Создание акватории. Для построения трехмерной модели акватории был использован игровой движок "Unity 3D", поддерживающий возможность создания скриптов на языке C# и импорт моделей из "Blender". В Unity присутствуют различные возможности для создания ландшафта, такие как создание модели вручную и при помощи карт высот. В данном примере использовалось ручное создание небольшого участка акватории, ограниченного скалами и различными природными объектами (рис. 2).

Рисунок 2. Трехмерная модель акватории

Для достижения большей реалистичности изображения на модель ландшафта были наложены текстуры камня и травы из стандартной библиотеки Unity, а так же добавлены различные 3D объекты, такие как деревья, кусты и т.д. Кроме того были использованы шейдеры для создания подводной обстановки. Предусмотрена возможность управления аппаратом с камерой от третьего лица, для которой включен параметр "дальность видимости" реализованный при помощи шейдера GlobalFog. При помощи контроллеров Rigidbody и Mesh Collider для объектов были реализованы такие физические свойства, как возможность столкновения с другими объектами и рельефом, и воздействие гравитации.

Заключение. В результате работы была воссоздана высокополигональная анимированная трехмерная модель аппарата с заданными параметрами материалов, которую можно использовать при моделировании сцены с различными процессами, а также реализована тестовая модель акватории с возможностью управления аппаратом.

ЛИТЕРАТУРА

1. Базовый курс Blender//Blender3D. URL: http://blender3d.com.ua/blender-basics/ (дата обращения: 14.03.2016).
2. Обучающие материалы по Unity//Unity-learn. URL: https://unity3d.com/ru/learn/tutorials (дата обращения: 14.03.2016).

РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ ПО ПОДБОРУ МУЗЫКИ

А.Я. Браневский, А.М. Бесчетников
(г. Томск, Томский политехнический университет)
e-mail: bronzspawn@gmail.com, branevskij_aj@bw-sw.com

RECOMMENDER SYSTEM FOR MUSIC SELECTION

Y. Branevsky, A.M. Beschetnikov
(Tomsk, Tomsk Polytechnik University)

Development of advisory system for the selection of music, using collaborative filtering algorithm.

Recommender system, music selection, collaborative filtering, SPARK.

Введение. С ростом мировой экономики и соответствующего роста товарооборота, появляется потребность в совершенствовании «предложения» за счет алгоритмов, обеспечивающих подбор и рекомендацию товара целевому клиенту, который в нем заинтересован. Такой подход повышает конверсию и эффективность рекламы и интернет-магазинов, то есть увеличивает показатель перехода посетителей в покупателей.

Алгоритмы, основанные на коллаборативной фильтрации, используя статистику пользователя на веб-ресурсе (его оценки определенной продукции, покупки и т.д.) определяют на ее основе его вкусы и предпочтения. Все это дает возможность в подборе и рекомендации наилучшего предложения.
В наше время такие системы являются частью любого крупного веб-сервиса (Google, Яндекс, Youtube, Ebay и т.д.)

Алгоритм. Допустим у нас имеется какое - то количество триплетов (пользователь - товар - оценка) на их основании можно сформировать матрицу, приведенную ниже. (рис. 1)

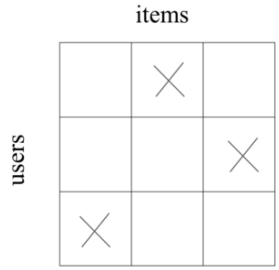


Рис. 1 – Матрица триплетов

Очевидно предположить, что такая матрица будет иметь множество пробелов – пустых ячеек (т.к. всегда будут иметь место неоцененные позиции и их окажется большинство). Таким образом, восстановление таких позиций является нашей задачей (это и будут предсказываемые рекомендации). Восстановим применяя широко-известный подход метод наименьших квадратов.

Для начала введем обозначения:

$$Q_{ui} = \begin{cases} r & \text{if user u rate item i} \\ 0 & \text{if user u did not rate item i} \end{cases}$$

Рис. 2 – Значения оценок

где Qui - оценка пользователем u некоторого предмета i

Также введем матрицу:

$$w_{ui} = \begin{cases} 0 & \text{if } q_{ui} = 0 \\ 1 & \text{else} \end{cases}$$

Рис. 3 – Значения оценок в матрице упрощенного типа

где Wui = 1 если пользователь u оценил предмет i, иначе 0.

