OSTIS-2014
(Open Semantic Technologies for Intelligent Systems)

УДК 004.02 : 004.9

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ВСПОМОГАТЕЛЬНЫХ АЛГОРИТМОВ ДЛЯ ПОСТРОЕНИЯ КЛАСТЕРНЫХ СТРУКТУР

Горюнов В.А., Жукевич А.И., Родченко В.Г.

Гродненский государственный университет имени Янки Купалы,
г. Гродно, Республика Беларусь

v.gorunov@grsu.by
san@grsu.by
rovar@mail.ru

При построении систем распознавания использование кластерных структур позволяет формализовать процедуру представления образов и эталонов классов в многомерном признаковом пространстве и в автоматическом режиме выполнить процедуры обучения и классификации. Процесс построения кластерных структур связан с использованием библиотеки программных процедур, реализующих вспомогательные алгоритмы кластеризации, вычисления метрик и межкластерных расстояний, критериев однородности.
Ключевые слова: интеллектуальная система, распознавание образов, кластерный анализ, алфавит классов, словарь признаков, классифицированная обучающая выборка.

Введение

В данной статье представлено описание и программная реализация вспомогательных алгоритмов для построения кластерных структур. Использование кластерных структур для формального представления образов классов при построении систем распознавания позволяет в автоматическом режиме на основе исходных данных классифицированной обучающей выборки (КОВ) реализовать процедуры обучения и окончательного принятия решений. [Родченко, 2006].

Формально весь процесс распознавания может быть реализован в результате выполнения следующей последовательности преобразований:

$$S \xrightarrow{F_1} C \xrightarrow{F_2} A \xrightarrow{F_3} T \xrightarrow{F_4} A^* \xrightarrow{F_5} E \xrightarrow{F_6} R$$

где S – алфавит классов; C – словарь наблюдаемых (измеряемых) характеристик; A – априорный словарь признаков (АСП); T – классифицированная обучающая выборка; A^* – уточненный словарь признаков для построения пространства решений; E – множество эталонов классов; R – множество решений; F_1 – алгоритм получения наблюдаемых характеристик; F_2 – алгоритм построения априорного словаря признаков; F_3 – алгоритм формирования классифицированной обучающей выборки; F_4 – алгоритм сепарирования признаков из априорного словаря по степени их информативности для построения пространства решений; F_5 – алгоритм построения образов эталонов классов в пространстве решений; F_6 – алгоритм процедуры принятия решения [Родченко, 2012].

Алфавит классов S, словарь наблюдаемых характеристик C и априорный словарь признаков A вырабатываются совместно экспертами в соответствующей прикладной области и аналитиками. Классифицированная обучающая выборка T формируется на основе наблюдений состояний объектов с учетом определенных S, C и A на этапах F_1 и F_2.

На основе данных КОВ и критериев однородности происходит сепарирование признаков из АСП и далее при отсутствии сдвигов среди признаков, выполняется построение кластерных структур объектов входящих в КОВ.

Признаки, значения которых по конкретному критерию оказались малоинформативными с точки зрения разделения образов классов исключаются из АСП. При этом выполняется исключение из КОВ строк соответствующих признакам, которые не влияют на процесс кластеризации. Однако можно отклонить предложение системы и приступить к формальному построению кластерных структур.

Классифицированная обучающая выборка фактически представляет собой прямоугольную матрицу числовых значений, которую можно

рассматривать и как один сплошной массив данных, не учитывая принадлежность отдельного объекта к конкретному классу. Программа позволяет в этом случае аттестовать АСП и произвести разбиение на классы, что достигается применением одного из алгоритмов кластеризации и таким образом получить рабочую КОВ. После чего результаты работы программы можно сохранить в файл, отредактировать, дополнить и т.п.

В случае, когда исходная КОВ – уже классифицирована, применяем алгоритмы аттестации АСП для сепарирования признаков по степени информативности и в результате получаем уточненный словарь признаков. Далее применяем алгоритмы кластеризации для аттестации КОВ на предмет пригодности при применении в алгоритме принятия решения. Для аттестации КОВ необходимо задать пороговое значение Q количества ошибочных включений векторов в состав другого класса. Если количество ошибок превышает Q, то предлагается возвратиться к самому началу алгоритма и сформировать новый вариант АСП.

