Disaggregation via Gaussian Regression for Robust Analysis of Heterogeneous Data

Nazanin Alipourfard, Keith Burghardt, Kristina Lerman

1 Introduction

Social data is often highly heterogeneous, coming from a population composed of diverse classes of individuals, each with their own characteristics and behaviors. As a result of heterogeneity, a model learned on population data may not make accurate predictions on held-out test data, nor offer analytic insights into the underlying behaviors that motivate interventions. To illustrate, consider Figure 1, which shows data collected for a hypothetical nutrition study measuring how the outcome, body mass index (BMI), changes as a function of daily pasta calorie intake. Multivariate linear regression (MLR) analysis finds a negative relationship in the population (red dotted line) between these variables. The negative trend suggests that—paradoxically—increased pasta consumption is associated with lower BMI. However, unbeknownst to researchers, the hypothetical population is heterogeneous, composed of classes that varied in their fitness level. These classes (clusters in Fig. 1) represent, respectively, people who do not exercise, people with normal activity level, and athletes. When data is disaggregated by fitness level, the trends within each subgroup are positive (dashed green lines), leading to the conclusion that increased pasta consumption is in fact associated with a higher BMI. Recommendations for pasta consumption arising from the naive analysis are opposite to those arising from a more careful analysis that accounts for the confounding effect of different classes of people. The trend reversal is an example Simpson's paradox, which has been widely observed in many domains, including biology, psychology, astronomy, and computational social science (Chuang et al., 2009; Kievit et al., 2013; Minchev et al., 2019; Blyth, 1972).

Social scientists analyze heterogeneous data with underlying structure (classes) using mixed effects models (Winter, 2013). This variant of linear regression uses random intercepts to model differences between classes and random slopes to model differences in regression coefficients within classes. Mixed effects models are used to describe non-independent observations of data from the same class, and they can even handle trend reversal associated with Simpson's paradox. Mixed effects models assume that classes are specified by a categorical variable, which can be constructed by disaggregating or binning data on an existing variable (Alipourfard et al., 2018a). In practice, however,

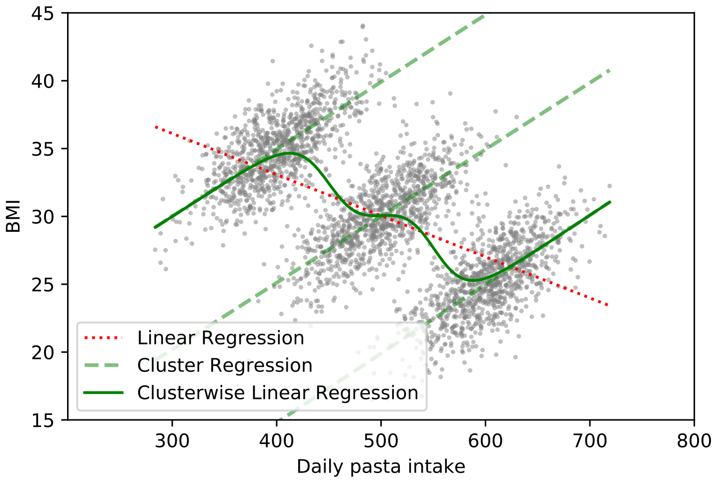


Figure 1: Heterogeneous data with three latent classes. The figure illustrates Simpson's paradox, where a positive relationship between the outcome and the independent variable exists for population as a whole (red dotted line) but reverses when the data is disaggregated by classes (dashed green lines).

these classes may not be known a priori, or be related to multiple variables. Instead, they must be discovered from data, along with the trends they represent. Many methods already exist for finding latent classes within data; however, they have various shortcomings. Unsupervised clustering methods disaggregate data regardless of the outcome variable being explained. In reality, distinct outcomes may be best described by different clusterings of the same data. Recent methods have tackled Simpson's paradox by performing supervised disaggregation of data (Alipourfard et al., 2018b; Fabris and Freitas, 2000). However, these methods disaggregate data into subgroups using existing features, and thus are not able to capture effects of latent classes (i.e., unobserved confounders). Additionally, they perform "hard" clustering, but assigning each data point to a unique group. Instead, a "soft" clustering is more realistic as it captures the degree of uncertainty about which group the data belong to. To address these challenges, we describe *Disaggregation via Gaussian Regression* [1] (*DoGR*), a method that jointly partitions data into overlapping clusters and estimates linear trends within them. This allows for learning accurate and generalizable models while keeping the interpretability advantage of linear models. Proposed method assumes that the data can be described as a superposition of clusters, or components, with every data point having some probability to belong to any of the components. Each component represents a latent class, and the soft membership represents uncertainty about which class a data point belongs to. We use an expectation maximization (EM) algorithm

[1] http://github.com/ninoch/DoGR

to estimate component parameters and component membership parameters for the individual data points. *DoGR* jointly updates component parameters and its regression coefficients, weighing the contribution of each data point to the component regression coefficients by its membership parameter. The joint learning of clusters and their regressions allows for discovering the latent classes that explain the structure of data.

Our framework learns robustly predictive and interpretable models of data. In validations on synthetic and real-world data, *DoGR* is able to discover hidden subgroups in data even in analytically challenging scenarios, where subgroups overlap along some of the dimensions, and identify interesting trends that may be obscured by Simpson's paradox, including in scenarios where the trends in data show Simpson's reversal. We show that our method achieves performance on prediction tasks comparable to or better than state-of-the-art algorithms, but in a fraction of the time, making it suitable for big data analysis.

2 Background

Regression is one of the most widely used methods in data analysis due to its flexibility, interpretability, and computational scalability (Fox, 1997). The most popular regression method uses linear models to learn the relationship between predictor variables (also known as fixed effects or features) and the outcome variable. However, linear model make assumptions about the data that are often violated by real-world data. One of these assumptions is independence of observations. In reality, however, observations may come from groups of similar individuals, violating the independence assumption. To account for individual differences within such groups, mixed-effect or multi-level models were developed (McLean et al., 1991). These models assume that similar individuals, or data samples, come from the same class, and that classes are specified by some categorical variable (Winter, 2013). In some cases, this variable can be created by binning a continuous variable and disaggregating data by these bins (Alipourfard et al., 2018a). However, existing methods do not work when differences between groups or individuals cannot directly be described by observed variables or features. In this chapter, we describe a method that captured differences between groups as latent confounders (i.e., latent clusters) that are learned from data. Our work improves the utility of linear models in the analysis of social data by controlling for these clusters. A number of previous works have attempted to tackle the issue of latent confounders. Clusterwise linear regression (CLR) (Späth, 1979) starts with initial clusters and updates them by reassigning one data point to another partition that minimizes a loss function. The method is slow, since it moves one data point at each iteration. Two other methods, WCLR (da Silva and de Carvalho, 2017) and FWCLR (da Silva and de Carvalho, 2018), improve on CLR by using k-means as their clustering method. These methods were shown to outperform CLR and other methods, such as K-plane (Manwani and Sastry, 2015). In Conditional Linear Regression (Calderon et al., 2018), the goal is to find a linear rule capable of achieving more accurate predictions for just a *segment* of the population, by ignoring outliers. One of the larger differences between their method and ours is that Conditional Linear Regression focuses on a small subset of data, while we model data as a whole, alike to CLR, WCLR, FWCLR and K-plane. The shared parameter λ across all clusters in WCLR and FWCLR makes these methods perform poorly if clusters have different variance, as we will show in our results section. Other methods have combined Gaussian Mixture Models and regression to create algorithms similar to our own (Sung, 2004; Ghahramani and Jordan, 1994). We call these methods *Guassian Mixture Regression (GMR)*. In contrast to these methods, however, we can capture relationships between independent and outcome variables through regression coefficients, which previous methods were unable to do. We also use *Weighted Least Squares* to fit our model, which makes it less sensitive to outliers (Cleveland and Devlin, 1988).

