Item2Vec: Neural Item Embedding for Collaborative Filtering

Oren Barkan[*^] and Noam Koenigstein[*]

[*]Microsoft
[^]Tel Aviv University

Abstract

Many Collaborative Filtering (CF) algorithms are item-based in the sense that they analyze item-item relations in order to produce item similarities. Recently, several works in the field of Natural Language Processing suggested to learn a latent representation of words using neural embedding algorithms. Among them, the Skip-gram with Negative Sampling (SGNS), also known as Word2Vec, was shown to provide state-of-the-art results on various linguistics tasks. In this paper, we show that item-based CF can be cast in the same framework of neural word embedding. Inspired by SGNS, we describe a method we name Item2Vec for item-based CF that produces embedding for items in a latent space. The method is capable of inferring item-to-item relations even when user information is not available. We present experimental results on large scale datasets that demonstrate the effectiveness of the Item2Vec method and show it is competitive with SVD.

Index terms – skip-gram, word2vec, neural word embedding, collaborative filtering, item similarity, recommender systems, market basket analysis, item-item collaborative filtering.

1. Introduction

Computing item similarities is a key building block in modern recommender systems. While many recommendation algorithms are focused on learning a low dimensional embedding of users and items simultaneously [1, 2, 3], computing item similarities is an end in itself. Items similarities are extensively used by online retailers for many different recommendation tasks. This papers deals with the overlooked task of learning item similarities by embedding items in a low dimensional space regardless of the users.

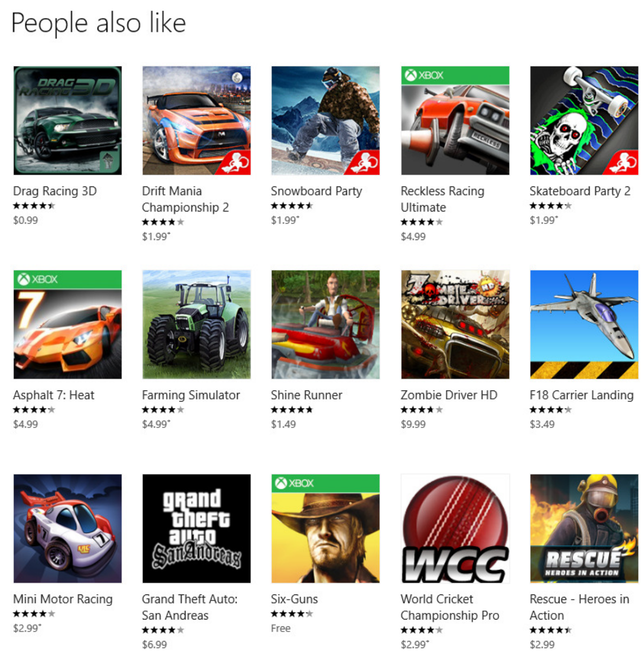

Figure 1: Recommendations in Windows 10 Store based on similar items to Need For Speed.

Item-based similarities are used by online retailers for recommendations based on a single item. For example, in the Windows 10 App Store, the details page of each app or game includes a list of other similar apps titled "People also like". This list can be extended to a full page recommendation list of items similar to the original app as shown in Fig.1. Similar recommendation lists which are based merely on similarities to a single item exist in most online stores e.g., Amazon, Netflix, Google Play, iTunes store and many others.

The single item recommendations are different than the more "traditional" user-to-item recommendations because they are usually shown in the context of an explicit user interest in a specific item and in the context of an explicit user intent to purchase. Therefore, single item recommendations based on item similarities often have higher Click-Through Rates (CTR) than user-to-item recommendations and consequently responsible for a larger share of sales or revenue.

Single item recommendations based on item similarities are used also for a variety of other recommendation tasks: In "candy rank" recommendations for similar items (usually of lower price) are suggested at the check-out page right before the payment. In "bundle" recommendations a set of several items are grouped and recommended together. Finally, item similarities are used in online stores for better exploration and discovery and improve the overall user experience. It is unlikely that a user-item CF method, that learns the connections between items implicitly by defining slack variables for users, would produce better item representations than a method that is optimized to learn the item relations directly.

