Constrained Logistic Regression for Discriminative Pattern Mining

Rajul Anand and Chandan K. Reddy

Department of Computer Science, Wayne State University, Detroit, MI, USA

Abstract. Analyzing differences in multivariate datasets is a challenging problem. This topic was earlier studied by finding changes in the distribution differences either in the form of patterns representing conjunction of attribute value pairs or univariate statistical analysis for each attribute in order to highlight the differences. All such methods focus only on change in attributes in some form and do not implicitly consider the class labels associated with the data. In this paper, we pose the difference in distribution in a supervised scenario where the change in the data distribution is measured in terms of the change in the corresponding classification boundary. We propose a new constrained logistic regression model to measure such a difference between multivariate data distributions based on the predictive models induced on them. Using our constrained models, we measure the difference in the data distributions using the changes in the classification boundary of these models. We demonstrate the advantages of the proposed work over other methods available in the literature using both synthetic and real-world datasets.

Keywords: Logistic regression, constrained learning, discriminative pattern mining, change detection.

1 Introduction

In many real-world applications, it is often crucial to quantitatively characterize the differences across multiple subgroups of complex data. Consider the following motivating example from the biomedical domain: Healthcare experts analyze cancer data containing various attributes describing the patients and their treatment. These experts are interested in understanding the difference in survival behavior of the patients belonging to different racial groups (Caucasian-American and African-American) and in measuring this difference across various geographical locations. Such survival behavior distributions of these two racial groups of cancer/non-cancer patients are similar in one location but are completely different in other locations. The experts would like to simultaneously (i) model the cancer patients in each location and (ii) quantify the differences in the racial groups across various locations. The problem goes one step further: the eventual goal is to rank the locations based on the differences in the cancer cases of the two racial groups. In other words, the experts want to find the locations where the difference in the predictive (cancer) models for the two racial groups is higher and the locations where such difference is negligible. Depending on such information,

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part I, LNAI 6911, pp. 92–107, 2011.

more health care initiatives will be organized in certain locations to reduce the racial discriminations in cancer patients [22].

In this problem, the main objective is not only to classify the cancer and non-cancer patients, but also to identify the discriminations (distribution difference) in the cancer patients across multiple subpopulations (or subgroups) in the data. The traditional solutions for this research problem partially addresses the dissimilarity issue, but fails to provide any comprehensive technique in terms of the prediction models. It is vital to develop an integrated framework that can model the discriminations and simultaneously develop a predictive model.

To handle such problems, the methods for modeling the data should go beyond optimizing a standard prediction metric and should simultaneously identify and model the differences between two multivariate data distributions. Standard predictive models induced on the datasets capture the characteristics of the underlying data distribution to a certain extent. However, the main objective of such models is to accurately predict on the future data (from the same distribution) and will not capture the differences between two multivariate data distributions.

1.1 Existing Methods

To find the changes between multivariate data distributions, we need to understand (i) the kind of changes and (ii) how to detect and model such changes. Prior work had emphasized on measuring change in the dataset using the

- difference in probability distribution between individual attributes [15,18]
- difference in the support level of patterns (attribute-value combinations) [8,4]

We term these works as '*unsupervised distribution change detection*'. These methods do not consider the underlying class distribution and how the class distribution changes between the datasets. Existing methods like contrast set mining, emerging pattern mining discussed in Section 2 provide rules with different support criteria (with statistical significance) within two classes. Contrast sets might provide class wise analysis in terms of patterns which differ in support but cannot quantitatively determine whether the overall class distribution between two datasets is different or to what extent the difference is. *The requirement of discrete data for contrast set among many open issues identified in [27] needs to be addressed as well.*

In the case of univariate data, methods such as KolmogorovSmirnov (KS) test [18] will provide information about whether two samples come from same distribution or not. In the multivariate case, an approach to take maximum KS test statistic among all possible orderings can provide some vital information, but again it's univariate analysis extended to multivariate data. Also the number of possible ordering increases exponentially with higher dimensions making the test statistic computationally expensive. The popular KL-divergence [15] also known as relative entropy does provide a change in distribution although non-symmetric in nature ($KL(A\|B) \neq KL(B\|A)$) and purely data oriented approach. Thus, all these methods provide some kind of information about patterns or test statistic. However, in this work, we are interested in finding whether the available data with respect to a class distribution require different classification model or not.

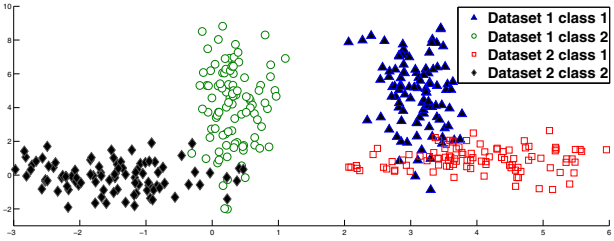
Our approach is to consider the change between datasets as the change in underlying class distributions. Our Supervised Distribution Difference (SDD) measure defined in Sec. 3.2 aims to detect the change in the classification criteria. To understand the kind of distribution changes we are supposedly trying to find can be illustrated using an example. Figure 1(a) visualizes two binary datasets. Performing any univariate or multivariate distribution difference analysis will give us the conclusion that these two datasets are "different" or provide us with some rules which differ in support (contrast set mining). We agree with such analysis, but only to the extent of considering these two datasets without their class labels. When we consider these two datasets having two classes which need to be separated as much as possible using some classification method. **We conclude that these two datasets are not different in terms of their classification criteria**. A nearly similar Logistic Regression (LR) classifier (or any other linear classification models) can be used to separate classes in both these datasets. Thus, *our criteria of finding distribution change is in terms of change in the classification model*.