Causal inference has been used to address the problem of confounders, including latent confounders (Louizos et al., 2017), and to infer the causal relationship between features and outcomes (Bareinboim and Pearl, 2016; Ranganath and Perotte, 2018; Wang and Blei, 2018). One difficulty with causal inference however, is that the focus is traditionally on one intervention (Ranganath and Perotte, 2018). Taking into account synergistic effects of multiple features is not well understood, but has been attempted recently (Ranganath and Perotte, 2018; Wang and Blei, 2018). With adequate datasets, these can help infer causal relationships between multiple causes, but certain causal assumptions are needed, which might not correspond to reality. In contrast, regression offers us the opportunity to understand relationships between each feature and an outcome, regardless of the dataset, even if we cannot make causal claims.

3 *DoGR* Method

Let $\mathcal{D} = \{d_1, d_2, d_3, ..., d_N | d_i = (x_i, y_i) \in \mathbb{R}^p \times \mathbb{R}\}$ be a set of N observations, or records, each containing a real-valued outcome y_i of an observation i and a vector of p independent variables, or features, $[x_{i,1}, x_{i,2}, ..., x_{i,p}]$. Regression analysis is often used to capture the relationship between each independent variable and the outcome. Specifically, MLR estimates regression coefficients $\beta_0, \beta_1, ..., \beta_p$ by minimizing the residuals of $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$ over all observations. However, parameters learned by the *MLR* model may not generalize to out-of-sample populations, as they can be confounded by Simpson's paradox. We can call this the "robustness" problem of regression.

As discussed in the introduction, Figure 1 illustrates this problem with synthetic data. The *MLR* model trained on the aggregate data gives $BMI = intercept - 0.61x$. This suggests a negative relationship (red dotted line) between the independent variable $x =$*Daily pasta intake* and the outcome BMI. However, there are three components each with a positive trend. Indeed, applied separately to each component, *MLR* learns the proper positive relationship (dashed green line) between *Daily pasta intake* and

BMI: $BMI = \beta_0 + \beta_1 x$, where β_0 is cluster's intercept, and β_1 is coefficient: $\{1.03, 1.08, 0.97\}$. Associations learned by the *MLR* model trained on the population-level data are not *robust* or *generalizable*. This could explain the *reproducibility problem* seen in many fields (Pashler and Wagenmakers, 2012; Smaldino and McElreath, 2016), where trends seen in some experiments cannot be reproduced in other experiments with different populations. While the underlying groups within the population may be the same, the later experiments may over-sample some of the groups during data collection. As a result, *MLR* trained at the population-level may find a different association between independent variable and the outcome. However, identifying each group (cluster) and analyzing it separately reveals the same trends in the new data. In summary, disaggregating heterogeneous data and performing cluster-wise regressions will identify *robust* and *generalizable* relationships in data.

3.1 Model Specification

The goal of this work is to learn robust and reproducible trends through a regression model that accounts for the latent structure of data, for example, the presence of three clusters in data shown in Fig. 1. Our model jointly *disaggregates* the data into K overlapping subgroups, or components, and performs *weighted linear regression* within each component. We allow components to overlap in order to represent the uncertainty about which component or subgroup an observation belongs to.

In what follows, we used capital letters to denote random variables and lowercase letters their values. We model the independent variable X of each component k as a multivariate normal distribution with mean $\mu_k \in \mathbb{R}^p$ and covariance matrix $\Sigma_k \in \mathbb{R}^{p \times p}$:

$$f_X^{(k)} \sim \mathcal{N}(\mu_k, \Sigma_k) \tag{1}$$

In addition, each component is characterized by a set of regression coefficients $\beta_k \in \mathbb{R}^{p+1}$. The regression values of the component k are

$$\hat{Y}^{(k)} = \beta_{k,0} + \beta_{k,1} X_1 + \beta_{k,2} X_2 + ... + \beta_{k,p} X_p, \tag{2}$$

with $y - \hat{y}^{(k)}$ giving the residuals for component k. Under the assumption of normality of residuals (DeSarbo and Cron, 1988), Y has a normal distribution with mean $\hat{Y}^{(k)}$ and standard deviation σ_k.

$$f_{Y|X}^{(k)} \sim \mathcal{N}(\hat{Y}^{(k)}, \sigma_k), \tag{3}$$

where $\hat{Y}^{(k)}$ is defined by Eq. 2. Under the assumption of *homoscedasticity*, in which the error is the same for all X, the joint density is the product of the conditional (Eq. 3) and marginal (Eq. 1) densities:

$$\begin{aligned} f_{X,Y}^{(k)}(x, y) &= f_{Y|X}(y|x) f_X(x) \\ &= \varphi(y; \hat{y}^{(k)}, \sigma_k) \varphi(x; \mu_k, \Sigma_k) \end{aligned} \tag{4}$$

Multiplication of $f_{Y|X}(y|x)f_X(x)$ can be converted to a normal distribution: $f^{(k)}_{X,Y}(x,y)$ comes from $\mathcal{N}(\mu'_k, \Sigma'_k)$, where $\mu'_k = [\mu^{(1)}_k, \mu^{(2)}_k, \ldots, \mu^{(d)}_k, \hat{y}^{(k)}]$ and $\Sigma'_k = \begin{bmatrix} \Sigma_k & 0 \\ 0 & \sigma_k \end{bmatrix}$ is a block matrix.

3.2 Model Learning

The final goal of the model is to predict the outcome, given independent variables. This prediction combines the predicted values of outcome from all components by taking the average of the predicted values, weighed by the size of the component. We define ω_k as the weight of the component k, where $\sum_k \omega_k = 1$. We can define the joint distribution over all components as $f_{X,Y}(x,y) = \sum_{k=1}^{K} \omega_k \times f^{(k)}_{X,Y}(x,y)$.
Then the *log-likelihood* of the model over all data points is:

$$\mathcal{L} = \sum_{i=1}^{N} log\big(\sum_{k=1}^{K} \omega_k \times f^{(k)}_{X,Y}(x_i, y_i)\big) \tag{5}$$

The formula here is same as the *Gaussian Mixture Model*, except that the target y_i is a function of x_i. To find the best values for parameters $\theta = \{\omega_k, \mu_k, \Sigma_k, \beta_k, \sigma_k | 1 \leq k \leq K\}$ we can leverage the *Expectation Maximization* (EM) algorithm. The algorithm iteratively refines parameters based on the expectation (E) and maximization (M) steps.

3.2.1 E-step

Let's define $\gamma_{i,k}$ (*membership parameter*) as probability that data point i belongs to component k. Given the parameters θ_t of last iteration, membership parameter is:

$$\gamma_{i,k} = \frac{\omega_k \times f^{(k)}_{X,Y}(x_i, y_i)}{\sum_{k'} \omega'_k \times f^{(k')}_{X,Y}(x_i, y_i)} \tag{6}$$

Thus, the E-step disaggregates the data into clusters, but it does so in a "soft" way, with each data point having some probability to belong to each cluster.

