Canonical Least Squares Clustering on Sparse Medical Data

Igor Gitman
Machine Learning Department
Carnegie Mellon University
igitman@andrew.cmu.edu

Jieshi Chen
Auton Lab
Carnegie Mellon University
jieshic@andrew.cmu.edu

Artur Dubrawski
Auton Lab
Carnegie Mellon University
awd@cs.cmu.edu

Abstract

We explore different applications of Canonical Least Squares (CLS) clustering on a corpus of sparse medical claims from a particular health insurance provider. We find that there are several reasons why most conclusions based on CLS clusters might be misleading, especially when the data is significantly sparse. We illustrate these findings by performing a number of synthetic experiments with the most focus on the sparsity issue since it has not been explored in the literature before. Based on the insights from the synthetic experiments we show how CLS clustering can be potentially applied to identify hospital peer-groups: hospitals that share similar operational characteristics. In addition we demonstrate that CLS clustering can be used to improve prediction results for the patients length of stay in the hospitals.

1 Introduction

Data clustering can be useful in various applications. These applications can be roughly divided into two groups: data analysis applications and prediction applications. The goal of the first group is to find some new properties of the data by examining obtained clusterings. The goal of the second group is to improve results of prediction models by adding cluster labels as features. Regardless of the application, the goal of the clustering algorithm is to find groups of objects that are similar according to some predefined criteria. One popular choice of similarity criteria is L_2 distance in feature space. In that case such well-known algorithms as k-means [21] or some form of hierarchical clustering [23] can be used. When feature space contains some dependent or target variable (e.g. produced by linear combination of other features), it might be reasonable to seek the groups of objects that have the same relation between dependent and independent variables (e.g. the same linear regression coefficients). In that case, some form of cluster-wise linear regression (CLR) can be used, e.g. [27] or [9]. When there are multiple dependent variables and one seeks to find clusters with different correlation patterns, canonical correlation analysis (CCA) clustering [10] can be applied. Recently, there has been proposed a canonical least squares (CLS) clustering method [18] that achieves similar goal to CCA, but is more robust and its results usually have better interpretations.

In this paper we explore potential applications of CLS to a corpus of medical claims of patients from a particular health insurance provider. For each claim there are 87 different fields characterizing that claim. These fields include: patients id, claim id, provider id, admittance/discharging dates, patient age, gender, diagnosis related group (DRG) and other. We first preprocess this data filtering outliers and not useful features and then produce two different aggregations: claim-level data (with each object corresponding to one claim) and hospital-level data (with each object corresponding to one provider, which is usually a hospital). More details about dataset and feature selection/aggregation processes are given in section 3.

One notable feature extracted from this data is patient length of stay in the hospital (or average length of stay for a certain DRG in case of the hospital-level data). This feature can be used as a measure

of quality of care for the hospital or claim. There are many business applications that could benefit from finding groups of claims or hospitals that have similar correlation patterns with quality of care. One example of such application is finding comparable groups of hospital or "peer groups" that have similar operational characteristics and similar length of stay dependence on those characteristics. These groups can then be used to assess hospitals performance fairly by comparing them with each other only inside their peer group and not across all hospitals. On a claim level, finding clusters that have similar correlation patterns can be useful for prediction purposes. It is reasonable to assume that different claims are not homogeneous in terms of their correlation with length of stay. For example, some patients can have different reactions to certain diseases or types of treatment. Thus, identifying regression-based clusters and using different regression models for them could potentially improve the prediction accuracy for a length of stay.

In this paper we demonstrate applications of CLS towards these two goals: correlation-based data analysis and improvement of prediction results. We start by reviewing the related work for both of these problems in section 2. We describe dataset used in this paper in more details in section 3 and give a brief introduction into CLS algorithm in section 4. In section 5 we describe a number of preliminary experiments that lead us to believe that a straightforward application of CLS to a claim-level data analysis is likely to give misleading results. We identify 3 main problems:

1. Standard metrics evaluating CLS performance overestimate goodness of linear fit to the data.
2. CLS is prone to finding non-intuitive correlations that only exists in the data by chance.
3. Applying CLS on sparse data is complicated since there are multiple good solutions, with some extreme cases when CLS problem becomes ill-defined.

The first two problems were mentioned in the literature before in the work of Brusco et al [5] and Vicari et al [28]. In this paper we give a few more intuitive examples illustrating the issues as well as provide more general and practical formulations of possible solutions in sections 6.1 and 6.2. The sparsity problem has not been explored before and thus we give a detailed analysis of this problem by running a series of synthetic experiments in section 6.3. In section 7.1 we demonstrate how CLS can be used to improve traditional methods of hospital peer-groups creation. Finally, in section 7.2 we introduce two novel approaches to prediction with regression-based clustering: *predictive CLS* which combines CLS objective with k-means and uses a separate classification model to predict CLS labels at test time; and *constrained CLS*, which uses a user-defined set of constraints on some features that are known at test time and thus could be used to derive test labels. We show that these methods perform better than other models, such as linear regression, random forest [4] and k-plane: a similar regression-based clustering algorithm proposed in [22]. We summarize and conclude the paper in section 8.

2 Related work

2.1 Interpretability analysis

When there is only one dependent variable, CLS is equivalent to CLR, proposed by Spath in 1979 [27] which has been extensively studied in the literature. Hennig [12] explores the problem of identifiability of linear regression mixtures and proves a set of necessary conditions. In this work we show with simple examples and experiments that strong sparsity can also lead to non-identifiable mixtures which was not studied in the original work by Hennig. Although we do not provide any complete theoretical results it is a potential direction of future research.

The problem of CLR overestimating the goodness of linear fit was first observed by Brusco et al [5]. The authors argue that even when target variable is generated independently from other features, CLR will find a solution with surprisingly good coefficient of determination (R^2) and mean squared error (MSE). They show that MSE comprises of 2 terms: within-cluster distances and between-cluster distances. Since CLR finds regression coefficients and clusters simultaneously, it might optimize between-cluster distance only (by grouping objects with similar target values), ignoring within-cluster correlations. The authors propose a different metric (based on hypothesis testing) to assess the performance of CLR adjusted for such a good behaviour when features and targets are independent. We reinforce the analysis of Brusco et al [5] by providing exact theoretical solution for the uniformly distributed targets. We also suggest a different adjusted metric which only requires running CLR 2

times and thus is exponentially faster to compute in the general CLS case when the dimension of target variables is greater than 1.

Vicari et al [28] expands on the work of Brusco et al [5] showing that since CLR doesn't distinguish between within-cluster and between-cluster distances it can converge to a very non-intuitive solutions. The authors devise an algorithm that has different models for relations within one cluster and relations between different clusters. The final algorithm can be seen as combining CLR objective with k-means objective which has also been explored by [6], [22], [8]. We reinforce these observations with simple examples and provide a general regularized reformulation of CLS, with k-means regularization being a particular instance.

2.2 Hospital peer-groups

The traditional approaches to finding hospital peer groups usually utilize standard clustering techniques. Klastorin [17] is using hierarchical clustering to identify peer groups. Alexander et al [1] suggests to use k-means and factor analysis instead which provides a better way to understand how good the given clustering is. These techniques were tested by Zodet and Clark [31], MacNabb [20] and Kang et al [16] on the hospital data from the state of Michigan, Canada and South Korea correspondingly. Another approach to defining hospital peer groups was developed by Byrne et al [7]. The authors suggested that peer groups might not be mutually exclusive and created a nearest neighbors based algorithm for identifying the peer groups centered at each hospital.

