедура выполняется для всех элементов формируемого кластера. Далее выбирается следующий элемент из списка формируемого кластера. По его номеру выполняется поиск в горизонтальном списке и к формируемому кластеру добавляются новые точки из его дочернего списка ближайших элементов, после чего элемент из горизонтального списка и сам список удаляются.

Процедура формирования кластера заканчивается после исчерпания списка просматриваемых точек кластера. При этом необходим определенный порядок выбора и записи элементов в список кластера. Реализуется вариант: запись новых в хвост списка формируемого кластера, выбор элемента для поиска новых ближайших из головы. Список формируемого кластера после определенного числа итерации подключений практически не пополняется, так как все элементы уже вошли в кластер, что означает окончание итеративной процедуры формирования кластера.

После выхода из цикла по списку ближайших соседей для родительского списка на текущем шаге формируется кластер. Формализо-

вано кластер записан в вертикальном списке текущего элемента горизонтального списка, а все элементы, входящие в сформированный кластер, удалены из горизонтального списка.

Процедура повторяется для следующего элемента горизонтального списка, который еще не принадлежит ни одному из сформированных ранее кластеров.

Оценка качества кластеризации. Для полученной цепочки кластеров вычисляются средние значения и оценки среднеквадратического отклонения по каждому кластеру. Для полученного в результате кластеризации числа кластеров и структуры проводится оценка качества кластеризации: вычисляется среднеквадратический критерий качества. В зависимости от полученного значения критерия процесс кластеризации осуществляется для других значений параметра кластеризации. Критерием окончания является выполнение условия: норма разности значений функционала качества на соседних шагах алгоритма должна быть меньше заданной точности.

Результаты моделирования. В программном модуле для отображения процесса решения задачи используются два основных диалоговых окна: отображения входной выборки и отображения результатов кластеризации. Сначала отображается окно вывода точек выборки, каждая точка обозначается номером (или именем), при вводе выборка перемешана.

После выполнения кластеризации осуществляется разбиение точек по кластерам и вычисляется среднее для каждого кластера. Точки разных кластеров закрашиваются разным цветом, каждая точка выборки выводится с соответствующим ей номером кластера, точки одного кластера объединяются в сеть, и отображается средняя точка кластера.

Результаты кластеризации объектов на плоскости приведены на рис. 2–6. Отображаются два окна: окно просмотра точек выборки и окно точек после кластеризации.

Заключение. Авторами разработан гибридный алгоритм кластеризации, не требующей априорной информации ни о числе кластеров, ни о форме выборки. Кластеризация осуществляется в два этапа.

На первом этапе для каждого элемента выборки определяется локальное сгущение, центром которого является сам элемент. В свою очередь, локальное сгущение определяется как список ближайших соседей. На втором этапе на основе методов определения связных графа осуществляется кластеризация путем слияния отдельных сгущений в кластеры.

Алгоритм обеспечивает кластеризацию в предположении слабого пересечения кластеров: кластеров типа слабого сгущения, типа изолированного облака, среднего сгущения с центром, сильного кластера. В случае пересекающихся кластеров алгоритм кластеризации позволяет получить начальное приближение для разбиения другими методами.

