64-я научная конференция профессорско-преподавательского состава ЮУрГУ

Интеграция алгоритма кластеризации Fuzzy *c*-Means в СУБД PostgreSQL

Р.М. Миниахметов

Кафедра системного программирования
Южно-Уральский государственный университет

Челябинск

12 апреля 2012 г.

Актуальность исследования

- Анализ сверхбольших объемов данных
- Программное обеспечение с открытым исходным кодом
- Интеллектуальный анализ данных (Data Mining) и реляционные СУБД
- Нечеткая кластеризация в медицинских исследованиях
  - Количество различных параметров более 600
  - Частота дискретизации 10 мс
  - Объем необработанных данных по пациенту за день исследований более 20 МБайт

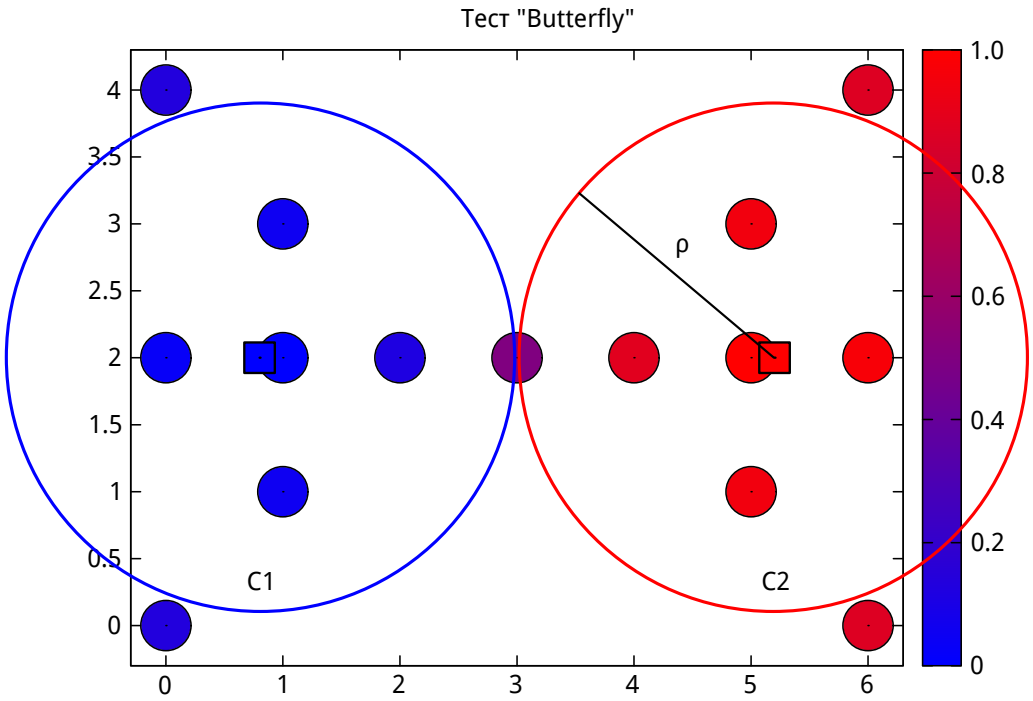
Цель и задачи

Цель работы — интегрировать алгоритм нечеткой кластеризации данных в СУБД PostgreSQL.

Задачи:

1. Проектирование алгоритма нечеткой кластеризации данных на языке реляционных баз данных SQL
2. Реализация алгоритма, адаптированного для СУБД PostgreSQL
3. Проведение тестирования разработанного алгоритма
4. Проведение вычислительных экспериментов по исследованию производительности разработанного алгоритма

Нечеткая кластеризация



- k — количество кластеров;
- N — количество векторов;
- m — степень нечеткости целевой функции;
- $x_i \in X$ — i-й вектор данных входного множества X, $|X| = N$;
- $c_j \in C$ — центр кластера j, вектор размерности d (*центроид*);
- C — множество центроидов, $|C| = k$;
- u_{ij} — функция принадлежности;
- $\rho(x_i, c_j)$ — функция расстояния.