Item similarities are also at the heart of item-based CF algorithms that aim at learning the representation directly from the item-item relations [4, 5]. There are several scenarios where item-based CF methods are desired: in a large scale dataset, when the number of users is significantly larger than the number of items, the computational complexity of methods that model items solely is significantly lower than methods that model both users and items simultaneously. For example, online music services may have hundreds of millions of enrolled users with just tens of thousands of artists (items).

In certain scenarios, the user-item relations are not available. For instance, a significant portion of today's online shopping is done without an explicit user identification process. Instead, the available information is per session. Treating these sessions as "users" would be prohibitively expensive as well as less informative.

Recent progress in neural embedding methods for linguistic tasks have dramatically advanced state-of-the-art natural language processing (NLP) capabilities [6, 7, 8, 9]. These methods attempt to map words and phrases to a low dimensional vector space that captures semantic and syntactic relations between words. Specifically, Skip-gram with Negative Sampling (SGNS), known also as Word2Vec [8], set new records in various NLP tasks [7, 8] and its applications have been extended to other domains beyond NLP [10, 11].

In this paper, we propose to apply SGNS to item-based CF. Motivated by its great success in other domains, we suggest that SGNS with minor modifications may capture the relations between different items in collaborative filtering datasets. To this end, we propose a modified version of SGNS named Item2Vec. We show that Item2Vec can induce a similarity measure that is competitive with an item-based CF using SVD. It is important to clarify that we do not claim to achieve the state-of-the-art, but merely to show another successful application of SGNS for item-based CF. Therefore, we choose to compare our method to a SVD-based model and leave the comparison to other more complex method to a future research.

The rest of the paper is organized as follows: Section 2 overviews related work in the context of SGNS, in Section 3 we describe how to apply SGNS for item-based CF, in Section 4 we describe the experimental setup and present empirical qualitative and quantitative results.

2. Skip-gram with negative sampling (SGNS) – Related work

SGNS is a neural word embedding method that was introduced by Mikolov et. al in [8]. The method aims at finding words representation that captures the relation between a word to its surrounding words in a sentence. In the rest of this section, we provide a brief overview of the SGNS method.

Given a sequence of words $(w_i)_{i=1}^{K}$ from a finite vocabulary $W = \{w_i\}_{i=1}^{W}$, the Skip-gram objective aims at maximizing the following term:

$$\frac{1}{K}\sum_{i=1}^{K}\sum_{-c\le j\le c, j\ne 0}\log p(w_{i+j} \mid w_i) \tag{1}$$

where c is the context window size (that may depend on w_i) and $p(w_j \mid w_i)$ is the softmax function:

$$p(w_j \mid w_i) = \frac{\exp(u_i^T v_j)}{\sum_{k\in I_W}\exp(u_i^T v_k^T)} \tag{2}$$

where $u_i \in U(\subset \mathbb{R}^m)$ and $v_i \in V(\subset \mathbb{R}^m)$ are latent vectors that correspond to the target and context representations for the word $w_i \in W$, respectively, $I_W \triangleq \{1,...,|W|\}$ and the parameter m is chosen empirically and according to the size of the dataset.

Using Eq. (2) is impractical due to the computational complexity of $\nabla p(w_j \mid w_i)$, which is a linear function of the vocabulary size $|W|$ that is usually in size of $10^5 - 10^6$.

Negative sampling comes to alleviate the above computational problem by the replacement of the softmax function from Eq. (2) with

$$p(w_j \mid w_i) = \sigma(u_i^T v_j) \prod_{k=1}^{N} \sigma(-u_i^T v_k)$$

where $\sigma(x) = 1/1 + \exp(-x)$, N is a parameter that determines the number of negative examples to be drawn per a positive example. A negative word w_i is sampled from the unigram distribution raised to the 3/4rd power. This distribution was found to significantly outperform the unigram distribution, empirically [8].

In order to overcome the imbalance between rare and frequent words the following subsampling procedure is proposed [8]: Given the input word sequence, we discard each word w with a probability $p(discard \mid w) = 1 - \sqrt{\frac{\rho}{f(w)}}$ where $f(w)$ is the frequency of the word w and ρ is a prescribed threshold. This procedure was reported to accelerate the learning process and to improve the representation of rare words significantly [8]. In our experiments, we observed the same improvements when applying subsampling.