1.2 Need for Constrained Models

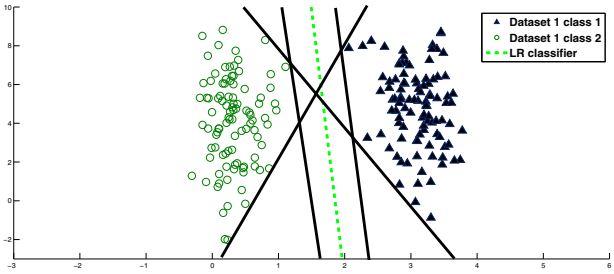
The above discussion clearly states that the differences in multivariate data distributions based on "model" is different from previous "data" based approaches. As such, inducing models on the datasets and finding difference between them can provide us some information about the similarity of the datasets. However, there is one pertinent question related to this discussion. Which model can accurately represent the data? From Fig. 1(b), we can observe that there are many options for classification models within the boundaries indicated by bold lines, representing the maximum possible width of classification margin, whereas the dotted line represent the optimized LR classifier model. Any classifier between these 'bold' boundaries will have the same accuracy (100% in this case). Similarly, it is shown in Fig. 1(c) for the second dataset.

Based on the parameter values, the LR model can be located between any of the class boundaries and yet represent the data accurately. Fig. 1(d) shows the LR model (bold line) obtained using D_1 and D_2 combined together. A constrained LR model obtained for each dataset will be nearly same, since the combined model itself is a reasonable representation of both datasets individually. Thus, the supervised distribution difference will be reported as zero (or close to zero). Whereas, using LR model obtained separately on each dataset (dotted lines) will report significant difference between the two datasets despite each individual LR model being close to the maximum margin classifier in this case. Thereby, just using classification models directly to obtain *SDD* will vary in the results when used for comparing two datasets.

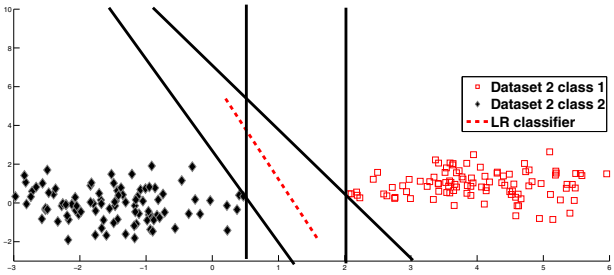
In the case of high-dimensional datasets coupled with non-linearly separable case, the number of potential classifier models required to represent dataset increase rapidly. Thus, the model representing the data for comparison has to be chosen carefully. The basic idea of building constrained models is to provide a baseline which fairly represents both the datasets for comparison. Then, this baseline model can be altered to generate specific model for each dataset. By doing this, we reduce the number of models



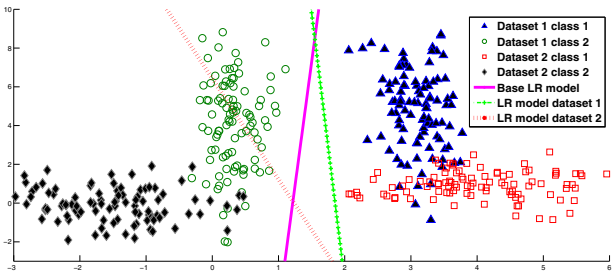
(a)



(b)



(c)



(d)

Fig. 1. (a) Two binary datasets with similar classification criteria (b) Dataset 1 with linear class separators (c) Dataset 2 with linear class separators and (d) Base LR model with Dataset 1 and Dataset 2 model

available for representing the datasets. By placing an accuracy threshold on the selected models, we further reduce the number of such models and simultaneously ensure that the new models are still able to classify each dataset accurately.

In this paper, *we propose a new framework for constrained learning of predictive models that can simultaneously predict and measure the differences between datasets by enforcing some additional constraints in such a way that the induced models are as similar as possible*. Data can be represented by many forms of a predictive model, but not all of these versions perform well in terms of their predictive ability. Each predictive modeling algorithm will heuristically, geometrically, or probabilistically optimize a specific criterion and obtains an optimal model in the model space. There are other models in the model space that are also optimal or close to the optimal model in terms of the specific performance metric (such as accuracy or error rate). Each of these models will be different but yet will be a good representation of the data as long as its predictive accuracy is close to the prediction accuracy of the most optimal model induced. In our approach, we search for two such models corresponding to the two datasets under the constraint that they must be as similar as possible. The distance between these two models can then be used to quantify the difference between the underlying data distributions. Such constrained model building is extensively studied in the unsupervised scenarios [3] and is relatively unexplored in the supervised cases. We chose to develop our framework using LR models due to their popularity, simplicity, and interpret ability which are critical factors for the problem that we are dealing with in this paper.