В обоих случаях в финале приходим к реализации алгоритма F_5, в результате которого получается E – множество эталонов классов. Принятие решения по распознаваемому классу происходит на основании исследования принадлежности каждого объекта этого класса к одному из классов множества E или же распознаваемый класс выделяется в отдельный джокер-класс.

1. Реализация процедуры построения эталонов классов в пространстве решений.

Для построения эталонов классов в пространстве решений формально необходимо построить кластерные структуры, разбив все объекты классифицированной обучающей выборки на группы, используя значения их признаков, рассматривая КОВ, как сплошной массив данных, ещѐ не разбитый предварительно на классы.

Если же КОВ представляет собой множество классов, состоящих из объектов признакового пространства, то для аттестации признаков предполагаем, что все признаки информативны – необходимо выполнить построение кластерных структур для объектов входящих в КОВ.

Аттестация признаков реализована посредством построения кластерных структур для заданной КОВ без учета отношения объектов к классу. Затем осуществляется проверка на предмет ошибочного включения объектов в классы исходной КОВ. Если количество ошибок не превышает заданного порогового значения, то признаки считаются пригодными для построения эталонов образов классов.

После выполнения этого этапа работа с КОВ и АСП одинакова для обоих вариантов и на основе КОВ $T=\{X_1^*,X_2^*,\ldots,X_k^*\}$ формируются эталоны классов $E=\{E_1,E_2,\ldots,E_k\}$, где эталон i-го класса $E_i^T=(e_{1i} \quad ,\ldots, \quad e_{pi})$ и $e_{ij}=\frac{1}{m_j}\sum_{j=1}^{m_j} x_{ij}^*$, где X_i^* ($i=\overline{1,k}$) – уточненные классы КОВ, после выполнения процедуры сепарирования априорного словаря признаков, p – количество признаков, попавших в уточненный словарь.

Сепарирование признаков априорного словаря производится на основе применением непараметрических критериев однородности. При отсутствии сдвига для очередного проверяемого признака по отношению ко всем остальным признакам, делается вывод о его пригодности для включения признака в уточненный словарь.

Для реализации процедуры обучения в автоматическом режиме разработана программа, которая реализована в виде многомодульного Windows-приложения. Модули выполнены по принципу плагинов, представленные в виде отдельных динамически подключаемых библиотек.

Существует несколько типов модулей которые реализуют алгоритмы вычисления метрики и межкластерных расстояний, которые в свою очередь используются в алгоритмах кластеризации, а так же модули проверки критериев однородности.

На данный момент реализованы алгоритмы кластеризации:

- FOREL [Загоруйко, 1999];
- FOREL-2 – модификация алгоритма FOREL, позволяющая строить заданное количество кластеров [Загоруйко, 1999];
- объектная кластеризация [Жукевич, 2010];
- равномерная кластеризация;
- равномерная кластеризация-2 – модификация алгоритма равномерной кластеризации, позволяющая строить заданное количество кластеров.

А так же модули с реализацией алгоритмов вычисления метрики (евклидова метрика, взвешенная евклидова метрика, метрика Махаланобиса, метрика Хэмминга); вычисления межкластерных расстояний (методы дальнего и ближнего соседа, метод центра тяжести, метод средних расстояний); критерии однородности (Е-критерий, критерий Хаги) [Кобзарь, 2006].

По мере написания новых алгоритмов кластеризации, признаков однородности, используемых метрик, методов расчета межкластерных расстояний, подключение их к программе происходит динамически.

Критерии аттестации АСП также реализованы в виде плагинов. Выбор нужных алгоритмов осуществляется посредством индексированных выпадающих списков (рисунок 1).

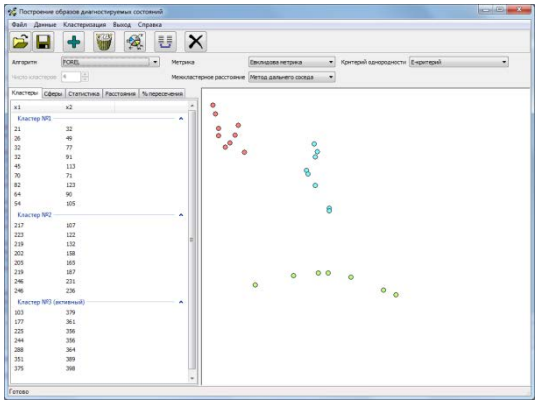

Рисунок 1 – Главное окно программы

Для случая размерности АСП равного двум (два признака) возможен интерактивный ввод данных с использованием манипулятора типа «мышь» или загрузка данных из файла в формате Microsoft Excel. В этом случае в правой рабочей области программы осуществляется наглядная визуализация результатов выполнения программы (рисунок 2).