3.2.2 M-step

Given the updated membership parameters, the M-step updates the parameters of the model for the next iteration as θ_{t+1}:

$$\omega_k = \frac{\sum_i \gamma_{i,k}}{N}$$

$$\mu_k = \frac{\sum_i \gamma_{i,k} x_i}{\sum_i \gamma_{i,k}}$$

$$\Sigma_k = \frac{\sum_i \gamma_{i,k}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i \gamma_{i,k}}$$

In addition, this step updates regression parameters based on the estimated parameters in θ_t. Our method uses *Weighted Least Squares* for updating the regression coefficients β_k for each component, with γ as weights. In the other words, we find β_k that minimizes the *Weighted Sum of Squares (WSS)* of the residuals:

$$WSS(\beta_k) = \sum_i \gamma_{i,k}(y_i - (\beta_{k,0} + \beta_{k,1}x_{i,1} + ... + \beta_{k,p}x_{i,p}))^2.$$

Using the value of β_k, the updated σ_k would be:

$$\sigma_k = \frac{\sum_i \gamma_{i,k}(y_i - \hat{y}_i^{(k)})^2}{\sum_i \gamma_{i,k}}$$

Intuitively, μ_k shows us where in $\mathbb{R}^p$ the center of each subgroup k resides. The further a data point is from the center, the lower its probability to belong to the subgroup. The covariance matrix Σ_k captures the spread of the subgroup in the space $\mathbb{R}^p$ relative to the center μ_k. Regression coefficient β_k gives low weight to outliers (i.e., is a *weighted* regression) and captures the relationship between X and Y near the center of the subgroup. Parameter σ_k tells us about the variance of the residuals over fitted regression line, and ω_k tells us about the importance of each subgroup.

3.3 Prediction

For test data x, the predicted outcome is the weighted average of the predicted outcomes for all components. The weights capture the uncertainty about which component the test data belongs to. Using equation 3, the best prediction of outcome for component k would be $\hat{Y}^{(k)}$, which is the mean of *conditional outcome value* of the component k:

$$\hat{y} = \sum_{k=1}^{K} \omega_k \times (\beta_{k,0} + \beta_{k,1}x_1 + ... + \beta_{k,p}x_p) \tag{7}$$

The green solid line in figure 1 represents the predicted outcome $\hat{y}$ as function of x. The solid green line, is weighted average over dashed-green lines.

4 Data

We apply our method to a synthetic and real-world data sets, including large social data sets described below.

The **Synthetic** data consists of two subgroups, with the same mean on x, but different variances, as shown in Figure 2. The variance of x for the top component is 600 and for the bottom component is 100. The number of data

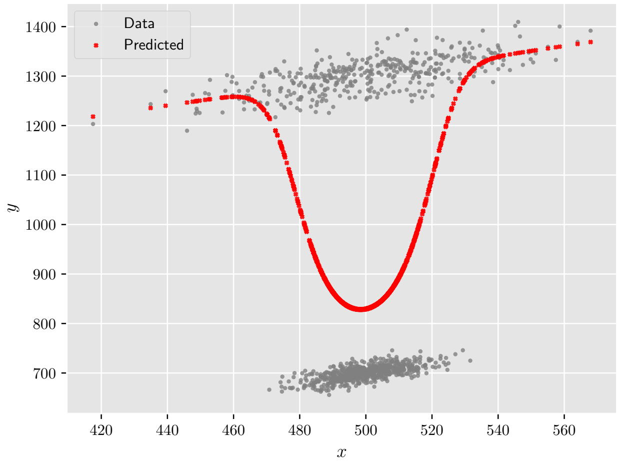


Figure 2: Synthetic data with two components centered on $x = 500$, but with different variances. Gray (lighter) points are data, and red (darker) points are predicted outcomes made by our method.

points in bottom component is $3,000$ and in top component $2,000$. The y value for the bottom component is $y = 200+x$ and the top component it is $y = 800+x$, with 20 as variance of residual.

The **Metropolitan** data comes from a study of emotions expressed through Twitter messages posted from locations around Los Angeles County (Lerman et al., 2016). The data contains over six million geo-referenced tweets from 33,000 people linked to US census tracts through their locations. The demographic and socioeconomic characteristics of tracts came from the 2012 American Fact Finder.[2] A tract is a small region that contains about 4,000 residents on average. The tracts are designed to be relatively homogeneous with respect to socioeconomic characteristics of the population, such as income, residents with bachelors degree or above, and other demographics. Of more than 2000 tracts within this Los Angeles County, 1688 were linked to tweets.

Emotional *valence* was estimated from the text of the tweets using Warriner, Kuperman, and Brysbaert (WKB) lexicon (Warriner et al., 2013). The lexicon gives valence scores—between 1 and 9—which quantify the level of pleasure or happiness expressed by a word, for 14,000 English words. Since valence is a proxy for the expressed happiness, the data allows us to investigate the social and economic correlates of happiness. We use valence as the outcome in data analysis.

The **Wine Quality** data combines two benchmark data sets from UCI repository[3] related to red and white wines (Cortez et al., 2009). The data contains records about 4898 white wine and 1599 red wine samples of wine

[2] http://factfinder.census.gov/
[3] https://archive.ics.uci.edu/ml/datasets/wine+quality

quality and concentrations of various chemicals. *Quality* is a score between 0 and 10, and is typically around 5. Our goal is to model wine quality based on physicochemical tests. Note that the type of wine (red or white) is only used to evaluate the learned components.

The **New York City Property Sale** data[4] contains records of every building or building unit (such as an apartment) sold in the New York City property market over a 12-month period, from September 2016 to September 2017. We removed all categorical variables like *neighborhood* and *tax class*. We use each property's *borough* to study relevance of the components to different neighborhoods in NYC; however, we do not use it in analysis. The outcome variable is *sale price*, which is in million dollars, and it ranges from 0 to $2,210$.

The **Stack Overflow** dataset contains data from a question-answering forum on the topic of computer programming. Any user can ask a question, which others may answer. We used anonymized data representing all answers to questions with two or more answers posted on Stack Overflow from August 2008 until September 2014.[5] We created a user-focused data set by selecting at random one answer written by each user and discarding the rest. To understand factors affecting the length of the answer (the outcome, which we measure as the number of *words* in the answer), we use features of answers and features of users. Answer features include the number of *hyperlinks*, and *lines of code* the answer contains, and its *Flesch readability* score (Kincaid et al., 1975). Features describing answerers are their *reputation*, *tenure* on Stack Overflow (in terms of *percentile* rank) and experience, i.e., the total *number of answers* written during their tenure. We also use activity-related features, including *time since previous answer* written by the user, *session length*, giving the number of answers user writes during the session, and *answer position* within that session. We define a session as a period of activity without a break of 100 minutes or longer. Features *acceptance probability*, *reputation* and *number of answers* are only used to evaluate the learned components.

4.1 Preprocessing

When variables are linearly correlated, regression coefficients are not unique, which presents a challenge for analysis. To address this challenge, we used Variance Inflation Factor (VIF) to remove multicollinear variables. We iteratively remove variables when their VIF is larger than five. For example, in the Metropolitan data, this approach reduced the number of features from more than 40 to six, representing the number of residents within a tract who are $White$, $Black$, $Hispanic$ and $Asian$, as well as percent of adult residents with $Graduate$ degree or with incomes $Below\ Poverty$ line. Table 1, represents information about all data sets.

