Кластеризация

Clustering

http://mmds.org

Mining of Massive Datasets

Jure Leskovec, Anand Rajaraman, Jeff Ullman

Данные большой размерности

- По облаку точек данных требуется понять его структуру

Задача кластеризации

- Дан набор точек, с понятием расстояния между точками, сгруппировать точки в некоторое число кластеров, таким образом, чтобы
  - Члены кластера были близки/похожи друг на друга
  - Члены разных кластеров были не похожи
- Как правило:
  - Точки находятся в пространстве большой размерности

- Схожесть определяется с помощью меры расстояний
  - евклидова, косинусная, Жаккара, редакционное расстояние, ...

Почему кластеризация – сложная задача?

- Кластеризация в двух измерениях выглядит просто
- Кластеризация малых объемов данных выглядит просто
- В большинстве случаев это так и есть
- Многие приложения привлекают не 2, а 10 или 10000 измерений

- Пространства высокой размерности выглядят по-другому: Почти все пары точек расположины примерно на одном расстоянии

Задача кластеризации: галактики

- Каталог 2 миллиардов «небесных объектов» представляет объекты по их излучению в 7 измерениях (частотных диапазонах)

- Задача: Кластеризовать по схожим объектам, например, галактики, находящиеся рядом звезды, и т.д.

- Sloan Digital Sky Survey

Задача кластеризации: музыкальные компакт-диски

- Интуитивно: музыка делится на категории, и покупатели предпочитают несколько категорий

  – Но что такое категории в действительности?

- Представим компакт-диск по множеству покупателей, которые его купили

- Похожие компакт-диски имеют похожий набор покупателей

Задача кластеризации: музыкальные компакт-диски

Пространство всех компакт-дисков:

- Представим себе пространство с одним измерением для каждого покупателя
  - Значения в измерении могут быть только 0 или 1
  - Компакт-диск – это точка в этом пространстве $(x_1, x_2, \ldots, x_k)$, где $x_i = 1$ ттогда i-й покупатель купил этот компакт-диск

- Для Amazon размерность составляет десятки миллионов

- Задача: найти кластеры схожих компакт-дисков

Задача кластеризации: документы

- Представим документ как вектор $(x_1, x_2, \ldots, x_k)$, в котором $x_i = 1$ ттогда i-е слово (по некоторому порядку) встречается в документе
  - В действительности не имеет значения, что k бесконечно; т.е. мы не ограничиваем набор слов
- Документы со схожими наборами слов могут быть об одной и той же теме

Меры схожести: косинусная, Жаккара и евклидова

- Как и в случае компакт-дисков, у нас имеется выбор, когда мы представляем документы в виде наборов слов или шинглов:
  - Наборы как векторы: Мера схожести – косинусное расстояние
  - Наборы как множества: Мера схожести – расстояние Жаккара
  - Наборы как точки: Мера схожести – евклидово расстояние

Обзор: методы кластеризации

- Иерархические:
  - Агломерационный (снизу вверх)
    - Изначально каждая точка – кластер
    - Многократно объединяем два «ближайших» кластера в один
  - Разделяющий (сверху вниз)
    - Начать с одного кластера из рекурсивно его разбивать
- Присвоение точек

– Хранить набор кластеров

– Точки принадлежат «ближайшему» кластеру

Иерархическая кластеризация

- Ключевая операция: Многократное объединение двух ближайших кластеров

- Три важных вопроса:
  - Как представить кластер, состоящий из более одной точки?
  - Как определить «близость» кластеров?
  - Когда прекращать объединение кластеров?

Иерархическая кластеризация

- Ключевая операция: Многократное объединение двух ближайших кластеров

- (1) Как представить кластер из множества точек?

  – Ключевая проблема: При объединении кластеров, как нам представить «расположение» кажого кластера, чтобы определить, какая пара кластеров наиболее близка?

- Евклидов случай: у каждого кластера есть центроида = среднее его точек

- (2) Как определить «близость» кластеров?
  - Измерять расстояние между кластерами по расстоянию между центроидами

А что в неевклидовом случае?

А что в неевклидовом случае?

- Единственные «расположения», о которых мы можем говорить, – это сами точки
  - т.е., нет никакого «среднего» между двумя точками
- Подход 1:
  - Как представить кластер множества точек? Кластроида = точка, «ближайшая» к другим точкам

– Как определить «близость» кластеров? Относится к кластроиде так, как если бы это была центроида, при вычислении расстояний между кластерами

«Ближайшая» точка?

