Информационные процессы, Том 17, № 2, 2017, стр. 159–163

Исследование эффекта добавления негативного сэмплирования при обучении факторизационных машин в задачах построения рекомендательных систем

Д.Е.Беляков, В.В.Кантор

Московский физико-технический институт (государственный университет), Долгопрудный

Поступила в редколлегию 23.06.2017

Аннотация—Важным вопросом при построении рекомендательных систем является выбор позитивных и негативных примеров для обучения. В частности, часто мы знаем о том, что объект понравился пользователю (покупка товара, просмотр и высокая оценка фильма и т.д.), но не имеем сведений о том, что пользователю точно не нравится. В данной работе показано, что добавление искусственных негативных примеров положительно сказывается на качестве рекомендаций не только в том случае, когда негативных примеров нет, но и тогда, когда они есть. Рассмотрено несколько стратегий добавления негативных примеров, в частности, основанные на популярном методе обучения представлений слов word2vec, показавшие заметное повышение качества рекомендаций на исторических данных.

КЛЮЧЕВЫЕ СЛОВА: рекомендательные системы, факторизационные машины, негативное семплирование, word2vec.

1. ВВЕДЕНИЕ

Рекомендательные системы широко используются в различных интернет-сервисах. Коллаборативная фильтрация является одним из наиболее широко используемых методов для построения таких систем. Она использует шаблоны предпочтений пользователя, получаемые из большого количества исторических данных, для построения рекомендаций. Одним из наиболее распространенных сценариев построения рекомендаций является составление на каждой итерации работы системы списка фиксированного размера N, в котором представлены наиболее релевантные пользователю объекты. Для оценивания качества таких рекомендаций используются метрики ранжирования, такие как $Precision@k$ и $Recall@k$.

Одним из важных вопросов, возникающих при построении рекомендательных систем, является неполнота данных. Зачастую известна лишь информация о том, что объект понравился пользователю, и ничего не известно об объектах, которые ему не понравились. В этом случае можно лишь делать предположения о таких негативных примерах, основанные на доступной в исторических данных информации.

В данной работе показано, что стратегия выбора негативных примеров сильно сказывается на качестве получаемых рекомендаций. Рассмотрены несколько стратегий к добавлению негативных примеров в обучающую выборку для построения рекомендательной системы с помощью *факторизационных машин* [1]. Кроме этого, показано, что добавление искусственных негативных примеров положительно сказывается на качестве рекомендаций и в случае, когда в данных присутствует информация о реальных негативных примерах.

2. ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ

2.1. Слабые негативные примеры

Одним из способов построения рекомендаций при отсутствии негативных примеров является одноклассовая коллаборативная фильтрация. В данном подходе в качестве слабых негативных примеров для пользователя считаются все объекты, с которыми не взаимодействовал пользователь. Слабыми они считаются потому, что при таком подходе: 1) среди этих объектов находятся и те, которые потенциально могут понравиться пользователю; 2) количество таких объектов для каждого пользователя, как правило, намного больше, чем известных позитивных примеров. Поэтому при использовании одноклассовой коллаборативной фильтрации все объекты, с которыми не провзаимодействовал пользователь, берутся с небольшим весом. Примером такого подхода могут служить рекомендательные системы, построенные с помощью wALS [2].

2.2. Сильные негативные примеры

При использовании данного подхода негативными примерами считаются не все объекты, с которыми не взаимодействовал пользователь, а только лишь некоторое подмножество таких объектов. При этом, в отличии от предыдущего подхода, негативные примеры берутся с весами, равными позитивным. Негативные примеры в данном случае выбираются случайно из тех объектов, с которыми не взаимодействовал пользователь. Подробнее об этом подходе можно прочитать в [3].

2.3. Word2Vec для рекомендательных систем

Word2Vec – это группа методов, позволяющих отобразить слова из некоторого множества текстов в непрерывное векторное пространство [4]. Отображение основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), в векторном представлении будут иметь близкие координаты векторов-слов. В word2vec применяются два основных алгоритма обучения: CBOW и Skip-gram. В первом случае предсказывается слово основываясь на окружающем его контексте. Во втором же случае для слова предсказывается его контекст.

Применительно к рекомендательным системам в качестве слов используются идентификаторы объектов, а в качестве текстов - наборы позитивных примеров для каждого пользователя. Таким образом, в полученном представлении объекты, часто вместе встречающиеся у различных пользователей, будут иметь близкие координаты векторов. Основываясь на близости векторов для объектов, можно строить рекомендации для пользователей [5].

