On the Consistency of Average Embeddings for Item Recommendation

WALID BENDADA*, Deezer Research & LAMSADE, Université Paris Dauphine, PSL, France
GUILLAUME SALHA-GALVAN, Deezer Research, France
ROMAIN HENNEQUIN, Deezer Research, France
THOMAS BOUABÇA, Deezer Research, France
TRISTAN CAZENAVE, LAMSADE, Université Paris Dauphine, PSL, France

A prevalent practice in recommender systems consists of averaging item embeddings to represent users or higher-level concepts in the same embedding space. This paper investigates the relevance of such a practice. For this purpose, we propose an expected precision score, designed to measure the consistency of an average embedding relative to the items used for its construction. We subsequently analyze the mathematical expression of this score in a theoretical setting with specific assumptions, as well as its empirical behavior on real-world data from music streaming services. Our results emphasize that real-world averages are less consistent for recommendation, which paves the way for future research to better align real-world embeddings with assumptions from our theoretical setting.

CCS Concepts: • **Information systems** → *Recommender systems*; • **Computing methodologies** → *Learning latent representations.*

Additional Key Words and Phrases: Recommender Systems, Representation Learning, Embedding Vectors, Average Embeddings.

ACM Reference Format:
Walid Bendada, Guillaume Salha-Galvan, Romain Hennequin, Thomas Bouabça, and Tristan Cazenave. 2023. On the Consistency of Average Embeddings for Item Recommendation. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3604915.3608837

1 INTRODUCTION

Modern recommender systems often leverage representation learning techniques to summarize similarities between recommendable items [20, 23, 29, 30]. These techniques learn low-dimensional vectorial representations of these items, also known as *embedding vectors* or simply *embeddings*, in a common vector space where item proximity should reflect user preferences (for a *collaborative filtering* system [16]) or resemblance of item characteristics (for a *content-based* system [15]). By computing similarity metrics such as the inner product or Euclidean distance between embeddings, the recommender system can subsequently identify new items similar to the ones each user has interacted with [10, 14].

A prevalent practice associated with the use of such embeddings in industry-oriented research and applications consists of *averaging item embeddings* to obtain embeddings for users or higher-level concepts in the same vector space [4, 8, 9, 20, 21, 25]. As an illustration, Spotify learns embedding representations of listening sessions by averaging pre-computed embeddings of the music tracks listened to during these sessions [9]. This service also computes "long-term" user embeddings, used for recommendation purposes, by averaging the session embeddings associated with each user. Deezer computes embeddings for several types of recommendable music collections, such as playlists and

*Contact author: research@deezer.com

Manuscript submitted to ACM

albums, by averaging embeddings of the music tracks present in these collections [4]. Yahoo averages embeddings of the news articles previously browsed by each user to represent them in the same embedding space as articles and provide personalized news recommendations [20]. Alibaba averages side information embeddings, including category and brand embeddings, to obtain product embeddings for cold start recommendation on the Taobao e-commerce platform [25].

However, despite its prevalence, this averaging practice is often adopted without explicit justification from a theoretical standpoint. As we detail in Section 2, the rationale for averaging item embeddings mainly stands in the simplicity and scalability of this approach [4, 6]. Yet, it is unclear to which extent averaging item embeddings guarantees to provide faithful user or higher-level concept representations for recommendation. For instance, assuming we represent a user by the average embedding of their previously consumed items, to what degree would the neighboring items of this average constitute relevant recommendations for this user? While the impact of averaging and other pooling operations has been studied in other fields, notably natural language processing (NLP) [2, 5], to our knowledge, the consistency of averaging operations remains relatively understudied in the specific context of a recommender system.

In this short paper, we propose to investigate these important considerations, making the following contributions:

- Firstly, we define $\text{Consistency}_k(\mathcal{X})$, a general expected precision score introduced in this study to measure the consistency of an average embedding relative to the items it summarizes, from a recommendation standpoint.
- Secondly, we examine the consistency of averaging operations in a *theoretical* setting with general assumptions on item embeddings, providing an in-depth analysis of the expression of $\text{Consistency}_k(\mathcal{X})$ in this setting.
- Thirdly, we analyze the *empirical* behavior of this score on real-world data. Our experiments consider three variants of large-scale music track embeddings obtained from the music streaming service Deezer [4].
- Lastly, we discuss the discrepancies between our theoretical and empirical results, emphasizing that real-world averages are less consistent for recommendation. This discussion paves the way for future research to better align real-world data with our theoretical assumptions. Overall, we believe this study will be insightful for researchers and practitioners aiming to improve the faithfulness of average embeddings in their recommender systems.

This paper is organized as follows. In Section 2, we introduce the average embedding consistency problem more precisely. We subsequently present our proposed $\text{Consistency}_k(\mathcal{X})$ score. We report our theoretical analysis of this score in Section 3, and discuss our experimental results on real-world data in Section 4. Finally, we conclude in Section 5.

2 PROBLEM FORMULATION AND EVALUATION

2.1 Preliminaries

2.1.1 Mathematical Notation. Throughout this paper, we consider a set $\mathcal{X} = \{X_i\}_{1 \leq i \leq N}$ of $N \in \mathbb{N}^*$ real-valued vectors of dimension $d \in \mathbb{N}^*$, i.e., $\forall i \in \{1, \dots, N\}, X_i = (X_{i,1}, X_{i,2}, \dots, X_{i,d})^\intercal \in \mathbb{R}^d$. We denote by $\mathcal{X}_k$ the set of subsets of $\mathcal{X}$ of cardinality k, for any $k \in \{1, \dots, N\}$. For any vector $z \in \mathbb{R}^d$ and $k \in \{1, \dots, N\}$, we define $\mathcal{X}_k(z) \in \mathcal{X}_k$ as the set of the k *nearest neighbors*[1] of z among the N elements of $\mathcal{X}$, according to some *similarity metric* $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, i.e.,

$$\mathcal{X}_k(z) = \underset{\mathcal{Y} \in \mathcal{X}_k}{\arg\max} \sum_{y \in \mathcal{Y}} s(z, y). \tag{1}$$

Moreover, for any subset $\mathcal{U} = \{u_i\}_{1 \leq i \leq k} \subseteq \mathcal{X}$ of k vectors, we define the *center* or *average* of $\mathcal{U}$ as $\mu_{\mathcal{U}} = \frac{1}{k} \sum_{u_i \in \mathcal{U}} u_i$, and we denote the complementary set of $\mathcal{U}$ within $\mathcal{X}$ as $\overline{\mathcal{U}}$, i.e., $\overline{\mathcal{U}} = \mathcal{X} \setminus \mathcal{U} = \{X \in \mathcal{X}, X \notin \mathcal{U}\}$.

