Кластеризация и визуализация экспериментальных данных в автоматизированной системе тематического анализа информации

м.н.с. Титов А.С.
НИИ механики МГУ

План доклада

- Что такое кластеризация и основные области её применения
- Этапы кластеризации, основные алгоритмы кластеризации
- Алгоритм гравитационной кластеризации
- Алгоритм визуализации многомерных данных

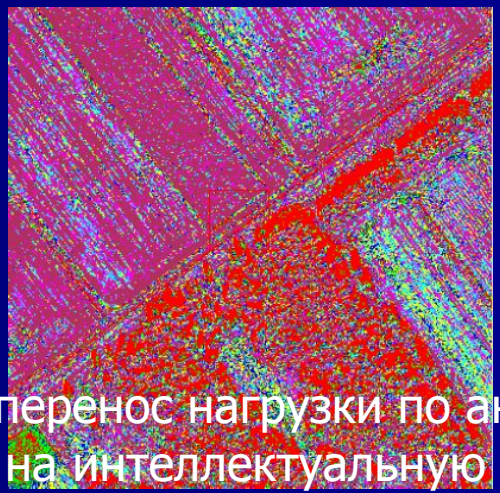
Кластеризация (кластерный анализ)

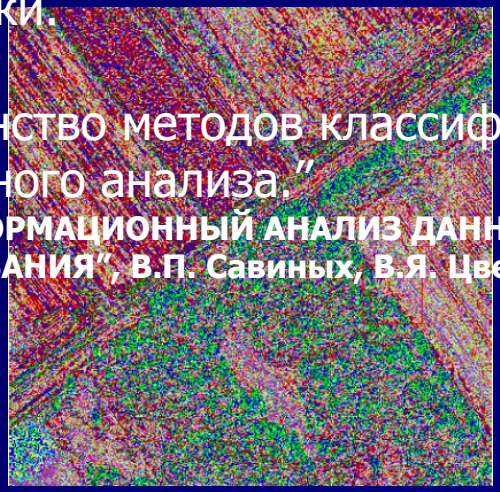
“Кластерный анализ – задача разбиения заданной выборки объектов на непересекающиеся подмножества, называемые кластерами так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались.”

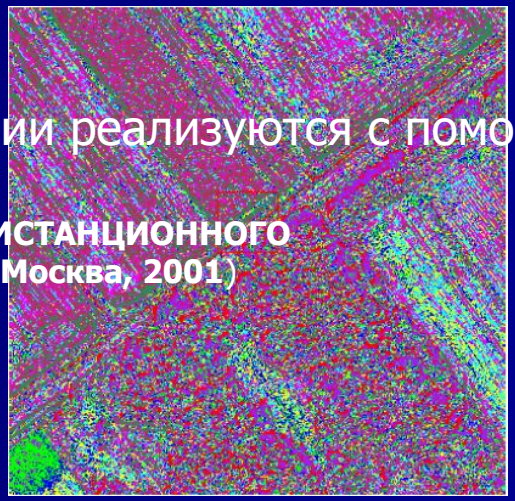
Процесс кластеризации достаточно трудоёмкая задача в особенности из-за большого шагов при её решении и возможности проводить различные изменения параметров на каждом из них.

Поэтому необходим алгоритм, который не требует задания дополнительных параметров и автоматически определяет число кластеров в исходных данных.

Дистанционное зондирование Земли







“Задачей классификации является перенос нагрузки по анализу и обработки информации с человека на интеллектуальную технологию обработки.

…

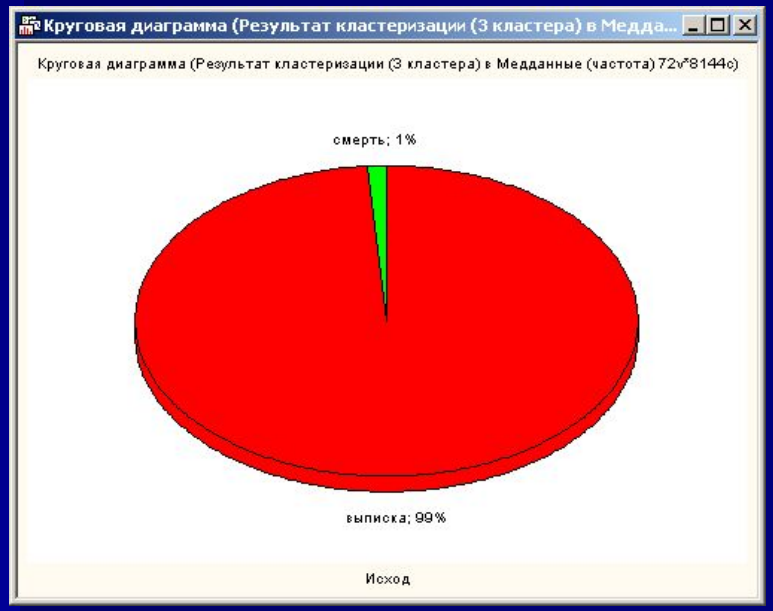
Большинство методов классификации реализуются с помощью кластерного анализа.”

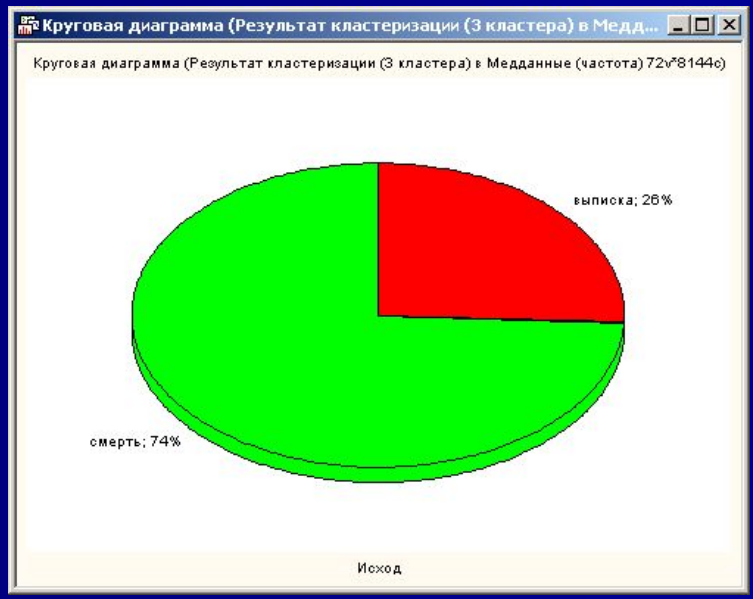
(“**ГЕОИНФОРМАЦИОННЫЙ АНАЛИЗ ДАННЫХ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ**”, **В.П. Савиных, В.Я. Цветков, Москва, 2001**)

Анализ данных Скорой помощи

Данные: база для кластерного анализа 1* (69v * 8144c)

	5 Исход	6 Операция	7 Пол	8 Число койко-дней	9 Клинический анализ крови
1	выписка		Мужской	1	1
2	выписка		Женский	4	0
3	выписка		Мужской	0	0
4	выписка		Женский	4	1
5	выписка		Женский	2	1





Традиционно кластеризация включает в себя следующие шаги:

1. Определить цель исследования. Это может быть или определение кластерной структуры данных (проверка наличия кластеров в данных) или разбиение объектов на группы по заранее определенному принципу, например, разбить множество книг по авторству или по их тематике.
2. Преобразование имеющихся данных в объекты для их кластеризации, например, определение набора слов, которые характеризуют тексты, определение переменных, характеризующих объекты, вычленение подпоследовательностей из генома.
3. Преобразование выбранных объектов в “компьютерный вид”, т.е. сопоставление значениям переменных числовые значения.
4. Задание метрики пространства. Для некоторых областей предпочтительны специфические метрики.
5. Выбор алгоритма кластеризации, что включает выбор параметров алгоритма, например, задание способа определения расстояния между группами точек, задание граничных параметров.
6. Интерпретация результатов кластеризации включает вычисление статистических характеристик каждого из кластеров и в результате сравнительного анализа проверяется соответствие заданному принципу разбиения. В случае несоответствия или отсутствия кластерной структуры проводится повторный процесс кластеризации с изменением настроек пунктов 1-4.

Краткий обзор методов кластеризации

- Классические алгоритмы
  - Алгоритмы, основанные на минимизации функционала
  - Алгоритм К – средних
  - Форель(FOREL), ИСОМАД(ISODATA), ПУЛЬСАР
  - Иерархические алгоритмы
- Современные алгоритмы
  - DBSCAN
  - BIRCH
  - CLARANS
  - CURE

Алгоритм К-средних

Описание алгоритма:

1. Задание числа кластеров k, на которые надо разбить входные данные.
2. Выбор k точек в исходном пространстве, именуемые как центры кластеров. На практике в основном выбирают случайные точки исходных данных.
3. Для каждой точки входных данных находится ближайший центр кластера. Группа точек, ближайшая к некоторому центру кластера, называются кластером.
4. Вычисление для каждого кластера С его центра масс , которая объявляется новым центром кластера.
5. Повтор процедуры с 3 шага в том случае, если не выполняется критерий остановки, например, “неизменность” кластеров за несколько последних итераций.

Алгоритм К-средних

Основная трудность, которая возникает при использовании алгоритма к-средних – необходимость задания числа кластеров. Для обхода этого недостатка возможны следующие варианты применения алгоритма, которые позволяют частично обойти это ограничение.

