MIXED-INTEGER EXPONENTIAL CONE PROGRAMMING IN ACTION: SPARSE LOGISTIC REGRESSION AND OPTIMAL HISTOGRAM CONSTRUCTION

by
SAHAND ASGHARIEH AHARI

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
August 2020

MIXED-INTEGER EXPONENTIAL CONE PROGRAMMING IN ACTION: SPARSE LOGISTIC REGRESSION AND OPTIMAL HISTOGRAM CONSTRUCTION

Approved by:

Asst. Prof. Dr. Burak Kocuk .....
(Thesis Supervisor)

Asst. Prof. Dr. Ezgi Karabulut Türkseven .....

Prof. Dr. İ.Kuban Altınel .....

Date of Approval: 18/08/2020.

ABSTRACT

MIXED-INTEGER EXPONENTIAL CONE PROGRAMMING IN ACTION: SPARSE LOGISTIC REGRESSION AND OPTIMAL HISTOGRAM CONSTRUCTION

SAHAND ASGHARIEH AHARI

Industrial Engineering M.Sc. THESIS, August 2020

Thesis Supervisor: Asst. Prof. Dr. Burak Kocuk

Keywords: mixed-integer conic programming, machine learning, sparse logistic regression, Kullback-Leibler divergence

In this study, two problems namely as, *Feature Subset Selection In Logistic Regression* and *Optimal Histogram Construction* are formulated and solved using solver *MOSEK*. The common characteristic of both problems is that the objective functions are *Exponential Cone-representable*. In the first problem, a prediction model is derived to predict the dichotomous dependent variable using labeled datasets which is known as *classification* in the context of machine learning. Different versions of the model are derived by the means of regularization and goodness of fit measures including Akaike Information Criteria, Bayesian Information Criteria, and Adjusted McFadden. Furthermore, the performance of these different versions are evaluated over a set of toy examples and benchmark datasets. The second model is developed to find the optimal bin width of histograms with the aim of minimizing *Kullback–Leibler divergence*, which is called *Information gain* in machine learning. The success of the proposed model is demonstrated over randomly generated instances from different probability distributions including Normal, Gamma and Poission.

ÖZET

KARMA TAMSAYILI ÜSTEL KONİK PROGRAMLAMA UYGULAMALARI: SEYREK LOJİSTİK REGRESYON VE ENİYİ HİSTOGRAM İNŞAASI

SAHAND ASGHARIEH AHARI

ENDÜSTRİ MÜHENDİSLİĞİ YÜKSEK LİSANS TEZİ, Ağustos 2020

Tez Danışmanı: Dr. Öğr. Üyesi Burak Kocuk

Anahtar Kelimeler: karma tamsayılı üstel konik programlama, makine öğrenmesi, seyrek lojistik regresyon, Kulllback Leibler uzaklığı

Bu çalışmada Öznitelik Altküme Seçmeli Seyrek Lojistik Regresyon ve Eniyi Histogram İnşaası Problemleri'nin gösterimleri verilmiştir. Ortak özelliği amaç fonksiyonlarının üstel konik programlama gösterimli olan bu iki problem, MOSEK çözücüsü ile çözülmüştür. İlk problemde, makine öğrenmesinde sınıflandırma olarak bilinen etiketli veri kümeleri üzerinde ikili bağımlı değişkeni tahmin etmek için bir model kurulmuştur. Bu modelin Akaike, Bayesçi ve Düzeltilmiş McFadden Bilgi Kıstasları gibi uyum iyiliklerini göz önünde bulunduran sürümleri çözülmüştür. Bu modellerin başarımı, rassal olarak üretilen ve literatürden alınan veri kümeleri üzerinde ölçülmüştür. İkinci model, histogramlarda Kullback-Leibler uzaklığını enküçükleyecek şekilde eniyi bölme genişliğini bulmak için geliştirilmiştir. Bu modelin başarımı Normal, Gamma ve Poisson olasılık dağılımlardan üretilen rassal veri kümeleri üzerinde ölçülmüştür.

ACKNOWLEDGEMENTS

This thesis is dedicated to my beloved family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATONS

AIC Akaike Information Criteria ..... viii, 8, 12, 13, 24, 25, 26, 46, 62

BIC Bayesian Information Criteria ..... viii, 8, 12, 24, 25, 46, 47, 48, 51, 62

DTC Decision Tree Classification 5

MCMC Monte Carlo Markov Chain 10

MICP Mixed-Integer Cone Programming 1, 2, 4, 13, 14, 16, 19, 28

MIEXP Mixed-Integer Exponential Cone Programming. xii, 4, 13, 23, 28, 29, 30, 32, 33, 39, 45, 46, 47, 48, 49, 50, 52, 53, 54, 56, 57, 58, 59, 60, 61, 62, 63

MISDP Mixed-Integer Semidefinite Programming 3, 4

MISOCP Mixed-Integer Second Order Cone Programming 2, 3

ML Maximum Likelihood 7

SVM Support Vector Machines 5, 6

1. Introduction

The aim of this thesis is to derive a mixed- integer exponential cone model for two different problems that lie in the intersection of optimization and machine learning. The first problem is *sparse logistic regression* and the second is *optimal histogram construction*. In this section, we review the related literature and the basic definitions.

1.1 Mixed-Integer Conic Optimization

In this section, we will represent the general form of conic programming and will discuss mixed-integer conic programming (MICP) models for different types of cones. We will also provide a brief review for the existing solution methods of MICPs and at the same time their applications in different areas.

Definition 1.1. *A set $K \subseteq \mathbb{R}^n$ is called a cone if $\lambda x \in K \quad \forall x \in K$, $\lambda \geq 0$.*

Definition 1.2. *A cone K is a regular cone if it is closed, convex, pointed, and full-dimensional.*

To give a definition for conic programs, in addition to the definition of a cone provided in Definition 1.1, we need to define a conic inequality.

Definition 1.3. *Let K be a regular cone. Then a conic inequality for the pairs of vectors u and v is defined as:*

$$u \geq_K v \iff u - v \in K.$$

By using Definition 1.3, we can define a conic programming problrm as in (1.1):

$$\min_x \left\{ c^T x | Ax \geq_K b \right\}, \tag{1.1}$$

in which we optimize a linear function over a conic inequality defined with respect to a cone K.

Conic programming is a generalization of linear programming since we can choose K as the cone of nonnegative orthant [1]. In the following parts, we will provide a review for the conic programming considering two widely used nonlinear cones.

Second-order cone programming (SOCP) and semidefinite programming (SDP) are among the most common conic programming classes. In this thesis, we are more interested in mixed-integer conic programming (MICP) which deals with conic optimization problems involving integer variables. There are many research studies in which different problems have been modeled using MICP formulations [2]. In this part, we will discuss solution methods and the applications of MICPs.

General formulation of mixed-integer second order cone programs (MISOCPs) is expressed as (1.2) [2]:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} \ c^T x \\ \text{s.t. } \|A_i x + b_i\|_2 \leq a_i^T x + d_i, \quad i = 1, \ldots, m, \end{cases} \tag{1.2}$$

where x is the vector of decision variables with size n and the set $\mathcal{X}$ is defined as

$$\mathcal{X} = \{(v, w) : v \in \mathbb{Z}^p, w \in \mathbb{R}^q\},$$

with

$$p + q = n, c \in \mathbb{R}^n, A_i \in \mathbb{R}^{m_i \times n}, b_i \in \mathbb{R}_i^m, a_i \in \mathbb{R}^n, d_i \in \mathbb{R}, \quad i = 1, \ldots, n.$$

First, we discuss the case where p is equal to zero, hence, (1.2) reduces to a convex optimization problem known as SOCP [3]. A group of convex optimization problems including Linear Programs (LPs), Quadratic Programs (QPs), and Quadratically Constrained Quadratic Programs (QCQPs) can be formulated as SOCPs [4]. The primal-dual interior point method can be extended to solve SOCPs [5, 6]. Moreover, as SOCP is a special case of Nonlinear Programming (NLP), the interior point method which has been proposed for NLP could possibly be extended for SOCPs.

The applications of SOCPs can be found in different fields including combinatorial optimization, robust optimization, finance, and engineering [3].

Second, we discuss the case $p \neq 0$ in which (1.2) represents an MISOCP. Generally, we can divide solution methods of MISOCPs into two groups. First group includes extension of mixed-integer linear programming (MILP) approaches while the second includes special-purpose mixed-integer nonlinear programming (MINLP) approaches. We note that interior point methods for SOCPs have some deficiencies when applied to MISOCPs, although they are computationally efficient and theoretically robust algorithms for SOCPs. One of these drawbacks is being difficult to warm start the algorithm due to the requirement of starting from a strictly feasible primal-dual pair of solutions [2]. Solution algorithms to solve MISOCPs are extensively discussed in the literature as well. Gomory cuts and tight relaxations [7], rounding cuts [8], and MILP methods applied to lifted polyhedral relaxation [9] are among approaches for finding the solution of MISOCPs. From application perspective, MISOCPs are a widely used class of problems in different fields [2] such as portfolio optimization, network design [10], delays in telecommunication networks [11], battery swapping stations on a freeway network [12], and power distribution systems [13].

Semidefinite programming is another important subclass in conic programming in which the aim is to minimize a linear function subject to a linear matrix inequality (LMI) [14]:

$$\begin{cases} \min\ c^T x \\ \text{s.t. } F_0 + x_1 F_1 + \ldots, + x_m F_m \succeq 0, \end{cases} \tag{1.3}$$

where $F_i \in \mathbb{S}^{n \times n}$ and $x \in \mathbb{R}^m$.

In fact, SDP can be interpreted as an extension of linear programming where matrix inequalities appear instead of element-wise inequalities of vectors. In other words, we consider the inequalities with respect to the cone of positive semi-definite matrices rather than the cone of non-negative orthant. The applications of SDP can be found in combinatorial optimization [15], control theory [16], and signal processing and communications [17]. Semidefinite programming is a powerful tool for tackling with non-convex quadratically constrained quadratic programs (QCQP) as well [17].

By the existence of some integer variables, SDPs turn to mixed-integer semidefinite programming (MISDPs). From application perspective, MISDPs are mainly divided

into two groups where the first group is reformulation of combinatorial optimization problems and the second is expressing structure of problems. Ellipsoidal uncertainty sets in robust optimization is a proper illustration of the latter group to show the importance of SDPs [18]. The main method for solving MISDPs is branch-and-bound, however, based on the structure of some specific problems such as graph partitioning [19] and max-cut problem [20], exceptional approaches have been proposed [21].

One of the other important cones is called exponential cone which is used in both models of this thesis namely as, *feature subset selection in logistic regression* and *optimal histogram construction*. As mixed-integer exponential cone programming (MIEXP) is the main idea of this research, we will cover the details in methodology section.

In this section the main ideas related to MICPs were summarized. In the next section, we will continue by a brief introduction for machine learning and a detailed discussion on an specific application of it due to its relevance with application problem of this thesis.

1.2 Machine Learning

Benefiting from the theory of statistics, machine learning is the process of learning from past experiences to construct a model which can be either descriptive, predictive, or both. Some applications of machine learning are regression, unsupervised learning, reinforcement learning, and learning associations [22]. In addition to the mentioned applications, there is another important application known as *classification*, which is also one of the main pillars of this thesis. Hence, the following section will be dedicated to this application.

1.2.1 Classification and Data Prediction Models

Classification in data mining is defined as associating each data point to one of the available predefined classes [23]. In other words, the problems in which the dependent variable is qualitative, meaning that it takes on values in one of finite different classes, are often categorized as *classification problems* [24]. In this section,

we will provide a review on the most common classification methods including the logistic regression which is the related classification method for this thesis.

I. **Naive Bayes**

In Naive Bayes classifications, the learning procedures become simple by the assumption that features are independent for a given class of dependent variable. Even under this simplistic assumption, the Naive Bayes classifier is confirmed to have a reasonable performance for some certain applications comparing to the other classification methods [25]. By focusing on the relations between data characteristics and the performance of Naive Bayes classifier, it is revealed that in the cases of functionally dependent or totally independent features, the Naive Bayes classifier demonstrates its best performance. From application perspective, the performance of Naive Bayes is significantly promising for medical diagnosis and text classification [26].

II. **Decision Trees**

Decision tree classification (DTC) approach is a subgroup of multi-stage decision making in which a compound decision is divided into several smaller and simpler decisions. DTC approach is mainly divided into three steps including i) choosing a proper structure for the tree, ii) selecting feature subsets at each level, and iii) defining a decision rule. The advantage of DTC is the elimination of high-dimensionality and estimation of a complicated global decision region using some manageable local decision regions. On the contrary, the drawbacks are the increase of search time in the presence of large number of classes and accumulation of errors for different levels of tree. [27].

III. **Instance-Based Learning**

The main idea of instance-based learning (IBL) is adopted from nearest neighborhood classification method which is extensively discussed in [24]. An IBL algorithm is described by a similarity function, concept description updater and classification function, and the goal is that the similar instances should have the same class label. The advantages of IBL methods are their simplicity and low updating cost. On the other hand, inflexibility of these methods to irrelevant features and being computationally expensive classifiers can be recognized as their disadvantages [28].

IV. **Support Vector Machines**

Support vector machines (SVM) is another widely used method for classification. The aim of this method is to find a separating hyperplane or hypersurface with maximal margin (and minimum classification error for non-linear SVM) that separates

the training data generally into two clusters and then labels the instances of test data considering the generated hyperplane (or hypersurface). The SVM model with a separating hyperplane is represented as in (1.4) [29]:

$$
\begin{cases}
\min\limits_{w,b,\epsilon} \sum\limits_{i=1}^{n} \epsilon_i + C\|w\|_2^2 & \\
\text{s.t. } y_i(w^T x_i + b) \geq 1 - \epsilon_i - v_i, & i = 1, \ldots, n \\
\quad \epsilon_i \geq 0 & i = 1, \ldots, n,
\end{cases} \tag{1.4}
$$

where n shows the number of data points and C is the penalty term for each misclassified data point predefined by the user. We note that there exist some methods such as grid search for parameter tuning [30].

Different extensions of SVM have been proposed in the literature. One of the important extensions deal with the imbalanced training data, where the cardinality of the existing classes are different from each other. To deal with this issue, weighted support vector machine (WSVM) model is proposed in [31]. In this method, the penalty term is taken proportional to the cardinality of each class in training dataset. Therefore, the penalty term will be greater for the misclassification of minority class. The formulation in the case of binary output data is given by (1.5).

$$
\begin{cases}
\min\limits_{w,b,\epsilon,v} \dfrac{1}{2}\|w\|_2^2 + \dfrac{C}{2|I^+|} \sum\limits_{i \in I^+} \epsilon_i^2 + \dfrac{C}{2|I^-|} \sum\limits_{i \in I^-} \epsilon_i^2 & \\
\quad \text{s.t. } y_i(w^T x_i + b) \geq 1 - \epsilon_i - v_i, & i \in I.
\end{cases} \tag{1.5}
$$

In [32], inspired by the ℓ_1-SVM (see e.g. [33]), a SVM model is developed for feature subset selection considering a budget constraint:

$$
\begin{cases}
\min\limits_{w,b,\epsilon,v} \sum\limits_{i=1}^{n} \epsilon_i & \\
\quad \text{s.t. } y_i(w^T x_i + b) \geq 1 - \epsilon_i, & i = 1, \ldots, n \\
\qquad l_j v_j \leq w_j \leq u_j v_j & j = 1, \ldots, p \\
\qquad \sum\limits_{j=1}^{p} c_j v_j \leq B & \\
\qquad v_j \in \{0, 1\} & j = 1, \ldots, n \\
\qquad \epsilon_i \geq 0 & i = 1, \ldots, n.
\end{cases} \tag{1.6}
$$

If $v_j = 1$ then the j^{th} feature is selected and the value of w_j can fluctuate between the lower bound l_j and upper bound u_j. The term c_j is the cost of choosing the feature j. The main disadvantage of using this model is that for a specific value of B, there exists multiple optimal solutions with a zero optimal value, especially in the cases where $n \ll p$. The extension of this work is presented in [34] to avoid the aforementioned criticism.

V. **Neural Networks**

Neural networks have been used in different areas including signal processing, expert systems, modeling and forecasting [35]. Generally, neural networks consist of input layer and output layer while there could be some intermediate layers called as hidden layers [36]. Each of the hidden layers contains a set of nodes which are identified by an activation function, threshold value, and a vector of weights. In the absence of hidden layers and adopting a sigmoidal activation function, the model will be equivalent to the logistic regression model [37].

VI. **Logistic Regression**

Logistic regression is a type of regression where the dependent variable takes only a finite set of values [38]. When the cardinality of this finite set of values is two, it is called binary logistic regression (these two values are usually zero and one). However, the independent variables are not restricted to a finite set of values and also can take continuous, categorical, or binary values.

Binary logistic regression is formally defined as follows: Given a dataset $\{x_i, y_i\}^n$, where $x_i \in \mathbb{R}^p$ (p is the total number of features) and $y_i \in \{0, 1\}$, one is interested in the conditional probability revealed in (1.7):

$$P(y_i = 1|x_i) = \frac{1}{1 + e^{\beta^T x_i}}. \tag{1.7}$$

The aim here is to estimate the parameter β. The most common methods for estimating this parameter are maximum likelihood (ML) and iteratively re-weighted least squares (IRLS). In the ML method, the estimator of β, denoted by $\hat{\beta}$, is obtained by maximizing the likelihood function (1.8):

$$\prod_{i=1}^{n} p(y_i|x_i). \tag{1.8}$$

There is no closed-form solution for $\hat{\beta}$, so in high dimensional datasets, especially when the number of features is significantly greater than that of instances, the estimation of $\hat{\beta}$ using these methods may not lead to an appropriate estimation.

However, some of these features are redundant which means that they are not influential to the independent variable value [39]. Therefore, we can reduce the number of features to have a better estimation of $\hat{\beta}$ by eliminating redundant features and using truly important ones when constructing a model. The action of finding appropriate features is called feature subset selection. We will introduce some prevailing feature subset selection methods in section 1.2.2.

1.2.2 Feature Selection

Feature subset selection is the process of choosing a subset of features to construct a model (see Definition 2.2 for the formal definition of feature selection). Major reasons of using feature subset selection fall into three main categories: (i) improving prediction accuracy, (ii) making the model easier to interpret, and (iii) decreasing the time of prediction [40]. In general, there are two important factors to be considered in variable selection. Firstly, developing a criterion to draw a comparison among the subsets and secondly, not being computationally expensive. A naive and inefficient way to find the best subset is to consider all possible combinations of features, which leads to 2^k subsets where, k is the number of independent variables. As the number of subsets increases exponentially with k, for large numbers of independent variables, this procedure is computationally intractable. In case of knowing the number of variables to be selected in advance, the problem could be solved by minimizing the sum of squared deviation. However, fixing the number of variables to be selected in advance is disadvantageous, hence, in [41] some Goodness of Fit Measures (GOf) including Adjusted R^2, Akaike Information Criteria (AIC), and Bayesian Information Criteria (BIC) are adopted to overcome this problem.

Feature selection methods fall into three main categories including Wrapper, Filter, and Embedded approaches [42]. In linear regression models, shrinkage methods are adopted to apply feature selection. In shrinkage method, unlike the models that fit a linear model containing a subset of predictors, a model is constructed by including all predictors and shrinking some of the coefficient estimates to zero by regularization techniques. Hence, shrinkage methods by estimating the values for some of the predictors' coefficient as zero, excludes those predictors from the model and develops a sparse model as well. [24]. Some important shrinkage techniques are as follows:

I. **Smoothly Clipped Absolute Deviation Penalty (SCAD)**

In [43], a variable selection model with a nonconcave penalized likelihood is introduced to overcome the problems of using stepwise selection methods including the computational costs and the ignorance of stochastic errors. The stepwise methods are illustrated in [25]. SCAD method selects the variables and simultaneously estimates the value of the coefficient of each selected variable. The penalty function used in this method (i) contributes to a sparse solution, (ii) assures the stability of model selection, and (iii) provides an unbiased estimation of the coefficients.

II. **Least Absolute Shrinkage and Selection Operator (LASSO)**

In regression models, one can estimate the coefficients by ordinary least squares (OLS), however, the drawbacks of this method are the low prediction accuracy and interpretability [44]. Lasso technique shrinks a subset of the coefficients estimation exactly to zero by adding constraints or regularization on the variables, hence, develops a sparse model to improve the prediction accuracy and provides an interpretable model at the same time [24].

III. **Minimax Concave penalty (MCP)**

Although the LASSO technique is fast, it is biased, hence, may lead to inconsistency in feature selection. To handle this, a nearly unbiased model known as $MC+$ is proposed in [45] which consists of two components. First component is a minimax concave penalty technique and the second is Penalized Linear Unbiased Selection (PLUS) algorithm. The mentioned study proves the promising performance of this algorithm in terms of accuracy and computational efficiency.

By means of penalized iteratively reweighted least squares (IRLS), the mentioned algorithms are also applicable for logistic regression as well. However, shrinkage methods require a proper selection of tuning parameters, which is a challenging task in the case of high-dimensionality [39]. Hence, we will now discuss another group of methods offered for feature selection known as group of Bayesian methodologies, which are free of parameter tuning. In Bayesian methodology, indicator models are widely used in high-dimensional situations to determine the truly important variables. In Indicator models, a binary variable is assigned to each feature which takes value one if the feature is important and zero otherwise. Bayesian variable selection methods consist of i) a prior to produce a posterior distribution, and ii) an approach for information extraction from posterior. Now, we review some of these methods.

I. **Bayesian Lasso**

There are a variety of models such as LASSO for variable selection and coefficient

estimation simultaneously with focus on consistency property. In [46], a Bayesian based formulation is developed which covers most of the versions of LASSO and simultaneously is superior to these versions in producing valid standard errors. This model which is called as Bayesian LASSO is built up on a geometrically ergodic Markov chain to extract information from posterior and revealed to have either similar or better performance in comparison with the other versions of LASSO.

II. **Iterated Conditional Modes/Medians (ICMM) method**

In [47], an algorithm is developed to construct empirical Bayesian variable selection. The motivation of this method is to handle the deficiencies of MCMC based Bayesian methods. MCMC algorithms are computationally expensive and the process of obtaining appropriate hyperparameters may be problematic for this algorithm. However, ICMM algorithms have the advantages of easy implementation and fast computation and make the best use of every single observation even if the sample size is small. The drawback of this algorithm is its tendency to generate many false negatives.

