УДК 004.942

Анализ и прогнозирование возникновения иерархических кластеров посредством самоорганизующихся фрактоидов

А.С.Семенов

Аннотация. Для реализации подсистемы планирования операциями космического аппарата, находящегося на значительном удалении от земли вводится метод анализа и прогнозирования информации на основе иерархических кластеров и модели поведения аппарата на основе самоподобных (фрактальных) шаблонов. Это позволяет существенно увеличить скорость обработки информации. Приводится алгоритм иерархической кластеризации, синхронизированный с моделью поведения космического аппарата. Обобщением модели поведения является самоорганизующаяся модель, названная фрактоид.

Ключевые слова: планирование, иерархический кластер; фрактальный шаблон; самоорганизующийся фрактоид.

1. Введение. Состояние проблемы и постановка задачи

Автономное планирование и составление расписаний, работающего на удалении в сотни миллионов километров от Земли космического аппарата, требует создания бортовой автономной программной системы планирования. Такая система, предназначенная для управления процессами составления расписания операций

космического аппарата, должна вырабатывать планы на основе целей высокого уровня, задаваемых с Земли, а также контролировать работу космического аппарата в ходе выполнения планов: обнаруживать, диагностировать и устранять неполадки по мере их возникновения.

Методы управления по данным, использующие технологии Data Mining [1] позволяют построить модель поведения системы автоматически, на основе имеющихся данных о нормальном поведении системы и позволяют обрабатывать телеметрические данные, поступающие от работающей системы, в режиме реального времени.

Технологии Data Mining решают задачи автоматизации поиска знаний посредством их интеллектуального анализа, с целью поиска скрытых закономерностей в больших, необработанных объемах. Процесс интеллектуального анализа данных включает, помимо поиска закономерностей, этапы сбора, подготовки данных и последующего анализа полученных результатов. К настоящему времени разработано множество алгоритмов и технологий.

Анализ телеметрических данных космического аппарата в реальном времени служит: для быстрого диагностирования появления аномалии в данных [2,3]; для отслеживания возникновения трендов в данных; для диагностирования появления тонких различий в поведении системы, являющихся ранними предвестниками возникновения проблем. Для проведения анализа может быть использована кластеризация данных в сочетании с моделью поведения представленной в виде фрактальных шаблонов.

Под кластеризацией будем понимать процесс организации объектов в различные группы, называемые кластерами, по схожим признакам [4].

Результат иерархической кластеризации – древообразная структура, называемая дендрограммой. Кластеризация рассматривается как обучение без учителя, что отличает ее от классификации (обучения с учителем). Кластеризация – разбиение множества объектов, например, документов на подмножества – кластеры. Количество кластеров может быть произвольным или фиксированным. Кластеры изначально не задаются, и

даже может быть неизвестно само множество кластеров. В первом разделе рассмотрен алгоритм иерархической (агломеративной) кластеризации. Во втором разделе приведен алгоритм фрактального построения древовидных структур по шаблону [5,6], что может служить для определения аномалий в модели поведения космического аппарата.

В третьем разделе показываются два метода анализа и прогнозирования кластеров данных посредством шаблона и функции пригодности. Обобщением является формирование модели поведения космическим аппаратом на основе самоорганизующегося фрактоида.

2. Алгоритм иерархической кластеризации

Алгоритм иерархической кластеризации состоит в последовательном объединении кластеров объектов, сначала самых близких, а затем все более отдаленных друг от друга [4,7]. Разные алгоритмы имеют разный результат, зависящий от способа вычисления расстояния d между кластерами x_i и x_j см. таб.1.

Таблица 1. Способы вычисления расстояния между кластерами

Способ	**Расстояние между двумя кластерами**
Минимальное расстояние между объектами (single-link)	$\min d(x_i,x_j)$
Максимальное расстояние между объектами (complete-link)	$\max d(x_i,x_j)$

Алгоритм кластеризации, заданный, например, способом по минимальному расстоянию, может быть адаптирован к другим способам.

Алгоритм 1. Пример кластеризации (по минимальному расстоянию):

Инициализация. L(k), k = 0 – уровень k-ой кластеризации

m = 0 – кластер с последовательностью чисел 0,1,....., (n-1)

1. Построить матрицу расстояний D = [d(i,j)], N*N, где N – количество объектов, число в i-том ряде j-той колонке – расстояние d между i-тым и j-тым объектами, см. таб. 2.

Таблица 2. Матрица расстояний, N = 6.

	a	b	c	d	e	f
a	0	10	14	3	7	15
b	10	0	5	8	4	6
c	14	5	0	12	9	1
d	3	8	12	0	2	13
e	7	4	9	2	0	11
f	15	6	1	13	11	0

2. Найти min по всем парам {r} и {s} в текущей кластеризации:
   d[{r},{s}] = min d[i,j],

   где d[{r},{s}] - расстояние между кластерами {r} и {s}.

   Например, d[{c},{f}] = min d[3,6] = 1

3. Определить каждый объект в отдельный кластер. Увеличить номер последовательности: m = m +1. Слить кластер {r} и {s} в единственный кластер, чтобы сформировать следующую кластеризацию m. Установить уровень этой кластеризации:

   L(m) = d[{r},{s}] Например, L(1) = d[{c},{f}]

4. Обновить матрицу расстояний D, удалив ряды {s} и колонки {s} кластеров и добавить ряд и колонку, соответствующую новому сформированному кластеру. Удаляем ряд {f} и колонку {f}

	a	b	{c,f}	d	E
a	0	10	14	3	7
b	10	0	5	8	4
{c,f}	14	5	0	12	9
d	3	8	12	0	2
e	7	4	9	2	0

Расстояние между новым кластером обозначается, как {r,s} и старым

кластером {k}, определяется как:

d[{k}, {r,s}] = min d[{k},{r}], d[{k},{s}]

Например:

d[{k}, {c,f}] = min d[{k},{c}], d[{k},{f}]

Процесс повторяется до тех пор, пока все N объектов не попадут в один кластер, содержащий все объекты, если это не так, то перейти к шагу 2.