- Применение алгоритма для нахождения большого числа групп, где не требуется интерпретация полученных кластеров, например, применяется в поисковых системах при обработке новостных сообщений. Выделяются группы документов (новостных сообщений), которые рассматриваются как подборка новостей по некоторому событию.
- Применение алгоритма с разным параметром числа кластеров 2,3,… и последующее сравнение результатов для определения наилучшего разбиения или для доказательства отсутствия четкой кластерной структуры в исходных данных.
- Использование параметра числа кластеров, полученного из исследований структуры исходных данных, например, построение дендрограмм для данных с разными метриками близости.

Алгоритм К-средних

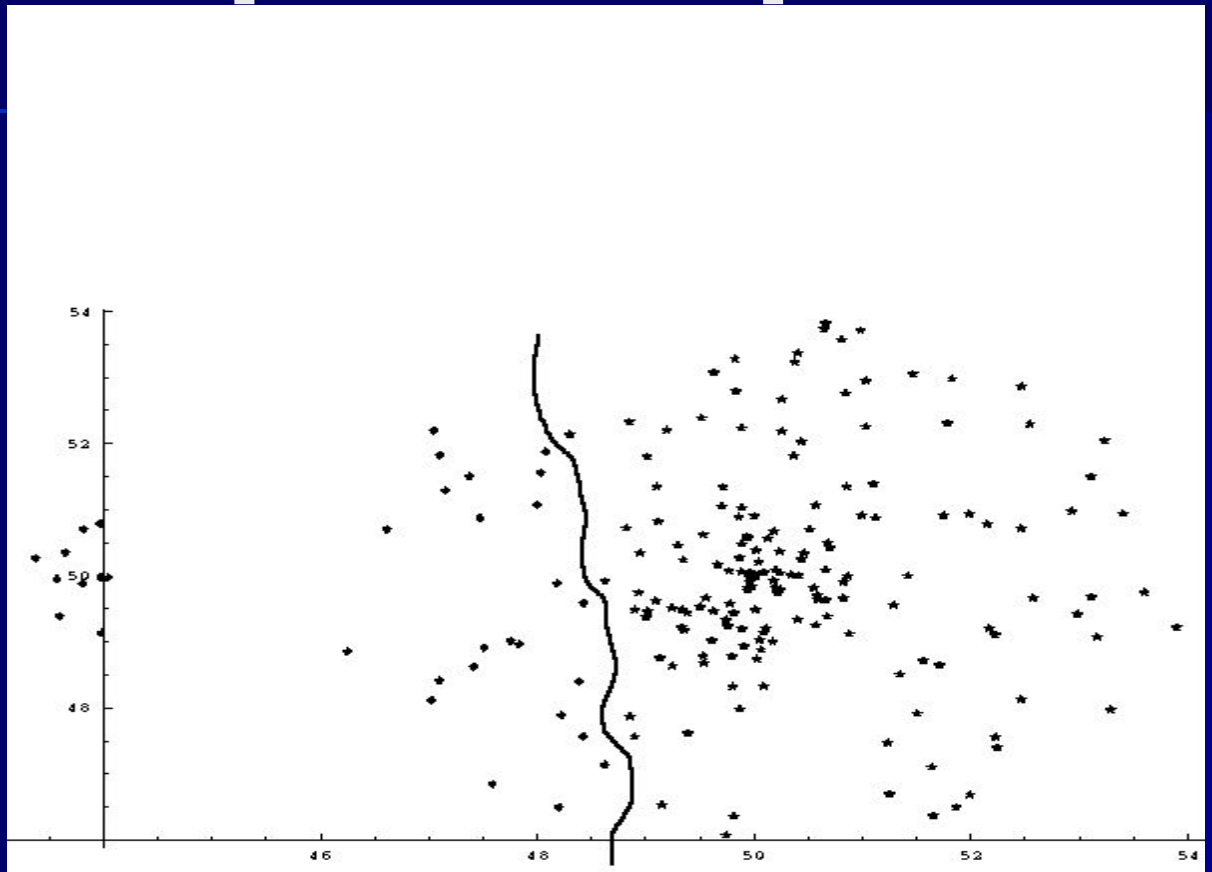
Основные достоинства алгоритма:

- простота использовании метода - задание только одного параметра (числа кластеров).
- один шаг алгоритма – O(n*K*k). На практике число шагов не велико.

К недостаткам алгоритма стоит отнести следующие обстоятельства:

- число кластеров необходимо задавать, нет автоматического определения числа кластеров в исходных данных;
- результаты кластеризации алгоритма к-средних могут содержать ошибки, как это представлено на рисунке.

Алгоритм К-средних



Постановка задачи кластеризации

Дано $X = \{\vec{X}_i\}_{i=1}^{m}, \quad \vec{X}_i \in R^n, \; d : R^n \to R$

Требуется разбить на множество групп (кластеров)

$$\{C_1, \ldots, C_m\}, \quad C_i \cap C_j = \emptyset, \quad i \neq j, \quad \bigcup_{i=1}^{m} C_i = X$$

чтобы точки из одной группы были “близки”, а из разных “различны”. Параметр m не задается.

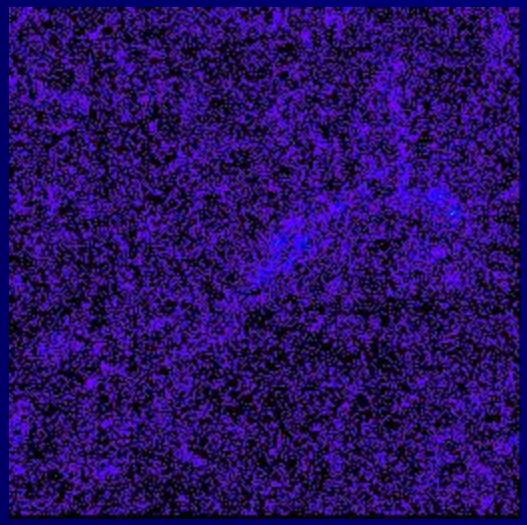
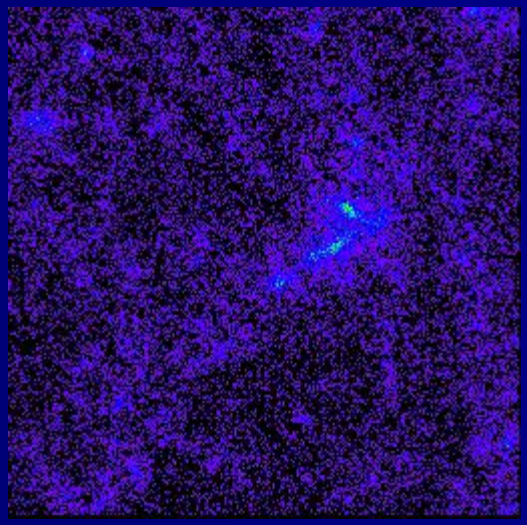
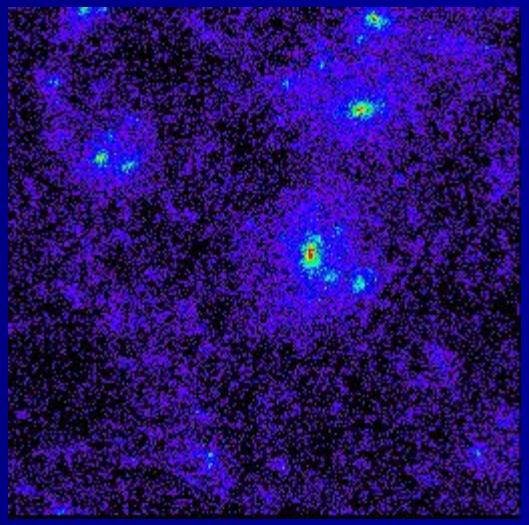
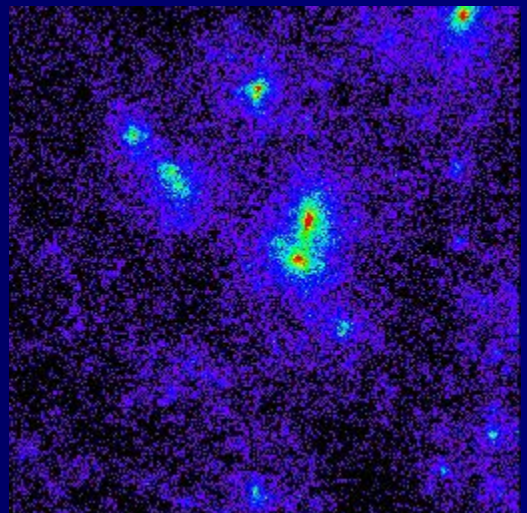
Требования к решению

- маштабируемость
- независимость от порядка данных
- независимость параметров алгоритма от данных

Алгоритм гравитационной кластеризации

- Общее описание
- Шаг 1: построение дерева объединений
- Шаг 2: построение “естественных” кластеров
- Шаг 3: построение кластеризации

Алгоритм гравитационной кластеризации: общее описание

$$F_{ij} = \frac{m_i m_j}{d^2(X_i, X_j)}$$

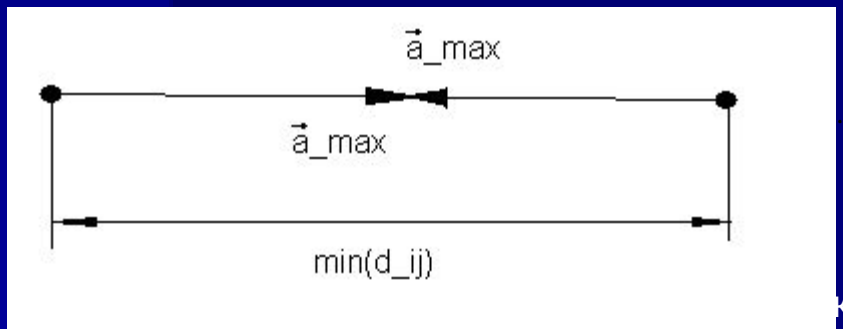
$d(\vec{X}_i, \vec{X}_j) < \frac{d_{\min}}{M_1}$

