Identifying Linear Models in Multi-Resolution Population Data Using Minimum Description Length Principle to Predict Household Income

CHAINARONG AMORNBUNCHORNVEJ, NAVAPORN SURASVADI, ANON PLANGPRASOPCHOK, and SUTTIPONG THAJCHAYAPONG, Thailand’s National Electronics and Computer Technology Center (NECTEC)

One shirt size cannot fit everybody, while we cannot make a unique shirt that fits perfectly for everyone because of resource limitations. This analogy is true for policy making as well. Policy makers cannot make a single policy to solve all problems for all regions because each region has its own unique issue. At the other extreme, policy makers also cannot make a policy for each small village due to resource limitations. Would it be better if we can find a set of largest regions such that the population of each region within this set has common issues and we can make a single policy for them? In this work, we propose a framework using regression analysis and Minimum Description Length (MDL) to find a set of largest areas that have common indicators, which can be used to predict household incomes efficiently. Given a set of household features, and a multi-resolution partition that represents administrative divisions, our framework reports a set \( C^* \) of largest subdivisions that have a common predictive model for population-income prediction. We formalize the problem of finding \( C^* \) and propose an algorithm that can find \( C^* \) correctly. We use both simulation datasets as well as a real-world dataset of Thailand’s population household information to demonstrate our framework performance and application. The results show that our framework performance is better than the baseline methods. Moreover, we demonstrate that the results of our method can be used to find indicators of income prediction for many areas in Thailand. By adjusting these indicator values via policies, we expect people in these areas to gain more incomes. Hence, the policy makers will be able to make policies by using these indicators in our results as a guideline to solve low-income issues. Our framework can be used to support policy makers in making policies regarding any other dependent variable beyond income in order to combat poverty and other issues. We provide the R package, MRReg, which is the implementation of our framework in the R language. The MRReg package comes with a documentation for anyone who is interested in analyzing linear regression on multi-resolution population data.

CCS Concepts: • Computing methodologies → Machine learning approaches; • Information systems → Data mining; • Applied computing → Sociology;

Additional Key Words and Phrases: Multi-resolution data, regression analysis, minimum description length, population data, model selection

This work was supported in part by the Thai People Map and Analytics Platform (TPMAP), a joint project between the office of the National Economic and Social Development Council (NESDC) and the National Electronic and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA), Thailand. Authors’ address: C. Amornbunchornvej (corresponding author), N. Surasvadi, A. Plangprasopchok, and S. Thajchayapong, Thailand’s National Electronics and Computer Technology Center (NECTEC), 112 Phahonyothin Road, Khlong Nueng, Khlong Luang District, Pathum Thani, 12120, Thailand; emails: {chainarong.amo, navaporn.surasvadi, anon.plangprasopchok, suttipong.thajchayapong}@nectec.or.th.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2020 Copyright held by the owner/author(s).
1556-4681/2020/12-ART15
https://doi.org/10.1145/3424670

ACM Transactions on Knowledge Discovery from Data, Vol. 15, No. 2, Article 15. Publication date: December 2020.
1 INTRODUCTION

Ending poverty in all its form everywhere has been recognized as the greatest global challenge in the 2030 Agenda for Sustainable Development [8]. Annually, there are at least 18 million human deaths caused by poverty [39]. A common goal is to increase the income of people living below the poverty line (e.g., 1.90 USD per day). However, poverty alleviation often requires comprehensive measures. In practice, implementing poverty alleviation programs would depend on the realities, capabilities, and level of development of each nation. In the past, there was a tendency to use a one-size-fits-all policy as a solution at the international level [29]. Even though one-size-fits-all policies enable uncomplicated deployment of government resources, in reality, they are not suitable in alleviating poverty as each region possesses different problems and socioeconomic characteristics [10]. The solution for one region might not work for another region even if they share some characteristics [29, 42]. If one were to drill down to the household-level poverty, one would often find that each household faces different problems in respect to demographic, health, education, living standards, and accessibility to available resources. Subsequently, lifting those household members above the poverty line in a sustainable way would often require targeted poverty alleviation with unique solutions. However, in reality, many developing countries often do not possess enough resources. Hence, it is challenging for policy makers to be able to optimize between cost-effective one-size-fits-all and expensive targeted individual-based approaches.

2 RELATED WORK

To combat poverty, one should find root causes of issues and all related factors that contribute to spreading poverty, such as health issues, lack of education, inaccessibility to public services, poor living conditions, lack of job opportunities, and so forth [3, 46]. To measure the degree of poverty, several works proposed to use the poverty indices, such as Human Poverty Index (HPI) [7], Multidimensional Poverty Index (MPI) [3, 4], and so forth. However, a specific poverty index covers only some aspects of population, which might not include the actual root cause predictors [4, 46]. Additionally, different areas typically have different issues and root causes of poverty [14, 40].

To infer which factors are the main causes of poverty in an area from population data, the first step is to find the associations between a dependent variable (e.g., poverty index) and independent variables (potential predictors). This is because the causal inference requires association relations among variables to be known before we can find causal directions [37, 38]. One of the approaches that is widely used in social science to find association relations is regression analysis [35]. It has been used as a part of framework in poverty index analysis in [24, 36]. Linear regression has been extended and used in various types of data beyond linear-static data, such as time-series data [9, 19, 48], non-linear data [6, 27], and so forth.

In this work, we are interested in finding variables that associate with population household incomes, which is one of the main components in several poverty indices [3, 4, 7].

However, resolutions of different areas (e.g., provinces vs. country levels) might have different predictors that are associated with household incomes. From the perspective of policy makers, placing the wrong policies to the areas might not be able to solve the issues. For example, urban
Fig. 1. Proposed framework overview. Given a dataset $S = \{\vec{y}, X\}$ and a set of multi-resolution clusters $C$ (Definition 3.1) as inputs, where a vector $\vec{y}$ contains realizations of dependent variables, and an $n \times l$ matrix $X$ contains realizations of independent variables, in the first step, the framework performs the parameter estimation (Section 4.1). Then, we use MDL (Section 4.2) and a cross-validation scheme (Section 4.3) to infer the maximal homogeneous partition in Section 4.4, which is the optimal-resolution partition that effectively represents population.

and rural areas have different issues of poverty [14]. Hence, it is important to find a way to select a model based on the right resolutions that truly represent issues of areas. According to Occam’s razor principle, we prefer a policy that is simple and covers issues of a larger population as an efficient policy. One of the model selection approaches that is widely used is Minimum Description Length (MDL) [22], which was introduced in the field of data analytics by [43]. It is used in a linear regression setting as a feature selection criterion to find the informative features [44]. Nevertheless, to the best of our knowledge, there is no framework that compares the regression models from different-area resolutions using MDL and asks which one is a better model as a predictor for household-income prediction.

2.1 Comparison with the Clustering Problem

Given a set of multi-resolution partitions as an input to our framework along with the population data (realizations of dependent and independent variables), we propose a framework that is able to infer a subset of multi-resolution partitions (Figure 1), which is a set of largest partitions s.t. each partition having a model that represents populations well in the terms of predictive performance (see Section 4.3). In contrast, in the clustering problem, the main goal is to find a set of clusters or partitions that is fitting the population data well w.r.t. some similarity function. Typically, the inputs of clustering are the number of clusters and/or some threshold(s) that define clusters and population data.

Hence, in a dataset that has only population data without a set of multi-resolution partitions, we can use some data clustering approach to infer the hierarchical clusters (multi-resolution partitions). Then, our proposed framework can be used to find the optimal partition in the hierarchical clusters that covers the entire population s.t. the partition models representing the population well in regard to prediction of the target dependent variable.

Policy makers might consider each cluster in the optimal partition from our framework as a group of individuals that shares the same indicators that can predict the target variable well. If policy makers want to change the value of target variable in the population, then they might make a policy for each cluster in our optimal partition instead of either making a single one-size-fits-all policy or having a single unique policy for each individual.

2.2 Comparison with Mixture Models

The concept of mixture models has been widely used for modeling complex distributions by using several mixture components for representing “subdistributions” or “subpopulations” [33].
Generally, an individual subdistribution is modeled using a “simpler” distribution, e.g., Gaussian distribution. Given a specific number of mixture components and a distribution type of mixture components, a finite mixture model will estimate the degree of membership of an individual datapoint to all mixture components as well as parameters of the components. Datapoints belong to the same partition if they have the highest membership degree to the same mixture component. Mixture models can be generalized for regression problems [16, 50]. Particularly, a regression function is used, instead of a distribution function, for representing a mixture component. Datapoints belonging to the same component would share a similar “behavior” according to the regression function and its variables.

In contrast to traditional data clustering techniques, a mixture model can form a data partition without any similarity function defined. The model, however, requires an assumption of subpopulations that are distributed or characterized according to a certain class of distributions or regression functions. Ideally, the mixture model framework would naturally fit to our problem if a set of policies is intended to apply to subpopulations, which can be clearly defined by a certain class of distributions or regression functions. Since it is not straightforward to impose structural constraints such as administrative regions (e.g., a pair of administrative regions at the same level cannot belong to the same subpopulation) into the inference algorithm, one can simply let the mixture model to flexibly “search” for subpopulation partitions and subsequently impose the administrative constraints if some partition violates such constraints. Our approach, however, explicitly takes the structural constraints into account by “searching” and evaluating the data partition along the administrative hierarchy. Consequently, the data partition from our approach always complies with the constraints and requires no additional step—the constraint checking—as in the mixture model case.

Table 1 briefly illustrates the difference between mixture of regression (MR) and our proposed framework. Both approaches assign individuals to a partition. While MR requires the number of clusters (components) as a parameter, our approach uses the multi-resolution partitions as an input. Additionally, instead of only assigning individuals to a partition, our approach can also evaluate how strong the individuals in the same cluster share the same model (see Section 4.3). Moreover, the time complexity of our approach is bounded for the exact solution, while the bound of mixture approach is unknown; however, there is an algorithm [31] that can provide the approximate solution with the bound $O(Nd)$, where $N$ is the number of individuals and $d$ is the number of dimensions.

### 2.3 Our Contributions

To fill the gaps mentioned above, in this article, we aim at formalizing the multi-resolution model selection problem in regression analysis using the MDL principle and propose the framework (Figure 1) as a solution of our model selection problem. The proposed framework is capable of doing the following:
Identifying Linear Models in Multi-Resolution Population Data

Table 2. Notations and Symbols

| Term and notation | Description |
|-------------------|-------------|
| n                 | Number of individuals |
| d                 | Number of independent variables |
| l                 | Number of parameters |
| D = \{1, \ldots, n\} | Set of individual indices |
| C                 | Cluster where \( C \subseteq D \) |
| \( C = \{C_{j,k}\} \) | Set of multi-resolution clusters where \( C_{j,k} \) is the \( j \)th cluster in the \( k \)th layer |
| \( C^* = \{C\} \) | Maximal homogeneous partition |
| X, x              | Independent variable and its realization |
| Y, y              | Dependent variable and its realization |
| S = \{(x_1, y_1), \ldots, (x_n, y_n)\} | Dataset containing realizations of X and Y |
| \( L(h) \)        | Number of bits of representation of a model \( h \) |
| \( L(S|h) \)      | Number of bits of representation of dataset \( S \) given a model \( h \) |
| \( L_R(x) \)      | Number of bits of representation of real-number vector \( x \) |
| \( \gamma \)      | Homogeneity threshold |
| \( \mathcal{H}, \mathcal{H}_1, \mathcal{H}_2 \) | Function class, set of models, and hypothesis space |
| \( I(C', H_1, H_2) \) | Model Information Reduction Ratio, where \( C \) is a partition of \( D \) |
| \( I(C'_1, C'_2, H) \) | Cluster Information Reduction Ratio, where \( C'_1 \) and \( C'_2 \) are partitions of \( D \) |

- **Inferring the optimal-resolution partition**: inferring the best-fitting partition that effectively represents population in regard to model prediction and MDL;
- **Inferring the optimal model for each area**: inferring the best-fitting model among candidate functions given area population data as well as choices of functions in a linear function class.

After formalizing the new problem, we provide an algorithm that guarantees finding the optimal partition, which is the output of the new problem. We evaluate our approach performance and demonstrate its application using both simulation and real-world datasets of Thailand’s population households. Our framework can be generalized beyond the context of poverty analysis.