Рассмотрим итерации формирования матрицы расстояний D на шаге 4.

При L(2) = d[{e},{d}] = 2

	{a,d,e}	b	{c,f}
{a,d,e}	0	10	14
b	10	0	5
{c,f}	14	5	0

При L(3) = d[{a},{d,e}] = 3

	a	b	{c,f}	{d,e}
a	0	10	14	3
b	10	0	5	8
{c,f}	14	5	0	12
{d,e}	3	8	12	0

При L(4) = d[{b},{c,f}] = 5

	{a,d,e}	{b,c,f}
{a,d,e}	0	10
{b,c,f}	10	0

Критерии кластеризации: монотонность, редуктивность, растяжение и сжатие оказывают существенное влияние на результат.

Монотонность. Функция расстояния $d(x_i,x_j)$ называется монотонной, если при каждом слиянии расстояние между объединяемыми кластерами увеличивается: $d_1 \leq d_2 \leq \ldots \leq d_n$, где d_n - расстояние между ближайшими кластерами, выбранными на n-ом шаге для слияния.

Редуктивность заключается в переборе лишь наиболее близких пар. Задаётся параметр, сокращающий перебор множества пар. Когда все такие пары будут исчерпаны, параметр увеличивается, и так далее, до полного слияния всех объектов в один кластер.

Растяжение и сжатие. По мере роста кластера, расстояние от него до других кластеров определяются через отношение $d_n/d(X, Y)$, где d_n - расстояние между ближайшими кластерами, объединяемыми на n-м шаге, X и Y - центры этих кластеров.

Если это отношение на каждом шаге больше единицы, то расстояние является растягивающим; если оно всегда меньше единицы, то сжимающим, иначе сохраняется метрика пространства. Растяжение способствует более чёткому отделению кластеров. Расстояние ближнего соседа является сильно сжимающим.

Рассмотрим визуализацию результатов кластеризации способом дендрограммы. Вид дендрограммы зависит от алгоритма кластеризации и выбора способа вычисления расстояния между объектами и кластерами. Для данной дендрограммы существуют различные способы построения соответствующих деревьев.

Если кластеризация обладает свойством монотонности, то дендрограмма строится без пересечений, а любой кластер представляется последовательностью кружков на вертикальной оси. Если процесс кластеризации не монотонен, то дендрограммы представляет собой клубок пересекающихся линий. По вертикальной оси откладываются расстояния d_n, по горизонтальной объекты. На рис. 1. приведена дендрограмма для рассматриваемого примера.

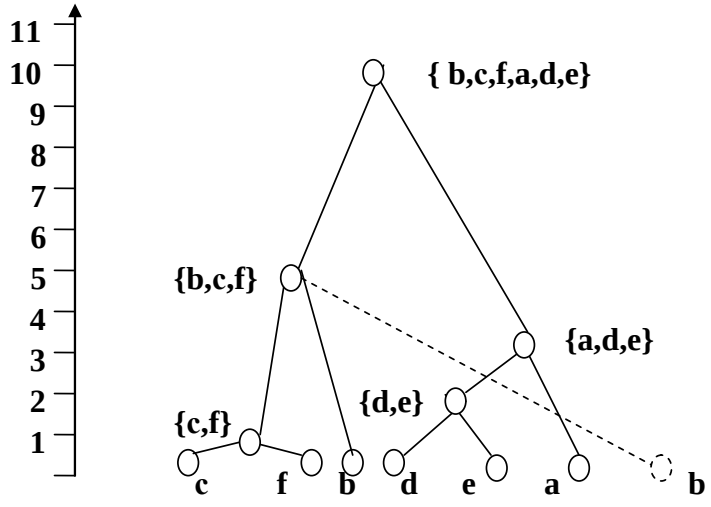

Рис. 1. Дендрограмма

Рассмотрим достоинства и недостатки алгоритма 1.

Достоинства:

1. Не нуждается в обучении.

2. Использование матрицы расстояний между объектами.

3. Инкрементное вычисление результата.

Недостатки:

1. Кластеризация совершается ступенчато, то есть два близко расположенных объекта объединяются и рассматриваются как один кластер. Это приводит к тому, что число объектов уменьшается и становится равным n - 1, то есть один кластер будет содержать два объекта, а n - 2 по одному.

2. Алгоритм недетерминирован, так как зависит от последовательности просмотра таблицы, поэтому расстояние между парами объектов в матрице D желательно упорядочивать. Например, при построении дендрограммы (дерево строится слева направо и затем снизу вверх), обьект {b} соединяется с кластером {b,c,f} и граф становится не планарным, то есть в нем есть пересечения между дугами (на рис. 1 дуга помечена пунктирной линией). Для того чтобы сделать граф планарным, надо объект {b} переместить в соответствующую позицию, что требует дополнительных вычислений. Если воспользоваться стратегией построения дендрограммы сверху вниз, то это потребует дополнительного пересчета матрицы D.

3. Для получения хорошего результата кластеризации необходимо: задавать порог – максимальное количество объектов в кластере.

4. Объектам жестко приписывается кластер, кластеры не пересекаются и нет возврата к предыдущему результату итерации.

5. Для поиска локального оптимального семейства объектов требуются локальные операции над деревьями.