Все имеющиеся алгоритмы кластеризации используют гипершары, из объединений которых строятся кластера. Поскольку визуализация предусмотрена только для случая двумерных векторов, то гипершары в этом случае представляют собой круги. Границы кругов изображены тонкими линиями. Контуры объединений кругов, принадлежащих одному кластеру, обведены более толстой линией. Области пересечения кластеров заштрихованы.

Каждому кластеру присваивается свой цвет, который генерируется автоматически. Границы кругов принадлежащих каждому кластеру, контуры кластера и точки, которые относятся к этому кластеру, рисуются одним и тем же цветом. Цвета подбираются автоматически так, чтобы они были как можно более хорошо различимы.

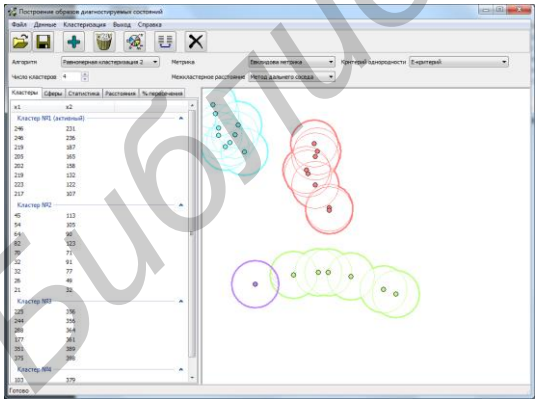

Рисунок 2 – Визуализация данных

На рисунке 2 представлены результаты кластеризации введенной КОВ для трех классов объектов. Для определения кластеров применили алгоритм равномерной кластеризации 2, с использованием евклидовой метрики $d_E\left(X_i, X_j\right) = \sqrt{\sum_{k=1}^{p}\left(x_i^{(k)} - x_j^{(k)}\right)^2}$, межкластерное расстояние вычисляется по методу дальнего соседа $\rho_{min} = \max_{X_i \in S_k, X_j \in S_m} d\left(X_i, X_j\right)$. Для проверки информативности признаков из АСП использовался непараметрический Е-критерий. На входе была загружена КОВ состоящая из 3-х классов, а на выходе получили 4-кластерную структуру, при этом два первых исходных класса образовали два отдельных кластера, а третий класс был разделен на два дополнительных класса. Результат работы алгоритмов построения кластерных структур существенно зависит от выбора используемой метрики.

Если объекты КОВ представляют собой векторы размерности более двух, то её загрузка в память, осуществляется только из файла в формате Microsoft Excel. При этом не проводится визуализация данных, но все характеристики построенных кластеров будут представлены на соответствующих закладках рабочего поля программы (рисунок 3).

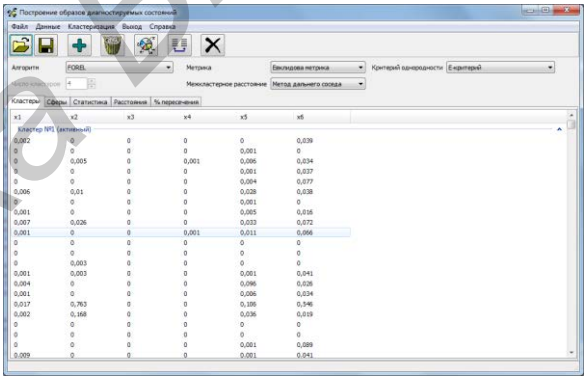

Рисунок 3 – Отображение данных

В остальном работа программы для векторов большей размерности ничем не отличается.

Результаты работы алгоритма построения кластеров и все их числовые характеристики можно сохранить в файл и в последующем, загрузив его, применить иной алгоритм кластеризации или иную метрику, алгоритм подсчета межкластерного расстояния, критерий однородности.

По внутренней структуре программный код реализации в полной мере использует архитектуру языка Object Pascal. В основе лежит обобщенная библиотека классов, которая должна быть одинакова и в главном модуле программы и в подключаемых библиотеках. Классы обобщенной библиотеки являются предками всех реализованных в программе классов, вызываемые методы которых в обобщенной библиотеке абстрактны и не имеют своей реализации, зато позволяют использовать принцип полиморфизма технологии объектно-ориентированного программирования.