[4] https://www.kaggle.com/new-york-city/nyc-property-sales

[5] https://archive.org/details/stackexchange

Dataset	*records*	*features*	*after VIF*	*outcome*
Synthetic	$5,000$	1	1	Y
Metropolitan	$1,677$	42	6	Valence
Wine Quality	$6,497$	11	5	Quality
NYC	$36,805$	7	4	Sale Price
Stack Overflow	$372,321$	13	5	Answer Length

Table 1: Data sets and their characteristics.

4.2 Finding Components

As first step in applying *DoGR* to data for qualitative analysis, we need to decide on the number of components. In general, finding the appropriate number of components in clustering algorithms is difficult and generally requires ad hoc parameters. For Gaussian Mixture Models, the *Bayesian information criterion (BIC)* is typically used. We also use BIC with $k \times (p^2 + 2p + 3)$ parameters, where k is number of components and p is number of independent variables. Based on the BIC scores, we choose five components for *Metropolitan* data. In our explorations, using three or six components gave qualitatively similar results, but with some of the components merging or splitting. With the same procedure, the optimal number of components for *Wine Quality* and *NYC* and *Stack Overflow* data is four.

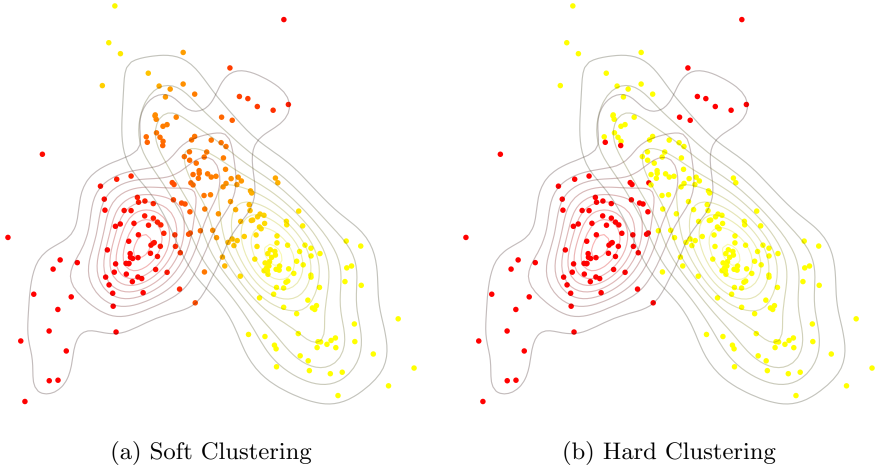
5 Results

Due to the unique interpretability of our method, we first use it to describe meaningful relationships between variables in the real-world data ("Qualitative Results"). We then show how it compares favorably to competing methods ("Quantitative Results").

5.1 Qualitative Results

We use a *radar chart* to visualize the components discovered in the data. Each colored polygon represents mean of the component, μ, in the feature space. Each vertex of the polygon represents a feature, or dimension, with the length giving the coordinate of μ along that dimension. For the purpose of visualization, each coordinate value of μ was divided by the largest coordinate value of μ across all components. The maximum possible value of each coordinate after normalization is one. The mean value of the outcome variable within each component is shown in the legend, along with 95% confidence intervals.

Our method also estimates the regression coefficients, which we can compare to MLR. We computed the p-value assuming coefficients in *DoGR* are equal to *MLR*. If β_0 and β_1 are two coefficients to compare, and $\sigma_{\bar{\beta_0}}$ and $\sigma_{\bar{\beta_1}}$ are the *standard errors* of each coefficient, then the z-score is $z = \frac{\beta_0 - \beta_1}{\sqrt{\sigma^2_{\bar{\beta_0}} + \sigma^2_{\bar{\beta_1}}}}$, from



(a) Soft Clustering

(b) Hard Clustering

Figure 3: The difference between hard and soft clustering for data analysis. Assuming we have two clusters (red and yellow); in hard clustering, the uncertainty of cluster assignment is not tractable in data analysis phase (e.g. studying the coefficients of the independent variables), since all the data-points have one of the main cluster colors (b). However, having the soft-clustering, we can get the approximated coefficient for each individual data-point (like difference range of orange in (a)), separately.

which we can easily infer the p-value assuming a normal distribution of errors (Paternoster et al., 1998).

We demonstrate the power of $DoGR$ to give novel insights into data. While other methods exist for disaggregating heterogeneous data, most partition the data into disjoint groups, using hard assignment of data-points to clusters, and then fit regression lines to each group separately. The main difference between $DoGR$ and these other methods is that our method is more flexible in giving information for analytical purposes. Figure 3 shows schematic representation of the *coefficients* in heterogeneous data. Each latent confounder has a unique coefficient and we represent them using one of two colors: red and yellow. Each data-point could be a member of any cluster at the same time (with different membership values), and as a result, each data-point could have a unique coefficient. The hard clustering methods are not able to offer this difference in coefficients. In Figure 3b, each data-point has a fixed coefficient (red and yellow), while in Figure 3a, each individual data-point has different coefficient based on their relative position from the center of the clusters. In the sections below, we apply $DoGR$ to a variety of heterogeneous data and show the insights the method produces.

5.1.1 Metropolitan

Figure 4 visualizes the components of the *Metropolitan* data and reports regression coefficients for two variables. Interestingly, the data is disaggregated along ethnic lines, perhaps reflecting segregation within the Los Angeles County (Figure 5 represents the Los Angeles County tracts colored based the component). The *Orange* component consists of census tracts with many highly educated white residents. It also has highest valence (5.86), meaning that people post the happiest tweets from those tracts. The *Purple* component, is ethnically diverse, with large numbers of white, Asian and Hispanic residents. It's valence (5.80) is only slightly lower than of the *Orange* component. The *Red* component has largely Asian and some Hispanic neighborhoods that are less well-off, but with slightly lower valence (5.76). The *Blue* and *Green* components represents tracts with the least educated and poorest residents. They are also places with the lowest valence (5.74 and 5.72, respectively). Looking at the regression coefficients, education is positively related to happiness across the entire population (*All*), and individually in all components, with the coefficients significantly different from *MLR* in four of the five components, *suggesting this trend was a Simpson's paradox*. However, the effect is weakest in the most and least educated components. Poverty has no significant effect across the entire population, but has a negative effect on happiness in poorest neighborhoods (*Blue*). Counter-intuitively, regression coefficients are positive for two components (*Orange* and *Red*). It appears that within these demographic classes, the poorer the neighborhood, the happier the tweets that originate from them.

5.1.2 Wine Quality

Figure 6 visualizes the disaggreagation of the *Wine Quality* data into four components. Although we did not use the type of wine (red or white) as a feature, wines were naturally disaggregated by type. The *Blue* component is almost entirely (98%) composed of high quality (average quality = 6.02) white wines. All the wines in the *Orange* component are white (average quality = 5.91), while the *Green* component is composed mostly (85%) of red wines with average quality 5.60. The lowest quality (*Red*) component, with average quality equal to 5.36, contains a mixture of red (43%) and white (57%) wines. In other words, we discover that in higher-quality wines, the type of wine can be determined with high accuracy simply based on its chemical components. Low quality wines appear less distinguishable based on their chemicals. We find *Chlorides* have a negative impact on quality of wine in all components (not shown), *Sugar* has positive impact on high quality white wines and red wines, and negative impact on low quality wines. Surprisingly, *Free Sulfur Dioxide* has a positive impact on high quality white wines, but a negative impact in other components. These are findings that may be important to wine growers, and capture subtleties in data that commonly-used *MLR* does not.