[1] This set might not be unique if z is equidistant to several elements of $\mathcal{X}$, in which case $\mathcal{X}_k(z)$ can be drawn uniformly from all complying sets.

2.1.2 From Mathematics to Recommender Systems. In the context of an embedding-based recommender system [16, 30], $\mathcal{X}$ would correspond to a set of d-dimensional embedding representations associated with all recommendable *items* in a catalog, while $\mathcal{U} \in \mathcal{X}_k$ would correspond to the embeddings of a subset of k items. For instance, $\mathcal{X}$ could represent music tracks on a music streaming service, and $\mathcal{U}$ could represent the tracks present in a playlist or album of length k [4, 9]. Alternatively, $\mathcal{X}$ could represent all products from an e-commerce platform and $\mathcal{U}$ the shopping cart of a user [25, 28]. At this stage, we do not make assumptions regarding the representation learning technique used to learn these embeddings.

2.2 Problem Formulation

2.2.1 Averaging Embeddings. As illustrated in Section 1, recommender systems frequently average item embeddings. Using the above notation, the action of averaging all embeddings of an item collection $\mathcal{U}$ to represent this collection translates to using $\mu_{\mathcal{U}}$ as the representation. As $\mu_{\mathcal{U}} \in \mathbb{R}^d$, one can interpret this vector as a new embedding in the same vector space as items, and, therefore, measure the similarity between $\mu_{\mathcal{U}}$ and other items using the similarity metric s.

2.2.2 How Relevant is this Practice? This paper aims to rigorously investigate the consistency of this averaging practice. For instance, if we represent a user by the average embedding $\mu_{\mathcal{U}}$ of the items $\mathcal{U}$ this user has consumed or liked, do the items similar to $\mu_{\mathcal{U}}$ in the embedding space (according to s) also constitute items that the user would like? In the same way, if $\mu_{\mathcal{U}}$ summarizes an album composed of the tracks in $\mathcal{U}$, are tracks similar to $\mu_{\mathcal{U}}$ also similar to tracks in $\mathcal{U}$?

Intuitively, for $\mu_{\mathcal{U}}$ to be faithful to $\mathcal{U}$, we expect $\mu_{\mathcal{U}}$ to remain similar to the original items from $\mathcal{U}$. For instance, the user embedding should remain similar to the items the user has already liked, and the album embedding to the tracks contained in the album. For this reason, this paper focuses on the following specific research question: *to which extent and under which conditions does $\mu_{\mathcal{U}}$ remain similar to items from $\mathcal{U}$, i.e., to the items used for its construction?*

2.2.3 Related Work. To our knowledge, this theoretical question remains understudied in recommendation. Existing research predominantly relied on averages for practical reasons. Average embeddings are faster and simpler to compute than alternatives such as neural aggregations [11, 12]. They have also been praised for their scalability, as they provide fixed-size representations independently of the number of items in $\mathcal{U}$ [4, 6]. Yet, recent research pointed out some of their limitations, e.g., to represent heterogeneous or contextual preferences [9, 22, 24]. We also acknowledge that the pros and cons of average embeddings have been studied for other applications, such as NLP tasks [2, 5, 18, 27]. While being out of our scope (averaging for recommendation), these studies confirm the importance of our research problem.

2.3 Problem Evaluation

To evaluate the consistency of $\mu_{\mathcal{U}}$ in accordance with our formulation in Section 2.2.2, we propose the following score:

$$\text{Consistency}_k(\mathcal{X}) = \mathbb{E}_{\mathcal{U} \in \mathcal{X}_k}\Big[\text{Precision}_k(\mathcal{U})\Big], \text{ where } \text{Precision}_k(\mathcal{U}) = \frac{|\mathcal{X}_k(\mu_{\mathcal{U}}) \cap \mathcal{U}|}{k} \in [0, 1], \tag{2}$$

for a given $k \in \{1, \ldots, N\}$. In essence, $\text{Precision}_k(\mathcal{U})$ measures the percentage of items from $\mathcal{U}$ among the k nearest neighbors of $\mu_{\mathcal{U}}$ in $\mathcal{X}$. A perfect precision of 1 indicates that $\mathcal{X}_k(\mu_{\mathcal{U}}) = \mathcal{U}$. Therefore, higher values of $\text{Consistency}_k(\mathcal{X})$ indicate that, on expectation, average embeddings computed from $\mathcal{X}$ will comprise more items used for their constructions in their neighborhood. The remainder of this paper provides an analysis of $\text{Consistency}_k(\mathcal{X})$ in a theoretical setting with assumptions of the distribution of embeddings in $\mathcal{X}$, and studies its empirical behavior on real-world embeddings.

3 A THEORETICAL ANALYSIS OF CONSISTENCY$_k(\mathcal{X})$

We dedicate Section 3 to our theoretical analysis of Consistency$_k(\mathcal{X})$. For clarity, we only present our setting, main results, and interpretation of these results in this section. We report all mathematical proofs in Appendices A and B.

3.1 Setting and Assumptions

We focus on the setting where $\mathcal{X}$ is a set of independent and identically distributed (i.i.d.) multi-dimensional random variables (r.v.). For every $X_i \in \mathcal{X}$, the elements $\{X_{i,j}\}_{1 \leq j \leq d}$ form a set of i.i.d uni-dimensional r.v. and we denote by μ, σ^2, γ, and κ the mean, variance, skewness, and kurtosis of their distribution, respectively. We assume that these moments are finite. s is the prevalent *inner product similarity*: $\forall (x, y) \in \mathbb{R}^d \times \mathbb{R}^d, s(x, y) = x^\intercal y = \sum_{i=1}^d x_i y_i$.

