**VIETNAM NATIONAL UNIVERSITY HCMC**

**INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**A blue and white logo

Description automatically generated with low confidenceDATA AGGREGATION WITH CLUSTERING**

**AND**

**DEMAND PREDICTION FOR RETAIL DATA**

By

NGUYEN DUNG MINH KHOA

A thesis submitted to the School of Computer Science and Engineering

in partial fulfillment of the requirements

for the degree of Bachelor of Computer Science

HCMC – 16/06/2022

# **ACKNOWLEDGEMENT**

The covid pandemic makes every aspect of life seem much more difficult, my undergraduate education progress also affected and procrastinated. With the devotion supportive from International University, Teachers as well as an advice of Instructors and senior members from “Seminar Bay Loan” gave me. Finally, I have finished a Thesis for a Bachelor of Computer Science program.

My Thesis would have been possible without the direct guideline and instructions from Associate Prof. Dr. Nguyen Thi Thuy Loan - my main mentor. Mrs. Loan has created a free space which helps to enhance my knowledge base not only on technical terms but the way to approach and analysis a problem as well. I personally influence and learn a lot from her working style.

I would like to extend my sincere thanks to Mr. Nguyen Van Sinh - Dean of Computer Science and Engineering Department who has taken responsibilities for my computer science’s course program. He also has the first instruction and hosts a meeting with many representatives from different companies to give me a chance to work under fresher in IT MWG. It helps me to understand the role of computer engineer and to enhance the knowledge in Information Technology domain.

I would like to extend my gratitude to the supportive from Mr. Pham Quoc Son Lam who resolve many of my confusion and guideline from my first year as a computer science student.

Thanks also to my friend – Nam Anh, Dinh Thuan and many others schoolmate, who share their point of views and help me a lot throughout Pre-Thesis and Thesis phases.

Last but not least, I am special thanks to Associate Prof. Dr. Vo Dinh Bay and others Postgraduate student for creating a favorable chance to learn about Data Mining field in “Seminar Bay Loan”.

# **TABLE OF COTENTS**

[ACKNOWLEDGEMENT 2](#_Toc106304384)

[TABLE OF COTENTS 3](#_Toc106304385)

[LIST OF FIGURES 5](#_Toc106304386)

[LIST OF TABLES 6](#_Toc106304387)

[ABBREVIATIONS 6](#_Toc106304388)

[ABSTRACT 7](#_Toc106304389)

[CHAPTER 1 - INTRODUCTION 8](#_Toc106304390)

[1.1 Background 8](#_Toc106304391)

[1.2 Problem Statement 8](#_Toc106304392)

[1.3 Scope and Objective 9](#_Toc106304393)

[1.3.1 Scope 9](#_Toc106304394)

[1.3.2 Goals 10](#_Toc106304395)

[1.4 Main Results and Contribution 10](#_Toc106304396)

[1.5 Structure of Pre-thesis 12](#_Toc106304397)

[CHAPTER 2 - LITERATURE REVIEW AND RELATED WORK 13](#_Toc106304398)

[2.1 Clustering Algorithms and Prediction 13](#_Toc106304399)

[2.2 Retail Operations and Demand Forecasting 14](#_Toc106304400)

[CHAPTER 3 - METHODOLOGY 16](#_Toc106304401)

[3.1 Model Formulation 16](#_Toc106304402)

[3.1.1 Model definition 16](#_Toc106304403)

[3.1.2 Model’s characteristics specification 18](#_Toc106304404)

[3.2 Data Aggregation with clustering 19](#_Toc106304405)

[3.2.1 Proposition 1 - The decentralized estimator satisfactions 21](#_Toc106304406)

[3.2.2 Proposition 2 - The aggregate estimator satisfies 22](#_Toc106304407)

[3.2.3 DAC algorithms 23](#_Toc106304408)

[3.2.3 PROPOSITION 3 - The algorithm satisfactions 25](#_Toc106304409)

[CHAPTER 4 – INSIGHTS OF POOLING DATA THROUGH DAC ALGORITHM 28](#_Toc106304410)

[4.1 PROPOSITION 4 [16] 28](#_Toc106304411)

[4.2 PROPOSITION 5 [16] 31](#_Toc106304412)

[4.3 PROPOSITION 6 [16] 33](#_Toc106304413)

[CHAPTER 5 – SIMULATED EXPERIMENTS 36](#_Toc106304414)

[5.1 Linear Regression 36](#_Toc106304415)

[5.2 Two Types of Items 43](#_Toc106304416)

[5.3 Logistic Regression 44](#_Toc106304417)

[CHAPTER 6 – CONCLUSION 47](#_Toc106304418)

[REFERENCES 49](#_Toc106304419)

# **LIST OF FIGURES**

[Table 5.1.1 – Used parameters in Section 5.1. 37](#_Toc106306056)

[Figure 5.1.1 - Linear regression comparison of prediction models under variety of metrics 38](#_Toc106306057)

[Figure 5.1.2 - Performance statistics over 100 independent trials (each value is the out-of-sample R-Squared). 40](#_Toc106306058)

[Figure 5.1.3 - Performance across items (each value is the out-of-sample MSE) 40](#_Toc106306059)

[Figure 5.1.4 - Comparison of prediction models (for linear regression). 41](#_Toc106306060)

[Figure 5.2 - Comparison of prediction models for a setting with two types of items (for linear regression). 43](#_Toc106306061)

[Figure 5.3 - Comparison of prediction models (for logistic regression). 46](#_Toc106306063)

# **LIST OF TABLES**

[Table 5.1 – Used parameters in Section 5.1. 37](#_Toc106307963)

[Table 5.3 - Performance across items (each value is the out-of-sample MSE). 45](#_Toc106307969)

# **ABBREVIATIONS**

GLM: Generalized Linear Model

MLE: Maximum-likelihood estimation

SKU: Stock-keeping unit

DAC: Data aggregation and clustering

# **ABSTRACT**

As a computer science student, I focus on the data mining fields of research to take advantage of what I have learn through 4-years education program that leverages mathematical and computational knowledges.

Having a change to work for MWG as a retailer in Vietnam, I find that retail industry as larger space where I can apply academic knowledge to enhance particular business processing. Promotion and Demand Forecasting is the most well-known keywork in retail that requires a lot of mathematical methodologies to solve big problems in term of optimizing business processes as well as firms’ resources.

In previous stage of researching (pre-thesis), I have discovered a lot of insight about promotion optimization and how to extend the topics to such a larger scope for multiple items and multiple supermarkets by Cohen et al. (2018). When the optimization model can yield many beneficial insights. The most important missing pieces to implement proposed models is demand forecasting parameter that is the reason I extend the scope of my researching and aim to evaluate demand metrics in retail. Demand metrics in retail show customer’s demand toward specific products that depends on certain product’s characteristics. When retails can predict accurate demand for specific product, they have to extend their results to matching those measurements to other products which share a certain feature with the indexed product because of a huge variety of products and stock-keeping units. These perspectives give two searching keys which are “Demand Prediction” and “Data Aggregation”. From that, I re-propose core algorithm called Data Aggregation and Clustering (DAC). Through theories and mathematical conduction, I have implemented the DAC models and proved its computational efficiently to segment an item to aggregation-level or cluster-level under different features in compare with common and well-known methods.

