# Theoretical Understandings of Product Embedding for E-commerce Machine Learning

Da Xu
Walmart Labs
Sunnyvale, California, USA
DaXu5180@gmail.com

Chuanwei Ruan*
Instacart
San Francisco, California, USA
RuanChuanwei@gmail.com

Evren Korpeoglu
Sushant Kumar
Kannan Achan
Walmart Labs
Sunnyvale, California, USA
[EKorpeoglu,SKumar4,KAchan]
@walmartlabs.com

## ABSTRACT

Product embeddings have been heavily investigated in the past few years, serving as the cornerstone for a broad range of machine learning applications in e-commerce. Despite the empirical success of product embeddings, little is known on how and why they work from the theoretical standpoint. Analogous results from the natural language processing (NLP) often rely on domain-specific properties that are not transferable to the e-commerce setting, and the downstream tasks often focus on different aspects of the embeddings. We take an e-commerce-oriented view of the product embeddings and reveal a complete theoretical view from both the representation learning and the learning theory perspective. We prove that product embeddings trained by the widely-adopted skip-gram negative sampling algorithm and its variants are sufficient dimension reduction regarding a critical product relatedness measure. The generalization performance in the downstream machine learning task is controlled by the alignment between the embeddings and the product relatedness measure. Following the theoretical discoveries, we conduct exploratory experiments that supports our theoretical insights for the product embeddings.

## CCS CONCEPTS

• **Mathematics of computing** → **Probability and statistics**; • **Information systems** → **Information retrieval**; • **Theory of computation** → **Machine learning theory**.

## KEYWORDS

Representation learning; Product relation; Information theory; Sufficient dimension reduction; Machine learning theory

*Work was done when the author was with Walmart Labs.


*WSDM '21, March 8–12, 2021, Virtual Event, Israel*

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441736

**ACM Reference Format:**
Da Xu, Chuanwei Ruan, and Evren Korpeoglu, Sushant Kumar, Kannan Achan. 2021. Theoretical Understandings of Product Embedding for E-commerce Machine Learning. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3437963.3441736

## 1 INTRODUCTION

Model interpretation and understanding play a critical role in e-commerce machine learning. Unlike the other domains where deep learning algorithms are being favored regardless of their black-box nature, model interpretability is often equally important as empirical performance in e-commerce, due to its closer connections with business and customers, as well as a more profound impact on revenue and social accountability including privacy, security and fairness [19, 31]. The underlying theoretical properties often justify model interpretation. Without sufficient theoretical understandings, model developers need to rely on intuitions and unverifiable assumptions to explain the inductive bias rather than providing reliable theoretical support and guarantee, which can easily result in mismatches between the purpose of model design and the actual working mechanism. It poses severe challenges on model diagnostics, which is an indispensable part of any industrial deployment. On the other hand, sacrificing the interpretability for a higher model complexity may not lead to better empirical performance. Several recent papers have challenged the state-of-the-art deep learning recommendation algorithms against the vanilla collaborative filtering, ending up finding worse performances from deep learning on various benchmark datasets [8, 21]. All the above concerns motivate our exploration of the theoretical perspective of product embeddings - the cornerstone for a considerable amount of machine learning models in e-commerce [4, 12, 25, 28–30].

Modern e-commerce machine learning favors the embedding models over classical feature-based approaches because of their computation efficiency and compatibility with more model architectures. Training product embeddings using the skip-gram negative sampling (SGNS) algorithm and its variants are highly efficient and scalable, even for billions of items and records [17, 18, 28]. A handful of open-source subroutines are available for easy implementation and modification under problem-specific needs [11]. By treating the product embeddings as vectors (usually of several hundred dimensions) encoded with useful product information, the computations of downstream tasks are simplified after converted

to the low-dimensional Euclidean space [15]. Replacing the product features with embedding in the downstream tasks significantly enriches the candidate models and reduces the feature engineering costs [5, 7]. The incompatibility issue is common for industrial problems, since the product features are often mixtures of quantitative and categorical variables that expand a huge irregular space that few modern machine learning models are suitable. Different approaches have been proposed to train product embeddings using SGNS with problem-specific modification, including *Item2vec* [4], *Prod2vec* [12, 25], *Triple2vec* [27], *MetaPath2vec* [9], *CompProd2vec* (complementary product embedding) [29] and product knowledge graph [30]. Despite the different formulations of the input data structure and regularization, the core component remains to be the SGNS induced by the input co-occurrence statistic (see Section 2 for details). The promising results from industrial applications also support the progress in the academic research on product embedding. Some of the papers have reported successful deployments, highlighting product embeddings as part of the mature solution for various online services.

Nevertheless, our understanding of product embedding is still inadequate to the classical models such as collaborative filtering and factorization machines [20, 22, 23]. Despite the conjectures and claims that product embedding encodes the useful features and relations, research in this domain has yet found an exact mapping to unveil how product embedding captures the signals and why they are useful for downstream tasks. The recent progress for the model understanding in the NLP domain, though highlighting certain aspects of the SGNS algorithm [2, 3, 6, 16, 24], does not directly transfer to the e-commerce setting due to different emphasis and data assumptions. We provide in-depth discussions in Section 2. In general, the product embedding is only partially understood, and the remaining unknown factors may still raise concerns from time to time and set barricades for the more thorough analysis and further improvements.

Our work is dedicated to providing an advanced theoretical understanding of product embedding for e-commerce machine learning. The first key result establishes the equivalence between training product embedding and finding the sufficient dimension reduction [10] of a *product relatedness measure* induced by the *co-occurrence statistic*. The product embedding, as a consequence, is optimal in an information-theoretical perspective. We then highlight several properties of the product relatedness measure, including its finite-sample tail bound and several domain-specific functionalities. The second key result shows the generalization bound for using product embedding in downstream machine learning tasks. It turns out that the generalization error is controlled by the alignment between the spectral spaces of product embedding and the product relatedness measure. In summary, we provide advanced theoretical understandings by answering: 1. what data distribution is product embedding representing; 2. how product embedding is representing the signal in e-commerce data; 3. why product embedding is useful for downstream tasks.

The practical implications of our results are two folds. Firstly, since product embedding is an (information-theoretical) optimal dimension reduction of the problem-specific product relatedness measure, its quality (meaningfulness) depends on the relatedness measure, which can be examined using the tail bound in Section 4. Secondly, the applicability (usefulness) of product embedding in downstream tasks can be examined in advance by checking how well they reconstruct the eigenspace of the product relatedness measure. They provide further understandings and some guidelines for obtaining more meaningful and useful product embedding, which we give a thorough exploration in our experiments. To the best of our knowledge, our paper provides the first advanced theoretical understanding of product embedding, and we conclude our contributions as follow.

- We establish the equivalence between training product embedding and sufficient dimension reduction with respect to the product relatedness measure.
- We provide the finite-sample tail bound and verify several domain-specific functionalities for the relatedness measure.
- We give a generalization bound for using product embedding in downstream tasks, and further illustrate our theoretical arguments via experiments.

## 2 BACKGROUND AND RELATED WORK

By convention, we use uppercase letters to denote random variables, lowercase letters to denote observations and scalars, bold-font letters to denote vectors and matrices, $D_{KL}(p \,\|\, q)$ to denote the Kullback-Leibler divergence between distribution $P$ and $Q$ with the corresponding density function $p, q$.

| | |
|---|---|
| $\mathcal{I}$, $\mathrm{Neg}(\mathcal{I})$ | The set of all products, and the negative samples drawn from $\mathcal{I}$. |
| $\mathcal{N}(i)$ | The neighborhood set for product $i \in \mathcal{I}$. |
| $\mathcal{D}(\mathcal{N})$ | The data generating mechanism with respect to the input data structure as well as the definition of neighborhood $\mathcal{N}(.)$. See Figure 1 for examples. We omit the dependency on $\mathcal{N}(.)$ for notation simplicity when no confusion arises. |
| $P_i(\mathcal{D})$, $P_{i,j}(\mathcal{D})$ | The marginal frequency of product $i$ and product pair $i, j$ for $i, j \in \mathcal{I}$, with respect to the data generating mechanism $\mathcal{D}$. |
| $n$, $N_i(\mathcal{D})$, $N_{i,j}(\mathcal{D})$ | The total number of records, and the (co-)occurrence statistics such that $P_i(\mathcal{D}) = N_i(\mathcal{D})/n$ and $P_{i,j}(\mathcal{D}) = N_{i,j}(\mathcal{D})/n$. |
| $\mathbf{z}_i, \tilde{\mathbf{z}}_i \in \mathbb{R}^d$ | The two embeddings for product $i \in \mathcal{I}$. The additional embedding helps handling the asymmetric product relations such that $\langle \mathbf{z}_i, \tilde{\mathbf{z}}_j \rangle \neq \langle \mathbf{z}_j, \tilde{\mathbf{z}}_i \rangle$. |
| $p(O = 1 \mid i, j)$ | The probability that product $i$ and $j$ co-occur in the same neighborhood, i.e. $p\big(1[j \in \mathcal{N}(i)]\big)$. |

**Table 1: Notations. Notice that $P_i(\mathcal{D})$, $P_{i,j}(\mathcal{D})$, $N_i(\mathcal{D})$ and $N_{i,j}(\mathcal{D})$ are random variables with their stochsticity induced by the data generating mechanism $\mathcal{D}(\mathcal{N})$.**

Product embeddings are trained on the input data that structured specifically to reflect particular product relations. The *skip-gram negative sampling* algorithm optimizes the embeddings to capture the desired product relation via inner products [17, 18]. Product

pairs from the same neighborhood are likely to have closer relations and are hence treated as positive samples. A fundamental difference between learning product embedding and word embedding is that the notion of neighborhood can be random variables in the e-commerce setting. The stochasticity in neighborhood may be induced by its own definition, e.g. the outcome of random walks (*MetaPath2vec* [9], *ProdNode2vec* [28]), or by the follow-up sampling steps, e.g. sampling the products from a given context window (*CompProd2vec* [29]). Therefore, the realization of neighborhood varies for different input data structures and problem settings (Figure 1), and the resulting co-occurrence statistics are random variables as well. To better notate the dependency on the underlying data generating mechanism as well as the neighborhood definition, we design our notation system as shown in Table 1.

