Marketplace Used Product Optimal Selling Price Prediction using LSH

Hemanth Kumar Jayakumar
Master of Computer Science
Rice University
Houston, Texas
hj51@rice.edu

Sharath Giri
Master of Data Science
Rice University
Houston, Texas
sg153@rice.edu

Anitesh Reddy Surakanti
Master of Computer Science
Rice University
Houston, Texas
as361@rice.edu

***Abstract*—A majority of the marketplaces provide immense features for the customization of products when listing them, but ignore the conundrum of devising an ideal price for the product to sell within a specified amount of time for a maximum profit gain. We propose a low-latency, high-accuracy solution combining LSH tables with Machine Learning that supports embedding similarity. We experiment with the latency, accuracy, and collision aspects of the product to identify a potential use case of hashing techniques in a prediction pipeline to import latency.**

***Index Terms*—Locality Sensitive Hashing, Machine Learning, Embedding Search, Price Prediction, Hierarchical Navigable Small World (HNSW)**

I. Introduction

The current technologies in the online marketplace come with the ability to post and sell any products, new or old, to users all across the globe. While this is a boon for the users, it comes with challenges, both for the consumers and the sellers. We address one of these challenges on the seller side: Identifying the right price to list an item. To be specific, take the scenario when a user wants to sell a product of his on the marketplace and has never an item with the characteristics showing up on his search when he tried to locate the proper price range. Or if the user did find one, but that product was sold in a month after listing and the user wants it gone within the next 7 days.

While it may seem like an easy problem on first look, countless factors affect what the price of a product should "ideally" be to get sold in a reasonable amount of time. We model this challenge as a price prediction problem, which comes under the category of recommendation systems. We propose an LSH-based pipeline that identifies the right amount of price, given the knowledge of the current marketplace, and all of its products, to a certain extent. We ensure the scalability of this solution, hence the consideration of LSH, as such solutions when used on applications such as Facebook Marketplace, require good enough latency before the prediction itself is rendered unusable.

Our proposal for solving the problem is as follows:

1) (Pre-processing step) Insert all the products in the marketplace into an LSH table, governed by a MinHash Hashing function over the Product title,
2) When calculating the price, hash the product to get all the buckets having similar items.
3) Using a similarity search, ideally using an embedding model, find the closest top k products.
4) Using an aggregation function, predict the price of the product from the top k given products. (eg. Weighted mean)
5) Create a time-series estimation to identify the price by including the sale time and list time of each product within the LSH bucket
6) Explore the potential use of Hierarchical Navigable Small World (HNSW) to improve accuracy and time to query (if time permits)

This research will delve into the theoretical underpinnings of LSH and its application in the domain of price estimation and product recommendations. By combining the probabilistic nature of LSH with the richness of historical marketplace data, we aim to develop a robust methodology for extracting valuable insights, predicting selling prices with accuracy, and enhancing user experience through targeted product recommendations.

As the digital marketplace landscape continues to evolve, the findings of this research promise to contribute not only to the optimization of pricing strategies for sellers but also to the enhancement of user satisfaction by facilitating more personalized and relevant product suggestions. Through the lens of probabilistic algorithms, this study seeks to advance the understanding of LSH's potential as a transformative tool in the analysis of historical marketplace data, opening new avenues for research and application in the realm of e-commerce analytics.

II. Problem Statement

Given a marketplace with its entire list of products gathered as a dataset, we identify the right price that a new item should be listed when it enters the marketplace and recommend it as accurately as possible.

Ideally, we would like to take the historical factor of the data, such as the time it took to sell, the year/month it was listed, etc. However, due to the lack of available existing datasets and the lack of time, we constrained the problem to a certain time at which the scraping was done to get all the live product listings on a marketplace such as Amazon or eBay.

III. Literature Review

Researchers have investigated locality-sensitive hashing (LSH) for efficient similarity search and nearest neighbor recovery in high dimensional spaces. Wei and Li [1] studied using LSH to accelerate query-book correlation computation for neural sequence models. Their method uses pre-calculated term vectors and limited integer similarity values to speed up kernel computation.

LSH has also been applied to large-scale machine-learning problems. Gonzalez-Lima and Ludena [2] proposed the use of LSH for SVM classification, where projection guide models are used to find approximations of solutions in different data sets Their method reduces the effective data size, and it accelerates training. For regression and prediction tasks, LSH forests [3] have been developed. These extend randomized forests with a hashing layer, enabling faster predictions and scalability.