6. Плохая масштабируемость при увеличении числа объектов n: наименьшая вычислительная сложность $O(n^2)$.

3. Алгоритм построения древовидной структуры по шаблону

В рассмотренном алгоритме 1 кластеризация осуществлялась как результат слияния двух кластеров (вершин) в кластер. Предлагается задавать кластеры в виде готовых шаблонов и из этих же шаблонов строить кластеры более высокого уровня, аналогично фрактальной структуре [5].

Шаблон $\mathcal{T}$ бинарного дерева будем определять как:

$\mathcal{T} = \alpha + \rho$, где (1)

α – левая цепочка символов, метка подкластера левого поддерева;

ρ – правая цепочка символов, метка подкластера правого поддерева;

+ – бинарный оператор конкатенации цепочек α и ρ. Результатом выполнения оператора + конкатенации цепочек является метка кластера.

P – множество порождающих правил, в соответствии с которыми на каждом шаге одновременно в цепочку нетерминальных символов подставляются нетерминальные символы, аналогично L-системе [5]. Пример правила: $\mathcal{P} = \{\mathcal{C} \rightarrow \mathcal{CG}, \mathcal{G} \rightarrow \mathcal{CG}\}$.

Цепочка символов $\mathcal{T}$, порожденная на шаге n, включает цепочку символов, полученную на предыдущем шаге n -1.

Перенастраиваемый алгоритм, названный $\mathcal{TL}$-система, связывает символьную динамику шаблона $\mathcal{T}$ с построением древовидной структуры [5].

Определение 1. $\mathcal{TL}$-системой $\mathcal{TL} = \{G, \bar{A}\}$ называется алгоритм генерации сложных фрактальных объектов, управляемый цепочкой символов, порождаемой по шаблону на основе задаваемых грамматических правил, где

$G = \{ \mathcal{V}, \mathcal{P}, \mathcal{A} \}$ - грамматика задаваемая для управления алгоритмом, где

$\mathcal{V}$ – множество нетерминальных символов;

$\mathcal{P}$ – конечное множество продукций или правил;

$\mathcal{A}$ – конечное множество аксиом или начальных значений, $axiom \in \mathcal{A}$ и $axiom \supset \Sigma$, где Σ – множество всех цепочек символов, включая пустой символ, порождаемых правилами $\mathcal{P}$.

$\bar{A} = \{ \mathcal{U}, \mathcal{T}, \alpha \}$ – перенастраиваемый алгоритм и его составляющие, где

$\mathcal{U}$ – множество терминальных символов, $\mathcal{V} \cup \mathcal{U} \subset \Theta$ – алфавит, конечное множество символов;

$\mathcal{T}$ – шаблон, состоящий из цепочек символов см. (1).

Будем обозначать $|\mathcal{T}|$ – число нетерминальных символов в шаблоне $\mathcal{T}$. Терминальному символу однозначно соответствует оператор фрактальной алгебры (см. далее). Для рассматриваемого шаблона (1) – это композиция +.

Соотношение между длинами цепочек левой и правой части шаблона $\mathcal{T} = \alpha + \rho$, однозначно определяет структуру кластера:

1. $|\alpha| = |\rho|$ кластер полностью сбалансированного бинарного дерева;
2. $|\alpha| > |\rho|$ кластер дерева Фибоначчи и его модификации;
3. $|\alpha| < |\rho|$ линейный массив, для кластеризации не применяется.

α – алгоритм, использует операторы фрактальной алгебры для построения объектов – поддеревьев по цепочкам символов шаблона.

$S = \{\Xi \mid \rho_1, ..., \rho_k\}$ – фрактальная алгебра, где $\rho_1, ..., \rho_k$ – множество уникально обратимых операторов композиции +, прототипирования ≡. $\mathcal{S}$ - определена над структурным пространством Ξ, которое необходимо для создания объектов, в виде алгоритма α.

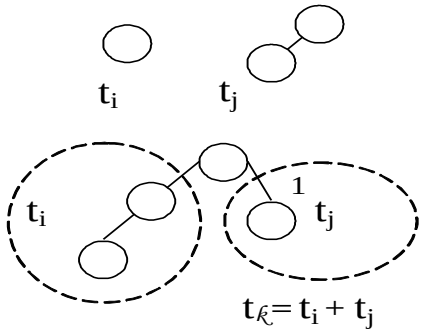

Рис. 2. Оператор композиции кластеров

Оператор композиции + алгоритма α соединяет два подкластера (поддерева) t_i и t_j в новый кластер (поддерево) t_k. Последовательность выполнения операции приведена на рис. 2: создается новая вершина, которая соединяет два поддерева.

W*(α, t)* – операция сохранить в памяти кластер t по имени цепочки символов шаблона α. Например, если на данном шаге выполнения алгоритма цепочка символов шаблона $\mathcal{T} = \mathcal{GC} + \mathcal{G}$, то в памяти сохранится кластер t по имени $\alpha = \mathcal{GC}$.

R*(ρ)* – операция получить из памяти кластер t по имени *ρ,* если кластер отсутствуе, то *null*.

Алгоритм α. Построение древовидной структуры кластеров по шаблону осуществляется слева на право и снизу вверх. Графическое представление строится подсистемой ввода-вывода.