III. **Expectation-Maximization based Variable Selection (EMVS)**

As mentioned before, the second ingredient of Bayesian variable selection is extracting information from the posterior. The common method for this step is using the MCMC method, however, EMVS is a deterministic model proposed as a substitute for MCMC, which is more efficient regarding the solution time and characterizing sparse models [48]. EMVS is highly sensitive to the initial value which may result in not obtaining the global optimal, however, this issue can be handled by a deterministic annealing variant. EMVS is a flexible method and revealed to have a promising performance when the number of features is considerably greater than the existing data points [49].

In this section, we introduced a set of commonly used methods for feature selection. In the next section, we will focus on the first factor of feature subset selection methods mentioned in section 1.2.2.

1.3 Statistical Measures for Prediction Models

As mentioned in section 1.2.2, for every feature subset selection method, a criterion should be defined to draw a comparison between the subsets. Here, we introduce

some of these criteria.

1.3.1 Traditional Fit Tests

I. R^2 **and Pseudo-**R^2 **statistics**: In Gaussian regression, one can use R^2 statistics or adjusted-R^2 as a goodness-of-fit measure. The definition of R^2 is as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = \frac{RSS}{TSS}, \tag{1.9}$$

where $\hat{y}_i$ is the estimation of actual y_i for each i and $\bar{y}$ is their mean. Here, the terms RSS and TSS stand for residual sum of squares and total sum of squares, respectively.

The value of R^2 is between 0 and 1, and the higher the value is, the better the model is fitted in terms of the training error. The main concern with R^2 is that the training error can be a poor estimator of the test error and by adding the predictors to the model, R^2 will always increase. Hence, the highest R^2 is achieved by adding all the predictors to the model. Although this scenario leads to a low training error, it can cause overfitting meaning that the model performs well over the training data but not over the test data. However, lowering the test error is of the main interest [24]. Hence, adjusted-R^2 (1.10) is defined by statisticians in which adding unimportant features will decrease the value of adjusted-R^2, due to the presence of k as a penalty factor.

$$R^2_{adj} = 1 - \frac{\frac{SSE}{n-k-1}}{\frac{TSS}{n-1}}, \tag{1.10}$$

where n is the number of instances, k is the number of selected predictors (features) and SSE is sum of squared errors. Same as R^2, higher values of adjusted-R^2 is preferred. However, unlike R^2, the value of adjusted-R^2 does not always increase by adding predictors meaning that once the truly important features are selected, adding noise predictors might decrease the adjusted-R^2 value [24].

Unfortunately, R^2 statistics could not be applied for logistic regression as it does not provide any information to decide among models [38]. However, in terms of logistic regression, Pseudo-R^2 statistics have been considered by the statisticians. There are

different types of Pseudo-R^2 but the widely used form is given as in (1.11):

$$1 - \frac{LL_F}{LL_C}, \tag{1.11}$$

where LL_F is the full model and LL_C is the intercept-only model [38]. Another commonly used Pseudo-R^2 statistics is called the *adjusted McFadden* (see 2.17 for the formal definition) which we will also use in our optimization.

II. **Likelihood ratio statistics**

Likelihood ratio test is used to figure out if adding a feature or a group of features modifies the performance of the model. The formulation is given by (1.12):

$$G = -2(LL_r - LL_f), \tag{1.12}$$

where LL_f and LL_r are the log-likelihood of the full and reduced model, respectively [38].

1.3.2 Information Criteria Tests

Information criteria test is a measure to evaluate the goodness of the fitted model when used for prediction of the test data. Two of the important subclass of information criteria are Akaike information criteria (AIC) and Bayesian information criteria (BIC), where both of them aim to maximize likelihood function while considering different terms to penalize the number of features added to the model. Here, we provide the definition of each criteria [50].

$$AIC = -2LL + 2k, \quad BIC = -2LL + k \log n,$$

where, LL represents loglikelihood function. k and n represent the number of predictors and data points, respectively.

1.4 Histograms

Pearson introduced histograms for the first time as an approximating tool to represent the distribution of a given dataset [51]. In histograms, the whole range of data has to be divided into a number of intervals, called as *bins*. Hence, it is required to find the breaking points of these non-overlapping intervals. Bin width is the most important parameter of a histogram which should be predetermined in a proper way to capture the essential structure of the data. In fact, bin width manages *over smoothing* or *under smoothing* of the estimation meaning that providing insufficient details and abundant details, respectively [52].

In [53], histograms are defined as nonparametric density estimators that are being used to summarize the data. Converging to the true density function is a crucial issue and simultaneously a motivation to find the methods culminating in an optimal histogram, hence, by the notion of minimizing the integrated mean squared error, a method was proposed to obtain the optimal histogram [53]. In another research study, histogram is considered as piecewise-constant model of the probability density and an algorithm is proposed based on Bayesian probability theory [54]. Additionally, Akaike information criteria (AIC) is also used to obtain optimal histogram with the aim of decreasing the subjectivity when identifying the smoothing parameter [55].

In this section, we considered some of the existing models for the problems in this thesis and covered the basic related concepts. In Chapter 2, we will explain our approach to derive MICP models for both of the application problems of this thesis.

1.5 Contribution

In this chapter, to model the sparse logistic regression and optimal histogram construction problems, we propose mixed-integer exponential cone (MIEXP) models with the aim of maximizing the likelihood function and minimizing the Kullback-Leibler divergence (KL-divergence), respectively. To the best of our knowledge, there has been no attempt to solve these problems using MICPs. The performance of our models are promising when comparing to the available methods in the literature. We will discuss the details of our approach in the methodology section.

2. Methodology

In this chapter, we will develop two MICP models where the first models is dedicated to feature subset selection in logistic regression and the second is generating optimal histograms with the aim of maximizing the likelihood function and minimizing the Kullback-Leibler divergence, respectively. The common characteristic of both problems is that their objective functions are *Exponential Cone-representable* and their feasible regions are mixed-integer linear representable.

In the first problem, we aim to derive a model for sparse logistic regression problem which is a trending research avenue. This interest stems from the presence of datasets with categorical outputs in many fields of study including medical science, social science, finance, and portfolio management. Notice that, linear regression will have a poor performance in constructing a prediction model over such datasets since the dependent variable takes a finite set of values. Hence, our motivation in the first problem is to derive an exact method by developing an MICP formulation, which has not been considered in the literature. Next, by adopting information criteria measures and regularization, we will modify the model and analyze the performance of each version.

In the second problem, we develop a model that can be used to obtain optimal histograms. In [51], Pearson introduced histograms for the first time as an approximating tool to represent the distribution of a given dataset. Bin width is the most important parameter of a histogram as it manages *over smoothing* or *under smoothing* of the estimation, meaning that providing insufficient details and abundant details, respectively [52]. In this model, our aim is to find the optimal bin width and at the same time breaking points with the aim of minimizing the KL-divergence of the given integer data to the generated histogram.

In Section 2.1, we will briefly explain linear regression and MILP model for Sparse Linear Regression which was our starting point for the sparse logistic regression model we propose in section 2.2. Finally, Section 2.3 will be dedicated to the optimal histogram construction model.

2.1 Linear Regression

Regression models are among the most important prediction models for both quantitative (taking numerical values) and qualitative (taking values from a finite set) variables. A regression model relates a dependent variable y to a function of known x and unknown α. Suppose that we have n data points $(\hat{y}_i; \hat{x}_{i1}, \hat{x}_{i2}, ..., \hat{x}_{iq})$, for $i = 1, \ldots, n$ in which $\hat{y}_i$'s are the values of dependent variables and $\hat{x}_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$ is the value of independent variable for the i^{th} data point. The following linear model is defined to predict the value of the dependent variable y given the independent variables x:

$$y = \alpha_0 + \alpha_1 x_1 + ... + \alpha_p x_p + \epsilon, \tag{2.1}$$

where ϵ is the error term for the data points and α_0 is the intercept.

Definition 2.1. *Let $q \geq 1$. Then, the ℓ_q norm of $x \in \mathbb{R}^n$ is given by*

$$||x||_q = \left(\sum_{i=1}^{n} |x_i|^q \right)^{\frac{1}{q}}.$$

Let $\hat{X}_{n \times p}$ denote a matrix whose (i, j) entry is the value of the dependent variable j in the i^{th} observation, $i = 1, \ldots, p$, $j = 1, \ldots, n$. A common procedure to find the values of the unknown coefficients α_i's is to minimize the sum of squared error:

$$\min_{\alpha} \|\hat{y} - \hat{X}^T \alpha\|_2^2, \tag{2.2}$$

where $\hat{X}$ and $\hat{y}$ are given. We note that it is possible to use other norms including ℓ_1-norm and ℓ_∞-norm.

2.1.1 Sparse Model for linear regression

Definition 2.2. *([24]) Variable selection is the method of excluding irrelevant variables which are not influential to the response in regression models with the aim of increasing the interpretability of the model.*

The main goals of variable (feature) selection are improving prediction accuracy and making the model easier to interpret. Although one may assume that the more variables are included to construct a model, the better the prediction accuracy is, it may lead to a phenomenon known as *overfitting*. Overfitting occurs when the model performs well for the training data and reveals a poor performance over the test data. Moreover, by eliminating irrelevant variables, the model might be easier to interpret. Hence, constructing sparse models is of a great interest. In the next part, we will discuss a model for sparse linear regression which is presented in [56].

First, for the ease of notation we define the operator "$\circ$" as the element-wise product of a matrix, also known as the *Hadamard product* [57].

Definition 2.3. *Given two matrices $A_{m\times n}$ and $B_{m\times n}$, the elements of matrix $C_{m\times n} = A \circ B$ is defined as*

$$c_{ij} = a_{ij} b_{ij}, \quad i = 1, \dots, m \quad j = 1, \dots, n.$$

To obtain a sparse (2.1), we define a binary vector $z \in \{0,1\}^{p+1}$ in which j^{th} element z_j takes value one if the j^{th} independent variable is chosen and zero otherwise (z_0 is associated to the intercept). Therefore, problem (2.2) is modified as the following:

$$\min_{\alpha, z} \ \|\hat{y} - \hat{X}^T(\alpha \circ z)\|_q \tag{2.3a}$$

$$\text{s.t. } z \in \{0,1\}^{p+1}. \tag{2.3b}$$

Notice that this model is non-linear due to the presence of the bilinear terms α, z, and the norm function. In the sequel, we will first linearize the bilinear terms by benefiting from the fact the z_j is binary and assuming that the magnitude of α_j is bounded by some constant M. After this common step for all the norms considered, we will obtain MICP or MILP formulations depending on the particular norm under consideration.

First, we define a new set of decision variables w such that:

$$w = \alpha \circ z,$$

where z_0 always takes value 1 to ensure the intercept to be always chosen.

Now, we can rewrite (2.3):

$$\min_{w,\alpha,z} \|\hat{y} - \hat{X}^T w\|_q \tag{2.4a}$$

$$\text{s.t. } -Mz \leq w \leq Mz \tag{2.4b}$$

$$\alpha - (e-z)M \leq w \leq \alpha - (e-z)(-M) \tag{2.4c}$$

$$z \in \{0,1\}^{p+1}. \tag{2.4d}$$

Notice that if an independent variable j is selected, then the corresponding w_j will be equal to α_j and it will be zero otherwise. The vector $e \in \mathbb{R}^n$ represents the vector of ones.

Following the previous discussion and using (2.4), sparse linear regression models will be presented with respect to ℓ_1, ℓ_2, and ℓ_∞ norms.

I. ℓ_1-based sparse linear regression model is given as in (2.5):

$$\min_{w,\alpha,z,u} \sum_{i=1}^{n} u_i \tag{2.5a}$$

$$\text{s.t. } y_i - x_i w_i \leq u_i \tag{2.5b}$$

$$-y_i + x_i w_i \leq u_i \tag{2.5c}$$

$$(2.4b), (2.4c), (2.4d).$$

II. ℓ_2-based sparse linear regression model is shown in (2.6):

Although ℓ_2 norm cannot be linearized, (2.6) can be solved as a second-order cone.

$$\min_{w,\alpha,z,u} \sum_{i=1}^{n} u_i \tag{2.6a}$$

$$\text{s.t. } (y_i - x_i w_i)^2 \leq u_i \tag{2.6b}$$

$$(2.4b), (2.4c), (2.4d).$$

III. ℓ_∞-based sparse linear regression model is given by (2.7):

$$\min_{w,\alpha,z,u} \; u \tag{2.7a}$$

$$\text{s.t. } y_i - x_i w_i \leq \tag{2.7b}$$

$$-y_i + x_i w_i \leq u \tag{2.7c}$$

$$(2.4b), (2.4c), (2.4d).$$

Finally, adding the following constraint to above models, we enforce an upper bound for the number of variables to be selected:

$$\sum_{i=1}^{n} z_i \leq K. \tag{2.8}$$

In (2.8), K is treated as a parameter rather than a variable, which means that the number of variables to be selected has to be known in advance and this can be considered as a disadvantage. However, there exist some approaches including Goodness of Fit (GOF) measure discussed in Section 1.3.2 to overcome this issue. As the sparse linear regression is not the main problem of interest in this thesis, we assume prior information on the number of variables to be selected through the implementation to just provide an insight for the sparse logistic regression model. See Appendix A for a preliminary experimental analysis of sparse linear regression.

2.2 Logistic Regression

Limitations of ordinary least squares regression to handle the binary variables as an outcome is a motivation to consider logistic regression for the classification purpose [25]. There are some important research areas ranging from medical science to social science in which there exist datasets that the value of dependent variable is limited to a finite set of values. The most common example is the specific group of medical tests which has positive and negative outcomes for the dependent variable. Additionally, logistic regression is not limited to the binary output and can handle the datasets in which the dependent variable has more than two classes. This type of logistic regression is called as "multinomial logistic regression".

Generally, in regression models, selecting truly important vectors can help to fit

a model which is more interpretable, especially when the number of independent variables is high [25]. As mentioned in Definition 2.2, this procedure is known as variable (feature) selection. Exact techniques and heuristic techniques are the two main approaches for dealing with the feature selection problem. Although exact techniques provide optimal solution, due to the fact that feature selection problem is NP-Hard [58], they cannot be applied efficiently for large datasets. For these settings, heuristic methods which provide proper solutions in a reasonable time might be preferred. Recently, feature subset selection in regression models have been the area of interest for many research studies. As an illustration, in [41], a mixed-integer second order cone programming model is introduced for feature subset selection in linear regression. Considering the logistic regression, [58] proposed a heuristic method based on Tabu search and [39] introduced an exact method based on Bayesian methodology. Our contribution in this part of thesis will be to provide an MICP formulation for feature subset selection in logistic regression. We will first, present some basic definitions.

Logistic Regression determines the conditional probability of the class Y given that $X = x$.

$$P(Y|X = x).$$

Definition 2.4. *Given $p \in [0,1)$, odds ratio is defined as $\frac{p}{1-p}$.*

Unlike the probability, odds is not bounded by 0 and 1. By taking the logarithm, we have a function known as log-odds:

$$\log(\text{odds}) = \log\left(\frac{p}{1-p}\right) \tag{2.9}$$

Observe that log-odds function can take any value from $-\infty$ to ∞. Hence, we can fit a linear regression model as follows:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \alpha_0 + \alpha_1 x_1 + ... + \alpha_n x_n,$$

where $p(x) = p(Y = 1|X = x)$. In the next parts, we will discuss background materials required to derive the sparse logistic regression model in Sections 2.2.1 and 2.2.2. Finally, we will present the model in Section 2.2.3.

2.2.1 Exponential cone

Exponential cone programming is an important subclass of conic programming which is also used to formulate both of the models in this thesis. Hence, we will discuss the exponential cones in this part.

Definition 2.5. *The exponential cone is defined as the set:*

$$K_{exp} = Cl\{x \in \mathbb{R}^3 : x_1 \geq x_2 e^{\frac{x_3}{x_2}}, \quad x_2 > 0\},$$

where Cl is the closure of a set.

Definition 2.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function. Then $g(x,t) = t\ f(\frac{x}{t}) : \mathbb{R}^n \times \mathbb{R}_{++} \to \mathbb{R}$ is called the perspective of f.*

We present the proof of the following lemma for completeness.

Lemma 2.1. *If f is a convex function, then the perspective of f is also a convex function.*

Proof. To prove this lemma we have to show that:

$$g(\lambda x_1 + (1-\lambda)x_2, \lambda t_1 + (1-\lambda)t_2) \leq \lambda g(x_1,t_1) + (1-\lambda) g(x_2,t_2).$$

By supposition we know that f is convex. Then we have:

$$\begin{aligned}
&g(\lambda x_1 + (1-\lambda)x_2, \lambda t_1 + (1-\lambda)t_2) \\
&= [\lambda t_1 + (1-\lambda)t_2] f(\frac{\lambda x_1 + (1-\lambda)x_2}{\lambda t_1 + (1-\lambda)t_2}) \\
&= [\lambda t_1 + (1-\lambda)t_2] f(\frac{\lambda x_1}{\lambda t_1 + (1-\lambda)t_2} + \frac{(1-\lambda)x_2}{\lambda t_1 + (1-\lambda)t_2}) \\
&= [\lambda t_1 + (1-\lambda)t_2] f(\frac{\lambda t_1}{[\lambda t_1 + (1-\lambda)t_2]} \frac{x_1}{t_1} + \frac{(1-\lambda t_2)}{[\lambda t_1 + (1-\lambda)t_2]} \frac{x_2}{t_2}) \\
&\leq [\lambda t_1 + (1-\lambda)t_2] (\frac{\lambda t_1}{[\lambda t_1 + (1-\lambda)t_2]} f(\frac{x_1}{t_1}) + \frac{(1-\lambda)t_2}{[\lambda t_1 + (1-\lambda)t_2]} f(\frac{x_2}{t_2})) \\
&= \lambda t_1 f(\frac{x_1}{t_1}) + (1-\lambda) t_2 f(\frac{x_2}{t_2}) \\
&= \lambda g(x_1,t_1) + (1-\lambda) g(x_2,t_2).
\end{aligned}$$

The above inequality holds due to the convexity of function f. □

Let $f(x_3) = e^{x_3}$. Then, by Definition 2.6, the perspective of f is $g(x_3, x_2) = x_2 e^{\frac{x_3}{x_2}}$.

Hence, the set:

$$Cl\{x \in \mathbb{R}^3 : x_1 \geq x_2 e^{\frac{x_3}{x_2}}, \quad x_2 > 0\},$$

is the closure of the epigraph of function g. In fact, the exponential cone is the closure of the epigraph of the perspective of the exponential function.

Lemma 2.2. *A function is convex if and only if the epigraph of the function is convex.*

Exponential function is a convex function and by Lemma 2.1 we conclude that $g(x_1, x_2)$ is a convex function as well. Finally, the convexity of the exponential cone can be proved by Lemma 2.2.

Considering the convex optimization problems, it is possible to represents an important subset of these problems in the form of conic programming and solve them using the existing solvers. Exponential cone is one of the widely used cones to transfer a convex optimization problem into an equivalent conic programming problem in the presence of exponential and logarithm functions. Logistic regression, Sparse logistic regression, and Geometric programming are among the problems which can be formulated as an exponential cone programming [59]. In this study, we will transfer the Sparse logistic regression and the Optimal histogram construction problems into their equivalent exponential cone programming versions.

The following definition will be the key in our analysis.

Definition 2.7. *The function* $f(\theta) = -\log(\frac{1}{1+e^{-\theta^T x}})$ *is called logistic cost function.*

Proposition 2.1. *The conic representation of logistic cost function is given by [60]*

$$\begin{cases} e^{-\theta^T x - t} \leq u \Rightarrow (u, 1, -(\theta^T x + t)) \in K_{exp} \\ e^{-t} \leq v \Rightarrow (v, 1, -t) \in K_{exp} \\ \frac{1}{u+v} \geq 1 \Rightarrow u + v \leq 1 \end{cases}$$

Proof. We have

$$t \geq \log(\frac{1}{1+e^{-\theta^T x}})^{-1} \Rightarrow e^t \geq 1 + e^{-\theta^T x} \Rightarrow \frac{e^t}{1+e^{-\theta^T x}} \geq 1 \Rightarrow \frac{1}{\underbrace{e^{-t}}_{v} + \underbrace{e^{-\theta^T x - t}}_{u}} \geq 1$$

$$\Rightarrow e^{-t} \leq v, \quad e^{-\theta^T x - t} \leq u, \quad \frac{1}{u+v} \geq 1.$$

□

2.2.2 Logistic Regression as an Exponential Cone Program

In logistic regression, one tries to fit the unknown coefficient vector (θ in 2.10) as a maximum likelihood estimator. The likelihood function is given by

$$L(\theta) = \prod_{i=1}^{n} f_\theta(x_i)^{y_i}(1 - f_\theta(x_i))^{(1-y_i)}, \tag{2.10}$$

where f is Probability Density Function (PDF) while x and y are the vectors of independent variable and dependent variable, respectively.

Log-likelihood function (2.11) is obtained by taking logarithm of (2.10) :

$$\begin{aligned} LL(\theta) &= \sum_{i=1}^{n} \ln\left(f_\theta(x_i)^{y_i}(1 - f_\theta(x_i))^{(1-y_i)}\right) \\ &= \sum_{i=1}^{n} \ln f_\theta(x_i)^{y_i} + \sum_{i=1}^{n} \ln(1 - f_\theta(x_i))^{(1-y_i)} \\ &= \sum_{i=1}^{n} y_i \ln f_\theta(x_i) + \sum_{i=1}^{n} (1 - y_i) \ln(1 - f_\theta(x_i)). \end{aligned} \tag{2.11}$$

Considering the Bernoulli distribution for the dependent variable y, we can replace $f_\theta(x_i)$ in (2.11) with $P(x) = P(Y = 1|X = x)$. The value of $P(x)$ is equal to $\frac{1}{1+e^{-\theta^T x_i}}$ and can be calculated by Logit function in equation 2.9.

