Attr2vec: A Neural Network Based Item Embedding Method

Peng FU, Jiang-hua LV, Shi-long MA and Bing-jie LI

State Key Laboratory of Software Development Environment, Beihang University, No. 37, Xueyuan Road, Haidian District, 100191 Beijing, China

Keywords: Skip-gram, Word2vec, Neural network, Item embedding.

Abstract. Mathematically modeling discrete objects to represent them in the form of low-dimensional vectors, which enable math calculations between discrete objects, is a key component of many computer systems. In recent years, Word2vec model has been proposed in natural language processing, and the low-dimensional vector embedding of words has achieved excellent results. In this paper, we extend the embedding model of word vector and add the representation method of object attributes so that the model can be applied to general objects. Besides, this paper verifies the validity of this model on a large-scale data set, indicating that it is superior to the traditional method SVD.

Introduction

Low-dimensional vector representations of objects [1, 2, 3] are a key part of many computer systems. Object embedding makes it possible to calculate between discrete objects so that mathematical methods can be used to find and quantify the relationship between different discrete objects, including but not limited to calculating cosine distances, calculating clusters, and so on. This method is widely used in user portrait and recommendation system. In this paper, we discuss a novel method of neural embedding by using the attributes of objects as priori information to get better object vector representations. Logs contain existing priori knowledge which can be used to find the relationship between objects and establish a model.

In general, item embedding uses user logs for low-dimensional embedding. The logs are usually formed as lists of items. For instance, one item in the click log indicates that the item has been clicked and one item in the impression log means the item has been impressed to the user.

The recent advances in neural embedding methods in natural language processing (NLP) tasks have greatly increased the ability of computers to solve problems [4, 5, 6, 7]. These methods map words and phrases to a low-dimensional vector space to capture semantic and syntactic relationships between words. Specifically, the neural network based language model NNLM [5] and embedding method Word2vec [6] sets new records for applications in a variety of natural language processing tasks. The Word2vec model treats a single word as a token and builds a low-dimensional vector representation of each word by co-occurrence of token sequences. This model abstract idea that has nothing to do with natural language processing allows the Word2vec model to actually be extended to more General areas [8, 9]. In this paper, we focus on put attributes into the embedding mode based on Word2vec so that the new model can model general objects with attributes, and therefore get the low-dimensional continuous vector representation.

The second section of this paper outlines the work related to other item embedding methods, the third section describes how to apply the attributes to item embedding, and the fourth section describes experiments and related results.

Related Works

Skip-gram with Negative Sampling (SGNS)

Skip-gram with negative sampling (SGNS) is an algorithm of Word2vec proposed by Mikolov et. al. in [8]. The goal of SGNS is to model words in natural language to obtain word vectors in
low-dimensional vector. The idea is to establish a co-occurrence window, looking for the relationship between the center word of the window and the surrounding words, and then capturing the relationship between all words and words by sliding the window. Given a word sequence \( \{w_i\}_{i=1}^{k} \) from limited vocabulary \( W = \{w_i\}_{i=1}^{W} \), the objective function of Skip-gram is to solve for a maximum likelihood function:

\[
\frac{1}{k} \sum_{i=1}^{k} \sum_{C \in C} \log p(w_{i+c}|w_i)
\]

Where \( C \) is the size of the context window, the context window size may depend on \( w_i \), and \( p(w_{i+c}|w_i) \) is the softmax function:

\[
p(w_i|w_j) = \frac{\exp(u_j^T v_i)}{\sum_{k=1}^{W} \exp(u_k^T v_i)}
\]

Where \( u_i \) and \( v_j \) are latent vectors that correspond to the target and the context, \( I_w = \{1, 2, \ldots, |W|\} \) and \( m \) is a hyper-parameter, usually related to the size of the vocabulary and \( m \) is usually in the size of 0.00001 to 0.000001.

The method has efficiency problems when the vocabulary is large, softmax would traversal for each word list, in terms of time is usually not acceptable. We can use Negative sampling to solve the problem.

Negative sampling is as following:

\[
p(w_i|w_j) = \delta(w_j|w_i) \prod_{k=2}^{N} \sigma(-u_i^T v_k)
\]

And:

\[
\sigma(x) = \frac{1}{1 + \exp(-x)}
\]

Where \( N \) is hyper parameter which indicates that \( n \) negative samples would be used to calculate the word vectors. A negative sample is sampled from the unigram distribution raised to the 3/4 power which is found to outperform the unigram distribution itself [8].

In the NLP tasks, there is a problem that the distribution of the word frequency is not uniform, such as the word frequency of "a" and "the" is very high, and the frequency of some uncommon words is very low, which makes the training uneven and leads to the word vector expressing ability weak, for this problem, Word2vec uses the subsampling method to downsample every word. We discard each word \( w \) with a probability:

\[
p(\text{discard}|w) = 1 - \frac{t}{\sqrt{f(w)}}
\]

Where \( f(w) \) is the frequency of the word \( w \) and \( t \) is a threshold hyper parameter. In this way, the sampling probability of high frequency words is punished, down sampling discard the high frequency words somehow, so it also accelerates learning procedure, and get better word vectors.