2.3 Prediction

There are multiple approaches on how to use regression-based clustering for prediction. Kang et al [15] suggests to use fuzzy clustering with Dirichlet prior so that it would be possible to obtain labels at test time. Bagirov et al [3] applies a modification of CLR to prediction of monthly rainfall, taking weighted average of different models based on the cluster sizes. Manwani et al [22] proposes k-plane method that combines CLR with k-means and uses the closest cluster centers as labels at test time. Our predictive CLS approach is similar to k-plane method, but in order to identify cluster membership we propose to train a separate model. We demonstrate that this modification leads to crucial difference in performance. We do not provide the comparison with other prediction methods since their CLR implementations are very different from CLS.

The idea of doing constrained cluster-wise regression was first proposed in [25], however, the authors do not explore potential applications to prediction.

3 Data

The dataset used in this project consists of medical claims of patients from a particular health insurance provider. In total there are around 1 billion claims, each characterized with 87 different fields (≈ 140 Gb of the raw data). As a preprocessing step we keep only inbound (registered in a hospital), non-empty claims for 2014 year. We delete claims containing mistakes, such as claims that have multiple patients associated with them, patients that have multiple gender or birthday or claims with total length of stay ≥ 60 days. We process the data to obtain the following set of features. Numerical: age, DRG weight, mean historic length of stay per DRG, mean historic length of stay per hospital, mean historic length of stay overall. Categorical: DRG, month of admission, hospital zipcode, was it observation stay, type of stay (emergency, urgent, elective, newborn or trauma), whether patient has been in this hospital before, whether patient had this DRG before. After dropping out claims with missing values and representing categorical features in one-hot encoding we obtain a corpus with ≈ 400000 claims with 872 features each. In order to further reduce the dimensionality we drop all the binary features that have value of 1 in less than a 1000 claims (e.g. certain rare DRGs or zipcodes). After the final preprocessing step data size is $\approx 400000 \times 146$.

After described preprocessing we construct a hospital-level aggregation of this data. To do that we aggregate claims belonging to the same hospital and obtain the following featurization: mean patient age, mean number of claims per year, mean DRG weight, proportion of claims of certain type, gender proportion, mean number of observational stay claims, mean number of claims for major diagnostic categories (MDC), which is an aggregation of DRGs, and mean length of stay per MDC.

After discarding all the hospitals with less than 20 claims per year we obtain 2197 hospitals with 64 features each.

Another important factor to note about this data is that hospital-level data is dense, while claim-level data is very sparse, which complicates application of CLS as we show in section 6.3. For both datasets we consider length of stay as target features. For claim-level data it is just one number per claim, while for hospital-level data we compute length of stay per MDC and thus its dimension is 26.

4 Canonical Least Squares clustering

CLS clustering can be applied to data that consists of 2 sets of features $X \in \mathbb{R}^{n \times d_1}$ and $Y \in \mathbb{R}^{n \times d_2}$. It is aimed to find partitions of the data that maximize canonical correlations [13] between corresponding X and Y inside clusters. The first canonical correlation between two sets of features for a particular object X_i, Y_i is defined as

$$\max_{u \in \mathbb{R}^{d_1}, v \in \mathbb{R}^{d_2}} \text{Corr}(X_i^T u, Y_i^T v) \tag{1}$$

The subsequent correlation coefficients can be found by solving problem 1 with additional constraints that previously found solutions $X^T u$ and $Y^T v$ (called canonical variables) should be uncorrelated with the new pair of canonical variables. The canonical correlation problem can be solved in a closed form with eigenvalue decomposition, see, for example [11].

CLS clustering has the following parameters: number of clusters k, number of canonical variables to consider m, data matrices X and Y. When $m = 1$, CLS clustering consists of iteratively performing the following 2 steps after randomly initializing cluster assignments.

CLS step. Let $R^{(i)} \in \mathbb{R}^{n \times n}$ be a diagonal matrix with $R^{(i)}_{ll} = 1$ if point l belongs to cluster i. Then, keeping label assignment fixed, solve:

$$\min_{u_i \in \mathbb{R}^{d_1}, v_i \in \mathbb{R}^{d_2}} \left[\sum_{i=1}^{k} \left\| R^{(i)}(X u_i - Y v_i) \right\|_2^2 \right], \text{subject to } v_i^T v_i = 1 \tag{2}$$

Labeling step. Keeping CLS coefficients fixed, assign each object x_l, y_l to cluster

$$\arg\min_i (y_l^T v_i - x_l^T u_i)^2 \tag{3}$$

Note, that CLS step does not exactly find canonical correlations, since the constraints are different. However, for the first component this problem can be still solved analytically and its solution has similar interpretation to canonical correlations. When the number of components is bigger than 1, corresponding CLS problem cannot be solved exactly and greedy approximation is used. We refer the reader to the original paper [18] for more details.

5 Preliminary data experiments

5.1 Data analysis

When we first ran CLS on a corpus of medical claims we noticed two discouraging observations. First, running CLS multiple times with different random initializations produced significantly different cluster assignments with similar objective values. To quantify this observation we measured the difference between obtained clusters using three common metrics for comparing label assignments: Adjusted Rand Index [14], Adjusted Mutual Information [29] and Maximum Kappa Statistic[1] [26]. The results of comparing 10 different CLS runs are presented in Figure 1. This problem is not unique for CLS clustering and is usually handled by running algorithm multiple times and choosing the assignment with the best objective value. However, since the objective value for all the runs is very similar, whatever conclusions we might draw about this data will be unreliable and will reflect the behaviour of the clustering algorithms rather then the real properties of the data. Unless these conclusions will be consistent across many different runs, even though label assignments are very different, which is not likely to happen. We explore the potential causes of this behavior in section 6.3.

[1]The Kappa statistic also allows to find the best match of cluster labels between different runs.

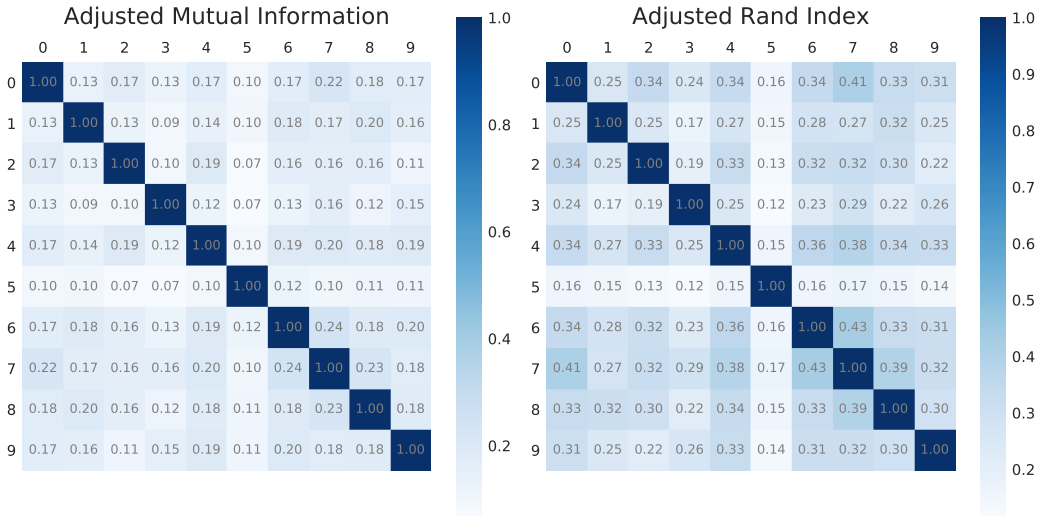


Figure 1: The results of applying CLS to the corpus of medical claims 10 times with different random initialization. We present here only adjusted Rand index and adjusted mutual information, because of the space constraints, although in general all three metrics agree that the obtained clusters are very different for all 10 runs. Examining the contingency matrix directly also confirms this observation.