3.2 Main Results and Interpretations

In this theoretical setting, we obtain the following results on average embeddings.

Proposition 1. Let $k \in \{2, \dots, N\}$, $\mathcal{U} \in \mathcal{X}_k$, $u_{\text{in}} \in \mathcal{U}$, and $u_{\text{out}} \in \overline{\mathcal{U}}$. Then, under the hypotheses of Section 3.1, both $s_{\text{in}} = s(u_{\text{in}}, \mu_{\mathcal{U}})$ and $s_{\text{out}} = s(u_{\text{out}}, \mu_{\mathcal{U}})$ converge in probability to normal distributions as d increases. Their respective means and variances are provided in the proof of Appendix A. Moreover, under such distributions, we have:

$$\mathbb{P}\Big(s(u_{\text{in}}, \mu_{\mathcal{U}}) > s(u_{\text{out}}, \mu_{\mathcal{U}})\Big) = \frac{1}{2}\Big(1 + \text{erf}(\sqrt{\frac{d\sigma^2}{2((2(k-1)+\kappa)\sigma^2 + 2k\gamma\mu\sigma + 2k^2\mu^2)}})\Big), \tag{3}$$

where erf denotes the Gauss error function [1]: $\text{erf} : x \mapsto \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$.

Proposition 1 is a consequence of the *central limit theorem* [7] applied to inner product similarities. Figure 1 illustrates that, for dimension values such as $d = 128$ (a common choice in recommendation applications), the normal distribution faithfully approximates the distribution of similarities, regardless of the original embedding distribution. Hence, Equation (3) provides a reliable approximation of the probability that an element from $\mathcal{U}$ will be closer to its center $\mu_{\mathcal{U}}$ than a random point from $\overline{\mathcal{U}}$. It is worth noting that the probability is guaranteed to be greater than 0.5, as the expression within the erf function is positive. We also observe that the probability increases with d, illustrating that, as the dimension increases, the chances for a vector from $\overline{\mathcal{U}}$ to have a stronger similarity with $\mu_{\mathcal{U}}$ than elements of $\mathcal{U}$ diminishes.

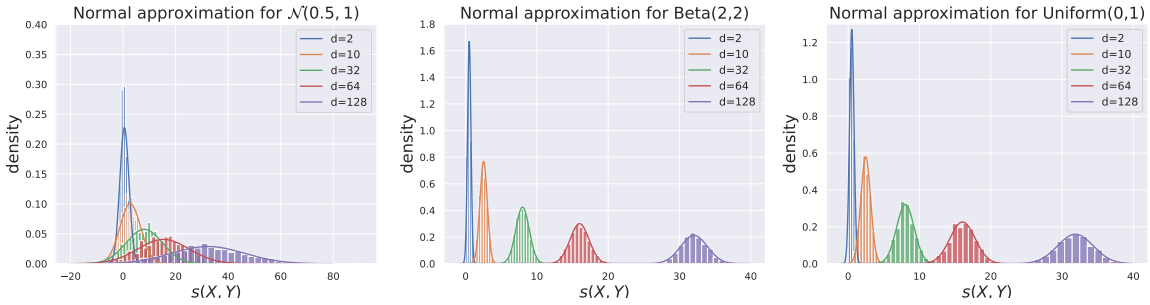
The case of centered embeddings ($\mu = 0$) is of particular interest. Indeed, for $\mu = 0$ and a fixed d, the probability becomes independent of σ and γ, and a decreasing function of k and κ. In essence, as k increases, it becomes increasingly difficult for $\mu_{\mathcal{U}}$ to remain similar to all $\mathcal{U}$ items, while remaining dissimilar to all $\overline{\mathcal{U}}$ items. The decrease with respect to the kurtosis κ relates to its interpretation as a measure of the propensity of a distribution to produce outliers [26]. Intuitively, the presence of outliers would impact the ability for $\mu_{\mathcal{U}}$ to remain similar to all $\mathcal{U}$ items and dissimilar to all $\overline{\mathcal{U}}$ items[2]. In the following, we continue to focus on $\mu = 0$, using results from Proposition 1 to express Consistency$_k(\mathcal{X})$.

Proposition 2. Under the hypotheses of Section 3.1 and approximated distributions of Proposition 1 with $\mu = 0$:

$$\text{Consistency}_k(\mathcal{X}) = \frac{1}{k} \sum_{i=1}^{k} \int_{-\infty}^{\infty} f_{\text{in},(i)}(x) \times F_{\text{out},(k-i+1)}(x)\, \text{d}x, \tag{4}$$

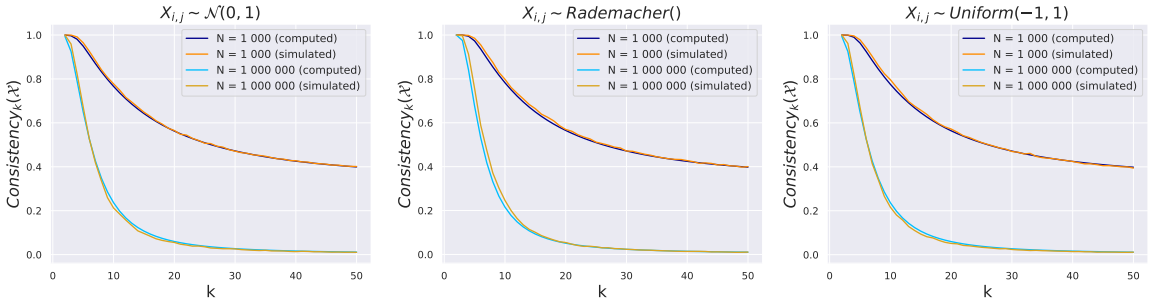
where explicit formulas for the $f_{\text{in},(i)}$ and $F_{\text{out},(k-i+1)}$ functions are provided in the proof of Appendix B.

[2]On the contrary, Equation (3) would be maximized by distributions with minimal kurtosis, such as a re-centered Bernoulli(0.5) or a Rademacher distribution ($X_{i,j} = 1$ or -1 with probability 0.5 each). We emphasize that these distributions are *discrete*, and might therefore allow for less precise similarity computations between vectors. This brings to light an interesting *trade-off* between similarity precision and average embedding consistency.