Алгоритм гравитационной кластеризации: построение дерева объединений

```
numpoints=n;
t=0;
while(numpoints!=1)
{
   объединение “близких” точек
   движение точек
}
```

- Объединение “близких”
  Две точки $\vec{X}_i$ и $\vec{X}_j$ объединяются, если $d(\vec{X}_i, \vec{X}_j) < \frac{d_{\min}}{M_1}$
  где $d_{\min}$ — минимальное расстояние между точками в начальный момент времени (t=0),
  а M_1 — первая константа алгоритма. Результатом объединения является точка $\vec{X}$

  с пересчитанными координатами: $\vec{X}_i = \frac{m_i \vec{X}_i + m_j \vec{X}_j}{m_i + m_j}$ $m_i = m_i + m_j$

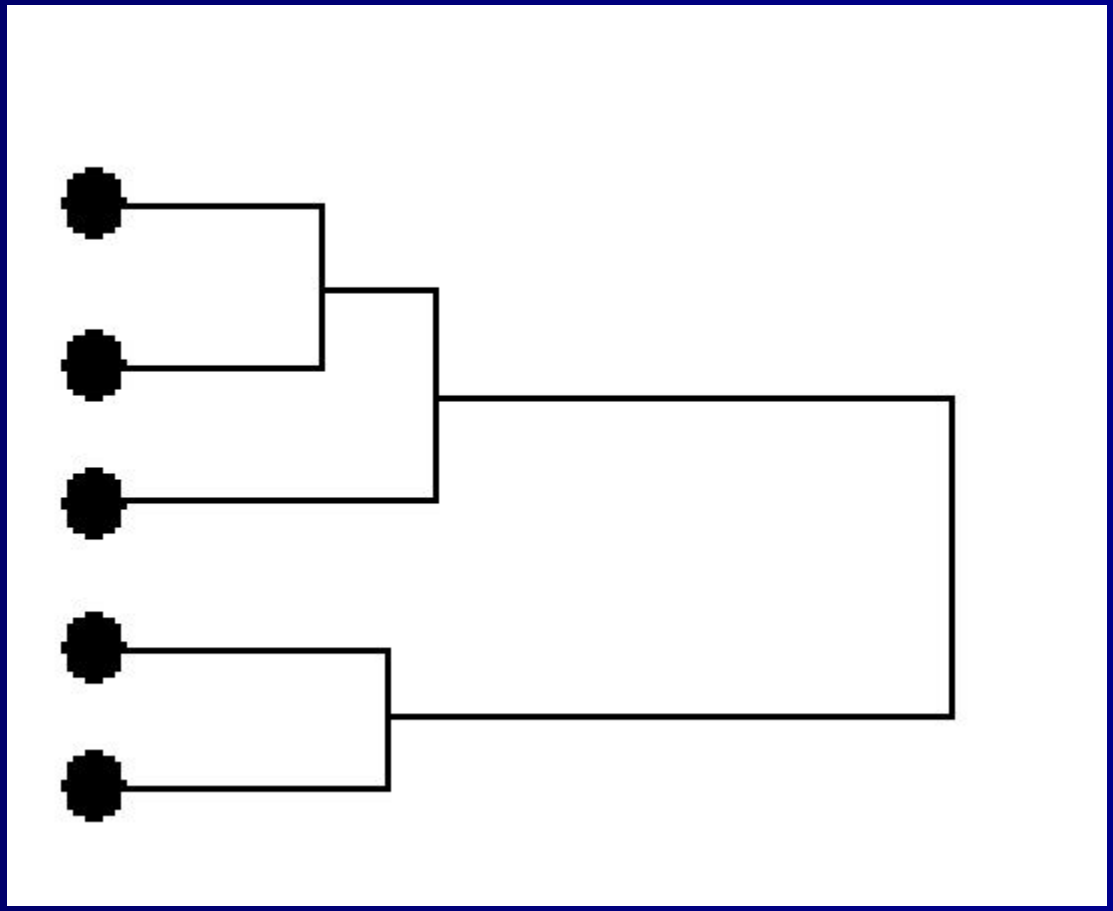


$\frac{\vec{r}_{ij}}{\sqrt{|\vec{r}_{ij}^2|}}$ $\vec{F}_i = \sum_{j \neq i} \vec{F}_{ji}$ $\vec{a}_i = \frac{\vec{F}_i}{m_i}$

$\Delta t = \sqrt{\frac{\min_{ij} d(\vec{X}_i, \vec{X}_j)}{\max_i |\vec{a}_i|}}$ $\vec{X}_i = \vec{X}_i + \frac{\vec{a}_i \Delta t^2}{2}$

