P-Companion: A Principled Framework for Diversified Complementary Product Recommendation

Junheng Hao[1], Tong Zhao[2], Jin Li[2], Xin Luna Dong[2]
Christos Faloutsos[2,3], Yizhou Sun[1], Wei Wang[1]
University of California, Los Angeles[1], Amazon.com[2], Carnegie Mellon University[3]
[jhao,yzsun,weiwang]@cs.ucla.edu,[zhaoton,jincli,lunadong,faloutso]@amazon.com

ABSTRACT

If one customer buys a tennis racket, what are the best 3 complementary products to purchase together? 3 tennis ball packs, 3 headbands, 3 overgrips, or 1 of each respectively? Complementary product recommendation (CPR), aiming at providing product suggestions that are often bought together to serve a joint demand, forms a pivotal component of e-commerce service, however, existing methods are far from optimal. Given one product, how to recommend its complementary products of different types is the key problem we tackle in this work. We first conduct an analysis to correct the inaccurate assumptions adopted by existing work to show that co-purchased products are not always complementary and further propose a new strategy to generate clean distant supervision labels for CPR modeling. Moreover, to bridge in the gap from existing work that CPR does not only need relevance modeling but also requires diversity to fulfill the whole purchase demand, we develop a deep learning framework, `P-Companion`, to explicitly model both relevance and diversity. More specifically, given one product with its product type, `P-Companion` first uses an encoder-decoder network to predict multiple complementary product types, and then a transfer metric learning network is developed to project the embedding of query product to each predicted complementary product type subspace and further learn the complementary relationship based on the distant supervision labels. The whole framework can be trained from end-to-end and is robust to cold-start products attributed to a novel pretrained product embedding module named `Product2vec`, based on graph attention networks. Extensive offline experiments show that `P-Companion` outperforms state-of-the-art baselines by 7.1% increase on the Hit@10 score with well-controlled diversity. Production-wise, we deploy `P-Companion` to provide online recommendations for over 200M products at Amazon and observe significant gains on product sales and profit.

KEYWORDS

Complementary product recommendation; Product relationship understanding; Product graph.

1 INTRODUCTION

Complementary product recommendation (CPR) has become increasingly critical for the success of online e-commerce such as Amazon, eBay, Taobao, etc. Such recommendations often help customers find a high-quality set of relevant products that are always bought and used together to facilitate a joint demand[1], e.g. phones and phone cases. CPR can inspire customers with more potential needs and is vital to bring better customer shopping experiences and trigger more revenue.

Throughout this paper, we define complementary product recommendation as a product-to-product recommendation problem: *given a "query" product, the goal is to recommend relevant and diverse products that are complementary to the "query" product such that they can be bought together to serve a joint intention.* In Figure 1, we show a comparative example to elaborate the requirements on generating high-quality "to-buy-together" recommendations. Given a tennis racket as a "query product", we compare three sets of recommendations. List 1 contains three other similar tennis rackets. List 2 contains three tennis balls and List 3 contains one tennis ball, one racket cover and one headband. Naturally we consider List 1, in general, are more towards substitute products and it is not likely to be purchased together in List 1. While both Lists 2 and 3 may be considered as reasonable recommendations, we consider List 3 a better option since it presents three different types of products that collectively better serve the customer's demand for tennis sport. This example illustrates that a desirable complementary production recommendation solution should take both relevance and diversity into consideration to fulfill customer's needs.

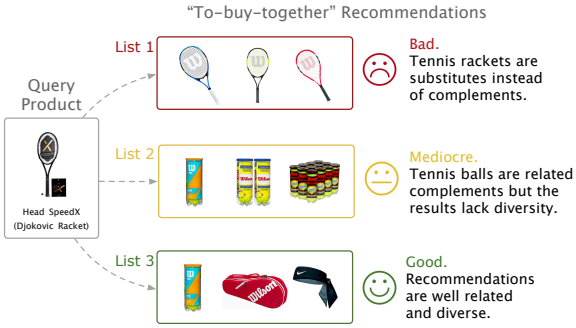

Figure 1: One example of "to-buy-together" recommendation on E-commerce. Good complementary recommendations require both relatedness and diversity.

Solving such complementary recommendation problem is non-trivial and challenging. Comparing with conventional similarity-based product-to-product recommender systems, it poses at least three challenges as follows. *C1: Complementary relationship is not symmetric and complementary recommendation is not simply based on similarity measurement.* For example, tennis rackets and headband are not similar at all to each other on textual or image features. Moreover, an SD card can be a complement product to a camera

[1]In economics, a complementary product is defined as a product whose use is directly related to the use of another base or associated product such that a surge in demand for one product results in an increase in demand for the other. We generalize this concept for e-commerce applications from the customer perspective, which refers "complements" to relevant products that are likely to be purchased and used together.

Table 1: Comparison between P-Companion and existing representative models: Co-Purchase, Sceptre and PMSC.

Property	Co-Purchase	Sceptre [17]	PMSC [23]	P-Companion
Asymmetric (C1)	✓	✗	✓	✓
Diversity (C2)	✗	✗	✗	✓
Cold-start (C3)	✗	limited	limited	advanced

but not inversely. These facts rule out most of the similarity-based approaches and require a different mechanism to model the complementary relationship. *C2: Complementary recommendation needs to consider diversity.* The recommendations are typically a set of products with diverse categories and functionalities that provide high utilization for customers' demand. As shown in Figure 1, a diversified recommendation basket, which includes three types of tennis-related products (List 3) is better than that with only one type (List 2). *C3: Complementary recommendation suffers for cold-start items.* Current methods [17, 23] often fail on such low-resource products, which widely exist in e-commerce platforms.