The rest of the paper is organized as follows: Section 2 discusses the previous works related to the problem described. Section 3 introduces the notations and concepts useful for understanding the proposed algorithm. Section 4 introduces the proposed constrained LR framework for mining distribution changes. The experimental results on both synthetic and real-world datasets are presented in Section 5. And finally, Section 6 concludes our discussion.

2 Related Work

In this section, we describe some of the related topics available in the literature and highlight some of the primary contributions of our work.

(1) Dataset Distribution Differences - Despite the importance of the problem, only a small amount of work is available in describing the differences between two data distributions. Earlier approaches for measuring the deviation between two datasets used simple data statistics after decomposing the feature space into smaller regions using tree based models [22,12]. However, the final result obtained is a data-dependent measure and do not give any understanding about the features responsible for measuring that difference. One of the main drawbacks of such an approach is that they construct a representation that is independent of the other dataset thus making it hard for any sort of comparison. On the contrary, if we incorporate the knowledge of the other class while building models for both the subgroups, they provide more information about the similarities and dissimilarities in the distributions. This is the basic idea of our approach. Some other statistical and probabilistic approaches [25] measure the differences in the data distributions in an unsupervised setting without the use of class labels.

(2) Discriminative Pattern mining - Majority of pattern based mining for different, unusual statistical characteristics [19] of the data fall into the categories of contrast set mining [4,14], emerging pattern mining [8], discriminative pattern mining [10,21] and sub-group discovery [11,16]. Applying most of these methods on a given dataset with two subgroups will only give us the difference in terms of the attribute-value pair combinations (or patterns) without any quantitative measures, i.e. difference of class distribution within a small space of the data and does not provide a global view of the overall difference. In essence, though these approaches attempt to capture statistically significant rules that define the differences, they do not measure the data distribution differences and also do not provide any classification model. The above *pattern mining algorithms do not take into account the change in distribution of class labels*, instead they *define the difference in terms of change in attribute value combinations* only.

(3) Change Detection and Mining - There had been some works on change detection [17] and change mining [26,24] algorithms which typically assume that some previous knowledge about the data is known and measure the change of the new model from a data stream. The rules that are not same in the two models are used to indicate changes in the dataset. These methods assume that we have a particular model/data at a given snapshot and then measure the changes for the new snapshot. The data at the new snapshot will typically have some correlation with the previous snapshot in order to find any semantic relations in the changes detected.

(4) Multi-task Learning and Transfer Learning - The other seemingly related family of methods proposed in the machine learning community is transfer learning [7,23], which adapts a model built on source domain D_S (or distribution) to make a prediction on the target domain D_T. Some variants of transfer learning had been pursued under different names: learning to learn, knowledge transfer, inductive transfer, and multi-task learning. In *multi-task learning* [5], different tasks are learned simultaneously and may benefit from common (often hidden) features benefiting each task. The primary goal of our work is significantly different form transfer learning and multi-task learning, since these methods do not aim to quantify the difference in the data distributions and they are primarily aimed at improving the performance on a specific target domain. These transfer learning tasks look for commonality between the features to enable knowledge transfer or assume inherent distribution difference to benefit the target task.

2.1 Our Contributions

The major distinction of our work compared to the above mentioned methods is that none of the existing methods explore the distribution difference based on a *'model'* built on the data. *The primary focus of the research available in the literature for computing the difference between two data distributions had been 'data-based', whereas, our method is strictly 'model-based'*. In other words, all of the existing methods utilize the data to measure the differences in the distributions. On the contrary, our method computes the difference using constrained predictive models induced on the data. Such constrained models have the potential to simultaneously model the data and compare multiple data distributions. Hence, a systematic way to build a continuum of predictive models is developed in such a manner that the models for the corresponding two groups

are at the extremes of the continuum and the model corresponding to the original data is lying somewhere on this continuum. *It should be highlighted that we compute the distance between two datasets from the models alone; without referring back to the original data*. The major contributions of this work are:

- Develop a measure of the distance between two data distributions using the difference between predictive models without referring back to the original data.
- Develop a constrained version of logistic regression algorithm that can capture the differences in data distributions.
- Experimental justification that the results from the proposed algorithm quantitatively capture the differences in data distributions.

3 Preliminaries

The notations used in this paper are described in Table 1. In this section, we will also describe some of the basic concepts of the Logistic Regression and explain the notion of supervised distribution difference.