(a) (b) (c)

Fig. 1. Histograms of inner product similarity values between 1 000 d-dimensional vectors with entries randomly drawn from $\mathcal{N}(0.5, 1)$ (Figure 1a), Beta(2, 2) (Figure 1b), or Uniform(0, 1) (Figure 1c) distributions, with $d \in \{2, 10, 32, 64, 128\}$. Histograms undergo a rightward shift as d increases, since similarity computations involve summing more elements (see Section 3.1). Curves correspond to normal approximations of similarity distributions. We observe that, for the largest values of d, the normal distribution faithfully approximates all similarity distributions, an important result to validate the approximations of Propositions 1 and 2.



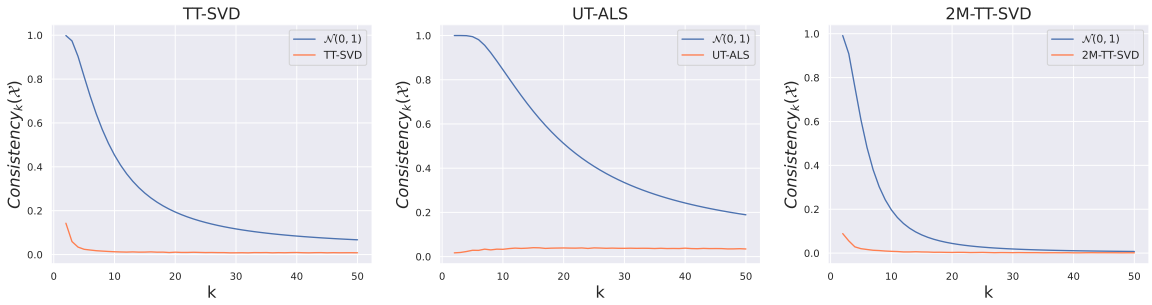
(a) (b) (c)

Fig. 2. Comparison of the Consistency$_k(\mathcal{X})$ scores obtained by a direct *computation* of Equation (4) to the ones estimated via numerical *simulations*, for $k \in \{2, \ldots, 50\}$, with $d = 128$, $N = 1\,000$ or $1\,000\,000$, and $X_{i,j} \sim \mathcal{N}(0, 1)$ (Figure 2a), $X_{i,j} \sim$ Rademacher() (Figure 2b), or $X_{i,j} \sim$ Uniform$(-1, 1)$ (Figure 2c). We used the Python library *scipy* to compute integrals from Equation (4). For numerical simulations, we sampled N vectors from the above three distributions. Then, for each k, we randomly picked a subset of k vectors and computed the precision of this subset (Equation 2). We repeated this operation 1 000 times for each value of k and reported averaged scores on figures.

Proposition 2 provides a useful approximated analytical expression of Consistency$_k(\mathcal{X})$. In Figure 2, we assess the accuracy of this expression by comparing, for various $X_{i,j}$ distributions and values of k, the Consistency$_k(\mathcal{X})$ score obtained from Equation (4) to the one estimated via numerical simulations. Our computed expression systematically coincides with the simulated score, validating the correctness of Proposition 2 and the relevance of our approximations.

Overall, we observe that Consistency$_k(\mathcal{X})$ decreases with k and the number of items $N = |\mathcal{X}|$. Regarding k, this result is coherent with our above interpretation of Proposition 1. Regarding N, such a result is also unsurprising. Indeed, the more vectors in $\mathcal{X}$, the greater the likelihood of some unrelated item embeddings becoming similar to $\mu_{\mathcal{U}}$ by chance.

Yet, in Figure 2, all scores remain quite high for $N = 1\,000$ (e.g., around 0.4 for $k = 50$). Importantly, for small values of k, the consistency of average embeddings remains close to 1 even for $N = 1\,000\,000$. Therefore, even with a large catalog of millions of items, as in music streaming services, averaging item embeddings appears as a consistent way to faithfully represent collections of a few items, provided that these embeddings comply with our theoretical assumptions.



(a) (b) (c)

Fig. 3. Consistency$_k(\mathcal{X})$ scores of centered versions of TT-SVD (Figure 3a), UT-ALS (Figure 3b), and 2M-TT-SVD embeddings (3c), for $k \in \{2, \dots, 50\}$ and the inner product similarity s. For comparison, figures also display the Consistency$_k(\mathcal{X})$ scores of embeddings generated from normal distributions, with the same dimension and number of items as the real-world embeddings under consideration. Average embeddings of real-world data are less consistent than those of data complying with our theoretical setting from Section 3.

4 FROM THEORY TO PRACTICE

4.1 Experimental Setting

In this Section 4, we analyze the empirical behavior of Consistency$_k(\mathcal{X})$ on real-world data, with the aim of discussing potential discrepancies with our results from Section 3. Our experiments focus on three *music track embeddings* datasets. As illustrated in this paper, music recommendation is especially prone to high-order concept learning: album, playlist, session, music genre, and user embeddings can all be obtained by averaging music track embeddings [3, 4, 9, 23].

Firstly, we consider two variants[3] of 50 000 music track embeddings publicly released by Deezer [4]. The first ones, denoted *TT-SVD embeddings*, consist of 128-dimensional vectors obtained by factorizing a track-track (TT) pointwise mutual information matrix computing track co-occurrences in Deezer playlists, using singular value decomposition (SVD) [4]. The second ones, denoted *UT-ALS embeddings*, are 256-dimensional vectors obtained by factorizing a user-track (UT) interaction matrix, using alternating least squares (ALS) [4].

Besides, we report results on *2M-TT-SVD embeddings*, a private dataset of two million 128-dimensional track embeddings, extracted from Deezer's production environment. These embeddings are computed using SVD on a co-occurrence matrix comparable to the TT-SVD one. We voluntarily omit technical details for confidentiality reasons.

4.2 Results and Discussion

Figure 3 reports our evaluation of Consistency$_k(\mathcal{X})$ scores for the three music track embeddings under consideration. Our experiments show that average embeddings of real-world data are less consistent for recommendation than those computed from embedding data explicitly complying with our theoretical setting from Section 3.