While most of the existing methods in recommender systems [21] focus on modeling user-item relationships by frequent pattern mining [8], matrix factorization [15], collaborative filtering [14, 20], or other neural network based recommenders [2], only a few [3, 8, 10, 16] target at explicitly modeling relationship between items. Among them, complementary relationship modeling has been scarcely investigated compared to the efforts made for modeling substitutes with similarity-based approaches. Recently, new approaches with behavior-based product graphs, which generally integrate product features and pair-wise relations obtained from customer behavioral data (such as co-purchase data), have been spotlighted and shown effective on complementary recommendation. Representative examples are Sceptre [17], which proposes a topic modeling method to infer networks of products, and PMSC [23], which incorporates path constraints in item pairwise relational modeling. However, these methods seek to distinguish substitutes and complements, fail to address these aforementioned challenges, and dive deep into modeling such properties of complementary recommendation, especially from the diversity perspective.

Besides the model incapability, existing research [17, 23] on CPR is particularly interested in using original product co-purchase and co-view data as labels for complementary and substitutable products. Also, these approaches mostly learn CPR models within one specific product category. However, when diving deep into co-purchase and co-view data, we find that these fundamental assumptions and settings are inaccurate and we have the following observations, which indicate that existing approaches on CPR is sub-optimal: first, product pairs from co-purchase and co-view records are not disjoint, and the amount of overlap heavily depends on product categories; second, co-purchase pairs (i.e. complementary relationship of products) widely exist across multiple product categories, different from the setting in current approaches where the models learn and experiment co-purchase within one category such as "electronics" or "baby". More details are discussed in Section 3.

To address the aforementioned issues in data collection and complementary relationship modeling, based on our data analysis through Amazon Mechanical Turk (MTurk) surveys, we first propose a novel collection schema to generate cleaner distant supervision labels and better capture the product relationships for complementary production recommendation. Moreover, to enable CPR across all product categories, we formulate the problem into a joint learning framework and develop a deep learning method, P-Companion(short for Product-Companion), as an end-to-end solution. Given a query product with its product type (e.g. TV), P-Companion jointly learns the complementary product type and the particular complementary products within the targeted complementary product type subspace. Specifically, P-Companion first uses the complementary type transition module to predict diverse target complementary product types, (e.g. from TV to wall mount and cables). Then, originated from transfer metric learning, a product prediction module is developed to project the embedding of query product to each of the predicted product type subspace and successfully obtain diversified complementary products among multiple types. P-Companion learns the complementary relationship between products based on the distant supervision labels from historical customer behaviors and the whole framework can be trained through end-to-end learning. Also, our model is robust to cold-start products attributed to the product embedding module Product2vecbased on graph attention networks. A comparison between P-Companion and existing representative models is summarized in Table 1. In summary, our contributions are listed in the following aspects:

- **Data Understanding.** We drop the inaccurate assumptions used in existing research on CPR. Based on observations and crowdsourced annotations on co-purchase and co-view, we propose a new approach to collect labels as distant supervision for CPR. (Section 3)
- **Methodology:** We propose a new model P-Companion, that considers both relevance and diversity in CPR modeling and yield diversified recommendations. We also introduce a graph attention based product embedding learning module that makes P-Companion robust to deal with cold-start products. (Section 4)
- **Performance:** Through new label collection schema and human evaluation by MTurk, experiments on real-world datasets show that P-Companion significantly outperforms state-of-the-art baselines by 7.1% improvement on Hit@10 score, and deliver reasonable and explainable recommendations with diversity across multiple product categories at Amazon in production. (Section 5)

2 PRELIMINARIES

In this section, we start with the definitions of Behavior-based Product Graph with product attributes and relationships used in this paper and present the formal problem formulation for diversified complementary product recommendation.

Behavior-based Product Graph (BPG) Let $\mathcal{I}$ denote the product/item[2] set, C_i denote item i's catalog features (e.g. product category, type, title and description), and $\mathcal{B} \in \mathcal{I} \times \mathcal{I}$ represent three relationships, (i.e. co-purchase $\mathcal{B}_{\text{cp}}$, co-view $\mathcal{B}_{\text{cv}}$ and purchase-after-view $\mathcal{B}_{\text{pv}}$) [3] between pairs of items, which are collected from customers' historical behaviors. In particular, for each item $i \in \mathcal{I}$, we assume there is a product type $w_i \in C_i$ that represents product i's functionality, such as hdmi-dvi-cable or over-ear-headphone.

[2] We use product and item interchangeably in this paper.

[3] More specifically, *co-purchase* means customers who purchased item x also purchased item y; *co-view* means customers who viewed item x also viewed item y; *purchase-after-view* means customers who viewed item x eventually bought item y.

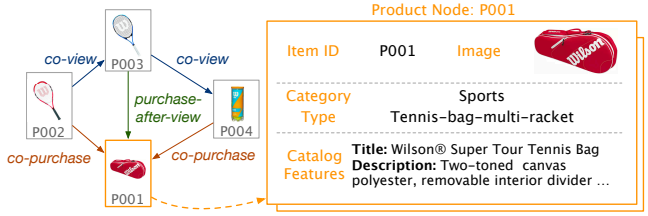

Figure 2: One snapshot of BPG. BPG is constructed with nodes as items with catalog features (type, etc) and edges as pairwise relations based on customer behavior.

Similarly, each item may also be associated with a general category, such as `electronics`. Such information can be viewed as a Behavior-based Product Graph (BPG) with products as "nodes", product types and other catalog features as "node attributes", and pairwise item relationships as "edges". Therefore, BPG is essentially a multi-relational attributed information network. Figure 2 presents a BPG snapshot of one tennis bag with multiple catalog features and related items connected by different relations.

Problem Formulation The complementary product recommendation problem in this work is formulated as follows. Taking *product catalog features* C (including title, item type, etc.) and customers behavior data $\mathcal{B}$ as inputs, we would like to learn a recommendation model $\mathcal{M}$ such that given a query item i with its product type w_i and diversity degree K, $\mathcal{M}$ can first predict K distinct complementary product types $\{w_k, k \in \{1, ..., K\}\}$ w.r.t w_i, and then generate K sets of items $\{S_{w_k}\}$ from each predicted complementary product type, aiming at optimizing their joint co-purchase probability $\sum_{k=1}^{K} \sum_{j \in S_{w_k}} \mathbb{P}_{\text{cp}}(\{i, j\})$.