As an unsupervised learning approach, the SGNS objective function is designed to characterize the contrasts between positive and negative samples using the sigmoid function:

$$
\begin{aligned}
p(O=1|i,j) &= \sigma(\mathbf{z}_i^\intercal \tilde{\mathbf{z}}_j) = \frac{1}{1+\exp(-\mathbf{z}_i^\intercal \tilde{\mathbf{z}}_j)}, \\
p(O=0|i,j) &= \sigma(-\mathbf{z}_i^\intercal \tilde{\mathbf{z}}_j) = \frac{1}{1+\exp(\mathbf{z}_i^\intercal \tilde{\mathbf{z}}_j)}, \\
\ell_{i,j} &= -\log p(O=1|i,j) + k \cdot \mathbb{E}_{k\sim \mathrm{Neg}(\mathcal{I})}\left[\log p(O=0|i,k)\right],
\end{aligned}
\tag{1}
$$

here $k$ is the number of negative samples. Without loss of generality, the modified objective functions for different product embedding algorithms can be summarized by:

$$
\ell = \sum_{i,j\in\mathcal{N}(i)} \ell_{i,j} + \mathrm{Reg}(\mathbf{Z},\tilde{\mathbf{Z}}),\ \mathbf{Z}=[\mathbf{z}_1,\ldots,\mathbf{z}_{|\mathcal{I}|}],\ \tilde{\mathbf{Z}}=[\tilde{\mathbf{z}}_1,\ldots,\tilde{\mathbf{z}}_{|\mathcal{I}|}],
\tag{2}
$$

where $\mathrm{Reg}(\mathbf{Z},\tilde{\mathbf{Z}})$ is the (implicit) problem-specific regularization on the product embeddings [1], e.g. products with similar features should have closer embeddings, or the regularization takes account of the user bias whenever user id is presented in the data. Set aside the regularizations, the NLP community has made considerable efforts to shed insights to the SGNS objective function [2, 3, 6, 16, 24]. Although the skip-gram model has been well understood [14], the negative sampling has changed the algorithm fundamentally. The major breakthrough was made in [16], showing that the optimal embedding is given by the point-wise mutual information (PMI) matrix. However, their result requires a full rank assumption on the embedding matrix, i.e. the embedding dimension is larger than the number of entities, which is not practical. Other papers that build direct connections between PMI and embedding either require a specific type of entity distribution or a underlying generative model from embeddings [3, 6, 24], which are not reasonable for the e-commerce setting. Two recent papers have provided probabilistic interpretation for embeddings trained by SGNS [1, 2], nonetheless, they focus on the compositional properties of trained embeddings rather than the training process and downstream tasks.

The notion of sufficient dimension reduction is inspired by the sufficient statistics of exponential family. Broadly speaking, when predicting $Y$ with $X$, if all the information in $X$ about $Y$ can be

[1]The original papers do not introduce their objectives by this form, and it is possible that an explicit expression for the regularizer is nonexistent. However, by viewing the algorithms in this way, we distinguish the core component and the problem-specific structures, which is necessary for analytical purposes.

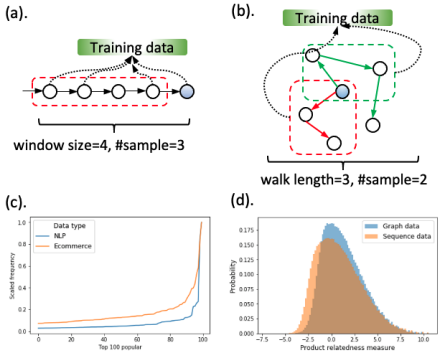


**Figure 1: (a), (b): Visual example of generating training data for SGNS using sequence and graph structure. The sampling steps induce the randomness in training data. (c). The scaled frequency for the 100 most occurred words/products in an NLP corpus and a pubic e-commerce data (see Section 6). Observe that the overlap only occurs for the top few cases. (d). The empirical distribution of the product relatedness measure $R_{i,j}$, for both the sequence-structured and graph-structured training data (as shown in (a) and (b)), generated from the pubic e-commerce data.**

compressed into a dimension reduction $f(X)$, i.e. $Y \perp X \,|\, f(X)$ or $p(Y|X)$ and $p\big(Y|f(X)\big)$ are the same, then $f(X)$ is a sufficient dimension reduction (SDR). Finding the SDR $f(X)$ is equivalent to solving a constraint minimax optimization [10]:

$$
\max_{f(X)} \quad \min_{\substack{q(y,x):\\ \int_x q(y,x)\mathrm{d}x=p(y),\, \int_y q(y,x)\mathrm{d}y=p(x),\\ \mathbb{E}_{q(x|y)}[f(X)]=\mathbb{E}_{p(x|y)}[f(X)]}} I\big(q(Y,X)\big),
\tag{3}
$$

where $I\big(q(Y,X)\big)$ is the Shannon mutual information, and $p(x)$, $p(y)$ corresponds to the marginal distribution of $p(x,y)$. Briefly put, the above minimax problem introduces a proxy distribution $q(y,x)$ induced by the SDR $f(X)$ (reflected by the third constraint under minimization) while maintaining the marginal distribution of $X$ and $Y$ (see the first two constraints under minimization). The SDR $f(X)$ is then optimized under the "maximum entropy principle", by maximizing the Shannon mutual information of $q(y,x)$. Intuitively speaking, SDR is finding the dimension reduction that compresses the maximum amount of information while still agreeing with the observed data distribution. Applying the variational principle and strong duality arguments, it has been shown in [10] that the dual problem is given by:

$$
\min_{q\left(y,f(x)\right)} D_{KL}\big(p(Y,X)\,\|\,q(Y,f(X))\big).
\tag{4}
$$

Sufficient dimension reduction is a powerful tool to examine the information-theoretical optimality of compressed representation, which we prove for the product embedding in the following section.

## 3 NONLINEAR PROJECTION AND SUFFICIENT DIMENSION REDUCTION

Formally, apart from the embedding regularization terms in (2), the objective function of SGNS for training product embedding has the alternative expression by aggregating the contribution to the

positive and negative samples from each item pair:

$$\ell(\mathcal{D}) = \sum_{i,j\in\mathcal{I}} N_{i,j}(\mathcal{D}) \log \sigma(\mathbf{z}_i^\intercal \tilde{\mathbf{z}}_j) + \frac{k}{n} N_i(\mathcal{D}) N_j(\mathcal{D}) \log \sigma(-\mathbf{z}_i^\intercal \tilde{\mathbf{z}}_j). \tag{5}$$

Note that $\ell_{\text{SGNS}}$ is a random variable because it conditions on the unknown data generating mechanism $\mathcal{D}$. When the input data, its structure and the definition of neighborhood are given, i.e. $\mathcal{D}$ is realized, the loss function is then fixed and observed. The loss function may seem peculiar at the first glance, but it is a special instance of nonlinear projection of a product relatedness measure:

$$R_{i,j} = \log\left\{ nN_{i,j}(\mathcal{D}) / \big(N_i(\mathcal{D}) N_j(\mathcal{D})\big) \right\}. \tag{6}$$

Taking the gradient of $\ell(\mathcal{D})$ with respect to $\mathbf{z}_i$ (the role of $\mathbf{z}_i$ and $\tilde{\mathbf{z}}_i$ is symmetric so we may consider either one), we obtain:

$$\nabla_{\mathbf{z}_i} \ell(\mathcal{D}) = \tilde{Z} \text{diag}(\mathbf{w}_i) \underbrace{\left\{ \sigma\big([R_{i,1}, \ldots, R_{i,|\mathcal{I}|}]\big) - \sigma(\langle \mathbf{z}_i, \tilde{\mathbf{Z}} \rangle) \right\}}_{\text{error term}}, \tag{7}$$

where $\text{diag}(\mathbf{w}_i)$ is the diagonal weight matrix with each instance is by given $\mathbf{w}_{i,j} = p_{i,j}(\mathcal{D}) + k p_i(\mathcal{D}) p_j(\mathcal{D})$. Bearing (7) in mind, let us revisit the weighted least square matrix factorization for $\left[R_{i,j}\right]_{i,j=1}^{|\mathcal{I}|}$ where the weights are also given by $\mathbf{w}_{i,j}$: $\ell_{\text{ls}}(\mathcal{D}) = \sum_{i,j} \mathbf{w}_{i,j} \big(R_{i,j} - \mathbf{z}_i^\intercal \tilde{\mathbf{z}}_j\big)^2$. The gradient with respect to $\mathbf{z}_i$ is given by:

$$\nabla_{\mathbf{z}_i} \ell_{\text{ls}}(\mathcal{D}) = \tilde{Z} \text{diag}(\mathbf{w}_i) \underbrace{\left\{ [R_{i,1}, \ldots, R_{i,|\mathcal{I}|}] - \langle \mathbf{z}_i, \tilde{\mathbf{Z}} \rangle \right\}}_{\text{error term}}. \tag{8}$$

Comparing (8) with (7), we see that the only difference lies in the error term which measures the deviation of projecting $\mathbf{z}_i$ to the space of $\tilde{\mathbf{Z}}$ and the target vector $[R_{i,1}, \ldots, R_{i,|\mathcal{I}|}]$:

- for the SGNS gradient in (7), the error is measured on the nonlinear space induced by the $\sigma(\cdot)$ transformation;
- for the lest-square objective in (8), the error is measured on the regular Euclidean space.

By comparing with the least-square matrix factorization, which characterizes linear project by the definition, we see that project embeddings trained by SGNS algorithm are essentially non-linear projections of the product relatedness matrix $\left[R_{i,j}\right]_{i,j=1}^{|\mathcal{I}|}$.

Remark 1 (Product relatedness measure and point-wise mutual information). *The point-wise mutual information (PMI) is defined by* $\log\left\{p(i,j)/p(i)p(j)\right\}$*, which is a data-specific measurement on how much information about entity $j$ is in entity $i$. Notice that the product relatedness measure is a random variable while the PMI is a fixed quantity. Upon a realization of the data generating mechanism $\mathcal{D}$, the $\text{PMI}_{i,j}$ computed by the observed data is an estimation of the the product relatedness measure $R_{i,j}$.*

Upon realizing that product embeddings are nonlinear projections, the next question is in what sense is the particular nonlinear projection optimal. Unlike linear projection whose optimality in the $\ell_2$ norm is well understood, there is no rule-of-thumb method to analyze nonlinear projections. However, our exploration from the SDR perspective is not by mere guessing. The key intuition is from the result in [16], that if there are no constraints on the embedding dimension, then the minimizer of the SGNS objective is exactly given by the corresponding product relatedness measure $\left[R_{i,j}\right]_{i,j=1}^{|\mathcal{I}|}$: $\big[R_{i,1}, \ldots, R_{i,|\mathcal{I}|}\big] = \arg\min_{\mathbf{z}_i} \ell(\mathcal{D})$, $i = 1, \ldots, |\mathcal{I}|$.