как , движение точек -

Алгоритм гравитационной кластеризации: построение дерева объединений

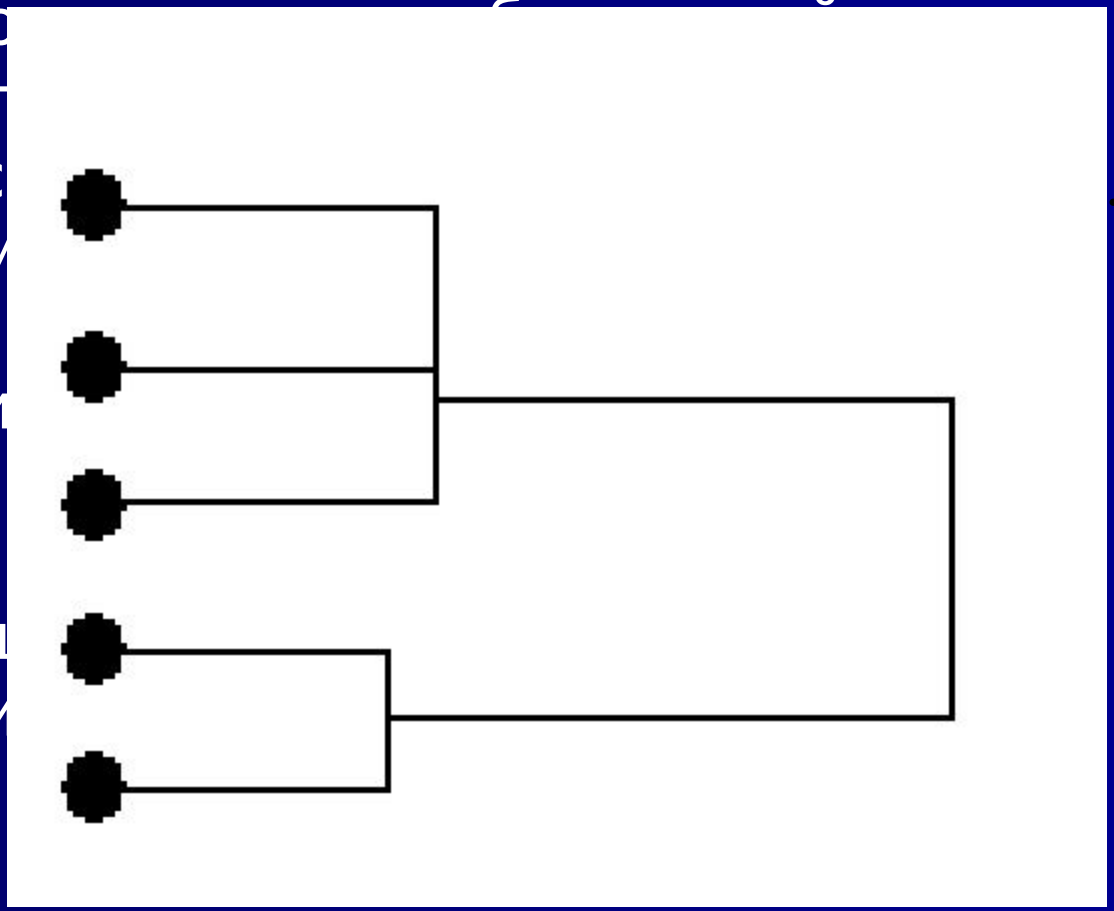


Алгоритм гравитационной кластеризации: построение “естественных” кластеров

- Упро
  Пуст
  расс
  врем

  Если

  (где
  верш
  роди

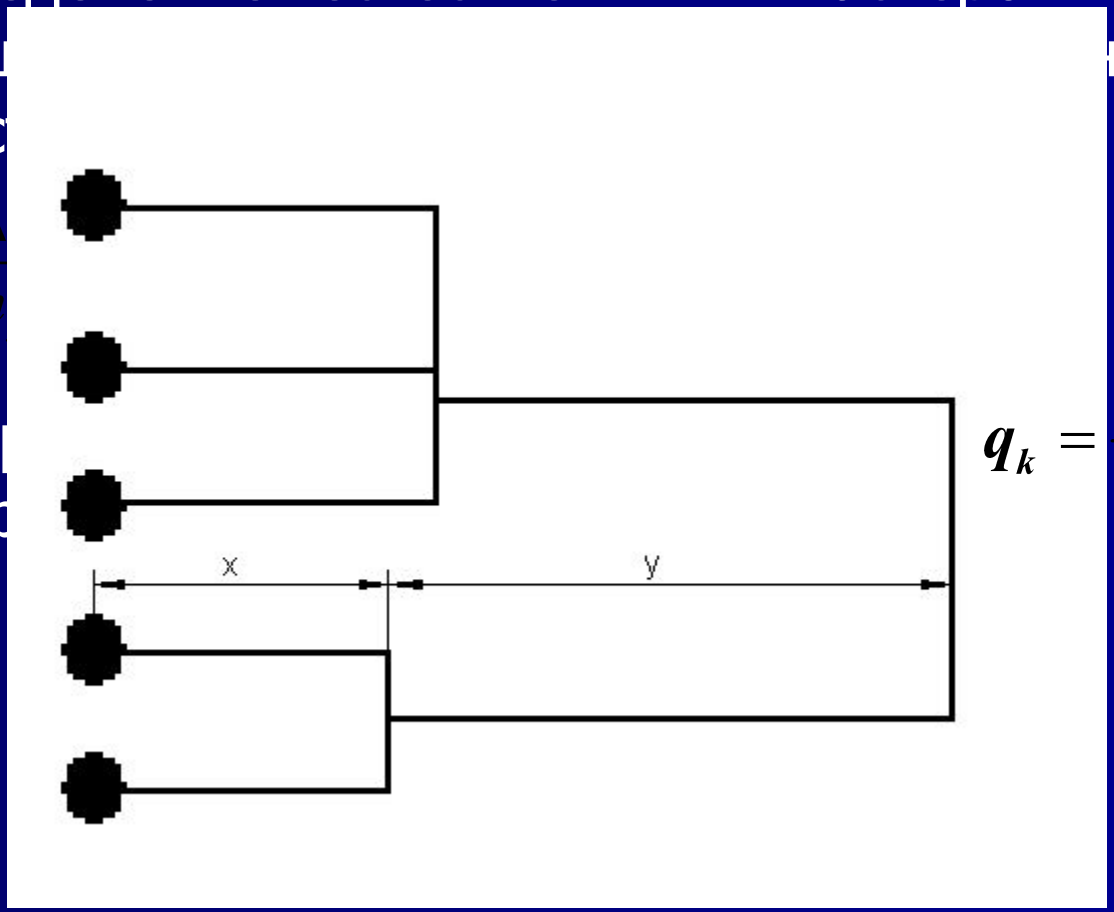


Алгоритм гравитационной кластеризации: построение “естественных” кластеров

- Определение “естественных” кластеров
  Верш … ным”
  клас…

$$\frac{\Delta}{min}$$

  и ве…
  коэф…

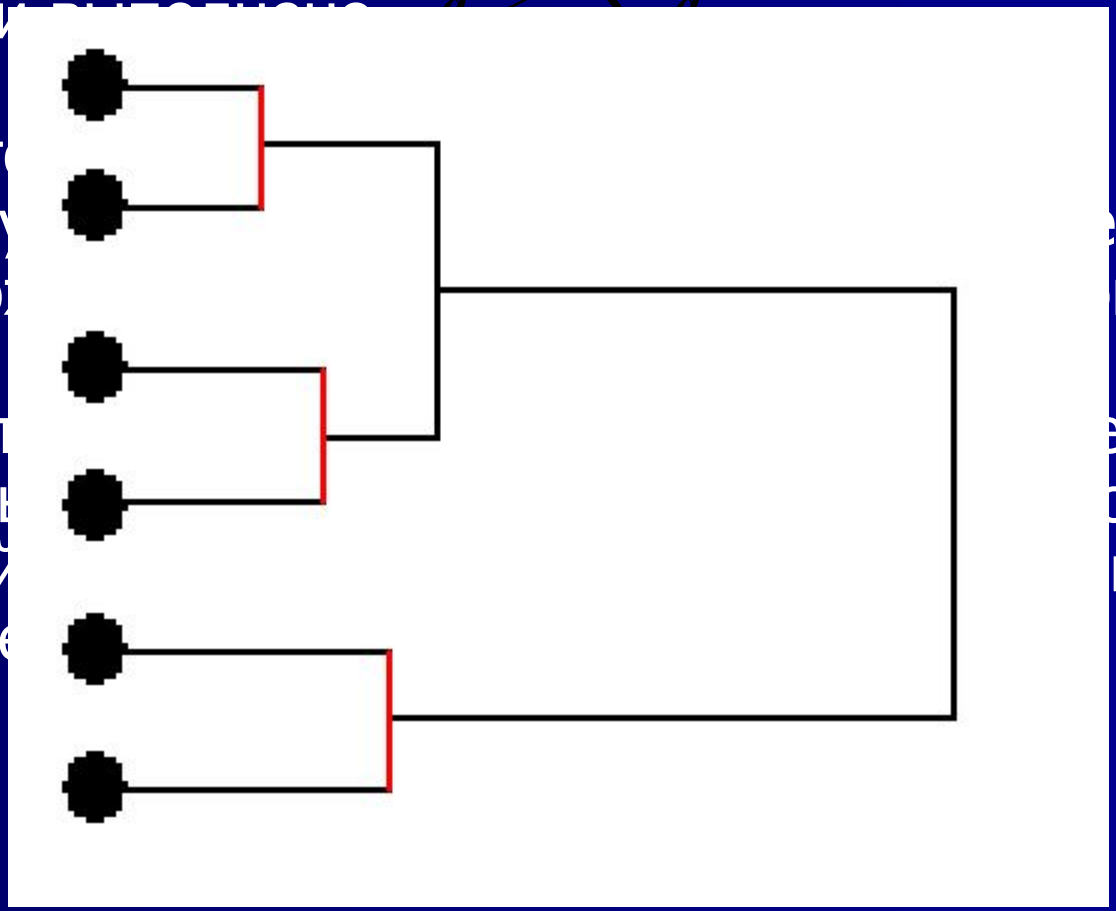


$$q_k = \frac{\Delta t_k}{min_j \Delta t_{i_j}}$$

Алгоритм гравитационной кластеризации: **построение кластеризации**

- Если выполнено $d_{ij} < \frac{1}{s}\sum_{k=1}^{s} d_{ik}$

то г
допу ленное»
мно цедура
для
соот ется
новы олнения
этой набор
непе



Достоинства

- Автоматическое определение числа кластеров
- Независимость настроек алгоритма от входных данных
- Дерево объединения даёт представление о ходе работы алгоритма, что позволяет проанализировать структуру данных
- Устойчивость алгоритма к изменениям входных данных

Устойчивость алгоритма гравитационной кластеризации

Определение *Алгоритм кластеризации устойчив на наборе данных* $\{\vec{X}_1,\ldots,\vec{X}_n\}$ *если существуют такие* $\delta_1>0,\ldots,\delta_n>0$ *что результаты кластеризации на наборах данных* $\{\vec{X}_1+\vec{\delta}_1,\ldots,\vec{X}_n+\vec{\delta}_n\}$ *и* $\{\vec{X}_1,\ldots,\vec{X}_n\}$ *совпадают, где* $|\vec{\delta}_1|<\delta_1,\ldots,|\vec{\delta}_n|<\delta_n$.

Определение *Дерево объединений для набора данных* $\{\vec{X}_1,\ldots,\vec{X}_n\}$ *будем называть устойчивым, если существуют* $\delta_1>0,\ldots,\delta_n>0$ *такие, что дерево объединений для набора данных* $\{\vec{X}_1,\ldots,\vec{X}_n\}$ *подобно дереву объединений для набора данных* $\{\vec{X}_1+\vec{\delta}_1,\ldots,\vec{X}_n+\vec{\delta}_n\}$ *где* $|\vec{\delta}_1|<\delta_1,\ldots|\vec{\delta}_n|<\delta_n$

Теорема (Устойчивость дерева объединений)

Теорема (Устойчивость алгоритма гравитационной кластеризации.)

Замечание *Теорема верна для полной гравитационной кластеризации и для алгоритма по m ближайшим точкам.*

Гравитационная кластеризация по m ближайшим точкам

Основная идея - учитывать влияние только m “ближайших” точек

$\{\vec{X}_{i_j}\}_{j=1}^{m}$ - m “ближайших” для точки $\vec{X}_i$

Пусть точки $\vec{X}_p$ и $\vec{X}_q$ объединяются

- первая стратегия

  если $\vec{X}_p \in \{\vec{X}_{i_j}\}_{j=1}^{m}$ или $\vec{X}_q \in \{\vec{X}_{i_j}\}_{j=1}^{m}$

  то пересчет m ближайших для точки $\vec{X}_i$
- вторая стратегия

  если $\vec{X}_q = \vec{X}_{i_s}$ то $\{\vec{X}_{i_1}, \ldots, \vec{X}_{i_{s-1}}, \vec{X}_p, \vec{X}_{i_{s+1}}, \ldots, \vec{X}_{i_m}\}$

Гравитационная кластеризация с использованием CF-дерева

CF-дерево, построенное по последовательности из n точек с параметрами B и T имеет максимальную глубину $h \le \frac{n-2}{B-1}$

минимальная глубина при T=0 $h \ge \log_B n - 1$

Основная идея – разбиваем множество на m групп, учитываем влияние на точку только точек из этой группы и остальных групп.

$$f(n_1,\ldots,n_m) = \sum_{i=1}^{m} n_i(n_i - 1 + m - 1) = \sum_{i=1}^{m} n_i^2 - \sum_{i=1}^{m} n_i + (m-1)n$$

$$(n_0+1,\ldots,n_0+1,n_0,\ldots,n_0),\ n_0 = \left\lfloor \frac{n}{m} \right\rfloor$$

