ЛОКАЛЬНО-ЧУВСТВИТЕЛЬНОЕ ХЕШИРОВАНИЕ ДЛЯ СЕГМЕНТАЦИИ РЫНКА

Дрозд П. С.
Факультет радиофизики и компьютерных технологий Белорусского государственного университета
Минск, Республика Беларусь
E-mail: drozdps@gmail.com

В этой работе предлагается новый метод сегментации рынка рынка путем кластеризации транзакционных данных клиентов, пользователей или покупателей в любой компании. Данный способ основан на алгоритме локально-чувствительного хеширования, который увеличивает эффективность и быстродействие последующей иерархической кластеризации в несколько раз. Показана высокая устойчивость подхода, алгоритм реализован в системе с использованием фреймворка Apache Spark.

Введение

В теории маркетинга существует понятие сегментации рынка, которое подразумевает разбиение клиентов или потенциальных клиентов на различные осмысленные группы. Решение этой задачи позволяет предприятию оптимальнее таргетировать своё предложение на определённую группу (или сегмент) потребителей. В рамках данной работы исследуются методы сегментации рынка с помощью кластерного анализа. Использование кластеризации позволяет создать и проверить гипотезы о существовании в исследуемой совокупности потребителей однородных групп. Для того чтобы добиться улучшения устойчивости и производительности кластеризации, был использован оригинальный подход на базе комбинации алгоритмов Bisecting Kmeans и локально-чувствительного хеширования. Для проверки предложенного подхода мы использовали открытый набор данных «Ta-Feng» компании ACM RecSys, который содержит информацию о покупках различных товаров, совершённых более чем 32 тысячами уникальных клиентов. Всего в наборе имеется 817741 запись, каждая из которых описывает совершённую покупателем транзацию с помощью 9 характеристик (дата проведения платежа, тип товара, сумма транзакции и т.д.).

I. Локально-чувствительное хеширование

Итак, после предварительных очистки, стандартизации и визуализации данных, было предложено использовать алгоритм локально-чувствительного хеширования (LSH - от англ. Locality-Sensitive Hashing), который традиционно применятся для решения задачи поиска ближайшего соседа[2]. Этот алгоритм позволяет быстро найти «похожие» точки в многомерном пространстве признаков, причем признаки могут быть представлены в различных типах шкал, а записи данных - иметь «выбросы» и/или пропуски. Специфика LSH заключается в том, что этот алгоритм оперирует со специальным набором «плохих» хеш-функций, которые, в отличии от обычных хеш-функций, должны генерировать коллизии на схожих образах. Таким образом, соседние точки скорее всего попадут в одну хеш-корзину. Пусть $\mathbf{x} = (x_1, x_2, ..., x_p)^T$ - точка в исходном p-мерном пространстве признаков, которая в рассматриваемом случае описывает одну транзакцию, совершённую одним клиентом/пользователем/покупателем. Зафиксируем максимальное значение всех координат этой точки: $C = sup\{x_1, x_2, ..., x_p\} + \epsilon, \epsilon \geq 0$. После этого преобразуем $\mathbf{x}$ в новый вектор размерности Cp по следующему правилу:

$$\begin{cases} \mathbf{v}(\mathbf{x}) = \Psi_C(x_1)\Psi_C(x_2)...\Psi_C(x_p) \\ \Psi_C(x_i) = \underbrace{11...11}_{x_i}\underbrace{00...00}_{C-x_i}, \forall i \leq p \end{cases}$$

Функция $\Psi_C(x_i)$ переводит значение i-ой компоненты $\mathbf{x}$ в последовательность из x_i единиц, за которыми следуют $C - x_i$ нолей. Например, если $\mathbf{x} = (3, 4)^T$ и $C = 5$, то $\mathbf{v}(\mathbf{x}) = (1110011110)^T$. Хеш-функция в алгоритме LSH вычисляет своё значение для вектора $\mathbf{x}$ путём конкатенации k битов (параметр локально-чувствительного хеширования) из $\mathbf{v}(\mathbf{x})$, порядковые индексы которых содержатся в предварительно сгенерированном множестве Υ, содержащем k случайных целочисленных элементов из $\{1, 2, 3, ..., Cp\}$. На практике генерируется l множеств $\{\Upsilon_1, \Upsilon_2, ...\Upsilon_l\}$ и соотвественно l хеш-функций, каждая из которых вычисляет своё значение для каждого вектора исходного набора.