3 DISTANT SUPERVISION LABEL COLLECTION FOR CPR PROBLEM

As mentioned in Section 1, we point out inaccurate assumptions and settings adopted by previous work. In this section, we elaborate with observations on real-world data and introduce a new collection schema to generate distant supervision labels for the CPR problem.

First, we observe product pairs from co-purchase and co-view records are not disjoint, and by analyzing 2 years' co-purchase and co-view records, we find there is a 20%+ overlap between co-purchase and co-view, which contaminates the multi-class signals for production relationship modeling and will confuse model learning. Moreover, we also observe that the amount of overlaps varies from product category to category. The overlap is much higher in apparels than in electronics and wireless, which is understandable as customers are more likely to buy 2 different shirts together than 2 TVs after viewing them. To clean up co-purchase, we conduct several annotation experiments via MTurk to investigate the best label collection schema among co-view, co-purchase, and purchase-after-view. In our MTurk experiments, given product pairs extracted from different combinations over co-view co-purchase and purchase-after-view, we ask annotators to label whether the two products are substitutable, complementary, or irrelevant. Throughout annotations on different combinations of co-view, co-purchase, and purchase-after-view, we observe that $\mathcal{B}_{\text{cp}} - (\mathcal{B}_{\text{pv}} \cup \mathcal{B}_{\text{cv}})$ which contains product pairs only in co-purchase records gives us the complement signals with highest MTurk voting score, which is 30% higher than pairs from co-purchase without processing.

Second, complementary relation among products is often observed across multiple categories. For example, it is quite often and understandable for a product such as "tennis racket" under the "sports" category to have potential complements "tennis shirts" under "apparels" or "tennis shoes" under category "shoes". Our observation shows that 32.93% of "electronics" products are purchased together with "home improvement", "office product", etc. To address this issue, we removed the product category restriction from [17, 23] and create a general dataset with $\mathcal{B}_{\text{cp}} - (\mathcal{B}_{\text{pv}} \cup \mathcal{B}_{\text{cv}})$ as basis across all product types for model learning, besides the electronics and grocery category (see Section 5.1). The final processed dataset contains 24M products from 34.8K product types with over 80M distant supervised complementary relationship labels for model training and evaluation.

4 MODELING

In this section, we present the main algorithm of `P-Companion` for complementary product recommendation. To enable end-to-end training for diversified CPR, we formulate `P-Companion` into a hierarchical multi-task learning framework, which enables the joint prediction of both complementary product types and complementary product items associated with each predicted product types. Figure 3 shows the high-level model architecture of `P-Companion`. The model has the three major components:

- **`Product2vec`: Graph-based Product Representation Learning**. It encodes item textual features and graph structures in BPG to learn product embeddings. It adapts the graph attention network (GAT [22]) for effective training and serves as the foundation of neural-based `P-Companion`. especially for cold-start products. The learned embeddings are used in both bootstrap model learning and inference. (Section 4.1)
- **Complementary Type Transition**. It learns the complementary transition in item type subspace, which model the asymmetric property for complementary product recommendation. Also, we can explicitly control the recommendation diversity by generating different complementary product types. (Section 4.2)
- **Complementary Item Prediction**. As the last step, we employ a projection function, with the query product embedding and predicted complementary type embeddings as input, to predict type-guided complementary products. Such subspace projection can help better perform diversified recommendation and then complementary production relationship is learned based on labels collected from Section 3 (Section 4.3).

The proposed `P-Companion` can be trained from end-to-end, and we explain the joint training objective function in Section 4.4 and supplementary details and analysis in Section 4.5. All the notations used in this paper are summarized in Table 2.

4.1 `Product2vec`

`Product2vec` is proposed to learn pretrained embedding representations for products that preserve similarities based on customer behavior data and catalog features. Based on BPG analysis results in Section 3, we observe by using $(\mathcal{B}_{\text{cv}} \cap \mathcal{B}_{\text{pv}}) - \mathcal{B}_{\text{cp}}$ links in BPG, products connected in this sub-graph are highly likely to be similar, which lead to the modeling assumption of `Product2vec` that their

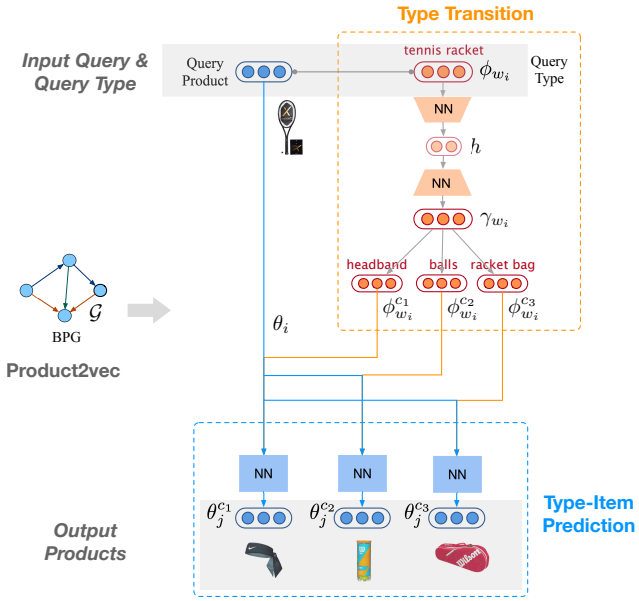

Figure 3: P-Companion model architecture for CPR. As an embedding-based recommender, it has three major components: type transition, item prediction, with product2vec for pretrained product embeddings as backend.

Table 2: Notations.

