УДК 004.93

ОБРАБОТКА МИКРОЧИПОВЫХ ДАННЫХ С ПОМОЩЬЮ АЛГОРИТМА НЕЧЕТКОЙ КЛАСТЕРИЗАЦИИ

А. С. Тараскина[1], **Е. С. Черемушкин**[1]

Рассматривается программа, реализующая генетический алгоритм, основанный на генетических процессах в биологических организмах.

Ключевые слова: генетические алгоритмы, ДНК-микрочип, кластеризация, итерационные методы, дендрограммы.

Введение. Во многих областях биомедицинских исследований экспрессию генов изучают с помощью ДНК-микрочипов [1]. Для анализа растущего объема данных, полученных с помощью этой технологии, кластеризация становится практически необходимой [2]. Методы кластеризации [3, 4] делятся на иерархические и итерационные (методы разбиений).

Иерархические алгоритмы связаны с построением дендрограмм. В агломеративных алгоритмах перед началом кластеризации все объекты считаются отдельными кластерами, которые в ходе алгоритма объединяются. Вначале выбирается пара ближайших кластеров, которые объединяются в один кластер. В результате количество кластеров уменьшается на единицу. Процедура повторяется, пока все классы не объединятся. На любом этапе объединение можно прервать, получив нужное число кластеров. Однако процедура иерархического кластерного анализа хороша для малого числа объектов и не годится для данных большого объема из-за трудоемкости агломеративного алгоритма и слишком больших размеров дендрограмм.

В итерационных алгоритмах данные сразу разбиваются на несколько кластеров, число которых оценивается исходя из условий. Далее элементы перемещаются между кластерами так, чтобы был оптимизирован некоторый критерий, например, минимизируется изменчивость внутри кластеров [5].

Целью настоящей статьи является описание нового алгоритма кластеризации, находящего близкое к оптимальному решение задачи кластеризации данных микрочипов на основе нечеткого алгоритма c-средних, и реализующей его программы.

1. Алгоритм нечетких c-средних. Подробно алгоритм рассматривается в [6, 7]. Исходной информацией для кластеризации является $(l \times n)$-матрица наблюдений $X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{l1} & x_{l2} & \dots & x_{ln} \end{pmatrix}$, где l — число объектов, n — число признаков (наблюдений) для каждого объекта.

Задача кластеризации состоит в разбиении множества объектов на группы (кластеры) “похожих” между собой объектов. В n-мерном метрическом пространстве признаков мерой “сходства” двух объектов будем считать расстояние между ними.

В данной работе применяется метод нечеткой кластеризации, согласно которому каждый объект может принадлежать с различной степенью принадлежности нескольким или всем кластерам одновременно. Число кластеров c считается заранее известным.

Кластерная структура задается $(c \times l)$-матрицей принадлежности $M = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1l} \\ m_{21} & m_{22} & \dots & m_{2l} \\ \dots & \dots & \dots & \dots \\ m_{c1} & m_{c2} & \dots & m_{cl} \end{pmatrix}$, где m_{ij} — степень принадлежности j-го элемента i-му кластеру.

Отметим, что матрица принадлежности должна удовлетворять следующим условиям:

1. $m_{ij} \in [0, 1]$, $i = \overline{1, c}$, $j = \overline{1, l}$;
2. $\sum\limits_{i=1}^{c} m_{ij} = 1$, $j = \overline{1, l}$, т.е. каждый объект должен быть распределен между всеми кластерами;

[1]Институт систем информатики РАН (ИСИ), просп. акад. М. А. Лаврентьева, д. 6, 630090, г. Новосибирск; e-mail: anna@biorainbow.com, annet@land6.nsu.ru

3. $0 < \sum_{j=1}^{l} m_{ij} < l$, $i = \overline{1,c}$, т.е. ни один кластер не должен быть пустым или содержать все элементы.