Модификации алгоритма гравитационной кластеризации

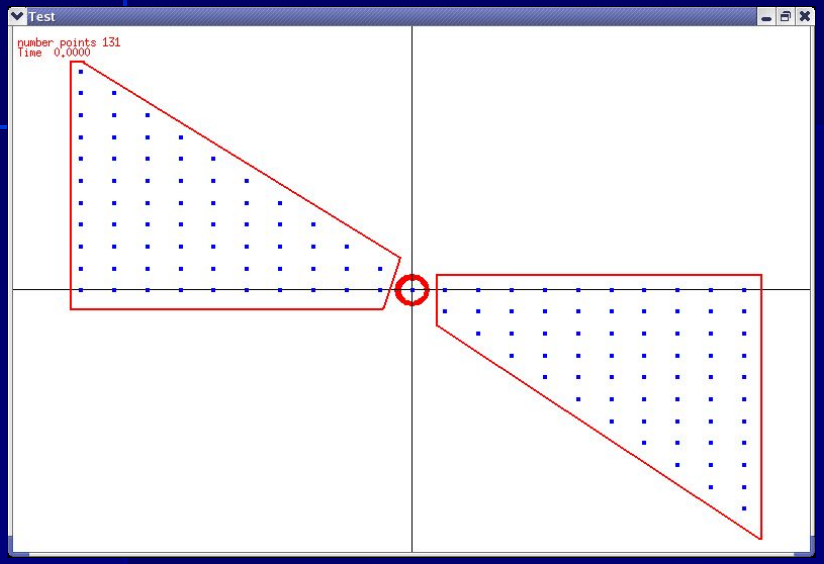
Предложены следующие модификации алгоритма

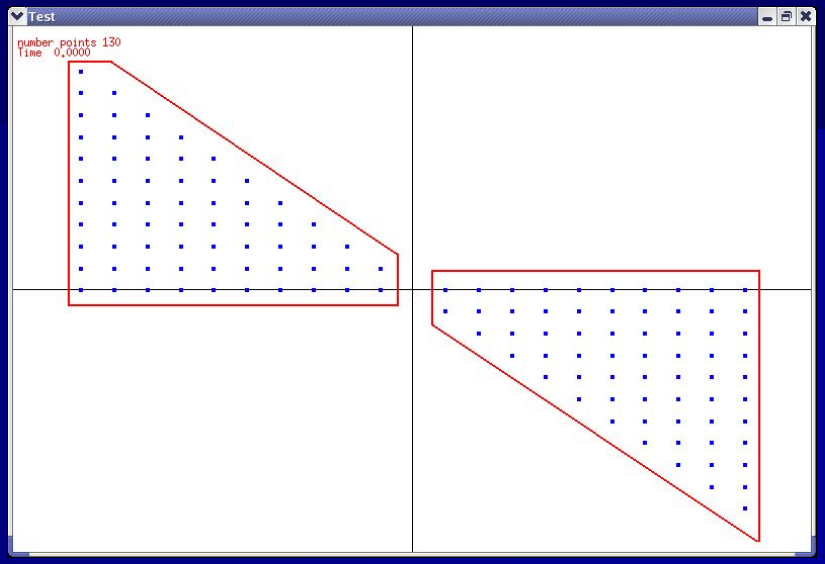
- Алгоритм гравитационной кластеризации по m ближайшим точкам
- Алгоритм гравитационной кластеризации с использованием CF- и R- деревьев
- Алгоритм гравитационной кластеризации с разбиением каждой группы точек на кластеры

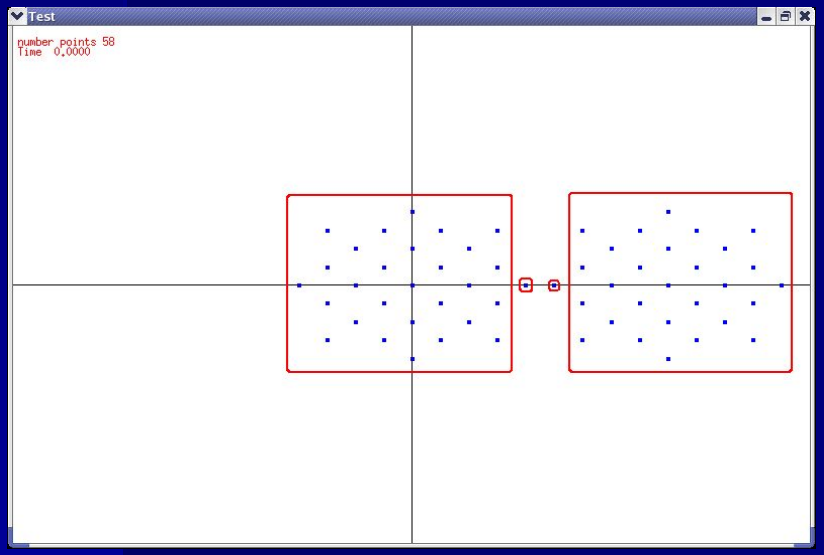
Оценка алгоритмической сложности

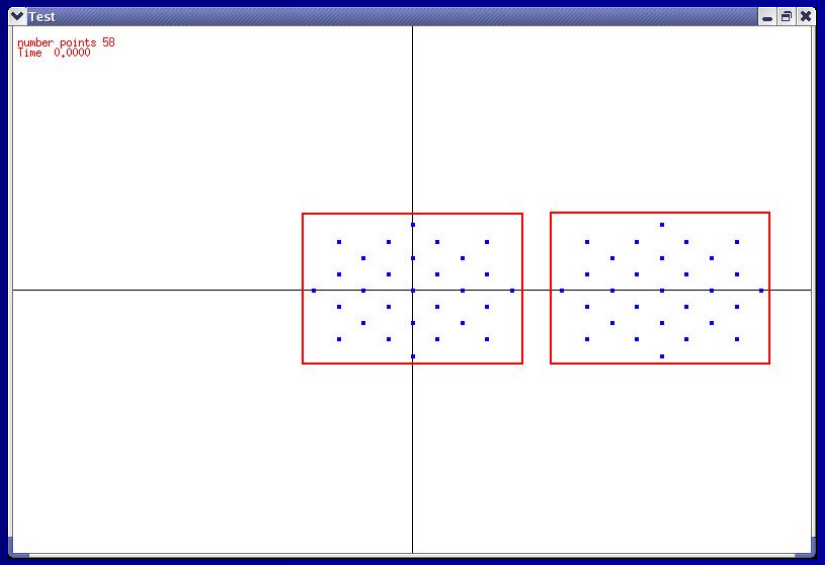
- Алгоритм гравитационной кластеризации $O(n^3 k)$
- Алгоритм гравитационной кластеризации по m-ближайшим $O(mn^2 k).$
- Алгоритм гравитационной кластеризации с использованием CF- и R- деревьев $O(n^2 \sqrt{n} k)$
- Алгоритм гравитационной кластеризации с разбиением каждой группы точек на кластеры ($O(n^2 k)$ - полный алгоритм, по 1-й ближайшей точке – $O(n^{4/3} k)$)