Symbol	Description
$i \in \mathcal{I}$	Item i in item set $\mathcal{I}$
w_i	Item type of i
θ_i	Product2vec embedding vector for item i
ϕ_{w_i}	Embedding vector for query type w_i
$\phi^c_{w_i}$	Embedding vector for complementary type w_i
$\theta_i^{w_c}$	Item i's projected embedding vector based on type w_c
γ_{w_i}	Complementary base vector predicted for type w_i
$y_{i,j}$	Binary label to indicate item i and j's relationship
$z_{i,j}$	Attention weight given item i and its neighbor item j
$\{W^{(k)}\}, \{b^{(k)}\}$	Learnable weight matrices and bias vectors
$\mathcal{T}$	All the item pairs used for model training
$\lambda, \lambda_i, \lambda_w, \epsilon$	different margin parameters in loss functions

embeddings should also to be close to each other. The learned embeddings will be then used as pretrained base representations for items, which foundational to support cold-start products in complementary recommendation. Figure 4 shows such Product2vec encoder model architecture.

Product2vec starts with taking each item's title and category features as input and applies an universal embedding module $FFN(\cdot)$, which consists of three feed-forward layers with a batch normalization layer, to all items i identically to obtain initial k-dimensional embeddings, as shown in Equation 1.

$$\theta_i = FFN(C_j) = \sigma\left(\sigma\left(C_i W^{(1)} + b^{(1)}\right) W^{(2)} + b^{(2)}\right) W^{(3)} + b^{(3)} \quad (1)$$

where C_i is the raw feature vector for item i's catalog and $\sigma(\cdot) = \tanh(\cdot)$ is a non-linear activation function. $W^{(1)}, W^{(2)} \in \mathbb{R}^{d \times d}, W^{(3)} \in \mathbb{R}^{d \times p}$ are weight matrices with the corresponding biased terms $b^{(1)}, b^{(2)}, b^{(3)}$.

Through $FFN(\cdot)$, each product item i has been transformed into p-dimensional representation θ_i. Then $\{\theta_i\}$ for products in $(\mathcal{B}_{cv} \cap \mathcal{B}_{pv}) - \mathcal{B}_{cp}$ sub-graph are fed into a graph attention (GAT) [22] layer, which selectively aggregates the neighbors from local connections, to fine-tune parameteres in $FFN(\cdot)$. More specifically, given an item i and the set of neighbor items $\{j\}$ in N_i, an attention vector $z_{i,j} \in \mathbb{R}^{|N_i|}$ is calculated based on θ_i and $\{\theta_j\}$ normalized on the softmax function, which can adaptively capture the similarities when summarizing over items $\{j\}$ in N_i,

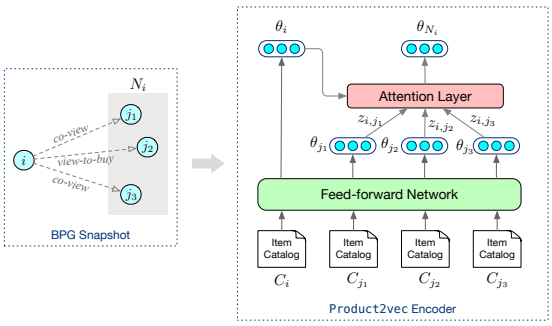

Figure 4: GNN-based Product2vec module architecture, which learns effective product embeddings given its textual features and aggregation from similar products.

$$z_{i,j} = \text{softmax}_j\left(\theta_i^T \theta_j\right) = \frac{\exp(\theta_i^T \theta_j)}{\sum_{j' \in N_i} \exp(\theta_i^T \theta_{j'})} \quad (2)$$

Thus the information of item i from weighted neighborhood aggregation N_i can be computed as, $\theta_{N_i} = \sum_{j \in N_i} z_{i,j} \theta_j$. For an item i with N_i, we mark it with a positive label $y_{i,N_i} = 1$. To conduct non-trivial model learning, for each item i, we create negative samples $\hat{N}_i$, as negative training instance (details explained in Section 4.5) with labels $y_{i,\hat{N}_i} = -1$. Therefore, the objective function of Product2vec is designed to optimize a hinge loss in Equation 2.

$$\begin{aligned} &\min \sum_{i \in \mathcal{I}} \left(l\left(y_{i,N_i}, f(i, N_i)\right) + l\left(y_{i,\hat{N}_i}, f(i, \hat{N}_i)\right)\right) \\ &= \min \sum_{i \in \mathcal{I}} \sum_{y \in \{\pm 1\}} \left\{ \max\left(\epsilon - y \cdot \left(\lambda - ||\theta_i - \theta_{N_i}||_2^2\right)\right)\right\} \end{aligned} \quad (3)$$

where $y = \{y_{i,N_i}, y_{i,\hat{N}_i}\}$, $f(\cdot)$ is the metric function, λ is the base distance to distinguish N_i and $\hat{N}_i$ and ϵ is the margin distance. Equation 3 aims at forcing the distance between θ_i and θ_{N_i} less than $\lambda - \epsilon$ while pushing θ_i far away from θ_{N_i} with a distance of at least $\lambda + \epsilon$.

It is noteworthy that $FFN(\cdot)$ is optimized by Equation 3. $FFN(\cdot)$ can be utilized to generate pretrained product embeddings for a large number of web-scale items with textural features after training, including cold-start items that help address the challenge **C1**. Such foundational embeddings act as input for subsequent complementary modeling on product items (Section 4.3).

4.2 Complementary Type Transition

In practical large-scale recommender systems, we typically do not consider all products as a candidate pool for each recommendation task, but use strategies to narrow down the candidates to only the relevant ones. This is an even critical requirement for complementary product recommendation as we need to take care of both relevance and diversity. In this subsection, we model such process as a complementary product type prediction task and propose a

neural model to (1) model the asymmetric relationship between query product type and complementary product types, e.g., camera and SD card; and (2) generate diversified complementary product types for further complementary production recommendations, e.g. SD card, filter, camera lens.