By replacing $f_\theta(x_i)$ in (2.11) with $P(x)$, we have:

$$\begin{aligned} LL(\theta) &= -\sum_{i=1}^{n} y_i \ln(1 + e^{-\theta^T x_i}) + \sum_{i=1}^{n} (1 - y_i)[-\theta^T x_i - \ln(1 + e^{-\theta^T x_i})] \\ &= -\sum_{i=1}^{n} \ln(1 + e^{-\theta^T x_i}) - \sum_{i=1}^{n} (1 - y_i)\theta^T x_i. \end{aligned} \tag{2.12}$$

Now, to estimate the unknown vector of θ, we maximize the log-likelihood function:

$$\max_{\theta} \quad \sum_{i=1}^{n} \ln(1 + e^{-\theta^T x_i})^{-1} - \sum_{i=1}^{n} (1 - y_i)\theta^T x_i. \tag{2.13}$$

With the aim of transforming problem (2.13) into a conic programming problem,

first, we reformulate it as:

$$\max_{\theta,t} \quad \sum_{i=1}^{n} t_i - \sum_{i=1}^{n}(1-y_i)\theta^T x_i \tag{2.14a}$$

$$\text{s.t.} \quad \ln(1+e^{-\theta^T x_i})^{-1} \geq t_i \quad i=1,\ldots,n. \tag{2.14b}$$

Second, we transform constraint (2.14b) into its equivalent conic form using Proposition 2.1. Finally, the equivalent conic form of logistic regression problem will be as:

$$\max_{\theta,t,u,v} \quad \sum_{i=1}^{n} t_i - \sum_{i=1}^{n}(1-y_i)\theta^T x_i \tag{2.15a}$$

$$\text{s.t.} \quad u_i + v_i \leq 1 \qquad i=1,\ldots,n, \tag{2.15b}$$

$$(v_i, 1, t_i) \in K_{exp} \qquad i=1,\ldots,n, \tag{2.15c}$$

$$(u_i, 1, -\theta^T x_i + t_i) \in K_{exp} \quad i=1,\ldots,n. \tag{2.15d}$$

In this section, we provided the conic formulation of logistic regression problem. In Section 2.2.3, we will derive a model for feature subset selection in logistic regression by adding a group of constraints to model (2.15). The aims of the constraints to be added are to force the model to select the truly important features and at the same time to estimate the value of corresponding elements of coefficient vector.

2.2.3 Sparse Logistic Regression Model

In sparse logistic regression model, the aim is to find the best subset of independent variables in the same fashion as what was discussed for sparse linear regression in Section 2.1.1. Hence, similar to the sparse linear regression, we define a binary vector $z \in \{0,1\}^p$ (intercept excluded) in which jth element z_j takes value one if the jth independent variable is chosen and zero otherwise . Finally, the MIEXP model for feature subset selection in logistic regression is given as follows:

$$\max_{\theta,t,u,v} \quad \sum_{i=1}^{n} t_i - \sum_{i=1}^{n}(1-y_i)w^T x_i \tag{2.16a}$$

$$\text{s.t.} \quad u_i + v_i \leq 1 \qquad i = 1,\ldots,n, \tag{2.16b}$$

$$(v_i, 1, +t_i) \in K_{exp} \qquad i = 1,\ldots,n, \tag{2.16c}$$

$$(u_i, 1, -w^T x_i + t_i) \in K_{exp} \qquad i = 1,\ldots,n, \tag{2.16d}$$

$$\sum_{j=1}^{p} z_j \leq k, \tag{2.16e}$$

$$-Mz \leq w \leq Mz, \tag{2.16f}$$

$$z_j \in \{0,1\}^p \tag{2.16g}$$

In this model the number of features to be selected is not a variable but a parameter. In the next section, we present modified versions of this model based on some GOF measures.

2.2.4 Modified Sparse Logistic Regression Model Using GOF Measures

As discussed before, the requirement to know the number of features to be selected in advance, can be considered as a disadvantage, especially when the number of features is high. To overcome this issue, we can use different goodness of fit measures including adjusted McFadden, Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) discussed in Section 1.3. In the next step, we will present the modification of model (2.16) based on these measures.

2.2.4.1 Adjusted McFadden Modification

Adjusted McFadden measure is defined as:

$$R^2_{adj} = 1 - \frac{logL(M_{full}) - k}{logL(M_{null})}, \tag{2.17}$$

where M_{full} is full model and M_{null} represents the case in which only intercept is used to construct a model [38]. In fact by adding term k, we are penalizing adding extra predictors to the model [25]. We note that in the Mcfadden model, "k" which

is the number of features to be selected is no longer a parameter but a variable. The modified model is as follows:

$$\max_{\theta,t,u,v,k} \quad \sum_{i=1}^{n} t_i - \sum_{i=1}^{n}(1-y_i)w^T x_i - k \tag{2.18a}$$

$$\text{s.t.} \quad (2.16b) - (2.16g) \tag{2.18b}$$

The same as adjusted-R^2 discussed in Section 1.3.1, we are interested in higher values of adjusted McFadden. Therefore, we have to minimize the term $\frac{logL(M)-k}{logL(M_{null})}$ in (2.17). As the value of likelihood is between zero and one, the logarithm of likelihood is less than equal to zero. Hence, considering the fact that $logL(M_{null})$ is a negative constant term, we just have to maximize the term $logL(M) - k$. Consequently, problem (2.18) is the modified version of problem (2.16) with respect to adjusted McFadden measure.

2.2.4.2 AIC and BIC Modification

In this section, information criteria measures of AIC and BIC will be considered to modify the model. The same as McFadden, the only change will be in the objective function.

Definition 2.8. *AIC and BIC measures are given by:*

$$AIC = -2LL + 2k, \qquad BIC = -2LL + ln(n)k$$

where k is the number of features and n is the number of instances [38].

By Definition 2.8, the modified models based on AIC and BIC are given by (2.19) and (2.20), respectively.

$$\min_{\theta,t,u,v,k} \quad -2\sum_{i=1}^{n} t_i + 2\sum_{i=1}^{n}(1-y_i)w^T x_i + 2k \tag{2.19a}$$

$$\text{s.t.} \quad (2.16b) - (2.16g) \tag{2.19b}$$

$$\min_{\theta,t,u,v,k} \quad -2\sum_{i=1}^{n} t_i + 2\sum_{i=1}^{n}(1-y_i)w^T x_i + ln(n)k \tag{2.20a}$$

$$\text{s.t.} \quad (2.16b) - (2.16g) \tag{2.20b}$$

Note that, AIC and McFadden are equivalent in the sense of optimization as if we multiply the objective function of (2.19) by the constant term $-\frac{1}{2}$, we will have the same objective function as in (2.18.)

We note that, in the case of existing highly correlated independent variables (features) in problems (2.18)-(2.20), a high variance model will be achieved and the higher the correlation is, the more unrealistic the model will be [61]. To handle this issue, a family of methods known as *regularization techniques* are introduced in the literature (see [62] for different methods of regularization and detailed theory). Moreover, regularization prevents overfitting, especially in the case that the size of training data is relatively small [63]. As a result, to avoid both overfitting and generating high variance models, we also considered the ℓ_2-regularization form of the problems (2.18)-(2.20) as well. This is a commonly used regularization technique and is also considered by [60] for logistic regression formulation.

Problem (2.21) represents the model in [60].

$$\min_{\theta,t,u,v} \quad -\sum_{i=1}^{n} t_i + \sum_{i=1}^{n}(1-y_i)w^T x_i + \lambda\|\theta\|_2, \tag{2.21a}$$

$$\text{s.t.} \quad (2.16b) - (2.16d)$$

The term $\lambda\|\theta\|_2$ in problem (2.21) is the mentioned ℓ_2-regularization term. From statistic perspective, this term can be considered as prior knowledge indicating that θ should not be very large while from the optimization perspective, it provides a trade-off between having a small θ and solving the problem [62]. In Chapter 3, we will report and discuss the computational results of these models over a group of toy examples and a bunch of benchmark datasets. In the next section, we present commonly used metrics to quantify the performance of binary classifiers.

2.2.5 Metrics For Binary Classifiers Performance

In this section, we will explain commonly used metrics which are required to evaluate the binary classifiers.

Definition 2.9. *([64]) An $m \times m$ matrix defined for a classifier to represent the predicted and actual classification, is called a confusion matrix. The rows of this matrix show the actual class and the columns are dedicated to the predicted class. Here, m is the number of existing classes.*

As we consider binary logistic regression in our study, the confusion matrix will be as Table 2.1. Here, based on Definition 2.9, the value of m is equal to two.

		Predicted class	
		P (+)	N (-)
Actual class	P (+)	TP	FN
	N (-)	FP	TN

Table 2.1 Confusion matrix for binary classifiers.

The definition of the terms used in Table 2.1 for the binary confusion matrix is provided in Table 2.2.

TP: true positive	FN: false negative
FP: false positive	TN: true negative

Table 2.2 Definitions of the confusion matrix elements.

Using the defined terms in Table 2.2, several metrics to quantify the performance of binary classifiers are introduced in Table 2.3 [65].

Accuracy	$\frac{(TP+TN)}{P+N}$
Error	$\frac{(FP+FN)}{P+N}$
Positive Predicted value	$\frac{TP}{TP+FP}$
Negative Predicted value	$\frac{TN}{TN+FN}$
Sensitivity	$\frac{TP}{P}$
Specificity	$\frac{TN}{N}$
FP Rate	$\frac{FP}{N}$
F Score	$\frac{2*Precision*Sensitivity}{Precision+Sensitivity}$

Table 2.3 Metrics for evaluating performance of the binary classifiers.

In Chapter 3, we will present the computational results of our proposed models for sparse logistic regression and will evaluate their performance using metrics in Table 2.3.

2.3 Optimal Histogram Construction

This section is dedicated to our MICP model for *Optimal Histogram Construction*. As mentioned in Section 1.4, several models have been proposed to find the optimal bins width, when constructing histograms. However, to the best of our knowledge, there has been no attempt to formulate this problem as an MICP. Hence, we propose an MIEXP model for this problem with the aim of minimizing KL-divergence as it is a measure to quantify how close an arbitrary distribution is to the true distribution. In this model, for a given data, we will fit a histogram in such a way that the KL-divergence of data to the fitted histogram is minimized. Note that both probability distributions must be defined on the same probability space.

In the next part, we will provide definition and conic representation of the KL-divergence function.

2.3.1 Kullback and Leibler (KL) Divergence

Kullback and Leibler introduced a sufficiency criteria (KL-divergence) by considering the divergence between statistical populations [66]. In fact, it measures how close the estimated probability distribution is to the reference distribution. The definition of KL-divergence for discrete random variables defined over the same probability space is as follows:

Definition 2.10. *KL-divergence for two discrete probability distribution P and Q is defined as:*

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right).$$

Here, $\mathcal{X}$ is the probability space that Q and P are defined on.

We note that the KL divergence does not define a distance between two distributions since it is not symmetric in general, that is,

$$D_{KL}(P||Q) \neq D_{KL}(Q||P)$$

The KL-divergence is also used extensively in machine learning and information theory, and is interpreted as the information gain if the Q is substituted by P [66].

In this model, we are interested in minimizing the discrete KL-divergence function provided in Definition 2.10:

$$\min_{Q} \sum_{x\in\mathcal{X}} P(x)\log\left(\frac{P(x)}{Q(x)}\right). \tag{2.22}$$

Equivalent optimization problem for problem (2.22) is as follows:

$$\min_{t,Q} \quad \sum_{x\in\mathcal{X}} t(x) \tag{2.23a}$$

$$\text{s.t.} \quad t(x) \geq P(x)\log\left(\frac{P(x)}{Q(x)}\right) \quad \forall x \in \mathcal{X}. \tag{2.23b}$$

Proposition 2.2. *([60]) Exponential cone representation of the inequality* $t \geq x\log\left(\frac{x}{y}\right)$ *is given by*

$$(y, x, -t) \in K_{exp}$$

Proof.

$$t \geq x\log\left(\frac{x}{y}\right) \Rightarrow t \geq -x\log\left(\frac{y}{x}\right) \Rightarrow e^{\frac{-t}{x}} \leq \frac{y}{x} \quad \Rightarrow xe^{\frac{-t}{x}} \leq y$$
$$\Rightarrow (y, x, -t) \in K_{exp}$$

□

Finally, using Proposition 2.2, MIEXP formulation of problem (2.23) is given by

$$\min_{Q,t} \quad \sum_{x\in\mathcal{X}} t(x) \tag{2.24a}$$

$$\text{s.t.} \quad (Q(x), P(x), -t(x)) \in K_{exp} \quad \forall x \in \mathcal{X} \tag{2.24b}$$

In this section, we provided the conic representation of the KL-divergence function. In Section 2.3.2, we will extend this model to introduce a MIEXP model for the optimal histogram construction problem by adding a set of constraints. In this model, we assume that we have m integer observations to be divided into k bins. The probability of each observation θ_i is defined as p_i. The aim is to associate each observation θ_i, $i = 1, \ldots, m$ with one of the k bins while minimizing the Kullback-Leibler divergence of given data to our fitted histogram. Without loss of generality, we assume that θ_i's are ordered.

2.3.2 Optimal Histogram Construction Problem

In this section, we will extend problem (2.24) by considering a set of assumptions to obtain an MIEXP model for optimal histogram construction problem. These assumptions are I) ensuring that at least one observation is dedicated to each bin, II) allocating each observation to exactly one bin, and III) if observation θ_i is allocated to bin j, then observation θ_{i+1} must be either allocated to bin j or $j+1$. Furthermore, we need to define a set of decision variables as follows:

$$\begin{cases} x_{ij} = 1 & \text{if observation } i \text{ belongs to bin } j \\ x_{ij} = 0 & \text{O.W} \end{cases}$$

$$q_i = \text{probability of } \theta_i \text{ in estimated distribution}$$

$$Q_j = \text{probability of bin } j \text{ in estimated distribution}$$

To clarify, we provide the following example. Let us define the vector of p=[0.1, 0.2, 0.4, 0.3], which represents the probability of observations θ_i's. If we assume $k = 2$ and the model associates the first two observations to the first bin and the remainder to the second bin, then the value of Q_1 and Q_2 are 0.3 and 0.7, respectively. Consequently, the values of q_1 and q_2 will be equal to 0.15 while those of q_3 and q_4 will be equal to 0.35.

Finally, the MIEXP model for this problem is given as:

$$
\begin{aligned}
&(2.25a) && \min_{q,t,x,Q} && \sum_{i=1}^{m} t_i && \\
&(2.25b) && \text{s.t.} && (q_i, p_i, -t_i) \in K_{exp} && \forall i, \\
&(2.25c) && && \sum_{i=1}^{m} x_{ij} \geq 1 && \forall j, \\
&(2.25d) && && \sum_{j=1}^{k} x_{ij} = 1 && \forall i, \\
&(2.25e) && && x_{ij} \leq x_{i+1,j} + x_{i+1,j+1} && \forall i, j, \\
&(2.25f) && && \sum_{j'=1}^{j} x_{ij'} + \sum_{j''=j+2}^{k} x_{i+1,j''} \leq 1 && \forall i, j, \\
&(2.25g) && && Q_j = \sum_{i'=1}^{m} p_{i'} x_{i'j} && \forall j, \\
&(2.25h) && && \frac{Q_j}{\sum_{i'=1}^{m} x_{i'j}} - (1 - x_{ij}) \leq q_i \leq \frac{Q_j}{\sum_{i'=1}^{m} x_{i'j}} + (1 - x_{ij}) && \forall i, j,
\end{aligned}
$$

where, $i = 1, \ldots, m$, $i' = 1, \ldots, m$, and $j = 1, \ldots, k$. The purpose of constraints (2.25c) and (2.25d) are to satisfy the assumptions I and II, respectively while both constraints (2.25e) and (2.25f) are added to the model to satisfy assumption III.

Observe that constraint (2.25h) is nonlinear. Therefore, to linearize this constraint, we write it as follows:

$$
\underbrace{Q_j - (1 - x_{ij}) \sum_{i'=1}^{m} x_{i'j}}_{\text{Expression 1}} \leq \underbrace{q_i \sum_{i'=1}^{m} x_{i'j}}_{\text{Expression 2}} \leq Q_j + (1 - x_{ij}) \sum_{i'=1}^{m} x_{i'j}.
$$

Linearization of Expression 1: We define new variables $u_{ii'j}$ such that:

$$
(1 - x_{ij}) x_{i'j} = u_{ii'j} \qquad \forall i, \forall i', \forall j
$$

Then the linear form will be as follows by defining new variables v_i:

$$
\begin{aligned}
&(2.26) && v_i = \sum_{i'=1}^{m} u_{ii'j} && \forall i, \forall j \\
&(2.26) && 0 \leq u_{ii'j} \leq x_{i'j} && \forall i, \forall i', \forall j \\
&(2.26) && (1 - x_{ij}) - (1 - x_{i'j}) \leq u_{ii'j} \leq (1 - x_{ij}) && \forall i, \forall i', \forall j
\end{aligned}
$$

Linearization of Expression 2: We define new variables $\alpha_{ii'j}$ such that:

$$\alpha_{ii'j} = q_i x_{i'j} \qquad \forall i, \forall i', \forall j$$

Then the linear form will be as follows by defining new variables w_i:

$$\begin{aligned} & w_i = \sum_{i'=1}^{m} \alpha_{ii'j} && \forall i, \forall j \\ & 0 \leq \alpha_{ii'j} \leq x_{i'j} && \forall i, \forall i', \forall j \\ & q_i - (1 - x_{i'j}) \leq \alpha_{ii'j} \leq q_i && \forall i, \forall i', \forall j \end{aligned} \tag{2.27}$$

Finally, the MIEXP model for optimal histogram construction is given by (2.28):

$$\min_{q,t,x,Q} \quad \sum_{i=1}^{m} t_i \tag{2.28a}$$

$$\text{s.t.} \quad (2.25b) - (2.25g), \tag{2.28b}$$

$$(2.26), (2.27), \tag{2.28c}$$

$$Q_j - v_i \leq w_i \leq Q_j + v_i \quad \forall i, j \tag{2.28d}$$

In Chapter 3, we will present the computational results of this model and evaluate its performance for different distributions.

3. Results and Discussion

In this chapter, we will present and discuss the computational results for both the feature subset selection in logistic regression and optimal histogram construction models in detail. For the former problem, we will start with an analysis over a group of toy examples, which arms us with a primary insight about the performance of our model over different settings such as the numbers of features, observations, and correlations between features. In these settings, as we already know the truly important features, it is easier to analyze the success of the models. However, in practice we have no prior information over the features. Therefore, we will also evaluate the performance of our models over a bunch of benchmark datasets which have been considered by the related works in the literature. Regarding the second model, the aim is to evaluate the performance of our proposed method in terms of KL-divergence with the case in which the bin widths are equal. The motivation of this setting is that Python packages acts in the same fashion and generates equal bin widths when constructing a histogram. To show the superiority of our model, we will assume different probability distributions including Normal, Gamma, and Poisson for the observations. Then, we will compare the KL-divergence of our fitted model to the given data with that of equal bin widths.

Finally, we note that solver Mosek is used for the implementation of both problems due to its capability of supporting MIEXPs.

3.1 Computational Results For Sparse Logistic Regression

In this part, we will report the computational results of our MIEXP model for feature subset selection in logistic regression problem for a group of toy examples and benchmark datasets. First, we will explain two procedures which we consider for all the datasets followed by an explanation for the parameters that all the models

have in common.

Procedure 1:

As discussed before, to evaluate the performance of a model, one should consider its performance over the test data once the model is trained by the training data. Hence, with the aim of dividing the datasets into training and test data, we used 10-fold cross validation. In this setting, the dataset is divided into ten folds of equal size where each fold in turn is used for testing and the remainder for training [67].

Procedure 2:

For the evaluation process, we require a group of measures to quantify the performance of the models. For this purpose, we use accuracy, F-score, and NPV (see Table 2.3 for the definition of each term). Accuracy deals with the true positive and true negative predictions while the main concern of F-score is false negative and false positive predictions. Finally, NPV is a measure for false negative prediction. Moreover, we also include sparsity and Harmonic mean of accuracy and sparsity, defined in [39], as other metrics. Sparsity is given by

$$\text{Sparsity} = 1 - \frac{\text{Number of non-zero elements of a vector}}{\text{Number of total elements of a vector}}, \tag{3.1}$$

and Harmonic mean is defined as

$$\text{Harmonic mean} = \frac{2}{\frac{1}{\text{Accuracy}} + \frac{1}{\text{Sparsity}}}. \tag{3.2}$$

Harmonic mean quantifies the trade-off between intrepretability and prediction accuracy of a model, meaning that keeping accuracy as high as possible while including less numbers of predictors (features) to the model. Notice that we are interested in higher values of all the measures defined in this part. Throughout this thesis, we will report the mean value of each measure obtained from 10-fold cross validation. We used *Scikit-learn package* to perform cross validation [68].

In this part, we will provide the explanations of the common parameters of the models.

I. **Success Rate of Bernoulli Distribution**:

As mentioned in Section 2.2.2, we assume Bernoulli distribution for the dependent variable y. Remember that we are interested in finding $P(x) = P(Y = 1|X = x)$.

In this thesis, we fix the success parameter of Bernoulli distribution (p) equal to 0.5 meaning that if $P(x) = P(Y = 1|X = x) \geq 0.5$, the dependent variable y takes value one and zero otherwise. Therefore, having equation (1.7) and the value of coefficient estimated by the model, we can calculate this probability for each of the observations and label them as 0 or 1 based on the value obtained for probability. In fact, we compute the probability given θ_i's and mark the observations as 0 or 1. Then, we compare them with the real values of dependent variables and calculate the binary metrics with respect to the obtained classes for the observations by the model and real classes.

II. **Regularization Parameter** λ:

For the models in which we assume regularization for the coefficients (see model (2.21) for instance), the regularization parameter λ should be defined in advance. Although an arbitrary value can be assigned for it, the value of λ is highly dependent to dataset. Hence, we prefer to tune this parameter for each of the datasets using the *glmnet* package in solver *R* [69].

III. **Seed number:**

Throughout the implementation, with the aim of reproducing the same folds when performing cross validation for all the models for a given dataset, we use a fixed seed number. This parameter provides us the opportunity to fairly evaluate and compare the performance of all the models in this thesis as they use the same combinations for the test and training data.

IV. **Big-M:**

In our proposed models for feature selection, Big-M constraint in model (2.16) plays a crucial role. Finding the value of big-M which provides an upper bound for the coefficients is problematic as it should be greater than the largest true coefficient and at the same time not too large in a way that it causes significant increase in time of obtaining the desired optimality gap. In the following part we will present available solutions to tackle with this issue and our findings in this regard.