Вход:

$\mathcal{T} = \alpha + \rho$ – шаблон

Выход:

t – кластер (дерево)

Инициализация:

level = 6 – высота кластера соответствует числу итераций

t = null – пустой кластер h = 0

h = 1 – строим кластер с одной вершиной

t = t + t – дерево с одним узлом, есть дерево высоты 1

W*(init,* ≡ t*)* – прототипирование t и сохранение прототипа по имени

Шаги:

for *h = 2* to *level* – построить кластер заданной высоты

t = M(t) + M(R(*ρ*)) – получить кластер по имени из памяти, композиция кластеров

rule (𝒯) – сгенерировать следующую цепочку символов по шаблону *𝒯*

W(*α,* ≡ t) – прототипирование и сохранение t в памяти по имени

end for

M – функция разметки кластера. Размечает построенную структуру t в соответствии с состоянием цепочек символов шаблона α и ρ. Алгоритмы могут выполняться асинхронно.

4. Фрактальный анализ и прогнозирование кластеризации

Под прогнозированием кластеризации будем понимать образование некоторого кластера еще несуществующих данных, которые могут появиться в результате работы некоторой динамической системы, причем расстояние между возможными кластерами определяется исходя из соответствия существующей кластеризации правилам построения кластеров алгоритма $\underline{\alpha}$.

Рассмотрим два метода анализа и прогнозирования кластеризации:

- по заданному шаблону. Визуализация дендрограммы осуществляется алгоритмом $\underline{\alpha}$;
- по функции пригодности, алгоритмом $\underline{\alpha}$ по шаблону строятся кластеры, шаблон может изменяться с целью большего соответствия функции пригодности.

4.1. Анализ и прогнозирование по заданному шаблону

Будем задавать различные правила генерации и начальные значения для шаблона $\mathcal{T}$ = α + ρ, управляющего алгоритмом $\underline{\alpha}$. Цепочкам символов α и ρ шаблона ставится в соответствие метка кластера, полученная алгоритмом $\underline{\alpha}$.

Целесообразно рассматривать шаблоны сбалансированного дерева и дерева Фибоначчи так, как между ними располагаются другие сбалансированные деревья.

Рассмотрим анализ и прогнозирование для кластеров типа: сбалансированного дерева, дерева без правого подкластера, дерева Фибоначчи и его модификации.

Вершины дерева изображаются кружками, ребрам присваиваются расстояния между кластерами. Вершины у основания деревьев (по горизонтали) помечаются объектами. Для некоторых вершин метки могут отсутствовать.

4.2. Сбалансированный кластер строится при | α | = | ρ |.

G = { $\mathcal{V}$={C, $\mathcal{G}$}, $\mathcal{P}$ = {$C \rightarrow C\mathcal{G}$, $\mathcal{G} \rightarrow C\mathcal{G}$}, $\mathcal{A}$ = {α = $\mathcal{G}$, ρ = C, *init* = ρ })

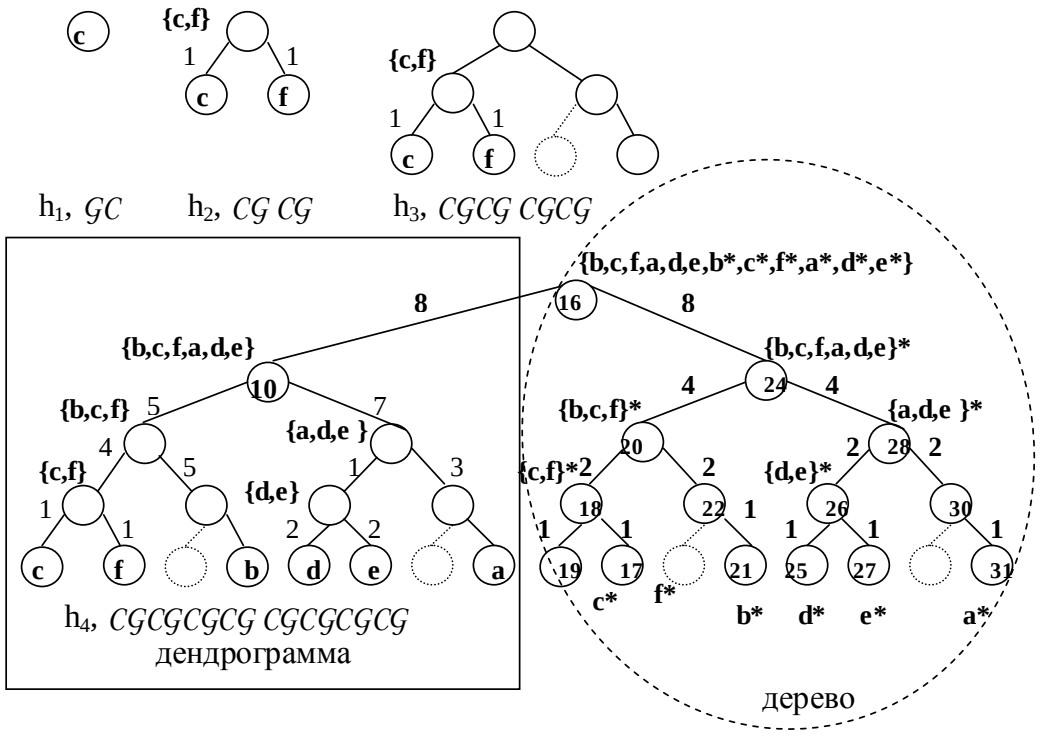

Рис. 3. Кластер сбалансированный, | α | =| ρ|

Анализ. На рис. 3. последовательно показаны шаги построения дендрограммы по шаблону сбалансированного дерева. На шаге h_1 вершине присваивается метка объекта {c}, h_2 - имеем кластер {c,f}, h_3 - построен шаблон, но метка не присвоена, кластер {b,c,f} размечается функцией M после построения кластера {a,d,e}, h_4 - построен кластер {b,c,f,a,d,e}. Не используемые вершины и дуги, показаны пунктирной линией. Дендрограмма выделена прямоугольником.