### 3 PROBLEM FORMALIZATION

Given a vector \( \mathbf{y} = (y_1, \ldots, y_n) \), an \( n \times d \) matrix \( \mathbf{X} = (x_1, \ldots, x_n)^T \), and a set of multi-resolution clusters \( C = \{C_{j,k}\} \) (Definition 3.1), \( y_i \in \mathbb{R} \) is a realization of dependent random variable \( Y \) of individual \( i \), \( \mathbf{X}(i,j) \in \mathbb{R} \) is a realization of independent random variable \( X_j \) of individual \( i \), and \( C_{j,k} \) is the \( j \) cluster in the \( k \)th layer of the multi-resolution cluster set, where individual \( i \) is a member of \( C_{j,k} \). In this work, the objective is to solve Problem 2, where a hypothesis space \( H \) is a linear function class. The details of definitions and the problem formalization can be found later in this section.

Specifically, in Section 3.1, we explore the concept of MDL. Then, in Section 3.2, we provide definitions as well as the formalization of Problem 2, which is the main problem we aim to solve in this article. In Section 3.3, we provide the properties of a linear model in the MDL setting. We also provide Table 2 as a reference for the notations and symbols used in this article.

### 3.1 MDL on Regression Models

Given a set of observation data below:

\[
S = \{(x_1, y_1), \ldots, (x_n, y_n)\},
\]

(1)
where \((x_i, y_i)\) are realizations of random variables \((X_i, Y_i)\) s.t. \(x_i \in \mathbb{R}^d\) and \(y_i \in \mathbb{R}\). The random variables \(X_1, \ldots, X_n\) are i.i.d. with an unknown distribution \(P_X\) or \(X_i \sim P_X\). The random variable \(Y_i\) has a relation with \(X_i\) as given below:

\[
Y_i = h^*(X_i) + \epsilon,
\]

where \(h^*(X_i)\) is an unknown function that is a member of some hypothesis space \(\mathbb{H}\) and \(\epsilon\) is a noisy constant s.t. \(X_i\) and \(\epsilon\) are statistically independent or \(X_i \perp \epsilon\).

In MDL, we would like to find the number of bits of the shortest representation that we can have for any given data based on some hypothesis space.

\[
\mathcal{L}(S|\mathbb{H}) = \min_{h \in \mathbb{H}} \mathcal{L}(h) + \mathcal{L}(S|h),
\]

where \(\mathcal{L}(h)\) is a number of bits of representation for a predictive model \(h\) and \(\mathcal{L}(S|h)\) is a number of bits of representation for data \(S\) given the predictive model \(h\). In our case, \(\mathcal{L}(h)\) is the number of bits we need to encode parameters and related information of function \(h\). \(\mathcal{L}(S|h)\) is the number of bits we need to encode \(S\) using \(h\).

\[
\mathcal{L}(h) = \sum_{j=1}^{l} \mathcal{L}_\mathbb{R}(p_j),
\]

\[
\mathcal{L}(S|h) = \sum_{i=1}^{n} \mathcal{L}_\mathbb{R}(x_i) + \sum_{i=1}^{n} \mathcal{L}_\mathbb{R}(y_i - h(x_i)),
\]

where \(l\) is the number of parameters of \(h\), \(p_j\) is the \(j\)th parameter of \(h\), \(\mathcal{L}_\mathbb{R}(x)\) is a function that returns the number of bits we need to encode a real number \(x\), and \(n\) is the number of data points in \(S\).

The model complexity in Equation (3) can be controlled by \(\mathcal{L}(h)\). If the two models perform the same, then Equation (3) chooses the less complex model that has lower \(\mathcal{L}(h)\) as a solution. See Section 7.2 for more discussion regarding how model complexity can be implemented and used to compare functions from different classes.

In regard to sparsity regularization, we can modify \(\mathcal{L}(h)\) to add the higher bits for a model representation (penalty) when the model \(h\) has a higher number of parameters (non-zero coefficients). See Section 7.3 for more discussion.

In this article, we make the following assumption.

**Assumption 1.** For any \(x_1, x_2 \in \mathbb{R}\) s.t. \(|x_1| \geq |x_2|\), we have \(\mathcal{L}_\mathbb{R}(x_1) \geq \mathcal{L}_\mathbb{R}(x_2)\).

We make Assumption 1 to represent the fact that a larger number contains a similar amount of or more information that we need to encode. In the function \(\mathcal{L}(S|h)\), there are two parts: encoding \(x\) or \(\sum_{i=1}^{n} \mathcal{L}_\mathbb{R}(x_i)\) and encoding \(y\) by \(h \in \mathbb{H}\) or \(\sum_{i=1}^{n} \mathcal{L}_\mathbb{R}(y_i - h(x_i))\). If \(h\) predicts \(y_i\) using \(x_i\) perfectly, then we expect to encode no information of \(y_i\), which makes \(\mathcal{L}_\mathbb{R}(y_i - h(x_i)) = 0\) bits. However, if \(h' \in \mathbb{H}\) makes a larger error of predicting \(y_i\) from \(x_i\) than \(h\), we expect to use more bits of encoding for \(y_i\) using \(h'\) and \(x_i\) than \(h\). Hence, Assumption 1 naturally represents this intuitive notion of having more bits of encoding for a function \(h'\) that makes more errors of predicting \(y_i\) from \(x_i\).

Now, we are ready to formalize the MDL-function inference problem.

**Problem 1:** MDL-function inference problem

**Input:** A set \(S = \{(x_1, y_1), \ldots, (x_n, y_n)\}\), and a hypothesis space \(\mathbb{H}\)

**Output:** An optimal MDL-function \(h^* = \arg\min_{h \in \mathbb{H}} \mathcal{L}(h) + \mathcal{L}(S|h)\)
3.2 Model Selection on Multi-Resolution Clusters

Given a set of individual indices \( D = \{1, \ldots, n\} \), where \( i \in D \) refers to the \( i \)th individual in a population, we can define a set of multi-resolution clusters as given below.

**Definition 3.1 (Multi-Resolution Cluster Set (MRC set)).** A set of multi-resolution clusters \( C = \{C_{j,k}\} \) of \( n_c \) layers, where \( k \in [1, n_c] \) and \( C_{j,k} \subseteq D \) is the \( j \)th cluster in the \( k \)th layer, has the following properties:

- If \( k > 1 \), then \( \forall C_{j,k}, \exists C_{j',k-1}, C_{j,k} \subseteq C_{j',k-1} \).
- All individuals belong to some cluster in each layer or \( \forall k, \bigcup_j C_{j,k} = D \).
- Clusters in the same layer are disjoint or \( \forall k, \bigcap_j C_{j,k} = \emptyset \).

An example of multi-resolution cluster set is a national administrative division of regions in a country where the first layer is the national level, the second layer is the province level, and the last layer is the village level. Each cluster in each layer is a specific group of individuals who are governed by the same local government at a particular level. For instance, at the province level, each cluster \( C \) represents citizens or households within a province \( C \). Next, we have a concept of clusters that all members share the same joint distribution \( P_{X,Y} \).

**Definition 3.2 (Homogeneous Cluster).** Given a cluster \( C \subseteq D \), and a set of population data \( S_C = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), where \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \). A cluster \( C \) is a homogeneous cluster of \( S_C = \{(x_i, y_i) | i \in C\} \), where \( x_i, y_i \) are realizations of \( X_i, Y_i \) respectively, if the following conditions hold:

1. \( \exists P_X, \forall i \in C, X_i \sim P_X \).
2. \( Y_i = h^*(X_i) + \varepsilon \) for some unknown function \( h^* \).

Here, \( \varepsilon \in \mathbb{R} \) is a noise random variable from some unknown truncated distribution with the bound \( [a, b] \) s.t. \( \varepsilon^* = \max(|a|, |b|) \), \( E[\varepsilon] = 0 \), and \( |\varepsilon| \leq \varepsilon^* \). All subsets of \( C \) are also homogeneous clusters.

The concept of homogeneous cluster represents a specific area (e.g., province, village) that shares a common relationship between a dependent variable and independent variables. For instance, a village \( A \) has the health issue that affects household incomes of all villagers. Suppose \( Y \) is a dependent variable of income and \( X \) is an independent variable of degree of health-issue severeness. If a village \( A \) is a homogeneous cluster, then \( |y_i| \propto |x_i| \) for any household \( i \) in this village. Because all individual households share the similar properties, policy makers should make a single set of policies to support the entire population within a homogeneous cluster.

Next, we need the concept of a cluster set that covers all individuals.

**Definition 3.3 (MRC Partition).** Given a multi-resolution cluster set \( C \), a set of clusters \( C_p \) is an MRC partition if \( C_p \subseteq C \), all clusters in \( C_p \) are disjoint, and \( \bigcup_{C_i \in C_p} C_i = D \).

An MRC partition represents a set of administrative divisions that cover all individuals within a country. A set of all provinces is an example of the MRC partition.

However, we need the MRC partition concept, which can include the subareas from mixed layers of the MRC set but still covers the entire population. This is because some provinces might be homogeneous clusters that share the same properties, while, in other provinces, we need to break down to the village layer to get the homogeneous clusters. Our goal is to use the concept of MRC partition to define a set of maximal homogeneous clusters (largest possible homogeneous cluster for a specific area) that cover the entire population.

**Definition 3.4 (Maximal Homogeneous Cluster).** Given a multi-resolution cluster set \( C \), and a set of population data \( S_C = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), where \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \), a cluster \( C^* \in C \) is a
maximal homogeneous cluster if $C$ is a homogeneous cluster, and there is no other homogeneous cluster $C'$ s.t. $C' \in C$ and $C^* \subset C'$.

The number of elements within an MRC partition might represent the number of sets of policies that policy makers should consider making in order to eliminate common issues. Next, we need to define a concept to call datasets that have the maximal homogeneous partition, which is the MRC partition that contains only maximal homogeneous clusters as members.

**Definition 3.5 (Multi-Resolution-Bivariate Set (MRB)).** Let $C$ be a multi-resolution cluster set, and $S_C$ be a set of $C$’s population data. $S_C$ is a multi-resolution-bivariate set if $S_C$ satisfies the following properties:

- There exists an MRC partition of homogeneous clusters, $C^* = \{C\}$, s.t. for each $C \in C^*$, $C$ being a maximal homogeneous cluster. We call $C^*$ as the “Maximal Homogeneous Partition”.
- Each homogeneous $C_i \in C^*$ has a separate function $h^*_i$ that might be different from any other function $h^*_j$ of $C_j \in C^*$, where $C_i \neq C_j$.

Suppose we have an MRB set $S_C = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and a function class $H$ (e.g., linear class, polynomial class, convex class). We can use the data in each cluster $C_{j,k}$ to estimate the parameters of function in $H$ to build a set of models $H_{j,k} \subset H$. Each model $h \in H_{j,k}$ has parameter values estimated from the data points $\{(x_i, y_i)\}$ in the cluster $C_{j,k}$. Let $\mathcal{H} = \bigcup H_{j,k}$ be a set of models of $C$, then we can formalize our new computational problem: MDL-function inference on multi-resolution data problem (Problem 2).

**PROBLEM 2:** MDL-function inference on multi-resolution data problem

**Input:** A set of individual indices $D = \{1, \ldots, n\}$, a multi-resolution cluster set $C = \{C_{j,k}\}$, an MRB set $S_C = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and a function class $H$

**Output:** A set of models $\mathcal{H}$ w.r.t. $H$ and $C$, a maximal homogeneous partition $C^* = \{C\}$, and a set of optimal-MDL functions $H^*_C = \{h^*_i\}$ where each $h^*_i \in H^*$ is the optimal MDL-function that belongs to a specific cluster $C \in C^*$ using the hypothesis space as $H$.