$$J_{FCM}(X, k, m) = \sum_{i=1}^{N} \sum_{j=1}^{k} u_{ij}^{m} \rho^2(x_i, c_j), \quad 1 < m < \infty$$

Алгоритм Fuzzy *c*-Means

Вход: X — входное множество, k — количество кластеров,
m — степень нечеткости, ε — точность кластеризации

Выход: U — матрица степеней принадлежности

// Инициализация

$s := 0$, $U^{(0)} := random(0..1)$

повторять

// Вычисление новых координат центроидов

$$C^{(s)} := (c_j), \text{ где } c_{jl} = \frac{\sum_{i=1}^{n} u_{ij}^{m} \cdot x_{il}}{\sum_{i=1}^{n} u_{ij}^{m}}$$

// Обновление матрицы степеней принадлежности

$$U^{(s+1)} := (u_{ij}), \text{ где } u_{ij} = \sum_{t=1}^{k} \left(\frac{\rho(x_i, c_j)}{\rho(x_i, c_t)} \right)^{\frac{2}{1-m}}$$

пока $\max_{ij} \{ |u_{ij}^{(s)} - u_{ij}^{(s-1)}| \} > \varepsilon$

Алгоритм pgFCM

Реляционная структура

Входное множество данных

Матрица X

	x_1 ⋯ x_d
1	1.0 ⋯ 2.1
⋮	⋮ ⋱ ⋮
n	3.4 ⋯ 2.9

$\Longrightarrow$

Таблица SH (горизонтальная)

i	x_1	⋯	x_d
1	1.0	⋯	2.1
⋮	⋮	⋱	⋮
n	3.4	⋯	2.9

Таблица SV (вертикальная)

i	l	val
1	1	1.0
⋮	⋮	⋮
n	d	2.9

№	Таблица	Семантика	Атрибуты	Записи
1	*SH*	Выборка векторов данных	$\underline{i}, x1, x2, \ldots, xd$	n
2	*SV*	Выборка векторов данных	$\underline{i, l}, val$	$n{\cdot}d$

Координаты центроидов

Матрица C

	x_1 ⋯ x_d
1	2.2 ⋯ 8.1
⋮	⋮ ⋱ ⋮
k	3.4 ⋯ 6.9

$\Longrightarrow$

Таблица C

j	l	val
1	1	2.2
⋮	⋮	⋮
k	d	6.9

№	Таблица	Семантика	Атрибуты	Записи
3	*C*	Координаты центроидов	*j*, *l*, *val*	*k*·*d*

Степени принадлежности

Матрица U

	1	⋯	k
1	0.2	⋯	0.1
⋮	⋮	⋱	⋮
n	0.8	⋯	0.1

$\Longrightarrow$

Таблица U (для шага s)

i	j	val
1	1	0.2
⋮	⋮	⋮
n	k	0.1

Таблица UT (для шага s+1)

i	j	val
1	1	0.2
⋮	⋮	⋮
n	k	0.1

№	Таблица	Семантика	Атрибуты	Записи
4	*U*	Степени принадлежности вектора x_i кластеру j на шаге s	$\underline{i, j}$, val	$n \cdot k$
5	*UT*	Степени принадлежности вектора x_i кластеру j на шаге $s+1$	$\underline{i, j}$, val	$n \cdot k$

Вспомогательные таблицы

№	Таблица	Семантика	Атрибуты	Записи
6	*SD*	Расстояния между x_i и c_j	*i*, *j*, *dist*	$n \cdot k$
7	*P*	Значение функции $\delta = \max_{ij}\{\lvert u_{ij}^{(s+1)} - u_{ij}^{(s)} \rvert\}$ на текущей итерации	*d*, *k*, *n*, *s*, *delta*	*s*