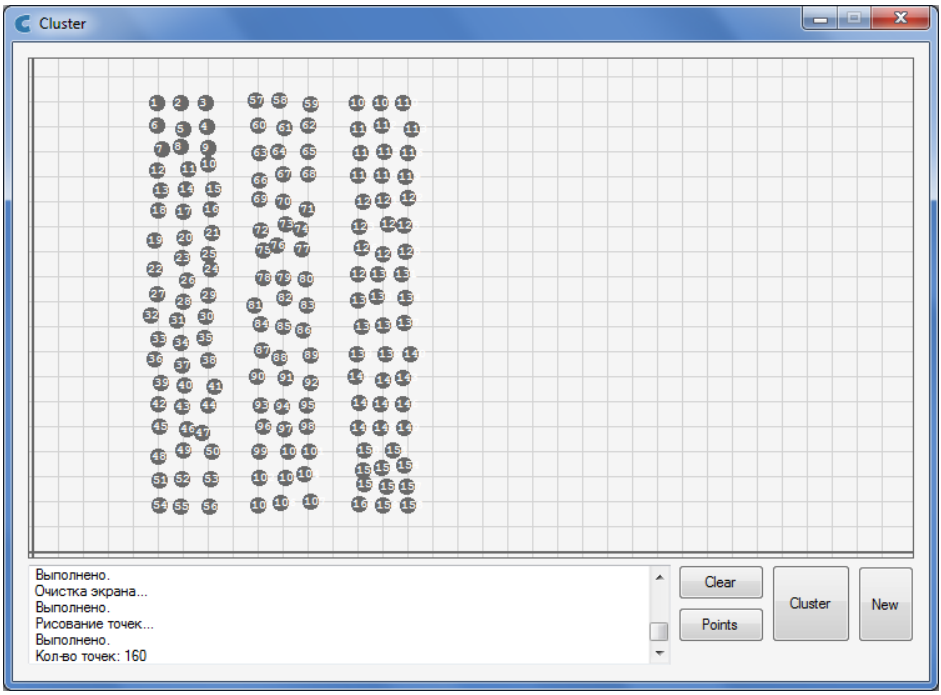

Рис. 2. Входная выборка 1

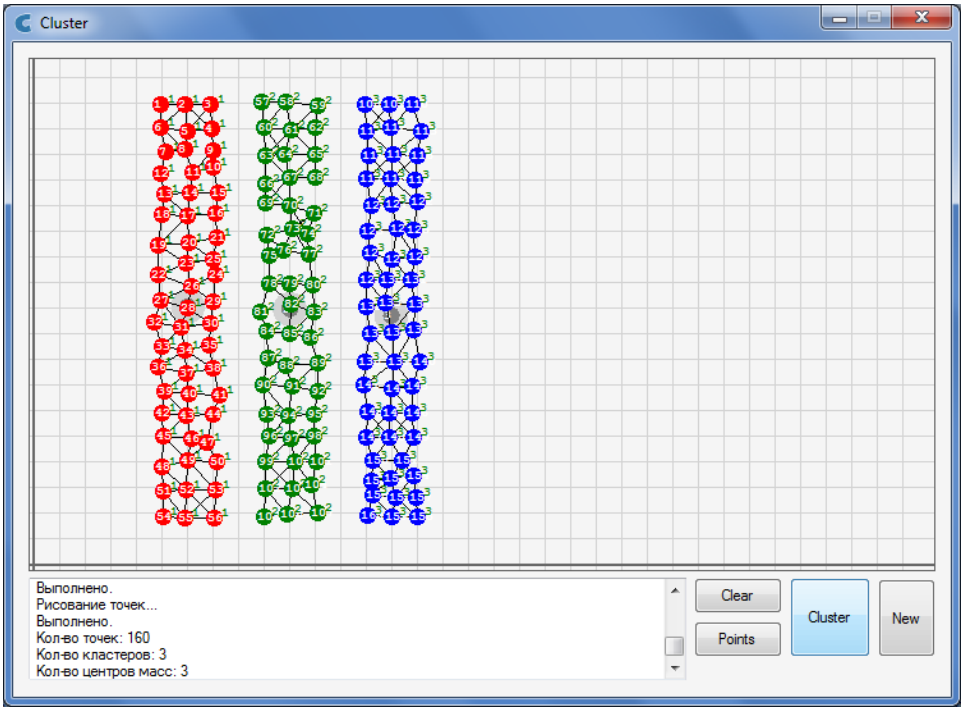

Рис. 3. Результат кластеризации выборки 1

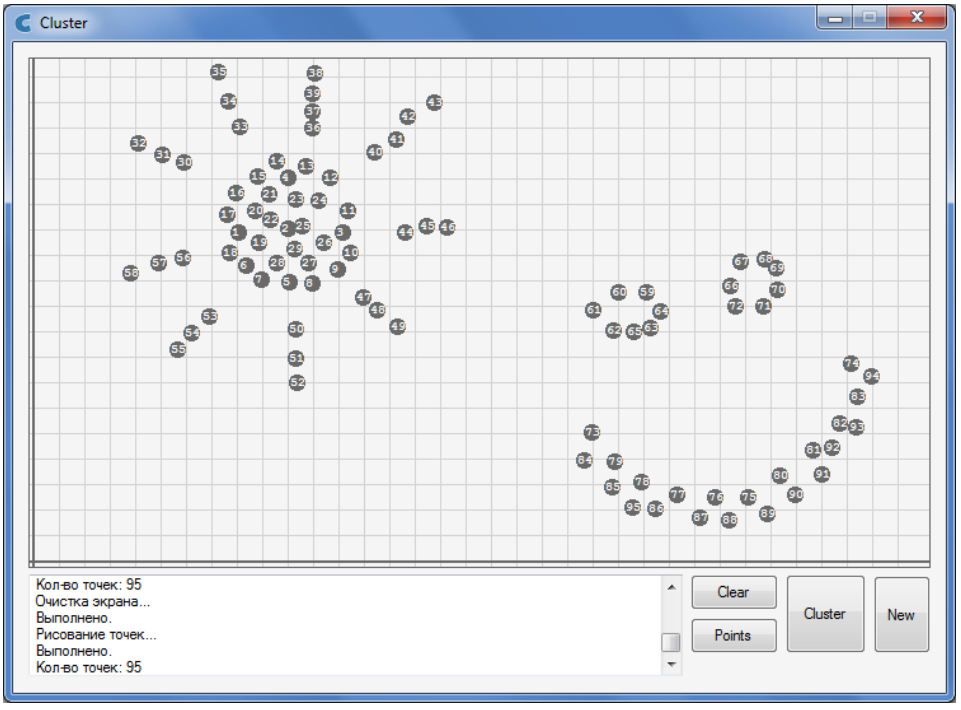

Рис. 4. Входная выборка 2

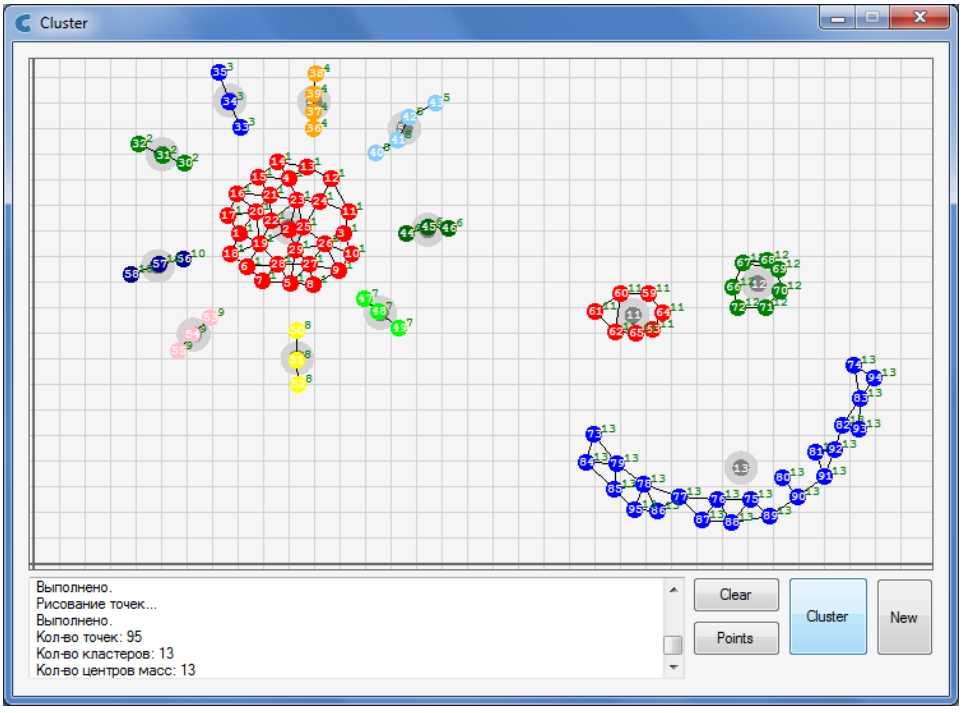

Рис. 5. Результат кластеризации выборки 2

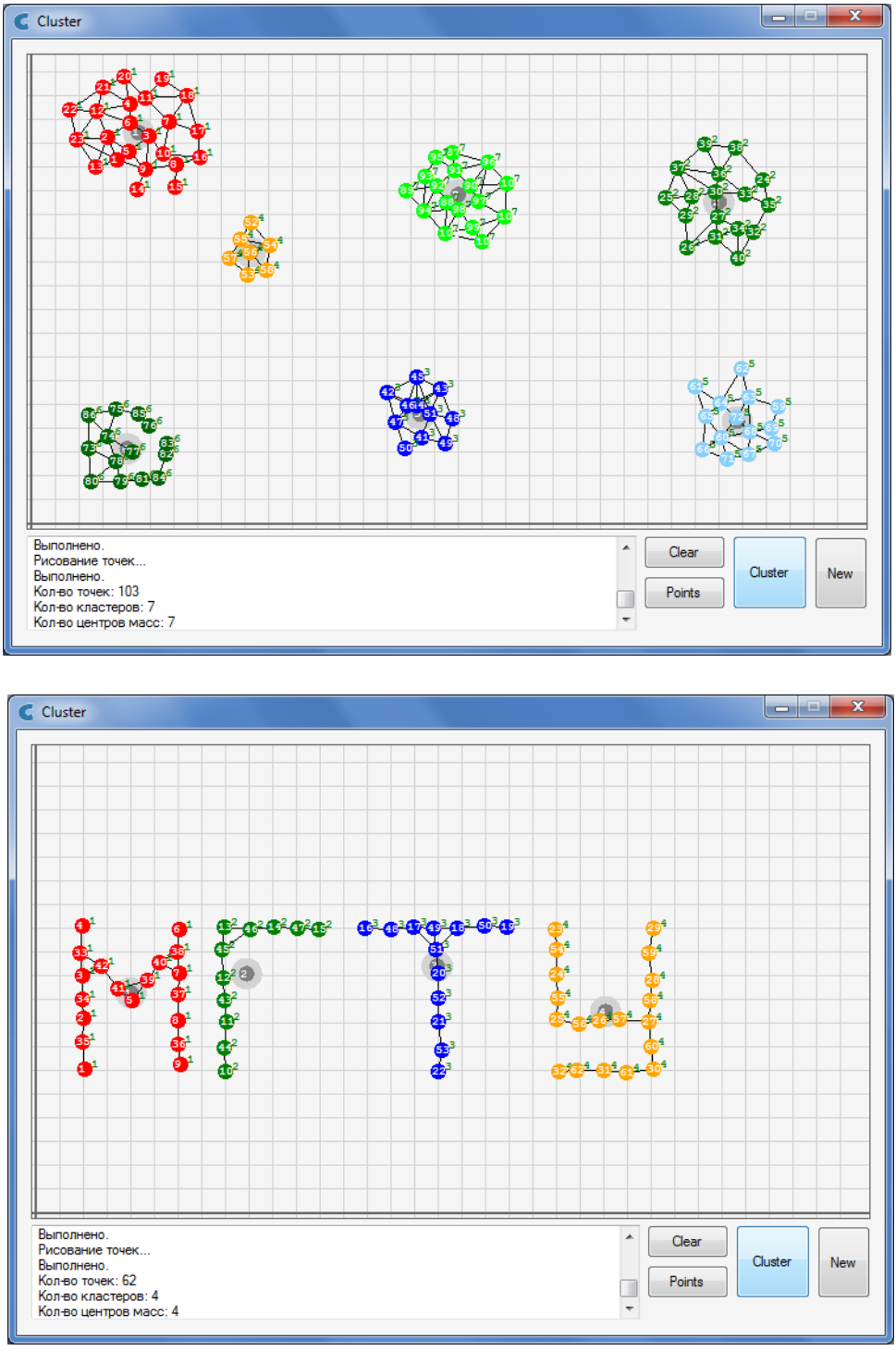

Рис. 6. Результаты кластеризации двух выборок

Полученные результаты могут использоваться не только в задачах, формулируемых как классификационные, но и в задачах исследования концептуальных схем группировки объектов и проверки гипотез, а также выделения схожих групп в совокупности временны́х рядов и сокращения размерности данных.

ЛИТЕРАТУРА

[1] Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. *Прикладная статистика: Классификация и снижение размерности*. Москва, Финансы и статистика, 1989.