### 3.3 Linear Function Properties

Suppose we have $H$ as a linear function class, and $C'$ as an MRC partition of $D$. We can have a measure of how well the models derived from $C'$ represent the set of population data $S$ in regard to MDL as given below:

$$
\mathcal{L}(C', H) = \sum_{C \in C'} \left( \mathcal{L}(H)_C + \mathcal{L}(C|H) \right),
$$

\hspace{1cm} (6)

$$
\mathcal{L}(H)_C = \mathcal{L}_\mathbb{R}(\hat{\beta}_C^1) + \mathcal{L}_\mathbb{R}(\hat{\beta}_C^d),
$$

\hspace{1cm} (7)

$$
\mathcal{L}(C|H) = \sum_{i \in C} \mathcal{L}_\mathbb{R}(x_i) + \sum_{i \in C} \mathcal{L}_\mathbb{R}(y_i - \hat{\beta}_1^C - x_i^T \hat{\beta}_C^d),
$$

\hspace{1cm} (8)

where $\hat{\beta}_C^1 \in \mathbb{R}$ and $\hat{\beta}_C^d \in \mathbb{R}^d$ are the estimators of linear function parameters based on the data-points in $C$. We have the next proposition to show that the sum of encoding bits of residuals from homogeneous clusters is less than the sum of encoding bits of residuals of the function whose parameters are approximated by the union of these homogeneous clusters.

**Proposition 3.6.** Let $C = \{C_1, \ldots, C_k\}$ be a set of homogeneous clusters of $S_C$, but the cluster $C' = \bigcup_{i \in C} C_i$ is not a homogeneous cluster and $H$ is a linear function class. Assume that the functions that generate $Y$ from $X$ in $S_C$ for each cluster $C_i \in C$ are in $H$. 

ACM Transactions on Knowledge Discovery from Data, Vol. 15, No. 2, Article 15. Publication date: December 2020.
If \( g' \) is a function whose parameters are approximated by \( C' \) and \( g_1, \ldots, g_k \) are functions whose parameters are approximated by \( C_1, \ldots, C_k \), respectively, where \( g' \) and \( g_1, \ldots, g_k \) are in \( H \), we have
\[
\sum_{i \in C'} L_R(y_i - g'(x_i)) \geq \sum_{j=1}^{k} \sum_{i \in C_j} L_R(y_i - g_j(x_i)).
\]

**Proof.** Because each cluster in \( C \) is homogeneous and the relation between \( Y \) and \( X \) is \( Y_i = \beta_1 + X_i^T \beta + \epsilon \). We can approximate \( \beta_1 \) and \( \beta \) using some estimator approach. Suppose \( \hat{\beta}_1, \hat{\beta} \) are estimators for \( \beta_1, \beta \), then we have
\[
\sum_{j=1}^{k} \sum_{i \in C_j} L_R(y_i - g_j(x_i)) = \sum_{j=1}^{k} \sum_{i \in C_j} L_R(y_i - \hat{\beta}_1, \beta - x_i^T \hat{\beta}_1) \leq |C'| L_R(\epsilon^*).
\]

In contrast, if two clusters \( C_1, C_2 \in C \) have different functions \( g_1, g_2 \), for example, \( g_1(X) = -X_1^T \beta + \epsilon \) while \( g_2(X) = -g_1(X) \), then by approximating \( \beta_1, \beta \) for \( g'(X) \) using datapoints from both \( C_1, C_2 \), the sum of residuals of datapoints from both clusters must be higher than \( \epsilon^* \). Precisely, according to Definition 3.2, \( |\epsilon| \leq \epsilon^* \) for any homogeneous cluster. Suppose \( g_1 \neq g_2 \) and \( C_1, C_2 \) are homogeneous clusters of \( g_1, g_2 \), respectively. If we approximate the parameters from datapoints of both \( C_1 \) and \( C_2 \), then we would have \( g' \) that is not the same as both \( g_1, g_2 \) but \( g' \) is something between \( g_1 \) and \( g_2 \). For \( g' \), the sum of encoding bits of residuals of \( C_1 \) is \( \sum_{i \in C_1} L_R(y_i - g'(x_i)) \leq |C_1| L_R(\epsilon^*) + \sum_{i \in C_1} L_R(g'(x_i) - g_1(x_i)) \), where the term \( \sum_{i \in C_1} L_R(g'(x_i) - g_1(x_i)) > 0 \) since \( g' \neq g_1 \). The same applies for \( C_2 \).

Hence, the sum of errors using \( g' \) to predict \( y \) must be greater than \( |C'| L_R(\epsilon^*) \).

Therefore, we have
\[
\sum_{j=1}^{k} \sum_{i \in C_j} L_R(y_i - g_j(x_i)) \leq |C'| L_R(\epsilon^*) < \sum_{i \in C'} L_R(y_i - g'(x_i)).
\]

Hence, \( \sum_{i \in C'} L_R(y_i - g'(x_i)) \geq \sum_{j=1}^{k} \sum_{i \in C_j} L_R(y_i - g_j(x_i)) \).

\[\square\]

We have the next theorem to state that the maximal homogeneous partition has the lowest number of encoding bits compared to any other MRC partitions.

**Theorem 3.7.** Let \( D \) be a set of individual indices, \( C \) be a multi-resolution cluster set, \( S_C \) be an MRB set, \( H \) be a linear function class, and \( C^* \) be the maximal homogeneous partition of \( S_C \). For any \( C' \) that is an MRC partition, \( L(C^*, H) \leq L(C', H) \).

**Proof.** Suppose \( L(C^*, H) \neq L(C', H) \). Because both \( C^* \) and \( C' \) are MRC partitions, all cluster members in both partitions are also members of the MRC set \( C \). It is obvious that \( L(C^*, H) \neq L(C', H) \) because some clusters between \( C' \) and \( C^* \) are different.

Let \( A = C^* \cap C' \). There are some regions \( C^* - A \) that make both partitions different. There are two cases that we need to consider.

Case 1: Suppose \( C \in C^* - A \) and \( C' = \{C_1, \ldots, C_k\} \subseteq C^* - A \), where \( \bigcup_{C_i \in C'} C_i = C \). Since \( C \) is a homogeneous cluster, the subsets of \( C \), which are \( \{C_1, \ldots, C_k\} \), must be the homogeneous clusters from the same \( X \sim P_X \) and \( Y = h(X) + \epsilon \). Because \( \bigcup_{C_i \in C'} C_i = C \) implies that we estimate the parameters of \( h \) from the same region, \( \sum_{C_i \in C'} L(S_{C_i}|H) \approx L(S_C|H) \).

\[
L(C', H) - L(C^*, H) = \sum_{C_j \in C'} (L(S_{C_j}|H) + L(H|C_j)) - L(S_C|H) - L(H).\]

Since \( H \) is a linear class and all homogeneous clusters in this case share the same \( h(X) \), then \( L(H)_C \approx L(H|C_j) \).

\[
L(C', H) - L(C^*, H) = (|C'| - 1)L(H)_C > 0.
\]
Case 2: Suppose $C = \{C_1, \ldots, C_k\} \subseteq C^* - A$ and $C' \subseteq C^* - A$, where $\bigcup_{C_i \in C} C_i = C'$. Since $C^*$ is a maximal partition, $C'$ cannot be a homogeneous cluster. In fact, different homogeneous clusters might have different functions $h^*$, where $Y_i = h^*(X_i) + \epsilon$. Let $g'$ be a model whose parameters are approximated by $C'$ and $g_1, \ldots, g_k$ be models whose parameters are approximated by $C_1, \ldots, C_k$, respectively. Again, we assume that $L(H)_{C'} \approx L(H)_{C_i}$.

$$L(C', H) - L(C^*, H) = L(S_{C'}|H) + L(H)_{C'} - \sum_{C_i \in C} (L(S_{C_i}|H) + L(H)_{C_i})$$

$$= \sum_{i \in C'} L_R(y_i - g'(x_i)) - \sum_{j=1}^k \sum_{i \in C_j} \left( L_R(y_i - g(x_j)) \right) - (|C'| - 1)L(H)_{C'}.$$

Typically, the number of parameters of linear functions is less than the number of individuals. Hence, it is safe to assume that $(|C'| - 1)L(H)_{C'} \ll \sum_{i \in C'} L_R(y_i - g'(x_i))$. According to Proposition 3.6, we have $\sum_{i \in C'} L_R(y_i - g'(x_i)) \geq \sum_{j=1}^k \sum_{i \in C_j} L_R(y_i - g(x_j))$. Hence,

$$L(C', h) - L(C^*, h) = \sum_{i \in C'} L_R(y_i - g'(x_i)) - \sum_{j=1}^k \sum_{i \in C_j} L_R(y_i - g(x_j)) \geq 0.$$

In both cases, $L(C', h) - L(C^*, h) \geq 0$. Therefore, $L(C^*, h) \leq L(C', h)$. \hfill $\square$

## 4 METHODS

In this section, we provide the details of the method that is used to solve Problem 2. Figure 1 gives an overview of our proposed framework. In the first step, the framework performs the parameter estimation (Section 4.1). Then, we use MDL (Section 4.2) and a cross-validation scheme (Section 4.3) to infer the maximal homogeneous partition in Section 4.4. We provide the time complexity of our method in Section 4.5.

### 4.1 Parameter Estimation

In linear models, to minimize the magnitude of residuals of prediction, the optimal $h^*$, which minimizes the residuals, can be inferred using the least-squares technique. Given $H$ is a linear function class, we have an optimization problem using the least-squares loss as given below:

$$h^* = \arg\min_{h \in H} \sum_{i=1}^n (y_i - h(x_i))^2. \quad (9)$$

The linear function is of the form

$$h(x_i) = \tilde{x}_i^T \hat{\beta} + \epsilon,$$

where $\tilde{x}_i$ is a vector of $x_i$ s.t. $\tilde{x}_i(1) = 1$ and $\tilde{x}_i(j) = x_i(j - 1)$, and $\epsilon \sim N(0, \tau^*)$ is a noisy value, where $\tau^*$ is an unknown variance. Hence,

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^{d+1}} \sum_{i=1}^n (y_i - \tilde{x}_i^T \beta)^2.$$  \quad (11)

We call $\hat{\beta}$ an ordinary-least-squares (OLS) estimator of $\hat{\beta}$. Suppose all columns in $X$ are linearly independent, then the optimal solution for Equation (11) can be solved by the closed-form equation as follows:

$$\hat{\beta} = \left( X_1^T X_1 \right)^{-1} X_1^T \tilde{y},$$  \quad (12)

where $X_1$ is an $n \times d + 1$ matrix s.t. the first column of $X_1$ containing only 1s, and for any other column $j > 1$, $X_1(i, j) = X(i, j - 1)$. 

ACM Transactions on Knowledge Discovery from Data, Vol. 15, No. 2, Article 15. Publication date: December 2020.
In this step, for each cluster \( C \in C \), we estimate \( \hat{\beta}_C \) using data points in cluster \( C \) \( ((x_i, y_i) \text{ s.t. } i \in C) \).

### 4.2 Minimum-Description-Length Measure of Models

To estimate \( L(C', H) \) in Equation (6), we need to compute \( L(H)_{C'} \) in Equation (7) and \( L(C|H) \) in Equation (8), which both require the estimation of \( L_{\mathbb{R}}(x) \).

Suppose we have two integer numbers \( y_1 \) and \( y_2 \), where \( |y_1| \ll |y_2| \). When we compress \( y_1, y_2 \) into a computer memory space, because the magnitude of \( y_1 \) is much smaller than \( y_2 \), we can approximately represent \( y_1 \) without using, say, all \( q \) bits needed to represent \( y_2 \).

For integer numbers, we need at least \( L_{\mathbb{R}}(y) = \lceil \log_2|y| \rceil + 1 \) bits to represent \( y \) in a binary representation, where \(+1\) bit is for the sign representation of \( y \). For a real number representation, the work in [32] provided the framework to implement the universal coding for real numbers s.t. for any \( y \in \mathbb{R} \), \( |E(y)| \approx |\log_2|y|| \), where \( |E(y)| \) is a number of bits of real-number representation in [32]. Hence, for simplicity, for any real number \( y \), we can approximately calculate \( L_{\mathbb{R}}(y) \) as given below:

\[
L_{\mathbb{R}}(y) = \begin{cases} 
\lceil \log_2|y| \rceil + 1, & |y| \geq 1 \\
1, & \text{otherwise}.
\end{cases}
\]  

(13)

If \( Y \) is a vector or matrix, then \( L_{\mathbb{R}}(Y) = \sum_i L_{\mathbb{R}}(y_i) \), where \( y_i \) is a real number element in \( Y \).