Table 1. Notations used in this paper

Notation	*Description*
D_i	i^{th} dataset
F_i	i^{th} dataset classification boundary
C	Regularization factor
L	Objective function
w_k	k^{th} component of weight vector w
W_j	j^{th} weight vector
$diag(v)$	Diagonal matrix of vector v
s^N	Modified Newton N^{th} step s
Z	Scaling matrix
H	Hessian Matrix
J^v	Jacobian matrix of $\|v\|$
ϵ	Constraint on weight values
eps	Very small value (1e-6)

3.1 Logistic Regression

In LR model, a binary classification problem is expressed by logit function which is a linear combination of the attributes [13]. This logit function is also considered as the log-odds of the class probabilities given an instance. Let us denote an example by $\boldsymbol{x}$ and its k^{th} feature as x_k. If each example is labeled either $+1$ or -1, and there are l number of features in each example, the logit function can be written as follows:

$$\log \frac{\Pr\left(y=+1|\boldsymbol{x}\right)}{\Pr\left(y=-1|\boldsymbol{x}\right)} = \sum_{k=0}^{l} w_k x_k = z \tag{1}$$

Here, $x_0 = 1$ is an additional feature called 'bias', and w_0 is the corresponding 'bias weight'. From Eq. (1), we have

$$\Pr(y = +1|\boldsymbol{x}) = \frac{e^z}{1+e^z} = g(z) \tag{2}$$

where, $g(z) = \frac{1}{1+e^{-z}}$. Let $(\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n})$ denotes a set of training examples and $(y_1, y_2, ..., y_n)$ be the corresponding labels. x_{ik} is the k^{th} feature of the i^{th} sample. The joint distribution of the probabilities of class labels of all the n examples is:

$$\Pr(y = y_1|\boldsymbol{x_1}) \Pr(y = y_2|\boldsymbol{x_2}) ... \Pr(y = y_n|\boldsymbol{x_n}) = \prod_{i=1}^{n} \Pr(y = y_i|\boldsymbol{x_i}) \tag{3}$$

LR will learn weights by maximizing the log-likelihood of Eq. (3):

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} \log \Pr(y = y_i|\boldsymbol{x_i}) = \sum_{i=1}^{n} \log g(y_i z_i) \tag{4}$$

where $z_i = \sum_{k=0}^{l} w_k x_{ik}$. To maximize Eq. (4), Newton's method which iteratively updates the weights using the following update equation is applied:

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \left[\frac{\partial^2 L}{\partial \boldsymbol{w} \partial \boldsymbol{w}}\right]^{-1} \frac{\partial L}{\partial \boldsymbol{w}} \tag{5}$$

$$\frac{\partial L}{\partial w_k} = \frac{\partial}{\partial w_k}\left(\sum_{i=1}^{n} \log g(y_i z_i)\right) = \sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i) \tag{6}$$

$$\frac{\partial^2 L}{\partial w_j \partial w_k} = -\sum_{i=1}^{n} x_{ij} x_{ik} g(y_i z_i) g(-y_i z_i) \tag{7}$$

To reduce higher estimation of parameters and to reduce over-fitting, a regularization term is added to objective function. By adding the squared L_2 norm and negating Eq. (4), the problem is converted to a minimization problem as shown in the following objective function:

$$L = -\sum_{i=1}^{n} \log g(y_i z_i) + \frac{C}{2} \sum_{k=1}^{l} w_k^2 \tag{8}$$

$$\frac{\partial L}{\partial w_k} = -\sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i) + C w_k \tag{9}$$

$$\frac{\partial^2 L}{\partial w_k \partial w_k} = -\sum_{i=1}^{n} x_{ik}^2 g(-y_i z_i) + C \tag{10}$$

3.2 Supervised Distribution Difference

Let D_1, D_2 be two datasets having the same number of features and the curve F_1 and F_2 represents the decision boundary for the dataset D_1 and D_2 correspondingly. D represents the combined dataset ($D_1 \cup D_2$) and F is the decision boundary for the combined dataset. For LR model, these boundaries are defined as a linear combination of attributes resulting in a linear decision boundary. We induce constrained LR models for D_1, D_2 which are as close as possible to that of D and yet have significant accuracy for D_1, D_2 respectively. In other words, F_1 and F_2 having minimum angular distance from F. Since, *there exists many such decision boundaries, we optimize for minimum angular distance from F that has higher accuracy.* Supervised Distribution Difference (SDD) is defined as the change in the classification criteria in terms of measuring the deviation in classification boundary while classifying as accurately as possible.

Definition 1. *Let w^A and w^B be the weight vectors corresponding to the constrained LR models for D_1 and D_2, then SDD is defined as follows:*

$$SDD(\boldsymbol{w^A}, \boldsymbol{w^B}) = \sqrt{\sum_k (w_k^A - w_k^B)^2} \tag{11}$$

4 Proposed Algorithm

We will now develop a constrained LR model which can measure the supervised distribution difference between multivariate datasets. Figure 2 shows the overall framework of the proposed algorithm. We start by building a LR model (using Eq.(8)) for the combined dataset D and the weight vector obtained for this base model is denoted by R. The regularization factor C for D is obtained using the best performance for 10-fold cross validation (CV) and then the complete model is obtained using the best value of C. Similarly, LR models on datasets D_1 and D_2 are also obtained. For datasets D_1 and D_2, the CV accuracy for the best C is denoted by Acc for each dataset. The best value of C obtained for each dataset is used while building the constrained model. After all the required input parameters are obtained, constrained LR models are separately learnt individually for the datasets D_1 and D_2 satisfying the following constraint: the weight vector of these new constrained models must be close to that of R (should not deviate much from R). To enforce this constraint, *we change the underlying implementation of LR model* to satisfy the following constraints:

$$|R_k - w_k| \leq \epsilon \tag{12}$$

where ϵ is the deviation we allow from individual weight vectors of model obtained from D. The upper and lower bound for each individual component of the weight vectors is obtained from above equation. To solve this problem, we now use constrained optimization algorithm in the implementation of constrained LR models.