# **CHAPTER 1 - INTRODUCTION**

## **1.1 Background**

For commerce, in general, and for retailers, in specific. The biggest question is “How could we promote our services and products, gain revenue and profit?”. That question brings us with the whole deep and wide Marketing field. Skip over many technical terms and stages, the promotion is the critical tool for boosting the sales and profit, which the retailer uses on a daily basis, especially in supermarkets, drugstores, convenience stores, food mart, electronic store, books store…vv. We have a variety of ways to promote our product and services, but the most powerful and effective is Price Promotion. But the retailers cannot simply launch the promotion program, they have to consider the others complex factors such as consumer’s demand, inventory status, stock-in-house, and many product’s correlation characteristics. This led us to specific topic which is data aggregation and demand prediction in retail industry.

In the materialism behavior in present, the retailer and most supermarkets use demand forecasting on every single date. Indeed, they may offer a thousand of product or sales programs satisfied to different consumer’s demand at the same time. So, demand forecasting which help to deeply understand about product’s characteristics relationship is used by retailers for different purchasing decisions from manufactures and distributors in order to fill in the gap of customer’s need. But product’s data aggregation is not an easy task that have been a challenge and potential topic in data mining field.

## **1.2 Problem Statement**

Retailers routinely collect large amount of transactional data, which are used to improve business operation, such as inventory management, pricing making, purchasing management and promotion decision. The most critical batching data tasks for retailers is to forecast the demand for each product’s stock-keeping unit (SKU). A general approach in practice is to classify SKUs into different departments (e.g., Mobile phone) and sometimes even into sub-categories (e.g., smart phone and normal phone) and then build forecasting models respectively. A typical demand forecasting model is a regression specification with the sales (or logarithmic of the sales) as the outcome variable and price, product’s life cycle, brand, functionality, and promotion’s frequency as features. The model’s correlation parameters are then estimated using retailer’s historical data.

In different retails background, a set of items has been in sales categories for a long time (old items), while other items were recently added to sales categories and introduced to the customers. But whereas the forecasting model for old product’ SKUs is generally easy due to rich data availability, accurately predicting the demand for newly introduced SKUs or product with a limited number of historical transactions is considerably more challenging. One may then wonder how the available data of old items from the same department could be leveraged to enhance the prediction of new items. Indeed, SKUs in the same department often share similar characteristics and, hence, tend to be affected by a particular feature in a similar way.

Due to insufficient data, the traditional approach of estimating a different model for each SKU is usually inefficient for new products or SKUs with noisy observations. This approach cannot identify the right aggregation level for each coefficient and does not find the underlying cluster structure of the coefficients.

## **1.3 Scope and Objective**

### **1.3.1 Scope**

An effective approach is to estimate certain coefficients at an aggregate level (i.e., by collecting the data across all SKUs and assuming a uniform coefficient). For example, it seems rational to believe that all items in the wearable category share the same seasonal patterns. Even though this approach has been widely adopted in the retail industry, no rigorous empirical method has been developed to formalize how this data aggregation process should be applied for demand prediction. In this thesis report, we seek to cross over this gap by modeling the tradeoff between data aggregation (i.e., pooling data from different items to reduce variance) and model flexibility (i.e., estimating a different model for each item to reduce bias) in a systematic fashion.

### **1.3.2 Goals**

As we mentioned, because of insufficient data, the traditional approach of estimating a different model for each SKU is usually inefficient for new products or SKUs with low data observations. This manual methodology cannot identify the right aggregation level for each feature’s coefficient and does not find the underlying cluster structure of the coefficients.

Based on available clustering methods (e.g., k-means), and proposed work of Cohen et al. (2019), we propose an efficient and integrated approach to conclude the coefficient of each feature while identifying the right level of data aggregation based on the statistical properties of the estimated coefficients. Our method also allows us to incorporate multiple aggregation levels and preserving model interpretability. From a theoretical perspective of Cohen et al. (2019), our approach obtains a consistent estimation, along with improved asymptotic properties. From a practical perspective, our method can easily be estimated using retail data and significantly improves out-of-sample prediction accuracy.

## **1.4 Main Results and Contribution**

We study the trade-off between data aggregation and model flexibility by optimally identifying the accurate level of aggregation for each feature, and the cluster structure of the items. We re-propose a practical method - as the Data Aggregation with Clustering (DAC) algorithm, which allows us to predict demand for product while optimally identifying which features should be estimated at the (1) item, (2) cluster, or (3) aggregate levels.

Our proposed algorithm first applies maximum-likelihood estimation to yield a different coefficient vector for each item (called the decentralized model). From that, it performs a hypothesis test (i.e., t-test) on the estimated coefficients of the decentralized model to identify the correct aggregation level for each feature. To characterize the cluster structure of the items, we use the k-means method on the estimated coefficients from the decentralized model (as opposed to using features’ average coefficient).

We first formulate the DAC algorithm’s theoretical properties. Specifically, we show that it obtains a stable estimation of the data aggregation levels and cluster structures. The estimated feature coefficients under DAC are homogeneous and predictable. Thus, if the data has enough observations, one can correctly discover the underlying data generating process. In addition to this reliable result, we derive improved prediction errors - variance, mean squared error, and generalization error are all smaller - relative to the commonly used MLE method applied in a decentralized fashion for each item. Furthermore, we show that if some items have large data while other items have limited data, DAC algorithm enhances the prediction accuracy for all items, with a more significant improvement for the items with limited data.

Armed with these theoretical results, we then show computational experiments based on both simulated and real data to illustrate the DAC algorithm’s significantly improved prediction accuracy relative to 15 different benchmarks.

These results highlight the important value of the DAC algorithm in better balancing the bias-variance trade-off, resulting in more accurate demand prediction. Moreover, this DAC algorithm can accurately recognize the correct data aggregation levels and recover cluster structure in the non-asymptotic regime (i.e., when the sample size of the data set is finite). Finally, we apply the DAC algorithm using two years of FMCG retail data and find that it can also help retailers uncover useful insights on the relationships between the different items.

## **1.5 Structure of Pre-thesis**

The structure of this report is classified as follows:

In Chapter 2, we mention related retail literature steams that help and develop the clustering data algorithms and demand prediction which can applied to retailer’s operation, especially in demand forecasting process.

In Chapter 3, hence, we construct the methodology for the thesis report. We introduce and specify our model for retail settings. We will state out the crucial computational challenges as the premise to describe the Data Aggregation and Clustering (DAC) algorithm.

In Chapter 4, we present the insights and analytical results from the re-proposed algorithm (DAC).

In Chapter 5, we will show some computational results on the stimulated data to prove DAC’s predictive and stable processing in compare with several benchmarks.

In Chapter 6, we will pre-process retail transactional FMCG data of MWG to have a practical point of view for DAC and constructed model.