Кластер {c,f} подобен кластеру {d,e}, кластер {b,c,f} подобен кластеру {a,d,e}, что следует из построения и разметки шаблонов.

Прогнозирование. Выполним следующий шаг построения алгоритма $\underline{\alpha}$. Результат построения, представляющий собой сбалансированное дерево, обведен пунктирной линией на рис. 3. Кластер {b,c,f,a,d,e} подобен {b,c,f,a,d,e}*, все их подкластеры тоже подобны. Знак * показывает, что кластер размечен по прототипу.

Для прогнозирования расстояния между объектами пометим дуги, соединяющие кластеры. Сбалансированное бинарное дерево размечается в соответствии с числовой последовательностью 1,2,4,8,…,2^n.

Одним из возможных вариантов определения расстояний между объектами является следующий: для дерева высоты h_5, ребро должно быть помечено 8, тогда вершина кластера помечена значением N=16. Такая разметка превышает предыдущую (10 для вершины и 5 для дуги), удовлетворяя условию дендрограммы, расстояние увеличивается по вертикали.

Кластер {b,c,f,a,d,e}* размечается сверху – вниз следующим образом; дуга левого и правого поддерева помечаются 2^{n-1}, левая вершина помечается числом $N_{h-1} = N_h - 2^{n-1}$, правая вершина помечается числом $N_{h-1} = N_h + 2^{n-1}$, что соответствует длинам цепочек символов шаблона алгоритм $\underline{\alpha}$.

Построен возможный прогнозируемый кластер расстояний между объектами {b,c,f,a,d,e,b*,c*,f*,a*,d*,e*}, удовлетворяющий следующим критериям: монотонности, растяжению при увеличении высоты дерева h и сжатию при уменьшении h.

Кластер без правого подкластера строится при | α | = | ρ |.

$G = \{ \mathcal{V}=\{C, \mathcal{G}\}, \mathcal{P}=\{C \rightarrow C\mathcal{G}, \mathcal{G} \rightarrow C\mathcal{G}\}, \mathcal{A}=\{\alpha = \mathcal{G}, \rho = C, init = \alpha\} \}$

Анализ. На рис. 4. последовательно показаны шаги построения дендрограммы по шаблону, в котором отсутствует правое поддерево. На шаге h_1 вершине присваивается метка объекта {c}, на шаге h_2 отсутствует правая ветка с объектом {f} имеем редуцированный кластер {c,f}, h_3 - построен шаблон, но метка не присвоена, кластер {b,c,f} с отсутствующей промежуточной вершиной (тоже редукция) размечается функцией M после построения редуцированного кластера {a,d,e}, h_4 - построен кластер {b,c,f,a,d,e}. Не используемые вершины и дуги, показаны пунктирной линией. Дендрограмма выделена прямоугольником.

Кластер {c,f} и {d,e} редуцированы, редуцированный кластер {b,c,f} подобен кластеру {a,d,e}.

Прогнозирование. Выполним следующий шаг построения алгоритма $\underline{\alpha}$. Результат построения, представляет собой редуцированное дерево, обведено пунктирной линией см. рис. 4. Кластер {b,c,f,a,d,e} подобен {b,c,f,a,d,e}*, все их подкластеры тоже подобны. Знак * показывает, что кластер размечен по прототипу.

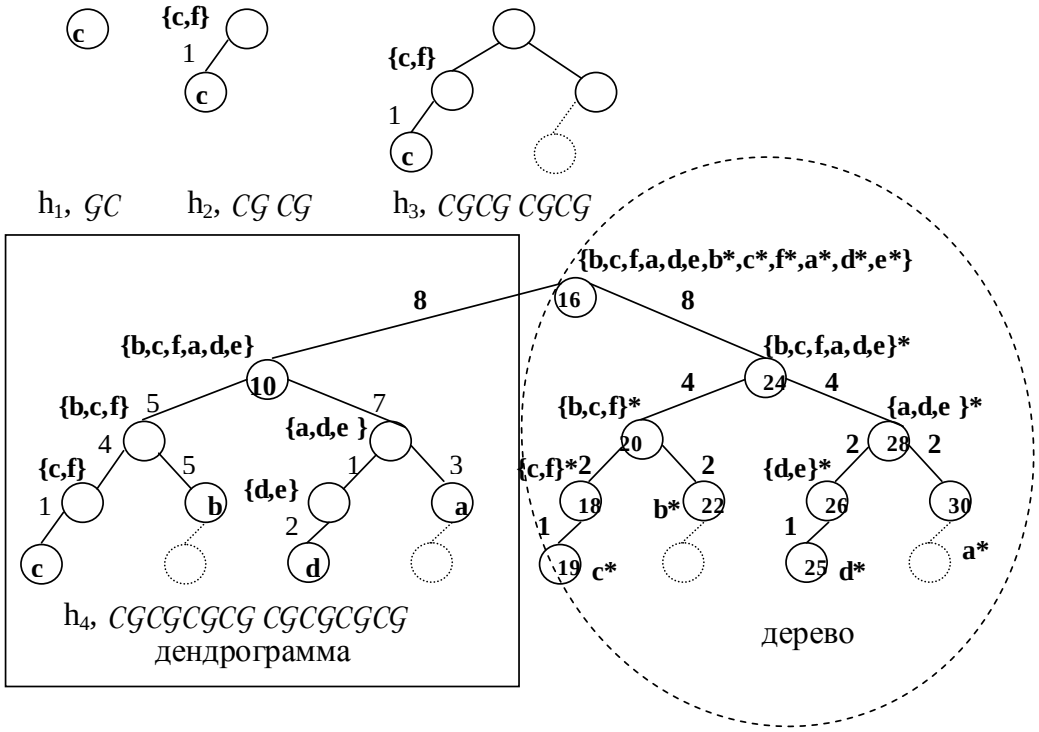

Рис. 4. Кластер без правого подкластера, | α | = | ρ |

Для прогнозирования расстояния между объектами пометим дуги, соединяющие кластеры, используя метод, рассмотренный для сбалансированного дерева.

