А. С. Тараскина

НЕЧЕТКАЯ КЛАСТЕРИЗАЦИЯ ПО МОДИФИЦИРОВАННОМУ МЕТОДУ *C*-СРЕДНИХ И ЕЕ ПРИМЕНЕНИЕ ДЛЯ ОБРАБОТКИ МИКРОЧИПОВЫХ ДАННЫХ

ВВЕДЕНИЕ

Во многих областях биомедицинских исследований экспрессию генов изучают с помощью ДНК-микрочипов [1]. Для анализа растущего объёма 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 1. 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 объёма 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 данной работы явилась разработка на основе нечеткого алгоритма *c*-средних нового алгоритма кластеризации, находящего близкое к оптимальному решение задачи кластеризации данных микрочипов.

1. АЛГОРИТМ НЕЧЁТКИХ *C*-СРЕДНИХ

Исходной информацией для кластеризации является матрица наблюдений $l \times n$

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ & & \dots & \\ x_{l1} & x_{l2} & \dots & x_{\ln} \end{bmatrix},$$

где l — число объектов, n — число признаков (наблюдений) для каждого объекта [6, 7].

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 нечёткой кластеризации, позволяющий каждому объекту принадлежать с различной степенью нескольким или всем кластерам одновременно. Число кластеров c считается заранее известным.

Кластерная структура задаётся матрицей принадлежности ($c \times l$ матрица):

$$M = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1l} \\ m_{21} & m_{22} & \dots & m_{2l} \\ & & \dots & \\ m_{c1} & m_{c2} & \dots & m_{cl} \end{bmatrix},$$

где m_{ij} — степень принадлежности j-го элемента i-му кластеру.

Отметим, что матрица принадлежности должна удовлетворять следующим условиям:

0) $m_{ij} \in [0,1], i = \overline{1,c}, j = \overline{1,l}$,

1) $\sum_{i=1}^{c} m_{ij} = 1, j = \overline{1,l}$, т.е. каждый объект должен быть распределён между всеми кластерами,

2) $0 < \sum_{j=1}^{l} m_{ij} < l, i = \overline{1,c}$, 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:

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

$d(v_i, x_j)$ — Евклидово расстояние между *j*-м объектом $x_j = (x_{j1}, x_{j2}, ..., x_{jn})$ и *i*-м центром кластера $v_i = (v_{i1}, v_{i2}, ..., v_{in})$,

$w \in (1, \infty)$ — экспоненциальный вес, определяющий нечёткость, размытость кластеров,

$V = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ & & \dots & \\ x_{c1} & x_{c2} & \dots & v_{cn} \end{bmatrix}$ — $c \times n$ матрица координат центров кластеров, элементы которой вычисляются по формуле $v_{ik} = \dfrac{\sum_{j=1}^{l} (m_{ij})^w x_{jk}}{\sum_{j=1}^{l} (m_{ij})^w}, k = \overline{1,n}$ (v).

Задачей является нахождение матрицы *M*, минимизирующей критерий *J*. Для этого используется алгоритм нечётких *c*-средних, в основе которого лежит метод множителей Лагранжа. Он позволяет найти локальный оптимум, поэтому для различных запусков могут получиться разные результаты.

На первом шаге матрица принадлежностей *M*, удовлетворяющая условиям 0)–2), генерируется случайным образом. Далее запускается итерационный процесс вычисления центров кластеров и пересчёта элементов матрицы степеней принадлежности:

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

где $d_{ij} = d(v_i, x_j)$ для $i = \overline{1,c}, j = \overline{1,l}$.

Вычисления продолжаются до тех пор, пока изменение матрицы M, характеризующееся величиной $\left\| M - M^* \right\|^2$, где M^* — матрица на предыдущей итерации, не станет меньше заранее заданного параметра остановки ε.

Сходимость алгоритма нечётких c-средних доказана в [8].

Остановимся на выборе значения w — экспоненциального веса. Чем больше это значение, тем матрица принадлежности более размазанная и при $w \to \infty$ элементы примут вид $m_{ij} = \frac{1}{c}$, что является плохим решением, т.к. все объекты с одинаковой степенью распределены по всем кластерам. Теоретически обоснованного правила выбора веса пока не существует, и обычно устанавливают $w = 2$.

2. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Локальный минимум, полученный с помощью алгоритма нечётких c-средних, зачастую отличается от глобального минимума. Поиск глобального минимума функционала J не осуществим ввиду большого объема вычислений, но существуют алгоритмы, получающие решение, близкое к глобальному минимуму.

Нами был использован генетический алгоритм (ГА), основанный на генетических процессах биологических организмов: биологические популяции развиваются в течение нескольких поколений, подчиняясь законам естественного отбора и по принципу «выживает наиболее приспособленный». Программа также может развивать 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 гарантирует обнаружения глобального решения за полиномиальное время. ГА не гарантируют и того, что глобальное решение будет найдено, но они хороши для поиска «достаточно хорошего» решения задачи «достаточно быстро».