The first derivative while obtaining LR model (Eq. (9)) is set to zero. In our model, a scaled modified Newton step replaces the unconstrained Newton step [6]. The scaled

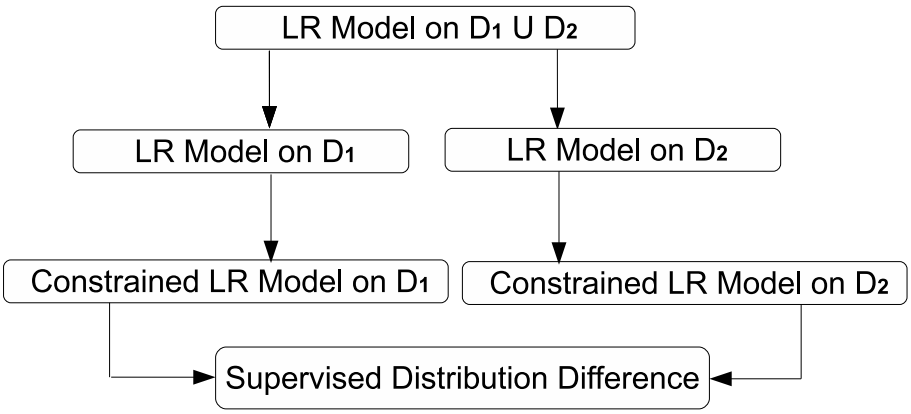


Fig. 2. Illustration of our approach to obtain Supervised Distribution Difference between two multivariate datasets

modified Newton step arises from examining the Kuhn-Tucker necessary conditions for Equations (8) and (12).

$$(Z(w))^{-2}\frac{\partial L}{\partial w} = 0 \tag{13}$$

Thus, we have an extra term $(Z(w))^{-2}$ multiplied to the first partial derivative of the optimization problem(L). This term can be defined as follows:

$$Z(w) = diag(|v(w)|^{-\frac{1}{2}}) \tag{14}$$

The underlying term $v(w)$ is defined below for $1 \leq i \leq k$

$$v_i = w_i - (R_i + \epsilon) \text{ if } \tfrac{\partial L_i(w)}{\partial w} < 0 \text{ and } (R_i + \epsilon) < \infty$$

$$v_i = w_i - (R_i - \epsilon) \text{ if } \tfrac{\partial L_i(w)}{\partial w} \geq 0 \text{ and } (R_i - \epsilon) > -\infty$$

Thus, we can see that the epsilon constraint is used in modifying the first partial derivative of L. The scaled modified Newton step for the nonlinear system of equations given by Eq. (13) is defined as the solution to the linear system

$$\hat{A}Zs^N = -\frac{\hat{\partial L}}{\partial w} \tag{15}$$

$$\frac{\hat{\partial L}}{\partial w} = Z^{-1}\frac{\partial L}{\partial w} \tag{16}$$

$$\hat{A} = Z^{-2}H + diag(\frac{\partial L}{\partial w})J^v \tag{17}$$

The reflections are used to increase the step size and a single reflection step is defined as follows. Given a step η that intersects a bound constraint, consider the first bound

constraint crossed by η; assume it is the i^{th} bound constraint (either the i^{th} upper or lower bound). Then the reflection step $\eta^R = \eta$ except in the i^{th} component, where $\eta_i^R = \eta_i$. In summary, our approach can be termed as constrained minimization with box constraints. It is different from LR which essentially performs an unconstrained optimization. After the constrained models for the two datasets D_1 and D_2 are induced, we can capture the model distance by the Eq. (11). Algorithm 1 outlines our approach for generating constrained LR models.

Algorithm 1. Constrained Logistic Regression

Input: Data (D), Accuracy of LR model on D (Acc), Threshold for accuracy loss (τ), Threshold for deviation (ϵ), Unconstrained LR model on D (R)
Output: Final model weight vector (W)