2. Реализованные алгоритмы построения кластерных структур.

Опишем в общем виде реализованные в библиотеке алгоритмы построения кластерных структур.

2.1. Реализация алгоритма FOREL

Данный алгоритм кластеризации автоматически определяет количество формируемых кластеров.

В результате работы этого алгоритма получаются кластера сферической формы. Количество кластеров зависит от радиуса сфер. Чем меньше радиус, тем больше получается кластеров.

При реализации алгоритма создаѐтся список векторов, которые ещѐ не были добавлены в кластеры. Первоначально он равен переданному списку векторов для кластеризации. Создаѐтся другой список, который будет содержать точки, которые присутствовали в прошлой «плавающей» сфере.

Далее организован цикл, который останавливается, когда список неиспользованных векторов пуст. Внутри него вычисляется минимальная сфера, которая содержит все неиспользованные точки. Есть ещѐ внутренний цикл, который вычисляет точки внутри текущей внутренней сферы, находящейся внутри сферы минимального радиуса. Радиус внутренней сферы равен k1 – радиус минимальной сферы, где k1 задаѐтся свойством метрики и по умолчанию имеет значение 0,9. Центр внутренней сферы – центр тяжести внутренних точек, полученных на предыдущем шаге алгоритма. Внутренний цикл продолжается пока прошлый список внутренних точек сферы не будет равен списку точек, полученных на данном шаге. Когда внутренний цикл завершает работу, внутренние точки подсферы добавляются в новый кластер и сама эта подсфера становится новым кластером.

2.2. Реализация алгоритма FOREL 2

Алгоритм FOREL 2 является модификацией алгоритма FOREL. Он требует явного указания количества кластеров, которые необходимо сформировать. При реализации этот алгоритм несколько раз вызывает метод базового алгоритма FOREL.

Сначала коэффициент k1 устанавливается в его значение по умолчанию, т.е. 0,9. При этом значении, как правило, достигается минимальное количество кластеров. При необходимости этот коэффициент можно устанавливать в значение более близкое к единице (но меньше единицы).

Далее организован цикл, который останавливается, когда число кластеров, полученных алгоритмом FOREL оказывается больше, чем необходимо или коэффициент k1 окажется меньше заданного значения. На каждом шаге цикла коэффициент k1 уменьшается на заданный шаг. Минимальное значение на данный момент равно 0,01. Шаг также равен 0,01:

$$0,01 \le k_1 \le 0,9. \quad (1)$$

Результаты, полученные во время последнего вызова алгоритма FOREL, служат в качестве результата вычисления алгоритма FOREL 2.

2.3. Реализация алгоритма объектной кластеризации

Этот алгоритм создаѐт один кластер для каждого множества точек, принадлежащих одному классу. Кластеры, получаемые при помощи этого алгоритма, представляют собой объединения гипершаров.

Для формального представления образа класса в многомерном признаковом пространстве строим соответствующий кластер. С этой целью создаѐм список векторов, которые ещѐ не были добавлены в кластеры. Первоначально он равен переданному списку векторов для кластеризации. Создаѐтся новый кластер – единственный для этого класса, который генерируется этим алгоритмом кластеризации, так как данный алгоритм не является разделяющим.

Процесс построения кластера начинается с поиска наиболее удаленного от других вектора класса и добавляется в кластер. Дальше организуется главный цикл, который останавливается, когда список неиспользованных векторов становится пустым.

Ищется вектор, ближайший к выбранному. Этот вектор добавляется в кластер. Создаѐтся шар с центром в предыдущей добавленной точке. Радиус – минимум от половины предыдущего сохранѐнного расстояния и половины текущего расстояния. Кроме того, вычисляются и запоминаются значения координат вспомогательного вектора, который указывает на середину отрезка, соединяющего два очередных экземпляра класса, и добавляется шар с центром, расположенным между прошлой и текущей точкой. Его радиус равен половине расстояния между текущей и предыдущей точкой. Алгоритм останавливается, когда не осталось ни одного неиспользованного вектора. Кроме того добавляется последний шар радиуса, равного половине последнего сохранѐнного расстояния и с центром в последней добавленной точке.

2.4. Реализация алгоритма равномерной кластеризации

2.4.1. *Подробное описание алгоритма*

Для работы алгоритма требуется хотя бы две точки пространства R^n.