ГА работает с популяцией — совокупностью особей, которые представляют собой возможные решения данной задачи. Каждая особь оценивается степенью её 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. Также существует возможность мутации особи, когда часть её характеристик случайным образом изменяется. Благодаря этому можно выйти из состояния локального оптимума, получить новое возможное решение.

3. ОПИСАНИЕ ПРОГРАММЫ

3.1. Данные

Для обработки могут использоваться два типа данных.

1. Микрочипы (microarray).

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

В качестве расстояния между генами берётся Евклидово расстояние в n-мерном метрическом пространстве. Координаты центров кластеров находятся по формулам (v).

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.

2. Матрицы расстояний.

Полученные некоторым образом матрицы расстояний между объектами можно использовать для кластеризации этих объектов. В этом случае в качестве исходных данных имеется симметричная матрица для системы из l объектов:

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1l} \\ d_{21} & d_{22} & \dots & \\ & & \dots & \\ d_{l1} & d_{l2} & \dots & d_{ll} \end{bmatrix}, \text{ где } d_{ii} = 0, d_{ij} = d_{ji}, i, j = \overline{1,l}.$$

Естественно, что 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 $\sum_{i=0}^{l} m_{ji} d_{ki} \quad (v_d)$.

3.2. Реализация

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

Шаг 1. Случайным образом создаётся начальная популяция с заданным числом особей *n*.

Для этого генерируются матрицы принадлежностей, а по ним определяются соответствующие особи (формулы (v) и (v_d)).

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. Производится 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 от-

боре члены популяции с более высокой приспособленностью с большей вероятностью будут выбираться чаще, чем особи с низкой приспособленностью. После отбора для каждой пары с некоторой вероятностью происходит двухточечный кроссовер [9]. Случайным образом выбирается первая точка — целое число от 0 до $c \cdot p_{cross}$, где c — число кластеров, а p_{cross} — процент признаков, который потомок должен получить от одного из родителей. Вторая точка отстоит от первой на $c(1-p_{cross})$ 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, определяемые данной особью.

Шаг 5. Мутация.

Все особи, полученные на предыдущем шаге, за исключением элитных, подвергаются мутации. С некоторой вероятностью случайное число элементов особи меняется на произвольные (разумеется, в границах, определяемых 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 4 шаг. Число переходов, т.е. жизненных циклов популяции, задаётся заранее.

Шаг 9. Наиболее приспособленная особь объявляется искомым решением задачи.

4. ДОПОЛНИТЕЛЬНЫЕ ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ.

4.1. Выбор параметра w

В работе [10] установлено, что значение $w = 2$ не подходит для данных, полученных с микрочипов. В нашей программе используются эксперимен-

тально установленные формулы для вычисления более подходящего значения, приведённые в вышеупомянутой работе.

Как было уже отмечено, при больших значениях w степени принадлежности становятся близки к $\frac{1}{c}$. Можно таким образом оценить граничное значение w_{ub}. Очевидно, 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 нечёткой кластеризации и коэффициента вариации cv для множества $Y_w = \left\{ d(x_i, x_j) \right\}^{\frac{2}{w-1}}, i \neq j = \overline{1,l}$, где $cv = \frac{\sigma(Y_w)}{\overline{Y_w}}$. По результатам экспериментов для нахождения w_{ub} было предложено уравнение $cv(Y_w) \approx 0.03n$, где n — размерность данных. Численное решение этого уравнения находится методом дихотомии.

В итоге, значение параметра выбирается следующим образом:

$$w = 1 + w_0, w_0 = \begin{cases} 1, w_{ub} \geq 10 \\ \frac{w_{ub}}{10}, w_{ub} < 10 \end{cases}.$$

4.2. Силуэт

Для оценки качества кластеризации можно использовать величину силуэта [11]. Допустим, ген x_i лежит в кластере C_r. При нечёткой кластеризации номер кластера определяется по максимальному значению степени принадлежности. Вычисляются значения $a(x_i) = \frac{1}{|C_r|} \sum_{x_j \in C_r} d(x_i, x_j)$ и $b(x_i) = \min \left\{ \frac{1}{|C_s|} \sum_{x_j \in C_s} d(x_i, x_j), r \neq s = \overline{1,c} \right\}$. Силуэт гена определяется как $s(x_i) = \frac{a(x_i) - b(x_i)}{\max(a(x_i), b(x_i))}$. Значение силуэта лежит в интервале $[-1; 1]$, если оно отрицательно, то ген считается плохо кластеризованным.

4.3. Входные данные и результаты

Данные для кластеризации считываются из файла, выбранного 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 также может работать с матрицей сходства, преобразуя её в матрицу расстояний.

Параметры алгоритма.