Результаты работы гравитационного алгоритма

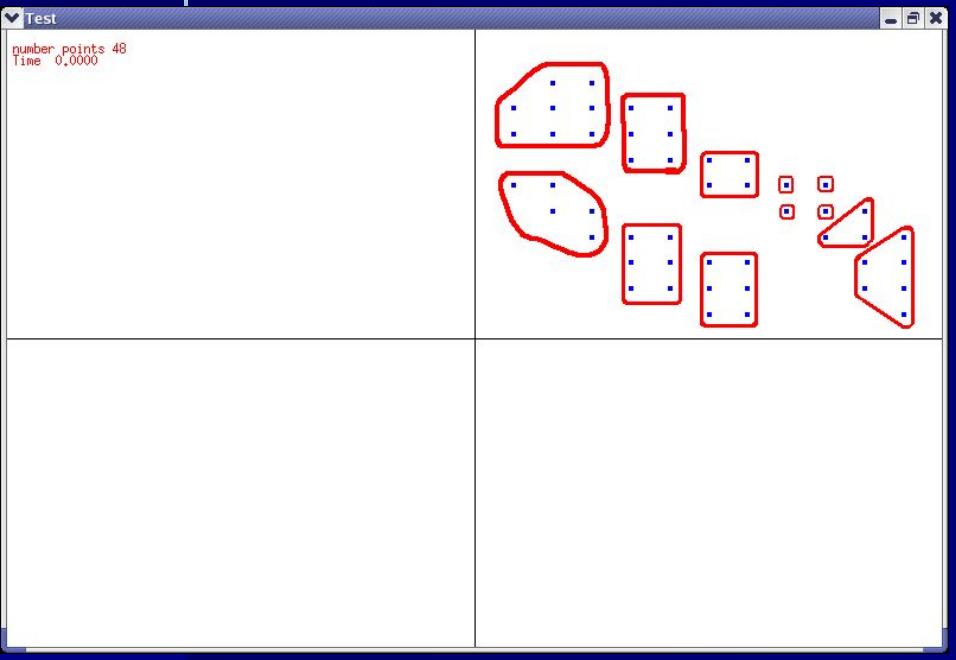


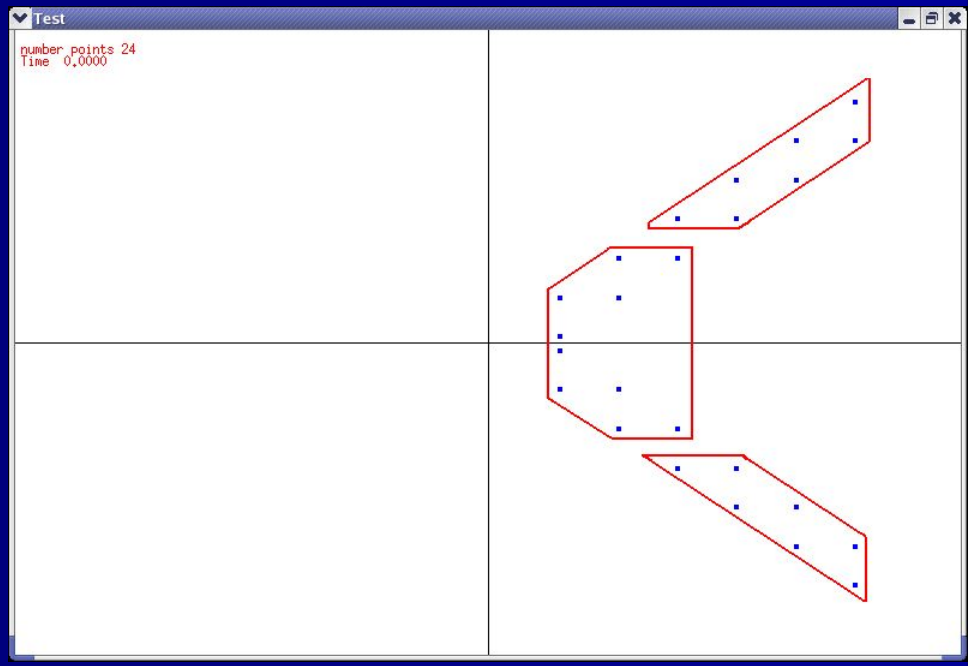






Результаты работы гравитационного алгоритма





Области применения: обработка потоковой информации

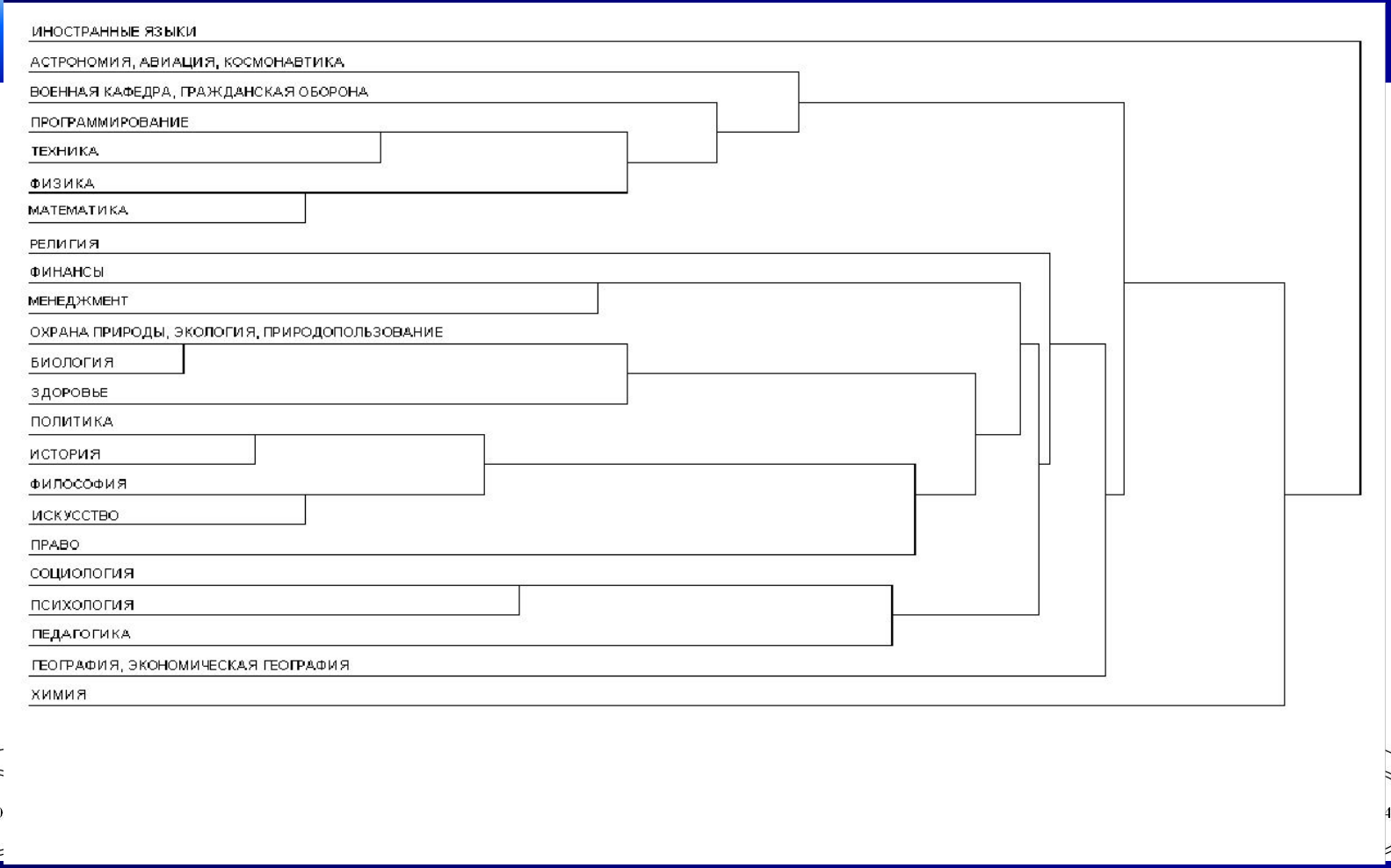
Рассматривается задача построения иерархии рубрик для системы рубрикации текстов

При обучении системы на вход модулю обучения подается массив обучающих текстов, по которым в соответствие каждой рубрике ставится массив статистических величин, называемых статистическим портретом рубрики. Каждый статистический портрет представляется точкой в многомерном пространстве с единичной массой, и на основе алгоритма гравитационной кластеризации автоматически строится дерево рубрик (рис). Следует отметить, что качество рубрикации с использованием такого автоматически построенного дерева рубрик, как правило, лучше, чем с использованием дерева, построенного человеком на основе его представления о смысловых взаимосвязях рубрик.

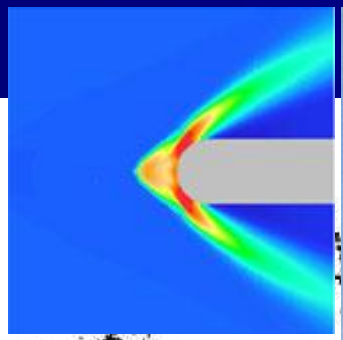
После построения дерева в процессе рубрикации вероятностный алгоритм производит спуск по дереву рубрик производя на каждом шаге процедуру выбора из небольшого количества вариантов на основе заданных обучающих выборок текстов.

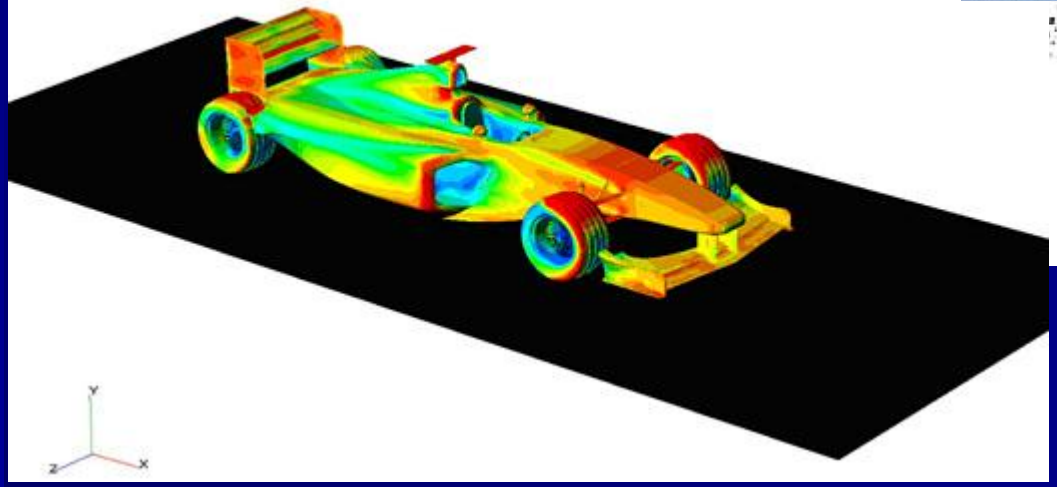
Области применения: обработка потоковой информации

Проведенные испытания показывают, что использование дерева повышает точность определения рубрики при большом количестве рубрик с 60-65% до 87-92%