P-Companion takes pairs of query product items $\{i\}$ and candidate complement items $\{j\}$ along with their types $\{(w_i, w_j)\}$ and distant supervision label $\{y_{i,j}\}$ according to Section 3 as inputs. To model the asymmetric relationship, for each type w, we assign two learnable embedding vectors $\phi_w, \phi^c_w \in \mathbb{R}^L$ to it, indicating its context position as query type or complementary type. Given a pair of items (i, j) with types w_i and w_j respectively, we use an encoder-decoder module to transform ϕ_{w_i}, the query embedding vector of w_i, to its complementary base vector γ_{w_i}, which will be used to predict complementary types for w_i, as shown in Equations 4 and 5:

$$h = \text{Dropout}\left(\text{ReLU}\left(\phi_{w_i} W^{(4)} + b^{(4)}\right)\right), \tag{4}$$

$$\gamma_{w_i} = hW^{(5)} + b^{(5)}, \tag{5}$$

where $W^{(4)} \in \mathbb{R}^{L \times \frac{L}{2}}$ and $W^{(5)} \in \mathbb{R}^{\frac{L}{2} \times L}$ are weight matrices for encoding and decoding types. Then, we optimize the relationship between the predicted type γ_{w_i} and ground-truth type $\phi^c_{w_j}$ with the label $y_{i,j}$ by using the hinge loss function in Equation (6).

$$\min \sum_{i,j \in \mathcal{T}} \left(\max\left\{0, \epsilon_w - y_{i,j}\left(\lambda_w - \|\gamma_{w_i} - \phi^c_{w_j}\|_2^2\right)\right\}\right), \tag{6}$$

where λ_w is the base distance to distinguish γ_{w_i} and $\phi^c_{w_j}$, ϵ_w is the margin distance. Similar to Equation 3, Equation 6 aims at forcing the distance between γ_{w_i} and $\phi^c_{w_j}$ to be lower than $\lambda_w - \epsilon_w$ when $y_{i,j} = 1$ while pushing $\phi^c_{w_j}$ far away from γ_{w_i} with at least $\lambda_w + \epsilon_w$ distance when $y_{i,j} = -1$.

4.3 Complementary Item Prediction

In this section, we present a learning approach to perform recommendations on product-granularity. Attributed to the generative power of encoder-decoder model from Section 4.2 that can generate different complementary product types, we would like to conduct complementary product recommendations within each predicted complementary product type subspace to ensure the diversity. Therefore, we adopt the idea from transfer metric learning [11] and propose to enrich the representation of a given query product from a single embedding vector to multiple embedding vectors transferred by each predicted complementary type embedding. As Figure 3 describes, by using *Complementary Type Transition* module to transfer query type embedding ϕ_{w_i} to its complementary base embedding γ_{w_i}, we design an item-embedding transition module that takes advantage of the predicted type embedding vector γ_{w_i} to project the original item embedding θ_i to different complementary subspaces via Equation (7).

$$\begin{aligned} \theta_i^{w_c} &= \theta_i \odot (\phi^c_{w_c} W^{(6)} + b^{(6)}), \\ s.t.,\ &||\phi^c_{w_c} - \gamma_{w_i}||_2^2 \le \beta, \end{aligned} \tag{7}$$

where $W^{(6)} \in \mathbb{R}^{L \times d}$, $\odot$ represents element-wise product and β is the similarity threshold to determine which complementary types will be used to recommend complementary items. We can also explicitly set how many complementary types for each query type, as the implementation in our experiment. Based on different complementary type embeddings $\{\phi^c_{w_c}\}$ that are close enough to γ_{w_i}, we can transfer item i's embedding θ_i to multiple complementary targets $\{\theta_i^{w_c}\}$. For each of complementary candidate j with its type w_j, we still use a hinge loss to optimize the objective function based on $\theta_i^{w_c}$, θ_j and label $y_{i,j}$ according to Equation (8).

$$\min \sum_{i,j \in \mathcal{T}} \max\left\{0, \epsilon_i - y_{i,j}\left(\lambda_i - \|\theta_i^{w_c} - \theta_j\|_2^2\right)\right\}. \tag{8}$$

w_c is selected based on β and γ_{w_i} or a preset parameter of different types. By controlling the number of different complementary types (or selection threshold), we can successfully achieve the diversified recommendation (**C1**), comparing to existing methods.

4.4 Joint Training

By complementary type transition module introduced in Section 4.2, P-Companion can automatically target different complementary product types for different query products, and further by using multiple predicted complementary product type as projection orientations, P-Companion can perform complementary product recommendations in an end-to-end fashion without separating the training in Section 4.2 and Section 4.3. In model training phrase, P-Companion jointly optimizes both on complementary type transition and item prediction objective functions based on Equation 6 and Equation 8. To strengthen the connection between the two objective functions, for each training instance, we force ϕ_{w_c} to be the same as γ_{w_i} in Equation 7. Once the model is well-trained, P-Companion sticks with Equation 7 to predict complementary items based on different complementary types. Therefore, the overall objective function can be written as Equation 9.

$$\begin{aligned} \min \sum_{i,j \in \mathcal{T}} &\alpha\left(\max\left\{0, \epsilon_i - y_{i,j}\left(\lambda_i - \|\theta_i^{w_j} - \theta_j\|_2^2\right)\right)\right\} \\ &+ (1-\alpha)\left(\max\left\{0, \epsilon_w - y_{i,j}\left(\lambda_w - \|\gamma_{w_i} - \phi^c_{w_j}\|_2^2\right)\right\}\right), \end{aligned} \tag{9}$$

where α is a hyper-parameter to control the trade-off between complementary type modeling and complementary item modeling.