The model uses stochastic gradient descent (SGD) to get convergent word vectors.
**Item2vec**

Item2vec [10] is a promotion of Word2vec in more general field, it is usually used in recommending system. In a recommending system, the log is formed as lists of items which can be regarded as item sequences generated by user behavior. It is noteworthy that the data in the real environment often have unusable data, such as incorrect operation caused by incorrect records, incomplete records and so on. By generalizing Word2vec, Item2vec has a good adaptability for real-world scenes.

Considering that the item sequences in recommending system is actually the same as token sequences in NLP tasks, Item2vec simply replaced the word with item in Word2vec, discarding the timing information in the item log sequence and preserves only the spatial information, that is, the co-occurrence context. Item2vec assumes a static context space in which timing information is useless, for example, (item1, item2) and (item2, item1) are indistinguishable in the item2vec model, so the formula for item2vec is:

\[
\frac{1}{k} \sum_{i=1}^{k} \sum_{t=1}^{T} \log p(y_t | p_t)
\]

The other part of Item2vec is the same as Word2vec.

**Attr2vec**

By generalizing word in Word2vec, Item2vec raise a model of calculating general item embedding. In NLP tasks, we only regard the word as a token, but the item in real-world usually contains more priori information. As shown in Fig. 1, from a movie record "Titanic", we can get priori information like: {rating: 7.8/10, genre: drama, director: James Cameron, stars: Leonardo DiCaprio, Kate Winslet, …} without analyzing the user behavior.

![Figure 1. Priori information of the move "Titanic".](image)

These properties play an important role in the qualitative characterization of the item, and because there is no information other than token in NLP, Word2vec does not do extra processing on the attributes so that item2vec does not consider the attributes, which makes the algorithm not fully make use of the priori knowledge, which in some way limit the effect of item embedding.
For this reason, we propose to expand the item embedding utilization information, make full use of the attribute information of the existing item to represent an item, and describe the vectorized representation of an item as:

\[ C_i = \sum_j w_j C_j \]  

(7)

Where \( C_i \) is the final item vector and \( w_j \) is the jth attribute’s weight, \( C_j \) is the vector of jth attribute. The equation means that one item vector is a weighted sum of all its available attributes, the vector of an item is determined by its multiple attributes.

Both Word2vec and Item2vec use SGD as an optimization algorithm. Attr2vec also use SGD. However, when the amount of data is large, if an item has k attributes on average, its computation amount will be k times of word2vec. On the other hand, the property is not a complete item, the same calculation as the item gradient will lead bias to the gradient. In order to eliminate such deviation, \( C_i \) does not participate in the gradient calculation when back propagating, using only item to calculate the gradients, and then the calculated gradients will be returned to the attributes and item at the same time. The final calculation, the item’s vector is the weighted sum of attribute vectors, the network is as Fig. 2 shows.

Figure 2. The Attr2vec network.

For Attr2vec, the algorithm can also be accelerated using the same Huffman coding as Word2vec [6]. The pseudo-code of Attr2vec accelerated with Huffman coding is as Fig. 3 shows:
Experimental Results

This section provides comparative testing and results of the Attr2vec method and the item embedding method of the traditional method singular value decomposition (SVD).

Datasets

We evaluate the methods on two different types of datasets.

The first experiment uses Million Song Dataset [11] as test dataset. This log includes data from 1.01 million users and 580,000 pop songs, including user's play log, correspondence between musicians and music, correspondence between music and genres, music tags and so on. The song categories are Pop / Rock, Electronic, Rap, Jazz, Latin, R & B, International, Country, Reggae, Blues, Vocal, "Folk", "New Age".

The second experiment uses the Dou-List dataset from Douban Movie, which includes scores for 100,000 movies by 60,000 users, 22 million user ratings and 580,000 labels. The dataset includes movie names, movie genres, movie starring, labels and other data.

Experimental Environment

The first experiment applies Attr2vec and SVD to the Million Song Dataset. The second experiment applies the Attr2vec method and SVD to the dataset from Douban Movie to find potentially-favorite movies. The optimization methods all use a method of stochastic gradient descent 20 epoch. For negative sampling, take $N = 15$ and embedding dimension is 100. The Subsampling parameter is set to 0.00001, depending on the size of the dataset.

Algorithm 1 Attr2vec using skip-gram and accelerated by Huffman encoding

1: for each $u$ in $\text{Context}(w)$ do
2: $e = 0$
3: for $j = 2 : |u| do$
4: $q = \text{sigmoid}(v(\omega)^T \theta^u_{j-1})$
5: $g = \eta(1 - d^u_e - q)$
6: $e := e + g \theta^u_{j-1}$
7: $\theta^u_{j-1} := \theta^u_{j-1} + gv(\omega)$
8: end for
9: $v(\omega) = v(\omega) + e$
10: for each attr in $u$ do
11: $v(\text{attr}) = v(\text{attr}) + e$
12: end for
13: end for

Figure 3. Pseudo-code of Attr2vec algorithm.