5.1.3 NYC Property Sales

The mean sale price of all properties in *New York City (NYC)* property sales data is $1.31 Million. The mean price, however, hides a large heterogeneity in the properties on the market. Our method tamed some of this heterogeneity by identifying four components within the sales data. Figure 7 shows that these components represent large commercial properties (*Orange*), large residential properties (*Green*), mixed commercial/residential sales (*Red*), and single-unit residential properties (*Blue*).

Table 2 shows what percentage of each component is made up of New York City's five boroughs. Large commercial properties (*Orange* component), such as office buildings, are located in Brooklyn and Manhattan, for example. These are the most expensive among all properties, with average price of more than $12 Million. The next most expensive type of property are large residential buildings (*Green* component)—multi-unit apartment buildings. These are also most likely to be located in Manhattan and Brooklyn. Small residential properties (*Blue* component)—most likely to be single family homes—are the least expensive, on average half a million dollars, and most likely to be located in Brooklyn and Queens, with some in Staten Island.

Component	Manhattan	Bronx	Brooklyn	Queens	Staten
Orange	**17**%	20%	**41**%	17%	5%
Green	**41**%	22%	**26**%	10%	1%
Red	8%	9%	**65**%	15%	3%
Blue	0%	13%	**37**%	**33**%	16%
All	3%	13%	40%	30%	14%

Table 2: DoGR components, and the boroughs that make up each component (rows might not add up to 100% due to rounding).

Regressions in this data set show several instances of Simpson's paradoxes. Property price in population-level data increases as a function of the number of *residential units*. In disaggregated data, however, this is true only for the *Green* component, representing apartment buildings. Having more residential units in smaller residential properties (*Red* and *Blue* components) lowers their price. This could be explained by smaller multi-unit buildings, such as duplexes and row houses, being located in poorer residential areas, which lowers their price compared to single family homes. Another notable trend reversal occurs when regressing on lot size (*land square feet*). As expected, there is a positive relationship in the *Blue* component with respect to lot size, as single family homes build on bigger lots are expected to fetch higher prices in New York City area. However, the trends in the other components are negative. This could be explained the following way. As land becomes more expensive, builders are incentivized to build up, creating multi-story apartment and office buildings. The more expensive the land, the taller the building they build. This is confirmed by the positive relationship with *gross square feet*, which are strongest for the *Orange* and *Green* components. In plain words, these components represent the tall buildings with small footprint that one often sees in New York City.

5.1.4 Stack Overflow

As our last illustration, we apply *DoGR* to Stack Overflow data to answer the question how well we can predict the length of the answer a user writes, given the features of the user and the answer.

DoGR splits the data into four clusters, as shown in Figure 8. The *Green* and *Red* components contain most of the data, with 47% and 39% of records respectively. The radar plot shows the relative strength of the features in each component, while the table above the plot shows features characterizing each discovered group. Except for *Percentile tenure*, these features were not used by *DoGR* and are shown to validate the discovered groups.

The *Orange* component (5% of data) contains very active (longer *Session Length*) users, who meticulously document their answers with many *lines of code* and *URLs*, so we can label them "power users". These are among the longest (high *Words*) and more complex (low *Readability*) answers in the data, and they also tend to be high quality (high *Acceptance Probability*). Surprisingly this group has newer users (lower *Percentile Tenure*), but they have high reputation (*Answerer Reputation*) and wrote more answers previously (higher *Number of Answers*). These users, while a minority, give life to Stack Overflow and make it useful for others. *Orange* component users have more *code lines* within shorter answers. This is in contrast to other groups, which tend to include more lines of code within longer answers. The brevity of *Orange* users when documenting code is an example of a trend reversal.

Another interesting subgroup is the *Blue* group (8% of data), which is composed of "veterans" (high *Percentile Tenure*), who write easy-to-read answers (high *Readability*) that are documented with many *URLs.* These users have a relatively high *reputation*, but they are selective in the questions they answer (lower *Number of Answers* than for the *Orange* users). Interestingly, tenure (*Percentile*) does not have an effect on length of the answer for these *Blue* users, while it has positive effect in other groups (i.e., more veteran users write longer answers). Negative effect of *URLs* on the length of the answer suggests that these users use *URL*s to refer to existing answers.

The *Red* (39%) and *Green* (47%) components contain the vast majority of data. They are similar in terms of user *reputation*, tenure (*Percentile*) and experience (*Number of Answers*), as well as the quality of the answers they write (*Acceptance Probability*), with the *Red* component users scoring slightly higher on all the measures. The main difference is in the answers they write: *Red* users do not include *URLs* in their answers, while *Green* users do not include *code.* Another difference between these groups is that *Green* users have longer answers than *Red* users, but as their answers become longer, they also become more difficult to read (lower *Readability*). In contrast, longer answers by *Red* users are easier to read. Overall, we find intuitive and surprising results from data that are largely due to DoGR's interpretability.

5.2 Quantitative Results

We compare the performance of *DoGR* to existing state-of-the-art methods for disaggregated regression: the three variants of CLR, WCLR (da Silva and de Carvalho, 2017), FWCLR (da Silva and de Carvalho, 2018), and GMR (Sung, 2004). We use *MLR*, which does not disaggregate data, as baseline.

5.2.1 Prediction Performance

For the prediction task, we use 5 × 5-fold nested cross validation to train the model on four folds and make predictions on the out-of-sample data in the fifth fold. As hyperparameters, we used $k = 1$–6 as potential number of components for all methods. For *WCLR and FCWCLR*, we set $\alpha = \{0.001, 0.01, 0.1, 10.0, 100.0,$
$1000.0\}$, and for *FWCLR* we set $m = \{1.1, 1.5, 2.0, 3.0\}$. We use grid search to find best hyperparameters. Table 3 presents results on our datasets and synthetic data. To evaluate prediction quality, we use *Root Mean Square Error (RMSE)* and *Mean Absolute Error (MAE).* When comparing the quality of cross-validated predictions across methods, we use the Kruskal-Wallis test, a non-parametric method to compare whether multiple distributions are different (Kruskal and Wallis, 1952). When distributions were significantly different, a pairwise comparisons using the Tukey-Kramer test was also done as post-hoc test, to determine if mean cross-validated quality metrics were statistically different (McDonald, 2014). Prediction results are shown in table 3. We use $*$ to indicate statistically significant ($p < 0.05$) differences between our method and other methods. Bolded values differ significantly from others or have lower standard deviation. Following (da Silva and de Carvalho, 2018), we use lower deviation to denote more consistent predictions.

WCLR and *FWCLR* do not perform well in **Synthetic** data, because variances of the two components are different, which these methods do not handle. This is a problem that exists in an arbitrary number dimensions; we observe in higher dimensions the gap between performance of *FWCLR* and our method *increases.*

For **Metropolitan** data, there is no statistically significant difference between methods: all cluster-based regression methods outperform *MLR* on the prediction task. This shows that any method that accounts for the latent structure of data helps improve predictions. For **Wine Quality** data, while the null hypothesis of equal performance of *GMR* and *DoGR* cannot be rejected, our method has a smaller standard deviation in both datasets. We were not able to successfully run *GMR* and *WCLR* on **NYC** data, due to exceptions (returning null as predicted outcome) and extremely long run time, respectively. It took three days for *FWCLR*, and four hours for our method to finish running on *NYC* data. Our method has significantly lower *MAE* compared to *FWCLR*. We were also not able to successfully run *GMR* and *WCLR* on **Stack Overflow** data for the same reasons. It took six days for *FWCLR* to run one round of cross-validation, after which we stopped it.