#### 4.2.1 Comparing Two Linear Models on the Same Datapoints

Given \( H \) is a linear function class and \( H_1, H_2 \subset H \), where \( H_1 \neq H_2 \), to compare \( H_1 \) and \( H_2 \) using \( L(C', H) \) in Equation (6) whose parameters are estimated from the same set of datapoints and the same set of clusters \( C' \), as well as considering that \( \sum_{i \in C} L_{\mathbb{R}}(x_i) \) in both \( L(C|H_1) \) and \( L(C|H_2) \) are the same, we have the following relation:

\[
L(C', H_1) - L(C', H_2) \propto \sum_{C' \in C'} \left( \sum_{i \in C} L_{\mathbb{R}}(y_i - h_1^C(x_i)) - \sum_{i \in C} L_{\mathbb{R}}(y_i - h_2^C(x_i)) + (L(H_1)_{C'} - L(H_2)_{C'}) \right),
\]

where \( h_1^C(x_i) \in H_1 \) is a linear function whose \( \hat{\beta} \) is estimated using datapoints in \( C \), and the same applies for \( h_2^C(x_i) \in H_2 \). The normalization of the equation given above is as follows:

\[
I(C', H_1, H_2) = \frac{\sum_{C' \in C'} \left( \sum_{i \in C} L_{\mathbb{R}}(y_i - h_1^C(x_i)) - \sum_{i \in C} L_{\mathbb{R}}(y_i - h_2^C(x_i)) + (L(H_1)_{C'} - L(H_2)_{C'}) \right)}{\sum_{C' \in C'} \left( \sum_{i \in C} L_{\mathbb{R}}(y_i - h_1^C(x_i)) + L(H_1)_{C'} \right)},
\]

(14)

where \( I(C', H_1, H_2) \in [-\infty, 1] \), and \( L(H_1)_{C'} \) and \( L(H_2)_{C'} \) are the numbers of bits needed for the compression of parameters of a function that is approximated by datapoints in \( C \) using \( H_1 \) and \( H_2 \), respectively. We call \( I(C', H_1, H_2) \) the "Model Information Reduction Ratio". The function \( I(C', H_1, H_2) \) represents how well we can reduce the size of our data using \( H_2 \) instead of \( H_1 \). For example, if \( I(C', H_1, H_2) = 0.5 \), it implies that \( H_2 \) reduces the size of space we need to encode the data by 50% compared to the size we need to encode the same data using \( H_1 \). In contrast, if \( I(C', H_1, H_2) = -0.4 \), it means using \( H_1 \) is still a better option since \( H_2 \) increases the size of space we need to encode the same data by around 40%.

#### 4.2.2 Comparing Two Sets of Clusters on the Same Datapoints

Given \( H \) is a linear function class, and \( C'_1 \) and \( C'_2 \) are two sets of clusters where they cover the same datapoints or \( \bigcup_{C' \in C'_1} = \bigcup_{C' \in C'_2} \). Assuming that \( L(H)_{C'} \) for each cluster have the same size \( c_H \), we can have "Cluster Information..."
Reduction Ratio $I(C'_1, C'_2, H)$ as given below:

$$I(C'_1, C'_2, H) = \frac{\sum_{i \in C'_1} \left( \sum_{i \in C_1} L_{\mathbb{R}}(y_i - h^{C_1}(x_i)) \right) - \sum_{i \in C'_2} \left( \sum_{i \in C_2} L_{\mathbb{R}}(y_i - h^{C_2}(x_i)) \right) + c_H(|C'_1| - |C'_2|)}{\sum_{i \in C'_1} \left( \sum_{i \in C_1} L_{\mathbb{R}}(y_i - h^{C_1}(x_i)) \right) + c_H|C'_1|},$$

where $h^{C_1}$ and $h^{C_2}$ are functions whose parameters are approximated using datapoints in clusters $C_1$ and $C_2$, respectively. The function $I(C'_1, C'_2, H) \in [-\infty, 1]$. $I(C'_1, C'_2, H)$ represents how well the different sets of clusters can encode the same information. For example, if $I(C'_1, C'_2, H) = 0.5$, it implies that $C_2$ reduces the size of space we need to encode the data by 50% compared to the size we can encode the same data using $C_1$. In contrast, if $I(C'_1, C'_2, H) = -0.4$, it means using $C_1$ is still a better option since $C_2$ increases the size of space we need to encode the same data by around 40%.

### 4.3 Homogeneity Measure of Clusters

In order to infer the degree of homogeneity of clusters, we deploy the cross-validation concept. It is typically used as a methodology to measure the generalization of models. If the model is generalized well, then it is capable of performing well in any given dataset outside of training data. It is typically used as a methodology to measure the generalization of models. If the model is generalized well, then it is capable of performing well in any given dataset outside of training data.

In order to infer the degree of homogeneity of clusters, we deploy the cross-validation concept. It is typically used as a methodology to measure the generalization of models. If the model is generalized well, then it is capable of performing well in any given dataset outside of training data. It is typically used as a methodology to measure the generalization of models. If the model is generalized well, then it is capable of performing well in any given dataset outside of training data.

$$\eta(C) = \frac{\sum_{C' \subset C} \left( \text{corr}(y_{C'}, h_{C-C'}(x_{C'})) \right)^2}{||C' : C' \subset C||},$$

where $h_{C-C'}$ is a function whose parameters are approximated using data points in $C - C'$, and $\text{corr}(y_{C'}, h_{C-C'}(x_{C'}))$ is a correlation between true values of $y$ in $C'$ and the predicted values of $y$ using $x'_{C'}$ in $C'$ by $h_{C-C'}$.

We estimate $h_{C-C'}$’s parameters using datapoints from the rest of subset clusters in $C$ except $C'$, then use $h_{C-C'}$ to predict $y$ using datapoints of $x$ in $C'$. If $C$ has no subsets in a multi-resolution cluster set $C$, then we can use a 10-fold cross-validation scheme to generate subsets of $C$ by uniformly separating members of $C$ into 10 subsets of equal size.

Note that Equation (10) in [11], which is used for cross-validation evaluation, is similar to $(\text{corr}(y_{C'}, h_{C-C'}(x_{C'})))^2$ in Equation (16). See [11] for more details regarding other indices that can be used to evaluate a linear model in a cross-validation scheme.

Next, we explore a theoretical property between $\eta(C)$ and homogeneity of clusters.

**Definition 4.1 ($\gamma$-Correlation-Linear-Learnable Dependency Relation).** Given a function of a set of datapoints $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^l$ is a realization of $X \sim P_X$, $y_i$ is a realization of $Y$, where $Y = h(X)$ s.t. $h(X)$ is a linear function, and a threshold $\gamma \in [0, 1]$. Let $S_1$ and $S_2$ be disjoint subsets of $S$. Assume that all dimensions in $x_i$ are linearly independent.

We say that $S_1$ and $S_2$ are $\gamma$-correlation-linear-learnable dependent if the following conditions are satisfied:

1. There exists the OLS estimator $\hat{\beta}$ of the linear function $g(x) = x^T \hat{\beta}$ s.t. $\hat{\beta}$ estimated using all datapoints from $S_1$ and $\text{corr}(y_{S_2}, g(x_{S_2})) \geq \gamma$, where $\text{corr}(y_{S_2}, g(x_{S_2}))$ is a correlation between the true values of $y_i$ in $S_2$ and predicted values of $y_i$ using $g(x)$ and $x_i$ in $S_2$.

2. The same must be true if we use $S_2$ for training the $g'$ function. We have $\text{corr}(y_{S_1}, g'(x_{S_1})) \geq \gamma$.

We denote $S_1 \approx_\gamma S_2$. 

ACM Transactions on Knowledge Discovery from Data, Vol. 15, No. 2, Article 15. Publication date: December 2020.
**Lemma 4.2.** A relation in Definition 4.1 is a dependency relation [1], which possesses the reflexive and symmetric properties.

**Proof.** First, we show that \( S_1 \cong_y S_1 \). Theorem 4 in [47] states that if all dimensions in \( x \) are linearly independent, then, using OLS and datapoints in \( S_1 \) to train a function \( g, y_i = g(x_i) \) for all \((x_i, y_i)\) in \( S_1 \). This implies \( \text{corr}(y_i, g(x_i)) = 1 \). Hence, the relation in Definition 4.1 is reflexive. For the symmetric property, for any \( S_1 \cong_y S_2 \), it implies \( S_2 \cong_y S_1 \) by definition. \( \square \)

By using \( \eta(C) \) in Equation (16), we can define the version of homogeneous cluster with the homogeneous degree property.

**Definition 4.3 (\( y \)-Homogeneous Cluster).** Given a set of individual indices \( D = \{1, \ldots, n\} \), and a set of population data \( S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), where \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \). A cluster \( C \subseteq D \) is a homogeneous cluster of \( S \) if \( \eta(C) \geq y \).

**Proposition 4.4.** Given \( C \) as a homogeneous cluster and a set of disjoint subsets \( C = \{C_1, \ldots, C_m\} \), where \( C_1 \subset C \) and \( \bigcup_{C_i \in C} C_i = C \). For any \( C_i \in C \), suppose \( S_{C_i} \) and \( S_{C-C_i} \) are sets of datapoints in \( C_i \) and \( C - C_i \), respectively. If \( S_{C_i} \cong_y S'_{C-C_i} \) for any \( C_i \in C \), then \( \eta(C) \geq y \), which implies \( C \) is a \( y \)-homogeneous cluster.

**Proof.** Since \( C \) is a homogeneous cluster, all datapoints \( x \sim X \in C \) are generated from the same distribution \( X \sim P_X \) and the relation \( Y = h(X) \) is the same for \( Y \) and \( X \) random variables in \( C \). For any \( C_i \in C \), because \( S_{C_i} \cong_y S'_{C-C_i} \), it implies that \( \text{corr}(y_{C_i}, g_{C-C_i}(x_{C_i})) \geq y \), where \( g_{C-C_i} \) is a function trained by datapoints in \( C - C_i \), and \( y_{C_i}, x_{C_i} \) are datapoints in \( C_i \). Therefore, by averaging all \( \text{corr}(y_{C_i}, g_{C-C_i}(x_{C_i})) \) from all \( C_i, \eta(C) \geq y \). \( \square \)

Proposition 4.4 suggests that if we know that all pairs of subsets \( C_i \) and \( C - C_i \) of \( C \) have a \( y \)-correlation-linear-learnable dependency relation, we can conclude that \( C \) is a \( y \)-homogeneous cluster. The dependency relation in Definition 4.1 can be checked by running the real data.

### 4.4 Inferring Maximal Homogeneous Partition

In this section, we propose the algorithm to solve Problem 2. To determine whether linear models provide an informative solution, we compare a linear model of each cluster \( C \) with the null model, which is the model that we predict \( y \) using the average of \( y, \bar{y} = \sum_{i \in C} y_i \). We use Equation (14) (Model Information Reduction Ratio \( I(C', H_1, H_2) \)) to compare linear and null models. We set \( C' = \{C\}, H_1 = \{h(x) = \bar{y}\}, H_2 = \{h(x) = \bar{y}' = \bar{x}_C^T \hat{\beta}_C\} \), where \( \hat{\beta}_C \) is estimated using datapoints in \( C \). Hence, \( \mathcal{L}(H_1)_C = \mathcal{L}(\bar{y}) \) and \( \mathcal{L}(H_2)_C = \mathcal{L}(\bar{y}' = \bar{x}_C^T \hat{\beta}_C) \).

Next, we show that our proposed algorithm (Algorithm 3) provides the solution for Problem 2.

**Proposition 4.5.** Given a multi-resolution cluster set \( C = \{C_{i,k}\} \), MRB set \( S_C \), and the threshold \( y = 0 \). Algorithm 3 always returns the maximal homogeneous partition.

**Proof.** Given \( S_C \) and \( C = \{C_{j,k}\} \), we prove that Algorithm 3 always provides a maximal homogeneous partition as an output. According to Theorem 3.7, the maximal homogeneous partition \( C^* \) always has \( \mathcal{L}(C^*, H) \) lower than or equal to that of any MRC partition. First, we show that Algorithm 3 provides an MRC partition.

In lines 4 and 5, the algorithm seeks the clusters from a top layer to a bottom one. It implies that if there exists a cluster \( C \in C^* \) in one of the above layers, then it is included in \( C^* \) before its subsets. The condition in line 7 prevents the algorithm from adding any subsets of the cluster members of \( C^* \); hence, all clusters in \( C^* \) are disjoint. For some cluster \( C \) that is not the last layer member of \( C \), either it is included in \( C^* \) in lines 8–11 or its subclusters are included in \( C^* \) for some later
**Algorithm 3:** Finding maximal homogeneous partition in linear function class

**Input:** A multi-resolution cluster set $C = \{C_{j,k}\}$, a set $S_C = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and a homogeneity threshold $\gamma$.