With a pre-specified dimension $d < |\mathcal{I}|$, the objective becomes: $\arg\min_{\mathbf{z}_i \in \mathbb{R}^d} \ell(\mathcal{D})$, which constraints the solution to the convex subspace of $\mathbb{R}^d$. The convexity of the constraint space often leads to nice relations between the unconstrained optimum and constrained optimum, e.g. the maximum-likelihood estimation leads to the locally optimal instance (in the constraint model space) in terms of the *KL divergence*. The SGNS objective, with scrutiny, is also maximizing a particular likelihood function. We formalize the above intuition in the following claim.

Claim 1. *Let $q\big(O \,\big|\, \mathcal{D}; \mathbf{Z}, \tilde{\mathbf{Z}}\big)$ be the co-occurrence probability computed by the embedding as in (1). At global optimum, the embedding matrices are given by the product relatedness matrix that gives the co-occurrence probability $p\big(O \,\big|\, \mathcal{D}; R\big)$. The minimizer of the SGNS objective function is characterized by:*

$$\underset{\mathbf{Z},\tilde{\mathbf{Z}} \in \mathbb{R}^d}{\text{minimize}} \; D_{KL}\Big(q\big(O \,\big|\, \mathcal{D}; \mathbf{Z}, \tilde{\mathbf{Z}}\big) \,\Big\|\, p\big(O \,\big|\, \mathcal{D}; R\big)\Big). \tag{9}$$

*According to (4), the product embedding is the sufficient dimension reduction of product relatedness measure with respect to the co-occurrence probability.*

The proof is provided in the appendix. Much of the analysis in this section holds for the general embedding settings as well. However, researchers from other domains, e.g. the NLP community, do not take the same approach. We explicitly consider the uncertainty from data generating mechanism, which is necessary for product embeddings because the input data structures and definitions of neighborhood are very different under various problems. On the other hand, the NLP community treats the data as fixed and given by the corpus, and the definition of neighborhood often has less impact on the outcome. In contrast, our particular interests in this type of results are driven by the pursuit of model interpretability for e-commerce machine learning.

Recognizing the product embedding as nonlinear projection plus sufficient dimension reduction of the product relatedness measure for the co-occurrence probability provides understandings of their nature. It is also an essential step towards analyzing their domain-specific theoretical properties, i.e. their meaningfulness and usefulness in the e-commerce setting, which are the topics of the next two sections.

## 4 PROPERTIES OF THE PRODUCT RELATEDNESS MEASURE

As a result of Claim 1, the understandings of the product relatedness measure can be transferred to the product embedding since it is the information-theoretical optimal compression in the $\mathbb{R}^d$ space. We explore two types of domain-specific properties for the product relatedness measure. The first property is concerned with the false association problem of product relation, which leads to a practical data cleaning procedure that reduces the noise in product embeddings. The second property relates to the *higher-order relation* and the *functional relation* perspective that is also unique to the e-commerce setting.

We often overlook the problem of false product association in e-commerce. Popular items are likely to co-occur with a large number of irrelevant items, so removing them from the dataset has become a common practice before model training. However, apart from the few globally popular items, false associations are also incurred by various factors including random user behavior, causing the effectiveness of standard data cleaning processes are sensitive to the underlying data distribution. The SGNS embedding algorithm and its variants are vulnerable to misspecified associations because they treat each co-occurrence with equal importance. The false association problem for the embedding model has not been studied before, mainly because it rarely raises concerns in the NLP setting. We explain the domain difference in two folds.

- The training data for NLP are extracted from established documents with high reliability. In e-commerce, the training data are often collected from user feedback, so the quality is often uncontrolled and the signal-to-noise ratio is not ideal.
- The number of tokenized words in an NLP training corpus if often of several magnitudes smaller than the number of products in an e-commerce dataset, so their frequency distribution can be different (Figure 1c)[2].

In analogy to the notation of false positive from hypothesis testing, we define false association for the product relatedness measure $R_{i,j} = \log\{nN_{i,j}(\mathcal{D})/(N_i(\mathcal{D})N_j(\mathcal{D}))\}$.

*Definition 1.* A false association of the product relatedness measure is to observe a large value of $R_{i,j}$ by chance.

First notice that the lower bound of $R_{i,j}$ is $-\infty$ so it can take negative values. If product $i$ and $j$ are not related, i.e. $N_i(\mathcal{D})$ is independent of $N_j(\mathcal{D})$, then we can expect $n\mathbb{E}[N_{i,j}(\mathcal{D})] = \mathbb{E}[N_i(\mathcal{D})] \cdot \mathbb{E}[N_j(\mathcal{D})]$, which implies that $\mathbb{E}[R_{i,j}] = 0$. In theory, we expect the unrelated products to have an ideal zero relatedness measure. However, there are random perturbations when the number of samples is insufficient (which is common in e-commerce dataset). The distributions of $R_{i,j}$ on real-world datasets are provided in Figure 1d, where a proportion of the values are negative. So the question is, how do we characterize our level of confidence when observing a relatively small value of $R_{i,j}$ so that we know it is safe to include the co-occurrence of $(i, j)$ to the training data?

The asymptotic properties of $R_{i,j}$ can be misleading in this case since the sample size is usually limited. The finite-sample tail bound, which characterizes the deviations from mean, appears to a reasonable choice; however, we need to be cautious because there are caveats in choosing the Hoeffding-type tail bound which tends to be loose on the boundary of $[0, 1]$ (see the Appendix for more detail). A careful manipulation using the Chernoff technique leads to a tighter tail bound that can be applied to derive confidence sets.

Lemma 1. *Define* $D_{KL}(a\|b) = p\log p/q + (1-p)\log(1-p)/(1-q)$ *for* $a, b \in (0, 1)$. *Then for* $\forall\epsilon > 0$*:*

$$\mathbb{P}\big(R_{i,j} \le -\epsilon\big) \le \exp\Big\{-nD_{KL}\Big(\frac{\mathbb{E}[N_i(\mathcal{D})]\mathbb{E}[N_j(\mathcal{D})]}{n^2e^{-\epsilon}} \,\Big\|\, \frac{\mathbb{E}[N_i(\mathcal{D})]\mathbb{E}[N_j(\mathcal{D})]}{n^2}\Big)\Big\}. \tag{10}$$

[2]We use the National Library of Medicine MeSH (Medical Subject Heading) as an example: https://www.dropbox.com/s/sd4yj1uqsqak4n1/d2016.bin?dl=1.

*The finite-sample lower confidence bound at $\alpha$-level, i.e. $p\big(R_{i.j} \le 0\big) \le \alpha$, is then given by:*

$$\log\frac{n^2p_\alpha}{\mathbb{E}[N_i(\mathcal{D})]\mathbb{E}[N_j(\mathcal{D})]}, \text{ where}$$
$$p_\alpha = \max\Big\{p \in [0,1] : D_{KL}\Big(\frac{N_{i,j}(\mathcal{D})]}{n}\,\Big\|\,p\Big) \le \frac{\log 1/\alpha}{n}\Big\}. \tag{11}$$

The proof is mostly technical and we leave it to the appendix. The result in Lemma 1 provides a sound criterion for detecting the truly unrelated product pairs. Compare with the other heuristic methods such as removing the top popular products, the confidence-interval approach is theoretically-grounded and is consistent across data distributions. While removing the noisy product pairs reduces false association and improves the quality of embedding, the approach requires the extra step of storing the observed $N_{i,j}(\mathcal{D})$ and conducting pairwise comparisons, so the computation and memory complexity will depend on the overall sparsity of the data.

In e-commerce, it is common that complementary products are combined into a combo (bundle) that possesses the overall functionality and contextual meaning of each product. The relations between the bundle and other products are often unchanged by the composition. For instance, `toothbrush` and `toothpaste`, are often bundled together, and the `brush+paste` bundle may inherit their relations with the other personal care products. This type of higher-order relation is essential and unique to e-commerce and should be captured by the product embedding whenever possible. In NLP, for example, when "`straw`" and "`berry`" are bundled together, the word "`strawberry`" has a different contextual meaning.

We find out that the product relatedness measure $R_{i,j}$ exactly constructs the higher-order relation, so in some sense the product embedding is indeed recognizing and leveraging the higher-order relation to characterizing product relations. Before we present the main discovery, we provide a heuristic definition for the higher-order relation using $R_{i,j}$.

*Definition 2.* Let $I$ be the random variable for products. Given a set of products $\{i_1, \ldots, i_k\}$, if there exists a product $i^* \in \mathcal{I}$ such that:

$$\mathbb{E}_{I|\{i_1,\ldots,i_k\}}\big[R_{i^*,I} - R_{j,I}\big] \ge 0, \tag{12}$$

for any other product $j \in \mathcal{I}$, then $i^*$ is the higher-order representation of $\{i_1, \ldots, i_k\}$.

We first provide heuristic understandings for the definition. For example, the `brush+paste` combo (assuming it exists as a valid product in $\mathcal{I}$) could be a higher-order representation of the product set {`brush`, `paste`}. The expectation term in (A.3) simply implies that compared with all other choices, the brush+paste combo, on average, has higher relatedness with all the products that may co-occur with {brush, `paste`} combined:

$$\sum_{\text{item}} p(\text{item} \mid \{\text{brush, paste}\})[R_{\text{brush+paste,item}} - R_{\text{other choice,item}}] > 0.$$

If the definition does lead to an $i^*$ that recovers the relation between $\{i_1, \ldots, i_k\}$ and other items, then the product relatedness measure is indeed capturing the higher-order relation among products. We show that there is an one-to-one mapping between (A.3) and an optimal information criterion that evaluates the distance of the conditional distributions induced by $i^*$ and $\{i_1, \ldots, i_k\}$.

Claim 2. *Let $i^*$ be defined by (A.3) for a meaningful product set $\{i_1, \ldots, i_k\}$ such that $\exists j \in \mathcal{I}: p(j|\{i_1, \ldots, i_k\}) > 0$. Then the higher-order product relation can be constructed using the distribution induced by $i^*$ and $\{i_1, \ldots, i_k\}$:*

$$i^* = \arg\min_{i \in \mathcal{I}} D_{KL}\Big(p\big(1\big[i \in \mathcal{N}(\{i_1, \ldots, i_k\})\big]\big) \,\Big\|\, p\big(1[i \in \mathcal{N}(i^*)]\big)\Big), \tag{13}$$

The proof is provided in the appendix. We also show that the direction from (13) to (A.3) also holds true, so there exists a duality between using KL-divergence and product relatedness measure to recognize and construct the higher-order product relations.