Specifically, one can not ensure that the average embedding of a collection of items will remain similar to the items present in this collection. Even for low values of k, the consistency of average embeddings does not surpass 14%, 6%, and 2 % for TT-SVD, 2M-TT-SVD, and UT-ALS embeddings, respectively. Consequently, even the average of two randomly selected embeddings would likely result in a vector whose two most similar neighbors will not be the selected embeddings themselves.

[3]We release our source code on GitHub to ensure the reproducibility of our experimental analysis on these two Deezer public datasets: https://github.com/deezer/consistency.

Regarding SVD-based embeddings (TT-SVD, 2M-TT-SVD), we observe that consistency scores drop as k increases, as in Figure 2. On the contrary, scores remain steady at around 2% for UT-ALS embeddings. This phenomenon highlights the dissimilarity in distributions of embeddings generated by different representation learning algorithms (SVD, ALS). In particular, the steady consistency of UT-ALS suggests that these embeddings might be more suitable for downstream applications involving average operations on large collections, while TT-SVD embeddings might be preferable for applications with low values of k, although more studies would be required in future work for confirmation.

Overall, our experiments also pave the way for future work to align real-world embeddings with the setting from Section 3, to improve the consistency of real-world averages. For instance, one could consider alternating SVD or ALS matrix reconstruction optimization steps with projections within the set of distributions complying with assumptions from Section 3. One could also examine adding a regularization term to the optimized loss during training, e.g., the Kullback-Leibler divergence of embeddings with a pre-selected complying distribution. In particular, these strategies could help to enforce the identical distribution of embedding dimensions. In addition, we note that, in Figure 3, we computed scores on *centered* embeddings. Besides being in line with Section 3, our tests revealed that this centering operation slightly improves the consistency of TT-SVD, UT-ALS, and 2M-TT-SVD average embeddings (albeit modifying initial similarities). Our future research will aim to further understand the impact of centering embeddings on consistencies.

5 CONCLUSION

This short paper proposed a rigorous study of the common practice consisting of averaging item embeddings in recommender systems. We provided a mathematical analysis of the consistency of these averaging operations in a general theoretical setting, as well as an empirical evaluation on real-world data. Our results revealed that real-world averages were less consistent than those computed in our theoretical setting. This sets the stage for future research directions, discussed in this paper, toward better aligning real-world data with our theoretical assumptions. Due to the prevalence of the embedding averaging practice in industry-oriented research and applications, we believe our study and proposed directions will be insightful for the recommendation community, and could eventually lead to the improvement of embedding-based recommender systems leveraging average embeddings for representation learning.

APPENDIX

In this supplementary section, we report the mathematical proofs of our Propositions 1 and 2 from Section 3.

A PROOF OF PROPOSITION 1

A.1 Preliminaries

Let X, Y, and Z be d-dimensional random variables (r.v.) composed of d independent and identically distributed (i.i.d.) uni-dimensional r.v. as elements. Distributions might differ between X,Y, and Z. Let $\mu_X, \sigma_X^2, \gamma_X, \kappa_X$ be the finite mean, variance, skewness, and kurtosis of the distribution of elements of X, respectively. Let $\mu_Y, \sigma_Y^2, \mu_Z, \sigma_Z^2$ be the finite mean and variance of the elements of Y and Z, respectively. When d increases, the following approximations hold:

$$\begin{aligned} s(X,Y) &\sim \mathcal{N}(\mu_{XY}, \sigma_{XY}^2), \\ s(X,X) &\sim \mathcal{N}(\mu_{XX}, \sigma_{XX}^2), \end{aligned} \tag{5}$$

with:

$$\begin{aligned} \mu_{XY} &= d \times \mu_X \mu_Y, \\ \sigma_{XY}^2 &= d \times ((\sigma_X^2 + \mu_X^2)(\sigma_Y^2 + \mu_Y^2) - \mu_X^2 \mu_Y^2), \\ \mu_{XX} &= d \times (\mu_X^2 + \sigma_X^2), \\ \sigma_{XX}^2 &= d \times (4\mu_X^2 \sigma_X^2 + 4\mu_X \gamma_X \sigma_X^3 + (\kappa_X - 1)\sigma_X^4). \end{aligned}$$

Indeed, we have $s(X,Y) = \sum_{i=1}^{d} X_i Y_i$. Each r.v. $X_i Y_i$ verifies $\mathbb{E}[X_i Y_i] = \mu_X \mu_Y$ and $\mathrm{Var}(X_i Y_i) = (\sigma_X^2 + \mu_X^2)(\sigma_Y^2 + \mu_Y^2) - \mu_X^2 \mu_Y^2$. $s(X,Y)$ being the sum of d i.i.d. r.v., according to the central limit theorem [7], it can be approximated by a normal distribution of expectation $d \times \mathbb{E}[X_i Y_i]$ and variance $d \times \mathrm{Var}(X_i Y_i)$ for large values of d, leading to the approximation for $s(X,Y)$ in Equation (5). A similar reasoning leads to the approximation for $s(X,X)$ in this same Equation (5).

Moreover, regarding covariances of these similarities, we have:

$$\begin{aligned} \mathrm{Cov}(s(X,Y), s(X,Z)) &= d \times \sigma_X^2 \mu_Y \mu_Z, \\ \mathrm{Cov}(s(X,X), s(X,Y)) &= d \times \mu_Y (\gamma_X \sigma_X^3 + 2\mu_X \sigma_X^2). \end{aligned} \tag{6}$$