Nevertheless, we tried to estimate how well CLS clusters fit this data. In order to do that we conducted the following experiment: we first ran CLS with k clusters on the whole data. Then, we split each CLS cluster into 10 folds and for each fold we trained two models: "data model" which used all the data except that fold and "clusters model" which used all the data only from that cluster except the chosen fold. Then we obtained predictions for the hold-out fold using both models, and this process was repeated for all folds in all clusters. After all clusters were evaluated we obtained 2 sets of predictions for the whole dataset: one for data model and one for clusters model for which we compute R^2 and MSE. Comparing these metrics we can quantify how much better this data can be described using k best linear predictions instead of one[2].

We observed that CLS shows a very good linear fit for this data. Even with $k = 2$, clusters model $R^2 \approx 0.86$ vs $R^2 \approx 0.27$ for the data model and MSE is decreased ≈ 3 times. Using 8 clusters, $R^2 \approx 0.96$ for the clusters model and MSE is decreased ≈ 20 times comparing to the data model. However, as pointed out in [5], this is not an indication of a good linear structure in the data. Such good values for R^2 and MSE can be obtained even if Y is replaced with random numbers, simply because CLS can group observations with similar targets together. That is, instead of finding the "real" linear patterns of the data, CLS is prone to fitting non-existing correlations that can always be found by grouping the points that happened to be on one line even if the generation process for these points was completely different. Even though this problem was already described in [5], we reinforce their findings with our observations and propose a different metric for adjustment for such noise that doesn't involve running CLS many times. The details are presented in section 6.1.

6 Synthetic experiments

6.1 Good fit on random data

From the work of Brusco et al [5] and from our preliminary experiments on the medical data we know that CLS will have good R^2 value even when there is no correlation between X and Y. To understand the intuition behind this phenomena, consider a simple case when X consists of all zeros

[2]It should be noted that this process is not the same as the usual 10 fold cross-validation since we only compute R^2 and MSE ones for all data points. It is also possible to run the usual cross-validation for each cluster separately and then consider the obtained $10k$ estimations as one set, but this is not very informative since different clusters often show very different prediction results. We also recorded the usual cross-validation metrics for each cluster and they show similar patterns to the estimation process reported in this paper.

and $Y \in \mathbb{R}$ is uniformly sampled from $[a, b]$. Since X is zero matrix, each model CLS fits will have only a bias term, equal to the mean of the selected points Y_i for cluster i (and therefore MSE will equal to the Var $[y]$ in that cluster). Thus, assuming the number of points is big enough, CLS optimization problem is equivalent to finding a separation of $[a, b]$ into k segments, such that the total variance across all of the segments is minimized:

$$\min_{s \in \mathbb{R}^{k-1}} \left[\text{Var}_{y \sim U(a, s_1)} [y] + \text{Var}_{y \sim U(s_1, s_2)} [y] + \cdots + \text{Var}_{y \sim U(s_{k-2}, s_{k-1})} [y] + \text{Var}_{y \sim U(s_{k-1}, b)} [y] \right]$$
$$\text{s.t. } a < s_1 < \cdots < s_{k-1} < b$$

This problem has an intuitive solution with s_i being equally distributed in $[a, b]$, i.e. $s_i = a + i\frac{b-a}{k}$. Defining $s_0 = a, s_k = b$ and mean of cluster i with $\bar{y}_i$ we can compute theoretical value of R^2 and MSE achieved on this data with k clusters and $n >> 1$ data points:

$$1 - R^2 = \frac{\frac{1}{n} \sum_{c=1}^{k} \sum_{y \in [s_i - s_{i-1}]} (y - \bar{y}_i)^2}{\frac{1}{n} \sum_{y \in [a,b]} (y - \bar{y})^2} \approx \frac{\frac{k}{n} \sum_{y \in [s_1 - s_0]} (y - \bar{y}_1)^2}{\frac{1}{n} \sum_{y \in [a,b]} (y - \bar{y})^2} \approx \frac{\text{Var}_{y \sim U(s_1 - s_0)} [y]}{\text{Var}_{y \sim U(a,b)} [y]} = \frac{1}{k^2}$$

since n is big and each cluster has approximately $\frac{n}{k}$ elements. Thus, with k clusters, MSE is going to be k^2 times bigger, than with 1 cluster and $R^2 = \frac{k^2 - 1}{k^2}$. So, even for this random data with no correlation between X and Y and when linear models are restricted to bias only, $R^2 = 0.75$ with just 2 clusters and $R^2 \approx 0.98$ for 8 clusters. Running a simple simulation confirms the theoretical computations with CLS being consistently able to find the optimal solution.

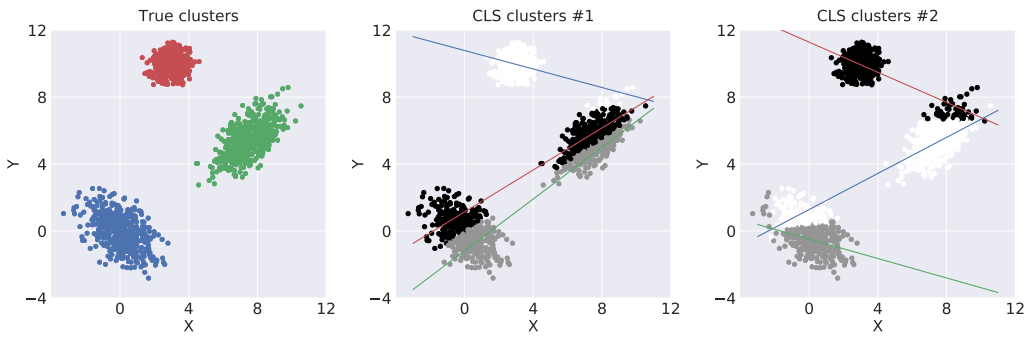
This simple experiment motivates the need to use a different metric of CLS performance that would account for such a good performance when there is no dependence between X and Y. Brusco et al [5] suggested to do a hypothesis testing, which consists of running CLS on random permutations of targets Y and comparing the obtained objective values with objective value for the true Y. Although this gives a viable metric to assess the actual degree of within-clusters linear fit, it requires running CLS multiple times. In this paper we propose another metric, which requires running CLS only 2 times and generalizes the concept of R^2 directly. We call this metric *clusterwise coefficient of determination* and denote with R^2_c. Note that, standard R^2, essentially, compares the performance of the chosen regression method to the best possible regression performance when there is no correlation between X and Y. Thus, in order to compute R^2_c we propose to divide the MSE of a CLS ran on (X, Y) (denoted with $\text{MSE}_{\text{CLS}}(X, Y)$) by the MSE of CLS ran on (0, Y) (denoted with $\text{MSE}_{\text{CLS}}(0, Y)$). The complete formula is given below:

$$R^2_c = 1 - \frac{\text{MSE}_{\text{CLS}}(X, Y)}{\text{MSE}_{\text{CLS}}(0, Y)}$$

It might be possible to compute $\text{MSE}_{\text{CLS}}(0, Y)$ efficiently without running it on the data second time, but finding such an algorithm is a direction of future research.