Finally, U and V are estimated by applying a stochastic optimization with respect to the objective in Eq. (1).

3. Item2Vec – SGNS for item-based CF

In the context of CF data, the items are given as user generated sets. Note that the information about the relation between a user and a set of items is not always available. For example, we might be given with a dataset that is generated from orders that a store received, without any information about the identity that sent the order. In other words, there are scenarios where multiple sets of items might belong to the same user, but this information is not provided. In Section 4, we present experimental results that show that our method handles these scenarios as well.

We propose to apply SGNS to item-based CF. The application of SGNS to CF data is straightforward once we realize that a sequence of words is equivalent to a set or basket of items. Therefore, from now on, we will use the terms "word" and "item" interchangeably.

By moving from sequences to sets, the spatial / time information is lost. We choose to discard this information, since in this paper, we assume a static environment where items that share the same set are considered similar, no matter in what order / time they were generated by the user. This assumption may not hold in other scenarios, but we keep the treatment of these scenarios out of scope of this paper.

Since we ignore the spatial information, we treat each pair of items that share the same set as a positive example. This implies a window size that is determined from the set size. Specifically, for a given set of items, the objective from Eq. (1) is modified as follows:

$$\frac{1}{K} \sum_{i=1}^{K} \sum_{j \neq i}^{K} \log p(w_j \mid w_i)$$

Another option is to keep the objective in Eq. (1) as is, and shuffle each set of items during runtime. In our experiments we observed that both options perform the same.

The rest of the process remains identical to the algorithm described in Section 2. We name the described method "Item2Vec".

In this work, we used u_i as the final representation for the i-th item and the affinity between a pair of items is computed by the cosine similarity.

Other options are to use v_i, the additive composition, $u_i + v_i$ or the concatenation $\left[u_i^T \ v_i^T\right]^T$. Note that the last two options sometimes produce superior representation.

4. Experimental Results

In this section, we provide an empirical evaluation of the Item2Vec method. We provide both qualitative and quantitative results depending whether a metadata about the items exists. As a baseline item-based CF algorithm we used item-item SVD.

4.1 Datasets

We evaluate the methods on two different types of datasets, both private.

The first dataset is user-artist data that is retrieved from the Microsoft XBOX Music service. This dataset consist of 9M events. Each event consists of a user-artist relation, which means the user played a song by the specific artist. The dataset contains 732K users and 49K distinct artists.

The second dataset contains physical goods orders from Microsoft Store. An order is given by a basket of items without any information about the user that made it. Therefore, the information in this dataset is weaker in the sense that we cannot bind between users and items. The dataset consist of 379K orders (that contains more than a single item) and 1706 distinct items.