Indeed:

$$\begin{aligned} \mathrm{Cov}(s(X,Y), s(X,Z)) &= \mathbb{E}[s(X,Y)s(X,Z)] - \mathbb{E}[s(X,Y)]\mathbb{E}[s(X,Z)] \\ &= \mathbb{E}[(\sum_{i=1}^{d} X_i Y_i)(\sum_{j=1}^{d} X_j Y_j)] - d \times \mu_X \mu_Y \times d \times \mu_X \mu_Z \\ &= \mathbb{E}[\sum_{i=1}^{d} (X_i Y_i X_i Z_i + \sum_{\substack{j=1 \\ j \neq i}}^{d} X_i Y_i X_j Z_j)] - d^2 \times \mu_X^2 \mu_Y \mu_Z \\ &= \sum_{i=1}^{d} (\mathbb{E}[X_i^2]\mathbb{E}[Y_i]\mathbb{E}[Z_i] + \sum_{\substack{j=1 \\ j \neq i}}^{d} (\mathbb{E}[X_i]\mathbb{E}[Y_i]\mathbb{E}[X_j]\mathbb{E}[Z_j])) - d^2 \times \mu_X^2 \mu_Y \mu_Z \\ &= d \times ((\sigma_X^2 + \mu_X^2)\mu_Y \mu_Z + (d-1)(\mu_X^2 \mu_Y \mu_Z)) - d^2 \times \mu_X^2 \mu_Y \mu_Z \\ &= d \times \sigma_X^2 \mu_Y \mu_Z. \end{aligned}$$

We obtain the second covariance of Equation (6) with similar computations. Notice how $\mu_Y = 0$ or $\mu_Z = 0$ implies that similarities are uncorrelated and, therefore, also independent when they follow multivariate normal distributions [13].

A.2 Distribution of $s_{\mathbf{in}} = s(u_{\mathbf{in}}, \mu_{\mathcal{U}})$ and $s_{\mathbf{out}} = s(u_{\mathbf{out}}, \mu_{\mathcal{U}})$

Using the distributive property of the inner product s, we have:

$$s_{\text{out}} = \frac{\sum_{i=1}^{k} s(u_{\text{out}}, u_i)}{k},$$

i.e., the sum of k correlated identically distributed normal distributions. The expectation of the sum is thus:

$$\mathbb{E}[s_{\text{out}}] = d \times \mu^2.$$

We compute its variance using Bienaymé's identity [17] stating that, for k r.v. $(A_i)_{1 \leq i \leq k}$, then:

$$\text{Var}\,(\sum_{i=1}^{k} A_i) = \sum_{i=1}^{k} \text{Var}\,(A_i) + \sum_{\substack{i,j=1 \\ i \neq j}}^{k} \text{Cov}\,(A_i, A_j) = \sum_{i,j=1}^{k} \text{Cov}\,(A_i, A_j).$$

Also considering $A_i = s(u_{\text{out}}, u_i)$, and using the first half of both Equations (5) and (6), we obtain:

$$\begin{aligned}\text{Var}\,(s_{\text{out}}) &= \frac{kd(\sigma^4 + 2\sigma^2\mu^2) + k(k-1)d\sigma^2\mu^2}{k^2} \\ &= d\frac{(\sigma^4 + (k+1)\sigma^2\mu^2)}{k}.\end{aligned}$$

Besides, under preliminary approximations:

$$s_{\text{in}} = \frac{s(u_{\text{in}}, u_{\text{in}}) + \sum_{i \neq \text{in}} s(u_{\text{in}}, u_i)}{k}$$

is the sum of k correlated normal distributions, i.e., a normal distribution with:

$$\mathbb{E}[s_{\text{in}}] = d(\mu^2 + \frac{\sigma^2}{k}).$$

Using Bienaymé's identity and Equations (5) and (6), we get:

$$\begin{aligned}\text{Var}\,(s_{\text{in}}) &= d \times \frac{4\mu^2\sigma^2 + 4\mu\gamma\sigma^3 + (\kappa-1)\sigma^4 + (k-1)(\sigma^4 + 2\sigma^2\mu^2) + 2(k-1)\mu(\gamma\sigma^3 + 2\mu\sigma^2) + (k-1)(k-2)\sigma^2\mu^2}{k^2} \\ &= \frac{d \times (k(k+3)\mu^2\sigma^2 + 4\mu\gamma\sigma^3 + (\kappa + k - 2)\sigma^4)}{k^2}.\end{aligned}$$

A.3 Distribution of $s_{\mathbf{diff}} = s_{\mathbf{in}} - s_{\mathbf{out}}$

Using preliminary approximations, and since the difference of two normally distributed r.v. is also normally distributed [13], we obtain that $s_{\text{diff}} = s(u_{\text{in}}, \mu_{\mathcal{U}}) - s(u_{\text{out}}, \mu_{\mathcal{U}})$ follows a normal distribution with:

$$\begin{aligned}\mathbb{E}[s_{\text{diff}}] &= \mathbb{E}[s(u_{\text{in}}, \mu_{\mathcal{U}})] - \mathbb{E}[s(u_{\text{out}}, \mu_{\mathcal{U}})] \\ &= d\frac{\sigma^2}{k}.\end{aligned}$$

To compute its variance, we remark that:

$$\begin{aligned} s_{\text{diff}} &= s(u_{\text{in}}, \mu_{\mathcal{U}}) - s(u_{\text{out}}, \mu_{\mathcal{U}}) \\ &= \frac{1}{k}(s(u_{\text{in}}, u_{\text{in}}) + s(u_{\text{in}}, \sum_{j \neq \text{in}}^{k} u_j) - s(u_{\text{out}}, u_{\text{in}}) - s(u_{\text{out}}, \sum_{j \neq \text{in}} u_j)) \\ &= \frac{s(X,X) + s(X,Z) + s(Y,X) + s(Y,Z)}{k}, \end{aligned}$$

with $X = u_{\text{in}}$, $Y = -u_{\text{out}}$ and $Z = \sum_{j \neq \text{in}} u_j$. Since X, Y, and Z are three independent multidimensional normal distributions, for which dimensions are i.i.d. with expectation $\mu, -\mu, (d-1)\mu$ and variance $\sigma^2, \sigma^2, (d-1)^2\sigma^2$, we once again use Bienaymé's identity with Equations (5) and (6) to obtain:

$$\text{Var}\,(s_{\text{diff}}) = d\frac{(2(k-1)+\kappa)\sigma^4 + 2k\gamma\mu\sigma^3 + 2k^2\sigma^2\mu^2}{k^2}.$$

