# Optimizing logistic regression coefficients for discrimination and calibration using estimation of distribution algorithms

**V. Robles · C. Bielza · P. Larrañaga · S. González · L. Ohno-Machado**

**Abstract** Logistic regression is a simple and efficient supervised learning algorithm for estimating the probability of an outcome or class variable. In spite of its simplicity, logistic regression has shown very good performance in a range of fields. It is widely accepted in a range of fields because its results are easy to interpret. Fitting the logistic regression model usually involves using the principle of maximum likelihood. The Newton–Raphson algorithm is the most common numerical approach for obtaining the coefficients maximizing the likelihood of the data.

This work presents a novel approach for fitting the logistic regression model based on estimation of distribution algorithms (EDAs), a tool for evolutionary computation. EDAs are suitable not only for maximizing the likelihood, but also for maximizing the area under the receiver operating characteristic curve (AUC).

Thus, we tackle the logistic regression problem from a double perspective: likelihood-based to calibrate the model and AUC-based to discriminate between the different classes. Under these two objectives of calibration and discrimination, the Pareto front can be obtained in our EDA framework. These fronts are compared with those yielded by a multiobjective EDA recently introduced in the literature.

**Keywords** Logistic regression · Evolutionary algorithms · Estimation of distribution algorithms · Calibration and discrimination

**Mathematics Subject Classification (2000)** 62J12 · 90C59 · 90C29

## 1 Introduction

Logistic regression modeling is employed in many fields (Hosmer and Lemeshow 2000). The outcome variable is binary, while the explanatory variables are of any type, lending great flexibility to this approach. Experimental results have shown that logistic regression can perform at least as well as a more complex classifier in a variety of data sets (Baumgartner et al. 2004; Kiang 2003), and this approach compared favorably with many supervised machine learning techniques: $k$-nearest neighbors, discriminant analysis, neural networks, support vector machines, and decision trees.

As in other simpler regression models, logistic regression applies the maximum likelihood principle for parameter estimation. The model that linearly relates the log of the odds of the response and the explanatory variables gives rise to complex nonlinear likelihood equations in the unknown parameters (Ryan 1997). Therefore, special numerical methods for their solution are required.

The so-called Newton–Raphson method is commonly used to solve the likelihood equations numerically. Although the method requires inverting a matrix and exhibits some dependence on the initial starting conditions for convergence to be guaranteed, it shows good performance overall (Minka 2003). One of the aims of the current paper is to tackle this maximization problem for parameter estimation by using a recent optimization heuristic called *estimation of distribution algorithms* (EDAs) (Larrañaga and Lozano 2002). In this sense, we contribute towards the (currently poor) application of optimization heuristics in statistic estimation and modelling problems (Winker and Gilli 2004).

EDAs are evolutionary algorithms that are among the best-known stochastic population-based search methods. These algorithms construct an explicit probability model from a set of selected solutions which is then conveniently used to generate new promising solutions in the next iteration of the evolutionary process. Other evolutionary algorithms, like genetic algorithms, have been used in logistic regression but for performing feature subset selection (Vinterbo and Ohno-Machado 1999a; Nakamichi et al. 2004), not for estimating the parameters or investigating the model performance from several points of view. To our knowledge, this is the first exploration on how EDAs can be used in this context.

The search for the parameters, both in a numerical or in an evolutionary way, tries to attain an appropriate model in the sense of maximizing the chances of obtaining the data given the fitted model. The proximity of the true and the observed probability for a given set of observations, usually measured using calibration indices, is an important criterion of a model performance. However, it does not suffice. The logistic regression model outputs the probability of a certain event occurring. This probability can be used to predict the class. For classification, high discriminatory ability to differentiate between the classes is at least as important as calibration. In fact, the recommendation of assessing the model performance by considering both calibration and discrimination has been clearly asserted, e.g., in seminal texts on logistic regression (see Hosmer and Lemeshow 2000, p.163).

Since good calibration does not necessarily mean good discrimination and vice versa, both types of measures should be analyzed in logistic regression models. Therefore, we specifically study the behavior of two model performance measures:

the maximum (log) likelihood (for calibration) and the area under the receiver operating characteristic curve (AUC) (for discrimination).

Among the most outstanding strengths, our new EDA framework can be flexible enough to cope with the parameter estimation when the optimization is based on calibration or on discrimination, or even on other model performance measures like the Brier score or any multi-objective measure. Moreover, the bi-objective space of calibration against discrimination can be explored to depict the relationship between both objectives, allowing us to estimate the Pareto front with the non-dominated points.

The paper is organized as follows. Section 2 reviews the logistic regression model and the derivation of the likelihood equations. Section 3 includes the model performance measures we will use to assess calibration and discrimination. Section 4 deals with different estimation methods of performance when the model faces future (unseen) data. Section 5 describes a method for searching the logistic regression parameters based on EDAs, emphasizing the advantages of this new approach. Section 6 shows the experiments with several data sets and the potentiality of our method. Section 7 highlights the benefits of our new approach. Finally, we discuss in Sect. 8 the conclusions and lines of future research.