3. ГЕНЕРАЦИЯ НЕГАТИВНЫХ ПРИМЕРОВ С ПОМОЩЬЮ WORD2VEC

Для генерации негативных примеров прежде всего для каждого пользователя определим его *профиль* как последовательность идентификаторов понравившихся ему объектов. Далее обучим модель word2vec [1] на множестве профилей пользователей. Далее, сопоставим каждому профилю вектор как средний для векторов идентификаторов входящих в профиль объектов.

По обученной модели для каждого пользователя можно получить два набора объектов: наиболее похожих на профиль и наиболее непохожих на него. Это будут, соответственно, объекты, вектора которых наиболее близки к вектору профиля или наиболее удалены от него.

Негативные примеры для каждого пользователя генерируются одним из трех следующих способов:

[1] Для обучения модели использовалась библиотека gensim для python

1. берутся наиболее похожие на профиль объекты, но не входящие в него (**Head**),
2. берутся наиболее непохожие на профиль объекты (**Tail**),
3. берутся в разных долях объекты из обоих наборов (**Combined**).

Мотивацией использования первого способа является то, что близкие к пользователю, но проигнорированные им объекты, скорее всего ему не нравятся. В противном случае он бы обратил на них внимание. Мотивация второго способа - пользователю неинтересны наиболее удаленные от его профиля объекты, т.е. это и есть негативные примеры.

4. ПОСТАНОВКА ЭКСПЕРИМЕНТОВ

Для оценки качества рекомендаций после добавления негативных примеров различными способами обучалась модель на 90% исторических данных с помощью факторизационных машин [2]. Полученные с помощью модели рекомендации оценивались метриками ранжирования *Precision@k* и *Recall@k*, сравниваясь с оставшимися 10%.

Для сравнения качества использовались следующие исторические данные:

– **MovieLens Implicit (MLI)**. Данные об оценках фильмов, которые выставили пользователи. В качестве положительных примеров для пользователей использовались фильмы с оценкой ≥ 3.
– **Foursquare (FS)**. Данные об отметках о посещении пользователями мест в Москве. В качестве положительных примеров для пользователей использовались посещенные ими места.
– **MovieLens (ML)**. Данные об оценках фильмов, которые выставили пользователи. В качестве положительных примеров для пользователей использовались фильмы с оценкой ≥ 3, в качестве негативных - фильмы с оценкой < 3.

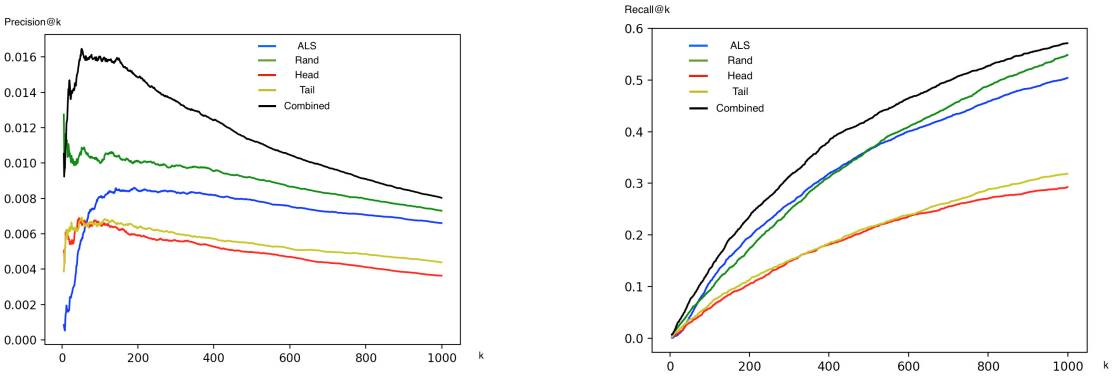

Рис. 1. Сравнение методов на данных MLI по Precision@k и Recall@k

Как можно видеть на Рис. 1, на данных MLI лучший результат показывает комбинированные подход к добавлению негативных примеров. Кроме этого, худшее качество показали методы Head и Tail. Объясняется это тем, что в первом случае мы добавляем много таких объектов, которые могли бы понравиться пользователю, как негативные примеры. Объекты, добавляемые же во втором случае, достаточно легко определяются как неинтересные пользователю, и не помогают построить более точные рекомендации. Похожую картину можно наблюдать и в случае данных FS на Рис. 2 и ML на Рис. 3.