Finding the value of big-M is completely dependent to the the dataset. Therefore, one way is trial and error to find this value although it is not efficient. We refer the reader to [70] in which this approach is considered. An alternative is to use *SOS type 1* reformulation introduced in [56]. In this approach, big-M constraint is replaced by the following term based on our notation:

$$\text{SOS type } 1 : \{1 - z_j, w_j\} \qquad j = 1, \ldots, p. \tag{3.3}$$

SOS type 1 constraint (3.3) satisfies the condition in which no more than one variable can be nonzero (either w_j or $1-z_j$). Hence, it is equivalent to the big-M constraint. This method is free of choosing big-M value, however, we cannot implement this method in our models as solver MOSEK does not support SOS type 1 constraints.

To set the value of big-M, we also try to extend the data-driven method for sparse linear regression proposed in [56] to our sparse logistic regression model. Inspired by the mentioned study, solving the following sets of optimization problems will provide the sufficient value for big-M.

$$\max_{\theta} \quad \theta_j \tag{3.4a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \ln(1+e^{-\theta^T x_i})^{-1} - \sum_{i=1}^{n}(1-y_i)\theta^T x_i \leq UB. \tag{3.4b}$$

$$\min_{\theta} \quad \theta_j \tag{3.5a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \ln(1+e^{-\theta^T x_i})^{-1} - \sum_{i=1}^{n}(1-y_i)\theta^T x_i \leq UB. \tag{3.5b}$$

Here, UB is an upper bound for the optimal value of problem (2.15) and θ_j represents the j^{th} element of the coefficient vector θ. Let u_j^+ and u_j^- for $j=1,\ldots,p$, be the optimal solutions for (3.4) and (3.5), respectively. Then, the sufficient value for big-M can be set as:

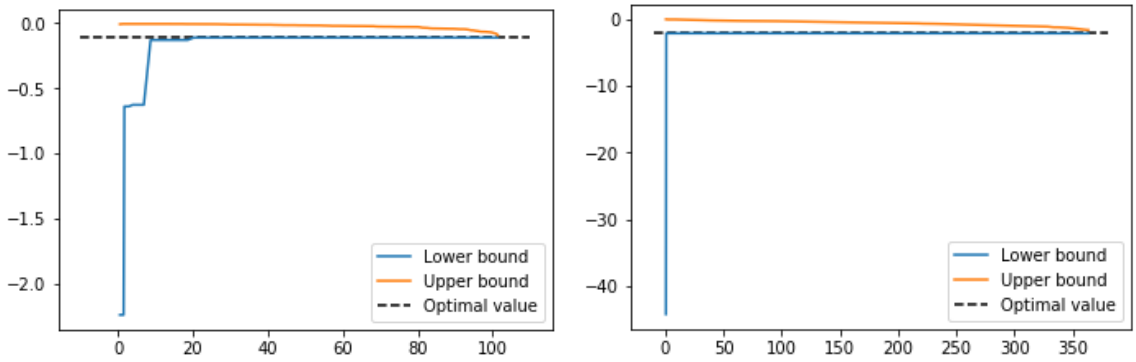
$$\text{big-}M := \max_j\{|u_j^+|, |u_j^-|\}.$$

In sparse logistic regression, the UB in (3.4) and (3.5) can be set to zero. The reason is, in model (2.15), we are maximizing the log-likelihood function. knowing that the likelihood function generates values between zero and one, it can be concluded that zero provides an upper bound for the log-likelihood function.

Finally, we note that at least one of problems (3.4) and (3.5) is unbounded. In fact, by ignoring trivial case where $x_i = 0$, if observation $x_i > 0$ problem (3.4) will be unbounded and if $x_i < 0$, then problem (3.5) will be unbounded. This analysis shows that it is not easy to find an appropriate value for big-M in advance. Therefore, in our experimental analysis, we solve our MIXEP models with different big-M values such as 10, 100, and 1000.

V. **Termination Criteria**:

In case of high dimensionality, especially when he number of observations is far less than the number of features, the solver may fail to obtain the desired optimality gap. Notice that adding constraint (2.16e), to the problem (2.15), makes it NP-Hard [56]. Therefore, we set a time based termination criteria throughout the implementation. We also mention that for some data sets the lower bound immediately reaches to the optimal value, however, it takes time to certify the optimality by the upper bound. (Recall that we have a maximization problem in this case). We provide Figure 3.1 to visualize this phenomenon.



(a) $p = 70, n = 25, k = 5$ (b) $p = 100, n = 100, k = 7$

Figure 3.1 The convergence of lower bound and upper bound by time. This figure is provided for two toy examples. Although, the lower bound reaches the optimal value in a few seconds, it takes time for the upper bound to certify optimality.

Based on Figure 3.1, for the cases in which there exist high optimality gap, it is possible that the incumbent feasible solution is (nearly) optimal or nearly optimal solution, however this may not be proven in general.

3.1.1 Experimental Analysis for the Toy Examples

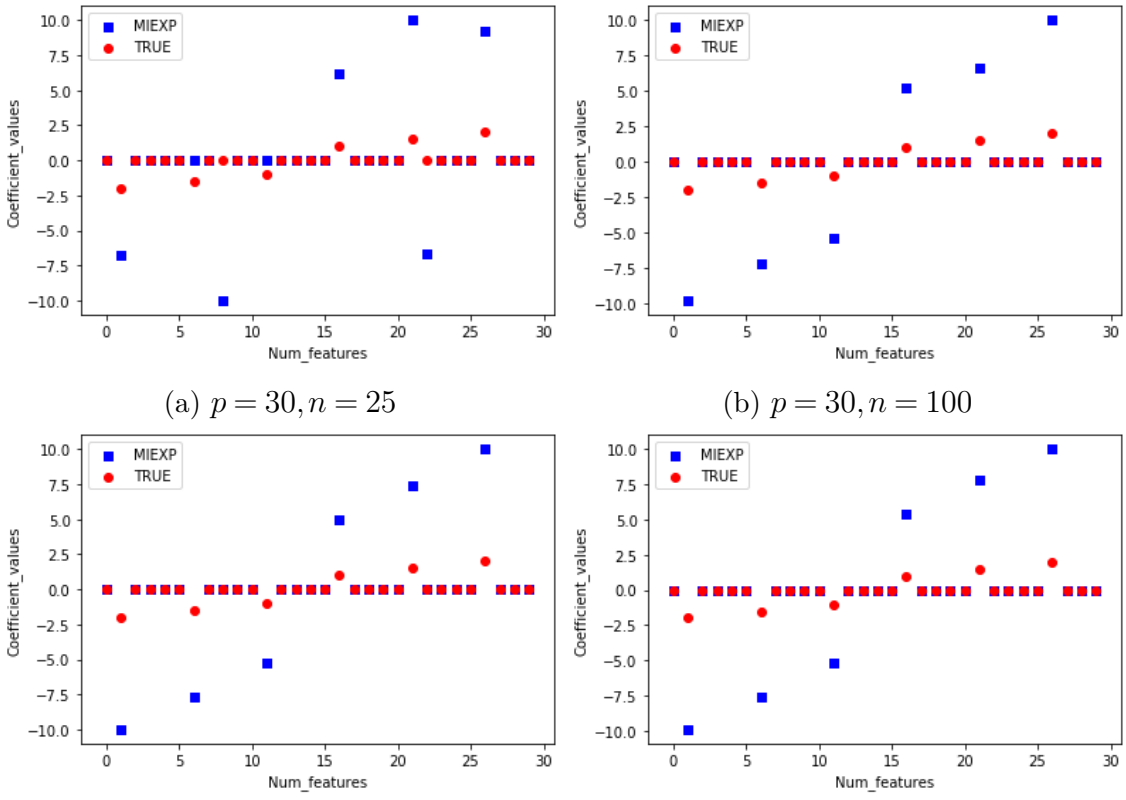
In our implementation, we will first evaluate the performance of the models over a group of toy examples and scenarios provided in [39]. The mentioned study, introduced a Bayesian methodology for feature subset selection in logistic regression. Throughout these examples p and n shows the number of features and data points (observations), respectively.

Example 1: In this case, we fix the number of features as $p = 30$ and the number of observations alters as $n = 25, 100, 200, 500$, respectively. Each dataset is drawn from the multivariate normal distribution $\mathcal{N} \sim (0, I_p)$. Note that in this case, using

identity matrix as covariance matrix leads to generation of totally independent features. Finally, we assume the coefficients of all features to be zero except those with indices 1, 6, 11, 15, 21, and 26 where they have the value equal to -2.0, -1.5, -1.0, 1.0, 1.5, 2.0, respectively.

Example 2: In this case, we include more features and set $p = 50$ and consider cases with $n = 35, 50, 100, 200$. Each dataset is drawn from the multivariate normal distribution $\mathcal{N} \sim (0, \sum)$ with $\sum_{ij} = 0.8^{|i-j|}$. Here, there are correlations between pair of features. Finally, we set the coefficients of all features to be zero except those with indices 1, 11, 21, 31, and 41, which are all equal to 0.8.

Considering Example 1 and Example 2, we will evaluate the performance of our model both in terms of selecting true features (interpretablity) and prediction accuracy. Due to the structure of these examples, we have prior information about the number of true features. Hence, we can simply use model (2.16). For these examples, setting big-M value to 10 is more than enough considering the magnitude of the coefficients. Moreover, the termination criteria is fixed to 60 seconds for these examples.



(a) $p = 30, n = 25$ (b) $p = 30, n = 100$

(c) $p = 30, n = 200$ (d) $p = 30, n = 500$

Figure 3.2 Performance of the model in selecting the true coefficients for Example 1.

Figure 3.2 reveals that for cases 3.2b-3.2d, the model selects all the true coefficients, however in Figure (3.2a), it includes 2 wrong features. As we can see, the number of observation in this case is relatively small which affects the performance of the model.

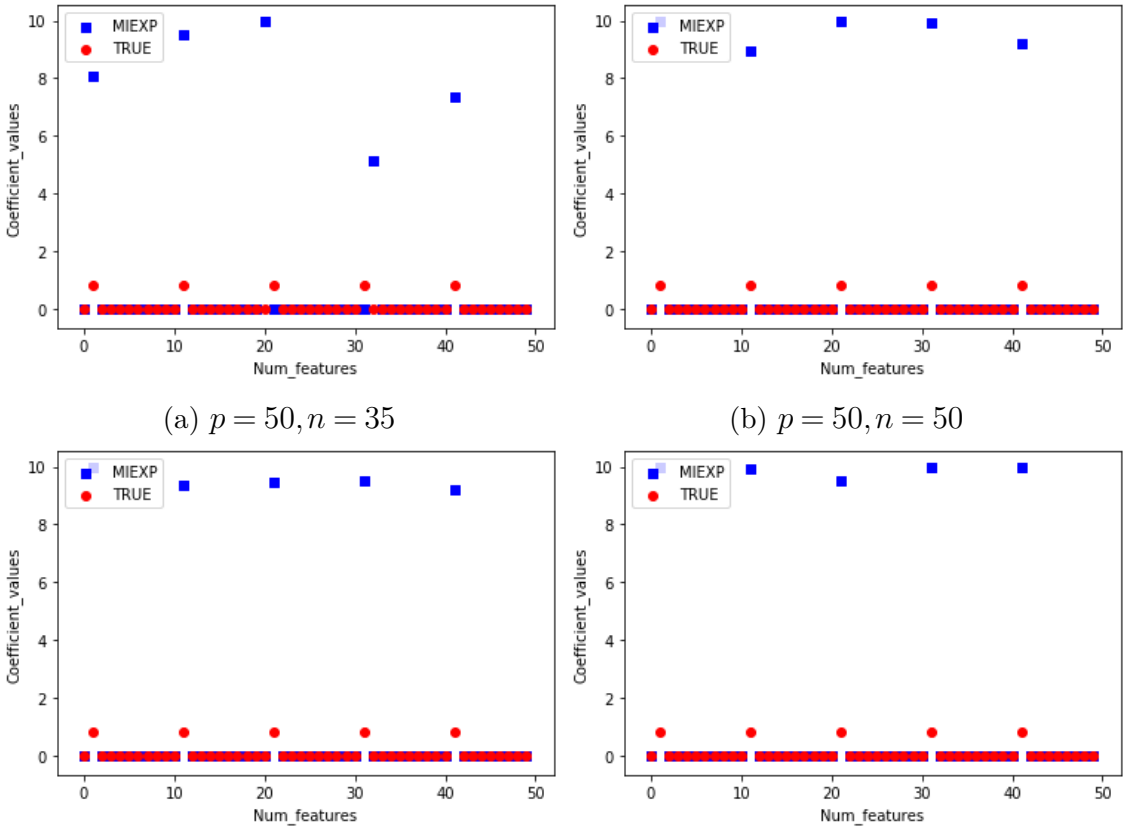
Another interesting inference from this figure is related to the value of coefficients. As discussed in Section 3.1, if there exist relatively sufficient observations, the model selects the value of the largest coefficient equal to the big-M value and scales other coefficients accordingly. We can observe that when the number of observations is large enough, scaling of coefficients by the model might be more accurate (this pattern can be detected by comparing Figures 3.2b and 3.2c). As an illustration, in Figure 3.2d, the values of coefficients found by the MIEXP model are almost 5 times greater than the true coefficient values, because in this case the ratio of big-M value to the largest true coefficient value, which are ten and two, respectively, is equal to five. These figures are provided to evaluate the performance the model in terms of intrepretability which is the capability of selecting the true coefficients. In Table 3.1, we will evaluate the performance of the model in terms of prediction accuracy.

n	Acc.	F-score	Harmonic mean	NPV	Opt gap.
25	0.8166	0.9333	0.7873	0.4000	0
100	0.9900	0.9857	0.8847	1.0000	0
200	0.9850	0.9873	0.8827	1.0000	0
500	0.9900	0.9901	0.8848	0.9917	0

Table 3.1 Performance of the model in terms of binary metrics for Example 1.

As we can see from the Table 3.1, the model performs nearly perfectly in terms of prediction as well. We can see the significant improvement in the performance of the model once the number of observations increases from 25 to 100 and it has almost the same performance when the number of observations are 100, 200, and 500, respectively.

Figure 3.3 and Table 3.2 show the performance of the model for the second example. The performance of the model is similar as in Example 1 in terms of both intrepretability and prediction, hence, the same conclusions as Example 1 can be drawn for Example 2 as well.



(a) $p = 50, n = 35$ (b) $p = 50, n = 50$

(c) $p = 50, n = 100$ (d) $p = 50, n = 200$

Figure 3.3 Performance of the model in selecting the true coefficients for Example.

n	Acc.	F-score	Harmonic mean	NPV	Opt gap.
35	0.7333	0.9314	0.7740	0.7166	0.0660
50	0.9200	0.8333	0.9045	0.9750	0.0000
100	0.9800	0.9818	0.9378	0.9600	0.0000
200	0.9950	0.9941	0.9450	1.0000	0.0000

Table 3.2 Performance of the model in terms of binary metrics for Example 2.

In the next step, we will consider four different scenarios to evaluate the performance of the model for differently structured datasets. We will briefly explain each scenario.

Scenario 1:

For this scenario, we assume $p = 100$ and $n = 50, 80, 110$. The value of coefficients with indices 1, 36, and 71 are set to be 2.5 while the others are zero. Each dataset is drawn from the multivariate normal distribution $\mathcal{N} \sim (0, \sum)$ with $\sum_{ij} = 0.94^{|i-j|}$.

Scenario 2:

For the purpose of evaluating the performance of the models in a less sparse situation,

we increase the number of true coefficients in this scenario. Hence, the coefficients with the indices $1, 16, 31, \ldots, 91$ will have the value of 2.5 while others are set to be zero. Moreover, we consider the multivariate normal distribution $\mathcal{N} \sim (0, \sum)$ with $\sum_{ij} = 0.8^{|i-j|}$ for datasets. Remaining parameters are the same as Scenario 1.

Scenario 3:

In this scenario, we will change the magnitude of the value of the coefficients by assuming that the coefficients with the indices $1, 16, 31, \ldots, 91$ will have the value of 0.6 while others are set to be zero. The rest of the structure is identical to the Scenario 2.

Scenario 4:

In this scenario, we will consider the combination of previous scenarios for high dimensional cases. Hence, we assume the value of p to be 300 while n takes the values 200, 300, and 400, respectively. There exist 15 true coefficients in this scenario where the value of coefficients with the indices $1, 21, 41, \ldots, 181$ are assumed to be 0.6 and those with indices $201, 221, \ldots, 281$ take the value 2. The value of remaining coefficients are set to be zero. Moreover, each data set is drawn from the multivariate normal distribution $\mathcal{N} \sim (0, \sum)$ with $\sum_{ij} = 0.8^{|i-j|}$.

We will use model (2.16) for these scenarios as we know the number of true coefficients in advance. Table 3.3 shows the performance of the model for each scenario in terms of prediction accuracy.

n	Acc.	F-score	Harmonic mean	NPV	Opt. gap
50	0.9800	0.9700	0.9739	1.0000	0.0000
80	0.9750	0.9523	0.9718	1.0000	0.0000
110	1.0000	1.0000	0.9847	1.0000	0.0000

(a) Scenario 1

n	Acc.	F-score	Harmonic mean	NPV	Opt. gap
50	0.8200	0.8547	0.8490	0.7000	0.6678
80	0.8625	0.8290	0.8917	0.9266	0.9169
110	1.0000	0.9637	1.0000	1.0000	0.3368

(b) Scenario 2

n	Acc.	F-score	Harmonic mean	NPV	Opt. gap
50	0.8200	0.8547	0.8490	0.7000	0.6802
80	0.8625	0.8290	0.8917	0.9266	0.9183
110	1.0000	1.0000	0.9637	1.0000	0.3589

(c) Scenario 3

n	Acc.	F-score	Harmonic mean	NPV	Opt. gap
200	0.8250	0.8206	0.8785	0.8605	0.9999
300	0.9066	0.9276	0.9204	0.9204	0.9998
400	0.9375	0.9413	0.9434	0.9521	0.9991

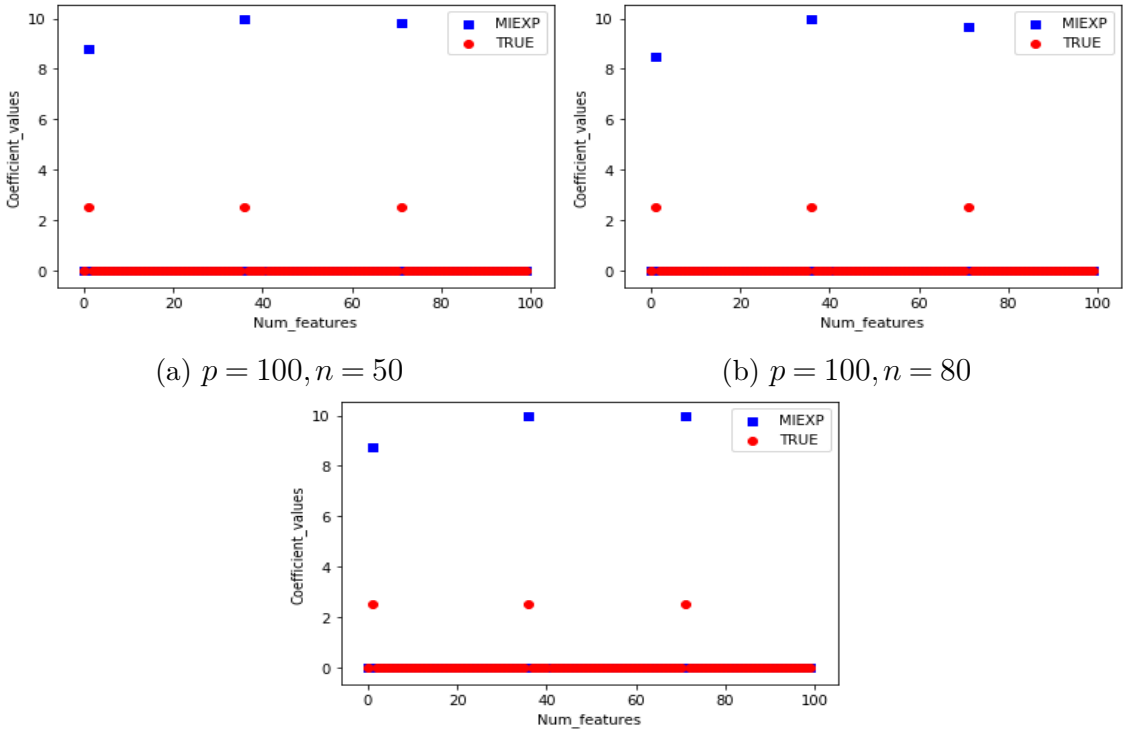
(d) Scenario 4

Table 3.3 Performance of model (2.16) for the scenarios in terms of prediction.

As we can see from Table 3.3, by increasing the number of true coefficients in Scenario 2 compared to Scenario 1, the model fails to obtain the desired optimality gap and the performance of the model decreases in terms of prediction accuracy. However, when the number of observations is high (in these cases 110), the difference between performance of the model over these datasets is insignificant. Comparing

Scenario 2 with Scenario 3, we can observe that decreasing the magnitude of the coefficients (weak signal strength) has not affected the performance of the model. Considering Scenario 4, we can see that in the case of high dimensionality, the model completely fails to obtain the desired optimality gap, however the prediction performance of the model for this scenario is still quite satisfactory. This may stem from occurrence of the same pattern we discussed in Figure 3.1. Overall, our model reveals a significantly better performance for all the toy examples and scenarios in comparison to the methods discussed in [39].

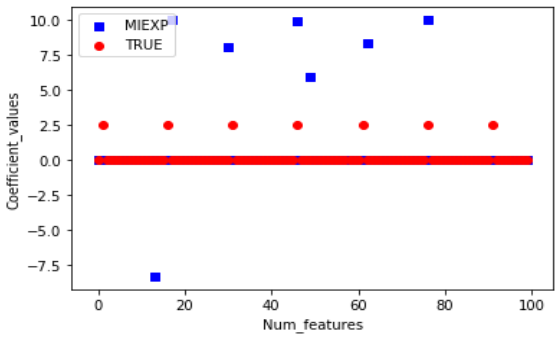
In the next step, we will discuss the performance of our method in terms of selecting the true coefficients for each of the scenarios. Figures 3.4-3.7, represents the mentioned performance for the Scenarios 1 to 4, respectively.