## 2 Logistic regression

Logistic regression (Hosmer and Lemeshow 2000) is a standard method to describe the relationship between a response (or dependent) variable which is binary and several explanatory (or predictor) variables called covariates. When it is used for classification purposes, the response variable is the class variable $C$ predicted through covariates $X_1, \ldots, X_k$. In this context, logistic regression becomes a powerful supervised classification paradigm that provides explicit probabilities of classification that can be used to provide class label information. This approach falls into the category of discriminative classifiers, in the sense that they model the probability of the class given the covariates, in contrast to generative classifiers that model the joint probability of the class and the covariates (Ng and Jordan 2001). As opposed to other methods like discriminant analysis, strong assumptions like normal distribution of the covariates given the class are not required. Also, covariates can be given in a quantitative (continuous or discrete) or qualitative scale.

The logistic regression classifier is induced from a (training) data set $\mathcal{D}_N$ containing $N$ independent samples $\mathcal{D}_N = \{(c_j, x_{j1}, \ldots, x_{jk}),\ j = 1, \ldots, N\}$, drawn from the joint probability distribution on $(C, X_1, \ldots, X_k)$. In this paper, we focus on the two category classification problem, although the ideas could be readily extended to the multi-category case. Thus, $C$ can only take 0 and 1 values, where label $c_j = 1$ means that the $j$th input pattern $\mathbf{x}_j = (x_{j1}, \ldots, x_{jk})$ is in the first class (i.e., observation $j$ has the feature that $C$ represents), while $c_j = 0$ means $\mathbf{x}_j$ does not have the feature, and therefore belongs to the other class. The classification model will be used for assigning labels $c_j$ to new instances that are not part of the training set, and therefore are only characterized with the values of the predictor variables.

Let $\pi_{\mathbf{x}}$ denote $P(C = 1|\mathbf{x}) = P(C = 1|X_1 = x_1, \ldots, X_k = x_k)$. Then the *logit* model is defined as:

$$\log \frac{\pi_{\mathbf{x}}}{1 - \pi_{\mathbf{x}}} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k, \tag{1}$$

or equivalently,

$$\pi_{\mathbf{x}} = \frac{e^{(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}}{1 + e^{(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}}, \tag{2}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)$ denotes the vector of regression coefficients including a constant or intercept $\beta_0$.

Therefore, the model specifies $\pi_{\mathbf{x}}$ as the dependent variable to be a function of the predictor variables. Since $C$ is dichotomous, its expected value is $E(C|\mathbf{x}) = \pi_{\mathbf{x}}$, and we search for a relationship between the expected response and the covariates.

Regression coefficients are usually estimated from the data by means of the maximum likelihood estimation method. Given the training data set, the likelihood function is $\mathcal{L}(\boldsymbol{\beta}) = \prod_{j=1}^{N} \pi_{\mathbf{x}_j}^{c_j} (1 - \pi_{\mathbf{x}_j})^{1-c_j}$, where $\pi_{\mathbf{x}_j}$ is stated in (2). Maximum likelihood estimators (MLE) $\widehat{\beta}_i$ are obtained by maximizing $\mathcal{L}$ with respect to $\boldsymbol{\beta}$, or equivalently, by maximizing $\log \mathcal{L}$ with respect to $\boldsymbol{\beta}$.

We have that

$$\begin{aligned} \log \mathcal{L}(\boldsymbol{\beta}) &= \sum_{j=1}^{N} \left( c_j \log \pi_{\mathbf{x}_j} + (1 - c_j) \log(1 - \pi_{\mathbf{x}_j}) \right) \\ &= \sum_{j=1}^{N} c_j \log \frac{\pi_{\mathbf{x}_j}}{1 - \pi_{\mathbf{x}_j}} + \sum_{j=1}^{N} \log(1 - \pi_{\mathbf{x}_j}) \end{aligned}$$

and using (1) and (2),

$$\log \mathcal{L}(\boldsymbol{\beta}) = \sum_{j=1}^{N} c_j \left( \beta_0 + \beta_1 x_{j1} + \cdots + \beta_k x_{jk} \right) - \sum_{j=1}^{N} \log\left(1 + e^{(\beta_0 + \beta_1 x_{j1} + \cdots + \beta_k x_{jk})}\right).$$

Thus, the following system of $k + 1$ equations and $k + 1$ parameters—called the likelihood equations—has to be solved:

$$\begin{aligned} \frac{\partial \log \mathcal{L}(\boldsymbol{\beta})}{\partial \beta_0} &= \sum_{j=1}^{N} c_j - \sum_{j=1}^{N} \frac{e^{(\beta_0 + \beta_1 x_{j1} + \cdots + \beta_k x_{jk})}}{1 + e^{(\beta_0 + \beta_1 x_{j1} + \cdots + \beta_k x_{jk})}} = 0, \\ &\vdots \\ \frac{\partial \log \mathcal{L}(\boldsymbol{\beta})}{\partial \beta_k} &= \sum_{j=1}^{N} c_j x_{jk} - \sum_{j=1}^{N} x_{jk} \frac{e^{(\beta_0 + \cdots + \beta_k x_{jk})}}{1 + e^{(\beta_0 + \cdots + \beta_k x_{jk})}} = 0. \end{aligned}$$