6.2 Converging to a non-intuitive solution

As shown in the previous section, standard metrics for CLS evaluation can indicate that there is a linear structure, when, in fact, there is no correlation between X and Y at all. However, even when the data was indeed generated from k linear models, CLS might converge to a very non-intuitive solution. To illustrate that, consider a simple case when X and Y are one-dimensional variables (Figure 2 (a)). Visual examination of the data shows that there are 3 CLS clusters (i.e. linear models generating data): one with strong positive correlation (green), second with strong negative correlation (blue) and third with almost zero correlation between X and Y (red). However, running CLS on this data never produces the expected clustering assignment. Figure 2 (b), (c) demonstrates two typical examples of CLS clusters obtained from different random initializations. These solutions are preferred by CLS because they, in fact, have lower MSE than the intuitive solution. Indeed, even if CLS is initialized to the correct label assignment, on the first iteration, it will reassign some of the points from blue cluster to be in green cluster. This happens, because green cluster's linear model will go through the center of blue cluster and some of the blue points will be better explained by that model. This problem comes from the fact that combining different datasets together will often increase the chance of finding highly correlated subsets of the combined data simply because there



(a) (b) (c)

Figure 2: Qualitative assessment of CLS performance on a simple 2D example. The true clusters are depicted on the plot (a). Plots (b) and (c) show two different solutions of CLS, both of which yield better MSE than the correct clusters. This illustrates that CLS will often find non-intuitive and non-interpretable cluster assignments since they might be optimal from the point of view of CLS objective.

are more points to choose from. This is exactly what is happening in the proposed example, since green and blue points just happen to be on the same line (there would be no problem if the green cluster was moved down by subtracting 4 from it's Y values). However, with the increase of the number of objects and the dimension of the data, it is reasonable to expect that the problem will become more severe.

It is clear from this experiments, that CLS objective is not always aligned with our intuitive expectations about the quality of the obtained clusters. Thus, the optimization objective needs to be somehow changed in order to assign lower values to the solutions with preferred structure. Fortunately, there is an easy modification to CLS algorithm that would allow to add a regularization term to an objective function that can encourage solutions with desired properties. In order to do that, the labeling step of CLS has to be changed in the following way:

$$\arg\min_i \left[\left\| y_l^T V^{(i)} - x_l^T U^{(i)} \right\|_2^2 + \lambda \phi(i, x_l, y_l) \right]$$

Where $\lambda \geq 0$ is a regularization hyperparameter. The function $\phi(i, x_l, y_l)$ can be arbitrary, as long as it only depends on the current point x_l, y_l and some properties of the cluster i (e.g. $U^{(i)}$ and $V^{(i)}$). This function should encode the desired structure of the CLS clusters, besides having a good linear correlation between X and Y which is encouraged by the first term. One way to define $\phi(i, x_l, y_l)$ is to set it to k-means objective: $\phi(i, x_l, y_l) = \beta_x \left\| x_l - \bar{x}_i \right\|_2^2 + \beta_y \left\| y_l - \bar{y}_i \right\|_2^2$, where $\bar{x}_i$ and $\bar{y}_i$ are the X and Y centers of cluster i respectively. β_x and β_y can be used to control how compact the clusters are going to be with respect to X and Y spaces separately[3].

Defining ϕ in such a way we encourage the compactness of the clusters, i.e. the far away points (measured by the L_2 distance) are unlikely to be assigned in the same cluster even if their correlation pattern is similar. In some sense this formalizes our intuition about the good cluster assignment for the synthetic example from Figure 2. Indeed, the points from the middle of the blue cluster will usually have a better correlation fit with the green cluster. However, we would still expect them to have blue labels, since they are surrounded by blue points and are visually separated from green points. Indeed, with a wide range of β_x and β_y values, CLS converges to the expected solution using the proposed k-means regularization.

[3]Note that using independent β_x and β_y makes the objective overparametrized and thus λ can be always set to 1 without the loss of generality

6.3 Convergence to significantly different solutions

Finally we explore the issue of converging to multiple significantly different solutions that we observed during preliminary experiments. There are multiple problems that could be contributing to this behaviour with various extent. One possible problem is that there are no clear linear clusters in the data and thus, depending on the random initialization, CLS will converge to different solutions, with a significant amount of points, that can be assigned to multiple clusters without a noticeable increase in CLS objective. Another explanation could be that even if there is a linear structure in the data, CLS might be getting stuck in different local minimums, since only convergence to a local minimum is guaranteed. However, it is more likely that this behaviour is caused by a more fundamental issue associated with CLS that we illustrate with the following example.

Consider a simple example where $X \in \mathbb{R}^{2n \times 2k}, Y \in R$. The data is structured in such a way so that either first k or last k features can be non-zero for a given object. That is,

$$\forall x_i : \left[\sum_{j=1}^{k} x_{ij}^2\right] \left[\sum_{j=k+1}^{2k} x_{ij}^2\right] = 0$$

The first n targets are being generated from one linear model: $y_i = x_i^T w^{(1)}, i \leq n$ and the last n from another $y_i = x_i^T w^{(2)}, i > n$. The two linear models can be arbitrary as long as they don't have bias terms (which is justified when data has zero mean). Thus, the correct solution for CLS with 2 clusters would be to assign the first n objects into one cluster and the last n objects into another cluster with recovered coefficients equal $w^{(1)}$ and $w^{(2)}$. Since there is no noise in the data, this solution yields $R^2 = 1$. However, there is another label assignment that has perfect fit. Let's denote with s_{11} all the objects from cluster 1 that have first k features equal zero and with s_{12} all the objects from cluster 1 that have last k features equal zero. s_{21} and s_{22} are defined in the same way for cluster 2. It is easy to see, that combining (s_{11}, s_{22}) in one cluster and (s_{12}, s_{21}) in another cluster, CLS would find another optimal solution with $R^2 = 1$ and coefficients

$$\hat{w}^{(1)} = [w_1^{(1)}, \ldots, w_k^{(1)}, w_{k+1}^{(2)}, \ldots, w_{2k}^{(2)}]^T, \hat{w}^{(2)} = [w_1^{(2)}, \ldots, w_k^{(2)}, w_{k+1}^{(1)}, \ldots, w_{2k}^{(1)}]^T$$

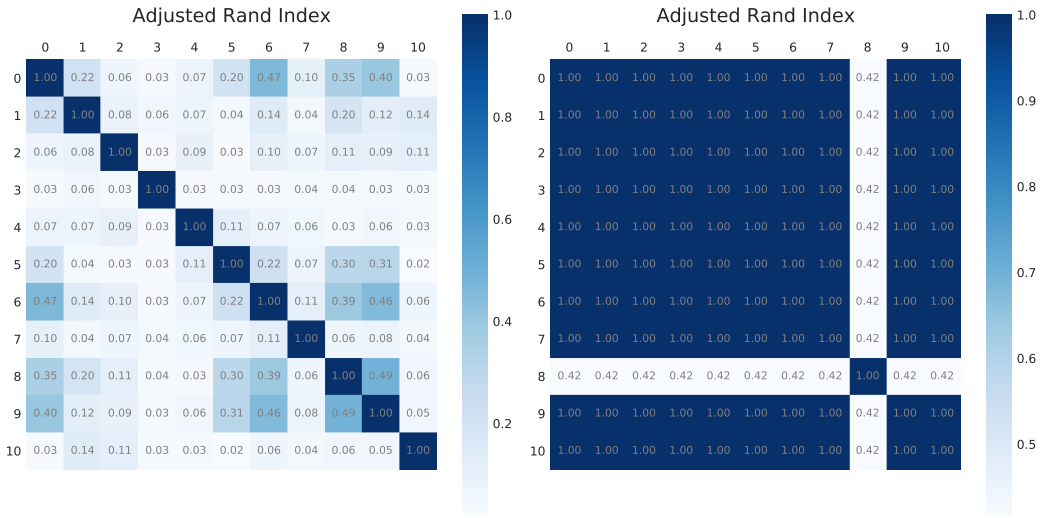
If the features are partitioned into more than 2 mutually exclusive groups or the correct number of clusters is bigger, then there are exponentially more equivalent solutions that CLS could find.