In the last chapter - Chapter 7 - is for the conclusion of what we have learned in this later stage of research (Thesis) and further contribution for this potential topic.

# **CHAPTER 2 - LITERATURE REVIEW AND RELATED WORK**

## **2.1 Clustering Algorithms and Prediction**

The present problems of demand prediction and clustering have been largely studied in the machine learning (ML) literature.

Donti et al. (2017) aim to develop new maximum-likelihood methods by training a predictive model to solve a nominal optimization problem. Although many studies have focused on general settings, it is not easy to apply existing methods to a retail setting where multiple levels of hierarchy may exist.

Elmachtoub and Grigas (2017) propose a new concept called ‘smart predict, then optimize’ (SPO). SPO’s key feature is that the loss function is computed based on comparing objective values which is generated by using predicted and observed data. The authors, later on, address the computational challenge and develop a tractable SPO version.

Jagabathula et al. (2018) propose an embedding model to segment a large population of customers into non-overlapping clusters with similar preferences.

Bertsimas and Kallus (2020) combine ideas from maximum-likelihood and operations research to propose a new prediction method. They solve a conditional stochastic optimization problem by incorporating various maximum-likelihood methods, such as local regression and random forests.

Currently, Liu et al. (2021) apply clustering techniques to predict the travel time of last-mile delivery services and optimize the order assignment.

## **2.2 Retail Operations and Demand Forecasting**

Retailers are always finding ways to improve operational procedures, such as inventory replenishment, supply chain management, and pricing management. These decisions require accurate demand prediction.

We refer to Fildes et al. (2019b) for a comprehensive review about this section, which show that forecasting managers in retail industry often face the dimensionality problem of having too many features consideration but lack of relevant data.

As reported by Cohen and Lee (2020), demand uncertainty is a major issue in designing efficient global supply chains. There is a wide researching aimed on developing methods for demand prediction in retail contexts. Well-developed and sophisticated models have been introduced in the last two decades to exploit the constantly increasing volume of data generated by retailers’ transactions.

Marketing papers, such as Van Heerde et al. (2000) and Mac ́e and Neslin (2004), study pre-promotion and post-promotion dips effect using linear regression models with lagged variables.

K ̈ok and Fisher (2007) develop a procedure to estimate substitution behavior in retail demand. Recent developments in demand prediction include the following three papers: Huang et al. (2014), who embed competitive information (including price and promotions) into demand prediction; Fildes et al. (2019a), who propose that promotional information can be quite valuable in improving forecast accuracy; and Huang et al. (2019), who further consider for the impact of marketing activities.

Ma et al. (2016) formulates a Lasso-based four-step methodological framework to overcome the problem of the ultra-high dimensionality of the feature space under multiple product categories.

In the operations management community, demand prediction models are a critical component for an optimization problem (e.g., Cohen et al. 2017, 2021). Specifically, Cohen et al. (2017) estimate a log-log demand model using supermarket data. The authors then solve the promotional optimization problem by developing an approximated approach based on linear programming. It was shown in the retail operations literature that responding to accurate demand forecasts can substantially increase profits (Caro and Gallien 2010).

Kesavan et al. (2010) show that incorporating the cost of goods sold, inventory, and gross margin information can substantially improve sales forecasting for retailers.

In recent years, the amount of data availability has grown exponentially, thus offering potential chances for research on demand prediction (Feng and Shanthikumar 2018).

In this thesis, our report proposes a new demand prediction method that can efficiently aggregate data from multiple items to improve prediction accuracy which called DAC-Algorithm.

# **CHAPTER 3 - METHODOLOGY**

## **3.1 Model Formulation**

We formulize the demand prediction model for retail contexts under the generalized linear model (GLM) framework. Specifically, we consider a retail department (e.g., fresh-foods or electronics) comprising *n* items (or SKUs). Each item has *m* historical observations (e.g., weekly sales transactions). In this section, we will show that our formulized model can be directly generalized to a setting that different items have a different number of observations.

### **3.1.1 Model definition**

We consider the model comprising *n* items, each item has *m* historical observations.

For item *i* (1 ≤ *i* ≤ n) and observation *j* (1 ≤ j ≤ m), we denote the (log-of-)sales as and the feature vector (e.g., price, promotion status, seasonality, functionality, and color) as which is i.i.d. with respect to observation *j* and independent with respect to item *i*. Without loss of generality, we assume that has a bounded support with and a positive definite second-moment matrix for each item *i*, that is, the smallest eigenvalue . This is a standard assumption to ensure identification and consistency in the statistics literature (e.g., Fahrmeir et al. 1985). The feature set is denoted by .

An important characteristic of our model is that a feature may affect the demand of a certain item at different data aggregation levels: (1) SKUs, (2) cluster, and (3) department.

More specifically, a feature may have the same impact on all items, captured by a uniform coefficient for all items in the department. We refer to such features as *shared* (i.e., department-level features), the set of which is denoted by . Here, we consider a setting where all the items belong to the same department to be compatible with the practical business retail context, where demand prediction is often performed for each department separately. We highlight, however, that our approach can be directly applied to a more general setting without a department structure.

Alternatively, a feature may have a different impact on different items, obtained different coefficient for each item. We refer to such features as *non-shared* (i.e., SKU-level features), the set of which is denoted by .

Finally, we suppose that the items are divided into different clusters so that some features have the same impact on items within the same cluster and a different impact on items from a different cluster. This phenomenon is captured by a consistent coefficient for all items in the same cluster (the coefficients are vary among different cluster). We refer to such features as *cluster-level features*, the set of which is denoted by .

Without losing of generality, we suppose that, for each cluster-level feature , the number of clusters or the way that the clusters are formed may be different. Thus, the whole feature set, , could be written as the union of three disjoint sets of features that affect the demand at different aggregation levels: . The aggregation structure , , and , and the cluster partition of the items are unknown a priori and will be estimated from data.

We assume that the ground truth follows the generalized linear model (GLM) specification. Specifically, the observations are generated from an exponential family distribution that includes *normal, binomial, gamma, Poisson, and inverse-normal distributions* as special cases. We refer to Fahrmeir et al. (1985) and McCullagh and Nelder (2019) for an introduction of the standard theory of GLM. Based on the three data aggregation levels of the features, we have

.

Here, is the cluster index that contains item with respect to feature represents the strictly increasing link function that setup the relationship between the linear predictor and the mean of the outcome variable, and 's are independent zero-mean random noises.

### **3.1.2 Model’s characteristics specification**

We use to denote the cluster with index that correspond to feature , where and form a partition of the items . We also call the cluster structure with respect to feature . We designate as the cluster that contains item with respect to feature . We also define if , and if . There are many commonly used link functions, and in practice, the function depends on the context. For example, if is the number of sold units of item in observation can be the identity function and, thus, the model reduces to a linear regression. On the other hand, if is a binary variable, can be the sigmoid function. Likewise, there exist other examples of link functions, such as logarithmic and inverse squared.