4.5 Model Details

Implementation details For Product2vec training (Equation 3), we use $(\mathcal{B}_{\text{cv}} \cap \mathcal{B}_{\text{pv}}) - \mathcal{B}_{\text{cp}}$ as positive examples and $\mathcal{B}_{\text{cp}} - (\mathcal{B}_{\text{pv}} \cup \mathcal{B}_{\text{cv}})$ as negative; for P-Companion training (Equation 9), we consider frequent co-purchase data as positive and $(\mathcal{B}_{\text{cv}} \cap \mathcal{B}_{\text{pv}}) - \mathcal{B}_{\text{cp}}$ as negative, which is according to our observations in Section 3. The negative sample ratio is fixed as 1.0 in subset datasets (see Section 5.1). Other parameters are set as follows: product embedding dimension $d = 128$, item type embedding dimension $L = 64$, margin parameters $\lambda = 1.0$ and $\epsilon = 1.0$. The trade-off parameter between type transition and item prediction is set as $\alpha = 0.8$.

Inference As shown in Figure 3, given a new product i during inference, we can obtain its product embedding θ_i (from Product2vec) and its type w_i (with type embedding ϕ_{w_i}). Based on the complementary type transition module, we can further retrieve the top-k complementary types $\{w_i^{c_1}, \ldots, w_i^{c_k}\}$. Later we can finally output complementary products $\{\theta_j\}$ by complementary item prediction module with the input of query product θ_i and multiple predicted complementary types $\{\phi^{c_1}_{w_i}, \ldots, \phi^{c_k}_{w_i}\}$.

5 EXPERIMENTS

In this section, we perform comprehensive experiments on product data from *Amazon.com* to evaluate recommendation performance of the proposed P-Companion framework.

5.1 Experiment Setup

Dataset We evaluate P-Companion on a real-world dataset obtained from *Amazon.com*, which includes 24MM of products with catalog features and customer behavioral data across 10+ product categories, containing similar product metadata as in [17, 18]. Statistics about all these three datasets are summarized in Table 3. We evaluate the model performance on query products sampled from all product categories and also provide details on two categories with large numbers of assigned products: electronics and grocery. Note that we don't apply any restrictions on the product categories when recommending complementary products regardless of query products, which means candidates can belong to any relevant product categories. This setup is consistent with our claim in Section 2 and different from existing work [17, 23].

Table 3: Dataset statistics.

Dataset	Electronics	Grocery	All-Groups
# Product items	97.6K	324.2K	24.54M
# Types	5.6K	6.5K	34.8K
# $\mathcal{B}_{\text{cp}}$ pairs	130.6K	804.1K	62.16M
# $\mathcal{B}_{\text{cv}}$ pairs	3.15M	8.96M	1,154M
# $\mathcal{B}_{\text{pv}}$ pairs	325.1K	1.105M	83.75M

Baselines We compare P-Companion with the following state-of-the-art baseline approaches. We use the default setting on hyperparameters for these baselines.

- **Co-purchase (CP)** As the most straightforward way, we can directly output the items in the co-purchase records for complementary recommendation.
- **Sceptre [17]** This approach utilizes topic modeling on item textual features from review text and logistic regression for substitute/complement classification. Category information is also applied with a sparse encoding technique.
- **PMSC [23]** Each product item has its source embedding and target embedding for query and candidate contexts. It adopts additional relation-aware parameters to model multiple item relations with path logic constraint loss and later feed in a neural network for classification.
- **JOIE [9]** It is designed for two-view knowledge graph embedding learning. We adapt JOIE to item-type views in BPG instead of entity-concept views in KG for complementary recommendation.

5.2 Evaluation on Co-Purchase Data

In real-world applications, only the best complementary products can be recommended due to the limited recommendation slots. To mimic the scenario, we choose to use an accurate ranking metric for evaluation rather than previously used link prediction accuracy [17, 23]. All the datasets used here are cleaned based on the strategy discussed in Section 2, and we use 80% for model learning and the rest 20% for evaluations. Given a query product i with its frequently co-purchased product pairs (i, j) as ground truth, we evaluate top-K recommendations S_K from P-Companion and all baselines by checking whether the model can successfully predict real co-purchased products.

Evaluation metrics A standard measurement for ranking tasks is the Hit@K score. Given a pair of items (query item i, co-purchased item j) in co-purchase test data, the Hit@K score is defined as,

$$Hit@k = \begin{cases} 1, & \text{if } j \in S_K \\ 0, & \text{else} \end{cases}, \quad K = 1, 3, 10, 60, \dots$$

where S_K is the K-element list of recommendations from the model. We report both Hit@K scores on both item level and type level (if applicable). Specifically, for JOIE and P-Companion which have the capability for complementary product type prediction, we first predict the top-3 complementary types and see whether the model can successfully predict the correct item type of j. As for the item level, we evaluate all models' ability to predict the exact co-purchased product j as a complementary product w.r.t. query product i.

To validate the effect of diversity in the recommendation, we also experiment on different settings of P-Companionas well as all baselines. Given the same set of query products, we examine the top 60 recommendations from all methods. Different from baselines, for P-Companion, we construct the top 60 recommendations by interleaving recommendations from multiple predicted complementary product types. In particular, we test P-Companion in the following four settings separately during the inference stage: recommend top 60 items only from top-1 predicted complementary type (denoted as "1 type × 60 items") together with interleaving top M items from top K complementary product types ("K types × M items/type"), where $M \times K = 60, K = 1, 3, 5, 6$.

Results We summarize the results in Table 4. P-Companion outperforms all baselines in the item level prediction with an average relative gain of 4.2% compared to the strongest baseline. In terms of type level, Comparing to JOIE with the item-type view, P-Companion improves by 9.9% on the Electronics dataset and by 4.5% on the Grocery dataset. We observe similar phenomena on the larger "All-Group" dataset with a relative 3.8% increase on item-level Hit score on average and increase on type-level against JOIE[4]. We believe this is due to the following reasons, (i) P-Companion infers complementary products by targeting the complementary type first rather than only modeling product relationships in Sceptre and PMSC. Item types can be considered as functionality abstraction and a high-quality selection of complementary types can help deliver more accurate recommendation; (ii) When modeling the complementary relationship, P-Companion follows the diversity nature of complementary products to project the embeddings of query products to multiple subspaces, which avoid the dilution and contamination across product categories.