1: maxAcc $\leftarrow$ 0, s $\leftarrow$ 0, $modelFound \leftarrow false$
2: **while** modelFound $\neq$ true **do**
3: **for** $a \leftarrow$ s+0.01 to s+0.05 step 0.01 **do**
4: $\epsilon \leftarrow a \times$ R
5: $lower \leftarrow R - \epsilon$
6: $upper \leftarrow R + \epsilon$
7: i $\leftarrow$ 0
8: $L_i \leftarrow L(\acute{W}_i)$
9: **repeat**
10: $argmin_w$ -lnL(W) to compute $\acute{W_{i+1}}$ with constraints $lower \leq \acute{W_{i+1}} \leq upper$
11: $L_{i+1} \leftarrow L(\acute{W_{i+1}})$
12: $i \leftarrow i + 1$
13: **until** $\frac{L_{i+1} - L_i}{L_i} < eps$
14: **if** (Acc - Acc($\acute{W_{i+1}}$))/ Acc $\leq \tau$ and maxAcc $<$ Acc($\acute{W_{i+1}}$) **then**
15: W $\leftarrow \acute{W_{i+1}}$
16: maxAcc $\leftarrow Acc(\acute{W_{i+1}})$
17: modelFound $\leftarrow$ true
18: **end if**
19: **end for**
20: s $\leftarrow$ s + 0.05
21: **end while**

Most of the input parameters for the constrained LR algorithm are dataset dependent and are obtained before running the algorithm as can be seen in the flowchart in Fig. 2. The only parameter required is τ which is set to 0.15. However, depending on the application domain of the dataset used, this value can be adjusted as it's main purpose is to allow for tolerance by losing some accuracy while comparing datasets. The constraint ϵ is varied systematically using variable a on line 4. This way, we gradually set bounds for weight vector to be obtained (lines 5, 6). The weight vector for the optimization is initialized with uniform weights (line 8). Line 10 employs constrained optimization using bounds provided earlier and terminates when the condition on line 13 is satisfied. The tolerance value eps is set to 1e-6. After the weight vector for a particular constraint is obtained, we would like to see if this model can be considered for representing the

dataset. Line 14 checks whether the accuracy of the current model is within the threshold. It also checks if the accuracy of the current model with previously obtained model and the better one is chosen for further analysis. The best model in the range of 1% to 5% constraint of base weight vector R is selected. If no such model is found within this range, then we gradually increase the constraint range (line 20) until we obtain the desired model. The final weight vector is updated in line 15 and is returned after the completion of full iteration. The convergence proof for the termination of constrained optimization on line 10 is similar to the one given in [6].

5 Experimental Results

We conducted our experiments on five synthetic and five real-world datasets [2]. The binary datasets are represented by triplet (dataset, attributes, instances). The UCI datasets used are (blood, 5, 748), (liver, 6, 345), (diabetes, 8, 768), (gamma, 11, 19020), and (heart, 22, 267). Synthetic datasets used in our work have 500,000 to 1 million tuples.

5.1 Results on Synthetic Datasets

First, a synthetic dataset with 10 attributes is generated using Gaussian distribution with a predefined mean and standard deviation (μ, σ). Here, two datasets D_1, D_2 are created with the same feature space, but the features that are important for class separation are different in both the datasets. These "significant" features are known a priori. Obtaining unconstrained LR models on each of the datasets independently will provide weightage for features, but only the highly significant features can be found using such models (normally the features that are already familiar in the application domain). Identifying the '*differential features*', which we define as features that are more important in one dataset but less in the other dataset, was not possible using unconstrained models. Using our constrained models, we were able to identify the differential features by ranking them in order of high magnitude by calculating the difference between the weight vectors for each feature. Since the ground truth is known in this case, we were able to identify most of the differential features correctly. We repeated our experiments by varying the differential features in both the datasets to remove any bias for a particular experiment.

Table 2 highlights the difference in the weight vectors obtained from one such experiment. The difference is between individual component of the weight vectors for LR and constrained LR model on the two datasets. Bold numbers correspond to the highly differential features obtained from constrained LR based on ground truth. We can notice that LR model does not necessarily produce high absolute scores for these attributes and gives higher absolute scores for other attributes while our method accurately captures the differential features.

Based on the ground truth available to us, we highlighted (in Fig. 3) the significance of features given by LR and constrained LR method based on Table 2. The features are ranked from 10 to 1 where 10 being most highly differential and 1 being least differential. From the figure, it can be observed that LR models were only able to capture features 2, 5 and 1 which are similar to the ground truth. The constrained LR model on the other hand was much closer to the ground truth most of the times.

Table 2. Difference between weight vectors for constrained and unconstrained LR models

LR	*Constrained LR*	*LR*	*Constrained LR*
-3.3732	**-0.8015**	1.2014	**0.4258**
-0.8693	0	0.0641	0.0306
-1.2061	-0.0158	-0.5393	0.1123
-1.6274	0	-3.5901	0
5.0797	**0.9244**	0.7765	0.0455