1. Находится точка, наиболее удалѐнная от всех остальных. Расстояние от неѐ до ближайшей точки обозначим M. Минимальное расстояние между точками обозначим m. Введѐм величину δ такую,

что $m \le \delta \le M$, например среднее между m и M. Все кластеры будут состоять из объединений гипершаров радиуса δ, далее просто «шаров».

2. Возьмѐм любые две точки из всех имеющихся, расстояние между которыми минимально.

3. Если расстояние между ними больше δ, то определим два новых кластера, в первый из них добавим шар с центром в первой точке, а во второй – шар с центром во второй точке. Эти две точки исключаются из рассмотрения. Переходим к пункту 6.

4. Если расстояние между двумя выбранными точками $\le \delta$, то определим новый кластер и добавим в него два шара с центрами в этих точках. Эти точки исключаем из рассмотрения

5. Все точки, которые попали в новый кластер, но ещѐ не исключены из рассмотрения, исключаем из рассмотрения. Добавляем в кластер шары с центрами в этих точках. Повторяем этот шаг, пока в новом кластере будут содержаться нерассмотренные точки.

6. Если все точки исключены из рассмотрения, то прекращаем работу алгоритма.

7. Если осталась только одна нерассмотренная точка, то определяем новый кластер, содержащий шар с центром в этой точке, и прекращаем работу алгоритма.

8. Если осталось хотя бы две точки, то переходим к пункту 2.

Изменяя δ, можно изменять количество кластеров. Чем меньше δ, тем больше в результате получается кластеров.

Как вариант, δ может быть взято как среднее арифметическое расстояний между всеми точками. Или M может быть вычислено как среднее арифметическое расстояний между всеми точками, а δ – по первоначальной формуле. Те или иные подходы могут быть более или менее удобными при работе с различными исходными данными.

2.4.2. *Реализация*

В начале алгоритма находится минимальное расстояние между всеми имеющимися точками. Для этого перебираются все возможные пары векторов и как только находится меньшее расстояние, чем были найдены до этого, это расстояние записывается в переменную и т.д.

Для нахождения максимально удалѐнного вектора от остальных перебираются все векторы. От каждого из них находится минимальное расстояние до всех остальных. Далее выбираем тот вектор, для которого это вычисленное значение максимально.

Вычисляется расстояние соседства в зависимости от коэффициента k1 и максимальной и минимальной величины. Вычисляется радиус будущих шаров. Создаѐтся список векторов, которые ещѐ не были добавлены в кластеры. Первоначально он равен переданному списку векторов для кластеризации. Организуется внешний цикл, который заканчивается тогда, когда список неиспользованных векторов становится пустым.

Создаѐтся новый кластер. Ищутся два ближайших вектора по аналогии с поиском минимального расстояния в начале алгоритма. Эти два вектора добавляются в кластер и удаляются из списка неиспользуемых векторов. В последний кластер добавляются шары стандартного размера с центрами в этих точках. Все точки из списка неиспользованных точек, которые не дальше расстояния соседства от точек, включенных в последний кластер, тоже добавляются в последний кластер. Аналогичным образом добавляются и шары.

Если таких двух точек не нашлось, то все оставшиеся точки добавляются в отдельные кластеры. Соответственно добавляется один шар стандартного размера в каждый из этих кластеров.

2.5. Реализация алгоритма равномерной кластеризации 2

Равномерная кластеризация 2 – это модификация алгоритма равномерной кластеризации, которая может строить заданное количество кластеров.

Этот алгоритм несколько раз вызывает метод базового алгоритма равномерной кластеризации в процессе выполнения основного бесконечного цикла.

После каждого вызова этого алгоритма сохраняется предыдущее количество полученных векторов. Предыдущее число кластеров и текущее число кластеров сравниваются и анализируются. Если текущее число кластеров меньше необходимого, а предыдущее – больше, тогда цикл прекращается и наоборот. Если число построенных кластеров равно требуемому, то цикл тоже прерывается. В остальных случаях, если число кластеров слишком большое или слишком малое, то коэффициент k1 алгоритма равномерной кластеризации корректируется в соответствующую сторону. Также цикл прекращается при достижении этим коэффициентом крайних значений.

Окончательные результаты получаются во время последнего вызова алгоритма равномерной кластеризации.