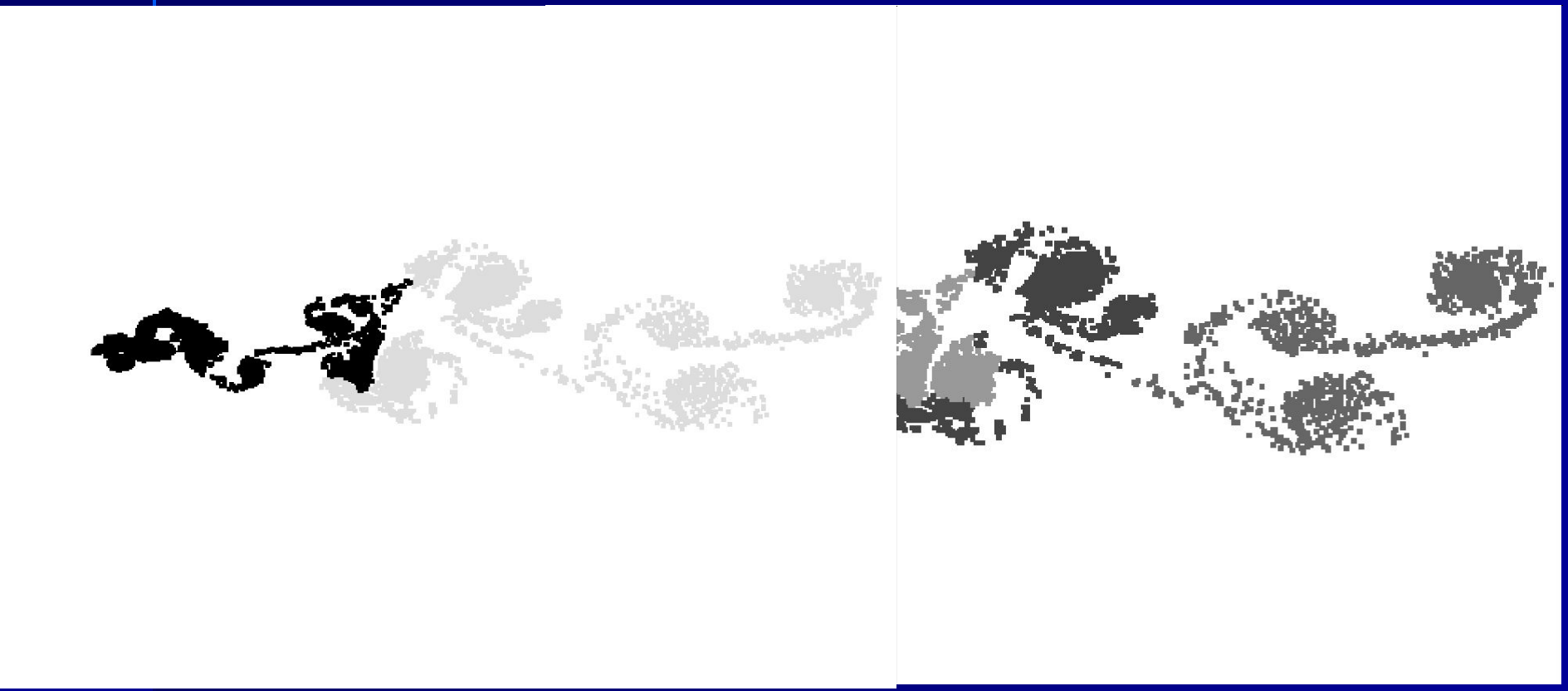
Области применения: **Минимизация вычислений в задаче обтекания**





Минимизация вычислений в задаче обтекания

Цель: разбиение множества вихрей на кластеры, которые изолированы друг от друга, где число кластеров равняется числу процессоров. При этом в каждой кластере разбиение вихрей на компактные группы.



Пример разбиения множества вихрей на группы

Задача визуализации

Сопоставление набору точек из многомерного пространства точек на плоскости с качественным отображением:

1) кластерной структуры;
2) расположения данных в исходном пространстве;
3) взаимосвязи между точками.

Требования к алгоритму визуализации

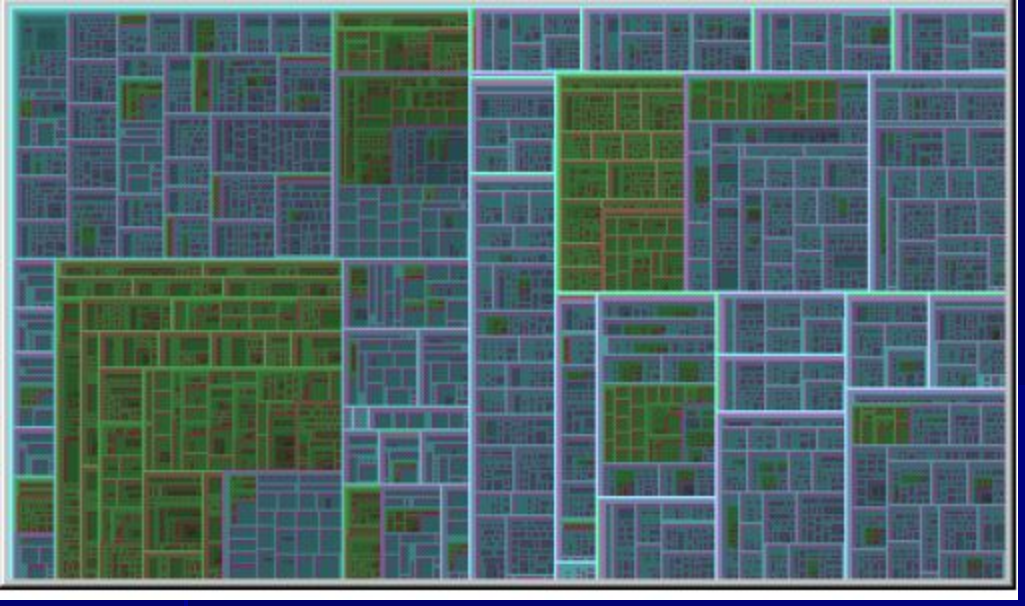
1) интуитивно понятное изображение
2) простота в навигации по данным
3) эффективное использование области для изображения
4) отображение взаимосвязи между данными
5) отображение пространственной структуры данных

Существующие подходы

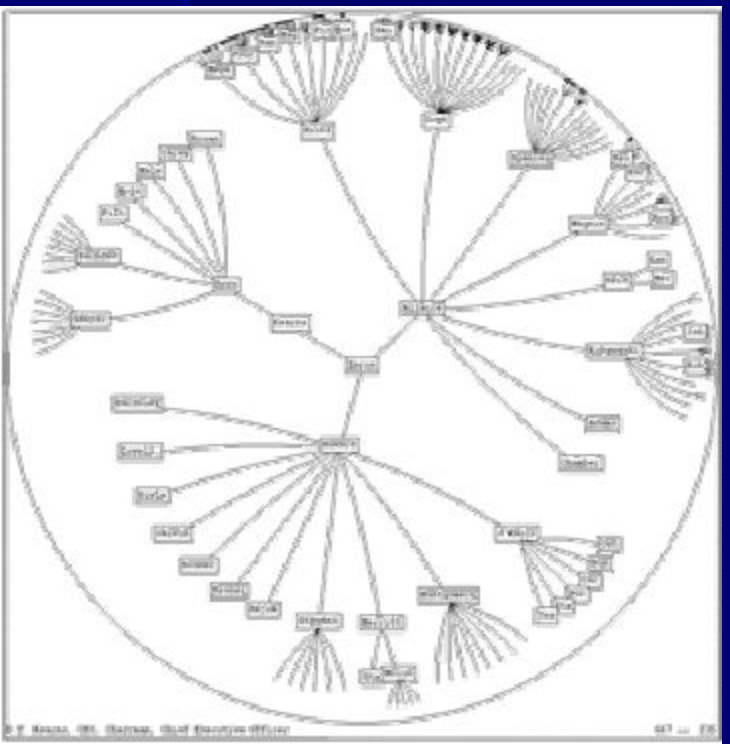
1)Многомерное шкалирование

$$\sum_{j=1}^{n}\sum_{i=1}^{n}\left(\frac{d_2(\vec{X}_i,\vec{X}_j)-a_{ij}}{a_{ij}}\right)^2$$

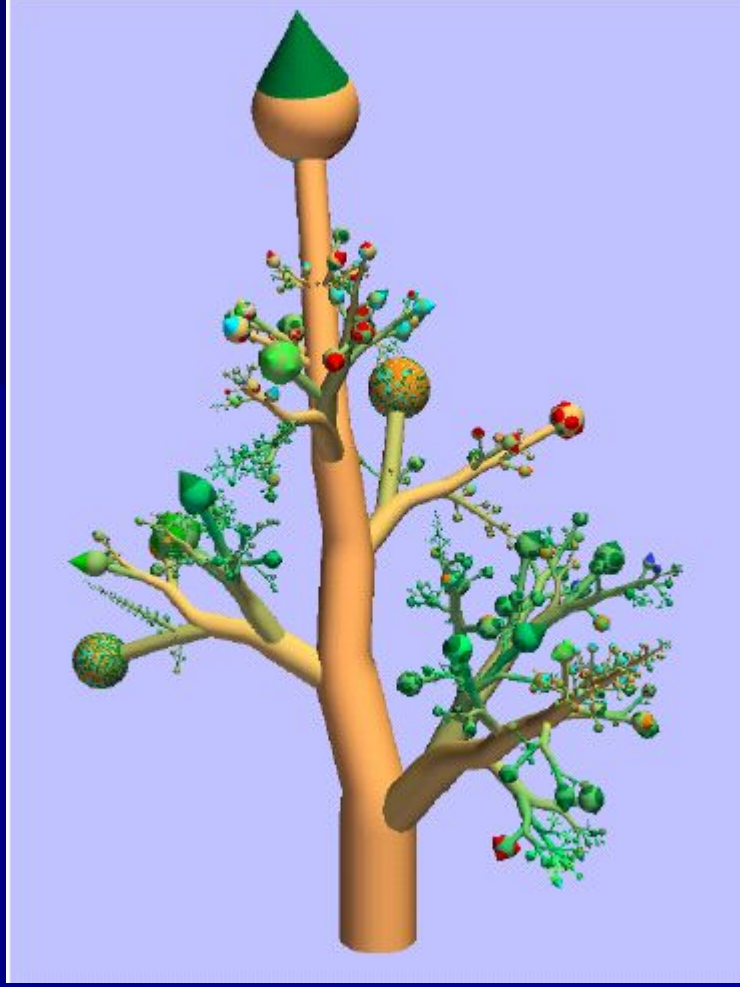
2) TreeMaps
3) Botanical Tree
4) Star Tree
5) Hyperbolic Display
6) Визуализация графов



Treemaps



Star Tree



Botanical Tree

Соответствие предъявленным требованиям существующих методов

	Интуит. понятная визуализация	Простота в навигации	Эфф. испол. прост-ва	Отобр-е взаимосвязи	Отобр-е кластерной структуры данных
Многом. шкалир.	+	-	+	+	-
TreeMaps	-	+	+	-	-
Botanical Tree	+	+	+	-	-
Star Tree	+	-	+	-	-
Hyper Tree	+	+	+	-	-
Граф	+	+	+	+	-

Визуализация многомерных данных

Определение Визуализация — множество $V = \{P_1, \ldots, P_l\}$, где $P_i = \{\overline{X}_{i_1}, \ldots, \overline{X}_{i_k}\}, \overline{X}_{i_j} \in X.$

Определение Визуализатор — множество визуализаций, т.е. $\{V_i\}_{i=1}^{s}$. Число s называется размером визуализатора.