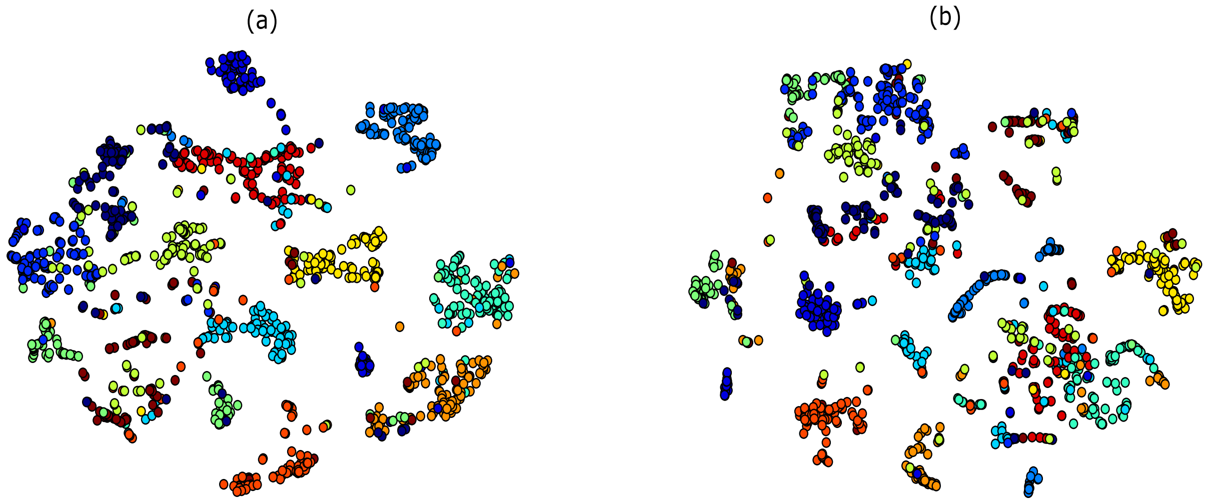

Figure 2: t-SNE embedding for the item vectors produced by Item2Vec (a) and SVD (b). The items are colored according to a web retrieved genre metadata.

4.2 Systems and parameters

We applied Item2Vec to both datasets. The optimization is done by stochastic gradient decent. We ran the algorithm for 20 epochs. We set the negative sampling value to $N = 15$ for both datasets. The dimension parameter m was set to 100 and 40 for the Music and Store datasets, respectively. We further applied subsampling with ρ values of 10^{-5} and 10^{-3} to the Music and Store datasets, respectively. The reason we set different parameter values is due to different sizes of the datasets.

We compare our method to a SVD based item-item similarity system. To this end, we apply SVD to a square matrix in size of number of items, where the (i, j) entry contains the number of times (w_i, w_j) appears as a positive pair in the dataset. Then, we normalized each entry according to the square root of the product of its row and column sums. Finally, the latent representation is given by the rows of $US^{1/2}$, where S is a diagonal matrix that its diagonal contains the top m singular values and U is a matrix that contains the corresponding left singular vectors as columns. The affinity between items is computed by cosine similarity of their representations. Throughout this section we name this method "SVD".

4.3 Experiments and results

The music dataset does not provide genre metadata. Therefore, for each artist we retrieved the genre metadata from the web to form a genre-artist catalog. Then we used this catalog in order to visualize the relation between the learnt representation and the genres. This is motivated by the assumption that a useful representation would cluster artists according to their genre. To this end, we generated a subset that contains the top 100 popular artists per genre for the following distinct genres: 'R&B / Soul', 'Kids', 'Classical', 'Country', 'Electronic / Dance', 'Jazz', 'Latin', 'Hip Hop', 'Reggae / Dancehall', 'Rock', 'World', 'Christian / Gospel' and 'Blues / Folk'. We applied t-SNE [12] with a cosine kernel to reduce the dimensionality of the item vectors to 2. Then, we colored each artist point according to its genre.

Figures 2(a) and 2(b) present the 2D embedding that was produced by t-SNE, for Item2Vec and SVD, respectively. As we can see, Item2Vec provides a better clustering. We further observe that some of the relatively homogenous areas in Fig. 2(a) are contaminated with items that are colored differently. We found out that many of these cases originate in artists that were mislabeled in the web or have a mixed genre.

TABLE 1
INCONSISTENCIES BETWEEN GENRES FROM THE CATALOG TO THE ITEM2VEC BASED KNN PREDICTIONS (K = 8)

Artist name	Genre in catalog (retrieved from the web)	Genre predicted by Item2Vec based Knn (consistent with Wikipedia)
DMX	R&B / Soul	Hip Hop
LLJ	Rock /Metal	Hip Hop
Walter_Beasley	Blues / Folk	Jazz
Sevendust	Hip Hop	Rock / Metal
Big Bill roonzy	Reggae / Dancehall	Blues / Folk
Anita Baker	Rock	R&B / Soul
Cassandra Wilson	R&B / Soul	Jazz
Notixx	Reggae / Dancehall	Electronic

TABLE 2
A COMPARISON BETWEEN SVD AND ITEM2VEC ON GENRE CLASSIFICATION TASK FOR VARIOUS SIZES OF TOP POPULAR ARTIST SETS

Top (q) popular artists	SVD accuracy	Item2Vec accuracy
2.5K	85%	86.4%
5K	83.4%	84.2%
10K	80.2%	82%
15K	76.8%	79.5%
20K	73.8%	77.9%
10K unpopular (see text)	58.4%	68%

Table 1 presents several examples, where the genre associated with a given artist (according to metadata that we retrieved from the web) is inaccurate or at least inconsistent with Wikipedia. Therefore, we conclude that usage based models such as Item2Vec may be useful for the detection of mislabeled data and even provide a suggestion for the correct label using a simple k nearest neighbor (KNN) classifier.