Finally, using the cumulative distribution function of normal distributions [13], we get Equation (3):

$$\begin{aligned} &\mathbb{P}(s_{\text{diff}} > 0) = 1 - \mathbb{P}(s_{\text{diff}} \leq 0) = \frac{1}{2}(1 - \text{erf}(-\frac{\mathbb{E}[s_{\text{diff}}]}{\sqrt{2\text{Var}\,(s_{\text{diff}})}})) \\ \iff &\mathbb{P}(s(u_{\text{in}}, \mu_{\mathcal{U}}) > s(u_{\text{out}}, \mu_{\mathcal{U}})) = \frac{1}{2}(1 + \text{erf}(\sqrt{\frac{d\sigma^2}{2((2(k-1)+\kappa)\sigma^2 + 2k\gamma\mu\sigma + 2k^2\mu^2)}})). \end{aligned}$$

B PROOF OF PROPOSITION 2

This proof starts by evaluating $p_{(\frac{i}{k})} = \mathbb{P}(\text{Precision}_k(\mathcal{U}) = \frac{i}{k})$ for $i \in \{1, \ldots, k\}$, and then computes $\text{Consistency}_k(\mathcal{X}) = \mathbb{E}[\text{Precision}_k(\mathcal{U})] = \sum_{i=1}^{k} \frac{i}{k} \times p_{(\frac{i}{k})}$ under the studied hypotheses. To begin with, we set:

$$p^{+}_{(\frac{i}{k})} = \mathbb{P}(\text{Precision}_k(\mathcal{U}) \geq \frac{i}{k}), \text{ for } i \in \{1, \ldots, k+1\}.$$

As i takes integer values:

$$p_{(\frac{i}{k})} = p^{+}_{(\frac{i}{k})} - p^{+}_{(\frac{i+1}{k})}.$$

Assuming $p^{+}_{(\frac{k+1}{k})} = 0$ by definition, we have:

$$\mathbb{E}\Big[\text{Precision}_k(\mathcal{U})\Big] = \sum_{i=1}^{k} \frac{i}{k} \times (p^{+}_{(\frac{i}{k})} - p^{+}_{(\frac{i+1}{k})}) = \frac{1}{k}\sum_{i=1}^{k} p^{+}_{(\frac{i}{k})}. \tag{7}$$

Let us denote by $s_{\text{in},(i)}$ the distribution of the i^{th} highest value of $S_{\text{in}} = \{s(u_{\text{in}}, \mu_{\mathcal{U}}), u_{\text{in}} \in \mathcal{U}\}$, and by $s_{\text{out},(k-i+1)}$ the $(k-i+1)^{\text{th}}$ highest value of $S_{\text{out}} = \{s(u_{\text{out}}, \mu_{\mathcal{U}}), u_{\text{out}} \in \overline{\mathcal{U}}\}$, for $i \in \{1, \ldots, k\}$. Using this notation, we observe that:

$$p^{+}_{(\frac{i}{k})} = \mathbb{P}(s_{\text{in},(i)} > s_{\text{out},(k-i+1)}).$$

Indeed, as illustrated in Figure 4, we can show that $s_{\text{in},(i)} > s_{\text{out},(k-i+1)} \iff \text{Precision}_k(\mathcal{U}) \geq \frac{i}{k}$:

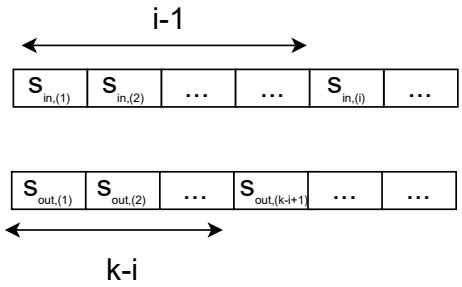


Fig. 4. Ranking s_{in} and s_{out} statistics.

- If $s_{\text{in},(i)} > s_{\text{out},(k-i+1)}$ then at most $k-i$ elements of S_{out} are greater than $s_{\text{in},(i)}$. Also, by definition, exactly $i-1$ elements of S_{in} are greater than $s_{\text{in},(i)}$. So, at most $k-1$ similarities are greater than $s_{\text{in},(i)}$ overall. Thus, $s_{\text{in},(i)}$ is one of the k highest values of $S_{\text{in}} \bigcup S_{\text{out}}$ and $\text{Precision}_k(\mathcal{U}) \geq \frac{i}{k}$.
- If $\text{Precision}_k(\mathcal{U}) \geq \frac{i}{k}$, then at least i elements of S_{in} are in the top-k of $S_{\text{in}} \bigcup S_{\text{out}}$, including $s_{\text{in},(i)}$. That leaves at most $k-i$ slots available in the top-k. Since, by definition, exactly $k-i$ elements of S_{out} are greater than $s_{\text{out},(k-i+1)}$, $s_{\text{out},(k-i+1)}$ is necessarily outside of the top-k elements of $S_{\text{in}} \bigcup S_{\text{out}}$, and so $s_{\text{in},(i)} > s_{\text{out},(k-i+1)}$.

In our setting, we derive $s_{\text{in},(i)}$ by noticing that the distribution of the i$^{\text{th}}$ highest value of S_{in} is also the distribution of its $(k-i)^{\text{th}}$ lowest value. Hence, $s_{\text{in},(i)}$ is the $(k-i)^{\text{th}}$ order statistic of S_{in}. As $\mu = 0$, we have that, given two distinct elements of $\mathcal{U}$, say u_1 and u_2, $\text{Cov}\left(s(u_1, \mu_{\mathcal{U}}), s(u_2, \mu_{\mathcal{U}})\right) = 0$, which implies that all elements of S_{in} are i.i.d. and that the probability density function of the $(k-i)^{\text{th}}$ order statistic of S_{in} is [19]:

$$f_{\text{in},(k-i)}(x) = \frac{k!}{\sigma_{\text{in}} 2^{k-1}(k-i)!\,(i-1)!}\phi\left(\frac{x-\mu_{\text{in}}}{\sigma_{\text{in}}}\right)\left(1+\text{erf}\left(-\frac{x-\mu_{\text{in}}}{\sqrt{2}\sigma_{\text{in}}}\right)\right)^{k-i}\left(1-\text{erf}\left(-\frac{x-\mu_{\text{in}}}{\sqrt{2}\sigma_{\text{in}}}\right)\right)^{i-1},$$

with:

$$\mu_{\text{in}} = d\frac{\sigma^2}{k},$$
$$\sigma_{\text{in}} = \sigma^2\frac{\sqrt{d(\kappa+k-2)}}{k},$$
$$\phi : x \mapsto \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt.$$