Утверждение Размер каждого визуализатора, состоящего из различных визуализаций, не превосходит 2^{2^n}

Множество визуализаций, соответствующих каждой вершине дерева объединений, образует визуализатор, называемый деревом визуализации.

Утверждение Размер дерева визуализации не превосходит $2n-1$

Дерево визуализации

- Каждая визуализация отображается как набор точек.
- Конкретное состояние визуализатора – некоторая конфигурация точек на плоскости.
- Визуализатор – интерактивное отображение состояний визуализатора со следующим интерфейсом:
  - Переход на нижний уровень (отображение внутренней структуры данных, соответствующих точке, и их окружения)
  - Переход на верхний уровень (уменьшение масштаба данных)
  - Изменение «точки в центре» (перемещение по данным)

Составляющие предлагаемого подхода

- Отображение данных, находящихся в “центре” и группировка данных, находящихся вне “центра”, с использованием иерархии.
- Инструменты модификации: раскрытие точки, расположение любой видимой точки в “центре”.
- Отображение небольшого числа точек за счет группировки данных.
- Взаимосвязь между данными определяется расстоянием между ними.
- Отображение структуры данных, используя иерархию данных, построенной гравитационным алгоритмом кластеризации.

Построение дерева визуализации (визуализатора)

При построении визуализатора на основе дерева объединений выполняются следующие шаги:

- Сопоставление визуализации точек на плоскости с помощью минимизации функции

$$S(\vec{x}_1,\dots,\vec{x}_n)=\sum_{j=1}^{n}\sum_{i=1}^{n}\left(\frac{d(\vec{x}_i,\vec{x}_j)-a_{ij}}{a_{ij}}\right)^2$$

- Сопоставление визуализаций. Основные шаги:
  - Выполнение смещения $\vec{x}_1+t,\dots,\vec{x}_n+t,$ где $t=\vec{x}_c-\vec{x}_1,$
  - Выполнение поворота

Обеспечение плавности перехода от одной конфигурации к другой

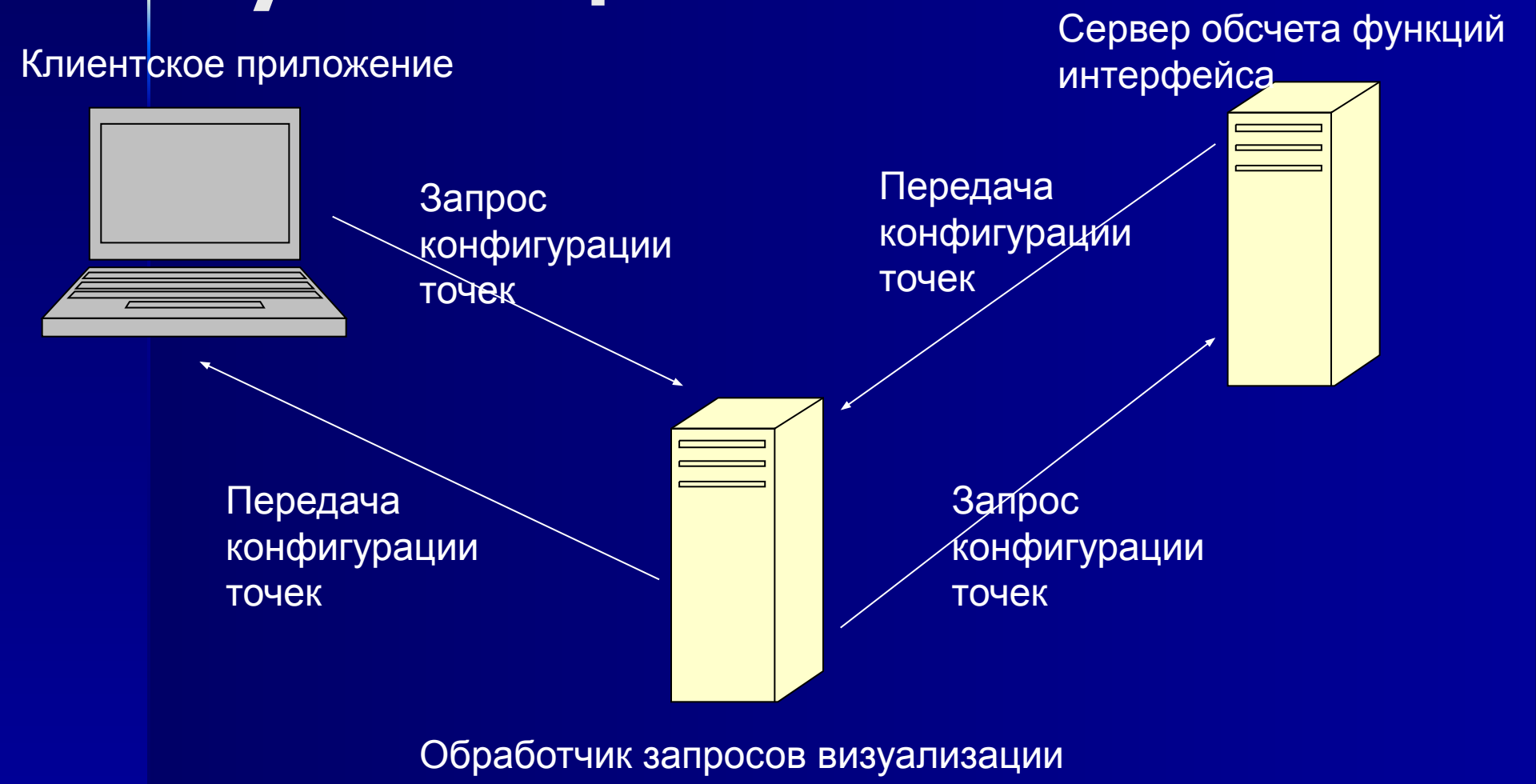
- **Смещение**
  - В любой конфигурации всегда должна быть «точка в центре» с некоторыми фиксированными координатами
- **Поворот**
  - Осуществление поворота таким образом, чтобы сохранялась направленность данных

Структура хранения визуализатора

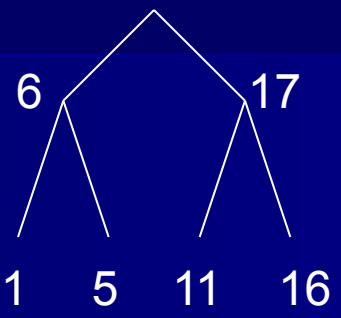
Структура хранения визуализации данных – дерево (дерево визуализации), вершины которого содержат следующую информацию:

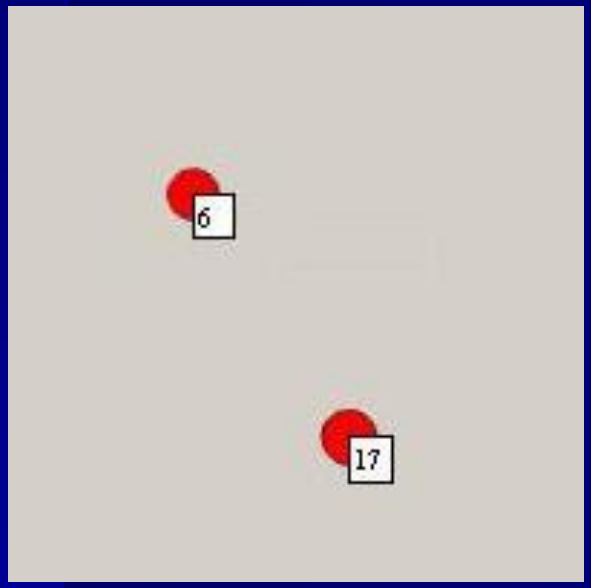
- указатель на родителя;
- указатели на детей;
- указатели на вершины, отображаемые вместо с данной;
- координаты вершин, отображаемых вместе с данной (визуализация).

Схема взаимодействия элементов системы визуализации

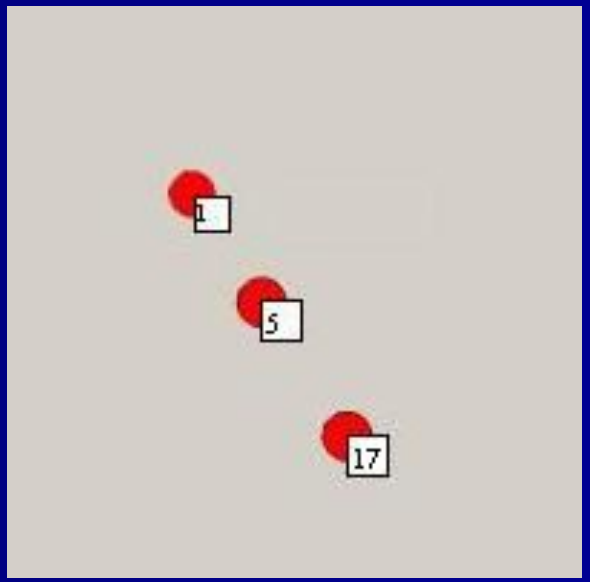


Пример





Раскрытие точки 6



Спасибо за внимание