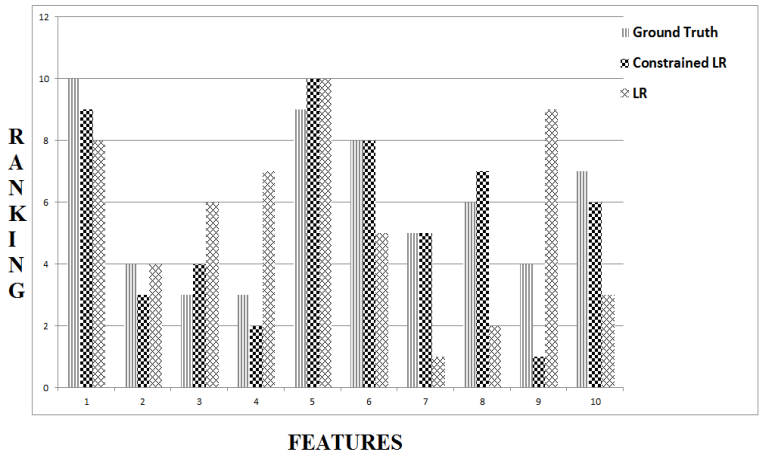


Fig. 3. Feature Ranking vs. Number of Features

5.2 The Comparison of the Distance Measure

Let $NM.F_{num}$ denote a dataset with N million tuples generated by classification function num. After computing the distance between the datasets, the main issue to be addressed is: *how large a distance should be there in order to ensure that the two datasets were generated by different underlying processes?* The technique proposed in [12] answers this question as follows: If we assume that the distribution G of distance values (under the hypothesis that the two datasets are generated by the same process) is known, then we can compute G using bootstrapping technique [9], and we can use standard statistical tests to compute the significance of the distance d between the two datasets. The datasets were generated using the functions F_1, F_2, and F_4 respectively. One of the datasets is constructed by unifying D with a new block of 50,000 instances generated by F_4 where $D = 1M.F1$. $D_1 = D \cup 0.05M.F_4$, $D_2 = 0.5M.F_1$, $D_3 = 1M.F_2$, and $D_4 = 1M.F_4$.

Prior work [12] devised a "data-based" distance measure along with derived a method for measuring statistical significance of the derived distance. The experiments conducted on synthetic datasets are explained in [1]. The distance value computed on these datasets by [12] can be taken as the ground truth and our experiments on these datasets follow the same pattern as that of earlier results. Table 3 highlights that relative ranking

among datasets for distance is same. Note that *the distances are not directly comparable* ([12] and Constrained LR), only ranking can be compared based using the distance computed.

Table 3. The distances of all four datasets by constrained LR and Ganti's method [12]

Dataset	*Dist by [12]*	*Dist by Constrained LR*
D_1	0.0689	0.00579
D_2	0.0022	0.004408
D_3	1.2068	0.022201
D_4	1.4819	0.070124

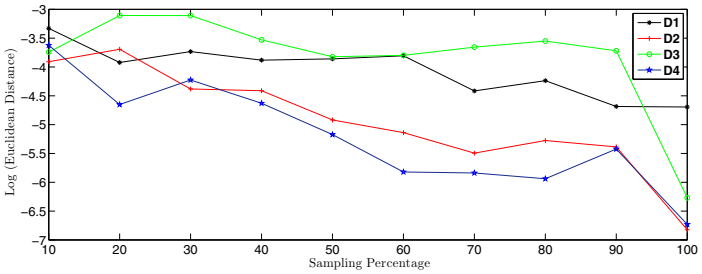


Fig. 4. Log(Euclidean distance) vs. sampling percentage

5.3 The Sensitivity of the Distance Measure

We will first show that the distance measure calculated by our algorithm precisely captures the differences between data distributions. In [20], the authors developed a systematic way of generating datasets with varying the degree of differences in data distributions. To achieve similar goal, we generate datasets exhibiting varying degrees of similarity. We created random subsamples of a given dataset, D of the size p, where p is varied as 10%, 20%, ..., 100%, with a stepsize of 10%. Each subsample is randomly chosen 5 times and model distances calculated are averaged to remove bias in the models. Each of these subsamples is denoted by D_p, where $D_{100} = D$. Now, using the proposed algorithm, we calculated the distance between D and D_p using Algorithm 1. *We expect the calculated distance between D and D_p to decrease as p increases* and to approach zero when $p = 100\%$. Figure 4 shows the result on synthetic datasets used above (Sec. 5.2). These datasets are big and thus there is a significant change in the class distribution even at small sampling levels. However, the distance is still small as expected and decreases monotonically. In Figure 5, we plot the model distances against the sampling size p for real-world datasets. Here, as we can observe that the class distribution is nearly uniform and thus SDD metric does not change much except for the case of less than 10% samples. The constant model distance and sudden drop in SDD after 10% sampling indicate that more than 10% samples of the data resemble class distribution closely since the induced models are nearly similar with low value of SDD. Thus, we can say that our metric captures the class distribution difference accurately.