**Output:** A set of models $\mathcal{H}$, a maximal homogeneous partition $C^* = \{C\}$, a set of optimal-MDL functions $H^* = \{h^*\}$ of $C^*$, and the minimum $\eta$ of clusters $\gamma'$.

1. For $C_{j,k} \in C$ do
   1.1. Calculate $\hat{\beta}_{j,k}$ using Equation (12) on data points in $C_{j,k}$;
   1.2. Append $h(x)_{j,k} = \bar{x}^T \hat{\beta}_{j,k}$ to $\mathcal{H}$;
2. For $k \leftarrow 1$ to $n_c$ do
   2.1. For $C_{j,k} \in C$ do
      2.1.1. // $I\left(\{C_{j,k}\}, H_0, H_{lin}\right)$ is cached to be used later
      2.1.2. Calculate $I\left(\{C_{j,k}\}, H_0, H_{lin}\right)$ where $H_0 = \{h(x) = \bar{y}\}$, $H_{lin} = \{h(x) \in \mathcal{H}\}$;
      2.1.3. If there is no $C \in C^*$ s.t. $C_{j,k} \subset C$ then
         2.1.3.1. // $C_{j,k}$ is not at the last layer
         2.1.3.2. If $\exists C \in C, C \subset C_{j,k}$ then
            2.1.3.2.1. Let $C' = \{C\}$ where $C \subset C_{j,k}$;
            2.1.3.2.2. Calculate $I\left(C', \{C_{j,k}\}, H_{lin}\right)$;
            2.1.3.2.3. Append $C_{j,k}$ to $C'$ if $I\left(C', \{C_{j,k}\}, H_{lin}\right) > 0$ and $\eta(C_{j,k})_c \geq \gamma'$;
         2.1.3.3. // $C_{j,k}$ is at the last layer
      2.1.4. Else
         2.1.4.1. Append $C_{j,k}$ to $C^*$;
   2.2. End
   3. End
   4. For $C \in C^*$ do
      4.1. If $I\left(\{C\}, H_0, H_{lin}\right) > 0$ then
         4.1.1. Append $h(x)_C \in \mathcal{H}$ to $H^*$ as an optimal MDL-function of $C$ where $h(x)$ is $C$'s linear function;
      4.2. Else
         4.2.1. Append $h(x)_C = \bar{y}_C$ to $H^*$ as an optimal MDL-function of $C$ where $\bar{y}_C$ is a mean of $y_i$ in $C$;
   5. End
   6. End
   7. Set $\gamma' = \min_{C \in C^*} \eta(C)$;
   8. Return $H, C^*, H^*, \gamma'$;

The iteration of the loop. If no subclusters of $C$ are included until the last layer, then all subclusters of $C$ in the last layer are included in $C^*$ by default. Hence, $C^*$ covers all individuals because the union of the first-layer clusters must cover all individuals, and later-layer clusters are subsets of some first-layer clusters. Hence, $C^*$ is an MRC partition.

Second, we show that $C^*$ is the maximal homogeneous partition. In line 11, we include the cluster $C$ in $C^*$ only if $I\left(\{C_{1}, \ldots, C_k\}, \{C\}, H_{lin}\right) > 0$. For the homogeneous cluster $C$ and its subsets $C_1, \ldots, C_k$, by Case 1 in Theorem 3.7, $I\left(\{C_{1}, \ldots, C_k\}, \{C\}, H_{lin}\right) > 0$. Since our algorithm performs top-down searching, it always gets a homogeneous cluster $C$ in the highest possible layers before its subsets in line 11. Hence, Algorithm 3 provides the maximal homogeneous partition.

For completeness, given $C^*$ as any maximal homogeneous partition, we prove that there is only one possible unique $C^*$ that Algorithm 3 provides and no other $C'$ exists. Suppose $C_1^*, C_2^*$ are maximal homogeneous partitions of $S_C$ and $C = \{C_{j,k}\}$. Let $A = C_1^* \cap C_2^*$, and there are some clusters in $C_1^* - A$ that are different from clusters in $C_2^* - A$. Let us assume that $C \in C_1^* - A$ and
\{C_1, \ldots, C_k\} \subseteq C'_2 - A$, where $\bigcup C_i = C$. According to Case 1 in Theorem 3.7, because $C$ and its subsets are homogeneous, the length of encoding by $C$ must be smaller than using $\{C_1, \ldots, C_k\}$. Hence, $C'_2$ is not a maximal homogeneous partition, which is a contradiction!

Therefore, Algorithm 3 always returns the maximal homogeneous partition that is unique. □

The $\eta(C_j, k)_{cv} \geq \gamma$ condition in line 11 guarantees that all homogeneous clusters are $\gamma$-homogeneous clusters. By setting $\gamma$, Algorithm 3 provides the maximal homogeneous partition s.t. all homogeneous clusters being $\gamma$-homogeneous clusters. Otherwise, it provides a set of MRC partitions that contains the $\gamma$-homogeneous clusters from the highest possible layer that can be found and the clusters from the last layers of the population that have no $\gamma$-homogeneous clusters in any layer.

Let $C'$ be the homogeneous partition with some non-$\gamma$-homogeneous clusters generated by the algorithm the first time. We can have

$$\gamma' = \min_{C' \in C'} \eta(C').$$

(17)

After we set the threshold $\gamma = \gamma'$ and run Algorithm 3 for the second time, the result of the algorithm is the maximal homogeneous partition with all $\gamma'$-homogeneous clusters. This is true since we know that all clusters in $C'$ are $\gamma'$-homogeneous clusters. By running the algorithm again the second time, if the result $C'_j$ is not the same as the first running result $C'$, by the restriction in line 11, the different parts must be $\gamma'$-homogeneous clusters. However, the experts in the field should determine whether $\gamma'$ is appropriate for their problems.

4.5 Time Complexity

The least-squares approach has the time complexity as $O(n^2 d)$, where $n$ is the number of individuals and $d$ is the number of $X$ dimensions. Given $n_{\text{max}}$ is the number of individuals in the largest cluster and $|C|$ is the total number of clusters from all layers, Algorithm 3 has a time complexity as $O(|C| n_{\text{max}}^2 d)$, where $n_{\text{max}} \leq n$. The lower bound is $\Omega(n^2 d)$.

5 EXPERIMENTAL SETUP

We use both simulation and real-world datasets to evaluate our method performance.

5.1 Simulation Data

Given a multi-resolution cluster set $C = \{C_{j, k}\}$, we assign some subsets of $C$ to be a maximal homogeneous partition $C^* = \{C\}$. We generate realizations of independent random variables $X_1, \ldots, X_K$ from normal distributions, where $X_k \sim N(0, 1)$. For all members in each cluster $C_j \in C^*$, their random variables $Y_j, X_j$ have the same joint distribution. We have the following relation between $X_j$ and $Y_j$:

$$Y_j = c_1 \cdot X_j + c_2,$$

(18)

where $c_1, c_2 \in \mathbb{R}$, and $\forall k \neq j, Y_j \perp X_k$. This guarantees that each $C_j \in C^*$ has a different function $Y = h(X)$.

In our simulation setting, we have $X_1, \ldots, X_{20}$. Each individual $i$ has a pair of values $(y_i, x_i)$, where $x_i = (x_{1,i}, \ldots, x_{20,i})$. We have four types of simulation datasets for linear models. Each type has a different $C = \{C_{j, k}\}$ and $C^* = \{C\}$ (see Figure 2). Each cluster in the last layer of any dataset have 10,000 individuals as members. Hence, a type 1/2/3 dataset has $\tilde{y} = (y_1, \ldots, y_{80000})$, and $80000 \times 20$ matrix $X$, while a type 4 dataset has $\tilde{y} = (y_1, \ldots, y_{240000})$, and $240000 \times 20$ matrix $X$.

Moreover, we also generated two types of datasets that have $Y = h(X)$ as a non-linear function. The first one is a dataset type of exponential function:

$$Y_j = c_1 \cdot e^{X_j} + c_2.$$  

(19)
The second type is the polynomial function:

\[ Y_j = c_1 \cdot X_j^d + c_2, \]  

(20)

where \( c_1, c_2 \in \mathbb{R} \), and \( \forall k \neq j, Y_j = X_k \). We set \( d = 3 \) as the polynomial degree. We generated these non-linear datasets based on a multi-resolution cluster set of type 4 datasets in Figure 2. The parameter settings of both types are the same as the type 4 datasets except that there are 100 individuals per cluster for the last layer.

We use these simulation datasets to evaluate whether our framework can infer the correct \( C^* \). We generated 100 datasets for each type and used them to report the averages of framework performance for each dataset type. We define true-positive (TP) cases of prediction as those individuals in clusters that are within the ground-truth maximal homogeneous partition \( s.t. \) these clusters are also within the predicted maximal homogeneous partition. The false-negative (FN) cases of prediction denote those individuals in clusters within the ground-truth maximal homogeneous partition that are not in the predicted maximal homogeneous partition. The false-positive (FP) cases of prediction denote individuals that belong to clusters outside the maximal homogeneous partition, but the predicted results claim that these clusters are in the maximal homogeneous partition. We use TP, FN, and FP to compute precision, recall, and F1-score values in the Section 6.

Lastly, the distributions of all types of simulation data are shown in Figure 3.

5.1.1 Motivation of the Simulation Design. The MPI [3, 4] is used by the United Nations Development Programme (UNDP) to measure acute poverty around the world. By design, MPI considers several factors that can contribute to increasing poverty. In Thailand, there are five main dimensions that government policy makers consider in MPI: health, living condition, education, financial status, and access to public services (see Table 3 for more details). Note that MPI typically analyzes at the household level.

Given \( Q \) as a matrix derived from surveys, where \( Q(i, j) \) is a binary value of whether household \( i \) fails the dimension \( j \) in MPI: 1 for a fail status and 0 for a pass status. In other words, a row \( i \) of \( Q \) represents a vector of poverty statuses of household \( i \). For example, suppose \( j \) is the dimension of household average income, then if household \( i \) has average income below the government...
Fig. 3. Distributions of simulation data. (a)–(d) Distributions of independent and dependent variables of simulation type 1 to type 4. (e) Distributions of variables of exponential type. (f) Distributions of variables of polynomial type.

threshold, then \( Q(i, j) = 1 \). In the analysis, suppose \( i \) fails \( d_i \) dimensions. Then, \( i \) is considered to be poor if \( \bar{d}_i = d_i / n_0 \) is below the government threshold, where \( n_0 \) is the number of dimensions. Now, we are ready to mathematically define MPI.

\[
M_0 = q_0 \times a_0,
\]

where \( q_0 \in [0, 1] \) is the ratio of the number of poor people to the number of people in the whole population, \( a_0 \) is the average of \( \bar{d}_i \) among poor people, and \( M_0 \in [0, 1] \) is MPI. A higher \( M_0 \) implies more issues of poverty.