Another interesting property we investigate is inspired by the famous finding from NLP that $\mathbf{z}_{\texttt{king}} - \mathbf{z}_{\texttt{men}} = \mathbf{z}_{\texttt{queen}} - \mathbf{z}_{\texttt{women}}$. The simple analogy from NLP is not useful in e-commerce for obvious reasons, however, when a group of functionally-related product pairs are presented, we ask the question of whether their product embeddings can be combined to obtain a meaningful representation for their functional relation. A motivating example will be: (`TV`, `remote control`), (`XBox`, `handle`) and (`laptop`, `mouse`), which all reflect the complementary relation in electronics. So is it possible that $(\mathbf{z}_{\texttt{TV}} - \mathbf{z}_{\texttt{remote control}}) + (\mathbf{z}_{\texttt{XBox}} - \mathbf{z}_{\texttt{handle}}) + (\mathbf{z}_{\texttt{laptop}} - \mathbf{z}_{\texttt{mouse}})$ captures the functional representation of the complementary relation in electronics?

We formalize the setup by denoting a relation $r$ by $\xrightarrow{r}$, e.g. handle $\xrightarrow{\text{complement}}$ XBox. Again, we focus on the product relatedness measure $R_{i,j}$, where we use the shorthand $\vec{R}_i = [R_{i,1}, \cdots, R_{i,\mathcal{I}}]$. Following the previous discussion, for a group of product pairs satisfying $\xrightarrow{r}$: $\mathcal{D}_r \equiv \big\{(i,j)|i \xrightarrow{r} j\big\}$, we define $\mathbf{z}_r = \sum_{i \xrightarrow{r} j} \vec{R}_j - \vec{R}_i$. The following claim validates $\mathbf{z}_r$ as the representation for relation $r$ according to $\mathcal{D}_r$.

Claim 3 (Informal). *Let $\mathcal{D}_r$ and $\mathbf{z}_r$ be defined above. For a product pair $i^* \xrightarrow{r} j^*$ not included in $\mathcal{D}_r$, we have:*

$$\vec{R}_{j^*} = \vec{R}_{i^*} + \mathbf{z}_r + \epsilon, \tag{14}$$

*where $\epsilon$ is the residual term that is negligible under mild conditions.*

The details are relegated to the appendix. Claim 3 characterizes the generalization property of the functional relation captured by the product relatedness measures. In summary, the product relatedness measure is capable of capturing two of the essential e-commerce-specific relations: the higher-order relation and the functional relation among products, without relying on additional context or supervision.

Nevertheless, how do the-above explorations on product relatedness measure lead to practical usage other than providing model interpretation? We have shown that the product embeddings are nonlinear projections of the product relatedness measure $R$ so we can not expect (A.3) and (14) to hold for product embedding as well. However, recall that product embedding is also the sufficient dimension reduction for $R$, so $\mathbf{z}_i$ is the best compression of $\vec{R}_i$ in the $\mathbb{R}^d$ space. It remains challenging to develop the optimal strategy that best leverages the higher-order relation and functional relation information in the product embedding space. However, we may still proceed with more straightforward (perhaps sub-optimal) practices such as adding the product embeddings in a shopping cart as the cart's representation, and conduct clustering based on the product embedding differences to detect and establish functional relations.

# 5 THE GENERALIZATION BOUND OF PRODUCT EMBEDDINGS

Even though we have recognized product embedding as the sufficient dimension reduction of the product relatedness measure, which suggests certain optimally, the performance of product embedding in downstream tasks can still depend on the problem instance, e.g. the embedding dimension $d$. Intuitively speaking, a small $d$ is insufficient for compression, while an overlarge $d$ can introduce extra noise due to the random initialization in the SGNS algorithm. In this paper, we study the product-level downstream tasks, e.g. product classification using the product embedding. Tasks that take a set (sequence) of products as input often employ complicated models, e.g. recurrent neural network, whose generalization performance can be intractable. Notice that the entity-level tasks are also unique to the e-commerce world, since in NLP, the downstream tasks are for the sentence or document level.

Before we characterize the generalization performance of product embedding, we point out that our problem is fundamentally different from the ordinary generalization theory in supervised learning:

- For the ordinary supervised learning, we study the generalization error from $\mathbf{X}_{\text{train}}$ to $\mathbf{X}_{\text{all}}$, i.e. how the optimal model for the training data performs globally [26].
- In our problem, we study how the model trained with the embedding $\mathbf{Z}$ generalizes against the original setting where the the model is trained using $\mathbf{X}$.

Specifically, the model using $\mathbf{Z}$ is trained by minimizing: $\mathcal{L}(\mathbf{Z}) = \mathbb{E}\Big[\frac{1}{|\mathcal{I}|}\sum_i \phi\big(f_{\theta_1}(\mathbf{z}_i), y_i\big)\Big]$, and the model using $\mathbf{X}$ is trained by minimizing: $\mathcal{L}(\mathbf{X}) = \mathbb{E}\Big[\frac{1}{|\mathcal{I}|}\sum_i \phi\big(f_{\theta_2}(\mathbf{x}_i), y_i\big)\Big]$, where $\phi(\cdot,\cdot)$ is the loss function that is L-Lipschitz in both arguments[3]. Here, $\mathbf{X}$ can be given by the product relatedness matrix $[R_{i,j}]_{i,j=1}^{|\mathcal{I}|}$, and we use the notation $\mathbf{X}$ to be consistent with the supervised learning literature. It is usually the case that people investigate on the linear model such that $\mathbf{y} = \mathbf{X}^\intercal\boldsymbol{\theta}_2 + \boldsymbol{\epsilon}$. However, this generic setting is not applicable to our problem because the data matrix $\mathbf{X}$, e.g. given by $[R_{i,j}]_{i,j=1}^{|\mathcal{I}|}$, has a very different geometric structure from product embedding $\mathbf{Z}$:

- empirical evidence shows that the elements of $\mathbf{Z}$ are mostly between $(-1, 1)$, while the elements of $\mathbf{X}$ can be unbounded;
- the product embedding lies in the Euclidean subspace of $\mathbb{R}^d$, where $\mathbf{X}$ can have arbitrary manifold data structure with a different dimension.

Therefore, we need a standardization protocol that works for both $\mathbf{Z}$ and $\mathbf{X}$, which leads us to the singular value decomposition:

$$\mathbf{Z} = \mathbf{U}(Z)\Sigma(Z)\mathbf{V}(Z)^\intercal, \text{ and } \mathbf{X} = \mathbf{U}(X)\Sigma(X)\mathbf{V}(X)^\intercal,$$

[3] In the case of binary classification, we do not explicitly assume $y_i$ is categorical. When $y_i$ is given by the score (of the positive class), we simply adapt the multi-class logistic loss, e.g. $\phi(f(x_i), y_i) = \sigma(y_i)\log\sigma(f(x_i)) + (1-\sigma(y_i))\log(1-\sigma(f(x_i)))$, where $\sigma(\cdot)$ is the sigmoid function. The loss function is Lipschitz if both $y_i$ and $f(x_i)$ are bounded, which is a mild assumption.

so $\mathbf{U}(Z)$ and $\mathbf{U}(X)$ are both orthonormal basis. The intuition is that we now think of $\mathbf{y}$ as generated by $\mathbf{U}(X)$ to avoid the above-mentioned issues. Then we consider the average-case setting (where the parameters $\theta$ follow an unknown distribution $N(0, \Sigma)$) with a proper scaling $\|\Sigma\| \leq 1$:

$$\mathbf{y}_0 = \mathbf{U}(X)^\intercal \boldsymbol{\theta}_0, \;\; \mathbf{y} = \mathbf{y}_0 + \boldsymbol{\epsilon} \;\text{ for } \boldsymbol{\theta}_0 \sim N(0, \Sigma), \boldsymbol{\epsilon} \sim N(0, \sigma^2 \mathbf{I}). \quad (15)$$

As such, we are interested in the average-case loss:

$$\mathbb{E}_{\theta_0, \epsilon}[\mathcal{L}(\mathbf{Z})] = \mathbb{E}_{\theta_0, \epsilon}\Big[\sum_i \phi\big(\hat{f}(\mathbf{z}_i), \mathbf{y}_{0,i}\big)\Big].$$

It becomes clear at this point that we study the random-effect setting because the observations are no longer independent, since we assume that they are generated by $\mathbf{U}(X)$ instead of $\mathbf{X}$. Therefore, in the above average-case loss, the expectation with respect to $\theta_0$ incurs because the "clean testing data" $y_0$ is generated under $\theta_0 \sim N(0, \Sigma)$. As for the $\epsilon$, it also occurs under the expectation because $\hat{f}(\cdot)$ is estimated using the "noisy training data" $\mathbf{y} = \mathbf{y}_0 + \boldsymbol{\epsilon}$ with a given $\mathbf{y}_0$, so $\hat{f}(\cdot)$ is a (implicit) function of $\boldsymbol{\epsilon}$.

Theorem 3. *Let $f_{\theta_1}(\mathbf{z}_i) = \mathbf{z}_i^\intercal \theta_1$ and $f_{\theta_2}(\mathbf{x}_i) = \mathbf{x}_i^\intercal \theta_2$. The generalization error for product embedding in the above setting follows:*

$$\mathbb{E}_{\theta_0, \epsilon}\big[\mathcal{L}(\mathbf{Z}) - \mathcal{L}(\mathbf{X})\big] \leq L\Big\{\Big(tr(\Sigma) - \underline{\lambda(\Sigma)}\big\|\mathbf{U}(X)^\intercal \mathbf{U}(Z)\big\|_F^2\Big)/|I|\Big\}^{1/2} + C, \quad (16)$$

*where $\underline{\lambda(\Sigma)}$ gives the smallest eigenvalue of $\Sigma$.*

The proof is relegated to the appendix. The significance of Theorem 3 is that the average-case generalization error of product embedding is controlled by the factor $\big\|\mathbf{U}(X)^\intercal \mathbf{U}(Z)\big\|_F^2$, i.e. how well the spectral space of $\mathbf{Z}$ aligns with the spectral space of $\mathbf{X}$. While the results reveal the special case under a linear model, it nevertheless provides a novel perspective for understanding the generalization performance of product embedding. We see that the "closeness" between $\mathbf{X}$ and $\mathbf{Z}$, which depends on the problem instance (loss function, generating model, learning model, etc.) as we show here, plays a critical part in the generalization bound. We leave it to the future work to derive the results for the more general models.