[2] Вапник В.Н., Червоненкис А.Я. *Теория распознавания образов.* Москва, Наука, 1974.

[3] Грешилов А.А. Лебедев А.Л. *Компьютерные обучающие пособия для решения задач математической статистики и математического программирования.* Москва, Изд-во МГТУ им. Н.Э. Баумана, 2011.

[4] Дюран Б., Оделл П. *Кластерный анализ.* Москва, Статистика, 1977, 128 с.

[5] Кормен Т., Лейзерсон Ч., Ривест Р. *Алгоритмы: построение и анализ.* Москва, МЦНМО, 2000.

[6] Мандель И.Д. *Кластерный анализ.* Москва, Финансы и статистика, 1988.

[7] Ахо А., Хоркровт Дж., Ульман Дж. *Построение и анализ вычислительных алгоритмов*. Москва, 1979.

[8] Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. *Объектно-ориентированное программирование*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2001.

[9] Русакова З.Н. *Динамические структуры данных и вычислительные алгоритмы: Visual C++*. Санкт-Петербург, Образовательные проекты, 2013.

[10] Шилдт Г. *Теория и практика C++*. Санкт-Петербург, BHV, 1996.

Статья поступила в редакцию 12.02.2014

Ссылку на эту статью просим оформлять следующим образом:
Русакова З.Н., Орел А.В. Моделирование кластеризации многомерных объектов в Visual C++. *Инженерный журнал: наука и инновации*, 2014, вып. 2.
URL: http://engjournal.ru/catalog/it/hidden/1200.html

Русакова Зинаида Николаевна — канд. техн. наук, ст. науч. сотр., доцент кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана. Автор более 60 научных работ в области вычислительной математики, интеллектуальных систем (*intelligent systems*), моделирования и интеллектуализации задач принятия решений, обработки данных. e-mail: z.n.rusakova@mail.ru

Орел Александр Валерьевич — студент факультета «Энергомашиностроение» МГТУ им. Н.Э. Баумана. e-mail: orel_alexandr@mail.ru