Here, the paper explores the use of LSH in conjunction with machine learning for market price forecasting. It attempts to leverage the advantages of LSH for rapid similarity searches to identify related products, and to apply ML to reclaimed products for reverse pricing accuracy. The impact could be a highly flexible pricing contribution to the internet on the market.

IV. Project Workflow

Below is the overall workflow of the project (as shown in figure1). As a first step we work on data gathering and follow it up with identifying similar products sold in the market place previously and then use the historic data of these identified products to estimate an optimal price to sell the users used product.

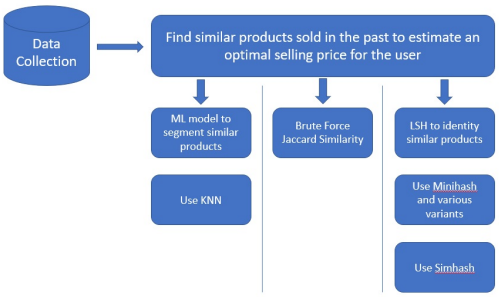

Fig. 1. workflow of the project

Initially, we intended to scrap data and use it build model to solve for the problem statement. Below are the key fields (shown in Table1) we were aiming to have extract from market places such as ebay, facebook etc. Unfortunately none of the market places holds past sold data, so we had to move away from the idea of web scrapping for data mining. As a fall back option we identified an eBay dataset of 30k records, containing the details of the product including product name, specifications, and the listing price.

Field	Discription
Product Name	Name of the product
Product Category	Category which the product belonged to
Product Manufacturer	The company which manufactured the product
Sold Price	Price at which the product was finally sold for
Initial listed Price	Initial price point the product was listed for
Listed days	Number of days the product was listed on market place
Used days	Number of days product has been already used
Other product attributes	Color, condition, size, weight, First time user etc

Fig. 2. Ideal data fields to be captured

Note, since it is listing price we will not be exposed to how long the product was listed for and the final sold price. This meant, that instead of solving for optimal price to be listed by used based on the urgency to sell, we had to settle on estimating optimal price point based on current listing (This is illustrated in figure2)

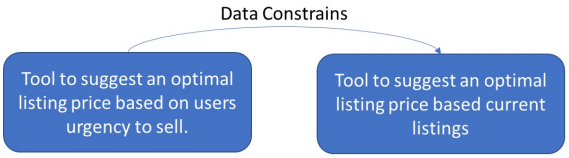

Fig. 3. Change in problem statement due to data constrain

Initially, we worked work with ML models such as KNN to segment the database based on selected attributes. Once the segmentation is done based on 90% of the data. Then we used 10% of the data to query and test for the suggested optimal selling price points. Following this, Jaccard similarity on n-grams of select attributes is used in a brute force manner to find closest products which are listed

in the market place currently. These product were intern used to estimate optimal listing price.

To improve user experience and avoid time lapse, LSH is deployed in form of minhash using n-grams and Jaccard similarity. We have tested for various hashtable sizes and number of hashtables to figure out best combinations which result in most accurate estimates. Further study is conducted by converting product attributes into embeddings and deploying simhash instead of using minhashes with n-grams. Embeddings where achieved by using Word2vec and flagembedding. Flagembedding Link

As a future scope and to future improve the model, we would like to explore Hierarchical Navigable Small World (HNSW) concept to identify similar products and then estimate optimal price. Based on the article 'Similarity Search, Part 4: Hierarchical Navigable Small World (HNSW)' HNSW seems to be a promising concept both from accuracy and time to query perspective.

V. Experimental Results

Since we are working with limited data attributes, best KPI to measure the accuracy of the model is the mean price variance and standard price deviation of the test products from the current listings. HYPOTHESIS: We expect that the similarity models perform better than KNN. LSH models should outperform brute force method due to the time taken for querying but accuracy should be slightly lower. Finally, we are expecting that using the concept of embeddings along with simhash should result in better accuracies compared to LSH using n-grams.

As previously stated we have identified an eBay dataset of 30k records, containing the details of the product including product name, specifications, and the listing price. This dataset is divided into 90:10% ratio as training and test sets.