We assume that is sub-Gaussian with parameter , i.e., for any , which is a standard assumption in the statistics and ML literature. We assume that all the observations are independent across both time periods and items. Extensions of the model and algorithm via vector auto-regression (resp. generalized least squares) to cases where observations may be correlated across time periods (resp. items) are discussed at the end of Chapter 3.

We also define as the coefficient of in the GLM specification of Eq. (1), so, if if and if . We denote as the true coefficient vector for item and as the true coefficient matrix for all items. Likewise, we use to denote an estimator of . So is the estimator for .

Based on the GLM specification in Eq. (1), we can specify the aggregation levels of the three different types of features. For a department-level feature , its coefficient is shared among all items. Thus, all items in the department will have the same coefficient for this feature. In comparison, for an SKU-level feature , its coefficient will differ across items (i.e., for . Finally, for a cluster-level feature , all items in the same cluster will have the same coefficient (i.e., for all with , the coefficient of and that of are both . Thus, the total number of coefficients for all items is . We use to indicate the number of items that have the same coefficient as item with feature (i.e., if if , and if Estimating the coefficient of some features at a certain level of aggregation is common in practice that retailers often use to estimate seasonality coefficients at the department level to avoid overfitting and capture the fact that the items follow the same seasonal patterns (i.e fashion items). Furthermore, when estimating the promotional effect on demand, such as cannibalization and halo effects, one may cluster several items together because promotions often have a similar impact on a segment of items. Assumption 1 below simplifies the exposition by avoiding the situation where two clusters have the same coefficient value. Without loss of generality, we make the following assumption throughout the paper for expositional and computational convenience.

## **3.2 Data Aggregation with clustering**

As discussed, simultaneously estimating the aggregation levels, cluster structures, and feature coefficients is computationally challenging and led to substantial prediction errors. In this section, we propose a novel approach that allows us to (1) identify the correct level of aggregation for each feature, (2) figure the elemental cluster structure of each SKU with regard to each feature, and (3) create a stable estimation of each feature coefficients. Our method is entirely data batching and can efficiently obtain these three goals in an integrated fashion while yielding an accurate demand prediction.

We begin our analysis by focusing on a (simple) special case of the generalized linear mode (GLM) in Eq. (1), where all the features are designated at the SKU level. In this case, the data-generating process can be written as

(2)

By comparing the model’s specifications in (1) and (2), we have for for , and for . We refer to model (2) as the decentralized model because each item is fitted in a decentralized fashion. Estimating the decentralized model is usually carried out through iterative re-weighted least squares, which ultimately lead to maximum-likelihood estimation (McCullagh and Nelder 2019 , for more details). We suppose that with each of items, the decentralized model is well formulated with a unique maximum-likelihood estimation solution, which is the situation for commonly used GLMs, such as linear and logistic regression. Estimating the decentralized model could be decomposed into estimating one model for each item separately. Using the data of item , we apply the maximum-likelihood estimation to figure out the estimated coefficients of this item, as follows:

where is the likelihood function associated with the data and the coefficient vector , and is the infinitely differentiable normalization mapping in the GLM with . We refer to the estimator as the decentralized estimator. Throughout this report, we parameterize the estimators with the sample size when we want to make this dependence explicit. For example, we use to denote a decentralized estimator with sample size . Besides, we define the Fisher information matrix with respect to the decentralized model of item as

where is the Hessian operator and the expectation is taken with respect to . We first show the following consistency and normality property of the decentralized estimator , which will be used as a building block for our subsequent analyses.

### **3.2.1 Proposition 1 - The decentralized estimator satisfactions**

A - CONSISTENCY:

As , (1) if , then (2) if , then and 3 if , then , where refers to convergence in probability.

B - NORMALITY:

For each item and each feature , there exist a threshold on the sample size and a constant such that if , we have, for any ,

.

Furthermore, for each item is asymptotically normally distributed with

.

where refers to convergence in distribution, refers to the multivariate normal distribution with mean and covariance matrix .

Proposition 1(A) proves that with abundant observations, we can consistently evaluate the feature coefficients by the decentralized model. It is observable that the decentralized estimator, , is consistent given that the decentralized model has a high amount of flexibility (which implies a low bias; see also Fahrmeir et al. 1985). Proposition 1(B) further shows that the coefficient estimation error of the decentralized approach is approximately normally distributed under both finite sample and asymptotic regimes. Therefore, if we have a large number of observations for each item, the prediction accuracy of the decentralized model will be high. However, two issues remain unaddressed with the decentralized estimation: (1) how can we realize the right aggregation level for each feature, and (2) how can we identify the items’ cluster structure. Moreover, the decentralized estimation may suffer from overfitting and, hence, admit a high variance. Addressing these issues will be the main focus in the remaining of this report. To evaluate the aggregation level and the underlying cluster structure based on the GLM specification in Eq. (1), we further introduce another special case of the model in which the data aggregation level and the cluster structure are known to the retailer. We refer to this case as the aggregate model and call its maximum-likelihood estimation the aggregate estimator, which we denote by

where if (1) or (2) and .

Thus, is the estimated coefficient of feature for item under the aggregate approach. As a counterpart of Proposition 1, the following result establishes the consistency and normality of the aggregate estimator .

### 

### **3.2.2 Proposition 2 - The aggregate estimator satisfies**

(A) CONSISTENCY:

As , (i) if , then ; if , then ; and (iii) if , then

(b) NORMALITY:

For each item and each feature , there exist a threshold on the sample size and a constant such that if , we have, for any ,

.

Furthermore, if , then converges in distribution to a zero-mean degenerate multivariate normal distribution as .

### **3.2.3 DAC algorithms**

Even though the aggregate model requires understanding the data aggregation level of each feature as well the cluster structure of the items, Proposition 1 help us to infer such critical information with high accuracy from the statistical properties of the decentralized approach. Hence, we will introduce the Data Aggregation with Clustering (DAC) algorithm (described in Algorithm 1), which help to estimates the coefficient of each feature for each item, as well as correctly identifies the aggregation levels and the underlying cluster structure of the items. We denote the cumulative distribution function (CDF) of a standard normal distribution by

**The DAC Algorithm is illustrated as below:**

Text, letter

Description automatically generated

In this report, we will use to indicate the Data Aggregation with Clustering algorithm initialized with a significance level and as the estimated feature coefficient matrix of the algorithm. We will call as the estimator. By leveraging the constancy and normality of the decentralized estimate (Proposition 1), we can perform hypothesis testing (i.e., the Wald test) to figure out the correct data aggregation levels and cluster structure.