Алгоритм pgFCM

Вход: SH — выборка векторов, k — количество кластеров,
m — степень нечеткости, eps — точность кластеризации
Выход: U — таблица степеней принадлежности

- - Инициализация
Создание и инициализация таблиц U, P, SV
повторять
- - Вычисления
Вычислить координаты центроидов (таблица C)
Вычислить расстояния (таблица SD)
Вычислить степени принадлежности $UT = \left(ut_{ij}\right)$ (таблица UT)
- - Обновление
Обновить таблицы P и U
- - Проверка завершения
пока $P.delta > eps$

Алгоритм pgFCM

Интерфейс функции pgFCM

- - Нечеткая кластеризация таблицы SH.
- - Вход: d - количество координат, k - количество кластеров,
- - m - степень нечеткости, eps - точность кластеризации.
- - Выход: -.
- - Результаты кластеризации хранятся в таблицах U и C.

```
CREATE OR REPLACE FUNCTION pgfcm(d INTEGER, k INTEGER,
  m NUMERIC, eps NUMERIC)
 RETURNS VOID
 LANGUAGE plpgsql
```

```
C1: INSERT INTO C
      SELECT R.j, SV.l,
             sum(R.s * SV.val) / sum(R.s) AS val
      FROM (SELECT i, j, U.val^m AS s
            FROM U) AS R, SV
      WHERE R.i = SV.i
      GROUP BY j, l;

C2: INSERT INTO SD
      SELECT i, j,
             sqrt(sum((SV.val - C.val)^2))) AS dist
      FROM SV, C
      WHERE SV.l = C.l
      GROUP BY i, j;
```

```
C3: INSERT INTO UT
      SELECT SD.i, j,
           SD.dist ^ (2.0 / (1.0 - m)) * SD1.den AS val
      FROM (SELECT i,
              1.0 / sum(dist ^ (2.0 / (1.0 - m))) AS den
            FROM SD
            GROUP BY i) AS SD1, SD
      WHERE SD.i = SD1.i;
```

Алгоритм pgFCM

Тест на наборе данных “Butterfly”