Заключение

При реализации систем распознавания для представления образов классов предлагается использовать кластерные структуры. Исходными данными для построения кластерных структур будут являться системы векторов в многомерном признаковом пространстве. Первоначальное описание этих векторов в математическом выражении представляет собой прямоугольную матрицу, с числом строк равным размерности

признакового пространства. Формальное же описание образов классов можно реализовать в виде кластерных структур, получаемых на основе применения различных алгоритмов кластеризации.

В статье рассмотрен ряд вспомогательных алгоритмов, на основе использования которых можно осуществлять представление формальных образов классов в виде кластерных структур и представлена программная реализация этих алгоритмов. Описаны основные принципы реализации на уровне алгоритмов и с точки зрения организации программного кода на основе применения технологии ООП. Программное приложение имеет многомодульную структуру и обеспечивает возможность динамического подключения новых реализаций алгоритмов кластеризации, вычисления метрик, критериев однородности для выполнения аттестации априорного словаря признаков.

Библиографический список

[Родченко, 2006] Родченко, В.Г. Об одном методе реализации процедуры обучения при построении системы распознавания образов / В.Г. Родченко. // Известия Гомельского государственного университета имени Ф.Скорины. - 2006. - №4. - С.73-76

[Родченко, 2012] Родченко, В.Г. Метод построения компьютерной системы диагностики на основе анализа данных обучающей выборки / В.Г. Родченко, Е.В. Олизарович, А.И. Жукевич // Искусственный интеллект. - 2012 .- №4. - С.381-386

[Загоруйко, 1999] Загоруйко, Н.Г. Прикладная методы анализа данных и знаний. / Н.Г. Загоруйко – Новосибирск: Изд-во Института математики СО РАН, 1999 – 264 с..

[Жукевич, 2010] Жукевич, А.И. Об одном методе построения формальных образов классов при реализации систем распознавания / А.И.Жукевич, В.Г.Родченко // Известия Гомельского государственного университета имени Ф.Скорины. – Гомель, 2010. – №5(62). – С.79-83.

[Кобзарь, 2006] Кобзарь, А.И. Прикладная математическая статистика. Для инженеров и научных работников. / А.И. Кобзарь – М.:ФИЗМАТЛИТ, 2006. – 816с.

SOFTWARE IMPLEMENTATION OF SUBSIDIARY ALGORITHMS FOR CONSTRUCTING CLUSTER STRUCTURES

Harunou V.A., Zhukevich A.I.,
Rodchenko V.G.

Yanka Kupala Grodno State University, Grodno, Republic of Belarus

v.gorunov@grsu.by

san@grsu.by

rovar@mail.ru

Cluster structures allow to formalize the procedure of presenting images and standards of classes in a multidimensional feature space when build recognition systems. Also it allows to automatically perform procedures of training and classification. The process of building cluster structures is associated with using of library of software procedures. The procedures implement the subsidiary clustering algorithms, algorithms to compute metrics and intercluster distance and homogeneity criteria.

Introduction

This article describes the software implementation of support algorithms for building of cluster structures. In an automatic mode the cluster structures allow to implement procedures for training and final decision-making based on an initial data of classified training sample. It is used to formal representation of images in the construction of class recognition systems.

Main Part

It is necessary to build cluster structures for construction of class standards in the solution space. It is necessary to divide all objects of classified training sample into groups using the values of their attributes.

If classified training set is a set of classes which consist of the feature space so we assume for certification that all features are informative and they need to build cluster structures for objects that included in the classified training set.

Certification features is implemented by constructing a cluster structures for a given classified training sample without a relationship between objects and class.

Then it is performed a check to find a mistaken inclusion of properties in the initial classes of classified training sample. If the number of errors does not exceed a threshold value, the signs are considered suitable for the construction of a master image classes.

There is a program to implement the training procedure in an automatically mode that is implemented as a modular Windows-application. Modules are executed as plugins that represented as separate dynamic link libraries.

There are several types of modules that implement algorithms of computation metrics and intercluster distances, clustering algorithms and checker modules of homogeneity criteria.

Conclusion

The article presents a set of auxiliary algorithms that can be used to implement of presentation of formal master image classes as a cluster structures. In addition, there is a software implementation of these algorithms. The article describes basic principles of implementation at the level of algorithms. The software application has a multimodule structure and provides the ability to dynamically connect new implementations of clustering algorithms, computing metrics and homogeneity criteria to perform certification of priori dictionary of features.