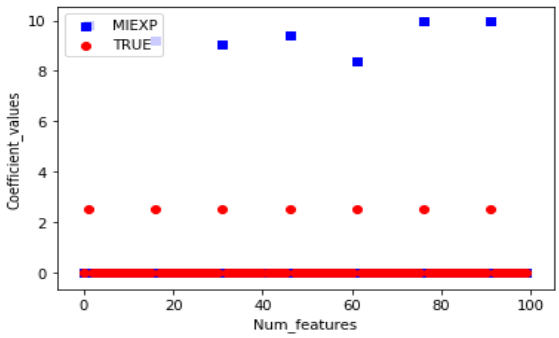
(a) $p = 100, n = 50$ (b) $p = 100, n = 80$

(c) $p = 100, n = 110$

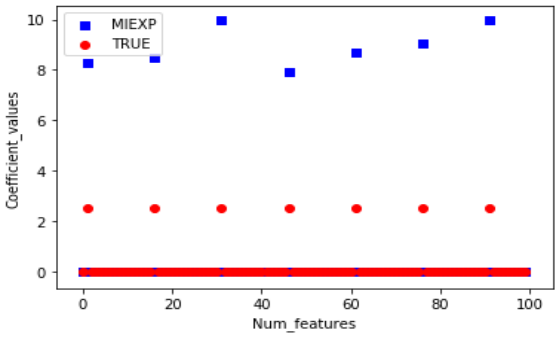
Figure 3.4 Performance of the model in terms of selecting the true coefficients for Scenario 1.



(a) $p = 100, n = 50$

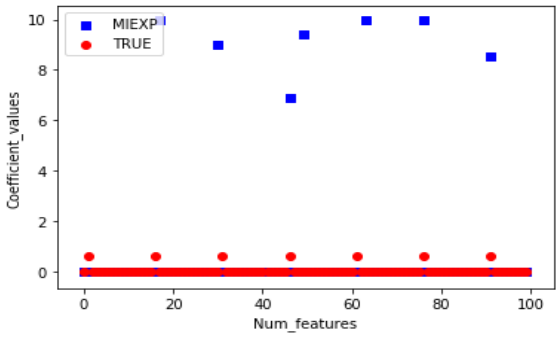


(b) $p = 100, n = 80$

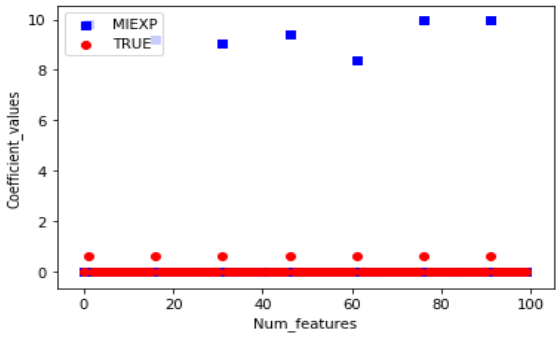


(c) $p = 100, n = 110$

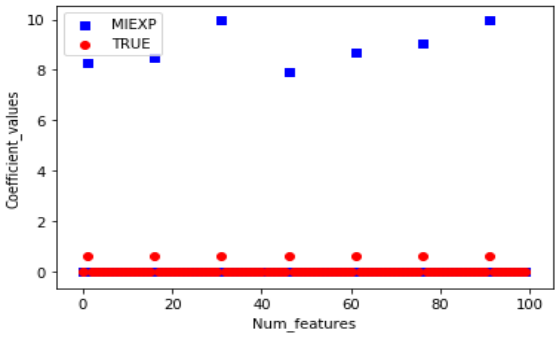
Figure 3.5 Performance of the model in terms of selecting the true coefficients for Scenario 2.



(a) $p = 100, n = 50$

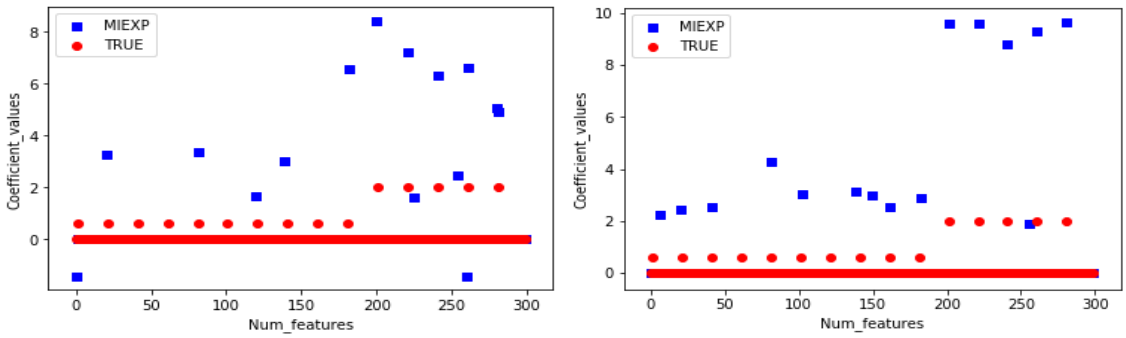


(b) $p = 100, n = 80$

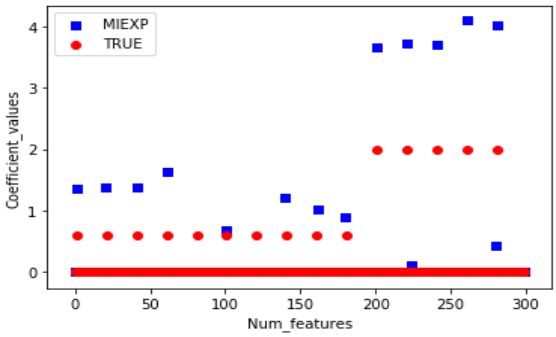


(c) $p = 100, n = 110$

Figure 3.6 Performance of the model in terms of selecting the true coefficients for Scenario 3.



(a) $p = 300, n = 200$ (b) $p = 300, n = 300$



(c) $p = 300, n = 400$

Figure 3.7 Performance of the model in terms of selecting the true coefficients for Scenario 4.

Figure 3.4, shows that when the number of true coefficients is small, the model performs very well for all the cases with different observation sizes. However, when the number of true predictors increases, the model fails to find all the true coefficients, especially when the sample size is relatively small (see Figure 3.5). For Scenario 3, the sample size plays the identical role as in Scenario 2, meaning that when the magnitude of the coefficients is small, the model requires more observations to perform well (see Figure 3.3c). Finally, Figure 3.3d represents that the weak signal strength (decreasing the magnitude of coefficients) affects the performance of the model in terms of finding the true features. Note that in this scenario the last five coefficients have been always chosen correctly due to their strong signal strength.

We presented and discussed the performance of our model over a set of toy examples. In the next part, we will consider a group of benchmark datasets from the literature.

3.1.2 Experimental Analysis for Benchmark Datasets

For sparse logistic regression model, we used a group of datasets from UCI Machine Learning Repository [71]. Overall, we will divide these datasets into two main groups as *easy* ($p < n$) and *hard* ($n < p$) datasets. The details of these two groups are shown in Table 3.4 and Table 3.5, respectively.

Dataset	n	p
Breast	699	10
Heart	270	14
Parkinson	195	23
Divorce	170	55
Sonar	208	60

Table 3.4 Easy datasets: $p < n$. p and n are the number of features and observations, respectively.

Dataset	n	p
LSVT	126	310
Colon	62	386
Leukemia	72	985

Table 3.5 Hard datasets: $p > n$. p and n are the number of features and observations, respectively.

We compare the performance of several models. Two of our models rely on continuous conic programs known as *Logistic Regression* (model(2.15)) and also its $\ell_2 - regularization$ version and abbreviated as *Regularized Log.Reg.* For all these datasets, we will consider three different versions of our model for the implementation purpose. The first model is constructed by adding ℓ_2-regularization term for the coefficients in (2.19) and (2.20) models and is named as *Regularized MIEXP.* The second version considers (2.19) and (2.20) models called as *MIEXP.* Finally in the last version, to prevent the model from associating very small values to the coefficients, we add a constraint to both (2.19) and (2.20) models which certifies the value of each coefficient to be greater than 0.1 if the model selects that coefficient. This version is denoted as *MIEXP with Lower bound For Coefficients.* Moreover, we repeat the experiments for different values of big-M which are 10, 100, and 1000, and also for different values of termination criteria which are 60, 300, and 600 seconds. We note that, for the datasets in which the desired optimality gap is obtained in 60 seconds, we do not repeat the experiment for the greater values of termination criteria.

Method	Time (sec)	Model type	Big M	Acc.	F-score	Harmonic mean	NPV	sparsity	Opt gap	Ave. of features
Regularized Log.Reg	60	NA	NA	**0.8444**	**0.8199**	0.0000	**0.8986**	0.0000	0	14.0
Logistic Regression	60		NA	0.8407	0.8170	0.0000	0.8909	0.0000	0	14.0
Regularized MIEXP	60	AIC	10^1	0.7778	0.7693	0.4044	0.7792	0.3286	0	9.4
			10^2	0.7778	0.7693	0.4044	0.7792	0.3286	0	9.4
			10^3	0.7778	0.7693	0.3961	0.7792	0.3214	0	9.5
		BIC	10^1	0.8222	0.7942	**0.6709**	0.8625	**0.5714**	0	6.0
			10^2	0.8222	0.7942	**0.6709**	0.8625	**0.5714**	0	6.0
			10^3	0.8222	0.7942	0.6600	0.8625	0.5571	0	6.1
MIEXP	60	AIC	10^1	0.8074	0.7801	0.3602	0.8518	0.2429	0	10.6
			10^2	0.8074	0.7801	0.3602	0.8518	0.2429	0	10.6
			10^3	0.8074	0.7801	0.3602	0.8518	0.2429	0	10.6
		BIC	10^1	0.8222	0.7942	**0.6709**	0.8625	**0.5714**	0	6.0
			10^2	0.8222	0.7942	**0.6709**	0.8625	**0.5714**	0	6.0
			10^3	0.8222	0.7942	0.6652	0.8625	0.5643	0	6.1
MIEXP with Lower bound For Coefficients	60	AIC	10^1	0.8074	0.7801	0.3602	0.8518	0.2429	0	10.6
			10^2	0.8074	0.7801	0.3602	0.8518	0.2429	0	10.6
			10^3	0.8074	0.7801	0.3602	0.8518	0.2429	0	10.6
		BIC	10^1	0.8222	0.7942	**0.6709**	0.8625	**0.5714**	0	6.0
			10^2	0.8222	0.7942	**0.6709**	0.8625	**0.5714**	0	6.0
			10^3	0.8222	0.7942	0.6652	0.8625	0.5643	0	6.1

Table 3.6 Performance of the models in terms of prediction for the Heart dataset.

As can be observed from Table 3.6, the regularized logistic regression has the best prediction performance for the Heart dataset. However, in terms of intrepretability, it reveals a highly poor performance by adding all the features to the model. On the other hand, the BIC model compromises the intrepretability and prediction accuracy which results in obtaining the highest harmonic mean value. In fact, although it constructs a model by six predictors, its performance is very close to the regularized logistic regression in terms of prediction accuracy. The impact of the regularization can be observed in the AIC models. In this case, the regularized versions constructs the model with less number of predictors in average compared to MIEXP and MIEXP model with lower bound. Additionally, the performances of the MIEXP and the MIEXP model with lower bound are identical for this dataset, meaning that MIEXP model already associates values bigger than 0.1 to the coefficients.

Method	Time (sec)	Model type	Big M	Acc.	F-score	Harmonic mean	NPV	sparsity	Opt gap	Ave. of features
Regularized Log.Reg	60	NA	NA	**0.9628**	**0.9446**	0.0000	**0.9746**	0.0000	0	10.0
Logistic Regression	60		NA	**0.9628**	**0.9446**	0.0000	**0.9746**	0.0000	0	10.0
Regularized MIEXP	60	AIC	10^1	0.9600	0.9405	0.4066	**0.9746**	0.2600	0	7.4
			10^2	0.9600	0.9405	0.4066	**0.9746**	0.2600	0	7.4
			10^3	0.9600	0.9405	0.4066	**0.9746**	0.2600	0	7.4
		BIC	10^1	0.9528	0.9292	**0.6002**	0.9723	**0.4400**	0	5.6
			10^2	0.9528	0.9292	**0.6002**	0.9723	**0.4400**	0	5.6
			10^3	0.9528	0.9292	**0.6002**	0.9723	**0.4400**	0	5.6
MIEXP	60	AIC	10^1	0.9600	0.9405	0.4066	**0.9746**	0.2600	0	7.4
			10^2	0.9600	0.9405	0.4066	**0.9746**	0.2600	0	7.4
			10^3	0.9600	0.9405	0.3939	**0.9746**	0.2500	0	7.5
		BIC	10^1	0.9528	0.9292	**0.6002**	0.9723	**0.4400**	0	5.6
			10^2	0.9528	0.9292	**0.6002**	0.9723	**0.4400**	0	5.6
			10^3	0.9528	0.9292	**0.6002**	0.9723	**0.4400**	0	5.6
MIEXP with Lower bound For Coefficients	60	AIC	10^1	0.9600	0.9405	0.4066	**0.9746**	0.2600	0	7.4
			10^2	0.9600	0.9405	0.4066	**0.9746**	0.2600	0	7.4
			10^3	0.9600	0.9405	0.3939	**0.9746**	0.2500	0	7.5
		BIC	10^1	0.9528	0.9292	**0.6002**	0.9723	**0.4400**	0	5.6
			10^2	0.9528	0.9292	**0.6002**	0.9723	**0.4400**	0	5.6
			10^3	0.9528	0.9292	**0.6002**	0.9723	**0.4400**	0	5.6

Table 3.7 Performance of the models in terms of prediction for the Breast dataset.

The performance of the models for the Breast dataset are almost the same as the Heart dataset with some minor differences. In this case, regularization does not affect the average number of selected features. Moreover, the AIC model has almost the same performance as the regularized logistic regression and logistic regression versions, due to the slight difference between the number of selected features. The BIC method has the best performance in terms of harmonic mean for the Breast dataset as well.

Method	Time (sec)	Model type	Big M	Acc.	F-score	Harmonic mean	NPV	sparsity	Opt gap	Ave. of features
Regularized Log.Reg	60	NA	NA	0.8411	0.8946	0.0083	0.5792	0.0043	0	22.9
Logistic Regression	60		NA	0.8468	0.8973	0.0249	0.6592	0.0130	0	22.7
Regularized MIEXP	60	AIC	10^1	0.8458	0.8972	0.8068	0.6333	0.7739	0	5.2
			10^2	0.8405	0.8936	0.8153	0.6333	0.7957	0	4.7
			10^3	0.8408	0.8936	0.7648	0.6333	0.7043	0	6.8
		BIC	10^1	0.8505	0.9018	**0.8562**	0.6167	**0.8652**	0	3.1
			10^2	0.8505	0.9008	0.8560	0.6167	**0.8652**	0	3.1
			10^3	0.8205	0.8807	0.7732	0.5767	0.7348	0	6.1
MIEXP	60	AIC	10^1	0.8458	0.8972	0.8068	0.6333	0.7739	0	5.2
			10^2	0.8508	0.9008	0.8114	0.6333	0.7783	0	5.1
			10^3	0.8621	0.9075	0.6503	0.6592	0.5261	0	10.9
		BIC	10^1	0.8505	0.9018	**0.8562**	0.6167	**0.8652**	0	3.1
			10^2	0.8455	0.8974	0.8452	0.6167	0.8478	0	3.5
			10^3	0.8305	0.8882	0.7681	0.5767	0.7174	0	6.5
MIEXP with Lower bound For Coefficients	60	AIC	10^1	0.8508	0.9008	0.8164	0.6333	0.7870	0	4.9
			10^2	0.8508	0.9008	0.8185	0.6333	0.7913	0	4.8
			10^3	**0.8721**	**0.9134**	0.6523	**0.6992**	0.5261	0	10.9
		BIC	10^1	0.8505	0.9018	**0.8562**	0.6167	**0.8652**	0	3.1
			10^2	0.8455	0.8974	0.8452	0.6167	0.8478	0	3.5
			10^3	0.8305	0.8882	0.7681	0.5767	0.7174	0	6.5

Table 3.8 Performance of the models in terms of prediction for the Parkinson dataset.

In Parkinson dataset, both BIC and AIC outperform the logistic regression models in terms of accuracy and F-score in most of the cases, although they include less predictors to the model. The same as previous datsets, BIC has the highest harmonic mean. For this dataset, we can observe the difference between the MIEXP and MIEXP with lower bound for coefficients for the AIC models. The MIEXP model with lower bound generates a more sparse model compared to MIEXP. The reason is that when the MIEXP constructs the model for this case, it associates values less than 0.1 to the coefficients.

Generally, we can conclude that for these datasets the BIC model shows the best performance in most cases and regularization caused to generate better models in terms of sparsity. We also note that the all the MIEXP models are solved to optimality for these three datasets.

Method	Time (sec)	Model type	Big M	Acc.	F-score	Harmonic mean	NPV	sparsity	Opt gap	Ave. of features
Regularized Log.Reg	60	NA	NA	**0.9765**	**0.9766**	0.0000	**1.0000**	0.0000	0.0000	55.0
Logistic Regression			NA	**0.9765**	**0.9766**	0.0000	**1.0000**	0.0000	0.0000	55.0
Regularized MIEXP	60	AIC	10^1	0.9588	0.9569	0.9431	0.9684	0.9291	0.1446	3.9
			10^2	0.9706	0.9701	0.9071	0.9709	0.8527	0.6842	8.1
			10^3	0.9471	0.9392	0.8327	0.9519	0.7455	0.8336	14.0
		BIC	10^1	0.9706	0.9699	0.9565	0.9900	0.9436	0.0438	3.1
			10^2	0.9706	0.9707	0.9244	0.9709	0.8836	0.6310	6.4
			10^3	0.9706	0.9688	0.8797	0.9784	0.8055	0.7916	10.7
	300	AIC	10^1	0.9706	0.9699	0.9516	0.9900	0.9345	0.0000	3.6
			10^2	0.9706	0.9699	0.9506	0.9900	0.9327	0.1793	3.7
			10^3	0.9706	0.9699	0.9380	0.9900	0.9091	0.4228	5.0
		BIC	10^1	0.9706	0.9699	0.9575	0.9900	**0.9455**	0.0000	3.0
			10^2	0.9706	0.9699	0.9565	0.9900	0.9436	0.1186	3.1
			10^3	0.9588	0.9545	0.9403	0.9718	0.9236	0.3393	4.2
	600	AIC	10^1	0.9706	0.9699	0.9516	0.9900	0.9345	0.0000	3.6
			10^2	0.9706	0.9699	0.9516	0.9900	0.9345	0.0663	3.6
			10^3	0.9588	0.9579	0.9441	0.9642	0.9309	0.2276	3.8
		BIC	10^1	0.9706	0.9699	0.9575	0.9900	**0.9455**	0.0000	3.0
			10^2	0.9706	0.9699	0.9575	0.9900	**0.9455**	0.0062	3.0
			10^3	0.9706	0.9699	0.9507	0.9900	0.9327	0.1591	3.7
MIEXP	60	AIC	10^1	0.9471	0.9440	0.9337	0.9559	0.9218	0.2028	4.3
			10^2	0.9588	0.9564	0.9424	0.9525	0.9273	0.4561	4.0
			10^3	0.9647	0.9597	0.9423	0.9718	0.9218	0.6492	4.3
		BIC	10^1	0.9647	0.9655	0.9498	0.9733	0.9364	0.1270	3.5
			10^2	0.9529	0.9474	0.9395	0.9468	0.9273	0.4635	4.0
			10^3	0.9529	0.9510	0.9387	0.9352	0.9255	0.6519	4.1
	300	AIC	10^1	0.9706	0.9699	0.9516	0.9900	0.9345	0.0000	3.6
			10^2	0.9647	0.9651	0.9491	0.9775	0.9345	0.1273	3.6
			10^3	0.9588	0.9578	0.9463	0.9517	0.9345	0.2915	3.6
		BIC	10^1	0.9706	0.9699	0.9575	0.9900	0.9455	0.0000	3.0
			10^2	0.9706	0.9674	**0.9576**	0.9809	0.9455	0.1008	3.0
			10^3	0.9706	0.9699	0.9537	0.9775	0.9382	0.2884	3.4
	600	AIC	10^1	0.9706	0.9699	0.9516	0.9900	0.9345	0.0000	3.6
			10^2	0.9647	0.9651	0.9491	0.9775	0.9345	0.0490	3.6
			10^3	0.9647	0.9655	0.9491	0.9608	0.9345	0.2032	3.6
		BIC	10^1	0.9706	0.9699	0.9575	0.9900	**0.9455**	0.0000	3.0
			10^2	0.9647	0.9622	0.9548	0.9684	**0.9455**	0.0112	3.0
			10^3	0.9706	0.9699	**0.9576**	0.9775	0.9455	0.1654	3.0
MIEXP with Lower bound on Coefficient	60	AIC	10^1	0.9647	0.9655	0.9441	0.9733	0.9255	0.2201	4.1
			10^2	0.9647	0.9651	0.9442	0.9775	0.9255	0.4689	4.1
			10^3	0.9588	0.9540	0.9395	0.9559	0.9218	0.6782	4.3
		BIC	10^1	0.9647	0.9651	0.9497	0.9900	0.9364	0.1324	3.5
			10^2	0.9706	0.9674	0.9499	0.9809	0.9309	0.4415	3.8
			10^3	0.9529	0.9467	0.9368	0.9268	0.9218	0.6883	4.3
	300	AIC	10^1	0.9706	0.9699	0.9516	0.9900	0.9345	0.0000	3.6
			10^2	0.9706	0.9704	0.9519	0.9900	0.9345	0.1383	3.6
			10^3	0.9706	0.9704	0.9489	0.9900	0.9291	0.3412	3.9
		BIC	10^1	0.9706	0.9699	0.9575	0.9900	0.9455	0.0000	3.0
			10^2	0.9647	0.9622	0.9548	0.9684	0.9455	0.1122	3.0
			10^3	0.9588	0.9531	0.9443	0.9601	0.9309	0.3565	3.8
	600	AIC	10^1	0.9706	0.9699	0.9516	0.9900	0.9345	0.0000	3.6
			10^2	0.9706	0.9704	0.9519	0.9900	0.9345	0.0583	3.6
			10^3	0.9765	0.9751	0.9547	0.9900	0.9345	0.2200	3.6
		BIC	10^1	0.9706	0.9699	0.9575	0.9900	**0.9455**	0.0000	3.0
			10^2	0.9706	0.9674	**0.9576**	0.9809	**0.9455**	0.0201	3.0
			10^3	0.9647	0.9622	0.9548	0.9684	**0.9455**	0.0505	3.0

Table 3.9 Performance of the models in terms of prediction for the Divorce dataset.

