

Реализация алгоритма поиска skyline points

04.09.2023

Понятие skyline-a

- ▶ Объекты представлены в виде точек, конкретная характеристика - одно измерение
- ▶ Точка доминирует над другой точкой, если она так же хороша или лучше во всех измерениях и лучше хотя бы в одном измерении.
- ▶ Skyline - множество точек, над которыми не доминируют никакие другие точки

Ограничения по входным данным

- ▶ 2D датасет
- ▶ Датасет не содержит дубликатов
- ▶ Значения координат - положительные вещественные числа

Алгоритм поиска ближайшего соседа (NNsearch)

- ▶ Применимость:
 - ▶ Ближайший сосед к точке $(0,0)$ согласно монотонной функции F будет принадлежать к skyline points
 - ▶ Если есть датасет D , и $D_m \in D$ содержит только точки со строго большими значениями координат (т. е. $x_m > x$ и $y_m > y$), тогда ближайший сосед к $(0, 0)$ из D_m будет принадлежать к skyline points как для D_m , так и для D

Алгоритм поиска ближайшего соседа (NNsearch)

► Идея:

- С помощью монотонной функции F найти ближайшую к $(0,0)$ точку
- Разделить пространство на 3 региона с помощью найденной точки (область - открытое множество)
- Исключить из рассмотрения область 3, а для регионов 1 и 2 применить данный алгоритм независимо

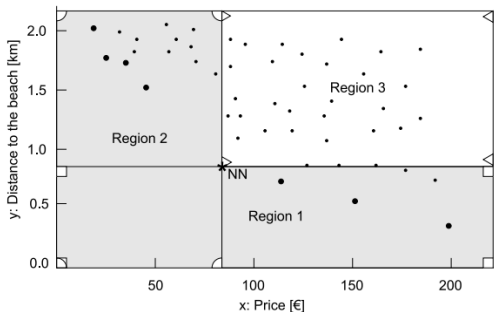
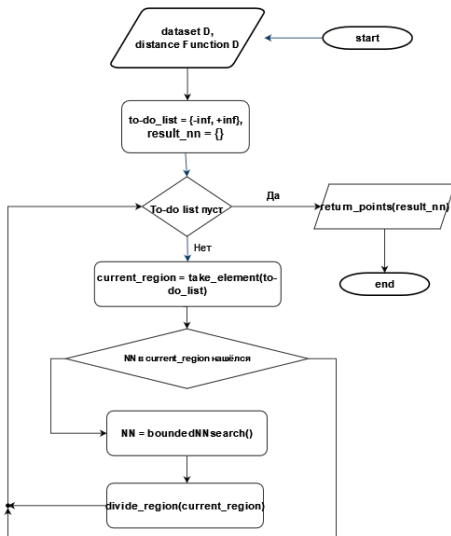
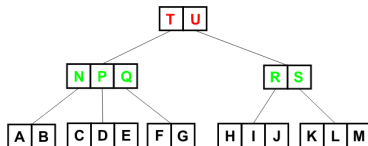
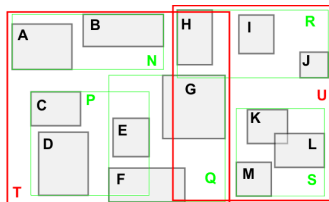


Схема алгоритма



R-tree



- ▶ Аналог B-дерева для пространственных объектов
- ▶ Оперирование предикатами "внутри" "пересекаются"...

R*-tree и Boost

- ▶ Шаблонная реализация взята из `boost::geometry::index::rtree`
- ▶ Использован подвид R*-дерево, в котором применена эвристика для минимизации перекрытий
- ▶ Применён алгоритм упаковки, значительно уменьшающий время на создание дерева и запросы к нему

Реализация функции поиска NN

```
bool boundedNNsearch(point_type & nn, rtree_type & tree, point_type region_border_point){  
  
    vector_type result_nn = {};  
    tree.query(bgi::nearest(point_type{0, 0}, 1) && bgi::covered_by(bg::model::box( {0, 0},  
    region_border_point)), std::back_inserter(result_nn));  
  
    if(result_nn.empty()){  
        return false;  
    }  
    nn = result_nn.front();  
    tree.remove(nn);  
  
    return true;  
}
```

Результаты и производительность

Конфигурация: AMD Ryzen 5 5600H 3.30 GHz, Windows 10 Home, VBox 7.0.8, 7GB RAM, Ubuntu 22.04

- ▶ Для каждого количества было проведено по 10 испытаний, представлено среднее время
- ▶ Большую часть времени занимает именно построение R^* -дерева

Усредненные результаты		
	100 000 точек	1 000 000 точек
Skyline	1-2	2-4
Время	0.163	1.923
Время R^*	0.156	1.914
Время опт.	0.181	0.223

Таблица: количество skyline points, время работы функции `get_skyline` [сек], время работы конструктора `rtee()` [сек], время с флагом оптимизации `o2` [сек]

Пример выдачи результатов

Красным отмечены точки skyline

```
Rtree creation time was 0.000473524 sec  
Random generated dataset of 500 points  
(62.5879, 32.2102)  
(51.0973, 123.321)  
(181.965, 15.9571)  
(26.0132, 166.472)  
(259.53, 6.42894)  
(11.7439, 272.748)  
(6.54526, 403.996)
```

```
Count of skyline points is 7  
Calculation time was 0.000730083 sec
```