Кластер расстояний между объектами {b,c,f,a,d,e,b*,c*,f*,a*,d*,e*}, удовлетворяет критериям: монотонности, растяжению при увеличении высоты дерева h и сжатию при уменьшении h, а также редукции по правому поддереву.

Прогнозируемые расстояния между кластерами {b}* и {b,c,f}*, {a}* и {a,d,e}* меньше, чем в сбалансированном дереве см. рис. 4.

4.3. Кластер Фибоначчи

Кластер Фибоначчи строится при | α | > | ρ |

$G = \{ \mathcal{V}=\{C, G\}, \mathcal{P}=\{C \rightarrow G,\ G \rightarrow GC\}, \mathcal{A}=\{\alpha = G, \rho = C, init = G\} \}$

Анализ. На рис. 5. последовательно показаны шаги построения дендрограммы по шаблону дерева Фибоначчи. На шаге h_1 вершине присваивается метка объекта {c}, на шаге h_2 отсутствует правая ветка с объектом {f} имеем редуцированный кластер {c,f}, h_3 - построен шаблон, но метка не присвоена, кластер {b,c,f} Фибоначчи (обведен пунктирной линией) подобен кластеру сбалансированного дерева с отсутствующей промежуточной вершиной {b,c,f}(см. рис. 4.). Кластер b,c,f} размечается функцией M после построения редуцированного кластера {d,e}, объект {a} редуцирован, h_4 - построен кластер {b,c,f,d,e}. Не используемые вершины и дуги отсутствуют, есть только одна промежуточная вершина в кластере {d,e}. Дендрограмма выделена прямоугольником.

Кластер {c,f} подобен кластеру {d,e}, количество подобных кластеров меньше по сравнению с предыдущими двумя шаблонами.

Прогнозирование. Выполним следующий шаг построения алгоритма $\underline{\alpha}$. Результат построения, представляет собой дерево Фибоначчи высотой h=5, обведено пунктирной линией см. рис. 5. Кластер {b,c,f,d,e} подобен {b,c,f,d,e}*, все их подкластеры тоже подобны. Знак * показывает, что кластер размечен по прототипу.

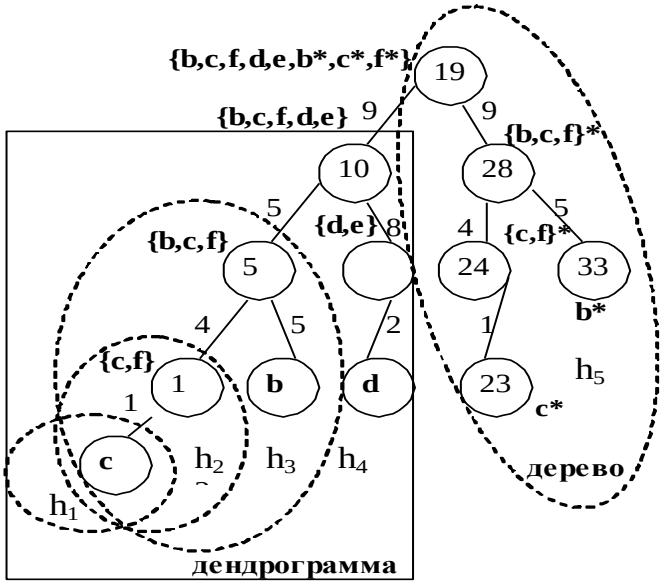

Рис. 5. Кластер Фибоначчи

Для прогнозирования расстояния между объектами пометим дуги, соединяющие кластеры. Дерево Фибоначчи размечается в соответствии с числовой последовательностью Фибоначчи $F_n = F_{n-1} + F_{n-2}$, например, 0,1,1,2,3,5,8,13,21,32,… .

Одним из возможных вариантов определения расстояний между объектами является следующий: для дерева высоты h_5, ребро должно быть помечено числом 9, так как 4+5, тогда вершина кластера помечена значением N=19. Разметка превышает предыдущую (10 для вершины и 5 для дуги), удовлетворяя условию дендрограммы, расстояние увеличивается по вертикали.

Кластер {b,c,f}* размечается сверху – вниз следующим образом; дуга правого и левого поддерева помечаются числами Фибоначчи, левая вершина помечается числом $N_{h-1} = N_h - F_{n-2}$, правая вершина помечается числом $N_{h-1} = N_h + F_{n-1}$.

Построен возможный прогнозируемый кластер расстояний между объектами {b,c,f,d,e,b*,c*,f*}, удовлетворяющий следующим критериям: монотонности, растяжению при увеличении высоты дерева h и сжатию при уменьшении h, редукции. Редукция по сравнению с предыдущими двумя шаблонами увеличивается, не рассматривается кластер {d,e}*.

Модифицированный кластер Фибоначчи. Начальные значения последовательностей Фибоначчи могут быть любыми, например, 0, 2, 2, 4, 6, 10, 16, …. Зададим начальную цепочку символов $\alpha = GCG$.