The main idea is “if one cannot reject the null-hypothesis that the estimated coefficients in the decentralized model and are the same, then it is very likely that feature is an aggregate-level feature or both items and belong to the same cluster”. A highlighted characteristic of DAC method is the usability of estimated coefficients as inputs to recognize the cluster structure of the items with respect to each cluster-level feature (see Step 6), as opposed to directly using features as in traditional clustering algorithms. We note that identifying the cluster structure in Step 6 of Algorithm 1 is very efficient since this step runs a single-dimensional -means. With every single item and feature respectively, we denote the estimated cluster that the item belongs to as . We also highlight that the last step of the DAC algorithm to fit an aggregate model can be regularized using a Lasso, Ridge, or elastic net penalty to prevent unnecessary features and reduce overfitting. This is very effective when there are high feature’s correlations that is regular in retail settings. Finally, we remark that the DAC algorithm has the same aspects as the two-stage heuristic proposed by Park et al. (2017) to address a special case of our problem, which has only cluster-level features and assumes identical cluster structures for all features. The two-stage heuristic fits the decentralized model in the first stage and then runs a multi-dimensional clustering algorithm for all the decentralized coefficients in the second stage. Our main contribution relative to this two-stage heuristic is included: (1) we adopt hypothesis testing in order to account for variety of data aggregation levels among different features; (2) we consider different cluster structures underlying different features; (3) we provide rigorous theoretical justifications for the validity of our approach and the edge of our method over the decentralized benchmark.

We now demonstrate that, with randomly high probability, the algorithm may identify the feature’s aggregation level, and the underlying cluster structure of the items with respect to each cluster-level feature. Under the algorithm with sample size , we define as the trigger on which the aggregation level of each feature and the cluster structure of each item for each cluster-level feature is correctly identified, namely, (1) and (2) is a permutation of for each . Being uniform with the decentralized and aggregate approaches, the algorithm generates a consistent and normal estimator .

### **3.2.3 PROPOSITION 3 - The algorithm satisfactions**

(A) CONSISTENCY:

There exists a positive probability strictly decreasing in with 0, such that

. (7)

Furthermore, is consistent, that is, as , (1) if , then (2) if , then ; and (3) if , then .

(b) NORMALITY:

For each item and each feature , there exist a threshold on the sample size and two constants and , such that if , we have, for any ,

. (8)

Mis-identifying the feature aggregation levels for the algorithm may cause by two types of errors in hypothesis testing (Step 2 in DAC Algorithm): (1) Type-I errors (i.e., rejecting the otherwise true null hypothesis) under which the algorithm falsely recognizes two identical coefficients as different from each other, and (ii) Type-II errors (i.e., not rejecting the otherwise false null hypothesis) under which the algorithm falsely recognizes two different coefficients to be the same. By the finite sample and asymptotic normality of the decentralized estimator (i.e., Proposition ), the Type-II errors of decay exponentially with the sample size , which shrink to 0 as approaches infinity. The Type-I errors of the algorithm is induced by the errors in each hypothesis test, which are controlled by the significance level . Although the Type-I error probability for each is upper bounded by , due to the notorious multiple hypothesis testing issue (see, e.g., Shaffer 1995 ), the total Type-I error probability of the algorithm is in general higher than Indeed, we leverage the asymptotic normality of the decentralized estimator to evaluate this probability as , which can be arbitrarily small with a proper choice of the significance level (see the proof of Proposition 3 for all the details). Equivalently, we can also follow adjust -values via Bonferroni correction from the simultaneous inference literature (see, e.g., Shaffer 1995) to implement the algorithm. In this case, the total Type-I error probability is upper bounded by , where is the (strictly decreasing) inverse of . For the practical implementation of the DAC algorithm using data, as we demonstrate in Chapter 5, the significance level for a single hypothesis test is a hyper-parameter to be fine-tuned via cross-validation. Finally, we remark that, for our demand prediction problem with heterogeneous data aggregation levels, Type-I errors are somehow acceptable in the sense that they will only cause model imprecision (i.e., the algorithm does not identify identical coefficients), but not misspecification, so that the estimation remains consistent and normal even under these errors (Proposition and (b)). Instead, Type-I errors will only affect the efficiency of the DAC estimator, giving rise to estimates with a higher variance. We will elaborate on this point in Section 4 below.

# **CHAPTER 4 – INSIGHTS OF POOLING DATA THROUGH DAC ALGORITHM**

The later parts of this thesis are focusing on characterizing the benefits of performing the pairwise tests and the clustering algorithm benchmarked against existing methodology in the literature. Specifically, we suggest three processing method from different perspectives:

(1) Comparative analysis between the DAC algorithm and the decentralized method in order to prove the value of data aggregation under efficient clustering.

(2) Simulated processing of the DAC algorithm versus several benchmarks, which show that the DAC algorithm can effectively differentiate and leverage data aggregation and the cluster structures.

(3) application of the DAC algorithm under real retail data, which showcases the practical value of our proposed method in improving demand prediction accuracy.

In this Chapter, we analyze the value of data aggregation through efficient clustering from theories by conducting several benefits of the aggregated model that relative to the decentralized model. To prove the DAC algorithm's benefits, we first examinate that if the true data-generating process has aggregate and/or cluster level features, the decentralized model presumes an overly complex model and, hence, will be prone to overfitting. To clarify this intuition, we leverage the asymptotic normality property of the MLE established by Propositions 1, 2, and 3 to obtain the following result on the estimation errors of the decentralized approach and the DAC algorithm.

## **4.1 PROPOSITION 4 [16]**

In Cohen et at. (2019), Proposition 4 is hold with the following statement:

(a) In case of the decentralized estimator , there exists a constant for each such that

(9)

(b) In case of the estimator , for the same set of constants , we have

(10)

where is the upper bound on the total Type-I error of which is characterized in Proposition 3 and the inequality is strict when .

(c) If and , there exists a threshold such that if , we have

. (11)

By the consistency of the decentralized and DAC estimators, if the number of observations becomes large, then the expected-squared-error of the estimated coefficient for each item and feature will drop off to zero. The positive constant is the asymptotic variance of the decentralized estimator . The ability to pool the data from different items makes the DAC estimator more powerful, and thus further reduce the variance of the estimates, as shown by the comparison result in Proposition . We emphasize that is the average variance of the estimated coefficient of feature for items in in case of using the decentralized estimator. Because of for , the condition can easily be satisfied. Furthermore, the Type-I error probability has as a upper bound, so that the DAC algorithm yields a smaller asymptotic error relative to the decentralized estimator for the coefficients of features at the department or cluster levels. In this case, for the same training sample, the DAC algorithm will use at least double of observations as the decentralized estimator to estimate the coefficient of such features. Hence, the estimation error will drop-off to zero much faster, especially when is large. In realistic context, a typical retail department consists of a large number of items , hence, the DAC algorithm will has a higher efficiency that relative to the decentralized estimator for cluster-level and aggregate-level features. Moreover, for these features, the value of the DAC algorithm in reducing the estimation variance of their coefficients will be stronger when the number of items increases, since it allows to pool more data to improve the estimation efficiency of these features.