## 6 EXPERIMENTS AND RESULTS

We design our experiments to provide empirical supports for our theoretical results, as well as to shed insights for future research and application with product embedding. All the reported numerical results are computed from ten repetitions. We use $d = 32$ unless specified otherwise.

**Dataset.**

We use the public *Instacart* dataset[4] for reproducibility. As for the experiments where the resource and information in public dataset do not satisfy our need, we use the proprietary *Walmart.com* datasets which we have full access. The *Instacart* data consists of ~50,000 grocery products, with the shopping records of ~200 thousand users and 3 million orders. The product catalog information, i.e. the *category* and *department* tags, can be used as labels for downstream classification task. We experiment on two types of data-generating mechanism $\mathcal{D}(\mathcal{N})$:

- **Sequences**. Choosing the users' sequential impression (purchase) as input data structure, and the neighborhood is defined by using the five previous purchases;
- **Graphs**. We build the *undirected* weighted graph using session-based purchase data. The neighborhood is then obtained by sampling five nodes according to the random walk outcome like the *Node2vec* [13].

**Optimality of sufficient dimension reduction.**

Directly examining the optimally of product embedding, which is the sufficient dimension reduction as we show in Claim 1, is infeasible so we need to rely on certain tasks. We consider:

**Task 1**. The next-item recommendation, where we use all but the last item for training, the second-to-last item for validation (if needed), and the last item for testing. The inner product $\langle \mathbf{z}_{\text{last item}}, \mathbf{z}_{\text{next item}} \rangle$ is used to rank the candidate items;

**Task 2**. The items' *department* classification, where we simply employ the multi-class logistic regression: $\text{softmax}(\mathbf{z}_i^\intercal \Theta)$, as the classifier. The model is trained using the *Scikit-Learn* package.

We show that product embedding achieves better performance than using the least-square linear dimension reduction (8) of the same $R_{i,j}$, when the embedding dimension is also $d = 32$. When estimating $R_{i,j}$ from the data $\mathcal{D}(\mathcal{N})$, we treat $\hat{r}_{i,j}$ as missing value if $(i, j)$ never co-occurs, and we treat the negative estimated values as zero (because in theory, $R_{i,j} = 0$ if the two products are unrelated). The comparisons of the results are provided in Table 2. We see that product embedding outperforms the linear dimension reduction in both tasks, supporting the optimality of product embedding as sufficient dimension reduction.

| Data | Instacart | | Walmart.com | |
|---|---|---|---|---|
| | **recommendation** | | | |
| Metric | AUC | NDCG | Recall@10 | NDCG@10 |
| LDR(seq) | 0.939(.008) | 0.142(.004) | 0.112(.005) | 0.058(.002) |
| **Emb**(seq) | *0.954*(.005) | *0.160*(.004) | *0.155*(.004) | *0.079*(.002) |
| LDR(graph) | 0.929(.010) | 0.139(.006) | 0.107(.008) | 0.053(.004) |
| **Emb**(graph) | *0.948*(.008) | *0.155*(.005) | *0.131*(.006) | *0.070*(.004) |
| | **classification** | | | |
| Metric | micro-F1 | macro-F1 | micro-F1 | macro-F1 |
| LDR(seq) | 0.483(.010) | 0.342(.013) | 0.312(.017) | 0.253(.013) |
| **Emb**(seq) | *0.509*(.008) | *0.470*(.010) | *0.395*(.018) | *0.313*(.011) |
| LDR(graph) | 0.487(.012) | 0.345(.019) | 0.316(.022) | 0.260(.017) |
| **Emb**(graph) | *0.513*(.010) | *0.476*(.014) | *0.404*(.018) | *0.317*(.014) |

**Table 2: Recommendation and classification results. *LDM* and *Emb* denote using linear dimension reduction and SGNS algorithm, and *seq* and *graph* indicates the data structure.**

**Improvement of removing false associations.**

According to the finite-sample confidence interval from Lemma 1, we do a full scan of the training data and remove the suspicious product co-occurrences with confidence level $\alpha \in \{0.3, 0.6, 0.9\}$ by treating them as zero. We then train the product embedding (using SGNS) and examine the performance via the same two tasks. The results are shown in Figure 2, where we see that a higher confidence level does lead to improvements for both tasks. The degree of improvement is more significant by moving from small to medium

[4] https://www.instacart.com/datasets/grocery-shopping-2017

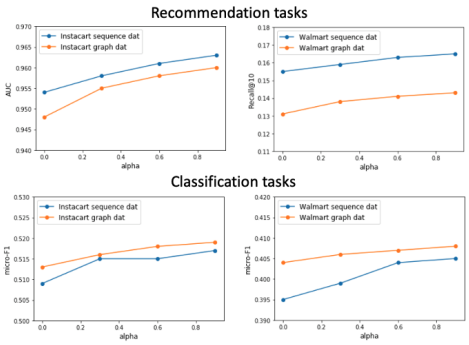


**Figure 2: The effectiveness of removing false associations according to the confidence bound in Lemma 1, for both the recommendation and classification task.**

$\alpha$. The improvement curves gets flattened under large $\alpha$, which can be caused by removing too much training data. The proposed preprocessing approach is overall effective, but the tradeoff between the quality and the size of data should be judged case-by-case.

**Cart-page recommendation: an example on exploring the higher-order relation.**

We obtain ~100,000 shopping cart snapshots with the customers' continual shopping records from *Walmart.com*. To examine the heuristic that by adding individual product embedding (**Add**) as the cart's embedding for the next-item (provided by the continual-shopping record) recommendation, we compare with:

**Baseline1**. Randomly select an item from the cart and use its embedding as the cart's embedding;

**Baseline2**. Use the most-recent added item's embedding as the cart's embedding;

**Oracle**. The best item embedding in retrospect, i.e. after we observe the user's next move and select the best item in the cart, as an ad-hoc approach;

**Enhanced**. Apply a simple dot-product self-attention to obtain the weights and use the weighted sum of the item embedding.

The cart-page recommendation performance is provided in Table 3. Compared with using a single product embedding, even when the single product is given by the oracle, combining the product embeddings leads to better performance under the simple addition. When the weight for each product is more carefully chosen, such as by using the dot-product attention mechanism, we observe a further improvement. The result is not surprising, since combining individual product embedding is becoming common in personalized recommendation. In this paper, we further justify this approach via the lens of higher-order product relations.

**Detecting product functional relations.**

Here, we provide a brief demonstration on some interesting results we obtained by clustering the embedding difference of product pairs, i.e. $\mathbf{z}_{\text{anchor}} - \mathbf{z}_{\text{reco}}$, for the 1,000 most popular anchor items with their top-10 recommendation (obtained by using the inner products of product embedding). All the items are selected from the *electronics* catalog on *Walmart.com*. We conduct both the K-means clustering and hierarchical clustering, with results shown in Figure 3. Under the correctly-specified number of clusters, K-means exactly detects the different functional relations for each department

| | Baseline1 | Baseline2 | Oracle | Add | Enhanced |
|---|---|---|---|---|---|
| Recall @10 | .051(.011) | .064(.003) | .093(.002) | .101(.002) | .112(.003) |
| NDCG @10 | .027(.004) | .030(.001) | .038(.001) | .044(.001) | .046(.001) |

| dimension | d=8 | d=16 | d=32 | d=64 |
|---|---|---|---|---|
| $S(Z,X)$ | 0.043 | 0.105 | 0.144 | 0.212 |
| micro-F1 | 0.206 | 0.358 | 0.404 | 0.439 |

**Table 3: Upper: the cart-page recommendation performances; Lower: the experiments for the generalization of embedding in classification task. Both experiments are conducted on the *Walmart.com* data.**

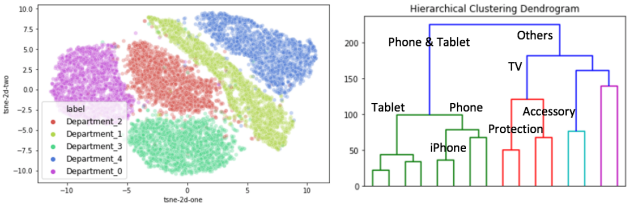


**Figure 3: Left: the clustering results (visualized after a t-sne projection to 2D) that detects the different product functional relations under each product department; Right: hierarchical clustering result (the branches are labelled after cross-checking with the ground-truth item catalog).**

of electronic products. To make sense of the hierarchical clustering result, we do a manual cross-checking to label the different branches from the dendrogram, leveraging the true department tag for the items. We find that hierarchical clustering keeps showing refined detection of finer-granulated product functional relations. Our discovery supports the result in Claim 3, and provide insights for understanding product relations via the pairwise embedding differences.

**Generalization performance of downstream tasks.**

To support our generalization results, we vary the embedding dimension $d \in \{8, 16, 32, 64\}$ as a control factor to obtain product embeddings that give different spectral *alignment score* $S(Z,X) := \left\|\mathbf{U}(X)^{\intercal}\mathbf{U}(Z)\right\|_F^2$, where the data matrix $X$ is given by the estimated $[R_{i,j}]$ matrix. Here, we specifically study the item classification task. As we conjectured, a larger dimension does lead to a higher score within the range we consider, where the results are provided in Table 3. We see that a higher spectral alignment score leads to better downstream classification performance, where the classifier is logistic-regression so the empirical result is consistent with our theoretical justifications. Our discussion may lead to methods that practically chooses $d$ in a data-adaptive fashion (which is out of the scope of this paper).

## 7 CONCLUSION

We thoroughly study the theoretical backgrounds of product embeddings by answering what they are, how they are unique to e-commerce, and why they are useful in downstream tasks. With both the technical derivations and intuitive explanations, we hope

this paper provides tools and reference for model interpretation and understanding, as well as developing more advanced techniques for representation learning in e-commerce.

## REFERENCES


[1] Carl Allen, Ivana Balazevic, and Timothy Hospedales. 2019. What the vec? towards probabilistically grounded embeddings. In *Advances in Neural Information Processing Systems*. 7467–7477.

[2] Carl Allen and Timothy Hospedales. 2019. Analogies explained: Towards understanding word embeddings. *arXiv preprint arXiv:1901.09813* (2019).

[3] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics* 4 (2016), 385–399.