This is an example of the data for which CLS problem is ill-posed, since there are multiple optimum solutions. The main reason why this data is adversarial for CLS is because there are groups of features that are, in some sense, independent from each other. The complete independence is achieved when for all objects, having non-zero features in one group implies that all features from the other group are exactly zeros or have no correlation with the target variable (i.e. corresponding linear regression coefficients are zeros). When this property is approximately satisfied (with few non-zero features or small correlation with target) we will say that such data has *weak feature interdependence*.

Although, the complete feature independence is somewhat unrealistic for real datasets, weak feature interdependence might to be a common property of sparse data. If that is the case, CLS problem for sparse data will be significantly ill-conditioned, meaning that there would be many different solutions with almost optimal objective value. To check this hypothesis we conducted a number of synthetic experiments aimed to estimate the quality of CLS solutions in the presence of sparse features.

For all of the experiments in this section we used the following setup. First, the number of objects n, the number of sparse features m_s, the number of dense features m_d and sparsity level p are chosen. Then, the matrix $X \in \mathbb{R}^{n \times (m_d + m_s)}$ is generated with $x_{ij} \sim U(-1, 1)$ if $j \leq m_d$ and $x_{ij} \sim \text{Bernoulli}(p)$ if $j > m_d$. Then data is randomly partitioned into k clusters (unless otherwise stated, k equal 4 was used) and for each cluster, linear regression coefficients and biases $w_c \in \mathbb{R}^{m_d + m_s}, b_i \in \mathbb{R}$ are generated. Finally, $Y \in \mathbb{R}^n$ is generated with $y_i = w_c^T x_i + b_c$ if (x_i, y_i) belongs to cluster c (note that no noise is added). After that CLS was run on this data 10 times with different random initializations. We compute the adjusted Rand index (ARI) for all pairs of found label assignments. The average ARI across all pairs is measuring mutual agreement between found clusterings. We also compute ARI of each of the found assignments with true labels. Its average is measuring CLS ability of restoring true labels.

Figure 3 shows results of CLS evaluation when data consists of $n = 1000$ objects, 8 features which are either all sparse with $p = 0.25$ (a) or all dense (b). Clearly, when features are dense, CLS has



(a) Sparse data

(b) Dense data

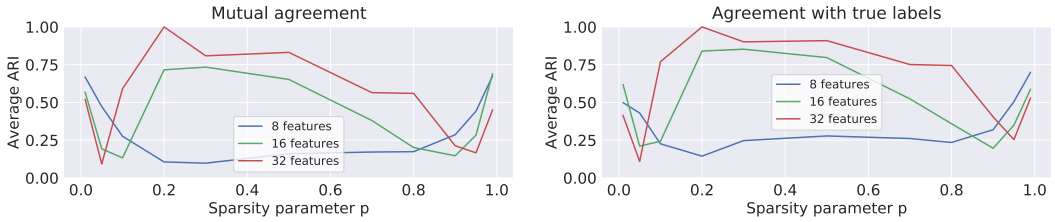
Figure 3: CLS ran 10 times on 1000 objects and (a) 8 sparse features (75% zeros), (b) 8 dense features. The first row (with index 0) shows the comparison with true labels. Clearly, optimization in the presence of sparse features is significantly more complicated. Average $R^2 = 0.95$, MSE $= 0.03$ which shows that found solutions are close to global minimum.

almost convex objective function, since 9 out of 10 runs converged to the true global minimum. However, when features are sparse, the optimization landscape becomes worse with a lot of good local minimums that CLS can converge to.

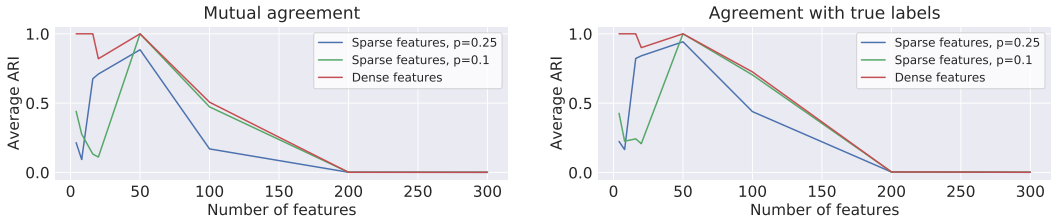
Figure 4 shows CLS dependence on (a) sparsity parameter, (b) number of features, (c) number of objects, (d) number of clusters, (e) proportion of sparse/dense features. Similar plots can be obtained when features are not $\{0, 1\}$, but $\{a, b\}$ for any a and b, which shows that the problem is not in the sparsity (i.e. presence of many zeros), but rather in the small variance in the feature values, which also leads to weak features interdependence. Figure 4 (a) shows a few interesting dependencies. First, the following trend can be seen for any number of features: for the extremely sparse data ($p = 0.01$) the accuracy of CLS solutions become reasonable good, tending to ARI $= 1.0$ as p goes to zero. Then, there is a short range of very bad performance (ARI < 0.2) after which performance becomes reasonable good with optimum around 0.5. The trend is approximately symmetric across 0.5. Although on the first glance this behaviour seems complicated, it actually aligns with our intuitive expectations. Indeed, consider the extreme case when $p = 0$. Then, CLS will find groups of similar targets Y and merge them into clusters. As we experimentally confirmed in section 6.1 for the case when $Y \sim U(a, b)$, CLS is able to consistently find the optimal solution. In these experiments, the correct solution is even easier to find, since all Ys inside clusters will be equal to the true intercept of the corresponding model[4]. With p close to 0.5, the performance of CLS is also reasonably good, since this amount of sparsity (for a fixed number of clusters and objects) introduces reasonable dependence between all subsets of features. The more features, the better performance becomes and the wider is the range of good performance[5]. However, when the amount of sparsity is small enough (but not too small to make the problem trivial), data enters the adversarial regime of weak dependence when the performance of CLS is very bad. The symmetry around 0.5 is expected since changing 0 to 1 or any other arbitrary number doesn't introduce new dependencies in the data. Figure 4 (b) shows that after certain number of features performance starts degrading both for sparse and dense models. Plots (c), (d) and (e) generally show that increasing number of objects, proportion of dense features or decreasing number of clusters makes CLS performance better. They also demonstrate significant difference in performance for the dense and sparse cases.

[4]Note, that when we introduced motivational example for weak feature interdependence, we considered the models with no bias. However, when bias is significantly different from zero, even though $X = 0$, CLS problem actually has a unique solution and becomes trivial to solve as we observe in the experiments.

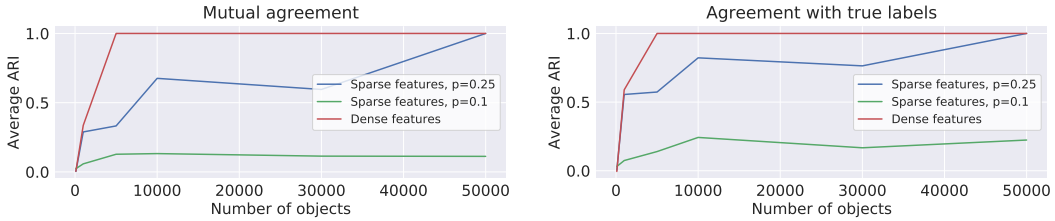
[5]Although, plot (b) shows that after certain number of features performance becomes significantly worse.