Based on the MPI concept, MPI cannot be used to find resolutions of common problems and dependency among dimensions (e.g., income and health issues). Hence, our aim is to identify whether other dimensions (e.g., health, education) affect incomes of households. We are interested in identifying the resolutions of common issues from specific dimensions that affect people’s incomes. Given that there are different linear dependencies between synthetic poverty dimensions (X variable) and income (Y variable) in simulation data in various resolutions, we want to, first, test whether our proposed framework can identify the correct resolution of models that share common dependencies between income and independent dimensions. Hence, the main purpose of our simulation is to use it as a sanity check whether our framework can correctly identify resolution w.r.t. our ground truth before deploying the proposed framework to the real-world datasets.
Table 3. Official Dimensions of MPI that Policy Makers of Thailand Currently Use to Design Policies that Are Related to Poverty Issues

| Main dimensions      | Subdimensions                                      |
|----------------------|-----------------------------------------------------|
| Health               | Birth weight records                                |
|                      | Hygiene and healthy diet                            |
|                      | Access to necessary medicines                       |
|                      | Work-out habits                                     |
| Living conditions    | Living in a reliable house                          |
|                      | Access to clean water                               |
|                      | Getting enough water for consumption                |
|                      | Living in a tidy house                              |
| Education            | Children at a pre-school age are prepared to go to school |
|                      | Children at a school age can attend to mandatory education |
|                      | Everyone in the household has attended at least high-school education |
| Financial status     | Adults (age 15–59) have reliable jobs               |
|                      | Seniors (age 60+) have incomes                      |
| Access to public services | Average income of household members                  |
|                      | Seniors can access public services in need          |
|                      | People with disabilities can access public services in need |

5.2 Real-World Data: Thailand’s Population Household Information

We obtained the dataset of Thailand household-population surveys from the Thai government. The surveys were collected in 2019. The surveys were used to estimate MPI [3, 4], which is the main poverty index that the United Nations (UN) is currently using. We used the surveys from two provinces, Khon Kaen and Chiang Mai, to perform our analysis. There are 353,910 households for Khon Kaen and 378,465 households for Chiang Mai. For each individual household, there are 30 dimensions that describe the characteristics of each household. Each dimension has three possible values: 1 means a good condition, 0 means no data, and −1 means a bad condition. These 30 dimensions can be categorized into five aspects: health, financial status, education, access to public services, and the living conditions. We remove the average household income dimension from the set of independent variables and make it a dependent variable for prediction. There are five layers of Thailand administrative divisions: (1) the nation (Thailand), (2) provinces (e.g., Khon Kaen and Chiang Mai), (3) amphoes, (4) tambons, and (5) villages. We created a multi-resolution cluster set from these administrative divisions. Our task is to infer which layers of these administrative divisions are predictive of income prediction given the household information.

Lastly, the distributions of real datasets are shown in Figure 4.

5.3 Baseline Method: Greedy Algorithm

To the best of our knowledge, since there is no method that we can compare against our approach directly, we compare our Algorithm 3 performance with the greedy algorithm in Algorithm 4 as a main baseline. The greedy algorithm infers the partition by greedily selecting the clusters that have the lowest root-mean-square error (RMSE) into the output partition. Hence, no algorithm can provide the partition that has the total RMSE as low as that of Algorithm 4. In line 3 of Algorithm 4, the RMSE of the cluster $C_{j,k}$ is calculated by fitting all datapoints in $C_{j,k}$ to make a model using linear regression. Then, we use the predicted values of the dependent variable from the inferred
Fig. 4. Distributions of real datasets. (Top) Distributions of independent and dependent variables of the Khon Kaen province. (Bottom) Distributions of variables of the Chiang Mai province.

**ALGORITHM 4:** Finding a partition using the greedy approach

**input:** A set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, a multi-resolution cluster set $C = \{C_{j,k}\}$

**output:** A set of models $\mathcal{H}$, a greedy partition $C^* = \{C\}$, a set of functions $H^* = \{h\}$ of $C^*$

1. for $C_{j,k} \in C$ do
2. | Calculate $\hat{\beta}_{j,k}$ using Equation (12) on data points in $C_{j,k}$;
3. | Calculate root mean square error (RMSE) for $C_{j,k}$;
4. | Append $h(x)_{j,k} = \bar{X}^T \hat{\beta}_{j,k}$ to $\mathcal{H}$;
5. end
6. Sort clusters in $C$ by RMSE and make $C_S$ as an ascendant sorted list of clusters;
7. for $C \in C_S$ do
8. | if there is no $C' \in C^*$ s.t. $C \subset C'$ then
9. | | Append $C$ to $C^*$;
10. end
11. for $C \in C^*$ do
12. | if $I(C, H_0, H_{\text{lin}}) > 0$ then
13. | | Append $h(x)_C \in \mathcal{H}$ to $H^*$ where $h(x)$ is $C$’s linear function;
14. | else
15. | | Append $h(x)_C = \bar{y}_C$ to $H^*$ where $\bar{y}_C$ is a mean of $y_i$ in $C$;
16. end
17. end
18. Return $H, C^*, H^*$;
model and the true values of the dependent variable to compute RMSE. In Section 6, we will show that our approach in Algorithm 3 provides the maximal homogeneous partition that has the total RMSE as low as Algorithm 4’s partitions.

The least-squares approach has the time complexity as \( O(n^2d) \), where \( n \) is the number of individuals and \( d \) is the number of \( X \) dimensions. Given \( n_{\text{max}} \) is the number of individuals in the largest cluster and \( |C| \) is the total number of clusters from all layers, Algorithm 4 has a time complexity as \( O(|C|n_{\text{max}}^2d) \), where \( n_{\text{max}} \leq n \). The time complexity is dominated by code lines 1–4 in Algorithm 4, which infers the model from data using the least-squares approach, which also occurs in Algorithm 3. This makes both Algorithm 3 and Algorithm 4 having the same time complexity. The lower bound is \( \Omega(n^2d) \).

5.4 Baseline Method: Finite MR Models
We compare our approach with the finite MR models developed by [20, 21, 30]. The mixture-model implementation is done in the R package “flexmix”. We set a number of components of mixture models (parameter \( k \)) corresponding to a number of homogeneous clusters within each dataset. We use a mixture model to show that even if mixtures of linear regression know the number of homogeneous clusters, in complicated scenarios, its performance is quite unstable. In contrast, our framework, which deploys the simple linear regression with multi-resolution partitions, can perform better than mixture models in the same complicated datasets.

Additionally, given \( n \) is the number of individuals and \( k = |C| \) is the number of clusters, Algorithm 3 has a time complexity as \( O(k \times n^2 \times d) \), while the time complexity of mixture models parameter estimates using the expectation-maximization (EM) algorithm [18] (deployed by flexmix) is \( O(k \times t_0 \times n \times d) \), where \( t_0 \in [1, \infty) \) is the number of time steps that the EM algorithm converges w.r.t. some predefined threshold. This makes MR running time unknown compared to our approach.

5.5 Software and Hardware Used in the Analysis
The computer that we used in this experiment is Lenovo Thinkpad T480s, with CPU Intel Core i7-8650U 1.9 GHz and RAM 16 GB. The software we used to conduct the experiment is the R studio version 1.2.5033 based on the R version 3.6.2. The R packages we used in the analysis are igraph [15], caret [28], and ggplot2 [51]. All experiments were conducted on Microsoft Windows 10. The implementation of our framework is in the form of the R package with documentation, which can be found in [5].

6 RESULTS
6.1 Simulation
In this section, we report the results of our analysis from simulation datasets (Section 5.1). We compared our approach (Algorithm 3) with the baseline methods from Sections 5.3 and 5.4.

The result of output partition from our approach is shown in Figure 5. The red nodes are clusters selected as members of maximal homogeneous partition. Comparing to the ground truth in Figure 2, our approach can infer the maximal homogeneous partition 100% correctly irrespective of whether the maximal homogeneous partition consists of clusters in the first layer (type 1 datasets), second layer (type 2 datasets), third layer (type 3 datasets), or multiple layers (type 4 datasets). In contrast, the result of output partition from the greedy algorithm in Figure 6 shows that it mostly selected the clusters in the last layers as the members of the partition. The greedy algorithm even included the lower clusters that are a subset of homogeneous clusters from the upper layer as the members of output partition. This is because the average RMSE of clusters from a lower layer is
Fig. 5. Inferred optimal partition from our method. The nodes represent clusters of population. The edges represent a subset relation between a parent cluster (head of arrow) and a child cluster (rear of arrow). The red nodes are clusters selected as members of output partition.

Table 4. Average RMSE Values of Approaches on Each Dataset Type

| Type   | OPT   | Greedy | Linear regression | Mixture of regression | $y_i - \bar{y}$ |
|--------|-------|--------|-------------------|-----------------------|-----------------|
| Type 1 Dataset | $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | $<10^{-6}$ | 6.30 |
| Type 2 Dataset | $<10^{-6}$ | $<10^{-6}$ | 5.30         | 0.56                  | 6.12 |
| Type 3 Dataset | $<10^{-6}$ | $<10^{-6}$ | 5.94         | 1.90                  | 6.35 |
| Type 4 Dataset | $<10^{-6}$ | $<10^{-6}$ | 5.58         | 12.67                 | 6.45 |

Each element in the table is the average of RMSEs from 100 datasets per dataset type. Each row represents results from a specific type of datasets. Each column represents a different method.

typically lower than that of the upper layer clusters. However, the difference between the average RMSE of the lower and upper layers is not significant (see T1–T3 datasets in Table 4). In fact, the lower layer clusters trend to have a lower RMSE compared to their superset cluster because the OLS seems to fit noise easier with a fewer number of individuals. This indicates that the greedy algorithm is sensitive to noise, which implies that it has an overfitting issue.

Table 4 illustrates the RMSE comparison between different methods. In all datasets, the results of our approach (OPT) are the same as that of the greedy one. In contrast, linear regression performs poorly in all datasets except the type 1 datasets. This is because the type 1 datasets have only a single homogeneous cluster, which implies that there is only a single linear function $h(x)$ that
generated $y$ from $x$. However, type 2/3/4 datasets have more than one linear function $h(x)$. By fitting a single linear regression function to these datasets, we expect to have a huge RMSE for these datasets (see Proposition 3.6). For an MR model (Section 5.4), it performs well for a dataset that possesses a fewer number of homogeneous clusters. Nevertheless, for the complicated datasets (type 4) that has 13 homogeneous clusters in the different layers, the MR was unable to perform well. This indicates that the MR, despite the generalization of linear regression, is quite unstable when the number of homogeneous clusters rises. The last column is the RMSE from the residual of difference between any $y_i$ and the mean of $y$ that we designate it as a null model. The results demonstrate that OPT and greedy perform a lot better than the null model.

In regard to the performance of inferring the maximal homogeneous partition, we reported the results of methods in Table 5. In the table, our method (OPT) can infer maximal homogeneous partition perfectly, while greedy performed mostly poorly except in type 3 datasets. This is because the greedy algorithm reports mostly the clusters in the last layer to be members of output partition and type 3 datasets have all clusters members of the maximal homogeneous partition in the last layer. For an MR, despite no access to multi-resolution partitions, it can retrieve homogeneous
Table 5. Average Precision, Recall, and F1 Score Values of Approaches on Each Dataset Type

| Type      | OPT | Greedy | Mixture of regression |
|-----------|-----|--------|-----------------------|
|           | Precision | Recall | F1  | Precision | Recall | F1  | Precision | Recall | F1  |
| Type 1    | 1   | 1      | 1  | 0.01      | 0.01   | 0.01 | 1         | 1      | 1   |
| Type 2    | 1   | 1      | 1  | 0.22      | 0.22   | 0.22 | 0.99      | 1      | 0.99|
| Type 3    | 1   | 1      | 1  | 1         | 1      | 1   | 0.96      | 1      | 0.97|
| Type 4    | 1   | 1      | 1  | 0.33      | 0.45   | 0.38 | 0.76      | 0.97   | 0.82|

Each element in the table is the average value from 100 datasets per dataset type.

Table 6. Average RMSE Values of Approaches on Non-LinearDatasets: Polynomial and Exponential Datasets

| Type      | OPT   | Greedy | Linear regression | Mixture of regression | $y_i - \bar{y}$ |
|-----------|-------|--------|-------------------|-----------------------|-----------------|
| EXP (LIN) | 8.41  | 7.42   | 12.91             | 5.00                  | 13.99           |
| Poly (LIN)| 16.64 | 13.46  | 22.74             | 12.13                 | 24.73           |
| EXP (EXP) | $<10^{-6}$ | $<10^{-6}$ | 12.26             | 5.16                  | 14.20           |
| Poly (EXP)| 14.59 | 13.45  | 22.43             | 12.40                 | 24.68           |

Each element in the table is the average of RMSEs from 100 datasets per dataset type. Each row represents results from a specific type of datasets. Each column represents a different method. The (LIN) implies that we used the linear function for all methods except $y_i - \bar{y}$ to fit the data, while the (EXP) implies we used an exponential function to fit the data except $y_i - \bar{y}$.

clusters well for datasets that have few homogeneous clusters (type 1/2/3). However, it performs fairly in the complicated type 4 datasets.

We also extended our approach and other methods to be able to fit data using the exponential function. We reported the results of using non-linear datasets in the analysis. Table 6 shows the average RMSE values of approaches on non-linear datasets: polynomial and exponential datasets. Our method (OPT) is not the best compared to the greedy and MR methods in regard to reducing RMSE.