In order to quantify the similarity quality, we tested the genre consistency between an item and its nearest neighbors. We do that by iterating over the top q popular items (for various values of q) and check whether their genre is consistent with the genres of the k nearest items that surround them. This is done by a simple majority voting. We ran the same experiment for different neighborhood sizes (k = 6, 8, 10, 12 and 16) and no significant change in the results was observed.

Table 2 presents the results obtained for $k = 8$. We observe that Item2Vec is consistently better than the SVD model, where the gap between the two keeps growing as q increases. This might imply that Item2Vec produces a better representation for less popular items than the one produced by SVD, which is unsurprising since Item2Vec subsamples popular items and samples the negative examples according to their popularity.

We further validate this hypothesis by applying the same 'genre consistency' test to a subset of 10K unpopular items (the last row in Table 2). We define an unpopular item in case it has less than 15 users that played its corresponding artist. The accuracy obtained by Item2Vec was 68%, compared to 58.4% by SVD.

Qualitative comparisons between Item2Vec and SVD are presented in Tables 3-4 for Music and Store datasets, respectively. The tables present seed items and their 5 nearest neighbors. The main advantage of this comparison is that it enables the inspection of item similarities in higher resolutions than genres. Moreover, since the Store dataset lacks any informative tags / labels, a qualitative evaluation is inevitable. We observe that for both datasets, Item2Vec provides lists that are better related to the seed item than the ones provided by SVD. Furthermore, we see that even though the Store dataset contains weaker information, Item2Vec manages to infer item relations quite well.

5. Conclusion

In this paper, we proposed Item2Vec - a neural embedding algorithm for item-based collaborative filtering. Item2Vec is based on SGNS with minor modifications.

We present both quantitative and qualitative evaluations that demonstrate the effectiveness of Item2Vec when compared to a SVD-based item similarity model. We observed that Item2Vec produces a better representation for items than the one obtained by the baseline SVD model, where the gap between the two becomes more significant for unpopular items. We explain this by the fact that Item2Vec employs negative sampling together with subsampling of popular items.

In future we plan to investigate more complex CF models such as [1, 2, 3] and compare between them and Item2Vec.