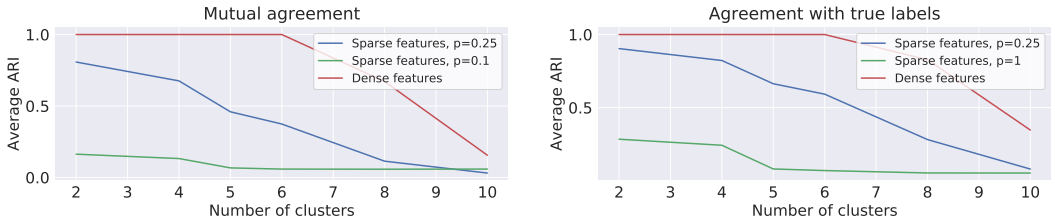
(a) 10000 objects, 4 clusters



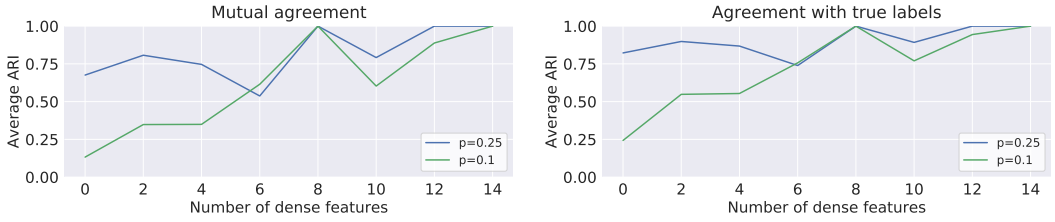
(b) 10000 objects, 4 clusters



(c) 16 features, 4 clusters



(d) 10000 objects, 16 features



(e) 10000 objects, 4 clusters, 16 features

Figure 4: These plots demonstrate how different parameters affect the quality of CLS solutions.

Overall, the following conclusions were obtained:

- CLS clustering cannot be directly applied in the presence of features with weak interdependence (which is the case when features are sparse), since optimization landscape becomes highly non-convex with many good local minimums. When data has subsets of features with zero feature interdependence, the problem becomes ill-defined, since there are multiple optimal solutions.
- Increasing proportion of dense/sparse features generally improves convergence.
- Increasing $\frac{number\ of\ objects}{number\ of\ features}$ ratio generally improves convergence.

- Increasing true number of clusters makes problem harder (performance degrades significantly more for the sparse case).
- Moving sparsity parameter closer to 0.5 generally improves convergence.

The main conclusion from these experiments is that in the presence of sparse features, even if there exists a unique global optimum, it is unlikely to be found. And in some cases there are multiple global optimums and thus, finding actual clusters and coefficient that generated the data is impossible. Therefore, if interpretation of the clusters is the goal, it is necessary to redefine problem that CLS is solving in order to have a unique solution. One way to do so is to introduce, so-called, cluster seeds: objects that are restricted to be in a certain cluster and in some sense define a prior solution to the problem. This prior is then refined when CLS recomputes clusters linear models and assigns new points. It is also possible to control how much influence seeds should bring to the cluster model by weighting seed objects (which is equivalent to implicitly adding the same seed objects to cluster multiple times). To make the contribution of seed and non-seed objects equal, seed weights could be set to $\sqrt{N_{ns}^{(i)}/N_s^{(i)}}$, where $N_{ns}^{(i)}, N_s^{(i)}$ are the number of seed and non-seed objects for cluster i. Of course, if the seed objects are independent and the number of seeds equals the number of features, CLS problem will have a unique solution, since seeds would fully define corresponding linear models. When seed objects are representative of the true clusters, one could hope that having even small number of seeds will provide enough information for CLS to converge to meaningful solutions. In case when data was not actually generated from a mixture of linear models, seeds can still be useful to define a clusters of interest. For example, if it is known that some objects behave differently from the others, placing them as cluster seeds will help to find more objects with similar characteristics.

7 Real-data experiments

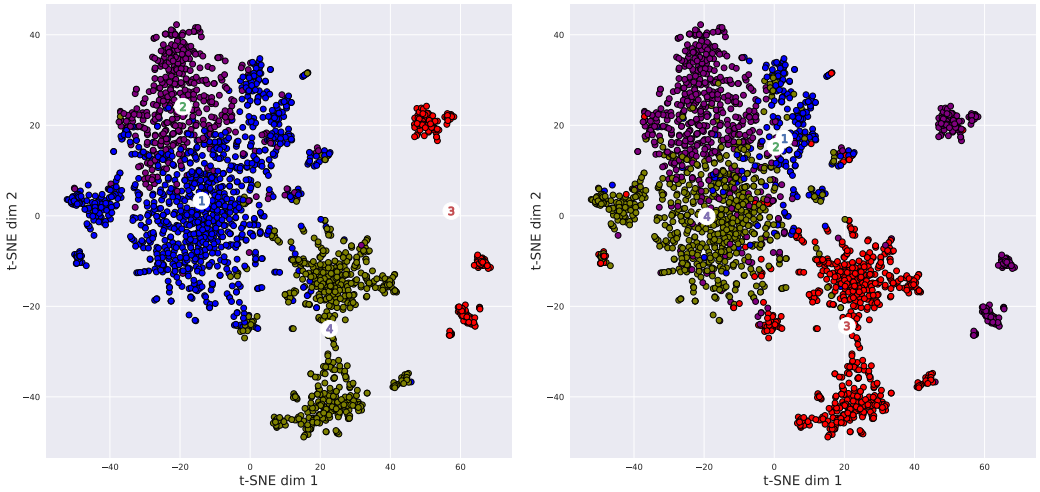
7.1 Data analysis

Since the analysis of sparse data is complicated and requires careful tuning, we restrict our attention to the dense corpus of hospital features. In this section we describe the application of CLS clustering to a hospital-level data with a focus on finding meaningful clusters that can potentially be used as peer groups. We show that CLS clustering can provide additional insights into the found peer groups and can be used to improve and refine final clusters with the correlation information. We start by applying k-means with 4 clusters to the hospital-level data which is a traditional approach of obtaining hospital peer groups. K-means was initialized using k-means++ algorithm [2] and best out of 100 iterations was chosen. We ignore the features corresponding to length of stay, since the goal is to obtain clusters based on their operational characteristics. The length of stay information will be used later to refine the analysis by applying CLS clustering. In order to visualize the results we use t-SNE technique [19] with 2 components. The found 4 clusters are depicted on Figure 5 (a).

In order to easier interpret the clustering results we use the following trick; for each cluster we train 2 models: SVM with L_1 regularization [30] and Random Forest [4] in order to classify points from one cluster vs all the rest. We tune the regularization parameter of SVM to have only few non-zero features and we also look at top-5 features according to feature importance produced by random forest. The features found using this technique can be used to understand what separates one cluster from all the rest and thus gain an intuition into why the hospitals were assigned to that cluster. The top-3 features for each cluster are:

1. 991 hospitals: “type newborn”, “MDC 6” (digestive system), “MDC 5” (circulatory system)
2. 416 hospitals: “type newborn”, “MDC 15” (newborns and neonates (perinatal period)), “MDC 14” (pregnancy, childbirth)
3. 169 hospitals: “MDC 20” (alcohol/drug use or induced mental disorders), “age”, “MDC 19” (mental diseases and disorders)
4. 586 hospitals: “age”, “MDC 23” (factors influencing health status), “type emergency’

Visually examining distributions of these features across the clusters we can further refine the understanding of obtained results. Cluster 2 seems to consists of hospitals specialized on pregnancy and childbirth. It has significantly higher average number of claims with type newborn, pregnancy and childbirth cases than all other clusters. Cluster 3 has a very high proportion of alcohol/drug illnesses as well as mental diseases comparing to other clusters. Cluster 4 has an average patient age close to 80 and seems to specialize on senior patients with a relatively small proportion of emergency



(a) k-means (b) CLS + k-means

Figure 5: Comparison of (a) k-means clustering with (b) CLS clustering combined with k-means. Different clusters are depicted with different colors. Cluster centers are marked with white circles, indicating cluster indices.

claims. Cluster 1 contains large amount of hospitals with big proportion of digestive or circulatory systems diagnosis, but a lot of hospitals from this cluster don't have evident specialization.