Method	Time (sec)	Model type	Big M	Acc.	F-score	Harmonic mean	NPV	sparsity	Opt gap	Ave. of features
Regularized Log.Reg	60	NA	NA	0.7505	0.7219	0.0000	0.7730	0.0000	0.0000	61.0
Logistic Regression			NA	0.7410	0.7095	0.0000	0.7798	0.0000	0.0000	61.0
Regularized MIEXP	60	AIC	10^1	0.7645	0.7449	0.7127	0.7852	0.6705	0.0564	20.1
			10^2	0.7500	0.7152	0.4866	0.7875	0.3738	0.3790	38.2
			10^3	0.7169	0.6875	0.4347	0.7331	0.3246	0.4713	41.2
		BIC	10^1	0.7119	0.6789	0.7731	0.7432	0.8541	0.0685	8.9
			10^2	0.7029	0.6539	0.7296	0.7717	0.7738	0.4286	13.8
			10^3	0.7555	0.7292	0.7021	0.7918	0.6721	0.5781	20.0
	300	AIC	10^1	0.7595	0.7406	0.7261	0.7780	0.7016	0.0200	18.2
			10^2	0.7357	0.7087	0.6329	0.7664	0.5639	0.2612	26.6
			10^3	0.7357	0.6981	0.6414	0.7943	0.5787	0.3381	25.7
		BIC	10^1	0.7217	0.7070	0.7907	0.7128	0.8770	0.0115	7.5
			10^2	0.7124	0.6751	0.7673	0.7577	0.8410	0.3148	9.7
			10^3	0.7219	0.6968	**0.8014**	0.7444	**0.9033**	0.4284	5.9
	600	AIC	10^1	0.7595	0.7406	0.7261	0.7780	0.7016	0.0106	18.2
			10^2	0.7314	0.6976	0.6462	0.7946	0.5984	0.2245	24.5
			10^3	0.7267	0.6922	0.6349	0.7803	0.5770	0.2955	25.8
		BIC	10^1	0.7217	0.7070	0.7907	0.7128	0.8770	0.0026	7.5
			10^2	0.6729	0.6420	0.7436	0.6917	0.8361	0.2577	10.0
			10^3	0.7033	0.6708	0.7815	0.7353	0.8918	0.3762	6.6
MIEXP	60	AIC	10^1	**0.7833**	**0.7659**	0.7320	**0.8046**	0.6902	0.0464	18.9
			10^2	0.7314	0.6994	0.5493	0.7640	0.4541	0.3757	33.3
			10^3	0.7507	0.7310	0.4293	0.7762	0.3049	0.6879	42.4
		BIC	10^1	0.7026	0.6748	0.7695	0.7323	0.8574	0.0647	8.7
			10^2	0.7021	0.6798	0.6679	0.7092	0.6672	0.4445	20.3
			10^3	0.6883	0.6422	0.5837	0.7473	0.6082	0.7279	23.9
	300	AIC	10^1	0.7595	0.7395	0.7235	0.7780	0.6951	0.0153	18.6
			10^2	0.7310	0.6995	0.6527	0.7576	0.5934	0.2289	24.8
			10^3	0.7260	0.6943	0.4987	0.7484	0.3852	0.5409	37.5
		BIC	10^1	0.7169	0.7016	0.7877	0.7128	0.8770	0.0109	7.5
			10^2	0.7405	0.7184	0.7462	0.7514	0.7623	0.3335	14.5
			10^3	0.7200	0.6970	0.6993	0.7518	0.7311	0.6105	16.4
	600	AIC	10^1	0.7595	0.7395	0.7235	0.7780	0.6951	0.0070	18.6
			10^2	0.7214	0.6875	0.6465	0.7465	0.5902	0.2058	25.0
			10^3	0.7212	0.6941	0.5231	0.7393	0.4148	0.4842	35.7
		BIC	10^1	0.7169	0.7016	0.7877	0.7128	0.8770	0.0032	7.5
			10^2	0.7162	0.6887	0.7499	0.7293	0.7984	0.2917	12.3
			10^3	0.7400	0.7142	0.7489	0.7752	0.7967	0.5777	12.4
MIEXP with Lower bound on Coefficient	60	AIC	10^1	0.7595	0.7326	0.7138	0.7964	0.6770	0.3566	19.7
			10^2	0.7550	0.7173	0.4804	0.7946	0.3590	0.6855	39.1
			10^3	0.7214	0.6803	0.3823	0.7805	0.2639	0.8417	44.9
		BIC	10^1	0.7176	0.6892	0.7791	0.7511	0.8607	0.4621	8.5
			10^2	0.6736	0.6526	0.6848	0.6731	0.7311	0.7285	16.4
			10^3	0.6740	0.6302	0.5118	0.7319	0.4803	0.8015	31.7
	300	AIC	10^1	0.7595	0.7395	0.7235	0.7780	0.6951	0.0168	18.6
			10^2	0.7360	0.7048	0.6328	0.7647	0.5623	0.2478	26.7
			10^3	0.7307	0.7010	0.5026	0.7504	0.3852	0.5353	37.5
		BIC	10^1	0.7169	0.7016	0.7877	0.7128	0.8770	0.0114	7.5
			10^2	0.7114	0.6867	0.7251	0.7231	0.7508	0.3488	15.2
			10^3	0.7119	0.6967	0.6551	0.7090	0.6639	0.6256	20.5
	600	AIC	10^1	0.7595	0.7395	0.7235	0.7780	0.6951	0.0070	18.6
			10^2	0.7360	0.7023	0.6494	0.7690	0.5852	0.2110	25.3
			10^3	0.7402	0.7167	0.5353	0.7493	0.4230	0.4856	35.2
		BIC	10^1	0.7169	0.7016	0.7877	0.7128	0.8770	0.0035	7.5
			10^2	0.6926	0.6503	0.7267	0.7357	0.7738	0.3032	13.8
			10^3	0.6929	0.6796	0.7063	0.6774	0.7770	0.5847	13.6

Table 3.10 Performance of the models in terms of prediction for the Sonar dataset.

Although Divorce and Sonar datasets are categorized as easy datasets based on our definition, in most cases the high number of predictors and the insufficient number of observations results in larger optimality gap. To overcome this issue, we set three different values for termination criteria which are 60, 300, and 600 seconds. By increasing the run time, the optimality gap decreases for all cases, even reaching to zero for some cases. We can observe that the higher the value of big-M is, the greater the optimality gap will be. This clarifies the importance of choosing the smallest possible big-M. Although there are large optimality gaps for Divorce and Sonar datasets, almost the same pattern as in first three datasets can be detected. BIC tends to construct more sparse models while keeping accuracy as high as possible. In Divorce dataset, the performance of our method in terms of intrepretability is significantly better that logistic regression models and very close to their performance in terms of prediction accuracy. Quite interestingly, our method, outperforms logistic regression in both terms of sparsity and prediction accuracy in the Sonar dataset.

Method	Time (sec)	Model type	Big M	Acc.	F-score	Harmonic mean	NPV	sparsity	Opt gap	Ave. of features
Regularized Log.reg	60	NA	NA	0.7077	0.6565	0.0051	0.6086	0.0026	0.0000	309.2
Logistic Regression			NA	0.5167	0.5156	0.0251	0.3340	0.0129	0.0000	306.0
Regularized MIEXP	60	AIC	10^1	0.6673	0.6012	0.7750	0.5896	0.9345	0.5771	20.3
			10^2	0.6833	0.6381	0.7610	0.5689	0.8735	0.9341	39.2
			10^3	0.4641	0.6216	0.6169	0.3316	**0.9865**	**0.9891**	4.2
		BIC	10^1	0.6763	0.6229	0.7808	0.5992	0.9374	0.6248	19.4
			10^2	0.7006	0.6432	0.7957	0.6376	0.9348	0.9235	20.2
			10^3	0.4237	0.5307	0.5514	0.1681	0.9787	0.9881	6.6
	300	AIC	10^1	0.6897	0.6333	0.7891	0.5856	0.9352	0.5156	20.1
			10^2	0.6423	0.5948	0.7380	0.5446	0.8784	0.9051	37.7
			10^3	0.7026	0.6315	0.7767	0.6660	0.9097	0.9703	28.0
		BIC	10^1	0.6429	0.6035	0.7599	0.5231	0.9397	0.5569	18.7
			10^2	0.7558	0.6900	0.8230	0.6870	0.9197	0.8845	24.9
			10^3	0.6173	0.5813	0.7433	0.4946	0.9616	0.9541	11.9
	600	AIC	10^1	0.6987	0.6325	0.7950	0.6412	0.9365	0.4912	19.7
			10^2	0.6910	0.6224	0.7672	0.5947	0.8826	0.8966	36.4
			10^3	0.6436	0.6473	0.7378	0.5787	0.8835	0.9580	36.1
		BIC	10^1	0.7154	0.6807	0.8055	0.6087	0.9400	0.5317	18.6
			10^2	0.7551	**0.7038**	0.8238	0.6578	0.9165	0.8708	25.9
			10^3	0.6263	0.5756	0.7561	0.5168	0.9832	0.9379	5.2
MIEXP	60	AIC	10^1	0.6981	0.6371	0.7931	0.6182	0.9345	0.5919	20.3
			10^2	0.6923	0.6209	0.7939	0.6120	0.9503	0.8977	15.4
			10^3	0.7474	0.6604	0.8327	0.7100	0.9552	0.9872	13.9
		BIC	10^1	0.6769	0.6194	0.7791	0.6034	0.9390	0.6186	18.9
			10^2	0.7404	0.6457	0.8269	0.7128	0.9506	0.9033	15.3
			10^3	0.7090	0.6297	0.8090	0.6412	0.9561	0.9881	13.6
	300	AIC	10^1	0.7141	0.6534	0.8031	0.6280	0.9365	0.5245	19.7
			10^2	0.7077	0.6304	0.8053	0.6370	0.9506	0.7852	15.3
			10^3	0.7308	0.6543	0.8212	0.6600	0.9552	0.9038	13.9
		BIC	10^1	0.6814	0.6130	0.7867	0.6057	0.9410	0.5538	18.3
			10^2	0.7551	0.6650	0.8373	0.7169	0.9523	0.7890	14.8
			10^3	0.7090	0.6297	0.8090	0.6412	0.9561	0.9053	13.6
	600	AIC	10^1	0.6756	0.6100	0.7795	0.6098	0.9365	0.4986	19.7
			10^2	0.6923	0.6185	0.7961	0.6120	0.9510	0.7620	15.2
			10^3	0.7308	0.6543	0.8212	0.6600	0.9552	0.8645	13.9
		BIC	10^1	0.6744	0.6076	0.7814	0.5973	0.9410	0.5283	18.3
			10^2	0.7385	0.6529	0.8277	0.6836	0.9532	0.7651	14.5
			10^3	0.7090	0.6297	0.8090	0.6412	0.9561	0.8651	13.6
MIEXP with Lower bound on Coefficient	60	AIC	10^1	0.6974	0.6526	0.7942	0.5870	0.9339	0.6089	20.5
			10^2	0.7006	0.6257	0.7997	0.6287	0.9497	0.9047	15.6
			10^3	0.7071	0.6305	0.8063	0.6307	0.9555	0.9864	13.8
		BIC	10^1	0.6603	0.6051	0.7707	0.5701	0.9390	0.6362	18.9
			10^2	**0.7641**	0.6705	0.8424	**0.7328**	0.9503	0.9098	15.4
			10^3	0.6686	0.5944	0.7829	0.5848	0.9552	0.9882	13.9
	300	AIC	10^1	0.7224	0.6452	0.8122	0.6648	0.9355	0.5342	20.0
			10^2	0.7077	0.6304	0.8053	0.6370	0.9506	0.7904	15.3
			10^3	0.7071	0.6305	0.8063	0.6307	0.9555	0.9113	13.8
		BIC	10^1	0.7308	0.6554	0.8175	0.6606	0.9416	0.5644	18.1
			10^2	**0.7641**	0.6739	**0.8425**	0.7294	0.9516	0.7978	15.0
			10^3	0.6769	0.6004	0.7887	0.5973	0.9561	0.9240	13.6
	600	AIC	10^1	0.7532	0.6910	0.8327	0.6691	0.9368	0.5062	19.6
			10^2	0.6994	0.6183	0.7993	0.6370	0.9506	0.7672	15.3
			10^3	0.7071	0.6305	0.8063	0.6307	0.9555	0.8749	13.8
		BIC	10^1	0.6994	0.6267	0.7985	0.6189	0.9419	0.5388	18.0
			10^2	**0.7641**	0.6739	**0.8425**	0.7294	0.9516	0.7760	15.0
			10^3	0.6769	0.6004	0.7887	0.5973	0.9561	0.8971	13.6

Table 3.11 Performance of the models in terms of prediction for the LSVT dataset.

Method	Time (sec)	Model type	Big M	Acc.	F-score	Harmonic mean	NPV	sparsity	Opt gap	Ave. of features
Regularized Log.Reg	60	NA	NA	0.7143	NAN	0.0062	0.9083	0.0031	0.0000	384.8
Logistic Regression			NA	0.3881	NAN	0.0021	**1.0000**	0.0010	0.0000	385.6
Regularized MIEXP	60	AIC	10^1	0.7905	0.8077	0.8667	0.8583	0.9790	0.4700	8.1
			10^2	0.8095	0.8392	0.8777	0.8583	0.9648	0.9209	13.6
			10^3	0.5762	0.6371	0.7130	0.6167	0.9873	0.9746	4.9
		BIC	10^1	0.7619	0.7838	0.8526	0.9083	0.9811	0.4685	7.3
			10^2	0.7619	0.8040	0.8440	0.8000	0.9645	0.9307	13.7
			10^3	0.6286	0.7258	0.7544	0.5000	0.9966	0.9505	1.3
	300	AIC	10^1	0.7619	0.7802	0.8483	0.8583	0.9811	0.3538	7.3
			10^2	0.7762	0.7973	0.8458	0.9500	0.9593	0.8638	15.7
			10^3	0.5286	0.5929	0.6637	0.5167	0.9966	0.9567	1.3
		BIC	10^1	0.7905	0.8238	0.8684	0.8833	0.9832	0.3480	6.5
			10^2	0.7000	0.7113	0.8023	0.8250	0.9697	0.8599	11.7
			10^3	0.6952	0.7847	0.8042	**0.6000**	**0.9969**	0.9282	1.2
	600	AIC	10^1	0.8095	0.8394	0.8770	0.9000	0.9821	0.3031	6.9
			10^2	0.7905	0.8120	0.8646	0.9167	0.9627	0.8372	14.4
			10^3	0.5619	0.6290	0.6896	0.5500	0.9946	0.9547	2.1
		BIC	10^1	0.7071	0.7658	0.8044	0.7000	0.9839	0.2992	6.2
			10^2	0.7643	0.7929	0.8465	0.8833	0.9733	0.8366	10.3
			10^3	0.6619	0.7514	0.7821	0.5667	0.9902	0.9264	3.8
MIEXP	60	AIC	10^1	0.7738	0.8075	0.8581	0.6833	0.9795	0.4919	7.9
			10^2	0.7619	0.7698	0.8505	0.7750	0.9858	0.8931	5.5
			10^3	0.7952	0.8124	0.8761	0.8500	0.9863	0.9851	5.3
		BIC	10^1	0.7476	0.7696	0.8384	0.8833	0.9821	0.5153	6.9
			10^2	0.7786	0.7974	0.8596	0.6250	0.9860	0.8831	5.4
			10^3	0.8190	0.8261	0.8901	0.7917	0.9855	0.9871	5.6
	300	AIC	10^1	0.7571	0.7767	0.8499	0.8500	0.9813	0.3591	7.2
			10^2	0.7786	0.7788	0.8596	0.8750	0.9860	0.6763	5.4
			10^3	0.7952	0.8124	0.8761	0.8500	0.9863	0.8306	5.3
		BIC	10^1	0.7643	0.7841	0.8494	0.8500	0.9832	0.3816	6.5
			10^2	0.7786	0.7974	0.8596	0.6250	0.9860	0.6804	5.4
			10^3	0.8357	0.8368	0.9009	0.8417	0.9858	0.8486	5.5
	600	AIC	10^1	0.7452	0.7772	0.8439	0.8500	0.9819	0.3305	7
			10^2	0.7786	0.7788	0.8596	0.8750	0.9860	0.6053	5.4
			10^3	0.8262	0.8429	0.8937	0.8500	0.9863	0.7858	5.3
		BIC	10^1	0.7286	0.7735	0.8240	0.8000	0.9845	0.2899	6
			10^2	0.7786	0.7974	0.8596	0.6250	0.9860	0.6071	5.4
			10^3	0.8214	0.8284	0.8934	0.9667	0.9858	0.7929	5.5
MIEXP with Lower bound on Coefficient	60	AIC	10^1	**0.8738**	**0.8894**	**0.9208**	0.9417	0.9788	0.5068	8.2
			10^2	0.7619	0.7698	0.8504	0.7750	0.9852	0.8981	5.7
			10^3	0.8429	0.8568	0.9045	0.8500	0.9863	0.9851	5.3
		BIC	10^1	0.7310	0.7534	0.8285	0.8833	0.9801	0.5368	7.7
			10^2	0.7786	0.7942	0.8595	0.6750	0.9858	0.8938	5.5
			10^3	0.8381	0.8457	0.9023	0.9083	0.9858	0.9868	5.5
	300	AIC	10^1	0.8429	0.8439	0.9021	0.9667	0.9803	0.3987	7.6
			10^2	0.7619	0.7698	0.8505	0.7750	0.9858	0.7281	5.5
			10^3	0.8286	0.8435	0.8941	0.8500	0.9863	0.9067	5.3
		BIC	10^1	0.7143	0.7617	0.8183	0.8000	0.9813	0.4332	7.2
			10^2	0.7786	0.7942	0.8596	0.6750	0.9860	0.7158	5.4
			10^3	0.8381	0.8457	0.9023	0.9083	0.9858	0.9240	5.5
	600	AIC	10^1	0.8405	0.8538	0.9007	0.8500	0.9819	0.3418	7
			10^2	0.7786	0.7788	0.8596	0.8750	0.9860	0.6213	5.4
			10^3	0.8262	0.8429	0.8937	0.8500	0.9863	0.7879	5.3
		BIC	10^1	0.6119	0.7548	0.7379	0.7500	0.9834	0.3408	6.4
			10^2	0.7786	0.7974	0.8596	0.6250	0.9860	0.6290	5.4
			10^3	0.8214	0.8266	0.8915	0.9083	0.9858	0.7950	5.5

Table 3.12 Performance of the models in terms of prediction for the Colon dataset.

Method	Time (sec)	Model type	Big M	Acc.	F-score	Harmonic mean	NPV	sparsity	Opt gap	Ave. of features
Regularized Log.Reg	60	NA	NA	0.9589	0.9381	0.0598	0.9583	0.0309	0.0000	954.6
Logistic Regression			NA	0.3589	0.5027	0.0046	0.0167	0.0023	0.0000	982.7
Regularized MIEXP	60	AIC	10^1	0.7357	0.6993	0.8299	0.6250	0.9938	0.9070	6.1
			10^2	0.6018	0.6712	0.7260	0.4050	0.9969	0.9522	3.1
			10^3	0.8250	0.7924	0.8918	0.7267	0.9902	0.9559	9.7
		BIC	10^1	0.4982	0.5521	0.6241	0.3100	0.9992	0.8869	0.8
			10^2	0.3607	0.5049	0.5110	0.0250	**0.9999**	0.9509	0.1
			10^3	0.5143	0.6679	0.6447	0.3217	0.9986	0.9579	1.4
	300	AIC	10^1	0.7500	0.7136	0.8503	0.6200	0.9933	0.8771	6.6
			10^2	0.7232	0.6775	0.8270	0.6417	0.9962	0.9345	3.7
			10^3	0.8500	0.7971	0.9092	0.8033	0.9902	0.9386	9.7
		BIC	10^1	0.6839	0.6444	0.7909	0.5933	0.9979	0.8389	2.1
			10^2	0.6000	0.5817	0.7423	0.4483	0.9985	0.9289	1.5
			10^3	0.7411	0.6875	0.8377	0.6467	0.9976	0.9460	2.4
	600	AIC	10^1	0.7482	0.7167	0.8488	0.5950	0.9929	0.8655	7.0
			10^2	0.7393	0.6851	0.8388	0.6833	0.9968	0.9194	3.2
			10^3	0.8625	0.7927	0.9169	0.8483	0.9904	0.9268	9.5
		BIC	10^1	0.8304	0.8222	0.8942	0.7600	0.9968	0.8139	3.2
			10^2	0.6125	0.5945	0.7529	0.4500	0.9982	0.9018	1.8
			10^3	0.7696	0.7013	0.8543	0.7550	0.9975	0.9109	2.5
MIEXP	60	AIC	10^1	0.8625	0.8963	0.9127	0.8450	0.9952	0.6618	4.7
			10^2	0.9857	0.9857	0.9913	0.9750	0.9979	0.8947	2.1
			10^3	0.9143	0.8690	0.9503	0.9183	0.9978	0.9871	2.2
		BIC	10^1	0.8607	0.8138	0.9210	0.8883	0.9957	0.6677	4.2
			10^2	0.9857	0.9857	0.9912	0.9750	0.9978	0.9069	2.2
			10^3	0.8857	0.8413	0.9336	0.8817	0.9978	0.9879	2.2
	300	AIC	10^1	0.8464	0.9373	0.9042	0.8233	0.9958	0.5712	4.1
			10^2	1.0000	1.0000	0.9989	1.0000	0.9979	0.8947	2.1
			10^3	0.8893	0.9319	0.9373	0.8817	0.9978	0.9871	2.2
		BIC	10^1	0.9018	0.8348	0.9442	0.9383	0.9963	0.5943	3.6
			10^2	0.9571	0.9357	0.9746	0.9550	0.9978	0.9069	2.2
			10^3	0.9143	0.8524	0.9503	0.9383	0.9978	0.9879	2.2
	600	AIC	10^1	0.8464	0.9536	0.9043	0.8233	0.9960	0.5245	3.9
			10^2	1.0000	1.0000	0.9989	1.0000	0.9979	0.8947	2.1
			10^3	0.8893	0.9484	0.9373	0.8817	0.9978	0.9871	2.2
		BIC	10^1	0.8750	0.7871	0.9300	0.9133	0.9965	0.5053	3.4
			10^2	0.9571	0.9357	0.9746	0.9550	0.9978	0.9069	2.2
			10^3	0.9143	0.8524	0.9503	0.9383	0.9978	0.9879	2.2
MIEXP with Lower bound on Coefficient	60	AIC	10^1	0.8054	0.9468	0.8807	0.8000	0.9951	0.6753	4.8
			10^2	0.9857	0.9857	0.9913	0.9750	0.9979	0.8947	2.1
			10^3	0.9571	0.9357	0.9746	0.9550	0.9978	0.9871	2.2
		BIC	10^1	0.8893	0.8135	0.9363	0.9217	0.9956	0.6777	4.3
			10^2	0.9857	0.9857	0.9912	0.9750	0.9978	0.9069	2.2
			10^3	0.9143	0.8750	0.9490	0.9300	0.9978	0.9879	2.2
	300	AIC	10^1	0.9179	0.9548	0.9517	0.9167	0.9957	0.6022	4.2
			10^2	**1.0000**	**1.0000**	**0.9989**	**1.0000**	0.9979	0.8947	2.1
			10^3	0.8589	0.9526	0.9164	0.8817	0.9978	0.9871	2.2
		BIC	10^1	0.8893	0.8246	0.9366	0.9250	0.9963	0.6137	3.6
			10^2	0.9429	0.9925	0.9640	0.9550	0.9978	0.9069	2.2
			10^3	0.8571	0.9527	0.9154	0.8850	0.9978	0.9879	2.2
	600	AIC	10^1	0.9143	0.9636	0.9494	0.9183	0.9958	0.5661	4.1
			10^2	1.0000	1.0000	0.9989	1.0000	0.9979	0.8947	2.1
			10^3	0.8589	0.9736	0.9164	0.8817	0.9978	0.9871	2.2
		BIC	10^1	0.9018	0.8437	0.9443	0.9250	0.9964	0.5653	3.5
			10^2	0.9429	1.0000	0.9640	0.9550	0.9978	0.9069	2.2
			10^3	0.8571	0.9736	0.9154	0.8850	0.9978	0.9879	2.2

Table 3.13 Performance of the models in terms of prediction for the Leukemia dataset.