In terms of the experiments to validate the benefit of diversified recommendations, the results are presented in Table 5. The best P-Companion diversity setup outperforms the best baseline model by a hit-score increase of 0.027 on Electronics (6 types) and 0.034 on Grocery (5 types). One can see that though the same number of items are recommended, with the increasing of complementary product types, P-Companion manages to provide a diverse recommendation explicitly and results in better item-level hit score, which further validates the diversity requirement for complementary product recommendation. Inspired by the observation, we may also reasonably recommend products of more types as complements for Electronics than Grocery, based on the results in Table 5.

[4]Due to the scalability issue, the results of PMSC is not available.

Table 4: Results of complementary recommendation based on distant supervision labels from co-purchase on the Electronics, Grocery and All-Categories datasets. P-Companion outperforms all baselines both on the type level and the item level.

Datasets	Electronics				Grocery				All Categories			
Level	Item Hit			Type Hit	Item Hit			Type Hit	Item Hit			Type Hit
Metrics	Hit@1	Hit@3	Hit@10	Hit@3	Hit@1	Hit@3	Hit@10	Hit@3	Hit@1	Hit@3	Hit@10	Hit@3
Sceptre	0.069	0.079	0.101	n/a	0.018	0.032	0.040	n/a	0.019	0.041	0.059	n/a
PMSC	0.112	0.135	0.169	n/a	0.024	0.053	0.087	n/a	n/a	n/a	n/a	n/a
JOIE	0.141	0.164	0.181	0.190	0.026	0.058	0.099	0.170	0.037	0.062	0.104	0.153
P-Companion	**0.145**	**0.170**	**0.187**	**0.206**	**0.030**	**0.063**	**0.104**	**0.177**	**0.037**	**0.068**	**0.108**	**0.161**

Table 5: Performance of P-Companion with different number of predicted item types on Electronics and Grocery dataset.

Dataset		Electronics	Grocery
Model & Setting		Hit@60	Hit@60
Sceptre		0.124	0.085
PMSC		0.179	0.139
JOIE		0.200	0.155
P-Companion	1 type × 60 items	0.138	0.088
	3 types × 20 items	0.198	0.153
	5 types × 12 items	0.222	**0.189**
	6 types × 10 items	**0.227**	0.187

Table 6: MTurk comparison between co-purchase recommendations and P-Companion's Top-3 recommendations interleaved from Top-3 complementary product types (PT).

Model	Co-Purchase	P-Companion		
		PT-1	PT-2	PT-3
% of Score 3	0.45	0.44	0.43	0.43
% of Score 2	0.25	0.27	0.27	0.27
% of Score 1	0.27	0.27	0.26	0.26
% of Score 0	0.03	0.02	0.04	0.04
Avg. Score	2.15	2.12	2.09	2.07

5.3 MTurk Evaluation

Although historical co-purchase data is good as distant supervision labels, it is far from complete and fails to include all possible truths on complements. Therefore, we leverage MTurk to evaluate the performance of P-Companion with an open-world assumption. In short, for each survey, we list 10 pairs of *query product* and *recommended product* with the question *"Given you decide to purchase the query product, would you be interested in purchasing the recommended products together?"*. For each survey question, we ask 5 different MTurk labelers to annotate their satisfaction for the recommendation and give scores from **0 (least satisfying)** to **3 (most satisfying)**. The results are summarized in Table 6. It is observed that P-Companion achieves comparable average scores with co-purchasewith similar percentage numbers on Score-3 recommendations. Meanwhile, P-Companion can provide much more diversified recommendations from multiple product types, compared to the approach that simply relies on co-purchase.

5.4 Case Study: CPR on Cold-Start Items

To evaluate the robustness of P-Companion in the production environment, we conduct analysis and case studies on cold-start products, with real-world recommendation examples.Cold-start items are defined as these products that have very limited (< 2) observed co-purchase records or even no co-purchase relations with other products. We compare the product and product type hits scores with baselines on cold-start items, and due to space limitation, we only show results in the Electronics category (other product categories show similar results) in Table 8. Still, P-Companion outperforms all baselines on low-resource items on the Electronics dataset with an average relative increase of 7.0% on the product item level and 10.9% on the product type level. Table 7 shows 4 online recommendation examples on some cold-start items. For example, for a query item "a pet-house", animal bowls and animal toys are recommended as diversified complements. Table 9 also provides examples that show reasonable complementary product type transitions to support such cold-start recommendation. One can observe that comparing with co-purchaserecords, P-Companion can generate diversified complementary product recommendations by automatically targeting on different complementary product types.

Table 7: Examples of complementary recommendation results from P-Companion on cold-start items.

Query Item	Co-Purchase	Top-3 P-Companion Recommendations
		
		
	None	
	None	

Table 8: Results on complementary recommendation on cold-start product items (H@k denotes Hit@k score).

Datasets	Electronics (*only cold-start items in testing*)			
Level	Item Hit score			Type Hit score
Metrics	H@1	H@3	H@10	H@3
Sceptre	0.049	0.065	0.081	n/a
PMSC	0.073	0.093	0.111	n/a
JOIE	0.107	0.136	0.157	0.138
P-Companion	**0.115**	**0.147**	**0.165**	**0.178**

5.5 Online Platform Performance

We further evaluate the performance of P-Companion in the production environment. With the launch of P-Companion on an AWS EMR cluster, we can generate recommendations for hundreds of millions of products on the Amazon platform within 2 hours, covering 90%+ page views. After deploying P-Companion for online

Table 9: Type transition examples. (Only top-3 transitions are listed for each type query.)