The reason that the MR outperforms OPT is that it tries to find the fitting models without boundary constraints of partitions, while OPT fits models w.r.t. the given boundary constraints. It is possible that there are some partitions that are not the same as the given partitions from the input that have lower RMSEs when we fit models using them. If such partitions exist, then the MR might use the unconstrained partitions to fit its models. Hence, the RMSEs of the MR can be lower than OPT’s in this case. Additionally, the assumption we implicitly made was that if all methods fit models under the partition constraints, then greedy has the lowest RMSE. Because the MR breaks this assumption, its RMSE can be lower than the RMSE of the greedy algorithm.

However, when we consider the average precision, recall, and F1 score values in Table 7 for the task of inferring the correct maximal homogeneous partitions, OPT performed better than all other methods.

Comparing linear and exponential model fitting, in the (LIN) rows of Table 7, linear model fitting of OPT performed well in exponential datasets, but it did not perform well on the polynomial datasets. In contrast, the result of exponential model fitting in the (EXP) rows shows that OPT performed well in both exponential and polynomial datasets. This indicates that our approach is able to be extended to infer maximal homogeneous partitions in non-linear datasets well.

In summary, the results in this section indicate that our approach has the same RMSE as the greedy one (Table 4); nevertheless, its performance is a lot better in the regard to inferring the maximal homogeneous partition (Table 5). For the MR model, it performed well for type 1/2/3...
datasets, while its performance decreased for type 4 datasets, which have clusters in the maximal homogeneous partition from multiple layers. In fact, the homogeneous clusters that are inferred by an MR might not be consistent with a given multi-resolution partition. However, we can deploy MRs to approximate homogeneous clusters when datasets come without multi-resolution partitions. But we still need our approach to estimate the degree of homogeneity of each cluster as well as using it to compare different ways of partitioning a population. Moreover, we can even use the MR as a kernel instead of using linear regression to predict our dependent variable.

6.2 Case Study: Inferring Informative-Administrative-Level Subdivision to Predict Population Incomes

We use household population data (Section 5.2) to demonstrate the application of our framework to help the policy maker make a policy. In reality, policy makers cannot create a unique policy for every village to solve the poverty issues because of the limited resources. On the other hand, making a single policy cannot solve all poverty issues since each region has their own unique problems. Hence, we propose to use our framework to find the largest level of administrative subdivision that has enough common issues (measured by $\eta$ in Equation (16)) so that the policy makers do not need to make a policy for each village. In our framework, a $\gamma$-homogeneous cluster is considered to be an informative subdivision helping the policy maker make a policy. In this case study, we set $\gamma = 0.05$, which requires a correlation between dependent variable $Y$ and predicted $\hat{Y}$ of around 0.22, which is closed to a moderate correlation (a correlation of around 0.3 is considered a moderate correlation [12, 13]).

Table 8 illustrates the RMSE of each method from two provinces: Khon Kaen and Chiang Mai. The result shows that our approach (OPT) has a bit higher RMSE compared to the greedy algorithm. In contrast, linear regression and the null model $y_i - \bar{y}$ have a larger RMSE. Linear regression represents the approach of finding one policy for all regions within a province to predict the income of households, while our approach is trying to find a unique policy for each specific region. The result in Table 8 indicates that each region has its own problem. This is why fitting linear regression for the entire province population performed poorly.

Table 7. Average Precision, Recall, and F1 Score Values of Approaches on Non-Linear Datasets: Polynomial and Exponential Datasets

| Dataset            | OPT Precision | OPT Recall | OPT F1  | Greedy Precision | Greedy Recall | Greedy F1  | Mixture of regression Precision | Mixture of regression Recall | Mixture of regression F1  |
|--------------------|---------------|------------|---------|------------------|--------------|------------|-------------------------------|----------------------------|----------------------------|
| EXP dataset (LIN)  | 0.83          | 0.87       | 0.85    | 0.30             | 0.43         | 0.35       | 0.19                          | 0.16                       | 0.17                       |
| Poly dataset (LIN) | 0.53          | 0.57       | 0.55    | 0.30             | 0.43         | 0.35       | 0.16                          | 0.16                       | 0.16                       |
| EXP dataset (EXP)  | 1.00          | 1.00       | 1.00    | 0.33             | 0.44         | 0.38       | 0.19                          | 0.18                       | 0.19                       |
| Poly dataset (EXP) | 0.92          | 0.94       | 0.93    | 0.30             | 0.43         | 0.35       | 0.16                          | 0.15                       | 0.15                       |

Each element in the table is the average value from 100 datasets per dataset type. The (LIN) implies that we used the linear function for all methods to fit the data while the (EXP) implies we used an exponential function to fit the data.

Table 8. RMSE Values of Approaches on Each Province Population

| Province   | OPT           | Greedy         | Linear regression | $y_i - \bar{y}$ |
|------------|---------------|----------------|--------------------|------------------|
| Khon Kaen  | 69,620.3      | 67,210         | 82,571             | 84,994.6         |
| Chiang Mai | 89,067.1      | 87,727.5       | 104,938            | 106,511          |

Each element in the table is the RMSE value. Each row represents results from a specific province. Each column represents a different method.
Table 9 shows the number of clusters that are the members of output partition in each method. The first layer is the province level, the second layer is the amphoe level, the third layer is the tambon level, and the fourth layer is the village level. The result indicates that our approach can detect informative subdivisions beyond the last layer, while the greedy approach can report only the clusters in the last layer.

After we got the maximal homogeneous partition, we need to validate whether the result is consistent with the ground truth of the government records. The following areas are examples of members of the maximal homogeneous partition.

First, in the Khon Kaen province, the “Subsomboon” tambon, which is in the third layer, has 11 villages in the fourth layer. A total of 8 out of 11 villages have the majority of poor people suffering from the lack of the education issue. Second, another tambon in the Khon Kaen province is Khu Kham. It has seven out of eight villages that face the lack of education issue among poor people. Third, in the Chiang Mai province, the San Sai amphoe (second layer) has 12 tambons. A total of 10 out of 12 tambons have finance and education as leading issues.

We can see that, for each tambon in the examples above, policy makers can make a single policy for all villages because the majority of subareas in each homogeneous area have almost the same issues.

Next, the non-homogeneous cases are provided below.

In the Khon Kaen province, Sila tambon (third layer) consists of 28 villages. There are 11 villages that have no poor people. There are 11 villages that have financial issues among the majority of poor people. There are five villages facing health issues among the majority of poor people in each village. One village has an equal number of poor people facing financial and health issues. We can see that the policy makers need to group villages in this tambon w.r.t. the issue types before making policies that fit each type of issue.

In the Chiang Mai province, the Chiang Dao amphoe (second layer) has seven tambons. Two tambons have financial issues among the majority of poor people. Two tambons have financial and education issues. Two tambons have financial and health issues. Lastly, one tambon has the education issue as a main problem. It is quite challenging for policy makers to place a single policy for the entire Chiang Dao amphoe.

The next is an example of how policy makers can use our system to combat poverty. One of the tambons in the output maximal homogeneous partition of our approach from the Khon Kaen province is “Subsomboon”. We used the $t$-test to find the variable importance of coefficients of regression that have values far from zero for this tambon. The null hypothesis is that the absolute of specific coefficient is zero, while the alternative hypothesis is that the absolute of coefficient is greater than zero. We reject the null hypothesis at $\alpha = 0.01$.

Out of 30 coefficients, there are only 4 coefficients for which we can reject the null hypothesis. However, the only positive coefficient is the indicator of whether a household has a savings account

---

\[1\] All reference data of government records used here are from https://www.tpmap.in.th/. We used the 2019 records of poor people as a ground truth in this result section.
with a bank. This implies that having a savings account is associated with income. The policy
maker should consider why some households can have a savings account to make a policy for
combating poverty in the Subsomboon tambon. For the Chiang Mai province, the San Sai amphoe
is one of the informative clusters in the output partition. The variables that are important and
have positive coefficients mostly are health issues in the household that might make the members
of household work inefficiently. Hence, the policy maker should consider to make a policy that
supports the health of people in the area.

In practice, the greedy algorithm performs well only when all homogeneous clusters are mostly
in the last layer. Even in this case, its performance is not significantly better than OPT. Moreover,
the time and space complexities of both algorithms are the same. Hence, OPT should always be
used to find the maximal homogeneous partition.

7 DISCUSSION

7.1 Inferring Multi-Resolution Partitions

In practice, the obvious case that a dataset of multi-resolution partitions can be found is when
a dataset is related to administrative subdivisions of a country. In each subdistrict in a specific
resolution, the data can be varied, such as household income, types of land use, distribution of
labor force, and so forth. In the eyes of policy makers, knowing which area shares similar models
for target properties makes them to have an easier way to manage resources or to place policies.

However, in some cases, there are no given multi-resolution partitions. For example, suppose
we divide a natural farming area into grids and we have information about the history of land use
of each block in the grid (e.g., the remaining resource). Our goal is to infer partitions of multiple
neighbor blocks that we can declare as either preserved areas to protect the land or non-preserved
areas. In this situation, we can use hierarchical clustering [17, 25, 45] (hclust function in R pro-
gramming [41]) to build a hierarchical tree of blocks. We can assign layers of partitions based on
the distance of nodes from the nearest leaf. Another approach is to use the MR method [20, 21, 30]
in Section 5.4. We might run the MR method to get the first-layer result, then, for each inferred par-
tition, we might run the method again to find the subdistricts. Nevertheless, this approach requires
users to set an appropriate number of clusters $k$.

7.2 Extension to Non-Linear Models

It is possible to extend our work to non-linear models if we care for only a number of bits and inform-
ation we need to explain models. However, in machine learning, there are issues of underfitting
and overfitting that we should consider.

To resolve underfitting issues, we try to find $h$ that minimizes $\mathcal{L}(S|h)$ in Equation (5). Neverthe-
less, a complex model typically fits data well, which makes $\mathcal{L}(S|h)$ small. To resolve the overfitting
issue, since we prefer a simpler model that has its performance close to complicated model, we try
to find $h$ that also minimizes $\mathcal{L}(h)$ in Equation (4).

We can say that $\mathcal{L}(h)$ represents the model complexity. However, there are many challenges
that we need to address in order to measure $\mathcal{L}(h)$ for any arbitrary $h$.

It is straightforward to compare $\mathcal{L}(h_i)$ against $\mathcal{L}(h_j)$ from the same function class (e.g., a linear
function class, an exponential function class) by measuring a number of bits we need to keep
coefficients and terms. For example, the model complexity of one linear function that has a fewer
number of terms with small-size coefficients can be considered less than that of a complicated
linear function that has many terms with large coefficients. In contrast, it is challenging to compare
two models from different classes and declare that one model is less complicated than another (e.g.,
sine function vs. logarithmic function).
7.3 Sparsity Regularization

In the context of sparsity regularization, we also want to get a model that has the minimum number of dimensions we used in the model along with the prediction performance $\mathcal{L}(S|h)$ (Equation (5)) and the model complexity $\mathcal{L}(h)$ (Equation (4)). Suppose $\beta = (p_1, \ldots, p_l)$ is a vector of parameters for a model $h$, and given a set of data $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and a penalty weight $\lambda$, we can have the following optimization problem to find the optimal $\beta$ from data that utilizes the sparsity regularization:

$$
\beta^\ast(\lambda) = \arg\min_{\beta \in \mathbb{R}^l} \frac{1}{n} \sum_{i=1}^{n} \text{loss}(y_i, f_{\beta}(x_i)) + \lambda \|\beta\|_0,
$$

where $\|\beta\|_0$ is the number of parameters in $\beta$ that have non-zero values, $f_{\beta}(x_i)$ is a function that returns the predicted value of $y_i$ using $\beta$ and $x_i$, and loss() is a loss function. Based on the optimization problem above, we can modify $\mathcal{L}(h)$ to utilize the sparsity regularization:

$$
\mathcal{L}(h, \lambda) = \sum_{i=1}^{l} \mathcal{L}_R(p_i) + \lambda \mathcal{L}_R(\|\beta\|_0).
$$

Hence, we have

$$
\mathcal{L}(S, \lambda)_{\mathbb{H}} = \min_{h \in \mathbb{H}} \mathcal{L}(h, \lambda) + \mathcal{L}(S|h).
$$

Equation (22) encourages a model $h$ that has a good performance (minimizing $\mathcal{L}(S|h)$) as well as using fewer parameters that have small magnitudes (minimizing $\mathcal{L}(h, \lambda)$).