Для оценки качества разбиения используется критерий разброса, показывающий сумму расстояний от объектов до центров кластеров с соответствующими степенями принадлежности:

$$J = \sum_{i=1}^{c} \sum_{j=1}^{l} (m_{ij})^w d(v_i, x_j),$$

где $d(v_i, x_j)$ — евклидово расстояние между j-м объектом $x_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ и i-м центром кластера $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$, $w \in (1, \infty)$ — экспоненциальный вес, определяющий нечеткость, размытость кластеров, $V = \begin{pmatrix} v_{11} & v_{12} & \ldots & v_{1n} \\ v_{21} & v_{22} & \ldots & v_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ v_{c1} & v_{c2} & \ldots & v_{cn} \end{pmatrix}$ — $(c \times n)$-матрица координат центров кластеров, элементы которой вычисляются по формуле

$$v_{ik} = \sum_{j=1}^{l} (m_{ij})^w x_{jk} \Big/ \sum_{j=1}^{l} (m_{ij})^w, \quad k = \overline{1,n}\ . \tag{1}$$

Задачей является нахождение матрицы M, минимизирующей критерий J. Для этого используется алгоритм нечетких c-средних, в основе которого лежит метод множителей Лагранжа. Этот алгоритм позволяет найти локальный оптимум, поэтому при повторных его использованиях могут быть получены разные результаты.

На первом шаге матрица принадлежностей M, удовлетворяющая условиям 1–3, генерируется случайным образом. Далее запускается итерационный процесс вычисления центров кластеров и пересчета элементов матрицы степеней принадлежности при $d_{ij} = d(v_i, x_j)$, $i = \overline{1,c}$, $j = \overline{1,l}$:

$$m_{ij} = \frac{1}{(d_{ij})^{2/(w-1)} \sum_{k=1}^{c} \frac{1}{(d_{kj})^{2/(w-1)}}} \quad \text{при} \quad d_{ij} > 0 \quad \text{и} \quad m_{kj} = \begin{cases} 1, & k = i, \\ 0, & k \neq i \end{cases} \quad \text{при} \quad d_{ij} = 0.$$

Вычисления продолжаются до тех пор, пока изменение матрицы M, характеризующееся величиной $||M - M^*||^2$, где M^* — матрица на предыдущей итерации, не станет меньше заранее заданного параметра останова ε. Сходимость алгоритма нечетких c-средних доказана в [8].

Остановимся на выборе значения экспоненциального веса w. Чем больше это значение, тем более размазанной становится матрица принадлежности; при $w \to \infty$ элементы примут вид $m_{ij} = 1/c$, что является неприемлемым решением, поскольку все объекты с одинаковой степенью принадлежности распределены по всем кластерам. Теоретически обоснованного правила выбора веса пока не существует (обычно выбирают $w = 2$).

2. Генетические алгоритмы. Локальный минимум, полученный с помощью алгоритма нечетких c-средних, зачастую отличается от глобального минимума. Поиск глобального минимума функционала J не осуществим из-за большого объема вычислений, однако существуют алгоритмы, вычисляющие решения, близкие к глобальному минимуму.

Нами был использован генетический алгоритм (ГА), основанный на генетических процессах в биологических организмах: биологические популяции развиваются в течение нескольких поколений, подчиняясь законам естественного отбора в соответствии с принципом “выживает наиболее приспособленный”. Наша программа, реализующая этот алгоритм, позволяет получать соответствующим образом закодированные решения, выбирая из них наиболее подходящее. Обычно ГА дают хорошие результаты для задач оптимизации многопараметрических функций, а именно такую задачу мы и решаем. Однако, как и другие методы эволюционных вычислений, они не гарантируют обнаружения глобального решения за полиномиальное время. ГА не гарантируют и того, что глобальное решение будет найдено, но они вполне пригодны для поиска “достаточно хорошего” решения задачи “достаточно быстро”.

ГА работает с совокупностью (популяцией) особей, которые представляют собой возможные решения данной задачи. Каждая особь оценивается степенью ее приспособленности, что соответствует тому, насколько “хорошо” данное решение задачи. Наиболее приспособленные особи могут скрещиваться и производить потомство. В результате получаются новые особи, сочетающие в себе “хорошие” характеристики, полученные от родителей. Возможность скрещивания менее приспособленных особей меньше, поэтому

признаки, которыми они обладали, будут элиминироваться из популяции в процессе эволюции. Итак, из поколения в поколение хорошие характеристики распространяются по всей популяции. Скрещивание наиболее приспособленных особей приводит к тому, что исследуются наиболее перспективные участки пространства поиска. В конечном итоге популяция будет сходиться к оптимальному решению задачи. Существует также возможность мутации особи, когда часть ее характеристик случайным образом изменяется. Благодаря этому можно выйти из состояния локального оптимума и получить новое возможное решение.