Query Type	Top-3 Type Recommendations
fajita-pan	cook-accessories, pot-holder, dutch-oven
roast-coffee-bean	fridge-coffee-cream, whole-bean, white-tea
fly-fish-line	fluorocarbon-fish-line, surf-fish-rod, fly-fish-reel

serving, we conduct online A/B testing on Amazon.com by splitting customer sessions randomly. For the control group, we use Co-purchase datasets for the recommendation, while for the treatment group, we show recommendations from P-Companion. We run the experiments for two weeks and observe relative +0.23% improvement on product sales, +0.18% improvement on profit gain, and all the results are statistical significance with p-value < 0.05. These results prove that by taking both diversity and relevance into consideration, CPR from P-Companion can significantly improve the customer shopping experience and helps them to better find their potential needs.

6 RELATED WORK

Complementary Product Recommendation (CPR) In the era of e-commerce, recommender systems are widely used to suggest relevant items given item features and user-item behaviors. Most of methods are based on collaborative filtering [14, 16], matrix factorization [15] and neural recommendation model [2, 25]. Different from most work focusing on modeling user-item relationship or similarity-based item-item relationship, we dive deep into complementary relationship discovery among items. The most straightforward way for CPR is based on frequent pattern mining and association rules [8]. Some recent works [12, 26] in this direction seek to classify whether two products are complementary or substitutable. Two representative examples is Sceptre [17] and PMSC [23] (see Section 5.1). However, they mainly operate on product level and lack diversity consideration [4, 24] in modeling and has limited capability on cold-start items. It is noteworthy that there is a thread of research on bundle list recommendation [1, 27], which aims at personalized recommendation based on user's purchase history and can be considered as a combinatorial problem. However, since we are not dealing with user-item relationship modeling, this line of research is out of the scope of our problem in this paper.

Network Embedding and Graph Neural Networks Learning representations from graph-structured data has been a spotlight in the past decade. Starting from random walk based methods (DeepWalk [19] and nodevec [5]), network embedding aims at representing nodes as low-dimensional vector representations, preserving both network topology structure and node features and easily perform subsequent graph analytic tasks. One of the state-of-the-art approaches is to use graph neural networks [7] such as graph convolutional networks (GCN) [13], GraphSAGE [6] and graph attention networks (GAT) [22]. In this paper, we adopt a GAT-based modelProduct2vecto learn product embeddings to support cold-start complementary product recommendations in P-Companion.

7 CONCLUSION

In this paper, we present P-Companion, an end-to-end solution for diversified complementary product recommendation. We first conduct data analysis to drop the inaccurate data assumptions adopted by existing work on CPR and propose a novel schema to obtain improved distant supervision labels for model learning. To model relevance and diversity, P-Companion jointly first models complementary product type transitions and design an innovative transfer metric learning component for complementary item recommendation associated with highly-related types. In addition, P-Companion can be applied on cold-start items facilitated by product representation learning module Product2vec. Experimental evaluation has shown the effectiveness of P-Companion in recommending relevant and diversified complementary items over baselines and demonstrated strong business values on our online production systems.

REFERENCES

[1] Jinze Bai, Chang Zhou, Junshuai Song, Xiaoru Qu, Weiting An, Zhao Li, and Jun Gao. 2019. Personalized Bundle List Recommendation. In *WWW*.
[2] Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *IEEE MLSP*.
[3] Arijit Biswas, Mukul Bhutani, and Subhajit Sanyal. 2017. Mrnet-product2vec: A multi-task recurrent neural network for product embeddings. In *ECML PKDD*.
[4] Lijun Chang, Chen Zhang, Xuemin Lin, and Lu Qin. 2017. Scalable Top-K structural diversity search. In *IEEE ICDE*.
[5] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *ACM SIGKDD*.
[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NIPS*.
[7] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).
[8] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. 2007. Frequent pattern mining: current status and future directions. (2007).
[9] Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. 2019. Universal representationlearning of knowledge bases by jointly embedding instances and ontological concepts. In *ACM SIGKDD*.
[10] Ruining He, Charles Packer, and Julian McAuley. 2016. Learning compatibility across categories for heterogeneous item recommendation. In *IEEE ICDM*.
[11] Junlin Hu, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. 2016. Deep Transfer Metric Learning. *Trans. Img. Proc.* 25, 12 (Dec. 2016), 5576–5588. https://doi.org/10.1109/TIP.2016.2612827
[12] Wang-Cheng Kang, Mengting Wan, and Julian McAuley. 2018. Recommendation Through Mixtures of Heterogeneous Item Relationships. In *CIKM*.
[13] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.
[14] Yehuda Koren and Robert Bell. 2015. Advances in collaborative filtering. In *Recommender systems handbook*.
[15] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* (2009).
[16] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* (2003).
[17] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *ACM SIGKDD*.
[18] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *ACM SIGIR*.
[19] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *ACM SIGKDD*.
[20] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. 2001. Item-based collaborative filtering recommendation algorithms. *WWW* (2001).
[21] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* (2009).
[22] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *ICLR* (2018).
[23] Zihan Wang, Ziheng Jiang, Zhaochun Ren, Jiliang Tang, and Dawei Yin. 2018. A path-constrained framework for discriminating substitutable and complementary products in e-commerce. In *WSDM*.
[24] Yue Wu, Jingfei Li, Peng Zhang, and Dawei Song. 2016. Learning to improve affinity ranking for diversity search. In *Asia Information Retrieval Symposium*.
[25] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* (2019).
[26] Yin Zhang, Haokai Lu, Wei Niu, and James Caverlee. 2018. Quality-aware neural complementary item recommendation. In *ACM RecSys*.
[27] Tao Zhu, Patrick Harrington, Junjun Li, and Lei Tang. 2014. Bundle recommendation in ecommerce. In *ACM SIGIR*.