We can infer from equation (10) that the estimation error of the DAC algorithm can be broken-down into two parts. For first part, bounds the error for the case when a Type-I error occurs so that the algorithm fails to identify the pooled feature coefficients. For this case, the estimation error is upper bounded by the error of the decentralized approach, , multiplied by the maximum probability that this case occurs . The second part, bounds the error for the case where the DAC algorithm correctly identifies the data aggregation levels and the cluster structures. In this case, the estimation error is upper bounded by the error of the aggregate estimator, , multiplied by the chance that the data aggregation levels, and the cluster structure are correctly specified as . As long as there are any aggregate level or cluster level features (i.e., , then the DAC algorithm can at least partially identify some (but not all) of the pooled coefficients so that the expected error will be strictly smaller relative to the decentralized model even if a Type-I error occurs. Hence, the inequality in Eq. (10) is strict when . Finally, we remark that the expected squared error driven Type-II errors of the DAC algorithm is of a lower magnitude relative to the advantage of the aggregate estimator over the decentralized and, thus, can be ignored without compromising the overall error bound of the algorithm.

## **4.2 PROPOSITION 5 [16]**

Proposition 4 illustrates the significant efficiency improvement (i.e., minimize the expected estimation error) of the DAC algorithm respect to the decentralized estimator under the presence of aggregate- and cluster-level features. We next analyze the way the DAC algorithm affects the mean squared error (MSE) of the predicted outcome under a fixed design for the linear regression model (i.e., and follows an . normal distribution with mean 0 and standard deviation . For any estimator , we define its MSE as

In Cohen et at. (2019), Proposition 5 is hold with the following statement under linear regression setting:

(a) The MSE of the decentralized estimator satisfies

(12)

where the expectation is taken with respect to the error terms .

(b) Assume that . The of the estimator satisfies

(13)

when is sufficiently large. Here, the expectation is taken with respect to both the randomness of the error terms and the DAC algorithm, is the expected number of coefficients to estimate in Step 7 of Algorithm 1, which is decreasing in and such that , and is the standard "Little-o Notation."

By comparing parts (a) and (b) of Proposition 5, we can see the fact that the expected MSE of the DAC estimator is substantially lower respect to the decentralized benchmark. Such an enhancement is demonstrated by the fact that our proposed DAC algorithm leverages hypothesis testing to pool data, so it lessens the number of model coefficients to estimate. This proves the power of aggregating data and reducing the model dimensionality in order to ultimately improve prediction accuracy.

Moreover, we study an important generalization of our base model where each item has a different number of observations . This setting fits well the scenario where some items have been offered for a long time (and, hence, have abundant data), whereas other items are new to the market (and, hence, have limited data). More specifically, we assume that, in the training set, item has observations for each . Namely, the larger , the more data is available for item . We define and .Hence, if and if . Then, the total number of data observations with the same coefficient for feature as is . To highlight the main intuition without getting trapped in technical details, we focus on the linear regression setting with independent features. Namely, we assume that for each item and for all data observations are . with mean 0 and variance 1 . The entire training dataset is denoted as , whereas the training dataset of item is denoted as .

We are interested in the generalization error ( of each item under different demand prediction methods in the random design setting, which measures the out-of-sample expected squared error assuming that the training and testing data are independently drawn from the same data generating process (see Hastie et al. 2009). For an estimation algorithm Dec, DAC trained on , where Dec refers to the decentralized approach, we define the estimator generated by as , where is the coefficient vector for item . The GE of for each item is defined as follows:

where the testing data is independently drawn from the same distribution as each training observation and the expectation is taken with respect to the randomness of both the training and testing data.

## **4.3 PROPOSITION 6 [16]**

The following proposition demonstrates the effectiveness of the DAC algorithm (in the asymptotic regime) for the case where different items have different data volumes. In Cohen et at. (2019), under a linear regression setting, the Proposition 6 is hold with the following statements if the estimators are trained on :

(a) The GE of the Dec estimator satisfies

(b) The GE of the estimator satisfies

where is the upper bound of the Type-I error of characterized in Proposition 3 and the inequality is strict if .

(c) If , we have, for each item , and sufficiently large ,

Furthermore, is convexly decreasing in and concavely increasing in for each and . Proposition 6 shows that when the sample size is sufficiently large, the DAC algorithm yields a lower generalization error relative to the decentralized approach for each item, regardless of the item's data availability. Furthermore, pooling data via our proposed DAC algorithm reduces the generalization error more substantially for items with a lower amount of data (i.e., smaller ). As expected, the prediction accuracy improvement of the DAC algorithm is strengthened when the number of features at the aggregate or cluster level, or , is higher. We further show the robustness of this result via extensive computational experiments in Section . The monotonicity result of in Proposition also indicates that if the sample size of one item increases, other items can successfully leverage this fact but with diminishing marginal returns, whereas the item itself could extract less additional information from other items' data. The proof of Proposition 6 relies on the bias-variance decomposition of the linear regression model. We also highlight that, similar to the estimation error characterized in Proposition 4, the generalization error of the DAC algorithm can be decomposed into two parts, where the first part, , measures the GE when Type-I errors occur, and the second part, , measures the GE when the DAC algorithm correctly identifies aggregation levels and cluster structures. Finally, we remark that Proposition 6 benchmarks the GE of different estimators relative to , which is the variance of the noise term and, thus, a lower bound of any GE by relying on the bias-variance decomposition (see, e.g., Eq. (7.9) of Hastie et al. 2009 and Eq. (54) in Appendix ).

To conclude this Chapter, we note that we have focused our analysis on the case where the sample size is sufficiently large. In practice, the sample size is often limited. Ultimately, one may question whether the value of pooling data via our proposed DAC algorithm remains significant in the small sample regime. Our simulation and real data studies (Chapter 5) clearly convey that the DAC algorithm efficiently identifies and leverages data aggregation and, hence, substantially improves the out-of-sample prediction accuracy relative to several common benchmarks.

# **CHAPTER 5 – SIMULATED EXPERIMENTS**

In this Chapter, we implement computational experiments using simulated data. We focus on the DAC algorithm's predictive performance and demonstrate the enhancement in prediction accuracy related to different benchmarks in the non-asymptotic regime (i.e., when the data sample size is moderated or small). We consider two GLM settings:

(1) Linear regression (i.e., see Section

(2) logistic regression (i.e., ; see Section 5.3).

The model's performance in linear regression is evaluated by the out-of-sample while in logistic regression, it is evaluated by the area under the ROC curve (AUC). We also tackle a deeper analysis in order to estimate the effluence of different parameters on the model's performance. All the results demonstrated in this Chapter is implemented from the modified source code from original authors.

## **5.1 Linear Regression**

We assume that the data is generated from the following linear model:

where are i.i.d. random variables. Each feature, , is generated randomly from a uniform distribution, and each coefficient is obtained from a uniform distribution. We hard code the number of clusters for all feature and vary the parameters one at a time (we observed similar results when 's value is varied). The definition and range of values for these parameters are reported in Table 5.1.1 below.

Table 5.1 – Used parameters in Section 5.1.

|  |  |
| --- | --- |
| **Parameter** | **Range of values** |
| Number of items |  |
| Number of features |  |
| Number of observations |  |
| Variance of the noise |  |
| Department-level probability | or |
| Cluster-level probability | or |

The parameters demonstrate the probability of given feature at aggregate levels while q for cluster’s level ones (different features are drawn independently). Thus, the remain probability is the case when the given features are at SKU level.