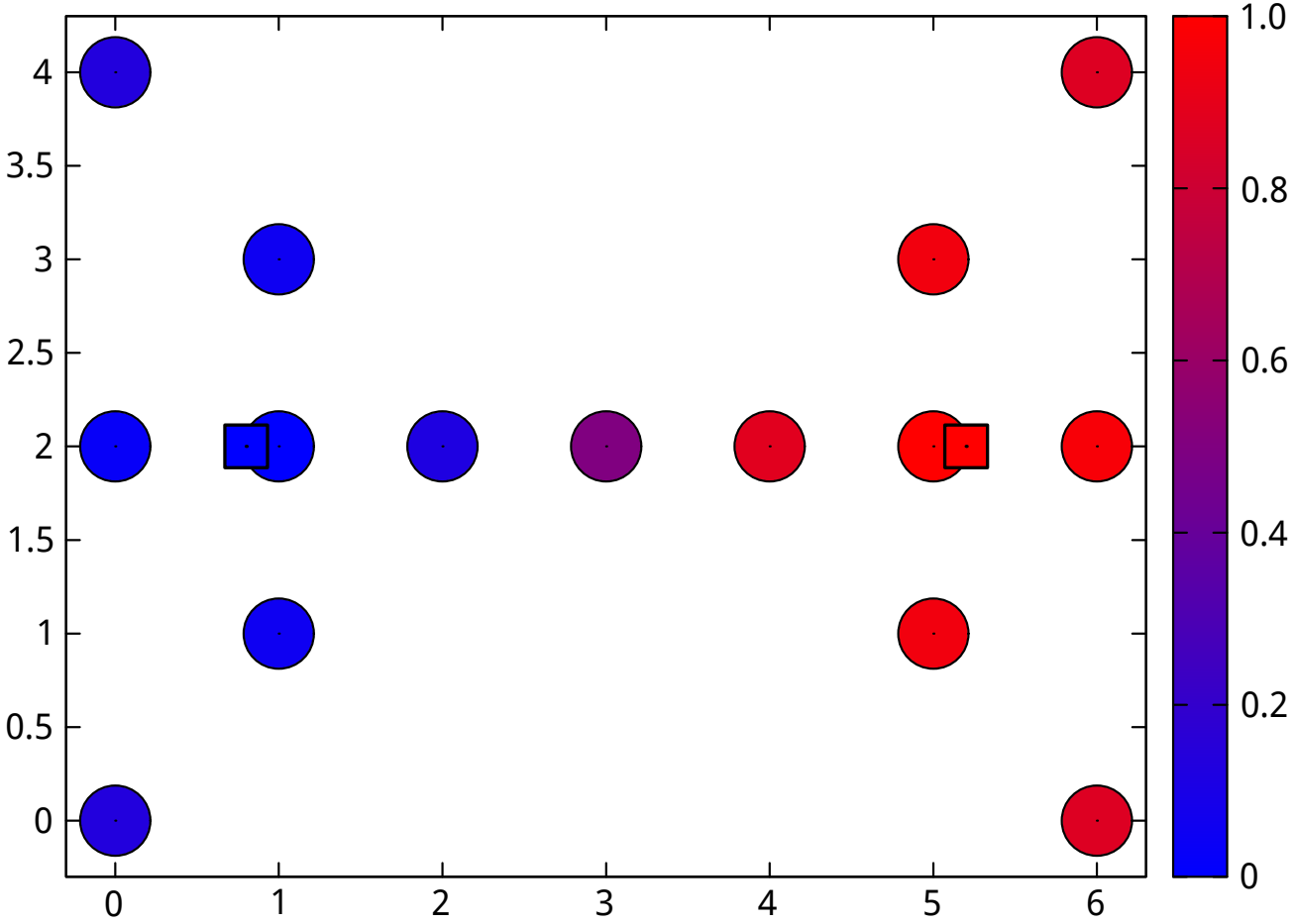


Таблица U

i	j	val
1	1	0.86
1	2	0.13
2	1	0.97
2	2	0.02
...	...	...
8	1	0.49
8	2	0.50
...	...	...
15	1	0.13
15	2	0.86