LSVT, Colon, and Leukemia datasets are categorized as hard datasets based on our definition. We can observe the impact of high dimensionality on the performance of the models through these datasets. By increasing the number of features, it takes more time for the upper bound to certify the optimality [56]. We can easily observe this for hard datasets. However, we can see that the models are of better performance in terms of prediction accuracy compared to logistic regression models while constructing sparse models which increases the intrepretability of the models. This implies that models generated by the logistic regression may be overfitted which can be easily observed in Leukemia dataset where the prediction accuracy is extremely low for this model. Overall, overfitting occurs when the model performs well over the training data but reveals a poor performance over the test data. To check if overfitting occurs for logistic regression model in Leukemia dataset, we can look at the accuracy, F-score, and NPV metrics over the training data for this version and obtained value 1 for all the metrics. Hence, it is a proof of overfitting. Moreover, we can observe that for LSVT, Colon, and Leukemia datasets, performance of the regularized logistic regression is significantly better than logistic regression model in terms of prediction accuracy. This observation represents the importance of regularization in these models. Finally, we note that for all the hard datasets, there exists at least one method which outperforms logistic regression models in both terms of intrepretability and prediction accuracy.

3.2 Computational Results For Optimal Histograms

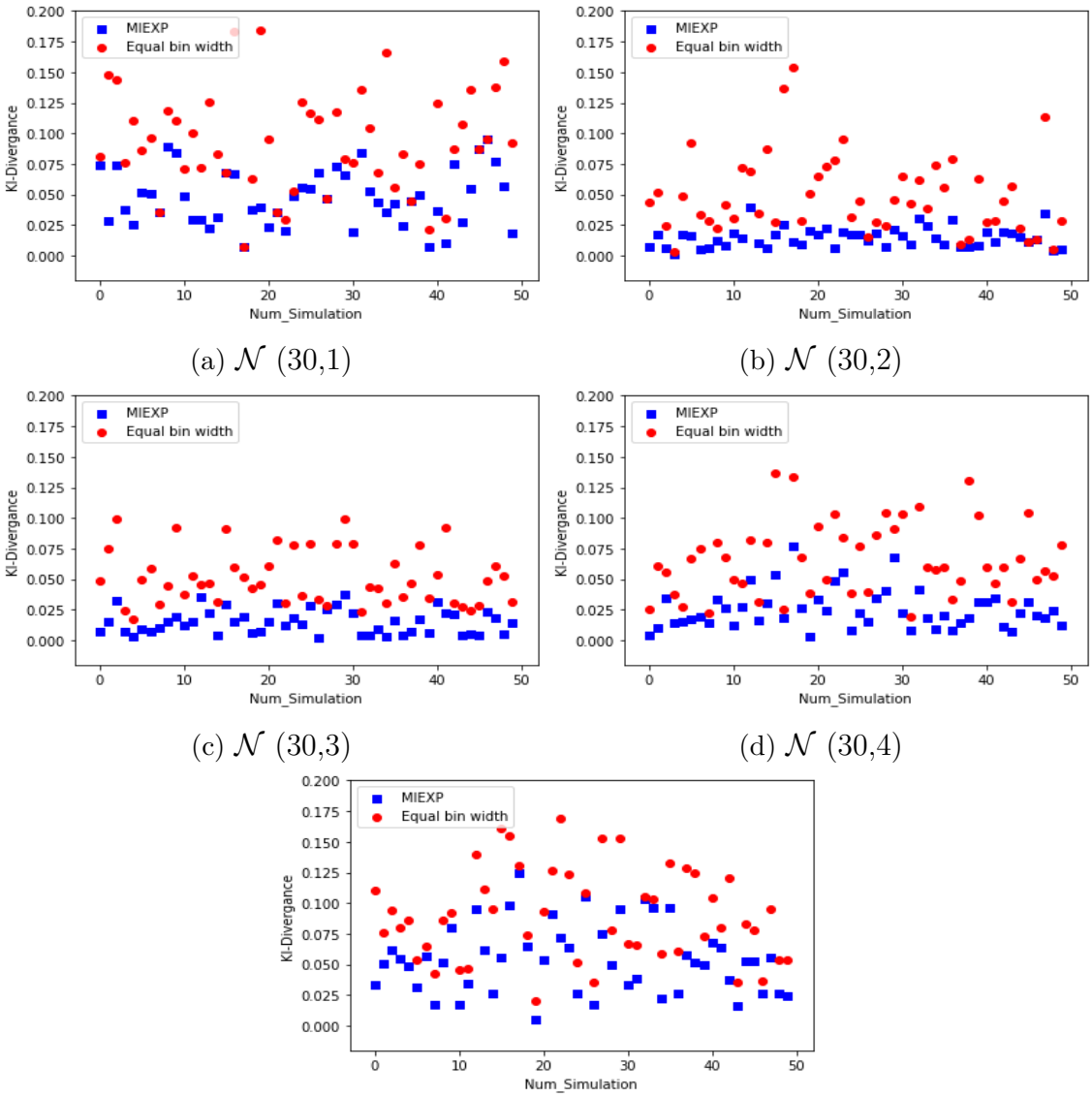
In this section, we present the computational results for the optimal histograms model. For this part, we consider three different distributions (Normal, Gamma, and Poisson) and perform 50 simulations with a sample size of 100 random integer variables. The goal of this model is to construct histograms with the aim of minimizing the KL-divergence. Hence, the KL divergence of the histograms generated with our proposed model and that of an equal bin width are compared. In this experiment we set the solution of Python package as an initial solution for our method. We set the number of bins for each simulation as follows:

$$\text{Number of bins} = \min\left\{10, \frac{|S_i|}{2}\right\}, \tag{3.6}$$

where, $|S_i|$ is the number of different values generated at each simulation $i = 1, \ldots, 50$. As an illustration, the value of $|S|$ for the vector [1,1,1,5,6,6,9,9,9,9] is 4. In the proceeding parts, we will present the performance of the model with regard to the KL-divergence for each of the distributions.

I. **Normal distribution**

When considering the Normal distribution, we perform five different sets of simulations by setting different values for the variance parameter. The mean of all the Normal distributions, assumed to be equal to 30, as it is not influential to the performance of the model. In Figure 3.8, we provide the results for the mentioned structures.



(a) $\mathcal{N}$ (30,1)

(b) $\mathcal{N}$ (30,2)

(c) $\mathcal{N}$ (30,3)

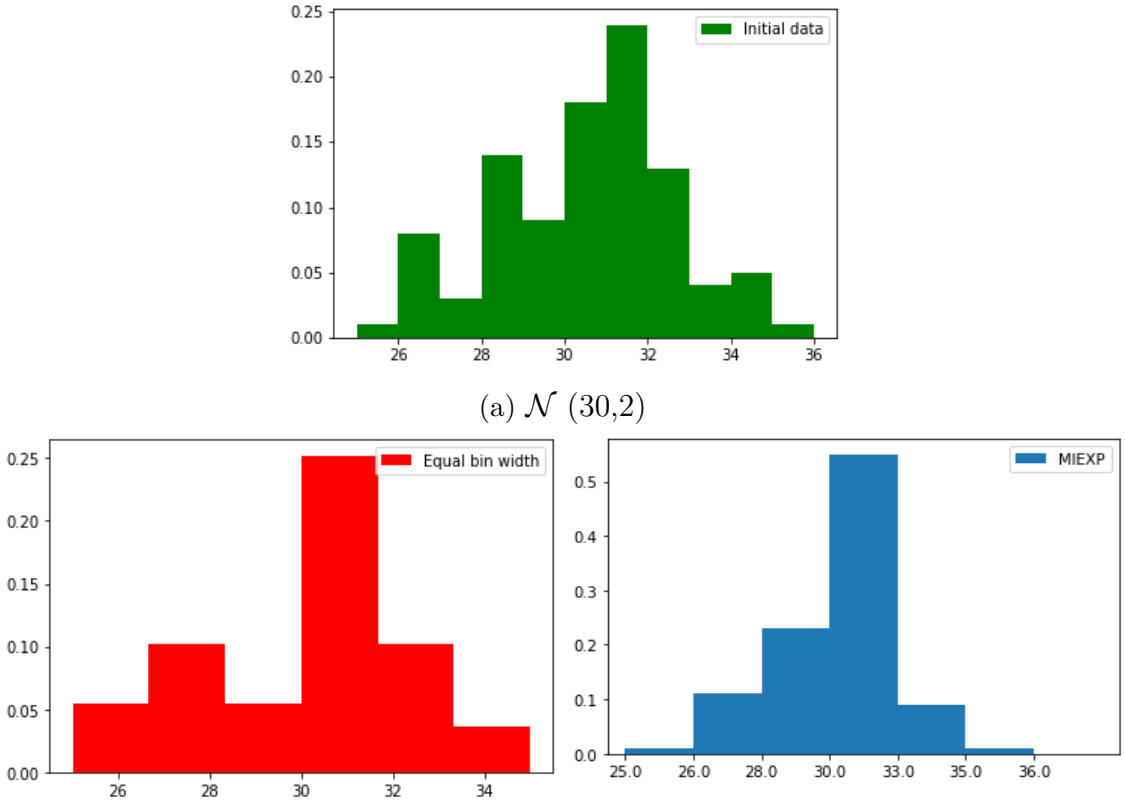
(d) $\mathcal{N}$ (30,4)

(e) $\mathcal{N}$ (30,5)

Figure 3.8 KL-divergence of MIEXP method and Equal bin width for Normal distributions.

By construction, the KL-divergence of MIEXP model is always smaller, however, for some instances, there are significant differences with respect to the value of KL-divergence. As an illustration, we can point the 48th instance of $\mathcal{N}$ (30,2). Figure 3.9 provides the histograms of this instance for the both models. In Figure 3.9, we provided the histogram of 48th simulation for the Normal distribution in which there is relatively significant difference between MIEXP model and equal bin widths in terms of KL-divergence value. The values for MIEXP and Equal bin widths are 0.03 and 0.11 ,respectively.

Generally, the larger the variance is, the harder the problem will be. The reason is that increasing the variance results in generation of a wider range of instances which results in adding more decision variables and constraints to the model.



(a) $\mathcal{N}$ (30,2)

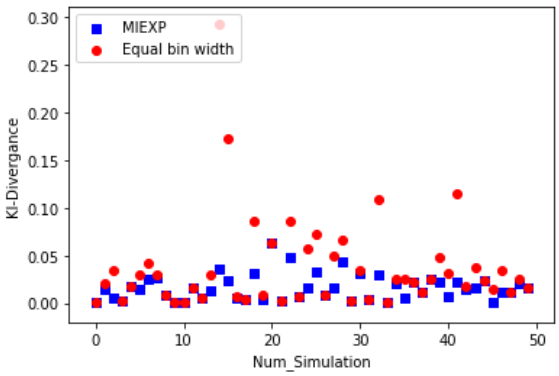
(b) $\mathcal{N}$ (30,2)

(c) $\mathcal{N}$ (30,2)

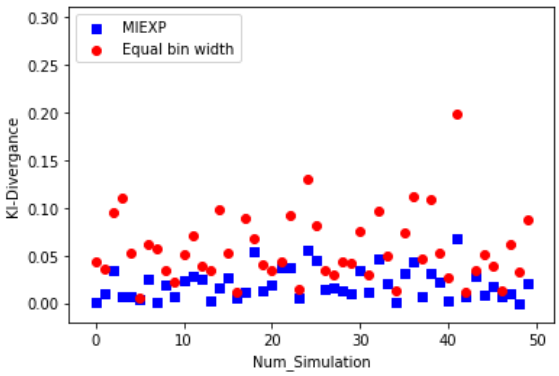
Figure 3.9 Histograms of initial data, MIEXP model, and Equal bin widths for the 48th simulation of $\mathcal{N}$ (30,2).

II. **Gamma distribution**

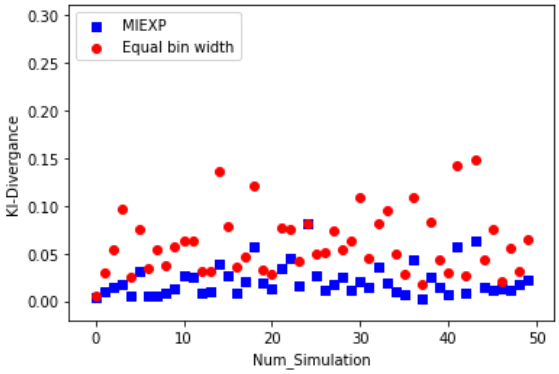
For Gamma distribution, we considered the five sets of parameters showed in (Figure3.10).



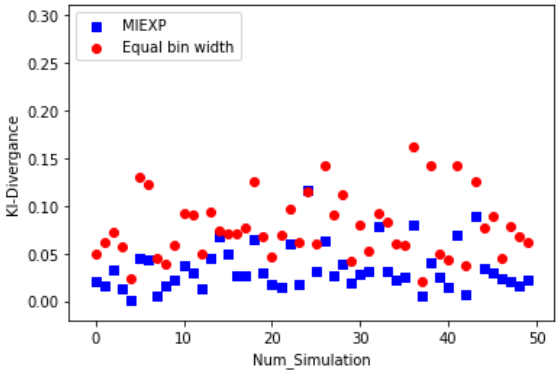
(a) Gamma (1,1)



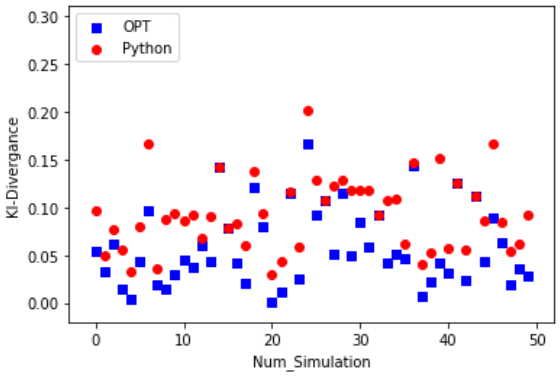
(b) Gamma (1,2)



(c) Gamma (1,3)



(d) Gamma (1,4)

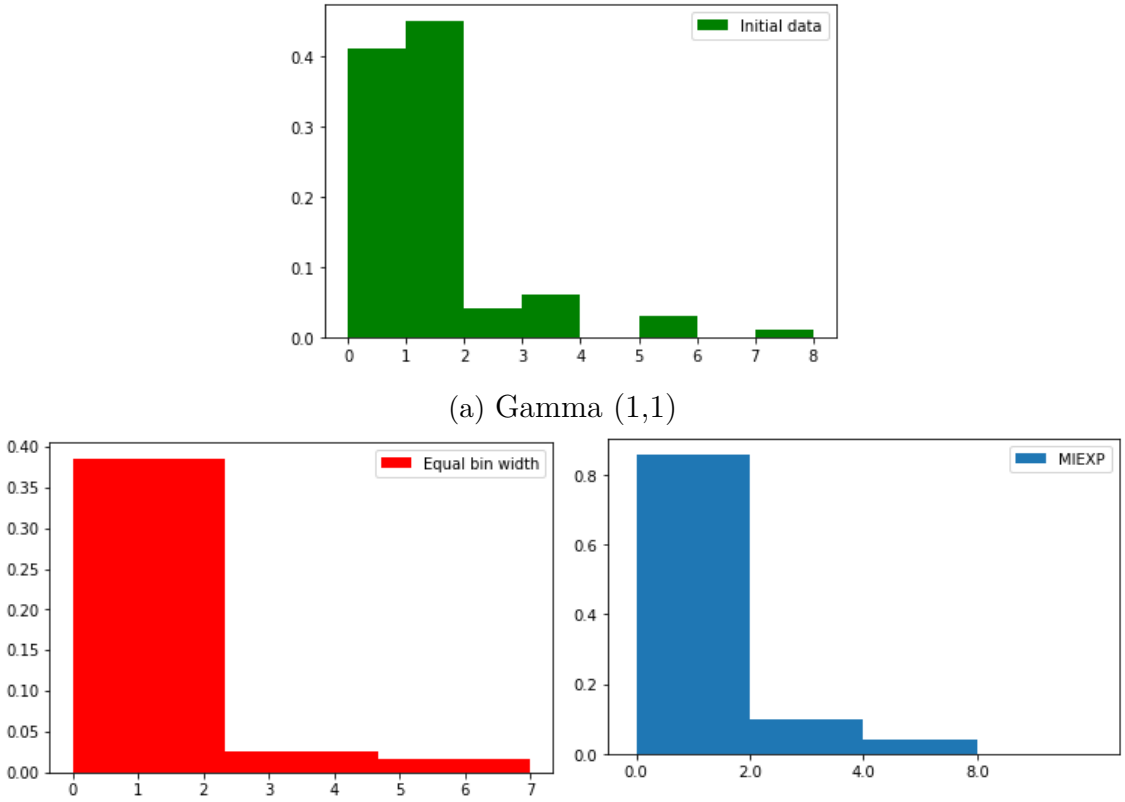


(e) Gamma (1,5)

Figure 3.10 KL-divergence of MIEXP method and equal bin widths for Gamma distributions.

For the Gamma distribution, the number of instances with significant difference in KL-divergence value is lower than that of Normal case. For this distribution we also provide the histograms generated by MIEXP and Equal bin width model for the 15th instance of Gamma (1,1). Similar to the case of Normal distribution, by

increasing the scale parameter of Gamma distribution, it takes more time for the solver to obtain the desired optimality gap. However, it is not as extreme as Normal distribution.



(a) Gamma (1,1)

(b) Gamma (1,1)

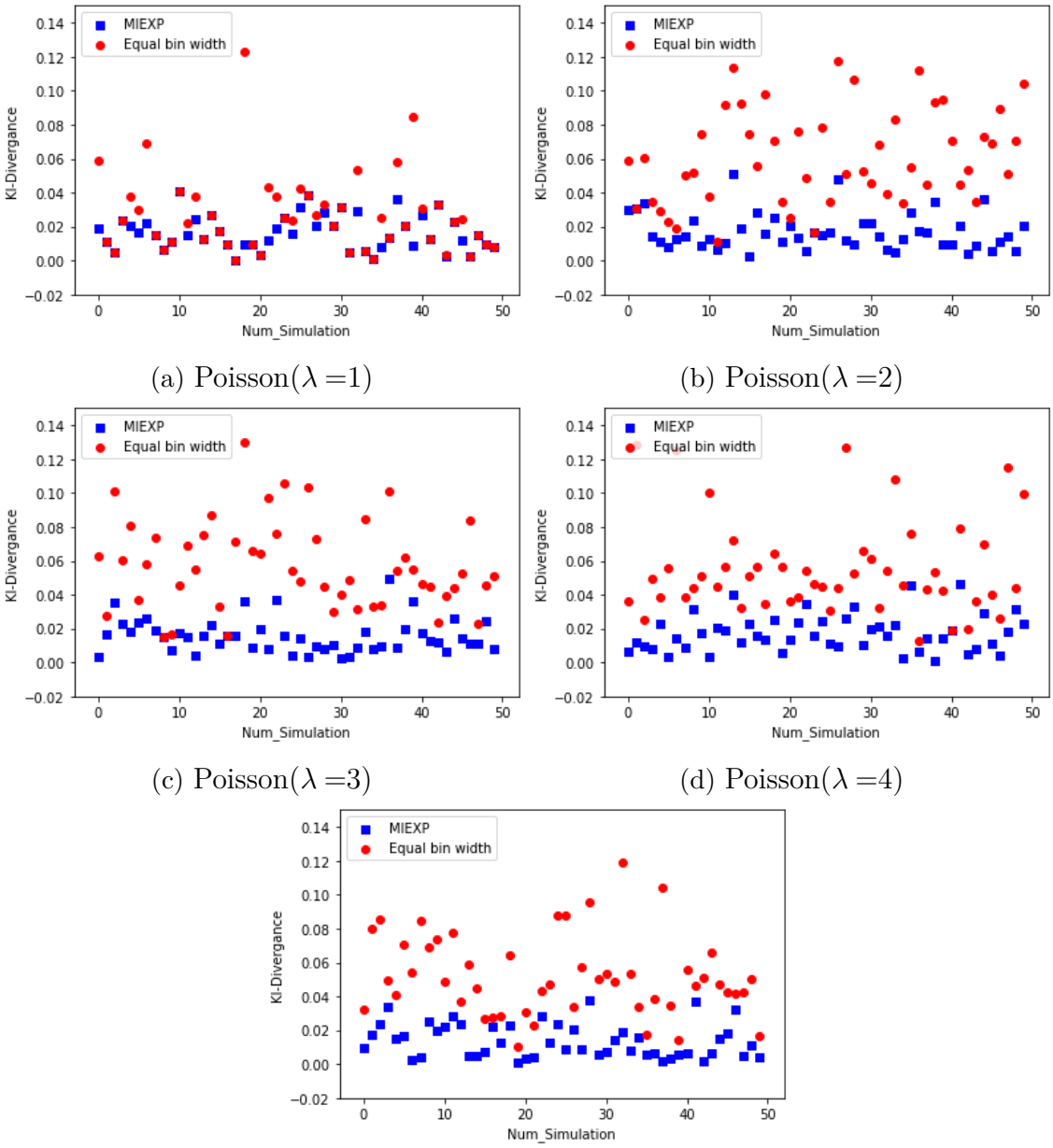
(c) Gamma (1,1)

Figure 3.11 Histograms of initial data, MIEXP model, and Equal bin widths for the 15th simulation of Gamma(1,1).