We need to pay attention that the DAC algorithm's implementation admits three hyper-parameters (i.e., , and ) in addition to the numbers of clusters 's. These three parameters represent the strictness of our algorithm in determining whether a feature should be aggregated. Initially, is the -value cutoff for statistical significance and is usually set at or (we apply in our model) . The parameters and represent thresholds for the ratio of non-rejected hypotheses. For example, assume that the percentage of non-rejected hypotheses for feature is (i.e., of the items have statistically close estimated coefficients). Then, we label feature as as a -level feature if and as a department-level feature if . Feature j is labeled as a cluster-level feature For any intermediate value . An optimal way to set the parameters and is by performing a cross-validation procedure. The parameters , , and give flexibility in the tolerance level of the algorithm. In this linear regression setting, we fine-tune these parameters by testing a grid of values. When implementing our algorithm with real data, we will carefully set their values using a rigorous cross-validation procedure.

In order to test the performance of the DAC algorithm, we consider the following four benchmarks: decentralized, decentralized-Lasso, centralized, and clustering. For each problem instance (i.e., a specific combination of parameters ), we modify 150 independent trials (i.e., datasets) and use as training and as testing for each trial. We then report in the average out-of-sample metrics across all items and observations. Below is a description of the methods we consider:

Graphical user interface, chart

Description automatically generated

Figure 5.1.1 - Linear regression comparison of prediction models under variety of metrics

1. DAC: We process our algorithm with , and .

2. Clustering: We first cluster the items using -means based on their features. We then fit an OLS model for each cluster.

3. Centralized: This is a naive OLS model which we suppose that for each feature, it will yield the same coefficient for all items.

4. Decentralized: We evaluate a simple OLS model for each item once at a time (i.e., models).

5. Decentralized-Lasso: This is same as the decentralized method, but we add an regularization term to each OLS model (we yield similar results when using a regularization based on Ridge regression or Elastic net).

As we can observe from Figure 5.1.1, DAC algorithm exceeds other 4 the benchmarks in all settings, in terms of out-of-sample . Similar results were observed for the MSE. As expected, as we increase the number of observations, the prediction accuracy of the DAC algorithm will also increase. Besides, we figure that the convergence DAC algorithm is faster in compare with the decentralized OLS method (not affected by regularization), so it enhances the benefit of the DAC algorithm in context of limited data availability. This clearly proves the optimal performance of data aggregation and clustering present in DAC algorithm. More interesting, the performance is not affected by the number of items or by the number of features (**indeed, the performance of each method depends on the proportion of the different feature types rather than the absolute number of features**). Last but not least, as expected, a higher negatively impacts the prediction accuracy of all methods, as it led to constraint in identifying the aggregation structure. So far, DAC algorithm has a substantial advantage relative to the four benchmarks.

Parameters:

Table

Description automatically generated

Figure 5.1.2 - Performance statistics over 100 independent trials (each value is the out-of-sample R-Squared).

Table

Description automatically generated

Figure 5.1.3 - Performance across items (each value is the out-of-sample MSE)

The applied parameters in figure 5.13 include:

To complement the results in Figure 5.1.1, we conduct statistical reports on the results of the different methods. In Figure 5.1.2, we report the performance statistics (measured by the out-of-sample ) over the 100 independent trials. In Figure 5.1.3, we report the performance statistics across different items (in this case, the per-item performance is captured by the MSE). As we can see from both tables, the DAC algorithm exceeds others 4 benchmarks in all metrics. In addition, the standard deviation-both across instances and across items-reduces significantly, hence suggesting that our proposed algorithm also decreases the variability of the performance.

Graphical user interface

Description automatically generated

Figure 5.1.4 - Comparison of prediction models (for linear regression).

Figure 5.1.4 demonstrates the performance of the methods when we modify the probability of the features in terms of aggregation level. When a large proportion of the features are at the department level (i.e., is getting close to 1), all five methods perform well (the DAC algorithm still performs best in all cases). However, for the case where the structure is more diverse, our algorithm significantly outperforms the four other benchmarks. Importantly, in all cases, the DAC algorithm leverages the structure of the problem by aggregating the data, ultimately yielding a higher prediction accuracy.

To further estimate the performance of the DAC algorithm, we consider two additional performance metrics:

(1) Measuring the capability to correctly identify the data aggregation levels of the features,

(2) Measuring the capability to accurately recover the cluster structure of the items with respect to cluster-level features.

To quantify the first performance metrics, we investigate the proportion of features for which the DAC algorithm correctly identifies their aggregation level. This metric is proposed in a similar fashion as the *Accuracy metric* used by Jagabathula et al. (2018). In order to quantify the second performance metrics, we measure the capability to recover the cluster structures using the index (see Rand 1971), which is constructed as

where is the number of pairs whose items belong to the same cluster and predicted to be in the same cluster, whereas is the number of item pairs that are in different clusters and predicted to be in different clusters, so is total number pairs of items. The capability which the algorithm is in recovering the true cluster structure depend on the value of RI. We consider the standard -means clustering algorithm as a benchmark to recover the cluster structure, where we first cluster the items into different clusters based on the average normalized values of the features , and then estimate a demand model for each cluster.

We underline that to compute those two metrics, we have to access to the ground truth values of the aggregation levels and the cluster structure, which is not the case in real data scenario. In such context, the possible way to measure the performance of DAC algorithm is computing the out-of-sample prediction accuracy. We consider the same computational setting as in Table 5.1.1.

## **5.2 Two Types of Items**

Here, we evaluate an interesting setting in which a subset of the items has limited data (referred to as 'new items'), while other items have rich data (referred to as 'old items'). The purpose is to prove that DAC algorithm can utilize the data of the old items to improve the accuracy of prediction for both types of items. These computational experiments enhance the analytical result derived in Proposition 6 by considering a non-asymptotic regime. Specifically, we consider a setting with items, features, , , and . It then fixes to the situation where the old items have four times more observations than the new items. We then vary the proportion of new items, captured by the parameter . When vary to 1, it corresponds to the setting where all the items have a lack of data (i.e., they all have observations). As previous experiment, we consider a setting of 100 independent trials and use the modified values of , and .

Graphical user interface, application

Description automatically generated

Figure 5.2 - Comparison of prediction models for a setting with two types of items (for linear regression).

Figure 5.2 present the computational results of this case. In the top panel, we show the average out-of-sample across all items in term of . As we expected, the DAC algorithm consistently outperforms all compared benchmark.

We note that the notion of "good performance" in this context is subjective, as it depends on the performance of alternative methods. We will consider below a benchmark approach for the task of recovering the cluster structure and convey the superiority of our approach. The benefit of the proposed algorithm relative to the decentralized OLS method increases when increases. In the bottom two panels, we spilt new items and old items in order to compute their out-of-sample in Panel b and Panel C respectively besides focusing on the comparison between DAC and decentralized methods. The results enhance two insights drawn from Proposition 6:

(1) the DAC algorithm improves the prediction accuracy for both types of items,

(2) the improvement is more considerable for the items with limited data.