Similarly, $s_{\text{out},(k-i+1)}$ is the $(N-2k+i)^{\text{th}}$ order statistic of S_{out} and so its cumulative density function is [19]:

$$F_{\text{out},(k-i+1)}(x) = \sum_{j=N-2k+i}^{N-k}\binom{N-k}{j}\left(1+\text{erf}\left(-\frac{x}{\sqrt{2}\sigma_{\text{out}}}\right)\right)^{j}\left(1-\text{erf}\left(-\frac{x}{\sqrt{2}\sigma_{\text{out}}}\right)\right)^{N-k-j},$$

with:

$$\sigma_{\text{out}} = \sigma^2\sqrt{\frac{d}{k}}.$$

Finally, we obtain:

$$\begin{aligned} p^{+}_{\left(\frac{i}{k}\right)} &= \mathbb{P}\left(s_{\text{in},(i)} > s_{\text{out},(k-i+1)}\right) \\ &= \int_{-\infty}^{\infty} f_{\text{in},(i)}(x) \times F_{\text{out},(k-i+1)}(x)\ \text{d}x, \end{aligned} \tag{8}$$

and, by injecting Equation (8) into Equation (7), we retrieve the expression from Equation (4) of Proposition 2.

REFERENCES

[1] Larry C Andrews. 1998. *Special Functions of Mathematics for Engineers*. Vol. 49. Spie Press.

[2] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-To-Beat Baseline for Sentence Embeddings. In *Proceedings of the 5th International Conference on Learning Representations*.

[3] Walid Bendada, Guillaume Salha-Galvan, Thomas Bouabça, and Tristan Cazenave. 2023. A Scalable Framework for Automatic Playlist Continuation on Music Streaming Services. In *Proceedings of the 46th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*. 464–474.

[4] Léa Briand, Guillaume Salha-Galvan, Walid Bendada, Mathieu Morlon, and Viet-Anh Tran. 2021. A Semi-Personalized System for User Cold Start Recommendation on Music Streaming Apps. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2601–2609.

[5] Ben Coleman. 2020. Why is it Okay to Average Embeddings? Technical post on: https://randorithms.com/2020/11/17/Adding-Embeddings.html.

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for Youtube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 191–198.

[7] Hans Fischer. 2011. *A History of the Central Limit Theorem: from Classical to Modern Probability Theory*. Springer.

[8] Mihajlo Grbovic and Haibin Cheng. 2018. Real-Time Personalization using Embeddings for Search Ranking at Airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 311–320.

[9] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. 2020. Contextual and Sequential User Embeddings for Large-Scale Music Recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 53–62.

[10] Lamis Hassanieh, Chadi Abou Jaoudeh, Jacques Bou Abdo, and Jacques Demerjian. 2018. Similarity Measures for Collaborative Filtering Recommender Systems. In *Proceedings of the 2018 IEEE Middle East and North Africa Communications Conference*. IEEE, 1–5.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[12] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-Based Recommendations with Recurrent Neural Networks. *Proceedings of the 4th International Conference on Learning Representations* (2015).

[13] Jean Jacod and Philip Protter. 2004. *Probability Essentials*. Springer Science & Business Media.

[14] Gourav Jain, Tripti Mahara, and Kuldeep Narayan Tripathi. 2020. A Survey of Similarity Measures for Collaborative Filtering-Based Recommender System. In *Soft Computing: Theories and Applications*. Springer, 343–352.

[15] Umair Javed, Kamran Shaukat, Ibrahim A Hameed, Farhat Iqbal, Talha Mahboob Alam, and Suhuai Luo. 2021. A Review of Content-Based and Context-Based Recommendation Systems. *International Journal of Emerging Technologies in Learning* 16, 3 (2021), 274–306.

[16] Yehuda Koren and Robert Bell. 2015. Advances in Collaborative Filtering. *Recommender Systems Handbook* (2015), 77–118.

[17] Michel Loève. 1977. Probability Theory I. *Graduate Texts in Mathematics* 4 (1977).

[18] Mridul K Mishra and Jaydeep Viradiya. 2019. Survey of Sentence Embedding Methods. *International Journal of Applied Science and Computations* 6, 3 (2019), 592–592.

[19] HN Nagaraja and HA David. 2003. *Order Statistics*. John Wiley & Sons, Inc.

[20] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-Based News Recommendation for Millions of Users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1933–1942.

[21] Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. 2020. PinnerSage: Multi-Modal User Embedding Framework for Recommendations at Pinterest. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2311–2320.

[22] Chanyoung Park, Donghyun Kim, Xing Xie, and Hwanjo Yu. 2018. Collaborative Translational Metric Learning. In *Proceedings of the 2018 IEEE International Conference on Data Mining*. IEEE, 367–376.

[23] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. 2018. Current Challenges and Visions in Music Recommender Systems Research. *International Journal of Multimedia Information Retrieval* 7, 2 (2018), 95–116.

[24] Viet-Anh Tran, Guillaume Salha-Galvan, Romain Hennequin, and Manuel Moussallam. 2021. Hierarchical Latent Relation Modeling for Collaborative Metric Learning. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 302–309.

[25] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-Scale Commodity Embedding for E-Commerce Recommendation in Alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 839–848.

[26] Peter H Westfall. 2014. Kurtosis as peakedness, 1905–2014. RIP. *The American Statistician* 68, 3 (2014), 191–195.

[27] Lyndon White, Roberto Togneri, Wei Liu, and Mohammed Bennamoun. 2015. How Well Sentence Embeddings Capture Meaning. In *Proceedings of the 20th Australasian Document Computing Symposium*. 1–8.

[28] Chen Wu and Ming Yan. 2017. Session-Aware Information Embedding for E-Commerce Product Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2379–2382.

[29] Zhe Yang, Bing Wu, Kan Zheng, Xianbin Wang, and Lei Lei. 2016. A Survey of Collaborative Filtering-Based Recommender Systems for Mobile Internet Applications. *IEEE Access* 4 (2016), 3273–3287.

[30] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.