3. Описание программы.

3.1. Данные. Для обработки могут использоваться два типа данных: микрочипы (microarray) и матрицы расстояний.

Данные, полученные в результате экспериментов с микрочипами, можно представить в виде матрицы наблюдений, где в строках располагаются гены, а в столбцах — их уровни экспрессии в различных экспериментах.

В качестве расстояния между генами берется евклидово расстояние в n-мерном метрическом пространстве. Координаты центров кластеров находятся по формулам (1).

Если данные нормализованы (нулевой средний уровень экспрессии для каждого гена и единичное среднеквадратичное отклонение), то в результате кластеризации получаются группы генов со сходным профилем экспрессии. В противном случае в один кластер попадают гены с близкими значениями экспрессии на протяжении всех экспериментов.

Полученные матрицы расстояний между объектами можно использовать для кластеризации этих объектов. В этом случае в качестве исходных данных задается симметричная матрица для системы из l объектов: $D = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1l} \\ d_{21} & d_{22} & \dots & d_{2l} \\ \dots & \dots & \dots & \dots \\ d_{l1} & d_{l2} & \dots & d_{ll} \end{pmatrix}$, где $d_{ii} = 0$, $d_{ij} = d_{ji}$, $i, j = \overline{1, l}$, а в качестве расстояний берутся элементы этих матриц. Непосредственные наблюдения являются “скрытыми”. Центры кластеров в этом случае совпадают с некоторыми из заданных объектов. Координаты по методу c-средних не вычисляются, а новым центром j-го кластера объявляется k-я вершина, минимизирующая сумму

$$\sum_{i=0}^{l} m_{ji} d_{ki}. \tag{2}$$

3.2. Реализация. В программе используется комбинация описанных выше алгоритмов (c-средних и генетического) [9]. В качестве члена популяции для микрочипов берется массив координат центров кластеров, а для матриц расстояний — массив номеров элементов, выбранных в качестве центров.

Шаг 1. Случайным образом создается начальная популяция с заданным числом особей n. Для этого генерируются матрицы принадлежностей, а по ним определяются соответствующие особи (формулы (1) и (2)).

Шаг 2. К каждой особи применяется метод c-средних, пока изменения на каждой итерации не станут меньше заданного параметра.

Шаг 3. Выбирается некоторое количество “элитных” особей с наименьшими значениями критерия.

Шаг 4. Выполняется скрещивание. Методом рулетки (roulette-wheel selection) из популяции выбирается пара особей. Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер сектора пропорционален соответствующей приспособленности, т.е. обратно пропорционален значению критерия. При таком отборе члены популяции с более высокой приспособленностью будут выбираться с большей вероятность чаще, чем особи с низкой приспособленностью. После отбора для каждой пары с некоторой вероятностью происходит двухточечный кроссовер [10]. Случайным образом выбирается первая точка — целое число от 0 до cp_{cross}, где c — число кластеров, а p_{cross} — процент признаков, который потомок должен получить от одного из родителей. Вторая точка отстоит от первой на $c(1-p_{\text{cross}})$ позиций. Обе родительские структуры разделяются в этих точках. Затем соответствующие центральные сегменты меняются местами и вновь объединяются с концевыми. Получаются два генотипа потомков. Кроссовер может не произойти, тогда на следующую стадию переходят неизмененные особи. Элитные особи также переходят в новое поколение без изменений. Число пар рассчитывается так, чтобы в новом поколении было то же количество особей n.

Для наших типов данных сегмент соответствует некоторому числу подряд идущих строк в матрице координат центров или элементов массива номеров. Таким образом, при кроссовере частично изменяются центры кластеров, определяемых данной особью.

Шаг 5. Полученные на предыдущем шаге особи (за исключением элитных) с заданной вероятностью подвергаются мутации. При этом для каждой мутирующей особи случайным образом выбирается несколько элементов, значения которых изменяются на произвольные из интервала, определяемого условиями.

Шаг 6. Снова с помощью c-средних обрабатываются новые и мутировавшие особи.