Next, we apply CLS clustering on the same data to gain additional insights into the found clusters. CLS uses the same set of features as k-means for independent variables X and seeks clusters that maximize correlations with length of stay per MDC features Y. Running standard CLS doesn't reveal any clear structure in the found clusters with very low CLS objective values indicating that it might have overfitted the data producing non-intuitive solutions. Thus, we use the method introduced in section 6.2 of combining CLS with k-means objective on X features only. With $\beta_x = 0.01, \beta_y = 0$ we obtain meaningful clusters that are depicted on Figure 5 (b). There are 2 interesting observations. Notice, that CLS doesn't change cluster 4 significantly, while merging clusters 2 and 3 and splitting cluster 1 into two clusters. This illustrates that k-means clusters 2 and 3 have similar correlation patters with length of stay and that cluster 1 is not correlation homogeneous. Examining CLS coefficients we can see that CLS clusters 1 and 4 (which k-means cluster 1 was split into) have opposite correlations with, for example, "MDC 1 length" (nervous system), "MDC 4 length" and (respiratory system) "MDC 9 length" (skin, subcutaneous tissue and breast). However, we want to emphasize that since CLS coefficients are harder to interpret and might not be intuitively meaningful, it is important to verify any conclusions with a human expert in the field.

7.2 Prediction

In this section we demonstrate that CLS clustering can be used to improve length of stay prediction accuracy for a claim-level data[6]. In this case Y consists of scalar values representing patient length of stay for a particular claim. It is not possible, however, to directly use linear regression models produced by CLS clustering, since it is not clear how to assign labels to test data points for which Y is unknown. One way to do it is to build a separate classification model, predicting CLS labels from features $X \in \mathbb{R}^d$. However, in many cases, d-dimensional planes produced by CLS clustering will overlap and thus it might be impossible to predict the correct labels considering only features X and ignoring the targets Y. A simple illustration of this problem is presented in Figure 6.

Manwani and Sastry [22] propose k-plane regression method that counteracts this problem by combining CLS objective with k-means on features X only[7]. At test time authors propose to compute

[6]Note that since the goal of this section is not interpretation of the results, it doesn't matter if CLS converges to the "true" clusters generating the data, as long as it is helpful for prediction. Thus, the sparsity problem described in section 6.3 does not matter as much in this case.

[7]We described this approach in section 6.2, however, aiming at different goal.

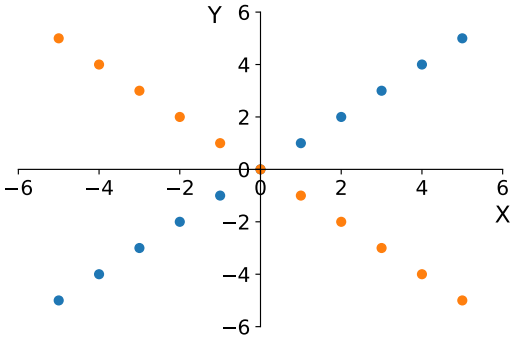


Figure 6: This plot illustrates that it might be impossible to predict CLS labels from features X only. In this case there are 2 linear regressions that CLS would find, depicted with orange and blue. However, ignoring Y will make points from different clusters identical.

k-means centers of each cluster and assign new objects to clusters with closest centers. However, since the objective function has 2 potentially contradicting terms, it is not guaranteed that this criteria will match points to the correct clusters even for the training data. Thus, even after combining CLS and k-means objectives, it might be beneficial to train additional classifier to predict CLS labels at test time. We call this method *predictive CLS* (CLS$_p$) and compare its performance to the k-plane regression method of Manwani and Sastry [22]. Table 1 contains the results of the comparison. To better understand the performance of both methods we used the following setup. First, the data is normalized: each feature is divided by standard deviation with no mean subtraction[8]. Then, CLS clustering with varying number of clusters k and regularization coefficient β_x was applied to the whole dataset and cluster labels were recorder. We report the cross-validation[9] performance of CLS models on the whole dataset in the second column of Table 1. These values show the performance of prediction with CLS models if the true labels were known at test time[10]. Then, the dataset was split into train (75%) and test (25%) subsets. We predict test CLS labels using Random Forest for CLS$_p$ and closest cluster for k-plane. The accuracy of this prediction is reported in columns 3 and 4. After that we compute predictions for the length of stay values on a test subset using linear regression models built with CLS for corresponding predicted clusters. These results are reported in columns 5, 6. Notice that the performance of both CLS$_p$ and k-plane crucially depends on the labels prediction accuracy. With $k = 2, \beta_x = 0$, even though only 13% of the points are classified incorrectly, the R^2 drops from potential 0.77 (when the true labels are known) to 0.21 for CLS$_p$ or -0.15 for k-plane which is significantly worse than running a simple linear regression. The reason for that is because different CLS clusters have radically different regression coefficients and thus for any incorrectly assigned label, the prediction might be arbitrary bad. Consider the example on Figure 6: if blue model is used for prediction of the orange dots, prediction becomes worse and worse farther from the origin. In that case the best possible prediction model at test time would be a weighted combination of the blue and orange models with equal weights. Even though this model will always predict zero, it is the best possible prediction when there is no prior knowledge about cluster labels for test objects.

We utilize this idea to improve the prediction results of CLS$_p$. Instead of taking only the answer of the model predicted by random forest, we take a weighted average over models for all clusters with weights being equal to probabilities that a point belongs to certain cluster, which random forest can compute. This approach is similar in spirit to performing a fuzzy CLR clustering (e.g. [9]) and could potentially be improved by incorporating fuzzy class memberships into the usual CLS procedure. We report the results of this weighting technique in columns 7 and 8 of Table 1. For k-plane weights are equal to the normalized distances to corresponding clusters.

[8]The reason for that is because the claim-level data is extremely sparse and thus it is possible to utilize fast sparse algorithms (e.g. LSQR [24] instead of the standard linear regression is used in all the models presented below) for models training. Subtracting mean, however, would change the sparsity of the data and models will become much slower to train.

[9]For the description of this cross-validation process, see section 5.

[10]Since we obtain cross-validation estimates of R^2, the only overfitting present in this evaluation comes from CLS itself: because we assume that we know the true labels.