Method	RMSE (±σ)	MAE (±σ)
Synthetic		
MLR	294.88 (± 1.236)*	288.35 (± 0.903)*
WCLR	261.14 (± 3.370)*	232.76 (± 2.682)*
FWCLR	261.27 (± 4.729)*	233.05 (± 3.772)*
GMR	257.36 (± 4.334)	219.15 (± 3.567)
DoGR	**257.32 (± 3.871)**	**219.11 (± 3.106)**
Metropolitan		
MLR	0.083 (± 0.0061)	0.062 (± 0.0033)
WCLR	0.083 (± 0.0029)	0.062 (± 0.0024)
FWCLR	**0.082 (± 0.0044)**	**0.061 (± 0.0021)**
GMR	0.083 (± 0.0043)	0.061 (± 0.0023)
DoGR	0.083 (± 0.0052)	0.061 (± 0.0031)
Wine Quality		
MLR	0.83 (± 0.018)*	0.64 (± 0.015)*
WCLR	0.83 (± 0.013)*	0.64 (± 0.011)*
FWCLR	0.80 (± 0.013)*	0.63 (± 0.009)*
GMR	0.79 (± 0.017)	0.62 (± 0.014)
DoGR	**0.79 (± 0.014)**	**0.62 (± 0.011)**
NYC		
MLR	13.36 (± 7.850)	2.20 (± 0.064)*
FWCLR	13.14 (± 7.643)	1.76 (± 0.321)*
DoGR	**11.88 (± 9.109)**	**1.40 (± 0.222)**
Stack Overflow		
MLR	60.69 (± 1.118)	37.74 (± 0.152)
FWCLR	60.47 (± 0.960)	37.25 (± 0.794)
DoGR	60.68 (± 1.298)	37.62 (± 0.314)

Table 3: Results of prediction on five data-sets. Asterisk $*$ indicates results that are significantly different from our method (p-value < 0.05). The bolded results have smaller standard deviation among the best methods with same mean of error.

Therefore, the mean reported in the table for *FWCLR* is the average of five runs, while for *MLR* and *DoGR* it is the average of 25 runs. Due to this discrepancy, we did not highlight any of the results in bold.

5.2.2 Run Time

To compare the run time of all algorithms, we performed one round of cross validation (not nested) for each method. The same machine (4-GB RAM, 3.0-GHz Intel CPU, Windows OS) was used for time calculation. The available code for *WCLR* and *FWCLR* methods are in *R* while the other methods are written in *Python*. Table 4 presents the run time in minutes. The slowest method is *WCLR*, while the fastest one is *MLR*. *WCLR* and *FWCLR* are sensitive to size of data, perhaps due to the many hyperparameters they need to tune. To find the best hyperparameters for *GMR* and *DoGR*, we ran the methods 6 times, while *WCLR* requires 36 runs and *FWCLR* 144 runs. Beside *NYC* and *Stack Overflow* datasets for which exceptions occur in *GMR*, the run time of *DoGR* is twice that of *GMR*. The reason *GMR* throws exception could be because of a singular covariance matrix. While our method is similar in spirit to *GMR*, out method is more stable, as shown on the *NYC* and *Stack Overflow* data. In addition, our method is interpretable, as it directly computes regression coefficients, while *GMR* represents relationships between variables via the covariance matrix. Covariance values are not guaranteed to have the same sign, let alone magnitude, as regression coefficients. Regression is therefore necessary to understand the true relationships between variables. Mathematically, *GMR* and *unweighted* regression can be converted to one another using linear algebra. It is not clear, however, whether the equivalence also holds for weighted regression.

Dataset	WCLR	FWCLR	GMR	*DoGR*
Synthetic	6.46	0.68	**0.76**	1.4
Metropolitan	37	9	**2**	4
Wine Quality	180+	36	**9**	16
NYC	600+	232	failed	**10**
Stack Overflow	-	7200+	failed	**170**

Table 4: Run time (in *minutes*) of one round of cross-validation (non-nested). *MLR* took less than a minute for all datasets. 'failed' indicates that the method could not run on the data due to exceptions. We time out WCLR exceeding sufficient amount of time to show the order of performance. **Bold** numbers indicate the fastest algorithm.

6 Discussion

In this chapter, we introduce *DoGR*, which softly disaggregates data by latent confounders. Our method retains the advantages of linear models, namely their interpretability, by reporting regression coefficients that give meaning to trends. Our method also discovers the multidimensional latent structure of data by partitioning it into subgroups with similar characteristics and behaviors. While alternative methods exist for disaggregating data, our approach is unique because it produces interpretable regressions that are computationally efficient.

We demonstrated the utility of our approach by applying it to real-world data, from data on wines to data on answers in question-answering website. We show that our method identifies meaningful subgroups and trends, while also yielding new insights into the data. For example, in the wine data set, it correctly separated high quality red wines from white, and also discovered two distinct classes of high quality white wines. In Stack Overflow data, it helped us identify important users like “veterans” and “power users.”

There are a few ways forward to improve our method. Currently, it is applied to continuous variables, but it needs to be extended to categorical variables, often seen in social science data. Moreover real data is often non-linear; therefore, our method needs to be extended to non-linear models beyond linear regression. In addition, to improve prediction accuracy, our method could include regularization parameters, such as ridge regression or LASSO. However, already in its current form, *DoGR* can yield new insights into data.

References

Alipourfard, N., Fennell, P. G., and Lerman, K. (2018a). Can you trust the trend?: Discovering simpson's paradoxes in social data. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 19–27. ACM.

Alipourfard, N., Fennell, P. G., and Lerman, K. (2018b). Using simpson's paradox to discover interesting patterns in behavioral data. In *Twelfth International AAAI Conference on Web and Social Media*.

Bareinboim, E. and Pearl, J. (2016). Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113:7345–7352.

Blyth, C. R. (1972). On simpson's paradox and the sure-thing principle. *Journal of the American Statistical Association*, 67(338):364–366.

Calderon, D., Juba, B., Li, Z., and Ruan, L. (2018). Conditional linear regression. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chuang, J. S., Rivoire, O., and Leibler, S. (2009). Simpson's paradox in a synthetic microbial system. *Science*, 323(5911):272–275.

Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553.

da Silva, R. A. and de Carvalho, F. d. A. (2017). On combining clusterwise linear regression and k-means with automatic weighting of the explanatory variables. In *International Conference on Artificial Neural Networks*, pages 402–410. Springer.

da Silva, R. A. and de Carvalho, F. d. A. (2018). On combining fuzzy c-regression models and fuzzy c-means with automated weighting of the explanatory variables. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE.

DeSarbo, W. S. and Cron, W. L. (1988). A maximum likelihood methodology for clusterwise linear regression. *Journal of classification*, 5(2):249–282.

Fabris, C. C. and Freitas, A. A. (2000). Discovering surprising patterns by detecting occurrences of simpson's paradox. In *Research and Development in Intelligent Systems XVI*, pages 148–160. Springer.

Fox, J. (1997). *Applied regression analysis, linear models, and related methods.* Sage Publications, Inc, Thousand Oaks, CA.

Ghahramani, Z. and Jordan, M. I. (1994). Supervised learning from incomplete data via an em approach. In *Advances in neural information processing systems*, pages 120–127.

Kievit, R., Frankenhuis, W. E., Waldorp, L., and Borsboom, D. (2013). Simpson's paradox in psychological science: a practical guide. *Frontiers in psychology*, 4:513.