Шаг 7. Из получившейся популяции элиминируются одинаковые организмы и вновь выбираются элитные.

Шаг 8. Переход на четвертый шаг. Число переходов, т.е. жизненных циклов, популяции задается заранее.

Шаг 9. Наиболее приспособленная особь объявляется искомым решением задачи.

4. Дополнительные функциональные возможности.

4.1. Выбор параметра w. В работе [11] установлено, что значение $w = 2$ не подходит для данных, полученных с микрочипов. В нашей программе используются экспериментально установленные в этой же работе формулы для вычисления более подходящего значения.

Как уже было отмечено, при больших значениях w степени принадлежности становятся близки к $1/c$. Можно оценить граничное значение w_{ub}. Очевидно, что значения m_{ij} зависят от расстояний между элементами и центрами кластеров. Центры кластеров близки к некоторым элементам (генам), поэтому можно предположить, что существует взаимосвязь результатов нечеткой кластеризации и коэффициента вариации cv для множества $Y_w = \left\{d(x_i, x_j)\right\}^{2/(w-1)}$, $i \neq j = \overline{1,l}$, где $cv = \sigma(Y_w)/\overline{Y}_w$. По результатам экспериментов для нахождения w_{ub} было предложено уравнение $cv(Y_w) \approx 0.03\,n$, где n — размерность данных. Численное решение этого уравнения находится методом дихотомии.

В итоге, значение параметра выбирается следующим образом: $w = 1 + w_0$, $w_0 = \begin{cases} 1, & w_{ub} \geqslant 10, \\ \dfrac{w_{ub}}{10}, & w_{ub} < 10. \end{cases}$

4.2. Силуэт. Для оценки качества кластеризации можно использовать величину силуэта [12]. Допустим, ген x_i лежит в кластере C_r. При нечеткой кластеризации номер кластера определяется по максимальному значению степени принадлежности. Вычисляются значения $a(x_i) = \dfrac{1}{|C_r|} \sum\limits_{x_j \in C_r} d(x_i, x_j)$ и $b(x_i) = \min\left\{ \dfrac{1}{|C_s|} \sum\limits_{x_j \in C_s} d(x_i, x_j),\ r \neq s = \overline{1,c} \right\}$. Силуэт гена определяется по формуле $s_(x_i) = \dfrac{a(x_i) - b(x_i)}{\max\left(a(x_i), b(x_i)\right)}$. Значение силуэта лежит в интервале $[-1; 1]$; если оно отрицательно, то ген считается плохо кластеризованным.

5. Входные данные и результаты. Данные для кластеризации считываются из файла, заданного пользователем. Тип данных (микрочипы, матрица расстояний) определяется самой программой. Столбцы отделены друг от друга символом табуляции. Файл не должен содержать пропущенных значений.

Микрочипы. Первая строка содержит названия всех столбцов. Каждая последующая — названия генов (один или более столбцов) и значения экспрессии для всех экспериментов.

Матрица расстояний. Первая строка: заголовок (строка без символов табуляции) и названия столбцов. Последующие строки: название, совпадающее с названием соответствующего столбца, и данные. Матрица значений симметрична, на диагонали — нули. Программа может работать также с матрицей сходства, преобразуя ее в матрицу расстояний.

Параметры алгоритма. Общие: число кластеров, параметр останова, определяющий точность вычислений, экспоненциальный вес.

Метод c-средних: число итераций–запусков программы со случайными начальными данными с выбором наилучшего результата.

Генетический алгоритм: число особей в популяции, число жизненных циклов популяции, число элитных особей, частота мутаций, вероятность кроссовера в процессе скрещивания, процентное соотношение признаков в кроссовере.

Результаты кластеризации частично выводятся в окне программы в виде списка элементов по кластерам со степенью принадлежности выше порогового значения, которое можно изменять.

Сохраненный файл с результатами содержит: параметры алгоритма, список генов по кластерам со степенью принадлежности выше порогового значения, матрицу принадлежностей, координаты центров кластеров, значения силуэтов, если они были вычислены пользователем.

6. Сравнительный анализ результатов. Тесты на искусственных наборах данных показали хорошую точность данного метода. Работа алгоритма была проверена на наборах данных, полученных в экспериментах по изучению клеточного цикла, которые можно найти на сайте [13].