Общие:

– число кластеров,

– параметр остановки, 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.

Сохранённый файл с результатами содержит:

– параметры алгоритма,

– список генов по кластерам со степенью принадлежности выше $\frac{1}{c}$,

– матрицу принадлежностей,

– координаты центров кластеров,

– значения силуэтов, если они были вычислены пользователем.

5. ТЕСТОВЫЙ ПРИМЕР И СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕЗУЛЬТАТОВ

Работа алгоритма была проверена на наборах данных, полученных в экспериментах по изучению клеточного цикла, которые можно найти на сайте [12].

Для кластеризации взяты нормализованные значения экспрессии для генов, участвующих в регуляции клеточного цикла, которые измерялись с периодичностью в 1 час. Мы провели разбиение генов на 5 кластеров по числу стадий клеточного цикла. Для каждого гена соответствующая стадия, а значит и кластер, были предсказаны с помощью алгоритма иерархической кластеризации [13]. Если предположить, что подобное предсказанное распределение точно, то отношение максимального числа генов одной стадии, попавших в один кластер к общему числу генов данной стадии, характеризует точность кластеризации с помощью нашего алгоритма. Результаты для двух наборов данных (47 и 27 измерений для каждого гена) представлены ниже:

Набор данных	Стадии	Номера кластеров					Отношение
		1	2	3	4	5	
1	G1/S	13	0	0	186	4	0.916256
	S	134	1	0	74	0	0.641148
	G2	33	128	51	0	20	0.551724
	G2/M	3	99	64	0	102	0.380597
	M/G1	0	8	2	1	175	0.94086
2	G1/S	6	1	0	118	23	0.797297

	S	1	12	0	29	114	0.730769
	G2	19	70	39	4	29	0.434783
	G2/M	68	80	58	2	3	0.379147
	M/G1	6	1	0	118	23	0.723684

Видно, что для некоторых стадий результаты согласуются с достаточно высокой точностью, а для некоторых – нет. Одной из причин может быть сходство профилей экспрессии у генов близких стадий (G2, G2/M). Также вполне вероятно, что предварительное распределение иерархическим алгоритмом отличается от истинного.

Ещё одно сравнение мы провели для генов, стадии активности которых были определены и описаны в различных работах по изучению клеточного цикла методами, отличными от кластеризации. Результаты представлены в таблице.

Набор данных	Стадии	Номера кластеров				Отношение
		1	2	3	4	
1	G1/S + G1	16	0	2	0	0.888889
	S	7	17	0	0	0.708333
	G2	0	0	1	5	0.833333
	G2/M	0	1	12	10	0.521739
2	G1/S + G1	0	12	2	2	0.75
	S	1	5	14	0	0.7
	G2	5	0	0	1	0.833333
	G2/M	2	0	0	13	0.866667

ЗАКЛЮЧЕНИЕ

Нечёткая 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 написана программа для кластеризации, в которой реализованы вышеизложенные методы. Дополнительно в программе присутствует возможность автоматического определения значения параметра размытости кластеров, подходящего для конкретного типа данных, и оценки качества кластеризации.

Также с помощью нашей программы можно осуществить разбиение объектов на группы, зная только попарные расстояния между ними, не задумываясь о координатном представлении этих объектов.

Программа реализована в среде Microsoft Visual Studio и доступна по адресу: http://biorainbow.com/fuzzyclustering/

СПИСОК ЛИТЕРАТУРЫ

1. Lockhart D. J. et al. Expression monitoring by hybridization to high-density oligonucleotide arrays // Nat. Biotechnol. — 1996. — Vol. 14. — P. 1675–1680.
2. Golub T.R. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring // Science. — 1999. — Vol. 286 (5439) — P. 531–537.
3. Anderberg, M. R. Cluster Analysis for Applications. Academic Press, New York, NY, 1976
4. Hartigan J. Clustering Algorithms. Wiley, New York, NY, 1975.
5. http://www.statsoft.ru/home/textbook/modules/stcluan.html
6. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику, гл.12.
7. Höppner F., Klawonn F., Kruse R., Runkler T. Fuzzy Cluster Analysis, Wiley,1999.
8. Bezdek, J.C. Pattern Recognition With Fuzzy Objective Functional Algorithms. Plenum Press, New York, 1981.
9. Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Mass., 1989.
10. Dembele D., Kastner P. C-means method for clustering microarray data // Bioinformatics. — 2003. — Vol. 19(8). — P. 973–980.
11. Rousseeuw J.P. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis // J. Comp. Appl. Math. — 1987. — Vol. 20 — P. 53–65.
12. http://genome-www.stanford.edu/Human-CellCycle/Hela/
13. Whitfield M. L. et al. Identification of Genes Periodically Expressed in the Human Cell Cycle and Their Expression in Tumors // Mol. Biol. Cell. — 2002. — Vol. 13 — P. 1977–2000.