Kincaid, J. P., R. P. Fishburnea, J., Rogers, R. L., and Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, U.S. Navy.

Kruskal, W. H. and Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621.

Lerman, K., Arora, M., Gallegos, L., Kumaraguru, P., and Garcia, D. (2016). Emotions, demographics and sociability in twitter interactions. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6446–6456. Curran Associates, Inc.

Manwani, N. and Sastry, P. (2015). K-plane regression. *Information Sciences*, 292:39–56.

McDonald, J. (2014). *Handbook of Biological Statistics.* Sparky House Publishing, Baltimore, 3rd edition.

McLean, R. A., Sanders, W. L., and Stroup, W. W. (1991). A unified approach to mixed linear models. *The American Statistician*, 45(1):54–64.

Minchev, I., Matijevic, G., Hogg, D., Guiglion, G., Steinmetz, M., Anders, F., Chiappini, C., Martig, M., Queiroz, A., and Scannapieco, C. (2019). Yule-simpson's paradox in galactic archaeology. *arXiv preprint arXiv:1902.01421.*

Pashler, H. and Wagenmakers, E. (2012). Editors' introduction to the special section on replicability in psychological science: A crisis of confidence? *Perspectives on Psychological Science*, 7(6):528–530.

Paternoster, R., Brame, R., Mazerolle, P., and Piquero, A. (1998). Using the correct statistical test for the equality of regression coefficients. *Criminology*, 36(4):859–866.

Ranganath, R. and Perotte, A. J. (2018). Multiple causal inference with latent confounding. *ArXiv preprint: 1805.08273*.

Smaldino, P. E. and McElreath, R. (2016). The natural selection of bad science. *Royal Society Open Science*, 3(9):160384.

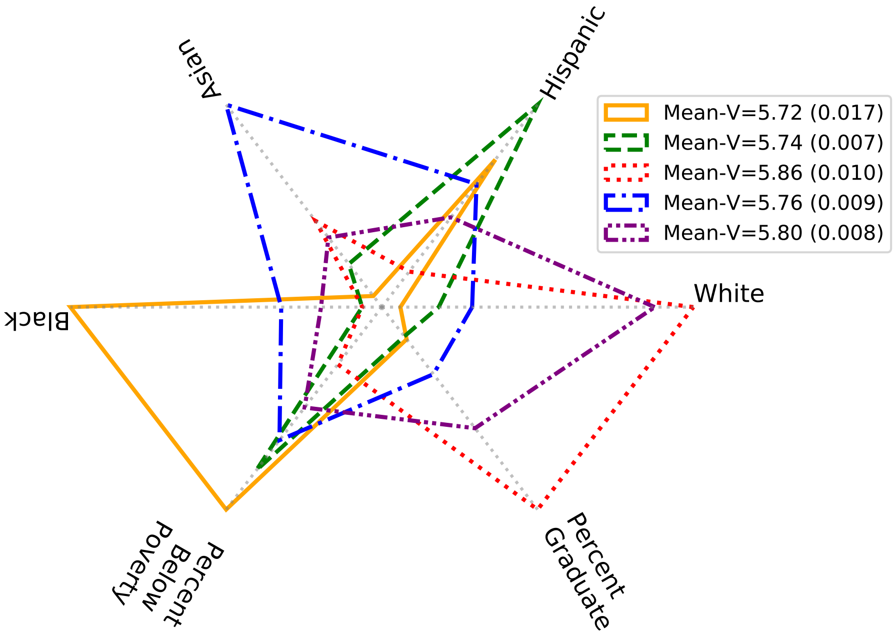
Späth, H. (1979). Algorithm 39 clusterwise linear regression. *Computing*, 22(4):367–373.

Sung, H. G. (2004). *Gaussian mixture regression and classification*. PhD thesis, Rice University.

Wang, Y. and Blei, D. M. (2018). The blessings of multiple causes. *ArXiv preprint: 1805.06826*.

Warriner, A. B., Kuperman, V., and Brysbaert, M. (2013). Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.

Winter, B. (2013). A very basic tutorial for performing linear mixed effects analyses. *arXiv preprint arXiv:1308.5499*.



			% Below Poverty	*% Graduate*
	mean	*std*	β	β
All	5.78	0.098	0.0000	0.0028
Orange (dotted)	5.86	0.092	0.0063***	0.0016*
Purple (dot-dot-dash)	5.80	0.077	0.0002	0.0030
Red (dash-dot)	5.76	0.080	0.0018***	0.0050***
Green (dashed)	5.74	0.073	0.0001	0.0056***
Blue (solid)	5.72	0.117	−0.0008***	0.0001***

* p-value$\leq$ 0.05; *** p-value$\leq$ 0.001

Figure 4: Disaggregation of the Metropolitan data into five subgroups. The radar plot shows the relative importance of a feature within the subgroup. The table report regression coefficients for two independent variables *Percent Below Poverty* and *Percent Graduate* for Multivariate Regression (MLR) of aggregate data and separately for each subgroup found by our method.