Below are the results of Machine learning model (KNN):

- *mean variance*: 0.724
- *median variance*: 0.699
- *time taken to query test set*: 14min & 42sec

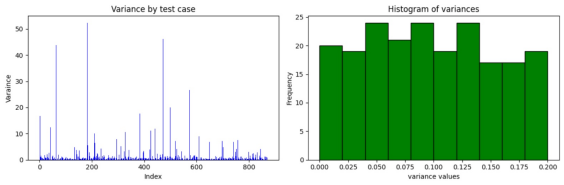

Fig. 4. Results of KNN with n=3

Below are the results from brute force method using n-grams Jaccard Similarity:

- *mean variance*: 0.5948
- *median variance*: 0.4257
- *standard deviation*: 0.5812
- *time taken to query test set*: 3min & 51sec

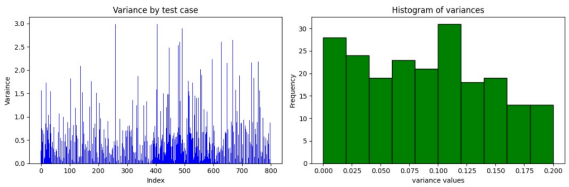

Fig. 5. Results of Bruteforce method using n-grams & Jaccard Similarity

Below are the results of Minhash LSH using n-grams:

- *mean variance*: 0.6521
- *median variance*: 0.4447
- *standard deviation*: 0.6559
- *time taken to query test set*: 16sec

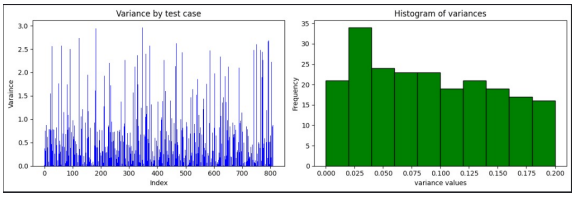

Fig. 6. Results of LSH method using n-grams & Jaccard Similarity

Below are the results from the simhash using Flagembedding & cosine similarity:

- *mean variance*: 0.5005
- *median variance*: 0.2053
- *standard deviation*: 2.9563
- *time taken to query test set*: 7min & 48sec

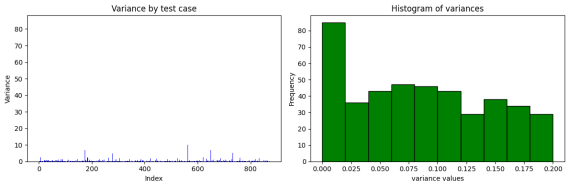

Fig. 7. Results of simhash using flagembedding & Jaccard Similarity

Below are the results from the Bruteforce using embeddings from flagembeddings & cosine similarity:

- *mean variance*: 0.4869
- *median variance*: 0.2075
- *standard deviation*: 2.5939
- *time taken to query test set*: 23min & 40sec

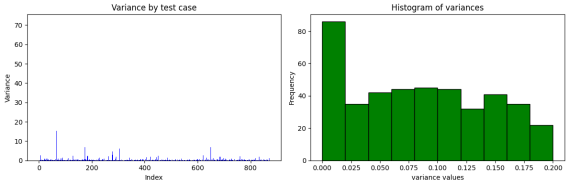

Fig. 8. Results of bruteforce using flagembedding & Cosine Similarity

VI. Conclusion

From the various studies performed, clearly identifying similarity based on embeddings along with cosine similarity of the product is resulting in similar accuracy in results. As expected bruteforce method is delivered at query times which are not practical to implement. Surprisingly, Simhash query time as well is very high and hence we cant go ahead with Simhash based on cosine similarities of the embeddings.

The accuracy of minhash based on jaccard similarity of n-grams is marginally lower compared to results from embedding & cosine similarity but the time for query is very low and is not depended on the database size. Hence based on the current studies, we can conclude that using minhash based on ngram & Jaccard Similarity is the way to go forward.

Surprisingly, KNN performance is comparable with that of other methods. Our Hypothesis did not suggest it. Overall, though minhash is producing best time, accuracy in general is not satisfactory. Primary reason for it is dataset used is very small. Firstly, for similarity search, we need to take more attributes into consideration and for better accuracy, more data points are required.

VII. Impact

The price prediction using LSH has several use cases. For example, In our use case, it is very frustrating and sub-optimal for users who like to sell used products on online platforms. This is because these cant estimate the price that is appropriate for the product taking into consideration the urgency to sell the product. if someone wants to sell a table on Facebook marketplace, such a tool will ensure that the seller gets a reasonable price and time to sell the product with a high guarantee. Combine that with the power of low latency from LSH and that makes it a highly user-friendly tool to use, even within commercial products.

References

[1] Locality Sensitive Hashing for Structured Data
Wei Wu and Bin Li
DOI : arXiv:2204.11209v2

[2] Gonzalez-Lima, Maria D. and Ludeña, Carenne C. Using Locality-Sensitive Hashing for SVM Classification of Large Data Sets
DOI : 10.3390/math10111812

[3] Bawa, Mayank and Condie, T. and Ganesan, Prasanna. LSH forest: Self-tuning indexes for similarity search
DOI : 10.1145/1060745.1060840,