TABLE 3

A QUALITATIVE COMPARISON BETWEEN ITEM2VEC AND SVD FOR SELECTED ITEMS FROM THE MUSIC DATASET

Seed item	Item2Vec – Top 5	SVD – Top 5
Justin Timberlake - Pop	• Rihanna - Pop • Beyonce - Pop • The Black Eyed Peas - Pop • Bruno Mars - Pop • Usher - Pop	• JC Chasez - Pop • Jordan Knight - Pop • Shontelle - Pop • *Nsync - Pop • The Black Eyed Peas - Pop
David Guetta-Electronic	• Avicii - Electronic • Calvin Harris - Electronic • Martin Solveig - Electronic • Major Lazer - Electronic • Deorro - Electronic	• Brothers - Electronic / Dance_Dance • The Blue Rose - Electronic / Downtempo • JWJ - Electronic / Dance_Progressive • Akcent - World_World Pop • JWC - Electronic / Dance_House
Britney Spears - Pop	• Miley Cyrus - Pop • Lady Gaga - Pop_Contemporary Pop • Christina Aguilera - Latin_Latin Pop • P!nk - Pop_Contemporary Pop • Ke$ha - Pop_Contemporary Pop	• Christina Aguilera - Latin_Latin Pop • Gwen Stefani - Rock_Indie / Alternative • Ke$ha - Pop_Contemporary Pop • Jessica Simpson - Pop_Contemporary Pop • Brooke Hogan - Pop_Contemporary Pop
Katy Perry -Pop	• Miley Cyrus - Soundtracks_Film • Kelly Clarkson - Pop_Contemporary Pop • P!nk - Pop_Contemporary Pop • Meghan Trainor - Pop_Contemporary Pop • Taylor Swift - Pop_Contemporary Pop	• Last Friday Night - Electronic / Dance_Dance • Winx Club - Kids_Kids • Boots On Cats - Rock_Indie / Alternative • Thaman S. - World_Indonesia • GMPresents - Pop_Singer-Songwriter
Dr. Dre – Hip Hop	• Game - Pop_Contemporary Pop • Snoop Dogg - Hip Hop • N.W.A - Hip Hop_Contemporary Hip Hop • DMX - R&B / Soul_R&B • Kendrick Lamar - Hip Hop	• Jack The Smoker - Hip Hop • Royal Goon - Hip Hop • Hoova Slim - Hip Hop • Man Power - Electronic / Dance_Dance • Ol'Kainry - World_Middle East
Johnny Cash -Country	• Willie Nelson - Country_Traditional • Jerry Reed - Country_Traditional Country • Dolly Parton - Country_Traditional • Merle Haggard - Country_Traditional • HANK WILLIAMS - Pop	• HANK WILLIAMS - Pop_Contemporary Pop • The Highwaymen - Blues / Folk • Johnny Horton - Pop_Contemporary Pop • Hoyt Axton - Pop_Contemporary Pop • Willie Nelson - Country_Traditional Country
Daft Punk -Electronic	• Kavinsky - Electronic / Breakbeat / Electro • Fatboy Slim - Electronic / Dance_House • Calvin Harris - Electronic / Dance_Dance • Moby - Electronic / Dance_Dance • Chromeo - Electronic / Dance_Dance	• Magic Hats - Pop_Contemporary Pop • Modjo - Electronic / Dance_Dance • Rinrse - Electronic / Dance_Dance • Mirwas - Electronic / Dance_Dance • Playboy - Pop_Contemporary Pop
Guns N' Roses - Rock	• Aerosmith - Rock_Metal & Hard Rock • Ozzy Osbourne - Rock_Metal • Bon Jovi - Rock_Mainstream Rock • Mtley Cre - Rock_Indie / Alternative • AC/DC - Rock_Metal & Hard Rock	• Bon Jovi - Rock_Mainstream Rock • Gilby Clarke - Rock_Indie / Alternative • Def Leppard - Rock_Indie / Alternative • Mtley Cre - Rock_Indie / Alternative • Skid Row - Rock_Indie / Alternative
Linkin Park - Rock	• Thirty Seconds To Mars - Rock_Indie • Evanescence - Rock_Indie / Alternative • System Of A Down - Rock_Metal • Nickelback - Rock_Indie / Alternative • Limp Bizkit - Rock_Metal & Hard Rock	• DJ David - Electronic / Dance_Ambient • Little Nicky Soundtrack - Rock • Fort Minor - Hip Hop • Entspannungsmusik Klavier Akademie - World • Evanescence - Rock_Indie / Alternative