Unfortunately, there is no analytical solution of these *nonlinear* equations for $\widehat{\beta}_i$, but we may resort to using numerical optimization methods. Among these, a general

choice is the Newton–Raphson numerical procedure (Thisted 1988) in which each iteration provides an updating formula given by

$$\widehat{\boldsymbol{\beta}}^{\text{new}} = \widehat{\boldsymbol{\beta}}^{\text{old}} + (\mathbf{X}^{\mathsf{t}}\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{t}}(\mathbf{c} - \widehat{\boldsymbol{\pi}}),$$

where $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$, $\mathbf{c}$ denotes the vector of response values $c_j$ $(j = 1, \ldots, N)$, $\mathbf{X}$ denotes an $N \times (k+1)$ matrix with each row given by $(1, \mathbf{x}_j)$, $\hat{\boldsymbol{\pi}}$ denotes the vector of estimated values at that iteration, i.e., its $j$th-component is $\hat{\pi}_{\mathbf{x}_j} = [1 + \mathrm{e}^{-(\hat{\beta}_0^{\text{old}} + \hat{\beta}_1^{\text{old}} x_{j1} + \cdots + \hat{\beta}_k^{\text{old}} x_{jk})}]^{-1}$, $j = 1, \ldots, N$, and $\mathbf{W}$ denotes a diagonal matrix with elements $\hat{\pi}_{\mathbf{x}_j}(1 - \hat{\pi}_{\mathbf{x}_j})$. This formula is used until a convergence criterion is achieved. Common convergence criteria consist of the detection of negligible changes in the log likelihood function, in the parameter estimates, or in the predictions. No single criterion appears superior to the others. In regard to starting estimates, the ones obtained using discriminant analysis turn out to be good and may speed up the convergence (Ryan 1997).

Minka (2003) compares eight different numerical algorithms for computing the MLEs in terms of computational complexity (total floating-point operations) and performance (log likelihood value achieved). The Newton–Raphson algorithm shows excellent performance and a rapid convergence rate.

Therefore, it may seem hard to design a better algorithm to approximate the $\widehat{\beta}_i$ MLEs for logistic regression. Since our estimation problem is an optimization problem, a promising alternative would be to try some optimization heuristics, which surprisingly have not been very commonly used in statistical estimation and modelling problems (Winker and Gilli 2004). We introduce here the estimation of distribution algorithms (EDAs) that, to the best of our knowledge, have never been used in this context. As far as evolutionary algorithms are concerned, we only know of a genetic algorithm employed to *select* variables in logistic regression (Vinterbo and Ohno-Machado 1999a; Nakamichi et al. 2004), not in the estimation problem.

The $\log \mathcal{L}$ function guides the search of $\widehat{\beta}_i$'s, trying to produce a model that fits, i.e., the observed sample values of the response variable agree with the values predicted by the model (or fitted values). This goodness-of-fit informs us about the effectiveness of the model in describing the response variable. A good fit provides a *calibrated* model.

However, when classification is a goal of the modeling and estimated probabilities are used to predict the class membership, the *discrimination* between the different classes may not be accurate even if the model fits the data well. Situations in which the logistic regression fits the data properly but yields poor classification have been reported elsewhere (see Hosmer and Lemeshow 2000, p.156 and p.163).

Therefore, both calibration and discrimination measures should be analyzed in logistic regression models. Among other interesting findings, we show that EDAs are flexible enough to cope with the estimation of $\beta$'s when the optimization is based on any of those measures.

## 3 AUC as a model performance measure in logistic regression

As noted above, good calibration does not necessarily entail good discrimination and vice versa. A seminal text on logistic regression claims that "model performance

should be assessed by considering both calibration and discrimination" (see Hosmer and Lemeshow 2000, p.163) and this is also our viewpoint.

One solution taken in the literature is to start off with a model that has good discrimination and then adjust its calibration, what is called model *recalibration* Harrell et al. (1984). In logistic regression, Harrell et al. (1996) propose shrinkage to recalibrate. However, this increases calibration only when the testing set is relatively large (Steyerberg et al. 2004), which is not always the case in practice. Vinterbo and Ohno-Machado (1999b) propose to alter the estimated probabilities by applying a discrimination-preserving transformation, which is empirically determined from a linear regression on the points in the calibration plot to move them closer to the identity line (the ideal calibration). A second transformation is still needed to keep the resulting predictions within 0 and 1. This method suffers from a number of limitations: It is not useful when the model gives the same estimates for every input, when the calibration plot points are spread in an alternating pattern around the identity line, or when many estimates are close to either 0 or 1.

The approach taken in this paper is different, since we search for a model that *directly* provides good overall performance by optimizing with respect to different performance measures with an evolutionary algorithm.

When classifying instances, since probabilistic model outputs are continuous, we have to transform those outputs into binary outcomes providing the predicted class. For a given cutpoint or threshold $t \in [0, 1]$, we can decide that the predicted class of an instance $j$ is $\hat{c}_j = 1$ if its estimated probability $\hat{\pi}_j \geq t$ and it is $\hat{c}_j = 0$ otherwise. Each $t$ results in a sensitivity and specificity pair. The plot of the values of 1-specificity (false positive rate) against sensitivity (true positive rate) over all possible cutpoints is the receiver operating characteristic (ROC) curve (see e.g. Pepe 2003).