For Gamma distribution, we provide the histograms of 15th simulations in which the values of KL-divergence for MIEXP model and Equal bin widths are 0.03 and 0.29 respectively. In the next part, we will provide the experimental analysis for the Poisson distribution.

III. **Poissondistribution**

For Poisson distribution, we consider the five sets of parameters showed in Figure.3.12.



(a) Poisson(λ =1)

(b) Poisson(λ =2)

(c) Poisson(λ =3)

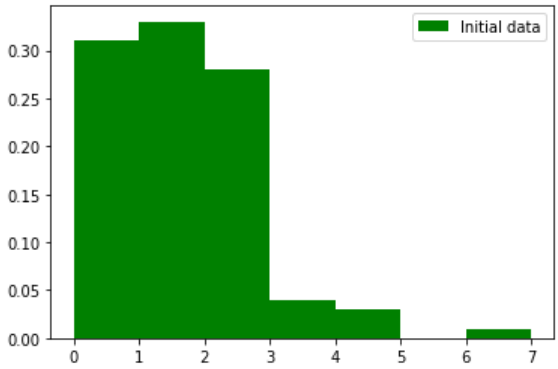
(d) Poisson(λ =4)

(e) Poisson(λ =5)

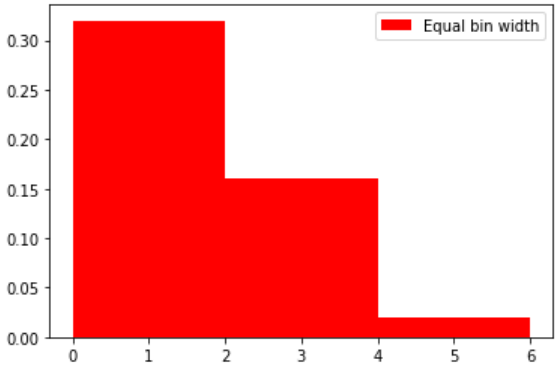
Figure 3.12 KL-divergence of MIEXP method and Equal bin widths for Poissondistributions.

Poisson distribution, shows a considerable difference between the KL-divergence of MIEXP and Equal bin width model. Moreover, the performance of the solver for the Poisson distribution in terms of solution time is relatively better than Gamma and Normal distributions. In fact, by increasing the parameter of Poisson distribution (λ), the run time increases similar to the Gamma and Normal distributions, however,

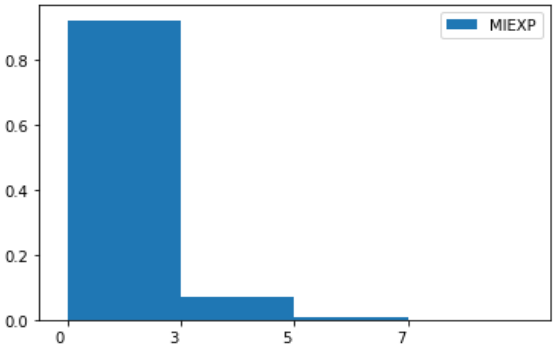
this increase in solution time is not as much as Normal and Gamma distribution. For this distribution, we provide the histograms of 18th instance of Poisson(λ =1).For disPoissontribution, we provide the histograms of 15th simulations in which the values of KL-divergence for MIEXP model and Equal bin widths are 0.009 and 0.123, respectively.



(a) Poisson(λ =1)



(b) Poisson(λ =1)



(c) Poisson(λ =1)

Figure 3.13 Histograms of initial data, MIEXP model, and Equal bin widths for the 18th simulation of Poisson(λ =1).

4. Conclusion

This thesis is dedicated to formulate and solve two problems from the intersection of machine learning and optimization called *feature subset selection in logistic regression* and *optimal histogram construction*. For both of the mentioned problems, we developed a mixed-integer exponential cone programming (MIEXP) model. To the best of our knowledge, there has not been any attempt in the literature to formulate these problems as a conic program. Our motivation for the first problem was the undeniable role of logistic regression in fitting a model for the datasets in which the value of dependent variable is limited to a finite set of values. The presence of this type of dataset in important research areas such as medical science has made logistic regression an area of research interest. In this thesis, given a binary output dataset, we aimed to fit a logistic regression model. Our main goal was to select the truly important independent variables when constructing the model, which is known as feature subset selection problem. There are multiple heuristic and exact methods for feature subset selection in logistic regression such as *stepwise model selection methods* and *Bayesian methodologies*. In this thesis, we developed a novel exact method using the conic representation of the maximum likelihood function. We also introduced modified versions of this model by considering widely used goodness of fit measures (GOF) including Akaike Information Criteria (AIC), Bayesian Information Criteria (BIC), and Adjusted McFadden. Additionally, we analyzed the impact of regularization on the performance of each version. For the evaluation purposes, the performance of the models in both terms of interpretability (constructing sparse models) and prediction accuracy were considered over the toy examples provided in [39] and a group of benchmark datasets from the literature. The performance of our models over the toy examples were significantly better than those methods provided in [39] and for the benchmark datasets, were either better or very close to these methods. Although our main focus was on the interpretability of the generated model by choosing the truly important features, the performance of this model in terms of prediction accuracy was highly satisfying as well. We note that, in the case of high dimensionality, especially when the number of features is larger than the number of observations, the model might fail to reach the desired optimality gap.

We showed that the reason might be slow convergence of the upper bound to certify the optimality gap while the current feasible solution might already be an optimal solution. One solution to reduce the optimality gap might be modifying the model by adding valid inequalities which can be considered in the future works.

In the second model, we introduced an MIEXP formulation for the optimal histogram construction problem. Our motivation to model this problem was its importance in the field of machine learning and information theory. Our solution approach was based on the conic representation for the KL-divergence function. In this problem, for a given set of integer data, we developed a model to fit a histogram to this data with the aim of minimizing the KL-divergence of the data to the fitted model through finding optimal bin widths in the histogram. We experimented with different probability distributions for the data including Normal, Gamma, and Poisson. Then, we compared the KL-divergence of our fitted model with the case of equal bin widths. The result showed that at the worst case, the KL-divergence value for our model was equal to that of equal bins width while in the most cases it was significantly lower. We observed that increasing the variance of the data will impose a computational burden in solving the model. The performance of the model in terms of solution time was significantly better for the Poisson distribution, especially in comparison with the Normal distribution.

Finally, we remind that the common characteristic of both problems considered in this thesis was that their objective functions were *Exponential Cone-representable*. For the future works, one might consider the sets and functions which have exponential cone representation such as (*entropy function*) and try to formulate related applications such as (*Decision Tree*) as an MIEXP problem.

Appendices

A Computational Results for Sparse Linear Regression

For the purpose of implementation, three different collections of random data points are generated considering different distributions for the error term as the following:

I. $\epsilon \sim$ *Uniform* (-1,1)

II. $\epsilon \sim$ *Normal* (0,1)

III. Standard-Cauchy

Toy Example:

To generate the data points, we assume ten independent variables (features), where the coefficients of the first five variables are set in the order of 10^2 while the rest five are set in the order of 10. The true coefficient vector is [250,-200,220,-190,180,-10,15,8,-13,17,10], and 150 data points and error terms are randomly generated by this predetermined coefficient vector. For the purpose of evaluating the performance of the model in the term of sparsity, two different upper-bounds are set for the k in Table 1 which are the number of variables to be selected.

$k = 10$	ℓ_1	Uniform	[10.09 249.97 -200.17 220.27 -189.96 179.99 -10.20 14.86 7.86 -13.00 17.07]
		Normal	[10.02 250.16 -199.56 220.06 -189.77 179.95 -9.74 15.36 7.68 -12.73 17.29]
		Cauchy	[10.24 249.90 -199.72 220.11 -189.61 179.82 -9.84 15.02 6.87 -12.76 16.86]
	ℓ_2	Uniform	[10.07 249.86 -200.12 220.16 -190.04 179.98 -10.14 14.88 7.84 -13.05 17.14]
		Normal	[10.02 249.94 -199.84 220.06 -189.66 179.94 -9.89 15.28 7.65 -12.97 17.08]
		Cauchy	[10.43 250.26 -198.76 221.98 -189.72 180.40 -10.18 14.00 6.99 -13.18 16.55]
	ℓ_∞	Uniform	[10.07 249.95 -199.94 220.07 -190.01 180.03 -10.11 14.94 7.96 -13.17 17.00]
		Normal	[10.15 249.88 -200.10 220.31 -189.43 179.86 -10.07 15.06 7.71 -12.99 17.07]
		Cauchy	[12.52 251.76 -193.11 228.18 -187.60 179.38 -6.04 13.91 10.21 -16.24 13.40]
$k = 5$	ℓ_1	Uniform	[6.50 249.70 -196.91 223.05 -192.84 171.67 0.0 0.0 0.0 0.0 0.0]
		Normal	[7.02 250.04 -195.51 222.49 -192.51 171.33 0.0 0.0 0.0 0.0 0.0]
		Cauchy	[8.64 251.48 -195.55 226.67 -191.27 173.31 0.0 0.0 0.0 0.0 0.0]
	ℓ_2	Uniform	[7.33 250.08 -198.92 222.11 -190.69 174.20 0.0 0.0 0.0 0.0 0.0]
		Normal	[7.28 250.18 -198.66 222.00 -190.36 174.21 0.0 0.0 0.0 0.0 0.0]
		Cauchy	[7.74 250.53 -197.47 223.95 -190.43 174.80 0.0 0.0 0.0 0.0 0.0]
	ℓ_∞	Uniform	[10.24 251.64 -193.29 218.81 -188.85 174.27 0.0 0.0 0.0 0.0 0.0]
		Normal	[11.06 250.43 -192.77 220.07 -189.25 171.87 0.0 0.0 0.0 0.0 0.0]
		Cauchy	[7.50 254.62 -189.26 231.09 -189.67 170.42 0.0 0.0 0.0 0.0 0.0]

Table 1 The estimated coefficient vectors using ℓ_1, ℓ_2, and ℓ_∞ norms with respect to different error term distributions and values of k.

In the case that k was equal to five the model chose the variables with highest coefficient absolute values. It can imply the significant effect of the variables with higher coefficient to generate a model that predict the dependent variable more accurately. Note that the first element of each vector is the intercept which is forced to be always chosen by the model.

In Table-2, the Mean Squared Error (MSE) of estimated coefficients for different values of k is reported.

	$k=10$			$k=8$			$k=5$		
	Normal	Uniform	Cauchy	Normal	Uniform	Cauchy	Normal	Uniform	Cauchy
ℓ_1	0.07	0.03	**0.06**	1.34	1.35	**1.37**	2.96	2.96	3.03
ℓ_2	**0.06**	0.01	0.43	**1.31**	**1.32**	1.39	**2.93**	**2.93**	**2.94**
ℓ_∞	0.08	**0.00**	2.91	1.36	1.34	2.80	3.15	3.159	3.53

Table 2 Mean squared error (MSE) of estimated coefficient vectors.

As Table 2 represents, when k is equal to 10 and the model is not sparse, the norms ℓ_1, ℓ_2, and ℓ_∞ performs better when the error term distribution is Cauchy, Normal, and Uniform, respectively. However, by decreasing the number of variables to be selected to ensure sparsity, in most of the cases ℓ_2 reveals a better performance (for k=8 and k=5).

The coefficient error are measured in Table 2 but to evaluate the performance of model, the training and test errors should be considered. For this purpose, we implemented 10-fold cross validation for 150 data points where 90 percent of the data used for training and remaining 10 percent for the test at each iteration. This procedure repeated for 5 iterations. Table 3 shows the average of training errors and Table 4 represents that of test errors.

	$k=10$			$k=8$			$k=5$		
	Normal	Uniform	Cauchy	Normal	Uniform	Cauchy	Normal	Uniform	Cauchy
ℓ_1	0.08	0.04	**0.76**	0.58	0.58	0.98	1.41	1.40	1.58
ℓ_2	**0.08**	0.04	0.74	**0.56**	**0.57**	**0.96**	**1.39**	**1.38**	**1.55**
ℓ_∞	0.09	**0.04**	1.49	0.60	0.61	1.51	1.49	1.497	1.85

Table 3 Train errors for sparse linear regression model.

	$k=10$			$k=8$			$k=5$		
	Normal	Uniform	Cauchy	Normal	Uniform	Cauchy	Normal	Uniform	Cauchy
ℓ_1	0.27	0.15	**1.79**	1.87	1.92	**2.84**	4.40	4.34	4.85
ℓ_2	**0.27**	0.15	1.96	**1.82**	**1.84**	2.87	**4.31**	**4.29**	**4.77**
ℓ_∞	0.30	**0.14**	4.77	1.89	1921.00	4.90	4.52	4.51	5.76

Table 4 Test errors for sparse linear regression model.

As it can be concluded from these two tables, almost the same pattern for MSE

applies for the training and test error meaning that in the presence of sparsity, the ℓ_2 norms has the best performance, in general.

BIBLIOGRAPHY

[1] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.

[2] Hande Y Benson and Ümit Sağlam. Mixed-integer second-order cone programming: A survey. In *Theory Driven by Influential Applications*, pages 13–36. INFORMS, 2013.

[3] Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

[4] Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear algebra and its applications*, 284(1-3):193–228, 1998.

[5] Yu E Nesterov and Michael J Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on optimization*, 8(2):324–364, 1998.

[6] Arkadii Nemirovskii and Katya Scheinberg. Extension of karmarkar's algorithm onto convex quadratically constrained quadratic problems. *Mathematical Programming*, 72(3):273–289, 1996.

[7] Mehmet Tolga Çezik and Garud Iyengar. Cuts for mixed 0-1 conic programming. *Mathematical Programming*, 104(1):179–202, 2005.

[8] Alper Atamtürk and Vishnu Narayanan. Cuts for conic mixed-integer programming. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 16–29. Springer, 2007.

[9] Juan Pablo Vielma, Shabbir Ahmed, and George L Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450, 2008.

[10] Yong Cheng, Sarah Drewes, Anne Philipp, and Marius Pesavento. Joint network optimization and beamforming for coordinated multi-point transmission using mixed integer programming. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3217–3220. IEEE, 2012.

[11] Hassan Hijazi, Pierre Bonami, and Adam Ouorou. Robust delay-constrained routing in telecommunications. *Annals of Operations Research*, 206(1):163–181, 2013.

[12] Ho-Yin Mak, Ying Rong, and Zuo-Jun Max Shen. Infrastructure planning for electric vehicles with battery swapping. *Management Science*, 59(7):1557–1575, 2013.

[13] Joshua A Taylor and Franz S Hover. Convex models of distribution system reconfiguration. *IEEE Transactions on Power Systems*, 27(3):1407–1413, 2012.

[14] Lieven Vandenberghe and Stephen Boyd. Applications of semidefinite programming. *Applied Numerical Mathematics*, 29(3):283–299, 1999.

[15] Farid Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM journal on Optimization*, 5(1):13–51, 1995.

[16] Lieven Vandenberghe and Stephen Boyd. A primal—dual potential reduction method for problems involving matrix inequalities. *Mathematical programming*, 69(1-3):205–236, 1995.

[17] Zhi-Quan Luo, Wing-Kin Ma, Anthony Man-Cho So, Yinyu Ye, and Shuzhong Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010.

[18] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

[19] Miguel F Anjos, Bissan Ghaddar, Lena Hupp, Frauke Liers, and Angelika Wiegele. Solving k-way graph partitioning problems to optimality: The impact of semidefinite relaxations and the bundle method. In *Facets of combinatorial optimization*, pages 355–386. Springer, 2013.

[20] Franz Rendl, Giovanni Rinaldi, and Angelika Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 121(2):307, 2010.

[21] Tristan Gally, Marc E Pfetsch, and Stefan Ulbrich. A framework for solving mixed-integer semidefinite programs. *Optimization Methods and Software*, 33(3):594–632, 2018.

[22] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[23] Neha Midha and Dr Vikram Singh. A survey on classification techniques in data mining. *IJCSMS (International Journal of Computer Science & Management Studies) Vol*, 16, 2015.

[24] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[25] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[26] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[27] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[28] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[29] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[30] Alfonso Rojas-Domínguez, Luis Carlos Padierna, Juan Martín Carpio Valadez, Hector J Puga-Soberanes, and Héctor J Fraire. Optimal hyper-parameter tuning of svm classifiers with application to medical diagnosis. *IEEE Access*, 6:7164–7176, 2017.

[31] Cho-Jui Hsieh, Si Si, and Inderjit Dhillon. A divide-and-conquer solver for kernel support vector machines. In *International conference on machine learning*, pages 566–574, 2014.

[32] Sebastián Maldonado, Juan Pérez, Richard Weber, and Martine Labbé. Feature selection for support vector machines via mixed integer linear programming. *Information sciences*, 279:163–175, 2014.

[33] Ji Zhu, Saharon Rosset, Robert Tibshirani, and Trevor J Hastie. 1-norm support vector machines. In *Advances in neural information processing systems*, pages 49–56, 2004.

[34] Martine Labbé, Luisa I Martínez-Merino, and Antonio M Rodríguez-Chía. Mixed integer linear programming for feature selection in support vector machine. *Discrete Applied Mathematics*, 261:276–304, 2019.

[35] Marcus D Odom and Ramesh Sharda. A neural network model for bankruptcy prediction. In *1990 IJCNN International Joint Conference on neural networks*, pages 163–168. IEEE, 1990.

[36] Moshe Leshno and Yishay Spector. Neural network prediction analysis: The bankruptcy case. *Neurocomputing*, 10(2):125–147, 1996.

[37] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5-6):352–359, 2002.

[38] Joseph M Hilbe. *Logistic regression models*. Chapman and hall/CRC, 2009.

[39] Chun-Xia Zhang, Shuang Xu, and Jiang-She Zhang. A novel variational bayesian method for variable selection in logistic regression models. *Computational Statistics & Data Analysis*, 133:1–19, 2019.

[40] Alberto Del Pia, Santanu S Dey, and Robert Weismantel. Subset selection in sparse matrices. *arXiv preprint arXiv:1810.02757*, 2018.

[41] Ryuhei Miyashiro and Yuichi Takano. Mixed integer second-order cone programming formulations for variable selection in linear regression. *European Journal of Operational Research*, 247(3):721–731, 2015.

[42] Nahúm Cueto-López, Maria Teresa García-Ordás, Verónica Dávila-Batista, Víctor Moreno, Nuria Aragonés, and Rocío Alaiz-Rodríguez. A comparative study on feature selection for a risk prediction model for colorectal cancer. *Computer Methods and Programs in Biomedicine*, 2019.

[43] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[44] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[45] Cun-Hui Zhang et al. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.

[46] Minjung Kyung, Jeff Gill, Malay Ghosh, George Casella, et al. Penalized regression, standard errors, and bayesian lassos. *Bayesian Analysis*, 5(2):369–411, 2010.

[47] Vitara Pungpapong, Min Zhang, Dabao Zhang, et al. Selecting massive variables using an iterated conditional modes/medians algorithm. *Electronic Journal of Statistics*, 9(1):1243–1266, 2015.

[48] Veronika Ročková and Edward I George. Emvs: The em approach to bayesian variable selection. *Journal of the American Statistical Association*, 109(506):828–846, 2014.

[49] MD Koslovsky, Michael D Swartz, Luis Leon-Novelo, Wenyaw Chan, and Anna V Wilkinson. Using the em algorithm for bayesian variable selection in logistic regression models with related covariates. *Journal of statistical computation and simulation*, 88(3):575–596, 2018.

[50] Sadanori Konishi and Genshiro Kitagawa. *Information criteria and statistical modeling*. Springer Science & Business Media, 2008.

[51] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

[52] MP Wand. Data-based choice of histogram bin width. *The American Statistician*, 51(1):59–64, 1997.

[53] David W Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[54] Kevin H Knuth. Optimal data-based binning for histograms. *arXiv preprint physics/0605197*, 2006.

[55] Charles C Taylor. Akaike's information criterion and the histogram. *Biometrika*, 74(3):636–639, 1987.

[56] Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The annals of statistics*, pages 813–852, 2016.

[57] Elizabeth Million. The hadamard product. *Course Notes*, 3(6), 2007.

[58] Joaquín Pacheco, Silvia Casado, and Laura Núñez. A variable selection method based on tabu search for logistic regression models. *European Journal of Operational Research*, 199(2):506–511, 2009.

[59] Santiago Akle Serrano. *Algorithms for unsymmetric cone optimization and an implementation for problems with the exponential cone*. PhD thesis, Citeseer, 2015.

[60] MOSEK ApS. Mosek modeling cookbook. 2020.

[61] Mark Schmidt. Least squares optimization with l1-norm regularization. *CS542B Project Report*, 504:195–221, 2005.

[62] Barbara Kaltenbacher, Andreas Neubauer, and Otmar Scherzer. *Iterative regularization methods for nonlinear ill-posed problems*, volume 6. Walter de Gruyter, 2008.

[63] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. Efficient l˜ 1 regularized logistic regression. In *Aaai*, volume 6, pages 401–408, 2006.

[64] Sofia Visa, Brian Ramsay, Anca L Ralescu, and Esther Van Der Knaap. Confusion matrix-based feature selection. *MAICS*, 710:120–127, 2011.

[65] Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. Consistent binary classification with generalized performance metrics. In *Advances in Neural Information Processing Systems*, pages 2744–2752, 2014.

[66] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[67] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[68] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[69] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[70] Hiroshi Konno and Rei Yamamoto. Choosing the best set of variables in regression analysis using integer programming. *Journal of Global Optimization*, 44(2):273–282, 2009.

[71] Dheeru Dua and Casey Graff. Uci machine learning repository, 2017.