Взяв нормализованные значения экспрессии для генов, участвующих в регуляции клеточного цикла, которые измерялись с периодичностью в один час, мы провели разбиение генов на пять кластеров по числу стадий клеточного цикла. Для каждого гена соответствующая стадия, а значит и кластер, были предсказаны с помощью алгоритма иерархической кластеризации [14]. Если предположить, что подобное предсказанное распределение является точным, то отношение максимального числа генов одной стадии, попавших в один кластер, к общему числу генов данной стадии характеризует точность кластеризации с помощью нашего алгоритма. Для некоторых стадий результаты согласовывались с достаточно высокой точностью (около 80 %), а для некоторых — нет. Одной из причин может быть сходство профилей экспрессии у генов близких стадий; вполне вероятно, что предварительное распределение иерархическим алгоритмом отличается от истинного.

Еще одно сравнение мы провели для генов, стадии активности которых были определены и описаны в различных работах по изучению клеточного цикла методами, отличными от кластеризации. Полученные результаты в среднем были лучше, чем в предыдущем примере.

Заключение. Нечеткая кластеризация по методу c-средних — это удобный подход для выделения генов, тесно связанных с заданными кластерами. Применяя его в комбинации с генетическим алгоритмом, можно найти решение задачи кластеризации, близкое к оптимальному.

Нами была составлена программа для кластеризации, в которой реализованы вышеизложенные методы. Программа допускает возможность автоматического определения значения параметра размытости кластеров, подходящего для конкретного типа данных, и оценки качества кластеризации. С помощью программы можно осуществить разбиение объектов на группы, зная только попарные расстояния между ними, без информации о координатном представлении этих объектов.

Программа реализована в среде Microsoft Visual Studio (http://biorainbow.com/fuzzyclustering/).

СПИСОК ЛИТЕРАТУРЫ

1. *Lockhart D.J., Dong H., Byrne M.C., Follettie M.T., Gallo M.V., Chee M.S., Mittmann M., Wang C., Kobayashi M., Horton H., Brown E.L.* Expression monitoring by hybridization to high-density oligonucleotide arrays // Nat. Biotechnol. 1996. **14**. 1675–1680.
2. *Golub T.R.* Molecular classification of cancer: class discovery and class prediction by gene expression monitoring // Science. 1999. **286(5439)**. 531–537.
3. *Anderberg M.R.* Cluster analysis for applications. NY: Academic Press, 1976.
4. *Hartigan J.* Clustering algorithms. NY: Wiley, 1975.
5. http://www.statsoft.ru/home/textbook/modules/stcluan.html
6. *Штовба С.Д.* Введение в теорию нечетких множеств и нечеткую логику (http://matlab.exponenta.ru/fuzzylogic/book1/index.php).
7. *Hoppner F., Klawonn F., Kruse R., Runkler T.* Fuzzy cluster analysis. NY: Wiley, 1999.
8. *Bezdek J.C.* Pattern recognition with fuzzy objective functional algorithms. NY: Plenum Press, 1981.
9. *Hall L.O., Ozyurt B., Bezdek J.C.* The case for Genetic Algorithms in Fuzzy Clustering // Proc. 98 IPMU. Paris, 1998. 288–295.
10. *Goldberg D.E.* Genetic algorithms in search, optimization, and machine learning. Reading: Addison-Wesley, 1989.
11. *Dembele D., Kastner P.* C-means method for clustering microarray data // Bioinformatics. 2003. **19(8)**. 973–980.
12. *Rousseeuw J.P.* Silhouettes: a graphical aid to the interpretation and validation of cluster analysis // J. Comp. Appl. Math. 1987. **20**. 53–65.
13. http://genome-www.stanford.edu/Human-CellCycle/Hela/
14. *Whitfield M.L., Sherlock G., Saldanha A.J., Murray J.I., Ball C.A., Alexander K.E., Matese J.C., Perou C.M., Hurt M.M., Brown P.O., et al.* Identification of genes periodically expressed in the human cell cycle and their expression in tumors // Mol. Biol. Cell. 2002. **13**. 1977–2000.

Поступила в редакцию
30.11.2005