## **5.3 Logistic Regression**

In the last simulated experiment, we conduct computational experiments for a classification problem when the logistic regression model is used for data-generating process. In which for and ,

That is the Sigmoid function. We use a similar setting as in Section and use the same values of , and as before. We generate the data matrix from a uniform distribution and the coefficients from a uniform distribution firstly. Hence, the variable ’s outcome is generated based on a Bernoulli distribution with parameter . As in the 5.1 case, we systematically vary each of parameters at a time. The parameters' value ranges are summarized in Table 5.3.

Table 5.3 - Performance across items (each value is the out-of-sample MSE).

|  |  |
| --- | --- |
| **Parameter** | **Range of values** |
| Number of items |  |
| Number of features |  |
| Number of observations |  |

Following several studies on binary classification problems, we take advantage of the AUC as the metric to analyze the performance of the different models. AUC is defined as the area under the receiver operating characteristic (ROC) curve (see, e.g., Bradley 1997). It can be interpreted as the probability that a prediction model is correctly ranking a random positive outcome higher than a random negative outcome. We compare DAC algorithm with to three benchmarks: decentralized, centralized, and clustering that we have defined in section 5.1 For each instance, we create 100 independent trials and report the average out-of-sample AUC scores.

Graphical user interface

Description automatically generated

Figure 5.3 - Comparison of prediction models (for logistic regression).

As we can see from Figure 5.3, DAC method exceed the benchmarks in all cases. Regardless of how we modify{n, m, d}, the DAC algorithm still outperforms the three other methods in terms of prediction accuracy. Furthermore, a similar result as in Figure 5.3 was also observed for the logistic regression model (the details are omitted to avoid repetition).

To summary, our simulated studies prove a stable and robust performance improvement for DAC algorithm in compare with several benchmarks (which are commonly used in practice and in the literature), even though the sample size is limited. Other benchmarks will be conducted in the further stages of research, where the prediction algorithms are implemented. The DAC algorithm efficiently aggregates data and correctly identifies the data aggregation levels of the features and the cluster structures of the items, thus ultimately improving the prediction accuracy for both regression and classification problems.

# **CHAPTER 6 – CONCLUSION**

Demand and sales forecasting is an important task faced by most retailers. By improving prediction accuracy and inferring insights on data aggregation, we can significantly leverage retailers’ decisions and profits. When modeling and estimating predictive models, retailers need to decide the aggregation level of each feature (e.g., functionality, promotion, and price). Some features may be estimated at the SKU level, while others are estimated at the department level, or at a cluster level. Traditionally, this problem was addressed by trial-and-error or by relying on domain experience. It is common to see such problems is converted and applied in such a data mining appliance and testing a multitude of model specifications until find the best aggregation level for each feature. Such an ad-hoc approach can be tedious and is not scalable for cases with a large number of features and items. The goal of this paper is to develop an efficient method to simultaneously determine

(1) The correct aggregation level of each feature

(2) The underlying cluster structures

(3) The estimated coefficients.

We re-propose a method referred to as the Data Aggregation with Clustering (DAC) algorithm. The DAC algorithm help to classify the correct aggregation level and identify the cluster structure of the items. This method is tractable even in multi-dimensionality scatter data, and it can significantly improve the efficiency in estimating the model parameters. We first derive several analytical results to demonstrate the validity and benefits of our proposed method. Specifically, we prove that the DAC algorithm obtains a consistent estimate, along with improved asymptotic properties relative to the decentralized method. HEHER We then extend the theory and implement the DAC algorithm using a large retail dataset. In all our computational tests, we observe that the proposed method significantly improves prediction accuracy relative to a multitude of benchmarks. Finally, we convey that our method can help retailers uncover useful insights from their data.

# **REFERENCES**

[1]. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1(14), 281–297.

[2]. Bernstein F, Modaresi S, Saur ́e D (2019) A dynamic clustering approach to data-driven assortment person- alization. Management Science 65(5), 2095–2115.

[3]. Cooper LG, Baron P, Levy W, Swisher M, Gogos P (1999) Promocast: A new forecasting method for promotion planning. Marketing Science 18(3), 301–316.

[4]. Cooper LG, Giuffrida G (2000) Turning datamining into a management science tool: New algorithms and empirical results. Management Science 46(2), 249–264.

[5]. Hu K, Acimovic J, Erize F, Thomas DJ, Van Mieghem JA (2019) Forecasting new product life cycle curves: Practical approach and empirical analysis. M&SOM 21(1), 66–85.

[6]. Park YW, Jiang Y, Klabjan D, Williams L (2017) Algorithms for generalized clusterwise linear regression. INFORMS Journal on Computing 29(2), 301–317.

[7]. Macé S, Neslin SA (2004) The determinants of pre- and postpromotion dips in sales of frequently purchased goods. J. Marketing Res. 41(3), 339–350.

[8]. Cohen, Maxime C. and Perakis, Georgia, Promotion Optimization in Retail (May 1, 2018). Available at SSRN: <https://ssrn.com/abstract=3194640>.

[9]. Cohen, Maxime C. and Kalas, Jeremy and Perakis, Georgia, Promotion Optimization for Multiple Items in Supermarkets (October 29, 2017). Available at SSRN: <https://ssrn.com/abstract=3061451>.

[10]. Ailawadi KL, Gedenk K, Lutzky C, Neslin SA (2007) Decomposition of the sales impact of promotion-induced stockpiling. J. Journal of marketing research. 44(3), 450–467

[11]. Wikimedia Foundation. (2022, May 25). Eigenvalues and eigenvectors. Wikipedia. Retrieved June 13, 2022, from <https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors>.

[12]. Probability distribution: Types of distributions. Analytics Vidhya. (2020, July 26). Retrieved June 13, 2022, from <https://www.analyticsvidhya.com/blog/2017/09/6-probability-distributions-data-science/?utm_source=blog&amp;utm_medium=everything-know-about-p-value-from-scratch-data-science>.

[13]. Kim, S. (2019, July 9). Bayes' classifier with maximum likelihood estimation. Medium. Retrieved June 13, 2022, from <https://towardsdatascience.com/bayes-classifier-with-maximum-likelihood-estimation-4b754b641488>.

[14]. Wikimedia Foundation. (2022, March 13). Rand index. Wikipedia. Retrieved June 13, 2022, from <https://en.wikipedia.org/wiki/Rand_index#:~:text=The%20Rand%20index%20or%20Rand,is%20the%20adjusted%20Rand%20index>.

[15]. AUC-Roc Curve in machine learning clearly explained. Analytics Vidhya. (2020, July 20). Retrieved June 13, 2022, from <https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/>.

[16] Cohen, Maxime C. and Zhang, Renyu and Jiao, Kevin, Data Aggregation and Demand Prediction (June 28, 2019). Forthcoming to Operations Research, Available at SSRN: https://ssrn.com/abstract=3411653 or http://dx.doi.org/10.2139/ssrn.3411653.