Как видно из Рис. 3, методы негативного семплирования Rand, Tail и Combined показали качество выше, чем без использования негативного семплирования (в этом случае берутся

[2] Для обучения модели использовалась библиотека pylibfm для python

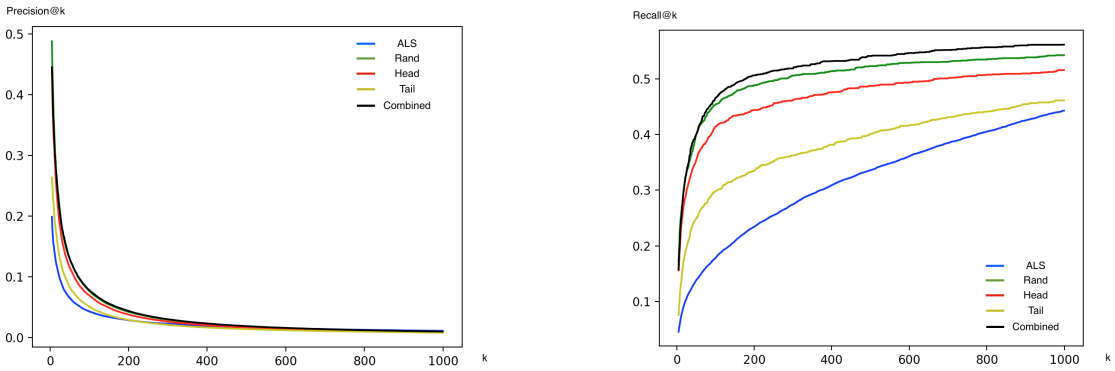

Рис. 2. Сравнение методов на данных FS по Precision@k и Recall@k

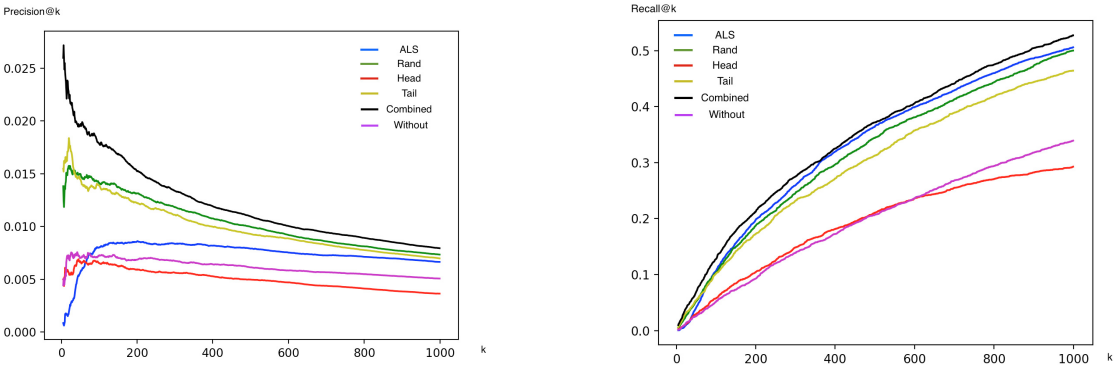

Рис. 3. Сравнение методов на данных ML по Precision@k и Recall@k

негативные примеры только те, что доступны в данных). Следовательно, негативное семплирование способно улучшить качество итоговых рекомендаций даже при наличии негативных примеров в исходных данных.

5. ЗАКЛЮЧЕНИЕ

По результатам работы были получены следующие результаты:

– Негативное семплирование сильно влияет на качество итоговых рекомендаций - как в лучшую, так и в худшую стороны.
– Для улучшения качества рекомендаций необходимы как очевидные негативные примеры, так и те, что похожи на понравившиеся ему объекты.
– Даже при наличии в данных негативных примеров можно добиться улучшения качества путем негативного семплирования.

СПИСОК ЛИТЕРАТУРЫ

1. Rendle S. *Factorization machines with libfm.* ACM Transactions on Intelligent Systems and Technology (TIST), 2012, vol. 3, no. 3, p. 57

2. Pan R., Zhou Y., Cao B., Liu N.N., Lukose R., Scholz M., Yang Q. One-class collaborative filtering. In: *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, 2008, pp. 502–511

3. Rendle S., Freudenthaler C. Improving pairwise learning for item recommendation from implicit feedback. *Proceedings of the 7th ACM international conference on Web search and data mining*, 2014, pp. 273–282

4. Mikolov T., Sutskever I., Chen K., Corrado G.S. Dean J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013, 3111–3119

5. Ozsoy, M.G. From word embeddings to item recommendation. *arXiv preprint arXiv:1601.01356*, 2016

D.E. Beliakov, V.V. Kantor

Selection of positive and negative samples is an important issue when we build a recommendation system. In particular, we often know that the user liked the item (buying a product, viewing and appreciating a movie, etc.), but we don't have information about what the user doesn't like. The purpose of this paper is to show that the addition of artificial negative samples improves the quality of recommendations even in the case when there are a real negative samples in the dataset.

KEYWORDS: recommendation systems, factorization machines, negative sampling, word2vec.