[4] Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[6] Ryan Cotterell, Adam Poliak, Benjamin Van Durme, and Jason Eisner. 2017. Explaining and generalizing skip-gram through exponential family principal component analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. 175–181.

[7] 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.

[8] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 101–109.

[9] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 135–144.

[10] Amir Globerson and Naftali Tishby. 2003. Sufficient dimensionality reduction. *Journal of Machine Learning Research* 3, Mar (2003), 1307–1331.

[11] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).

[12] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1809–1818.

[13] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[14] David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. A closer look at skip-gram modelling.. In *LREC*, Vol. 6. 1222–1225.

[15] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based Retrieval in Facebook Search. *arXiv preprint arXiv:2006.11632* (2020).

[16] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*. 2177–2185.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[19] Kyösti Pennanen, Taina Kaapu, and Minna-Kristiina Paakki. 2006. Trust, risk, privacy, and security in ecommerce. In *Proceedings of the ICEB+ eBRF Conference*.

[20] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[21] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. *arXiv preprint arXiv:2005.09683* (2020).

[22] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[23] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. Springer, 291–324.

[24] Karl Stratos, Michael Collins, and Daniel Hsu. 2015. Model-based word embeddings from decompositions of count matrices. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1282–1291.

[25] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 225–232.

[26] Martin J Wainwright. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge University Press.

[27] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. 2018. Representing and recommending shopping baskets with complementarity, compatibility and loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1133–1142.

[28] 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.

[29] Da Xu, Chuanwei Ruan, Jason Cho, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Knowledge-aware Complementary Product Representation Learning. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 681–689.

[30] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Product knowledge graph embedding for e-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 672–680.

[31] Rashad Yazdanifard, Noor Al-Huda Edres, and Arash Pour Seyedi. 2011. Security and privacy issues as a potential risk for further ecommerce development. In *International Conference on Information Communication and Management-IPCSIT*, Vol. 16.

# A PROOFS

We provide the proofs in this part of the paper.

## A.1 Proof for Claim 1

.

Proof. We start by considering the co-occurrence random variable $O$ as following a Bernoulli distribution such that $p(O=1)=\beta$, where $\beta \in (0,1)$ characterizes the global probability of having a positive sample, which for the SGNS it is given by $1/k+1$, since for each positive sample we generate $k$ negative samples.

Consequently, we have $p(i,j|O) = \begin{cases} p_i(\mathcal{D})p_j(\mathcal{D}) & if \quad O=0, \\ p_{i,j}(\mathcal{D}) & if \quad O=1. \end{cases}$

Applying the Bayes rule, we immediately have:

$$p(O=1 \mid i,j) = \frac{\beta p_{i,j}(\mathcal{D})}{\beta p_{i,j}(\mathcal{D}) + (1-\beta)p_i(\mathcal{D})p_j(\mathcal{D})}. \tag{17}$$

We point out that this definition of co-occurrence probability does not contradict the original definitions in (1), since we have:

$$\frac{\beta p_{i,j}(\mathcal{D})}{\beta p_{i,j}(\mathcal{D}) + (1-\beta)p_i(\mathcal{D})p_j(\mathcal{D})} = \frac{1}{1+\exp\Big(-\log\big(\frac{\beta p_{i,j}(\mathcal{D})}{(1-\beta)p_i(\mathcal{D})p_j(\mathcal{D})}\big)\Big)} = \sigma(\tilde{R}_{i,j}),$$

where $\tilde{R}_{i,j} = R_{i,j} + \log\frac{\beta}{1-\beta}$ is a shifted version of the original product relatedness measure. The impact of using the shifted original product relatedness on the loss function is negligible, because we now have:

$$\ell(\mathcal{D}) = \sum_{i,j\in\mathcal{I}} N_{i,j}(\mathcal{D}) \log\sigma(\mathbf{z}_i^\intercal \tilde{\mathbf{z}}_j) + \frac{k}{n} N_i(\mathcal{D}) N_j(\mathcal{D}) \log\sigma(-\mathbf{z}_i^\intercal \tilde{\mathbf{z}}_j) + f(\beta(k)),$$

and $f(\beta(k)) := f(k/(1+k))$ is irrelevant to the optimization variables.

The first-order necessary condition for the global optimal of the SGNS objective without dimension constraint, which we denote by:

$$\mathbf{Z}^*, \tilde{\mathbf{Z}}^* = \arg\min_{\mathbf{Z},\tilde{\mathbf{Z}}} \ell(\mathcal{D}),$$

is given by $\nabla_{\mathbf{z}_i}\ell(\mathcal{D}) = 0$ and $\nabla_{\tilde{\mathbf{z}}_i}\ell(\mathcal{D}) = 0$, for $i = 1,\ldots,\|\mathcal{I}\|$. According to (7), the first-order condition implies that $\langle \mathbf{z}_i^*, \tilde{\mathbf{z}}_i^* \rangle = R_{i,j}$ for all $i,j \in \mathcal{I}$, because $\sigma(\cdot)$ is a strictly increasing function. Hence, $\ell(\mathcal{D};\mathbf{Z}^*,\tilde{\mathbf{Z}}^*) := \min_{\mathbf{Z},\tilde{\mathbf{Z}}} \ell(\mathcal{D})$ is a fixed quantity that only depends on $\mathcal{D}$. Therefore, minimizing $\ell(\mathcal{D})$ is equivalent to finding :

$$\arg\min_{\mathbf{Z},\tilde{\mathbf{Z}}\in\mathbb{R}^d} \big\{\ell(\mathcal{D}) - \min_{\mathbf{Z},\tilde{\mathbf{Z}}} \ell(\mathcal{D})\big\} = \arg\min_{\mathbf{Z},\tilde{\mathbf{Z}}\in\mathbb{R}^d} \big\{\ell(\mathcal{D}) - \ell(\mathcal{D};\mathbf{Z}^*,\tilde{\mathbf{Z}}^*)\big\}.$$

According to the above argument on the implication of the first-order condition, the term $\ell(\mathcal{D};\mathbf{Z}^*,\tilde{\mathbf{Z}}^*)$ is given by:

$$\sum_{i,j\in\mathcal{I}} N_{i,j}(\mathcal{D}) \log\sigma(R_{i,j}) + \frac{k}{n} N_i(\mathcal{D}) N_j(\mathcal{D}) \log\sigma(-R_{i,j}).$$

Recall that $q\big(O \,\big|\, \mathcal{D};\, \mathbf{Z},\tilde{\mathbf{Z}}\big)$ is the co-occurrence probability computed by the embedding as in (1), and $p\big(O \,\big|\, \mathcal{D};\, R\big)$ is the co-occurrence probability when the embedding matrices are given by the product relatedness matrix (that happens for the unconstrained global optimum). By rearranging terms and extracting the factor of $n(k+1)$ to the front, it holds that:

$$\begin{aligned}
&n(k+1)\big(\ell(\mathcal{D}) - \min_{\mathbf{Z},\tilde{\mathbf{Z}}} \ell(\mathcal{D})\big) \\
&\propto \sum_{i,j\in\mathcal{I}} \Big\{\frac{1}{k} p_{i,j}(\mathcal{D}) \log\frac{\sigma(R_{i,j})}{\sigma(\mathbf{z}_i^\intercal \mathbf{z}_j)} + (1-\frac{1}{k}) p_i(\mathcal{D}) p_j(\mathcal{D}) \log\frac{\sigma(-R_{i,j})}{\sigma(-\mathbf{z}_i^\intercal \mathbf{z}_j)}\Big\} \\
&= \sum_{i,j\in\mathcal{I}} \Big\{\frac{1}{k} p_{i,j}(\mathcal{D}) \log\frac{p\big(O_{i,j}=1 \mid R\big)}{q\big(O_{i,j}=1 \mid \mathbf{z}_i,\tilde{\mathbf{z}}_j\big)} + (1-\frac{1}{k}) p_i(\mathcal{D}) p_j(\mathcal{D}) \log\frac{p\big(O_{i,j}=0 \mid R\big)}{q\big(O_{i,j}=0 \mid \mathbf{z}_i,\tilde{\mathbf{z}}_j\big)}\Big\} \\
&\overset{(a)}{=} \sum_{\substack{\alpha\in\{0,1\} \\ (i,j)\in\mathcal{D}}} \Big\{p\big(O_{i,j}=\alpha\big) \log\frac{p\big(O_{i,j}=\alpha \mid R\big)}{q\big(O_{i,j}=\alpha \mid \mathbf{z}_i,\tilde{\mathbf{z}}_j\big)}\Big\} \\
&= D_{KL}\Big(q\big(O \,\big|\, \mathcal{D};\, \mathbf{Z},\tilde{\mathbf{Z}}\big) \,\Big\|\, p\big(O \,\big|\, \mathcal{D};\, R\big)\Big),
\end{aligned} \tag{18}$$

where we use the shorthand $O_{i,j} = \alpha$ to denote the event $O=\alpha, i, j$; and step $(a)$ follows from (17).

Therefore, solving for $\min_{\mathbf{Z},\tilde{\mathbf{Z}}\in\mathbb{R}^d} \ell(\mathcal{D})$ is equivalent to finding:

$$\arg\min_{\mathbf{Z},\tilde{\mathbf{Z}}\in\mathbb{R}^d} D_{KL}\Big(q(O\,|\,\mathcal{D};\,\mathbf{Z},\tilde{\mathbf{Z}})\,\Big\|\,p(O\,|\,\mathcal{D};\,R)\Big),$$

which concludes the proof. □

## A.2 Proof for Lemma 1

Proof. We first prove the auxiliary case that with $X_1,\ldots,X_n$ being a sequence of independent Bernoulli random variables under mean $\mu$, and $\hat{\mu}$ given by: $\frac{1}{n}\sum_{i=1}^n X_i$, it holds for any $\epsilon\in[0,1-\mu]$ that:

$$\mathbb{P}\big(\hat{\mu}\le\mu-\epsilon\big)\le\exp\big(-nD_{KL}(\mu-\epsilon\,\|\,\mu)\big).$$

The above result is straightforward by using the Cramer-Chernoff bounding technique:

$$\begin{aligned}\mathbb{P}\big(\hat{\mu}\le\mu-\epsilon\big)&\le\mathbb{P}\Big(\exp\Big(\lambda\sum_{i=1}^n(\mu-X_t)\Big)\ge\exp(\lambda n\epsilon)\Big)\\&\le\frac{\mathbb{E}\big[\exp\big(\lambda\sum_{i=1}^n(\mu-X_t)\big]}{\exp(\lambda n\epsilon)}\\&=\big\{\mu\exp(\lambda(1-\mu-\epsilon))+(1-\mu)\exp(\lambda(\mu+\epsilon))\big\}^n,\end{aligned}$$

holds for any $\lambda>0$. Note that the above objective is convex in $\lambda$ (when $\lambda>0$), so the unique minimizer is given by: $\lambda^*=\log\frac{(\mu+\epsilon)(1-\mu)}{\mu(1-\mu-\epsilon)}$. We plug $\lambda^*$ back to the above expression and obtain: $\mathbb{P}\big(\hat{\mu}\le\mu-\epsilon\big)\le\exp\big(-nD_{KL}(\mu-\epsilon\,\|\,\mu)\big)$.