$G = \{ \mathcal{V}=\{C, G\}, \mathcal{P}=\{C \to G, G \to GC\}, \mathcal{A}=\{\alpha = GCG, \rho = C, init = \alpha\})$

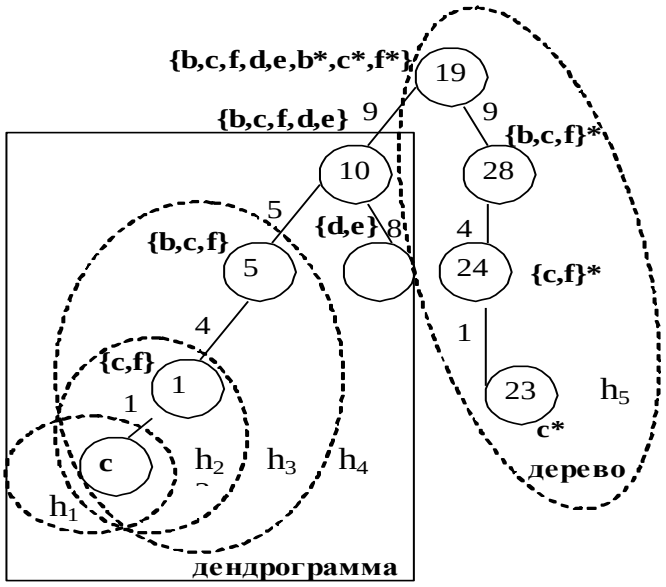

Рис. 6. Модифицированный кластер Фибоначчи

Анализ. На рис. 6. последовательно показаны шаги построения дендрограммы по шаблону модифицированного дерева Фибоначчи, изменены начальные условия построения. На шаге h_1 вершине присваивается метка объекта {c}, на шаге h_2 редуцированный кластер {c,f}, h_3 - редуцированный кластер {b,c,f} и далее h_4 - редуцированный кластер {d,e} и {b,c,f,d,e}. Все подученные кластеры редуцированы. Дендрограмма выделена прямоугольником. Подобные кластеры отсутствуют.

Прогнозирование. Выполним следующий шаг построения алгоритма $\underline{\alpha}$,. Результат построения, представляет собой дерево Фибоначчи высотой h=5, обведено пунктирной линией см. рис. 6. Кластер {b,c,f} подобен {b,c,f}*. Знак * показывает, что кластер размечен по прототипу.

Для прогнозирования расстояния между объектами пометим дуги, соединяющие кластеры, используя метод, рассмотренный для дерева Фибоначчи.

Построен возможный прогнозируемый кластер расстояний между объектами {b,c,f,d,e,b*,c*,f*}, удовлетворяющий следующим критериям: монотонности, растяжению при увеличении высоты дерева h и сжатию при уменьшении h, редукции. Редукция по сравнению с предыдущими шаблонами увеличивается, у кластеров отсутствуют объекты, кроме кластера {c,f} и {c,f}*.

4.4. Анализ и прогнозирование по функции пригодности

Уровень желательности различных разбиений и группировок, удовлетворяющий некоторой цели кластеризации, будем определять функцией пригодности (fitness).

Цели кластеризации на основе рассмотренных фрактальных шаблонов зависят от особенностей поставленной задачи:

1. Декомпозировать структуру множества объектов на подобные кластеры для ее понимания. Уменьшая число кластеров, упростить обработку данных и принятия решений, на основе стратегии ”разделяй и властвуй”, работая с каждым кластером по отдельности.

2. Редуцировать кластеры и тем самым объём хранимых объектов, оставив по одному подобному кластеру. Требуется высокая степень подобия объектов внутри каждого кластера.

3. Выделить кластеры не подобные ни одному из построенных.

Для реализации перечисленных целей функцию пригодности (fitness) будем задавать критериями кластеризации: монотонностью, растяжением и сжатием, редуктивностью. Результаты кластеризации визуализируются алгоритмом α в виде фрактальной структуры.

Обобщая сказанное, введем понятие самоорганизующейся фрактальной структуры, названной фрактоидом.

Определение 2. Алгебраическая структура $\mathcal{F}^{n}$ ($\mathcal{A}$,S,α,*fitness*) называется самоорганизующимся фрактоидом, где n – размерность фрактоида $\mathcal{F}$,

α – начальный компонент (самоподобное множество),

S – оператор преобразования, в данной статье задается $\mathcal{TL}$-системой см. определение 1.

$\mathcal{A}$ – самоподобное множество, представляющее собой заключительное состояние фрактоида, после шага n;

fitness – функция пригодности;

$\mathcal{A}$ – самоподобное множество называется структурным аттрактором, если удовлетворяет функции пригодности *fitness*.

Представим процесс кластеризации фрактоидами: сбалансированного дерева $\mathcal{T}^n(\mathcal{B}, S, \mathcal{B}_0)$, дерева без правого подкластера $\mathcal{T}^n(\mathcal{B}', S, \mathcal{B}'_0)$, дерева Фибоначчи $\mathcal{T}^n(Fib, S, Fib_0)$, модификацией дерева Фибоначчи $\mathcal{T}^n(Fib'', S, Fib''_0)$.

Определение 3. Начальный компонент α фрактоида размерности n можно записать фрактоидом n – 1, и подставить фрактоид, лучше удовлетворяющий функции пригодности *fitness*.

Пусть n = 1, имеем $\mathcal{T}^1(\mathcal{B}_1, S, \mathcal{B}_0)$;

Пусть n = 2, $\mathcal{T}^2(\mathcal{B}, S, \alpha) = \mathcal{T}^2(\mathcal{B}_2, S, \mathcal{B}_1)$, так как $\mathcal{T}^2$ предшествует $\mathcal{T}^1$, то $\mathcal{T}^2(\mathcal{B}, S, \alpha) = \mathcal{T}^2(\mathcal{B}_2, S, \mathcal{T}^1(\mathcal{B}_1, S, \mathcal{B}_0))$ – $\mathcal{T}^2$ построен из $\mathcal{T}^1$.

