Подход к интеграции интеллектуального анализа данных и реляционной СУБД на основе таблиц предвычислений

Т.В. Речкалов, М.Л. Цымблер

Южно-Уральский государственный университет (НИУ)

Предвычисления

Алгоритмы интеллектуального анализа данных (ИАД) требуют генерации большого количества вспомогательных данных. В ряде случаев данные могут быть повторно используемы. Идея состоит в том, чтобы *до исполнения* алгоритма ИАД вычислять вспомогательные данные и хранить их в РСУБД. При выполнении алгоритма данные запрашиваются из РСУБД.

Таблица предвычислений (ТПВ) представляет собой таблицу базы данных, содержащую предварительно вычисленные и повторно используемые данные для некоторого алгоритма ИАД. Целью использования ТПВ является ускорение алгоритма ИАД за счет уменьшения количества вычислений.

Общий вид ТПВ

Id1	Id2	…	IdR	Data1	Data2	…	DataP
id_{11}	id_{21}	…	id_{r1}	d_{11}	d_{21}	…	d_{p1}
id_{11}	id_{21}	…	id_{r2}	d_{11}	d_{21}	…	d_{p2}
…	…	…	…	…	…	…	…
id_{1q}	id_{2w}	…	id_{re}	d_{1t}	d_{2y}	…	d_{pu}

Столбцы ТПВ могут быть двух видов. В столбцах первого вида *{Id1, Id2, …, IdR}* хранятся внешние ключи. Их комбинация представляет собой составной первичный ключ. В оставшихся столбцах второго вида *{Data1, Data2, …, DataP}* хранятся предвычисленные результаты.

Алгоритм кластеризации K-Medoids

K-Medoids разбивает множество точек *D* на *k* кластеров, выбирая в качестве репрезентативного объекта кластера *медоид* – точку из *D*.

Алгоритм минимизирует целевую функцию $E = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, o_i)$

где *E* – сумма абсолютных ошибок всех точек *p* множества *D*, o_i – медоид кластера C_i

Временная сложность $O(k(n-k)^2)$ на одну итерацию

Псевдокод K-Medoids

Вход: число кластеров *k*, множество *D* из *n* точек.
Выход: множество *k* кластеров.
Метод:
произвольно выбрать *k* точек из *D* в качестве медоидов;
repeat
назначить каждую точку кластеру с ближайшим медоидом
for each кластер
случайно выбрать не-медоид o_{random}
вычислить цену перемещения o_j в o_{random}

$$S = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, o_i)^2 - \sum_{i=1}^{k} \sum_{p \in C_i} d(p, o_{random})^2$$

if *S<0* **then** заменить o_j на o_{random}
until нет переназначений точек между кластерами

ТПВ для алгоритма K-Medoids

	Id1	Id2	Distance
1	1	2	d_{12}
2	1	3	d_{13}
…	…	…	…
n^2-n	n	n-1	d_{nn-1}

Алгоритм K-Medoids выполняет кластеризацию конечного множества точек, используя в вычислениях значения расстояний от каждой точки до всех остальных. В силу этого в качестве ТПВ может использоваться матрица расстояний. В столбцах Id1 и Id2 хранятся номера точек. Столбец Distance предназначен для хранения предварительно вычисленных значений расстояний d_{ij} (расстояние между *i-й* и *j-й точками*).

Реализация

```
double dist(double *pt1, double *pt2, int dim);
double euclidDist(double *pt1, double *pt2, int dim);
double manhattanDist(double *pt1, double *pt2, int dim);

int n; // количество кластеризуемых точек
int dim; // количество координат кластеризуемых точек
double *data; // массив точек

double *preCompTab;

void fillTable(){
  for (int i = 0; i < n; ++i) {
    for (int j = 0; j < n; ++j) {
      preCompTab[i*n + j] = dist(data + i, data + j, dim);
    }
  }
}

double indexDist(int index1, int index2) {
  if(preCompTab != NULL){
    return preCompTab[index1 * n + index2];
  }
  return dist(data + index1, data + index2, dim);
}
```

Эксперименты

Аппаратная платформа – узел суперкомпьютера «Торнадо ЮУрГУ»: процессор Intel Xeon X5680, сопроцессор Intel Xeon Phi SE10X. Режим работы сопроцессора: *offload*. Данные: точки из R^2, тип double, синтетические.