- Как представить кластер множества точек? Кластроида = точка, «ближайшая» к другим точкам

- Возможные значения слова «ближайшая»:
  - Наименьшее максимальное расстояние до других точек
  - Наименьшее среднее расстояние до других точек
  - Наименьшая сумма квадратов расстояний до других точек

* Для метрики расстояния d кластроида c кластера C – это $\min_c \sum_{x \in C} d(x, c)^2$

Определение «близости» кластеров

- (2) Как определить «близость» кластеров?
  - Подход 2:
    Межкластерное расстояние = минимум расстояний между любыми двумя точками, по одной из каждого класса
  - Подход 3:
    Введем понятие «связности» (cohersion) кластеров, например, максимальное расстояние от кластроиды
    * Объединять кластеры, объединение которых получается наиболее связным

Связность (Cohersion)

- Подход 3.1: Использовать диаметр объединяемого кластера = максимальное расстояние между точками кластера

- Подход 3.2: Использовать среднее расстояние между точками кластера

- Подход 3.3: Использовать подход, основанный на плотности

  – Взять диаметр среднего расстояния, например, и разделить на число точек в кластере

Реализация

- Наивная реализация иерархической кластеризации:
  - На каждом шаге вычислить попарные расстояния между всеми парами кластеров, затем выполнить объединение
  - $O(N^3)$
- Тщательная реализация с ипользованием приоритетную очередь (priority queue) позволяет уменьшить время до $O(N^2 \log N)$

– Все равно слишком затратно для по-настоящему больших наборов данных, не помещающихся в памяти

Метод k-средних

(k-means clustering)

Алгоритм(ы) k-средних

- Предполагает евклидово пространство/расстояния
- Начать с выбора k, числа кластеров
- Инициализировать кластеры путем выбора одной точки на кластер
  - Например: Выбрать случайно одну точку, затем $k - 1$ других точек, каждую как можно дальше от предыдущих точек

Наполнение кластеров

- 1) Для каждой точки, поместить ее в тот кластер, центроида которого ближе всего
- 2) После того, как все точки распределены, обновить расположения центроид k кластеров
- 3) Перераспределить все точки к их ближайшим центроидам
  - Иногда перемещает точки между кластерами
- Повторять 2) и 3) до сходимости

– Сходимость: точки не «бегают» между кластерами и центроиды стабилизируются

Выбор правильного k

Как выбрать k?

- Пробовать различные k, наблюдая за изменением среднего расстояния до центроиды при увеличении k

- Среднее быстро убывает до подходящего k, затем изменяется мало

Алгоритм BFR

Обобщение k-средних на большие данные

Алгоритм BFR

- BFR [Bradley-Fayyad-Reina] – это вариант метода k-средних, предназначенный для обработки очень больших (хранящихся на диске) наборов данных

- Предполагает, что кластеры нормально распределены вокруг центроиды в евклидовом пространстве

  – Стандартные отклонения могут быть разными в разных координатах

    * Кластеры – эллипсы, ориентированный по осям

- Эффективный способ подытожить кластеры (требуемая память – O(кластеров), а не O(данных))

Алгоритм BFR

- Points are read from disk one main-memory-full at a time

- Большинство точек из предыдущих загрузок памяти подытоживаются простыми статистиками

- Для начала, из первоначальной загрузыки выберем начальные k центроиды некоторым sensible образом:

  – Возьмем k случайных точек

– Возьмем малую случайную выборку и кластер оптимальным образом

– Возьмем выборку; выберем случайную точку, а затем еще $k-1$ точек, каждую – настолько далеко от выбранных точек, насколько возможно

Три класса точек

Три набора точек, за которыми мы следим

- Discard set (DS):
  - Точки, достаточно близкие к центроиде, чтобы быть подытоженными
- Compression set (CS):
  - Группы точек, которые достаточно близки друг к другу, но не близки ни к какой существующей центроиде

- Эти точки подытоживаются, но не относятся ни к какому кластеру.