ROC curves describe the predictive behavior of a classifier independently of class distributions or error costs. Its use as a metric for comparing algorithms rather than using the classification error rate has been widely justified (see, e.g., Provost et al. 1998).

The ROC curve is commonly summarized by the area under the curve (AUC) (Hilden 1991; Bradley 1997). AUC ranges from 0 to 1, where perfect discrimination between both classes corresponds to an area of 1 (a horizontal line through the point (1,1)) and random classification corresponds to an area of 0.5 (the identity line).

There are many ways to compute the AUC (Fawcett 2003). We can use the trapezoidal rule after connecting the ROC curve points by straight lines (Horton et al. 2004), which is a poor computational strategy, known to underestimate the AUC if the number of points is limited (Hanley and McNeil 1982). AUC has an important statistical property: it is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. Thus, an intuitive way to proceed is by computing the *concordance index* or *c-index*, as follows. Let us create all the possible pairs of observations such that its first element has $c_j = 1$ and the second element has $c_j = 0$. Then, the $c$-index is the proportion of the time that the observation with $c_j = 1$ has the higher of the two probabilities, with ties resolved by tossing an unbiased coin (Harrell et al. 1996; Hanley and McNeil 1982). Without ties, the $c$-index is the AUC and is equivalent to

the Mann–Whitney $U$ statistic, which is another form of the Wilcoxon rank-sum test (Hajek et al. 1999). Under limited information—like having only a single point of the ROC curve—approximations to the AUC can be computed (van den Hout 2003). Other approximations, both parametric and non-parametric, have been proposed in the large literature on the subject, see a review in Lasko et al. (2005).

## 4 Model performance assessment

In the previous sections, we have introduced two performance measures of a model: $\log \mathcal{L}$ and AUC. To assess this performance that allows us to compare a model to other candidate models, its discrimination/calibration ability has to be checked on test data that is different from training data. By so doing, we estimate the generalization performance of our model.

Let us start our discussion by taking the error rate as the performance metric. After designing our logistic regression classifier, its (classification) error rate when using the model for classifying unseen (new) instances has to be estimated, or at least its *expected* error rate. A low error rate usually corresponds to high accuracy. When comparing error estimators, that should be as close as possible to the true error, one has to consider their bias and variance, since the composition of both defines the mean squared error. Unbiasedness (or at least a low degree of bias) and small variance are desirable. A large variance is of particular concern even with unbiasedness, since the estimate corresponding to a given sample can be often far from the actual error rate. Among these estimators, the *resubstitution* estimator, where the error is directly computed on the sample data itself, is simply a very fast but usually optimistic (i.e., low-biased) estimator of the true error. *Holdout* error estimation, with a training set for the modeling and a test set for testing the classifier, requires large sample sizes.

However, *cross-validation* error estimation is the most widely used method and provides a nearly unbiased estimate of the future error rate although perhaps at the expense of some variance. In *k-fold cross-validation* (Stone 1974), the data set is randomly partitioned into $k$ *folds* of approximately equal size. Each time $t \in \{1, 2, \ldots, k\}$ a fold is left out of the modeling process and used as a testing set. The cross-validation estimate of the error is computed by averaging the resulting error estimations from all folds. A 10-fold cross-validation will be the chosen method in this paper.

Any—more general—performance measures as $\log \mathcal{L}$ and AUC may be estimated from the data in a way similar to the one used in estimating the error rate.

## 5 Estimating the logistic regression coefficients with estimation of distribution algorithms

As stated in Sect. 2, the likelihood equations to be solved in order to obtain the values of parameters $\beta_0, \ldots, \beta_k$ cannot be resolved analytically (Hosmer and Lemeshow 2000). Several numerical algorithms for computing the MLE of the regression coefficients have been proposed in the literature (Minka 2003). However, the solutions provided by these procedures are likely to be improved in some circumstances.

In this section we present an introduction to estimation of distribution algorithms (EDAs), a recent population-based stochastic optimization heuristic (Larrañaga and Lozano 2002). We also describe a method for searching the optimal values of the logistic regression coefficients based on EDAs in continuous domains.

### 5.1 Estimation of distribution algorithms

It is possible to use optimization heuristics as an alternative way for the estimation of regression parameters. These optimization heuristics can be divided into local and population-based search methods. Evolutionary algorithms are among the best-known stochastic population-based search methods. They start from a random population of individuals—each of them representing a possible solution to the optimization problem—and iterate until some pre-defined stopping criterion is satisfied. At every iteration, usually called *generation*, a subset of individuals is selected. By applying some variation operators to the selected set, a new population is created. An example of evolutionary algorithms are *genetic algorithms* (GAs) (Goldberg 1989). The distinguishing feature of GAs is the application of the recombination and mutation operators. As mentioned before, GAs have been used in combination with logistic regression only in the selection of the covariates to be included in the model (Vinterbo and Ohno-Machado 1999a; Nakamichi et al. 2004).

