

# A Method for Recommending Consumption Bundles

Adarsh Senthil

November 18, 2023

## Abstract

To satiate the demand of a consumer, we can either provide the demanded consumption bundle or recommend similar consumption bundles the consumer may prefer. Similar consumption bundles that are under the budget and supply constraints can be recommended using item embeddings, consumer state embeddings and consumer indifference functions.

The main function of a marketplace is to satiate the consumer's demand. This can be done by either recommending consumption bundles based on the consumer's state, or, given a bundle chosen by the consumer that is over budget or under supply, recommend similar bundles that are under budget and in supply. When a consumer looks to buy items, they are usually in a particular state, and that state defines a similarity relation among the various combinations of items in the market. For instance, if a consumer is hungry and their preference is to eat fruits, then combinations of items consisting solely of fruits will be similar to each other. The similarities between various items allow consumers to have flexibility in satiating their demands, since the similarity between items allows the consumer to *substitute* one for the other.

In machine learning, when it is necessary to find the similarities between objects, we usually create embedding vectors for each of the objects and apply a similarity measure such as cosine similarity. To create embedding vectors for items in a market, we could use the product descriptions and

product images of each of the items. The embedding model can be trained to predict parts of the image and description, and the weights of the trained model can then be used as the embeddings. When one item is not in supply or is causing the bundle to be over budget, we can use the embedding vectors to recommend the top-k similar items that are in supply and under budget, thus making sure the consumer’s demand is still satiated. However, due to consumer preferences, two items that are similar in their embeddings may not be similar in the sense of being preferred equally by the consumer. For instance, smartphones from different brands could be similar in their embeddings, but consumer preferences may lead to one brand being preferred over the other. To state this in terms of similarity, if two items are such that the consumer does not prefer one over the other, then the consumer is *indifferent* between the two items, i.e. the two items are similar for the consumer.

In economics, the indifference function is used to plot the set of all consumption bundles that are similar. Two consumption bundles are similar if they yield the same utility for the consumer. Although utility can be measured by ubiquitous methods such as rating systems and comment sections, the measurements are heavily dependent on the state of the consumer. For example, a consumer who buys a bundle consisting of various fruits may give a high rating for it when they are hungry, but would give the same bundle a lower rating when they are not hungry. Thus, we surmise that it would be better to define the indifference function as the set of all consumption bundles that are mapped to the same consumer state. We could then optionally use the ratings of bundles to order the indifference curve and recommend the bundles with the highest rating. The indifference function of the consumer could be learned from the consumer’s purchase history by embedding the location, time and other parameters of previous purchases into a state vector. We then retrieve the consumption bundles the consumer bought when they were in that state, and mask  $j$  non-zero elements of each bundle, to train a machine learning model to predict those  $j$  elements, conditioned on the state. The trained model would then be the indifference function for the consumer.

To consolidate, the main function of a marketplace is to satiate the consumer’s demand. The consumer’s demand is highly dependent on their current state. The current state of the consumer can be retrieved by various methods, e.g. a questionnaire. After we know the current state, we give it as

input to the state embedding function, to get the embedding vector for that state. The state embedding is then given as input to the indifference function, which outputs the corresponding consumption bundles for that state. From this set, we can either recommend bundles that are under budget and in supply, or given a bundle chosen by the consumer that is over budget or under supply, recommend similar bundles that are under budget and in supply.

The mathematical objects needed for the recommender system are:

- $b$ : Budget of the consumer.
- $\mathbf{p} = [p_1 \cdots p_n]$ : Prices of each of the items.
- $\mathbf{s} = [s_1 \cdots s_n]$ : Supply of each of the items.
- $\mathbf{c} : A \rightarrow R^m$ : Embedding function that takes in consumer state data and converts it into a  $m$ -dimensional state embedding vector.
- $E = \{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$ : Set of  $k$ -dimensional item embedding vectors.
- $I = \text{span}(E)$ : Consumption bundle space. For a consumption bundle  $\mathbf{v} = \sum_{i=0}^k a_i \mathbf{e}_i$ , the scalars  $a_i$  are interpreted as being equal to  $p_i q_i$ , where  $p_i$  and  $q_i$  are the price and demanded quantity of item  $i$  respectively.
- $\mathbf{u} : I \rightarrow R^m$ : Consumer's indifference function.
- $\mathbf{u}^{-1}(\mathbf{x})$ : Consumer's indifference set with respect to state  $\mathbf{x}$ .
- $S = \{\mathbf{v} = \sum_{i=0}^k p_i q_i \mathbf{e}_i \in I \mid (\sum_{i=0}^k p_i \leq b) \wedge (q_i \leq s_i \ \forall \ s_i \in \mathbf{s})\}$ : Bundles that are under budget and in supply.

Using these objects, we would like to perform two tasks: 1) given a state vector, recommend bundles that are in its indifference set and are under budget and in supply; 2) given a bundle chosen by the consumer that is over budget or under supply, recommend bundles that are similar to it but are under budget and in supply.

For the first task, given a state vector  $\mathbf{c}(w) = \mathbf{x}$ , the set of recommendations for the state is given by the set  $R_{\mathbf{x}} = \mathbf{u}^{-1}(\mathbf{x}) \cap S$ . For the second task, let  $\mathbf{v}$  be the bundle chosen by the consumer. Let  $B(\mathbf{v}; r)$  be the  $k$ -ball with radius  $r$ , centered at  $\mathbf{v}$ . We would like the intersection  $B(\mathbf{v}; r) \cap S$  to be non-empty. Let  $r_{min}$  solve the equation  $\min_{r \in R} B(\mathbf{v}; r) \cap S \neq \emptyset$ . Then, the set of recommendations for the bundle is given by the set  $R_{\mathbf{v}} = B(\mathbf{v}; r) \cap S, r \geq r_{min}$ .

An additional implementation step that would be helpful in the long run is to create an open standard for the item embeddings, state embeddings, prices, supplied quantities and the indifference function. The state embeddings and the model representing the indifference function can be stored on the consumer's device, while the item embeddings, prices and supplied quantities can be stored on a public database server. Any marketplace application that follows the standard can then access the market information through the public server, while asking for the consumer's permission to access their state embeddings and indifference function. This ensures that the necessary data needed to power the markets is unaffected by the product differentiation strategies used by the creators of marketplace applications. The end goal would be to turn marketplace applications into a frontend for an open, standardised market backend, thereby reducing switching costs and mitigating vendor lock-in.

## References

1. David Begg, Gianluigi Vernasca, Stanley Fischer and Rudiger. *Economics 11th ed.* McGraw-Hill Education, 2014.
2. Charu C. Aggarwal. *Neural Networks and Deep Learning: A Textbook.* Springer Nature Switzerland AG, 2023.
3. Steven Roman. *Advanced Linear Algebra.* University of California, Irvine. Springer, 2005.