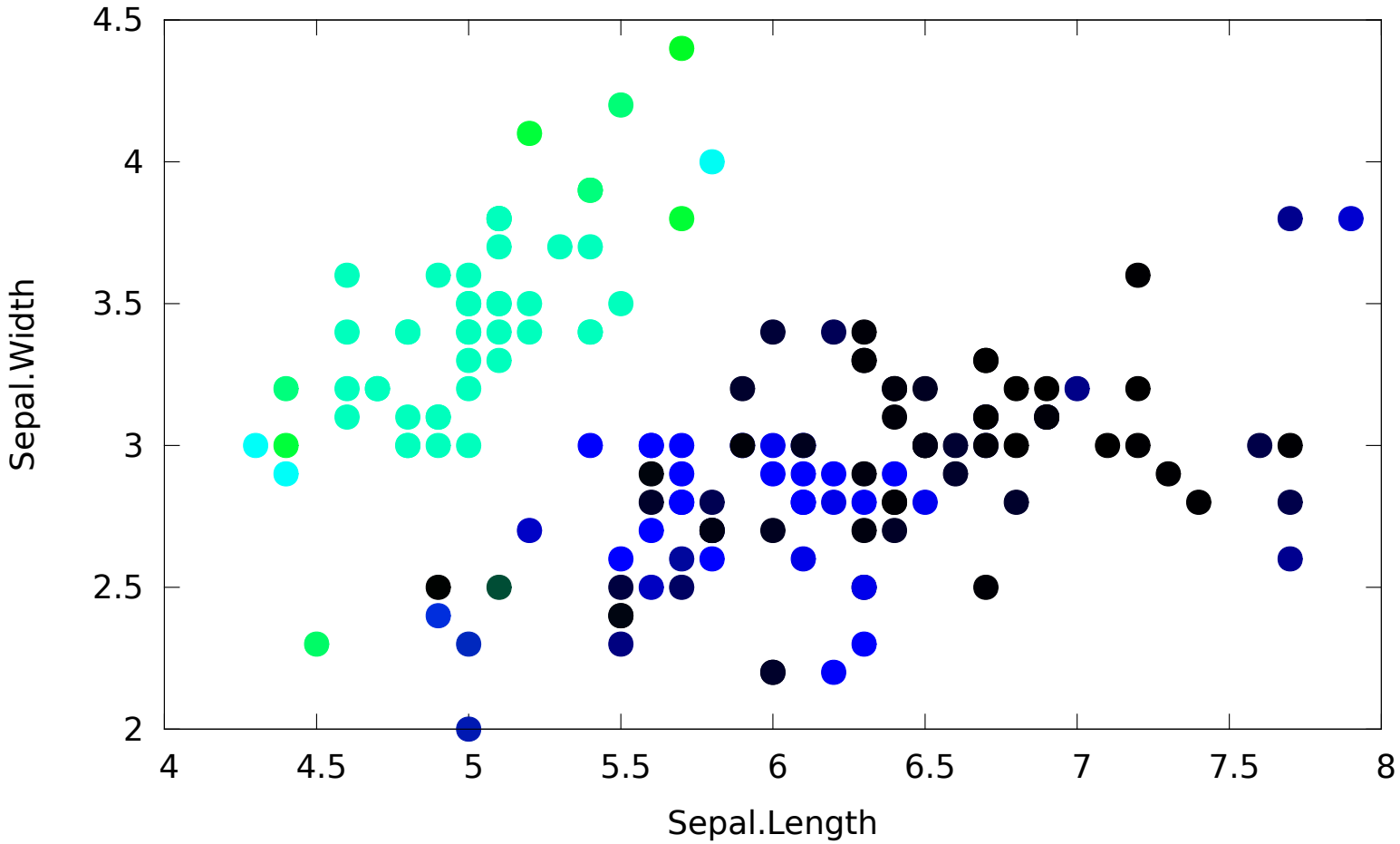
Таблица C

j	l	val
1	1	0.79
1	2	2.0
2	1	5.2
2	2	1.99

Параметры алгоритма: $d = 2$, $k = 2$, $m = 2.0$, $eps = 0.01$.