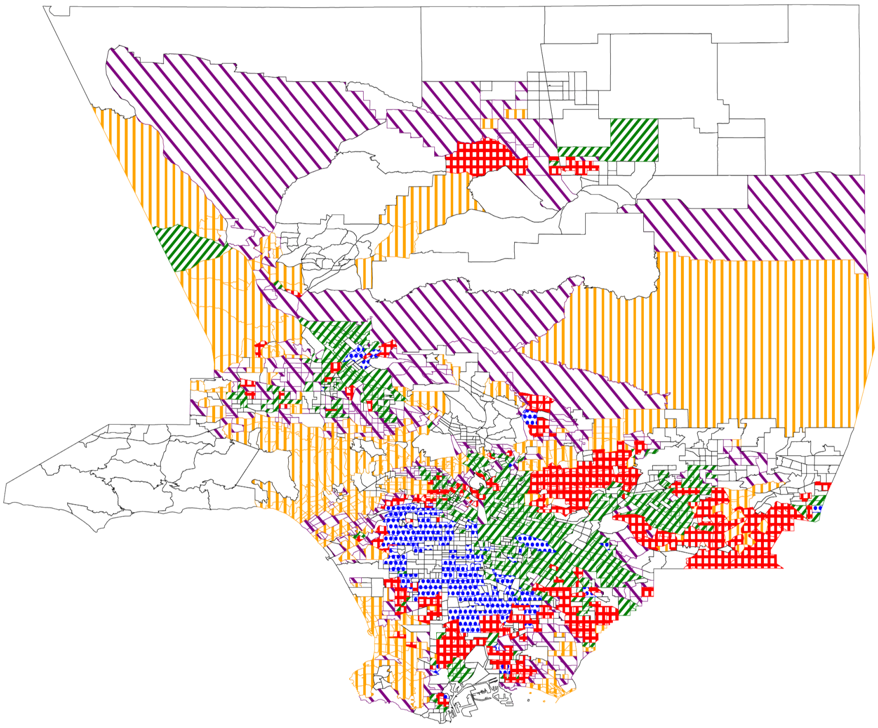
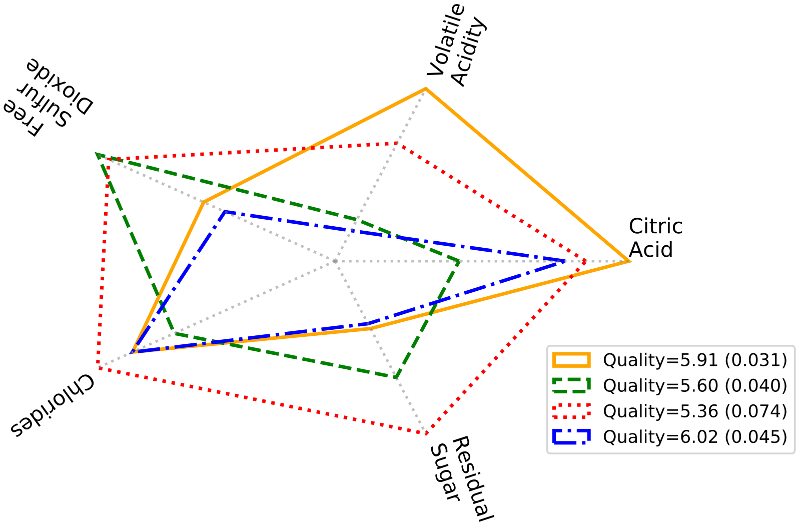
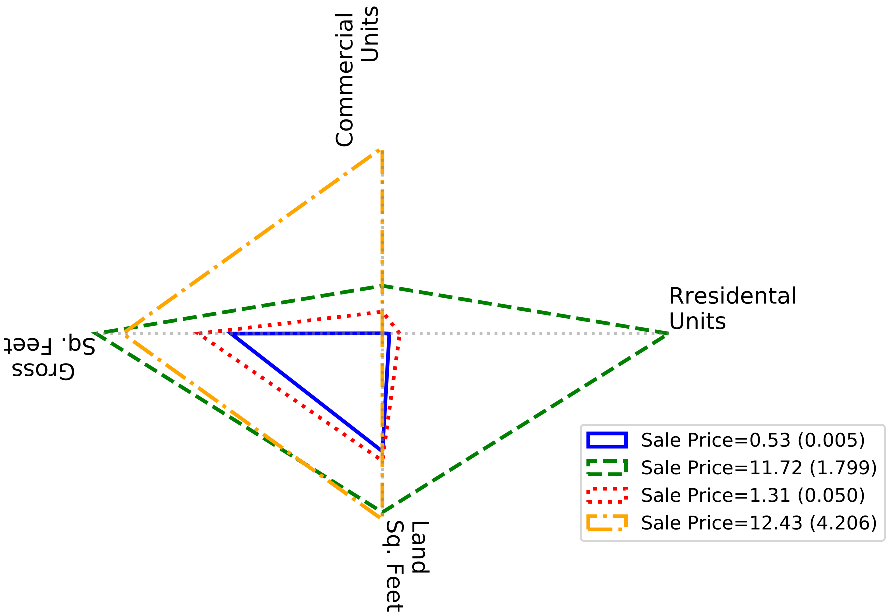


Figure 5: Map of Los Angeles county. The color represent the component for the Tract. Colors match the colors of Figure 4. Majority Asian Neighbourhoods is red (++), while majority Black neighborhoods is blue (*oo*). Affluent areas are orange (‖), while ethnically diverse areas are purple (\\\\). Majority Hispanic (East Los Angeles) neighborhoods is green (//).



	Wine Quality		*Citric Acid*	*SO2*	*Sugar*
	mean	*std err*	β	β	β
All			0.1044	−0.0008	−0.0175
Blue (dash-dot)	6.02	0.023	0.4299	0.0173***	0.4272***
Orange (solid)	5.91	0.016	−0.1545**	−0.0001	−0.0156
Green (dashed)	5.6	0.020	−0.0968	−0.0024	0.3227***
Red (dotted)	5.36	0.037	0.3039**	−0.011***	−0.004***
* p-value≤ 0.05; *** p-value≤ 0.001					

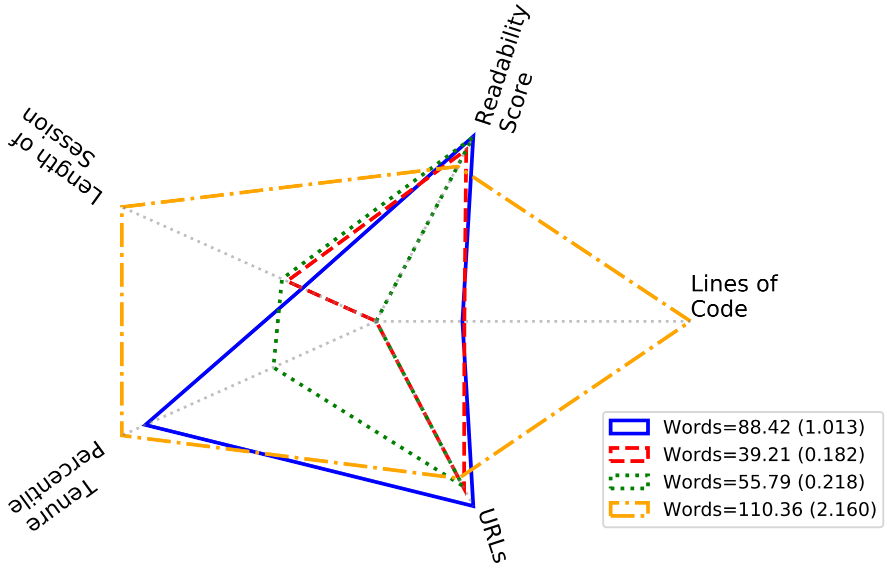
Figure 6: The value of the center of each component (μ) for four components of the Wine Quality data. The *Blue* and *Orange* components are almost entirely white wines, and the *Green* component is composed mostly (85%) of red wines. The lowest quality and smallest—*Red*—component is a mixture of red (43%) and white (57%) wines.



	Sale Price		*Resid. Units*	*Commer. Units*	*Gross Sq.Feet*	*Land Sq.Feet*
	mean	*std err*	β	β	β	β
All	1.31	0.079	0.036	0.039	2.681	0.595
Orange (dash-dot)	12.43	2.143	0.000	0.218	8.842	−1.956
Green (dashed)	11.72	0.916	0.078	1.008	5.329	−6.652
Red (dotted)	1.31	0.026	−0.142	−0.900	0.591	−0.348
Blue (solid)	0.53	0.004	−0.039	0.000	0.124	0.089
all significant: p-value $\leq$ 0.001						

Figure 7: The value of center (μ) for four components of NYC data. The numbers in legends are average price of the component's real estate and is in millions of dollars.

	Size	Acceptance Probability	Answerer Reputation	Num. of Answers	Percentile Tenure
Red	39%	0.26	338.17	16.02	0.43
Green	47%	0.24	314.35	14.56	0.42
Blue	8%	0.33	492.19	20.65	0.47
Orange	5%	0.32	784.62	40.21	0.40



	Words		*Codes*	*Readab.*	*Percnt.*	*URLs*
	mean	*std*	β	β	β	β
All	54.91	0.102	0.29	0.048	5.61	17.11
Red (dashed)	39.21	0.093	0.43	0.112	6.09	0
Green (dotted)	55.79	0.111	0	−0.175	2.64	3.62
Blue (solid)	88.42	0.517	0.17	−0.467	*-0.22*	−86.20
Orange (dash-dot)	110.36	1.102	−0.20	0.341	20.45	12.87

* *Italicized effects are not significant*

Figure 8: Disaggregation of Stack Overflow data into subgroups. The outcome variable is length of the answer (number of words). The radar plot shows the importance of each features used in the disaggregation. The top table shows average values of validation features, while the bottom table shows regression coefficients for the groups.