Нашей задачей является найти 2 таких вектора X (вектор стоблец) и Y (вектор строка) которые при перемножении дадут матрицу максимально близкую к Q (в общем случае X Y

могут быть матрицами, конечные формулы обобщены и на них в том числе). Запишем соотношение для yi (рис 4.)

$$X * y_i = Q_i$$
$$W_i * X * y_i = W_i * Q_i$$

Рис. 4 – Используемые формулы для расчетов

Qi это некоторое множество оценок всех пользователей предмета i, но т.к. некоторые пользователи не оценивают этот предмет i, можно домножить слева и справа на Wi. Но, такое соотношение не всегда будет иметь решение поэтому воспользуемся следующим подходом.

Найдем проекцию вектора Qi на пространство всех линейных комбинаций вектора X, таким образом найдем приближенный вектор Qproji для которого система будет гарантированно иметь решение так как Qproji будет какой-то из линейных комбинаций вектора X.

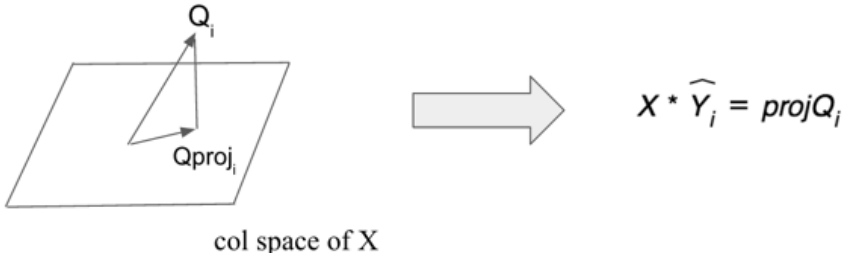
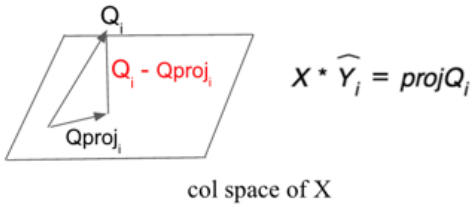


Рис. 5 – Проекция вектора Qi

Далее заметим, что разность Qi – Qproji перпендикулярны любому вектору в col space X. Воспользуемся свойством скалярного произведения получим соотношения для Xu Yi. (В примере ниже рис 6. разобран случай получения Yi Xu получается аналогичным образом)



$$X^T * (Q_i - projQ_i) = 0$$
$$X^T * (Q_i - X * \widehat{Y_i}) = 0$$
$$X^T * Q_i = X^T * X * \widehat{Y_i}$$
$$\widehat{Y_i} = (X^T * X)^{-1} * X^T * Q_i$$
$$\widehat{Y_i} = (X^T * W_i * X)^{-1} * X^T * W_i * Q_i$$

$$\widehat{Y_i} = (X^T * W_i * X)^{-1} * X^T * W_i * Q_i$$
$$\widehat{X_u} = (Y * W_u * Y^T)^{-1} * Q_u * W_u * Y^T$$

Рис. 6 – Математический вывод

На малых объемах данных такой способ будет показывать хороший результат, но при больших объемах могут возникнуть проблемы. Воспользуемся возможностями фреймворка обработки больших данных SPARK. Архитектура его такова, что в процессе вычисления информация разбивается на блоки и каждый обсчитывается на отдельном кластере, далее вся информация сливается на основной кластер и там идет возможная свертка результатов.

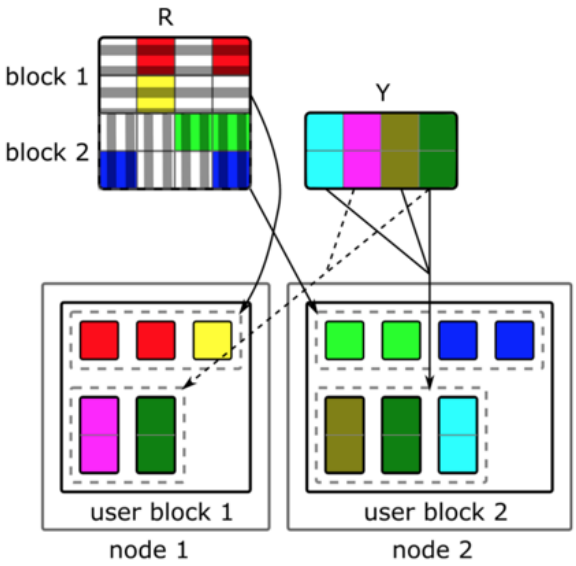


Рис. 7 – Блоки SPARK

На картинке выше показан пример разбиения на блоки вычисления вектора X. Для определенной его компоненты достаточно знать некоторые компоненты вектора Y (т. к. часть из них откидывается при перемножении с матрицей W) и некоторые компоненты матрицы Q. Пользуясь таким подходом можно существенно ускорить вычисления.