Another class of population-based search methods comprises those algorithms that use probabilistic modeling of the solutions instead of genetic operators. *Estimation of distribution algorithms* (Larrañaga and Lozano 2002; Lozano et al. 2006) are evolutionary algorithms that construct an explicit probability model from a set of selected solutions. This model can capture, by means of probabilistic dependencies, relevant interactions among the variables of the problem. The model can be conveniently used to generate new promising solutions.

Figure 1 shows a pseudo-code for a general EDA approach to optimization. At the beginning $M$ individuals, each of them representing a point of the search space, are generated at random. These $M$ individuals constitute the initial population and are evaluated by means of a fitness function. In a first step of the algorithm, a number $N$ ($N < M$) of individuals are selected according to a selection method. Next, the

---

(i) $D_0 \leftarrow$ Generate $M$ individuals randomly
(ii) $l = 1$
(iii) **do** {
(iv) $\quad D_{l-1}^{Se} \leftarrow$ Select $N < M$ individuals from $D_{l-1}$ according to a selection method
(v) $\quad p_l(z) = p(z|D_{l-1}^{Se}) \leftarrow$ Estimate the joint probability distribution from the selected individuals
(vi) $\quad D_l \leftarrow$ Sample $M$ individuals (the new population) from $p_l(z)$
(vii) } **until** a stopping criterion is met

---

**Fig. 1** Pseudo-code for the EDA approach to optimization

induction of a multidimensional probabilistic model that reflects the interdependencies between the variables in these $N$ individuals is carried out. The estimation of the joint density constitutes the bottleneck of EDAs, as different degrees of dependencies between the variables used to represent the individuals can be considered. In a third step, $M$ new individuals–the new population—are obtained from a simulation of the multidimensional probabilistic model learnt in the second step. These three steps are repeated until a stopping condition is met.

The main advantages of EDAs as compared to GAs are: (i) they avoid designing ad hoc crossover and mutation operators, as well as the tuning of the values of several associated parameters; (ii) they are able to express in an explicit manner, by means of a joint probability distribution, the relationships between the different variables used to represent a point of the search space, and (iii) they can incorporate from the beginning some knowledge we can have about the problem by imposing conditional independence relationships between those variables. These advantages, as well as the difficulties of using real coded GAs, necessary in our problem, have led us to choose EDAs as better suited for this paper.

EDAs have been successfully applied in machine learning, for instance in learning Bayesian networks from data (Blanco et al. 2003; Romero et al. 2004), in feature subset selection (Inza et al. 2000), and in different optimization problems, within k-nearest neighbors, clustering and neural networks paradigms (Larrañaga and Lozano 2002).

### 5.2 UMDA$_c^G$ approach for logistic regression

Let $Z_i$, with $i = 0, 1, \ldots, k$, represent a continuous random variable. A possible value of $Z_i$ is denoted $z_i$. Each continuous variable is associated with its corresponding parameter of the logistic regression model. In our case, $z_i$ represents a value for parameter $\beta_i$. Similarly, we use $\mathbf{Z} = (Z_0, Z_1, \ldots, Z_k)$ to represent a $k+1$ dimensional random variable and $\boldsymbol{z} = (z_0, z_1, \ldots, z_k)$ to denote one of its possible values. In this sense, $\boldsymbol{z} = (z_0, z_1, \ldots, z_k)$ refers to a value for the parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)$. The joint density function over $\mathbf{Z}$ is denoted by $p(\boldsymbol{z})$.

In order to reduce as much as possible the computational cost derived from the learning of the joint density function, $p(\boldsymbol{z})$, we have chosen the EDA approach called *UMDA*$_c^G$ (Larrañaga et al. 2000). UMDA$_c^G$ assumes that at each generation all variables are independent and normally distributed. Tacking these two assumptions into account, the joint density at each generation, $p_l(\boldsymbol{z})$, can be factorized as follows:

$$p_l(\boldsymbol{z}) = \prod_{i=0}^{k} p_l(z_i) = \prod_{i=0}^{k} \frac{1}{\sqrt{2\pi}\sigma_{il}} \mathrm{e}^{-\frac{1}{2}\left(\frac{z_i - \mu_{il}}{\sigma_{il}}\right)^2}. \tag{3}$$

The $2(k+1)$ parameters of the model, $\mu_{il}$ and $\sigma_{il}$ with $i = 0, 1, \ldots, k$, have to be estimated at each generation by means of the sample mean and standard deviation calculated from the selected individuals.

We can use EDAs not only to obtain the values of the parameters $\beta_0, \beta_1, \ldots, \beta_k$ that maximize the likelihood but also to optimize other model performance measures like the AUC. We propose the use of EDAs, specifically the UMDA$_c^G$ approach, to build two new algorithms that use different fitness functions:

- UMDA$_c^G$-logL which goal is to obtain the $\beta$'s in a logistic regression model, with the highest log $\mathcal{L}$ value;
- UMDA$_c^G$-AUC which goal is to obtain the $\beta$'s in a logistic regression model, with the highest AUC value.

Note that, unlike the traditional procedures to find parameters $\beta_i$'s as MLE, the EDA approach is able to use any optimization objective, regardless of its complexity or the lack of an explicit formula for its expression, like for the AUC objective.