7.4 Encoding Methods for Real Numbers

As a reminder, Assumption 1 states the following.

For any $x_1, x_2 \in \mathbb{R}$ s.t. $|x_1| \geq |x_2|$, we have

$$
\mathcal{L}_R(x_1) \geq \mathcal{L}_R(x_2).
$$

This assumption is true for the Single-Precision Floating-point Format (IEEE Standard 754 [23]), which is widely used in several well-known programming languages (e.g., Fortran, C, C++, C#, Java) to encode real numbers. However, the floating-point format makes every real number use an equal number of bits: 32 bits per number.

Nevertheless, the work by [32] provided the framework to implement the universal coding for real numbers. In this coding, the number of bits varies proportional to a real-number magnitude. Specifically, for any $y \in \mathbb{R}$,

$$
|\mathcal{E}(y)| \propto |\log_2|y||,
$$

where $|\mathcal{E}(y)|$ is the number of bits of real-number representation in [32]. In practice, if a specific application has a high variation of magnitudes of real numbers, then the encoding in [32] might
be able to reduce spaces required for keeping real numbers, and our framework can obviously be benefitted by this. For a specific application, one might check whether the encoding scheme for real numbers that complied with our assumption is suitable for being used under the application.

8 CONCLUSION

In this article, we addressed the challenges encountered by many policy makers: to find the largest possible areas with common problems where similar policies can be implemented and limited resources are optimally utilized. Given a set of target variables and predictors, as well as a set of multi-resolution clusters that represent administrative divisions as inputs, we proposed a framework to infer a set of largest informative clusters that have efficient predictors. Efficient predictors of each informative cluster cover the entire cluster population. We used both simulation datasets and real-world datasets of Thailand’s population household information to evaluate and to illustrate the application of our framework. The results showed that our framework performed better than all baseline methods. Moreover, in Thailand’s population household information, the framework can infer the predictors of people income for the specific areas, which can be used to guide policy makers to utilize this information to combat poverty. Particularly, simulated data show that our approach can estimate the homogeneity of each cluster and compare different ways of population partition, which would help formulate evidence-based policies as well as their assessments. This finding is confirmed with real-world data from the two major provinces of Thailand, where it is possible to cluster regions with similar problems up to the second (amphoe) level. More importantly, our framework can be used beyond the context of income prediction for any kind of regression analysis on multi-resolution clusters. Lastly, we also provide the R package, MRReg, which is the implementation of our framework in the R language with the manual given in [5]. The official link for MRReg at the Comprehensive R Archive Network (CRAN) can be found at https://cran.r-project.org/package=MRReg.

ACKNOWLEDGMENT

The authors would like to thank the National Electronics and Computer Technology Center (NECTEC), Thailand, for providing their resources to successfully complete this work.

REFERENCES

[1] IJsbrand Jan Aalbersberg and Grzegorz Rozenberg. 1988. Theory of traces. Theoretical Computer Science 60, 1 (1988), 1–82. DOI: https://doi.org/10.1016/0304-3975(88)90051-5
[2] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. 2012. Learning from Data, Vol. 4. AMLBook, New York, NY.
[3] Sabina Alkire, Usha Kanagaratnam, and Nicolai Suppa. 2019. The Global Multidimensional Poverty Index (MPI) 2019. OPHI MPI Methodological Note 47. Oxford Poverty and Human Development Initiative (OPHI), Oxford Department of International Development, University of Oxford.
[4] Sabina Alkire and Maria Emma Santos. 2010. Multidimensional Poverty Index 2010: Research Briefing. Oxford Poverty and Human Development Initiative (OPHI), Oxford Department of International Development, University of Oxford.
[5] C. Amornbunchornvej. 2020. MRReg: The MDL Multiresolution Linear Regression Framework in R. Retrieved May 11, 2020 from https://github.com/DarkEyes/MRReg.
[6] Senjian An, Wanquan Liu, and Svetla Venkatesh. 2007. Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression. Pattern Recognition 40, 8 (2007), 2154–2162.
[7] Sudhir Anand and Amartya Sen. 1997. Concepts or human development and poverty: A multidimensional perspective. In Poverty and Human Development: Human Development Papers 1997. United Nations Development Programme, New York, NY, 1–20.
[8] UN General Assembly. 2015. The 2030 Agenda for Sustainable Development. Technical Report. Resolution A/RES/70/1.
[9] Erdal Atukeren. 2010. The relationship between the F-test and the Schwarz criterion: Implications for Granger-causality tests. Economics Bulletin 30, 1 (2010), 494–499.

ACM Transactions on Knowledge Discovery from Data, Vol. 15, No. 2, Article 15. Publication date: December 2020.
[10] Julio Berdegué and Germán Escobar. 2002. Rural Diversity, Agricultural Innovation Policies and Poverty Reduction. Agricultural Research and Extension Network.

[11] Michael W. Browne. 2000. Cross-validation methods. Journal of Mathematical Psychology 44, 1 (2000), 108–132. DOI: https://doi.org/10.1006/jmps.1999.1279

[12] Jacob Cohen. 1988. Set correlation and contingency tables. Applied Psychological Measurement 12, 4 (1988), 425–434.

[13] Jacob Cohen. 2013. Statistical Power Analysis for the Behavioral Sciences. Routledge.

[14] Patrick Commins. 2004. Poverty and social exclusion in rural areas: Characteristics, processes and research issues. Sociologia Ruralis 44, 1 (2004), 60–75.

[15] Gabor Csardi and Tamas Nepusz. 2006. The igraph software package for complex network research. InterJournal Complex Systems (2006), 1695. Retrieved from http://igraph.org.

[16] R. D. de Veaux. 1989. Mixtures of linear regressions. Computational Statistics & Data Analysis 8, 3 (Nov. 1989), 227–245. DOI: https://doi.org/10.1016/0167-9473(89)90043-1

[17] D. Defays. 1977. An efficient algorithm for a complete link method. Computer Journal 20, 4 (Jan. 1977), 364–366. DOI: https://doi.org/10.1093/comjnl/20.4.364

[18] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological) 39, 1 (1977), 1–22.

[19] C. W. J. Granger. 1969. Investigating causal relations by econometric models and cross-spectral methods. Econometrica 37, 3 (1969), 424–438. Retrieved from http://www.jstor.org/stable/1912791.

[20] Bettina Grün and Friedrich Leisch. 2006. Fitting finite mixtures of linear regression models with varying & fixed effects in R. In Proceedings of Computational Statistics (Compstat ’06). Alfredo Rizzi and Maurizio Vichi (Eds.), Physica Verlag, Heidelberg, 853–860.

[21] Bettina Grün and Friedrich Leisch. 2007. Applications of finite mixtures of regression models. Retrieved from http://cran.r-project.org/web/packages/flexmix/vignettes/regression-examples.pdf.

[22] Mark H. Hansen and Bin Yu. 2001. Model selection and the principle of minimum description length. Journal of the American Statistical Association 96, 454 (2001), 746–774.

[23] David G. Hough. 2013. The IEEE standard 754: One for the history books. Computer 52, 12 (2019), 109–112.

[24] Neal Jean, Marshall Burke, Michael Xie, W. Matthew Davis, David B. Lobell, and Stefano Ermon. 2016. Combining satellite imagery and machine learning to predict poverty. Science 353, 6301 (2016), 790–794.

[25] Joe H. Ward Jr. 1963. Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58, 301 (1963), 236–244. DOI: https://doi.org/10.1080/01621459.1963.10500845

[26] Vladimir Koltchinskii. 2001. Rademacher penalties and structural risk minimization. IEEE Transactions on Information Theory 47, 5 (2001), 1902–1914.

[27] Georgios Kostopoulos, Stamatis Karlos, Sotiris Kotsiantis, and Omiros Ragos. 2018. Semi-supervised regression: A recent review. Journal of Intelligent & Fuzzy Systems 35, 2 (2018), 1483–1500. DOI: 10.3233/JIFS-169689

[28] Max Kuhn. 2020. caret: Classification and Regression Training. Retrieved from https://CRAN.R-project.org/package=caret R package version 6.0-86.

[29] Glada Lahn and Paul Stevens. 2017. The Curse of the One-Size-Fits-All Fix: Re-Evaluating What We Know about Extrac- tives and Economic Development. Technical Report. WIDER Working Paper.

[30] Friedrich Leisch. 2004. FlexMix: A general framework for finite mixture models and latent class regression in R. Journal of Statistical Software 11, 8 (2004), 1–18. DOI: https://doi.org/10.18637/jss.v011.i08

[31] Yuanzhi Li and Yingyu Liang. 2018. Learning mixtures of linear regressions with nearly optimal complexity. In Proceedings of Machine Learning Research, Vol. 75. 1125–1144.

[32] Peter Lindstrom, Scott Lloyd, and Jeffrey Hittinger. 2018. Universal coding of the reals: Alternatives to IEEE floating point. In Proceedings of the Conference for Next Generation Arithmetic (CoNGA ’18). ACM, New York, NY, Article 15, 14 pages. DOI: https://doi.org/10.1145/3190339.3190344

[33] G. J. McLachlan and D. Peel. 2000. Finite Mixture Models. Wiley, New York, NY.

[34] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Foundations of Machine Learning. MIT Press.

[35] Laura L. Nathans, Frederick L. Oswald, and Kim Nimon. 2012. Interpreting multiple linear regression: A guidebook for variable importance. Practical Assessment, Research & Evaluation 17, 9 (2012).

[36] Christopher Njuguna and Patrick McSharry. 2017. Constructing spatiotemporal poverty indices from big data. Journal of Business Research 70 (2017), 318–327. https://www.sciencedirect.com/science/article/abs/pii/S0148296316304921.

[37] Judea Pearl. 2009. Causality. Cambridge University Press.

[38] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. Elements of Causal Inference: Foundations and Learning Algorithms. MIT Press.

[39] Thomas Pogge. 2005. World poverty and human rights. Ethics & International Affairs 19, 1 (2005), 1–7.

[40] D. G. Pringle, Sally Cook, M. A. Poole, and Adrian Moore. 2000. Cross-Border Deprivation Analysis: A Summary Guide. Oak Tree Press.
[41] R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna. Retrieved from https://www.R-project.org/.

[42] Felix Rioja and Neven Valev. 2004. Does one size fit all?: A reexamination of the finance and growth relationship. *Journal of Development Economics* 74, 2 (2004), 429–447. DOI: https://doi.org/10.1016/j.jdeveco.2003.06.006

[43] J. Rissanen. 1978. Modeling by shortest data description. *Automatica* 14, 5 (1978), 465–471. DOI: https://doi.org/10.1016/0005-1098(78)90005-5

[44] Daniel F. Schmidt and Enes Makalic. 2012. The consistency of MDL for linear regression models with increasing signal-to-noise ratio. *IEEE Transactions on Signal Processing* 60, 3 (2012), 1508–1510.

[45] R. Sibson. 1973. SLINK: An optimally efficient algorithm for the single-link cluster method. *Computer Journal* 16, 1 (Jan. 1973), 30–34. DOI: https://doi.org/10.1093/comjnl/16.1.30

[46] Harald Strotmann and Jürgen Volkert. 2018. Multidimensional poverty index and happiness. *Journal of Happiness Studies* 19, 1 (Jan. 2018), 167–189. DOI: https://doi.org/10.1007/s10902-016-9807-0

[47] Peter Turney. 1994. A theory of cross-validation error. *Journal of Experimental & Theoretical Artificial Intelligence* 6, 4 (1994), 361–391.

[48] George Udny Yule. 1927. On a method of investigating periodicities in disturbed series, with special reference to Wolfer’s sunspot numbers. *Philosophical Transactions of the Royal Society of London Series A* 226 (1927), 267–298. https://ui.adsabs.harvard.edu/abs/1927RSPTA.226..267U/abstract.

[49] V. N. Vapnik and A. Ya. Chervonenkis. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications* 16, 2 (1971), 264–280. DOI: https://doi.org/10.1137/1116025

[50] Michel Wedel and Wayne S. DeSarbo. 1995. A mixture likelihood approach for generalized linear models. *Journal of Classification* 12, 1 (Mar. 1995), 21–55. DOI: https://doi.org/10.1007/BF01202266

[51] Hadley Wickham. 2016. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York, NY. Retrieved from https://ggplot2.tidyverse.org.

Received July 2019; revised August 2020; accepted September 2020