Проблемы.
Масштабируемость

С увеличением количества пользователей в системе, остро встает проблема производительности при масштабируемости. Например, имея 30 миллионов слушателей O(M) и миллион исполнителей O(N), алгоритм коллаборативной фильтрации со сложностью равной O(MN) становится слишком сложен для расчётов. Другой проблемой будет сложность в обновлении оценок уже поставленных пользователем, так как стандартная реализация данного алгоритма не предполагает онлайн-изменений.

- Синонимия

Стандартный алгоритм никак не анализирует семантику названий позиций. Таким образом, фразы «игрушки для детей» и «детские игрушки» не будут объединены в один объект, а будут рассматриваться как разные. Появляется потребность в препроцессинге данных.

- Нечестные оценки

Под действием различных факторов пользователь может выставлять оценку продукту, которая может не совпадать с его объективной оценкой. Такие действия вносят определенный шум в данные и может неблагоприятно сказаться на итоговых результатах. Таким образом, результаты могут быть искажены из-за форсированного продвижения продукции определенной фирмой, занимающейся накруткой рейтинга.

- Белые вороны

Всегда существуют пользователи, чье поведение отличается от типовых групп пользователей, их вкусы не совпадают с большинством. Именно, из-за этого рекомендации им могут быть затруднены.

Заключение. Вышеописанный алгоритм был реализован на языке программирования Scala и протестирован на больших объемах данных (в качестве исходных данных использовалась база логов известного веб-сервиса Last.fm). Примеры рекомендаций можно увидеть в презентации.

ЛИТЕРАТУРА

1. Рекомендательные системы [ИСТОЧНИК URL: https://ru.wikipedia.org/wiki/Рекомендательная_система].
2. Коллаборативная фильтрация [ИСТОЧНИК URL: https://ru.wikipedia.org/Коллабортивная _фильтрация].
3. Коллабортивная фильтрация [ИСТОЧНИК URL: https://habrahabr.ru/post/150399/].

ВИЗУАЛИЗАЦИЯ ГРАФОВ С ПОМОЩЬЮ СИЛОВЫХ АЛГОРИТМОВ

М.В. Демешко, А.Ю. Дёмин
(г. Томск, Томский политехнический университет)
e-mail: demeshkomaria@gmail.com

GRAPH VISUALIZATION USING FORCE ALGORITHMS

M.V. Demeshko, A.Y. Demin
(Tomsk, Tomsk Polytechnic University)

This article describes theory and realization of force-based graph visualization. There is an overview of aesthetic criteria of visualization. Article contains examples of visualization using force-based algorithms and describes the need to use different metrics.

Keywords: graph, force-directed, graph-visualization, graph drawing, graph metrics.

Задача визуализации графов встречается в таких областях как картография, анализ сетей (в том числе социальных), анализ программных зависимостей, анализ цитируемости публикаций и многих других. Не смотря на то, что в каждой из этих областей используются выборки данных разного объема, а также преследуются различные конечные цели, существует несколько эстетических критериев, которые позволяют оценить качество визуализации графа для любой задачи.

К этим критериям можно отнести отсутствие наложения вершин. В зависимости от задачи, вершины могут плотно примыкать друг к другу или отстоять на некоторое расстояние, но их наложение в любом случае затрудняет чтение графа. Следующий критерий – это минимизация числа пересечений ребер. Для его соблюдения в некоторых алгоритмах используются дуговые или изломанные ребра.

Для некоторых задач принципиальна укладка графа на минимально возможную площадь, то есть длины ребер не должны быть избыточными, а сам граф не должен выходить за область визуализации. Кроме того, одним из самых важных критериев является отражение топологии графа.

Соблюдения перечисленных критериев можно достичь использованием силовых алгоритмов визуализации. Они относятся к классу алгоритмов, использующих в своей работе физические аналогии.

Силовые алгоритмы описывают функцию, которая определяет идеальные параметры, к которым должен стремиться граф. Соответственно, при каждой итерации система прибли-