The parameters used to run the proposed algorithms based on EDAs and the method used for assessing the convergence may vary depending on the specific problem. The chosen values for our data sets are fully detailed in Sect. 6.2. As usual, the best individual in the last generation is chosen as solution.

Other EDA approaches that take into account more complex interactions among parameters $\beta_i$'s could be used, at the expense of the computational cost, but with the explicit modeling of their probabilistic conditional dependencies (see Larrañaga and Lozano 2002, Chap. 2).

## 6 Experiments

### 6.1 Data sets

The experimental study was carried out with six different data sets, described as follows:

- The `Breast` cancer data set intends to distinguish benign tumors from malignant tumors (benign/malignant).
- The `Diabetes` data set shows whether the patients have signs of diabetes (healthy/diabetes) according to *World Health Organization* criteria.
- The `ICU` data set contains patients who were part of a much larger study on survival of patients following admission to an adult intensive care unit (ICU). The major goal of this study is to develop a logistic regression model (or another valid method) to predict the probability of in-hospital survival (died/lived).
- The `Prostate` cancer data set involves a study of patients with prostate cancer. The goal of the analysis is to determine if variables measured at a baseline exam can be used to predict whether the tumor has penetrated (penetrated/not penetrated) the prostatic capsule.
- The purpose of the `UIS` data set is to compare treatment programs of different planned durations designed to reduce drug abuse and to prevent high-risk HIV behavior. The class variable is defined as having returned to drug use prior to the scheduled completion of the treatment program (remained drug free/ otherwise).
- The goal of the `Adult-r` data set is to predict whether income exceeds \$50 K/yr ($> 50$ K, $\leq 50$ K) based on census data. The original `Adult` data set includes 48,842 instances. By randomly selecting 8,000 instances maintaining the same proportion of positive and negative cases, the corresponding reduced version `Adult-r` is the data set used here.

**Table 1** Data sets characteristics

| Data set | Variables | Instances | Positive instances |
|---|---|---|---|
| `Breast` | 10 | 699 | 458 |
| `Diabetes` | 9 | 768 | 268 |
| `ICU` | 20 | 200 | 40 |
| `Prostate` | 8 | 380 | 153 |
| `UIS` | 8 | 575 | 147 |
| `Adult-r` | 14 | 8,000 | 1,912 |

The first two and the last data sets come from the UCI machine learning repository (Newman et al. 1998). The remaining data sets were obtained from (Hosmer and Lemeshow 2000). Table 1 contains the number of variables, the number of instances and the number of positive instances ($c_j = 1$) of each data set.

### 6.2 Implementation

To obtain the MLEs of $\beta$'s with the Newton–Raphson algorithm, we have used the R environment for statistical computing and graphics (R Development Core Team 2004; Ihaka and Gentleman 1996), which is freely available. MLEs are computed using the `glm()` function that takes into account the change between successive steps in parameter estimates to assess the convergence of the algorithm. We have called this algorithm R-glm. Besides, we have used the `somers2()` R function included in the `Hmisc` package for estimating the AUC. This function computes the $c$-index explained earlier.

For the new proposed algorithms based on EDAs, we have developed our own implementation in C++. EDAs were run with two different fitness goals: maximizing log likelihood (UMDA$_c^G$-logL) and maximizing the area under the ROC curve (UMDA$_c^G$-AUC).

The parameters used to run UMDA$_c^G$ were (see Fig. 1): (i) population size of 200 individuals ($M = 200$), (ii) the best 100 individuals were selected for the learning step ($N = 100$), and (iii) the change in the fitness value average between successive generations was chosen to assess the convergence of the algorithms. These parameters were tuned after some extra experiments.

As commented in Sect. 4, a 10-fold cross-validation was used to estimate model performance measures, both in R-glm and UMDA$_c^G$ algorithms (this process is shown in Fig. 2 for the EDA-based algorithms). Due to the stochastic nature of this validation method, each experiment was run five different times, therefore having a $5 \times 10$-fold cross-validation (Bouckaert and Frank 2004).

### 6.3 Comparison of algorithms

Table 2 summarizes the experimental results reflecting the average and standard deviation of the performance measures over all executions carried out with R (R-glm) and the two EDA algorithms. Note that for each algorithm we show not only the optimized measure but also the other measure in order to get an insight into the relationships between them.

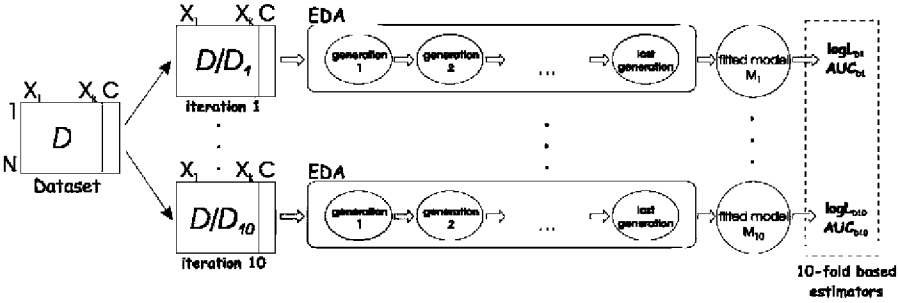


**Fig. 2** Assessing the performance of $\text{UMDA}_c^G$ algorithms using 10-fold cross-validation

**Table 2** Summary of the experimental results reflecting the average and standard deviation of all performance measures over the five executions, besides the average computation time for each fold. $\wedge$ means that EDA exhibits a statistically significant better behavior when compared to R-glm ($p$-value $< 0.05$). $\triangledown$ is for the opposite significance. Filled symbols are used for the performance measure that is being optimized by the associated algorithm