Если $\mathcal{T}^2$ и $\mathcal{T}^1$ – структурные аттракторы, некоторой динамической системы, то система эволюционирует от аттрактора $\mathcal{T}^1$ к $\mathcal{T}^2$.

Определение 4. Бифуркацией (альтернативой) будем называть процесс построения фрактоида после выполнения подстановки.

Пример. Пусть цель иерархической кластеризации максимально редуцировать кластеры, обозначим как max {..}, тогда алгоритм кластеризации при min d[i,j] описывается функцией пригодности:

fitness (max {..}, min d[i,j])

Самоорганизующийся фрактоид $\mathcal{T}^1(\mathcal{B}_1, S, \mathcal{B}_0,$ *fitness*) в результате бифуркаций будет реорганизовывать структуру, для удовлетворения заданной функции пригодности.

Например, $\mathcal{T}^2(Fib, S, \mathcal{T}^1(\mathcal{B}_1, S, \mathcal{B}_0))$ – после построения сбалансированного фрактоида размерности n = 1 возникает бифуркация и строится фрактоид Фибонначи.

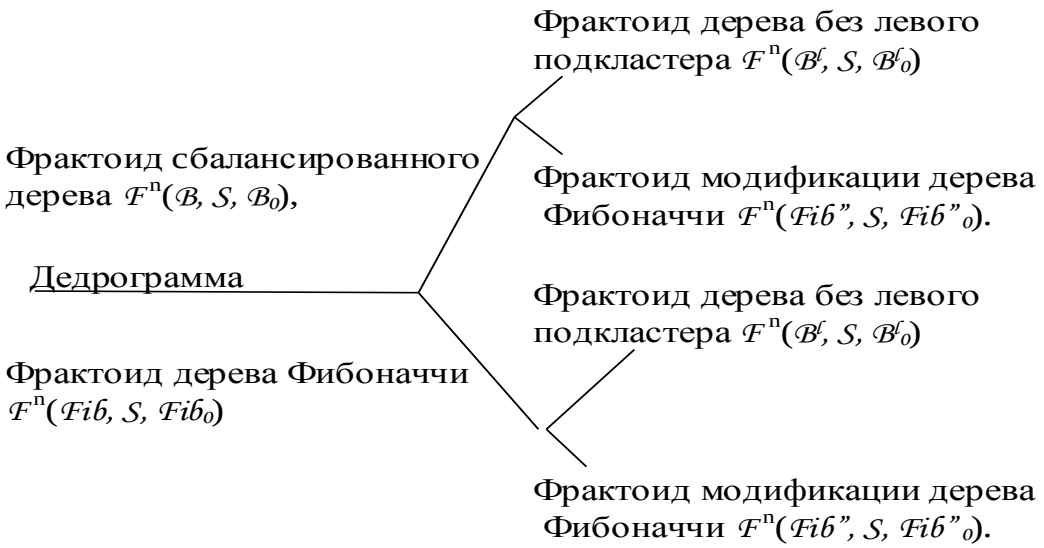

Рис. 7. Бифуркация и самоорганизация процесса кластеризации

Рис. 7. Иллюстрирует возникновение бифуркаций и самоорганизацию для рассматриваемого примера. Фрактоиды расположены в порядке уменьшения редукции. Анализ и прогнозирование, изложенные в разделе 3.1., являются составными частыми общего процесса самоорганизации.

Заключение

Текущие тенденции применения технологий Data Mining в космических аппаратах характеризуются непрерывным ростом объема архивных данных, количества систем, генерирующих телеметрические данные и ростом объемов самих телеметрических данных. Прослеживается тенденция к применению методов управления по данным.

Применение $\mathcal{TL}$-системы позволяет исследовать кластеризацию с точки зрения символьной динамики и грамматик языков, что открывает возможности прогнозировать динамику кластеризации и планировать поведение космического аппарата по фрактальным шаблонам.

Представление телеметрических данных в виде самоорганизующейся динамической системы фрактоид позволяет динамически изменять стратегии для поиска наиболее пригодных шаблонов, которые кодируют поведение космического аппарата.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. *Tanner S., Stein C., Graves S.J.,* "On-board Data Mining" in "Scientific Data Mining and Knowledge Discovery" by M.M. Gaber (Editor), Springer Verlag GmbH, 2009, pp. 345-376

2. *Chandola V., Banerjee A., Kumar V.,* "Anomaly Detection: A Survey", ACM Computing Surveys, Vol. 41(3), Article 15, July 2009 (PDF)

3. *Schwabacher M., Waterman R.,* "Pre-Launch Diagnostics for Launch Vehicles", IEEE Aerospace Conference, 2008. (PDF)

4. *Мандель И.Д.* Кластерный анализ. – М.: Финансы и Статистика, 1988.

5. *А.С.Семенов.* Построение класса фрактальных систем по шаблону на примере дерева Фибоначчи // Информационные технологии и Вычислительные системы. 2005. №2. с. 10-17.

6. *А.С.Семенов.* Фрактальное построение n-мерных гиперкубовых архитектур в структурном пространстве // Информационные технологии и Вычислительные системы. – М.: N1, 2007.

7. *Дюран Б.и Одел П.* Кластерный анализ. – М.: Статистика, 1977.

Семенов Александр Сергеевич, доцент Московского авиационного института (государственного технического университета), к.ф.-м.н.

e-mail: Semenov_Alex@yahoo.com