We now let $\mu_{i,j}=\frac{\mathbb{E}N_i(\mathcal{D})N_j(\mathcal{D})}{n^2}$ and $\hat{\mu}=\frac{N_i(\mathcal{D})N_j(\mathcal{D})}{n}$, which are the mean and empirical average of the Bernoulli random variables defined in our setting. It then holds that:

$$\begin{aligned}&\mathbb{P}\big(\hat{\mu}_{i,j}\le\mu_{i,j}-\epsilon\big)\le\exp\Big(-nD_{KL}\Big(\frac{\mathbb{E}N_i(\mathcal{D})N_j(\mathcal{D})}{n^2}-\epsilon\,\Big\|\,\frac{\mathbb{E}N_i(\mathcal{D})N_j(\mathcal{D})}{n^2}\Big)\Big)\\&\Leftrightarrow\mathbb{P}\big(\log\frac{\hat{\mu}_{i,j}}{\mu_{i,j}}\le\log\big(1-\frac{\epsilon}{\mu_{i,j}}\big)\big)\le\exp\big(-nD_{KL}(\mu_{i,j}-\epsilon\,\big\|\,\mu_{i,j})\big),\quad\text{define }\tilde{\epsilon}:\epsilon=\mu_{i,j}(1+\exp(\tilde{\epsilon}))\\&\Leftrightarrow\mathbb{P}\big(\log\frac{\hat{\mu}_{i,j}}{\mu_{i,j}}\le\tilde{\epsilon}\big)\le\exp\Big(-nD_{KL}\Big(\frac{\mu_{i,j}}{\exp(-\tilde{\epsilon})}\,\Big\|\,\mu_{i,j}\Big)\Big).\end{aligned}$$

Then notice that $D_{KL}(\cdot,\|\,\mu)$ is decreasing on $[0,\mu]$, so if $\epsilon$ is the (unique) solution of $D_{KL}(\mu-\epsilon\,\|\,\mu)=\alpha$ on $[0,\mu]$ for some $\alpha\in[0,D_{KL}(0\,\|\,\mu)]$, it holds that:

$$\big\{D_{KL}(\hat{\mu}\,\|\,\mu)\ge\alpha,\hat{\mu}\le\mu\big\}=\{\hat{\mu}\le\mu-\epsilon,\hat{\mu}\le\mu\}=\{\hat{\mu}\le\mu-\epsilon\}.$$

Consequently, using the result from the beginning of the proof, we have:

$$\mathbb{P}\big(D_{KL}(\hat{\mu}\,\|\,\mu)\ge\alpha,\hat{\mu}\le\mu\big)\le\exp(-n\alpha).$$

Next, we take $\tilde{p}_\alpha=\max\{\mu\in[0,1]:D_{KL}(\hat{\mu}\,\|\,\mu)\le\alpha\}$, and it is straightforward to verify that: $\tilde{p}_\alpha\ge\hat{\mu}$ and $D_{KL}(\hat{\mu}\,\|\,\cdot)$ is strictly increasing on $[\hat{\mu},1]$. Therefore, it holds that:

$$\{\mu\ge\tilde{p}_\alpha\}=\{\mu\ge\tilde{p}_\alpha,\mu\ge\hat{\mu}\}=\{D_{KL}(\hat{\mu}\,\|\,\mu)\ge D_{KL}(\hat{\mu}\,\|\,\tilde{p}_\alpha),\mu\ge\hat{\mu}\}=\{D_{KL}(\hat{\mu}\,\|\,\mu)\ge\alpha,\mu\ge\hat{\mu}\},$$

which directly leads to: $\mathbb{P}(\mu\ge\tilde{p}_\alpha)\le\exp(-n\alpha)$.

Now we replace $\mu$ by $\mathbb{E}N_i(\mathcal{D})\mathbb{E}N_i(\mathcal{D})/n^2$ and define $\tilde{\alpha}=\exp(-n\alpha)$, which gives the desired result in (11).

□

## A.3 Proof for Claim 2

Proof. Recall from Definition 2 that $I$ is the random variable for products. Given a set of products $\{i_1,\ldots,i_k\}$, if there exists a product $i^*\in\mathcal{I}$ such that:

$$\mathbb{E}_{I|\{i_1,\ldots,i_k\}}\big[R_{i^*,I}-R_{j,I}\big]\ge0,$$

for any other product $j\in\mathcal{I}$, then $i^*$ is the higher-order representation of $\{i_1,\ldots,i_k\}$.

Define shorthand $\vec{I}=\{i_1,\ldots,i_k\}$ as the combo of the $k$ items, and $p(i|j):=p\big(\mathbb{1}[j\in\mathcal{N}(i)]\big)$. Similarly, we use the shorthand: $p(\vec{I}):=p(i_1,\ldots,i_k)$ and $p(i|j)=p(\mathbb{1}[i\in\mathcal{N}(j)])$.

By rearranging terms and apply basic algebraic manipulations, it holds that:

$$
\begin{aligned}
D_{KL}\Big(p\big(\mathbb{1}\big[i \in \mathcal{N}(\{i_1,\ldots,i_k\})\big]\big) \,\Big\|\, p\big(\mathbb{1}\big[i \in \mathcal{N}(i^*)\big]\big)\Big) &= \sum_{e\in\mathcal{I}} p(e|\vec{I}) \log \frac{p(e|\vec{I})}{p(e|i^*)} \\
&= \sum_{e\in\mathcal{I}} p(e|\vec{I})\Big(\log \frac{p(\vec{I})}{\prod_{i\in\vec{I}} p(i)} - \log \frac{p(\vec{I}|e)}{\prod_{i\in\vec{I}} p(i|e)} + \log \frac{p(i^*|e)}{p(i^*)} - \log \prod_{i\in\vec{I}} \frac{p(i|e)}{p(i)}\Big), \\
&= \sum_{e\in\mathcal{I}} p(e|\vec{I})\Big(\log \frac{p(\vec{I})}{\prod_{j\in\vec{I}} p(j)} - \log \frac{p(\vec{I}|e)}{\prod_{j\in\mathbf{x}} p(j|e)} + \mathbf{R}_{i^*,e} - \sum_{i\in\vec{I}} \mathbf{R}_{i,e}\Big) \\
&= \mathbb{E}_{e|\vec{I}}\Big[\log \frac{p(\vec{I})}{\prod_{j\in\vec{I}} p(j)} - \log \frac{p(\vec{I}|e)}{\prod_{j\in\vec{I}} p(j|e)} - \sum_{i\in\vec{I}} \mathbf{R}_{i,e}\Big] + \mathbb{E}_{e|\vec{I}}\big[\mathbf{R}_{i^*,e}\big].
\end{aligned}
$$

Notice that the first term in the above expression is independent of $i^*$, and as a consequence, when:

$$
D_{KL}\big(p(I|\{i_1,\ldots,i_k\}) \,\big\|\, p(I|i^*)\big) \le D_{KL}\big(p(j|\{i_1,\ldots,i_k\}) \,\big\|\, p(I|j)\big)
$$

it must hold that: $\mathbb{E}_{I|\{i_1,\ldots,i_k\}}\big[\mathbf{R}_{i^*,I} - \mathbf{R}_{j,I}\big] \ge 0, \forall j \in \mathcal{I}$, which exactly recovers (A.3).

It is easy to see that each step in the above derivation is invertible, so we can also obtain the statement in the claim by starting from Definition 2. Together, they give the desired results. □

## A.4 Proof for Claim 3

Proof. We follow the setup from A.3, and define the following shorthand:

$$
\tau(\vec{I}) = \log \frac{p(\vec{I})}{\prod_{i\in\vec{I}} p(i)} \text{ and } \tau(\vec{I}|e) = \log \frac{p(\vec{I}|e)}{\prod_{i\in\vec{I}} p(i|e)}.
$$

Notice that $\tau(\vec{I})$ is measuing the mutual independency among $\vec{I}$, and $\tau(\vec{I}|e)$ measure the conditional independence of $\vec{I}|e$ for $e \in \mathcal{I}$. The mutual independency and conditional independence terms are usually very small in the e-commerce setting, and the degree to which this assumption is valid decides the quality of the approximation in the statement.

Recall that for a group of product pairs satisfying $\xrightarrow{r}$: $\mathcal{D}_r \equiv \big\{(i,j)|i \xrightarrow{r} j\big\}$, we define $\mathbf{z}_r = \sum_{i\xrightarrow{r}j} \vec{R}_j - \vec{R}_i$. We further decompose $\mathcal{D}_r$ into $\mathcal{D}_r^{(+)} \cup \mathcal{D}_r^{(-)}$, where $\mathcal{D}_r^{(+)} = \{i \,|\, \exists j \in \mathcal{I} \text{ s.t. } (i,j) \in \mathcal{D}_r\}$ and $\mathcal{D}_r^{(i)} = \{j \,|\, \exists i \in \mathcal{I} \text{ s.t. } (i,j) \in \mathcal{D}_r\}$. We make the decomposition because the product functional relations are often asymmetric, which means $i^* \xrightarrow{r} j^* \nLeftrightarrow j^* \xrightarrow{r} i^*$.