Алгоритм pgFCM

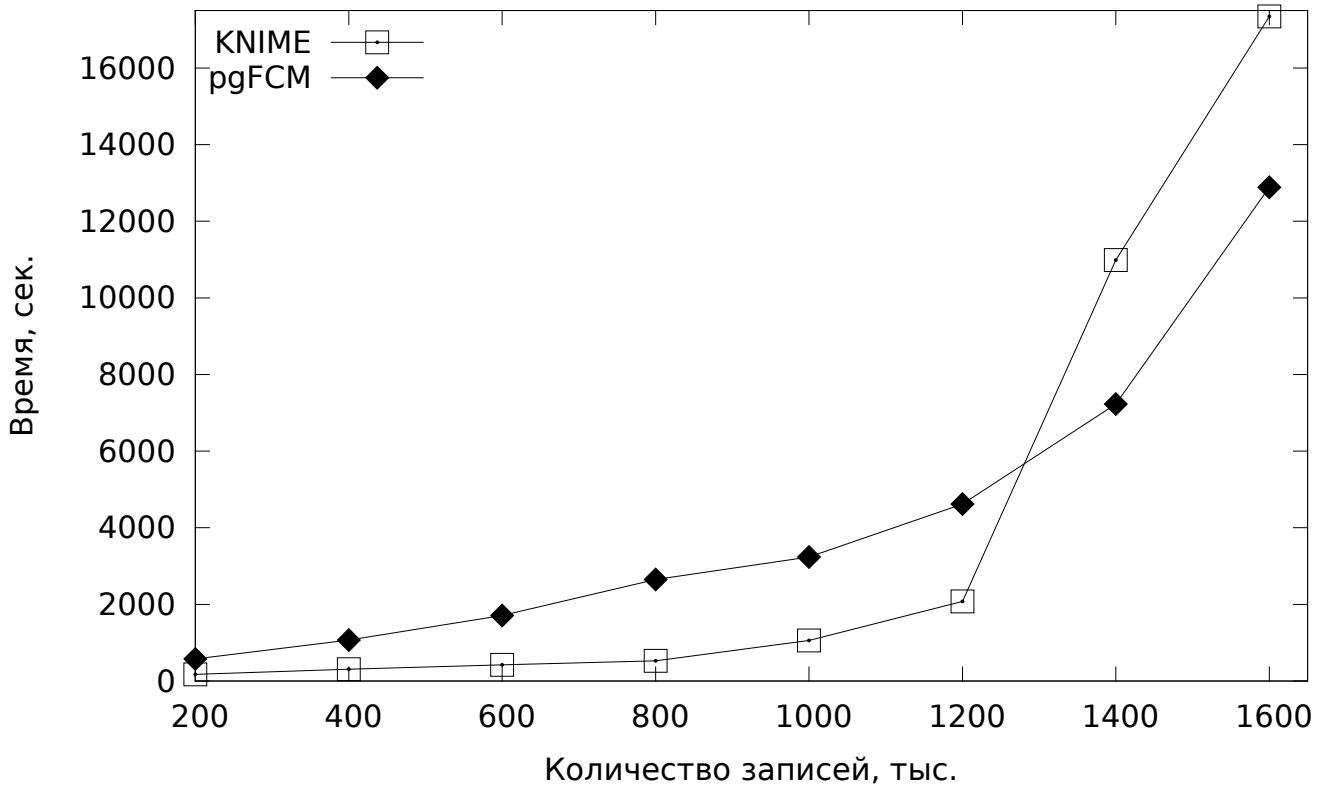
Тест на наборе данных “Iris”



Параметры алгоритма: $d = 5$, $k = 3$, $m = 2.0$, $eps = 0.01$.

Алгоритм pgFCM

Эксперименты по исследованию производительности



Параметры алгоритма: $d = 5$, $k = 3$, $m = 2.0$, $eps = 0.01$, $n = \overline{200\,000, 1\,600\,000}$.

Результаты работы

- Выполнено проектирование алгоритма нечеткой кластеризации на языке запросов SQL и схемы соответствующей реляционной базы данных.
- Выполнена реализация разработанного алгоритма для реляционной СУБД с открытым исходным кодом PostgreSQL.
- Выполнено тестирование на стандартных наборах данных Butterfly и Iris.
- Проведены эксперименты для исследования эффективности разработанного алгоритма на различных наборах данных.

Дальнейшие исследования

Дальнейшие исследования могут быть направлены на улучшение текущей реализации алгоритма pgFCM, а также на разработку параллельной версии алгоритма.

Алгоритм pgFCM

Эксперименты по исследованию производительности

N, тыс.	pgFCM, сек	KNIME, сек	JDBC1, сек	JDBC2, сек
200	578	174	3	25
400	1067	310	9	49
600	1711	423	13	75
800	2648	529	28	100
1000	3238	1061	95	125
1200	4620	2078	123	152
1400	7229	10989	161	178
1600	12888	17347	216	223

Параметры алгоритма: $d = 5$, $k = 3$, $m = 2.0$, $eps = 0.01$, $n = \overline{200\,000, 1\,600\,000}$.