TABLE 4

A QUALITATIVE COMPARISON BETWEEN ITEM2VEC AND SVD FOR SELECTED ITEMS FROM THE STORE DATASET

Seed item	Item2Vec – Top 5	SVD – Top 5
LEGO Dimensions Emmet Fun Pack	• LEGO Dimensions Bad Cop Fun Pack • LEGO Dimensions NinjagoTeam Pack • LEGO Dimensions The Simpsons: Bart Fun Pack • LEGO Dimensions Gimli Fun Pack • LEGO Dimensions Scooby-Doo! Team Pack	• Minecraft Foam Pickaxe • Disney INFINITY Toy Box Starter Pack for Xbox One • Minecraft for Xbox One • Terraria for Xbox One • Dragon Ball Xenoverse for Xbox One Full Game Download Code
Minecraft Lanyard	• Minecraft Diamond Earrings • Minecraft Periodic Table • Minecraft 17-Inch Enderman Plush • Minecraft Crafting Table • Minecraft 12-Inch Skeleton Plush	• Rabbids Invasion for Xbox One: The Interactive TV Show Season Pass Download Code • Mortal Kombat X Premium Edition for Xbox One Full Game Download Code • Minecraft Periodic Table • Middle-earth: Shadow of Mordor PC Game • Kinect Sensor for Xbox One
Skylanders SWAP Force Character - Star Strike	• Skylanders SWAP Force Character - Scorp • Skylanders SWAP Force Series 3 Character - Heavy Duty Sprocket • Skylanders SWAP Force Character - Lava Barf Eruptor • Skylanders SWAP Force Series 3 Character - Slobber Tooth • Skylanders SWAP Force Character - Boom Jet	• NBA Live 14 for Xbox One • Watch Dogs Limited Edition PC Game • Trials Fusion Season Pass Download Code for Xbox 360 • Minecraft Creeper Face Sticker • Disney INFINITY Figure: Woody
Disney INFINITY 2.0 Figure: Disney Originals Baymax	• Disney INFINITY 2.0 Figure: Disney Originals Maleficent • Disney INFINITY 2.0 Figure: Disney Originals Hiro • Disney INFINITY 2.0 Figure: Disney Originals Stitch • Disney INFINITY 2.0 Figure: Marvel Super Heroes Nick Fury • Disney INFINITY 2.0 Marvel Super Heroes Toy Box Game Discs	• Disney INFINITY 2.0 Figure: Disney Originals Stitch • Mega Bloks Halo UNSC Firebase • LEGO Dimensions The Simpsons: Bart Fun Pack • Mega Bloks Halo UNSC Gungoose • Sid Meier's Civilization: Beyond Earth (PC)
GoPro LCD Touch BacPac	• GoPro Anti-Fog Inserts • GoPro The Frame Mount • GoPro HERO3+ Standard Housing • GoPro Floaty Backdoor • GoPro 3-Way	• Titanfall Collector's Edition for Xbox One • GoPro The Frame Mount • Call of Duty: Advanced Warfare Day Zero Edition PC Game • Evolve PC Game • Total War: Rome 2 PC Game
Thrustmaster T3PA Pedal Set Add-On	• Thrustmaster TH8A Add-On Gearbox Shifter • Thrustmaster T3PA-PRO 3-Pedal Add-On • Thrustmaster TM Leather 28 GT Wheel Add-On • Astro Gaming A40 Headset for Xbox One • Thrustmaster Ferrari F1 Wheel Add-On	• Thrustmaster TX Racing Wheel Ferrari 458 Italia Edition • Tom Clancy's The Division for Xbox One • Xbox One Play and Charge Kit • Xbox One Chat Headset • Disney INFINITY 2.0 Figure: Disney Originals Tinker Bell
Surface Pro 4 Type Cover with Fingerprint ID (Onyx)	• UAG Surface Pro 4 Case (Black) • Surface Pro 4 Type Cover with Fingerprint ID (Onyx) • Jack Spade Zip Sleeve for Surface (Luggage Nylon Navy) • Microsoft Surface 65W Power Supply • PerfectFit Surface Pro 4 Antimicrobial CleanShield Premium Film Screen Protection	• Farming Simulator PC Game • Dell Alienware Echo 17 R3 AW17R3-834SLV Signature Edition Gaming Laptop • Bose SoundLink Around-Ear Wireless Headphones II • UAG Surface Pro 4 Case (Black) • Microsoft Surface Pro 4 - Customize your device
Windows Server 2012 R2	• Windows Server 2012 Remote Desktop Services 1-User CAL • Exchange Server 2010 Standard Edition 64-Bit 5-Client License • Windows Server 2012 5-User Client Access License • Exchange Server 2010 Standard Edition 5-User CAL • Windows Server 2012 Remote Desktop Services 5-Device CAL	• NBA Live for Xbox One – 4 600 NBA Points Download Code • Windows 10 Home • Mega Bloks Halo Covenant Drone Outbreak • Mega Bloks Halo UNSC Vulture Gunship • Windows 10 Pro

References

[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] Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web 2001 Apr 1 (pp. 285-295). ACM.

[5] Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE. 2003 Jan;7(1):76-80.

[6] 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.

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

[8] 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).

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

[10] Frome A, Corrado GS, Shlens 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).

[11] Lazaridou A, Pham NT, Baroni M. Combining language and vision with a multimodal skip-gram model. arXiv preprint arXiv:1501.02598. 2015 Jan 12.

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