It holds for any $e \in \mathcal{I}$ that:

$$
\begin{aligned}
&\mathbf{R}_{i^*,e} - \mathbf{R}_{j^*,e} \\
&= \log \frac{p(e|i^*)}{p(e|j^*)} + \log \prod_{q^+\in\mathcal{D}_r^{(+)}} \frac{p(e|q^+)}{p(e|q^+)} + \log \prod_{q^-\in\mathcal{D}_r^{(-)}} \frac{p(e|q^-)}{p(e|q^-)} \\
&= \sum_{q^+\in\mathcal{D}_r^{(+)}} \log p(q^+|e) - \sum_{q^-\in\mathcal{D}_r^{(-)}} \log p(q^-|e) + \log \frac{\prod_{q^-\in\mathcal{D}_r^{(-)}\cup i^*} p(e|q^-)}{\prod_{q^+\in\mathcal{D}_r^{(+)}\cup j^*} p(e|q^+)} \qquad (19) \\
&= \sum_{q^+\in\mathcal{D}_r^{(+)}} \mathbf{R}_{q^+,i^*} - \sum_{q^-\in\mathcal{D}_r^{(-)}} \mathbf{R}_{q^-,j^*} + \underbrace{\log \frac{p\big(e|i^*,\mathcal{D}_r^{(-)}\big)}{p\big(e|j^*,\mathcal{D}_r^{(+)}\big)} - \tau\big(i^*\cup\mathcal{D}_r^{(-)}\,\big|\,e\big) + \tau\big(j^*\cup\mathcal{D}_r^{(+)}\,\big|\,e\big) - \tau\big(i^*\cup\mathcal{D}_r^{(-)}\big) + \tau\big(j^*\cup\mathcal{D}_r^{(+)}\big)}_{\epsilon}.
\end{aligned}
$$

Consequently, we reach:

$$
\vec{R}_{j^*} = \vec{R}_{i^*} + \mathbf{z}_r + \epsilon,
$$

where $\mathbf{z}_r = \sum_{i\xrightarrow{r}j} \vec{R}_j - \vec{R}_i$ and the $\epsilon$ term is highlighted in the above expression.

As we mentioned in the beginning, in the expression of $\epsilon$, the mutual independence and conditional independence terms are usually negligible compared with $\mathbf{z}_r$, so the approximation of $\vec{R}_{j^*} \approx \vec{R}_{i^*} + \mathbf{z}_r$ can hold with fine granularity under a large sample size. □

## A.5 Proof for Theorem 3

Proof. Recall that the setting we study is:

$$
\mathbf{y} = \mathbf{U}(\mathbf{X})\boldsymbol{\theta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\theta} \sim N(0,\Sigma), \quad \boldsymbol{\epsilon} \sim N(0,\sigma^2\mathcal{I}),
$$

and we consider the loss function such as $\phi(y, \hat{y}) = \sigma(y)\log\sigma(y) + (1-\sigma(y))\log(1-\sigma(y))$, and assume that the loss function is $L$-Lipschitz in both arguments.

Let the noise-free label be given by: $\mathbf{y}_0 = \mathbf{U}(\mathbf{X})\boldsymbol{\theta}$. We define: $\boldsymbol{\theta}_1^* = \arg\min \sum_{i=1}^{|\mathcal{I}|} \phi(\mathbf{X}_i\boldsymbol{\theta}, y_i)$, $\boldsymbol{\theta}_2^* = \arg\min \sum_{i=1}^{|\mathcal{I}|} \phi(\mathbf{Z}_i\boldsymbol{\theta}, y_i)$ to be the optimum when using $\mathbf{X}$ and $\mathbf{Z}$ as the features, and their predictions are given by: $\mathbf{y}_1 = \mathbf{X}\boldsymbol{\theta}_1^*$ and $\mathbf{y}_2 = \mathbf{Z}\boldsymbol{\theta}_2^*$. Therefore, the empirical training loss using $\mathbf{X}$ is given by: $L(\mathbf{y}_1, \mathbf{y}) = \frac{1}{|\mathcal{I}|}\sum_{i=1}^{|\mathcal{I}|}\phi(\mathbf{y}_{1,i}, \mathbf{y}_i)$, and the empirical training loss using $\mathbf{Z}$ is given by: $L(\mathbf{y}_2, \mathbf{y}) = \frac{1}{|\mathcal{I}|}\sum_{i=1}^{|\mathcal{I}|}\phi(\mathbf{y}_{2,i}, \mathbf{y}_i)$. It is easy to verify that $L(\cdot,\cdot)$ is $L/\sqrt{|\mathcal{I}|}$-Lipschitz in both arguments.

On the other hand, recall that the average risk $\mathcal{L}$ using $\mathbf{X}$ and $\mathbf{Z}$ for predicting the "clean label" is given by: $\mathcal{L}(\mathbf{X}) = \mathbb{E}_{\boldsymbol{\epsilon}}\Big[\frac{1}{|\mathcal{I}|}\sum_{i=1}^{|\mathcal{I}|}\phi(\mathbf{X}_i\boldsymbol{\theta}_1^*, \mathbf{y}_i)\Big]$ and $\mathcal{L}(\mathbf{Z}) = \mathbb{E}_{\boldsymbol{\epsilon}}\Big[\frac{1}{|\mathcal{I}|}\sum_{i=1}^{|\mathcal{I}|}\phi(\mathbf{Z}_i\boldsymbol{\theta}_2^*, \mathbf{y}_i)\Big]$.

By the definition, it is easy to verify that:

$$\mathcal{L}(\mathbf{X}) = \mathbb{E}_{\boldsymbol{\epsilon}}\Big[\frac{1}{|\mathcal{I}|}\sum_{i=1}^{|\mathcal{I}|}\phi(\mathbf{X}_i\boldsymbol{\theta}_1^*, \mathbf{y}_i)\Big] \geq \mathbb{E}_{\boldsymbol{\epsilon}}\big[L(\mathbf{y}_0, \mathbf{y}_0)\big] = L(\mathbf{y}_0, \mathbf{y}_0). \tag{20}$$

Then it holds for all $\boldsymbol{\theta}_2$ that:

$$\begin{aligned}
\mathcal{L}(\mathbf{Z}) &= \mathbb{E}_{\boldsymbol{\epsilon}}\big[L(\mathbf{Z}\boldsymbol{\theta}_2^*, \mathbf{y}_0)\big] \\
&\leq \mathbb{E}_{\boldsymbol{\epsilon}}\Big[L(\mathbf{Z}\boldsymbol{\theta}_2^*, \mathbf{y}) + \frac{L}{\sqrt{|\mathcal{I}|}}\|\boldsymbol{\epsilon}\|_2\Big] \quad \text{(using the Lipschitz condition of } L\text{)} \\
&\leq \mathbb{E}_{\boldsymbol{\epsilon}}\Big[L(\mathbf{Z}\boldsymbol{\theta}_2, \mathbf{y}) + \frac{L}{\sqrt{|\mathcal{I}|}}\|\boldsymbol{\epsilon}\|_2\Big] \quad \text{(by the definition of } \boldsymbol{\theta}_2^*\text{)} \\
&\leq \mathbb{E}_{\boldsymbol{\epsilon}}\Big[L(\mathbf{Z}\boldsymbol{\theta}_2, \mathbf{y}_0)\Big] + 2\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\frac{L}{\sqrt{|\mathcal{I}|}}\|\boldsymbol{\epsilon}\|_2\Big] \quad \text{(again by using the the Lipschitz condition of } L\text{)} \\
&\leq L(\mathbf{Z}\boldsymbol{\theta}_2, \mathbf{y}_0) + 2L\sigma \quad \text{(by the definition of } \boldsymbol{\epsilon}\text{)}.
\end{aligned}$$

Also, by the textbook derivation, it holds that:

$$\min_{\boldsymbol{\theta}_2}\|\mathbf{Z}\boldsymbol{\theta}_2 - \mathbf{y}_0\|_2^2 = \big\|\mathbf{Z}(\mathbf{Z}^\intercal\mathbf{Z})^{-1}\mathbf{Z}\mathbf{y}_0 - \mathbf{y}_0\big\|_2^2 = \|\mathbf{y}_0\|_2^2 - \big\|\mathbf{U}(\mathbf{Z})^\intercal\mathbf{y}_0\big\|_2^2. \tag{21}$$

Combining (20), (A.5) and (21), we have:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\epsilon}}\Big[L\big(\mathbf{Z}\boldsymbol{\theta}_2^*, \mathbf{y}_0\big) - L\big(\mathbf{X}\boldsymbol{\theta}_1^*, \mathbf{y}_0\big)\Big] &= \mathbb{E}_{\boldsymbol{\theta}}\big[\mathcal{L}(\mathbf{Z}) - \mathcal{L}(\mathbf{X})\big] \\
&\leq \mathbb{E}_{\boldsymbol{\theta}}\big[L(\mathbf{Z}\boldsymbol{\theta}_2, \mathbf{y}_0) - L(\mathbf{y}_0, \mathbf{y}_0) + 2L\sigma\big] \\
&\leq \mathbb{E}_{\boldsymbol{\theta}}\Big[\frac{L}{|\mathcal{I}|}\|\mathbf{Z}\boldsymbol{\theta}_2 - \mathbf{y}_0\|_2 + 2L\sigma\Big] \quad \text{(using the Lipschitz condition of } L\text{)} \\
&\leq \mathbb{E}_{\boldsymbol{\theta}}\Big[\frac{L}{|\mathcal{I}|}\sqrt{\|\mathbf{y}_0\|_2^2 - \big\|\mathbf{U}(\mathbf{Z})^\intercal\mathbf{y}_0\big\|_2^2} + 2L\sigma\Big] \\
&\leq \frac{L}{|\mathcal{I}|}\sqrt{\mathbb{E}_{\boldsymbol{\theta}}\big[\|\mathbf{y}_0\|_2^2 - \big\|\mathbf{U}(\mathbf{Z})^\intercal\mathbf{y}_0\big\|_2^2\big]} + 2L\sigma \quad \text{(using Jensen's inequality)} \\
&\leq \frac{L}{|\mathcal{I}|}\sqrt{tr(\Sigma) - \underline{\lambda(\Sigma)}\big\|\mathbf{U}(\mathbf{Z})^\intercal\mathbf{U}(\mathbf{X})\big\|_2^2} + 2L\sigma.
\end{aligned}$$

In the last line we use the definition of $\mathbf{y}_0$ to obtain: $\mathbb{E}_{\boldsymbol{\theta}}\Big[\big\|\mathbf{U}(\mathbf{Z})^\intercal\mathbf{y}_0\big\|_2^2\Big] = \big\|\mathbf{U}(\mathbf{Z})^\intercal\mathbf{U}(\mathbf{X})\Sigma^{1/2}\big\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm. It is then easy to verify that $\big\|\mathbf{U}(\mathbf{Z})^\intercal\mathbf{U}(\mathbf{X})\Sigma^{1/2}\big\|_F^2 \geq \underline{\lambda(\Sigma)}\big\|\mathbf{U}(\mathbf{Z})^\intercal\mathbf{U}(\mathbf{X})\big\|_2^2$, which gives the desired result. □