| Data set | Algorithm | Model performance measures | | Computation time (s) |
|---|---|---|---|---|
| | | log $\mathcal{L}$ | AUC | |
| Breast | R-glm | $-70.97 \pm 2.34$ | $0.9936 \pm 0.0003$ | 0.24 |
| | $\text{UMDA}_c^G$-logL | $-70.36 \pm 0.84$ | $0.9937 \pm 0.0001$ | 2.80 |
| | $\text{UMDA}_c^G$-AUC | $-84.70 \pm 6.20$ $\triangledown$ | $0.9930 \pm 0.0015$ | 3.13 |
| Diabetes | R-glm | $-320.85 \pm 1.60$ | $0.8298 \pm 0.0008$ | 0.32 |
| | $\text{UMDA}_c^G$-logL | $-319.48 \pm 1.76$ | $0.8318 \pm 0.0011$ | 4.24 |
| | $\text{UMDA}_c^G$-AUC | $-1526.52 \pm 241.56$ $\triangledown$ | $0.8267 \pm 0.0082$ | 5.48 |
| ICU | R-glm | $-142.72 \pm 25.70$ | $0.7754 \pm 0.0183$ | 0.49 |
| | $\text{UMDA}_c^G$-logL | $-138.67 \pm 16.79$ | $0.7617 \pm 0.0068$ | 4.66 |
| | $\text{UMDA}_c^G$-AUC | $-497.73 \pm 40.19$ $\triangledown$ | $0.7773 \pm 0.0082$ | 9.89 |
| Prostate | R-glm | $-202.09 \pm 2.18$ | $0.8017 \pm 0.0041$ | 0.20 |
| | $\text{UMDA}_c^G$-logL | $-199.25 \pm 1.94$ | $0.8066 \pm 0.0051$ | 1.82 |
| | $\text{UMDA}_c^G$-AUC | $-206.02 \pm 2.60$ $\triangledown$ | $0.8096 \pm 0.0076$ ▲ | 2.84 |
| UIS | R-glm | $-320.77 \pm 0.64$ | $0.6220 \pm 0.0030$ | 0.18 |
| | $\text{UMDA}_c^G$-logL | $-318.98 \pm 0.95$ ▲ | $0.6307 \pm 0.0037$ $\triangle$ | 3.99 |
| | $\text{UMDA}_c^G$-AUC | $-485.15 \pm 29.92$ $\triangledown$ | $0.6417 \pm 0.0106$ ▲ | 6.94 |
| Adult-r | R-glm | $-2832.48 \pm 8.78$ | $0.8449 \pm 0.0015$ | 1.59 |
| | $\text{UMDA}_c^G$-logL | $-2843.57 \pm 5.37$ | $0.8442 \pm 0.0018$ | 218.37 |
| | $\text{UMDA}_c^G$-AUC | $-12821.53 \pm 418.47$ $\triangledown$ | $0.8447 \pm 0.0012$ | 371.59 |

The Mann–Whitney test was used to compute the statistical significance of the difference between the algorithms. A $\wedge$ $\triangle$ symbol (a $\triangledown$ symbol) in a value means that the corresponding EDA algorithm reveals a statistically significant better (worse) behavior than R-glm with a $p$-value $< 0.05$. Filled symbols are used for the performance measure that is being optimized by the associated algorithm. The absence of the sym-

bol means that there was not enough evidence to reject the null hypothesis of equal behavior of the algorithms.

Several conclusions can be extracted from Table 2 with respect to the algorithms:

- The UMDA$_c^G$-logL algorithm achieves at least the same results as the R-glm algorithm. Most of the time the differences are negligible. In fact, there is only a statistically significant difference in the $\log \mathcal{L}$ value for the `UIS` data set in favor of the EDA ($p = 0.01$), see the second row of each data set, under the $\log \mathcal{L}$ column. In contrast, R-glm is never statistically superior to the EDA. Note that in this case both algorithms are using the same fitness function, $\log \mathcal{L}$, but a completely different search strategy.

  While optimizing $\log \mathcal{L}$, we can record the corresponding AUC (second row, last column). In this case, AUC as a complementary measure, exhibits the same behavior when the search is carried out using R-glm and EDA algorithms. There is one statistically significant difference in the AUC value for the `UIS` data set in favor of the EDA ($p = 0.007$).
- The behavior of the UMDA$_c^G$-AUC algorithm is now analyzed. Here we compare the AUC outputted by UMDA$_c^G$-AUC algorithm, which is its objective function, and the AUC corresponding with the model that R-glm finds (see the third row of each data set, under the AUC column). For `Breast`, `Diabetes`, `ICU` and `Adult-r` data sets, the differences are negligible. However, for `Prostate` ($p = 0.05$) and `UIS` ($p = 0.007$) data sets, EDA is again statistically superior to R-glm.