II. Bisecting K-means

На предыдущем этапе (LSH) мы выбрали хеш-функции таким образом, чтобы их можно было определить в метрическом пространстве, так как для любого алгоритма иерархической кластеризации необходимо вычислять попарные расстояния между объектами. После этапа LSH все данные оказались разбиты на домены, объединяющие схожие транзакции, и каждому домену соответствовало какое-то значение хеш-функции, то есть если изначально нам необходимо было обрабатывать сотни тысяч объектов в исходном многомерном пространстве признаков, то уже после LSH можно работать с

тысячами или сотнями значений коллизионных хеш-функций. Кроме того, в некоторых случаях размерность значений хеш-функций можно было снизить по сравнению с размерностью исходных данных [2]. Для непосредственного кластерного анализа этих значений был применён алгоритм Bisecting K-means - высокоэффективная иерархическая версия K-means. Этот алгоритм является дивизионным, так как предполагает, что все точки изначально принадлежат одному глобальному кластеру. На каждой итерации он делит текущий кластер на два дочерних с помощью обычного K-means с фиксированным параметром $k = 2$. В некоторых случаях [3], Bisecting K-means на порядки производительнее K-means.

III. Система для сегментации рынка

На рисунке 2 представлена use-case диаграмма системы сегментации рынка, которая основана на предложенном методе. Для распараллеливания вычислительных задач система использует фреймворк Apache Spark.

Рис. 1 – Функциональная схема системы

IV. Результаты

После обработки системой набора из Ta-Feng было получено 6 осмысленных кластеров покупателей, которые представляли различные сегменты рынка (пенсионеры, VIP-клиенты и т.д.), а также также представлены ориентировочные советы (marketing insights) по оптимизации маркетинга в отношении этих групп (например, для максимизации прибыли следует предоставлять скидку пенсионерам в выходные и предпраздничные дни, так как средняя корзина в этом случае становится больше).

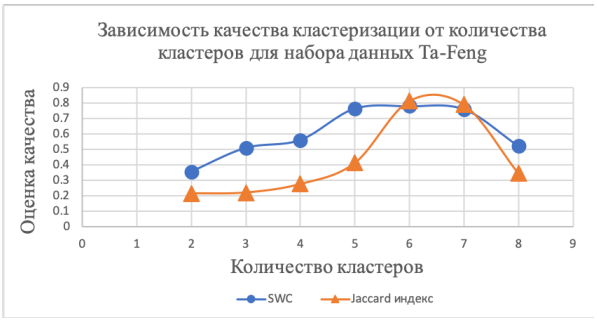

Рис. 2 – Анализ качества сегментации Ta-Feng

Ниже представлена итоговая схема работы предложенного подхода (коллизионное хеширование в LSH и последующая иерархическая кластеризация).

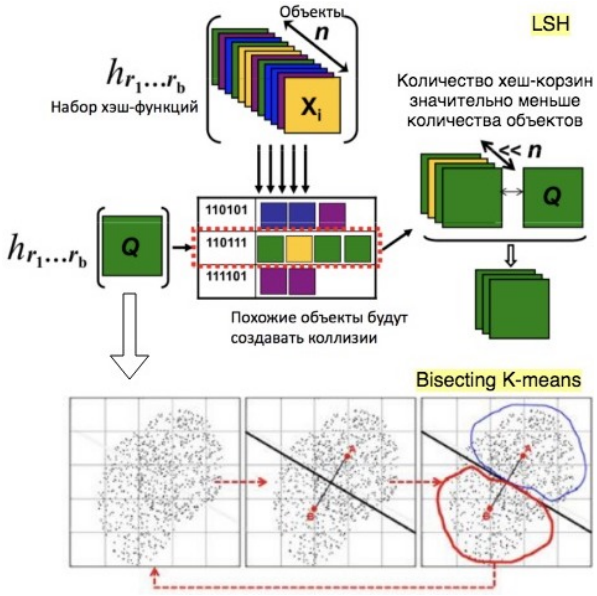

Рис. 3 – Схема предложенного метода

Заключение

Применение алгоритма локально-чувствительного хеширования непосредственно перед иерархической кластеризацией позволяет значительно снизить количество обрабатываемых объектов, а также (в некоторых случаях) снизить размерность пространства признаков. Разработана облачная система, демонстрирующая производительность и устойчивость описанного метода.

Список литературы

1. Drozd P. Locality-Sensitive Hashing for Customer Segmentation / P. Drozd // Open Readings 2019 abstract book / ed. E. Skliutas. – Vilnius, 2019. – P.103.
2. Koga, H. Hierarchial Clustering Algorithm Using Locality-Sensitive Hashing / H. Koga, T.Ishibashi, T. Watanabe // Discovery Science, 7 th International Conference, Padova, 2-5 oct. 2004 / ed. E. Suzuki. – Padova, 2014. – P.114-128
3. Fern, X.Z., Clustering ensembles for high dimensional data clustering / X.Z. Fern, C.E. Brodley // In Proc. International Conference on Machine Learning / ed. T. Fawcett. – Washington DC, 2003. – P.178-185