Table 1: Comparison of k-plane regression and predictive CLS. The baseline R^2 of a simple linear regression is 0.3.

k, β_x	R^2 train	CLS_p label accuracy	k-plane label accuracy	CLS_p R^2	k-plane R^2	weighted CLS_p R^2	weighted k-plane R^2
2, 0	0.77	0.87	0.80	0.21	-0.15	**0.40**	-0.22
2, 0.1	0.76	0.91	0.81	0.20	-0.56	0.39	-0.82
2, 0.5	0.54	0.95	0.93	0.24	0.05	0.34	-0.44
2, 0.7	0.48	0.97	0.95	0.30	0.12	0.34	-0.18
2, 1.0	0.55	0.97	0.97	0.30	0.28	0.36	-2.04
2, 10.0	0.33	0.99	1.00	0.33	0.33	0.33	0.24
8, 0.0	0.96	0.38	0.26	0.09	-0.17	0.38	-0.24
8, 0.1	0.88	0.86	0.86	0.19	0.13	**0.39**	-1.50
8, 0.5	0.60	0.97	0.97	0.30	0.28	0.36	-1.63
8, 0.7	0.55	0.98	0.98	0.32	0.31	0.36	-0.61
8, 1.0	0.53	0.98	0.98	0.32	0.30	0.35	-0.22
8, 10.0	0.34	0.99	1.00	0.33	0.33	0.34	0.24

Note that as expected, CLS_p algorithm gives better label prediction accuracy than k-plane. Also, increasing k-means coefficient β_x leads to better label predictions (columns 3, 4) for both algorithms, but at the same time to lower potential R^2 (column 2). Thus, k-plane and usual CLS_p don't perform well when β_x is small and are giving the best R^2 for $\beta_x = 10$ which is almost the vanilla k-means clustering. However, weighting technique significantly improves the accuracy of CLS_p since it doesn't yield catastrophically bad predictions for the incorrectly assigned objects anymore. This is not true with weighted k-plane, since distances to cluster centers are usually very similar and thus don't provide a good indication of the models confidence.

However, the final R^2 values for the length of stay prediction might still be biased since CLS clustering was applied to the whole dataset (otherwise it would be impossible to compare label prediction accuracy for the two methods). We obtain unbiased estimates by running CLS only on the train data and evaluating the best CLS_p models (weighted version, for $k = 2$ clusters $\beta_x = 0$, for $k = 8$ clusters $\beta_x = 0.1$) as well as the best k-plane models (usual version, for $k = 2$ clusters $\beta_x = 10$, for $k = 8$ clusters $\beta_x = 10$). This unbiased estimation yields similar results to the ones reported in the Table 1, indicating that CLS doesn't overfit the data. The new results are presented in Table 2.

There is another way to do a prediction with CLS clustering. Instead of combining it with k-means, it is possible to incorporate some natural constraints available in the data. One way to do so is to constraint claims based on their hospital membership. This means that we enforce all claims from the same hospital to be in the same cluster. This can be easily incorporated into CLS by changing the labeling step to compute per hospital argmin instead of per claim. By using this kind of constraints there is no problem in assigning labels at the test time since hospital membership is assumed to be known. We denote this method with CLS_c and report the results in Table 2. The constrained CLS shows the best R^2 beating even such a highly non-linear model as random forest with just 2 clusters. With 8 clusters the performance is improved further. The results might become better if additional tuning is performed, e.g. exploring different types of constraints and different numbers of clusters.

8 Conclusions

In this paper we have demonstrated that applying CLS clustering to sparse datasets might present significant problems for the interpretation of the results. With theoretical examples and synthetic experiments we have shown that this happens because in some cases CLS problem becomes ill-

Table 2: Comparison of different prediction models

	MSE	R^2
Linear Regression	0.71	0.30
Random Forest	0.59	0.42
K-plane (2 clusters)	0.70	0.31
K-plane (8 clusters)	0.67	0.33
CLS_p (2 clusters)	0.61	0.40
CLS_p (8 clusters)	0.62	0.38
CLS_c (2 clusters)	0.57	0.43
CLS_c (8 clusters)	**0.55**	**0.45**

defined, meaning that there might be exponentially many potential solutions and thus, different interpretations. We also provide additional insights and interpretations of the problems of finding non-intuitive solutions and overestimating the goodness of CLS fit that have been observed in the literature before. Based on this preliminary analysis we show how CLS clustering can be potentially applied to finding hospital peer-groups, which is an important problem for many health insurance providers. Finally, we propose new techniques to apply CLS for prediction and experimentally show that on the problem of predicting patient length of stay they perform better than k-plane regression, linear regression and random forest.

References

[1] J. A. Alexander, C. J. Evashwick, and T. Rundall. Hospitals and the provision of care to the aged: a cluster analysis. *Inquiry*, pages 303–314, 1984.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[3] A. M. Bagirov, A. Mahmood, and A. Barton. Prediction of monthly rainfall in victoria, australia: Clusterwise linear regression approach. *Atmospheric Research*, 188:20–29, 2017.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] M. J. Brusco, J. D. Cradit, D. Steinley, and G. L. Fox. Cautionary remarks on the use of clusterwise regression. *Multivariate Behavioral Research*, 43(1):29–49, 2008.

[6] M. J. Brusco, J. D. Cradit, and A. Tashchian. Multicriterion clusterwise regression for joint segmentation settings: An application to customer value. *Journal of Marketing Research*, 40(2):225–234, 2003.

[7] M. M. Byrne, C. N. Daw, H. A. Nelson, T. H. Urech, K. Pietz, and L. A. Petersen. Method to develop health care peer groups for quality and financial comparisons across hospitals. *Health services research*, 44(2p1):577–592, 2009.

[8] R. A. da Silva and F. d. A. de Carvalho. 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, 2017.

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

[10] X. Z. Fern, C. E. Brodley, and M. A. Friedl. Correlation clustering for learning mixtures of canonical correlation models. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 439–448. SIAM, 2005.

[11] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[12] C. Hennig. *Identifiability of Finite Linear Regression Mixtures*. Universität Hamburg. Institut für Mathematische Stochastik, 1996.

[13] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[14] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[15] C. Kang and S. Ghosal. Clusterwise regression using dirichlet mixtures. *Advances in multivariate statistical methods*, 4:305, 2009.

[16] H.-C. Kang, J.-S. Hong, and H.-J. Park. Development of peer-group-classification criteria for the comparison of cost efficiency among general hospitals under the korean nhi program. *Health services research*, 47(4):1719–1738, 2012.

[17] T. D. Klastorin. An alternative method for hospital partition determination using hierarchical cluster analysis. *Operations Research*, 30(6):1134–1147, 1982.

[18] E. Lei, K. Miller, and A. Dubrawski. Learning mixtures of multi-output regression models by correlation clustering for multi-view data. *arXiv preprint arXiv:1709.05602*, 2017.

[19] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[20] L. MacNabb. Application of cluster analysis towards the development of health region peer groups. In *Proceedings of the Survey Methods Section*, pages 85–90. Citeseer, 2003.

[21] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

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

[23] F. Murtagh and P. Contreras. Methods of hierarchical clustering. *arXiv preprint arXiv:1105.0121*, 2011.

[24] C. C. Paige and M. A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM transactions on mathematical software*, 8(1):43–71, 1982.

[25] A. Plaia. Constrained clusterwise linear regression. *New Developments in Classification and Data Analysis*, pages 79–86, 2005.

[26] C. Reilly, C. Wang, and M. Rutherford. A rapid method for the comparison of cluster analyses. *Statistica Sinica*, pages 19–33, 2005.

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

[28] D. Vicari and M. Vichi. Multivariate linear regression for heterogeneous data. *Journal of Applied Statistics*, 40(6):1209–1230, 2013.

[29] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[30] J. Zhu, S. Rosset, R. Tibshirani, and T. J. Hastie. 1-norm support vector machines. In *Advances in neural information processing systems*, pages 49–56, 2004.

[31] M. Zodet and J. Clark. Creation of hospital peer groups. *Clinical performance and quality health care*, 4(1):51–57, 1996.