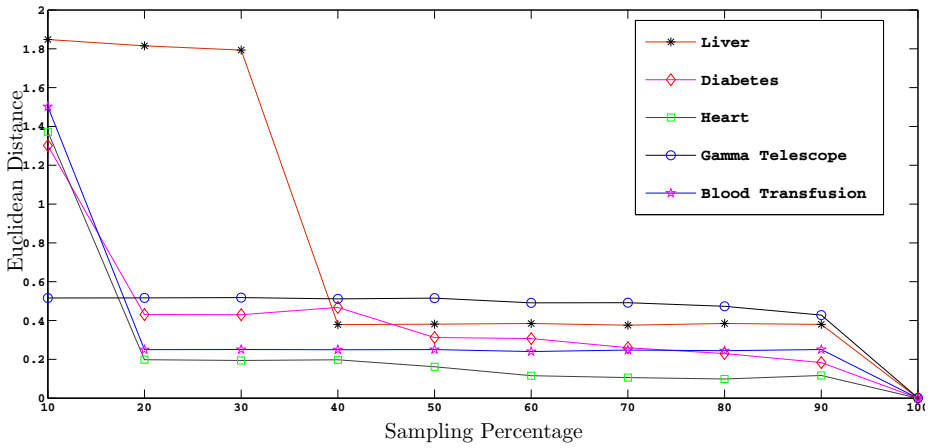


Fig. 5. Euclidean distance vs. sampling percentage

6 Conclusion

Standard predictive models induced on multivariate datasets capture certain characteristics of the underlying data distribution. In this paper, we developed a novel constrained logistic regression framework which produces accurate models of the data and simultaneously measures the difference between two multivariate datasets. These models were built by enforcing additional constraints to the standard logistic regression model. We demonstrated the advantages of the proposed algorithm using both synthetic and real-world datasets. We also showed that the distance between the models obtained from proposed method accurately captures the distance between the original multivariate data distributions.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Database mining: A performance perspective. IEEE Trans. Knowledge Data Engrg. 5(6), 914–925 (1993)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://archive.ics.uci.edu/ml/
3. Basu, S., Davidson, I., Wagstaff, K.L.: Constrained Clustering: Advances in Algorithms, Theory, and Applications. CRC Press, Boca Raton (2008)
4. Bay, S.D., Pazzani, M.J.: Detecting group differences: Mining contrast sets. Data Mining and Knowledge Discovery 5(3), 213–246 (2001)
5. Caruana, R.: Multitask learning. Machine Learning 28(1), 41–75 (1997)
6. Coleman, T.F., Li, Y.: An interior trust region approach for nonlinear minimizations subject to bounds. Technical Report TR 93-1342 (1993)
7. Dai, W., Yang, Q., Xue, G., Yu, Y.: Boosting for transfer learning. In: ICML 2007: Proceedings of the 24th International Conference on Machine Learning, pp. 193–200 (2007)
8. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 43–52 (1999)

9. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman and Hall, London (1993)
10. Fang, G., Pandey, G., Wang, W., Gupta, M., Steinbach, M., Kumar, V.: Mining low-support discriminative patterns from dense and high-dimensional data. IEEE Transactions on Knowledge and Data Engineering (2011)
11. Gamberger, D., Lavrac, N.: Expert-guided subgroup discovery: methodology and application. Journal of Artificial Intelligence Research 17(1), 501–527 (2002)
12. Ganti, V., Gehrke, J., Ramakrishnan, R., Loh, W.: A framework for measuring differences in data characteristics. J. Comput. Syst. Sci. 64(3), 542–578 (2002)
13. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, Heidelberg (2009)
14. Hilderman, R.J., Peckham, T.: A statistically sound alternative approach to mining contrast sets. In: Proceedings of the 4th Australasian Data Mining Conference (AusDM), pp. 157–172 (2005)
15. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. 22(1), 79–86 (1951)
16. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup discovery with cn2-sd. Journal of Machine Learning Research 5, 153–188 (2004)
17. Liu, B., Hsu, W., Han, H.S., Xia, Y.: Mining changes for real-life applications. In: Data Warehousing and Knowledge Discovery, Second International Conference (DaWaK) Proceedings, pp. 337–346 (2000)
18. Massey, F.J.: The kolmogorov-smirnov test for goodness of fit. Journal of the American Statistical Association 46(253), 68–78 (1951)
19. Novak, P.K., Lavrac, N., Webb, G.I.: Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. Journal of Machine Learning Research 10, 377–403 (2009)
20. Ntoutsi, I., Kalousis, A., Theodoridis, Y.: A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees. In: SIAM International Conference on Data Mining (SDM), pp. 810–821 (2008)
21. Odibat, O., Reddy, C.K., Giroux, C.N.: Differential biclustering for gene expression analysis. In: Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology (BCB), pp. 275–284 (2010)
22. Palit, I., Reddy, C.K., Schwartz, K.L.: Differential predictive modeling for racial disparities in breast cancer. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 239–245 (2009)
23. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2010)
24. Pekerskaya, I., Pei, J., Wang, K.: Mining changing regions from access-constrained snapshots: a cluster-embedded decision tree approach. Journal of Intelligent Information Systems 27(3), 215–242 (2006)
25. Wang, H., Pei, J.: A random method for quantifying changing distributions in data streams. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 684–691. Springer, Heidelberg (2005)
26. Wang, K., Zhou, S., Fu, A.W.C., Yu, J.X.: Mining changes of classification by correspondence tracing. In: Proceedings of the Third SIAM International Conference on Data Mining (SDM), pp. 95–106 (2003)
27. Webb, G.I., Butler, S., Newlands, D.: On detecting differences between groups. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 256–265 (2003)