Capstone Proposal

May 10, 2017

1 Machine Learning Engineer Nanodegree

1.0.1 Domain Background

Customer segmentation is the division of customers into discrete groups. It benefits companies in multiple ways:

- Optimize marketing and communication strategies
  Having a clear understanding of what segment a customer belongs to allows company to effectively communicate and market its product to him or her.
- Improving product offerings
  Customers in different segments have different needs and wants. Company can only improve it offerings to best satisfy these needs and wants after it understand what segment a customer belongs to

Traditionally, companies segments customer by combining different features of customers based on heuristics. For example, assigning a customer to the first cluster if he or she has spent more than a $1,000 in the past six months and made more than ten transactions.

Applying robust machine learning techniques to segment customer will bring about multiple benefits over the traditional approach. It can scale to much larger dataset and discover hidden patterns that can be missed by human. It allows the company to quantify how good its segmentation is based on mathematical quantities such as silhouette score. When the company acquire a new customer, it can quickly assign him or her to a cluster and adopt a communication and marketting strategy most appropriate for that cluster

1.0.2 Problem Statement

I want to apply algorithms and techniques learned in customer segments project such as PCA and k-means to a large dataset from a real-life retailer. The retailer specializes in healthy, handmade meals for busy individuals. Based on transactional data of customers, the problem is to segment customers into discrete groupings with similar features

With datasets range from hundreds of thousands to tens of millions of rows, scikit-learn running on a single computer may be too slow. As part of this project I'll investigate alternative methods that can scale up to run on multiple nodes or on GPU. I tentatively selected Tensorflow library from Google, a popular open-source library for machine learning computation.

Taking a step further, we will implement a system that can segment new customer in real-time using previously built model. This allows business to automatically include information about customer segmentation to a customer profile in order to optimize its offering and marketing for that particular customer. While this part doesn't focus on training a machine learning model, I believe knowing how to productize a machine learning model is a crucial skill of machine learning engineer

1.0.3 Datasets and Inputs

We have three types of datasets

- Customer-centric: Each row is for one customer and it contains features such as customer's dietary preference, their sign up date, their loyalty tiers. The raw dataset contains about 300,000 rows
- Transactions: Each row is for one tranaction that a customer perform. A customer can have many transactions. Transaction features includes customer id, date, total sales, location, fulfillment option i.e. delivery vs pick up, etc… This dataset has about seven million rows
- Transaction line item: Each row is for one item that were purchased as part of a transaction. Feature includes item id, its quantity and transaction id. This dataset has about twenty five million rows

1.0.4 Solution Statement

Typically, customer segmentation is a unsupervised learning problem where label for the data is missing or not reliable. We can use any clustering algorithm to achieve the segmentation. Without knowing the nature of the dataset at this point, I will choose Kmeans for its speed and simplicity. If necessary, Gaussian Mixture Model (GMM) can be used to address some shortcomings Kmeans has working with non-spherical or uneven clusters

1.0.5 Benchmark Model

We can randomly (and uniformly) assign samples into clusters and measure the performance of the assignments using an appropriate metric (see below). If our clustering algorithm has higher score, we can be confident that it performs better

1.0.6 Evaluation Metrics

There are multiple methods to measure performance of either Kmeans or GMM. At this point, it is unclear if we would have ground truth for the dataset. Therefore, the first metrics comes to mind is the Silhouette Coefficient which measures the coefficient between intra cluster distance & inter cluster distance.

$$s = \frac{(b-a)}{max(a,b)}$$

The higher the score, the better the separation.

If ground truth is available, we can use other metrics such as Adjusted Rand index and Mutual information based score

1.0.7 Project Design

Obtaining data Because we aim to segment customers, we need to convert non customer-centric dataset into customer-centric. We do that by aggragating the data into different features. For example, from the transactions dataset, we can get the dollar amount spent by a customer in the past 30, 60 or 90 days.

We can go further and aggregate the transaction line item into features such as how many times a customer has ordered a paleo dishes in the past 30, 60 or 90 days.

I can think of two types of features that will be aggregated from transactions & transaction line items dataset. Features that are related to the spending behavior and features that are related to food preference. It would be interesting to see what PCA deems as the first few principle components and how they are combined from these features

The end result of this step should be a single csv file where a customer and his or her attribute occupies one row

Exploratory analysis There are couple analysis I would perform to have a better understanding of the dataset

- Describing the data to learn about the nature of each features, i.e. its type and its range
- Visualize the distribution of each feature
- Create scatter plot for each pair of features. From this, we can obverse if there is any correlation between features

Preprocessing From the visualization perform in previous steps, if features are not normally distributed and are heavily skew, we can perform feature scaling. It'd would be interesting to apply Box-Cox test to feature scaling and compared it with a simple natural logarithm

Another preprocess step that we can do is removing outliers if applicable. We can use Tukey's Method for identfying outliers

PCA The dataset should have more than 20 features and more than 200,000 rows so it would be useful to apply PCA to reduce the number of features. Furthermore, PCA can tell you how the pricinpal components (new features) are composed of from the old features. This is helpful in understanding pattern in customer dataset

Clustering We should now have two datasets: the original and the PCA-transformed one. I'll take the following steps to apply a clustering algorithm to the dataset and observe the result

- Based on the nature of the dataset, select an appropriate clustering algorithm such as kmeans or GMM.
- Perform grid search using the selected algorithm to get the number of clusters that has the best silhouette score
- Compared with other clustering algorithms and make observation on their performance
- If possible, run the selected clustering algorithm on the original dataset and observe on its performance and silhouette score and compare it with the reduced dataset
- Visualize clusters and its central point
- Perform PCA to the dataset and select an appropriate number of components for the transformed dataset. These components should account for more than 90% variation

- I would apply an appropriate clustering algorithm such as Kmeans or Gaussian mixture model to the processed data with number of cluster starting from 2. This is essentially performing grid search in order to find the correct number of clusters that yields the best silhouette score

Production deploy In this section, I will describe plan to bring this trained clustering model to production. We will cover how to predict a datapoint as it become available as well as how to periodically update the model with latest data without having to perform manual work