- Retained set (RS):
  - Изолированные точки, ожидающие того, чтобы быть отнесенными к compression set

Подытоживание набора точек Для каждого кластера DS подытоживается по:

- Количество точек N
- Вектор SUM, i-я компонента которого – это сумма координат точек в i-м измерении
- Вектор $SUMQ$: i-я компонента = сумма квадратов координат в i-м измерении

Подытоживание точек: комментарии

- $2d + 1$ значений представляют кластер любого размера
  - $d =$ количество измерений
- Среднее в каждом измерении (центроида) может быть вычислена как SUM_i/N
  - $SUM_i = i$-я компонента вектора SUM
- Дисперсия множества DS кластера в размерности i – это: $(SUMSQ_i/N) - (SUM_i/N)^2$

  - А стандартное отклонение – это квадратный корень из этого значения

- Следующий шаг: Собственно кластеризация

«Загрузка в память» точек

Обработка «загрузки в память» точек (1):

- 1) Найти те точки, которые «достаточно близки» к центроиде кластера и добавить эти точки в этот кластер и DS
  - Эти точки настолько близки к центроиде, что их можно подытожить и затем отбросить
- 2) Использовать любой алгоритм кластеризации, выполняющийся в оперативной памяти, для того, чтобы разбить на кластеры оставшиеся точки и старый RS

– Кластеры идут в CS; точки-выбросы идут в RS

«Загрузка в память» точек

Обработка «загрузки в память» точек (2):

- 3) Множество DS: «Подкрутить» статистики кластеров, чтобы учесть новые точки
  - Добавить N'ы, SUM'ы, $SUMSQ$'ы
- 4) Рассмотреть возможность объдинения сжатых множеств в CS
- 5) Если это последний проход, объединить все сжатые множества в CS и все точки RS в их ближайший кластер

Немного подробностей...

- Q1) Как понять, что точка «достаточно близка» к кластеру настолько, что мы должны ее добавить в кластер?

- Q2) Как понять, что два сжатых множества (CS) могут быть объединены в одно?

Насколько близко «достаточно близко»?

- Q1) Нам нужен метод принятия решения о том, чтобы добавить новую точку в кластер (и выбросить ее)

- BFR предлагает два метода:
  - Расстояние Махаланобиса меньше, чем порог
  - Большое правдоподобие того, что точка принадлежит к ближайшей, в настоящее время, центроиде

Расстояние Махаланобиса

- Нормированное Евклидово расстояние от центроиды

- Для точки $(x_1, \ldots, x_d)$ и центроиды $(c_1, \ldots, c_d)$

  1. Нормировать в каждом измерении: $y_i = (x_i - c_i)/\sigma_i$

  2. Взять сумму квадратов y_i

  3. Взять квадратный корень

$$d(x, c) = \sqrt{\sum_{i=1}^{d} \left(\frac{x_i - c_i}{\sigma_i} \right)^2}$$

σ_i= стандартное отклонение точек в кластере по i-й размерности

Расстояние Махаланобиса

- Если кластеры нормально распределены в d измерениях, то после преобразований одно стандартное отклонение $= \sqrt{d}$
  - т.е., у 68% точек кластера расстояние Махаланобиса $< \sqrt{d}$
- Принять точек в кластер, если ее расстояние Махаланобиса $<$ некоторого порога, например, 2 стандартных отклонения

Следует ли объединять 2 CS-кластера?

Q2) Следует ли объединять 2 CS-подкластера?

- Вычислить дисперсию объединенного подкластера
  - N, SUM, $SUMSQ$ позволяют выполнить это вычисление быстро
- Объединить, если объединенная дисперсия ниже некоторого порога
- Множество альтернатив: По-разному обращаться с измерениями, учитывать плотность

Алгоритм CURE

Обобщение метода k-средних на кластеры произвольной формы

Алгоритм CURE

- Проблемы с BFR/k-means:
  - Предположение, что кластеры нормально распределены в каждом измерении
  - И оси фиксированы – эллипсы под углом НЕ допускаются
- CURE (Clustering Using REpresentatives):
  - Предполагает евклидово расстояние
  - Позволяет кластерам принимать любую форму

– Использует набор точек-представителей для представления кластеров

Начало CURE

Алгоритм в два прохода. Проход 1:

- 0) Выбрать случайную выборку точек, умещающуюся в оперативной памяти
- 1) Начальные кластеры:
  - Разбить эти точки на кластеры иерархически – группируем ближайшие точки/кластеры
- 2) Выбрать точки-представители

- Для каждого кластера, произвести выборку точек, настолько разбросанных, насколько возможно

- Из выборки выбрать представителей посредством их перемещения (например) на 20% ближе к центроиде кластера

Окончание CURE

Проход 2:

- Теперь перепросмотреть весь набор данных и перебрать каждую точку p в наборе данных

- Поместить ее в «ближайший кластер»

  – Нормальное определение «ближайшего»: Найти ближайшее представление к p и приписать его к кластеру представителя

Резюме

- Кластеризация:

- Алгоритмы:
  - Агломерационная иерархическая кластеризация:
    - Центроида и кластроида
  - k-means:
    - Инициализация, выбор k
  - BFR

– CURE