Where $\eta$ is the learning rate, the greater the learning rate, the greater the penalty for wrong decisions and the greater the span of correction for the intermediate vector.
Experiments

We visualize the data by t-distributed stochastic neighbor embedding (t-SNE) [12], and use the cosine kernel function to reduce the vector to 2-dimensional space so that the model has better visualization. Fig. 4 shows the results of different algorithms. Obviously, Attr2vec has a better effect of clustering than SVD. Some of the genres in SVD are not clear or even confused. On the other hand, Attr2vec separates different music genres more clearly.

![Figure 4. t-SNE comparison for SVD(left) and Attr2vec(right).](image)

To quantify the representation of low-dimensional vectors, we test the consistency of the type between an item and its k nearest neighbors. We examine the consistency of the type between an item and its nearest q items by iterating over the top q top entry. Table 1 shows the result when k is 8. Obviously, Attr2vec better than the SVD method in every item.

| Top(q) popular artists | SVD accuracy | Attr2Vec accuracy |
|------------------------|--------------|-------------------|
| 2.5K                   | 84.8%        | 87%               |
| 5K                     | 83.3%        | 86.1%             |
| 10K                    | 80%          | 82.3%             |
| 15K                    | 77%          | 79.8%             |

For the dataset from Douban Movie, we use cosine similarity as a measure of distance to find the "Potentially Likeable Movies" that are closest to a given movie. As you can see, Attr2vec has better found movies that are close to the genre, similar in content, and likely to be of interest of a given movie.

| Movie name | Recommended results 1 | Recommended results 2 | Recommended results 3 | Recommended results 4 |
|------------|-----------------------|-----------------------|-----------------------|-----------------------|
| The Shawshank Redemption | The Pursuit of Happyness | One Flew Over the Cuckoo's Nest | The King's Speech | The Truman Show |
| Spirited Away | Castle in the Sky | Howl's Moving Castle | The Lion King | My Neighbor Totoro |
| Infernal Affairs | Infernal Affairs 2 | Police Story | A Better Tomorrow | The Godfather |
| The Lord of the Rings: The Fellowship of the Ring | The Lord of the Rings: The Two Towers | Hobbits | Pirates of the Caribbean | The Hobbit: The Desolation of Smaug |
| Movie name | Recommended results 1 | Recommended results 2 | Recommended results 3 | Recommended results 4 |
|------------|-----------------------|-----------------------|-----------------------|-----------------------|
| The Shawshank Redemption | The Dark Knight | 12 Angry Men | The King's Speech | Pulp Fiction |
| Spirited Away | The Lion King | Howl's Moving Castle | Mononoke-hime | Forrest Gump |
| Infernal Affairs | Salinui chueok | Gwoemul | Police Story | The Godfather |
| The Lord of the Rings: The Fellowship of the Ring | The Lord of the Rings: The Fellowship of the Ring | Star Wars: Episode V - The Empire Strikes Back | The Godfather | The Godfather: Part II |

**Summary**

In this paper, Attr2vec model is proposed, which uses neural network to embed objects to low dimension vectors.

1. The model incorporates meta data as priori knowledge into embedding method, which makes the model outperforms SVD and matrix factorization in item clustering and classification.
2. Because of the use of negative sampling and skip-gram and more meta-information is added to the model, Attr2vec is more capable of presentation, therefore the model could recommend good items for user.

**Reference**

[1] Paquet, U., Koenigstein, N. (2013, May). One-class collaborative filtering with random graphs. In Proceedings of the 22nd international conference on World Wide Web (pp. 999-1008).

[2] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009 Aug 1(8):30-7.

[3] Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In Proceedings of the 25th international conference on Machine learning 2008 Jul 5 (pp. 880-887). ACM.

[4] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning 2008 Jul 5 (pp. 160-167). ACM.

[5] Mnih A, Hinton GE. A scalable hierarchical distributed language model. In Advances in neural information processing systems 2009 (pp. 1081-1088).

[6] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems 2013 (pp. 3111-3119).

[7] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013 Jan 16.

[8] Frome A, Corrado GS, Shlene J, Bengio S, Dean J, Mikolov T. Devise: A deep visual-semantic embedding model. In Advances in Neural Information Processing Systems 2013 (pp. 2121-2129).

[9] Lazaridou A, Pham NT, Baroni M. Combining language and vision with a multimodal skip-gram model. arXiv preprint arXiv:1501.02598. 2015 Jan 12.
[10] Barkan O, Koenigstein N. Item2vec: neural item embedding for collaborative filtering[C]//Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on. IEEE, 2016: 1-6.

[11] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

[12] Van der Maaten, L., & Hinton, G. Visualizing data using t-SNE. Journal of Machine Learning Research, (2008) 9(2579-2605), 85.