  When we look at the $\log \mathcal{L}$ values, recorded in our EDA algorithm as a complementary measure not used for the optimization (third row of each data set, under the $\log \mathcal{L}$ column), R-glm always displays statistically significant differences versus EDA ($p = 0.007$ always except for the `Prostate` data set, with $p = 0.03$).

Therefore, under the same optimization problem of maximizing $\log \mathcal{L}$, our EDA (UMDA$_c^G$-logL) algorithm behaves as a strong competitor of the R-glm algorithm, achieving similar and sometimes better results.

On the contrary, when we compare a certain measure yielded by algorithms that optimize different objectives, the results use to favor the algorithm that optimizes the measure. Thus, if we compare the $\log \mathcal{L}$ values outputted by the UMDA$_c^G$-AUC algorithm and the R-glm algorithm, the latter algorithm is better. However, when we compare the AUC values outputted by the UMDA$_c^G$-AUC algorithm and the R-glm algorithm, both algorithms almost tie, with the EDA algorithm being slightly superior.

In terms of computational time, EDA-based algorithms are slower than R-glm, but required times are quite reasonable. For the first five data sets, while R-glm ranges between 0.18 s and 0.49 s, UMDA$_c^G$-logL ranges between 1.82 s and 4.66 s and UMDA$_c^G$-AUC ranges between 2.84 s and 9.89 s. For the `Adult-r` data set, with more instances, R-glm needs 1.59 s while UMDA$_c^G$-logL and UMDA$_c^G$-AUC need 218.37 s and 371.59 s, respectively (see Table 2).

Note that optimizing an objective does not guarantee a good behavior of other possible objectives. How calibration and discrimination measures are related in logistic regression models is an aim in this paper and will be analyzed in the next subsections.

The results also show that optimizations with $\log \mathcal{L}$ as fitness function achieve good results in all performance measures while optimizations with AUC as fitness function achieve only good results in AUC values. Just another point of view: optimizing calibration results in optimizing discrimination, but optimizing discrimination does not result in optimizing calibration. As discussed in Sect. 3, $\log \mathcal{L}$ is a calibration measure while AUC is a discrimination measure.

It is possible to verify this situation by analyzing the evolution of the performance measures during the optimizations of the different fitness functions. For the `Diabetes` data set, the evolution of each measure, AUC and $\log \mathcal{L}$, during the maximization of $\log \mathcal{L}$ can be seen in Fig. 3. Similarly, the evolution during the maximization of AUC in the `Diabetes` data set can be found in Fig. 4. The behavior on the other five data sets is analogous.

Note that each logistic regression model, that is, specific values for the parameters $\beta_0, \beta_1, \ldots, \beta_k$, has associated AUC and $\log \mathcal{L}$ values represented in the curves of the figures. Since we have already compared the algorithms, from now on the models will be fitted by using the whole data set, i.e., by resubstitution.

As shown in Fig. 4 for the `Diabetes` data set during AUC maximization, when AUC values become stable (generation 12) at approximately 0.839, $\log \mathcal{L}$ values range from $-3064$ to $-1073$. However, as observed in Fig. 3, during $\log \mathcal{L}$ maximization, when $\log \mathcal{L}$ values become stable (generation 18) at approximately $-366$, AUC also reaches good and stable values. This means that $\log \mathcal{L}$ optimization leads to AUC optimization but not vice versa.

Moreover, the final $\log \mathcal{L}$ values when maximizing AUC are different depending on the execution. For example (see Table 2), in the `Diabetes` data set with stable final AUC values (mean 0.8267 with standard deviation 0.0082) there is a high

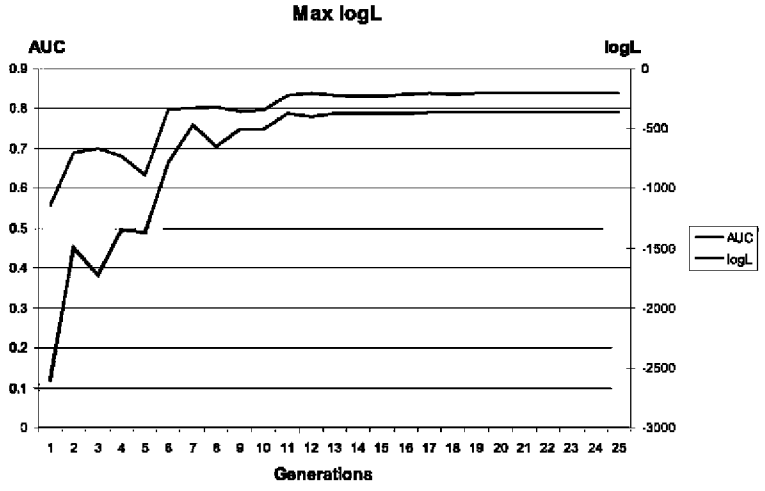


**Fig. 3** Evolution of $\log \mathcal{L}$ and AUC in the UMDA$_c^G$-logL algorithm for the `Diabetes` data set

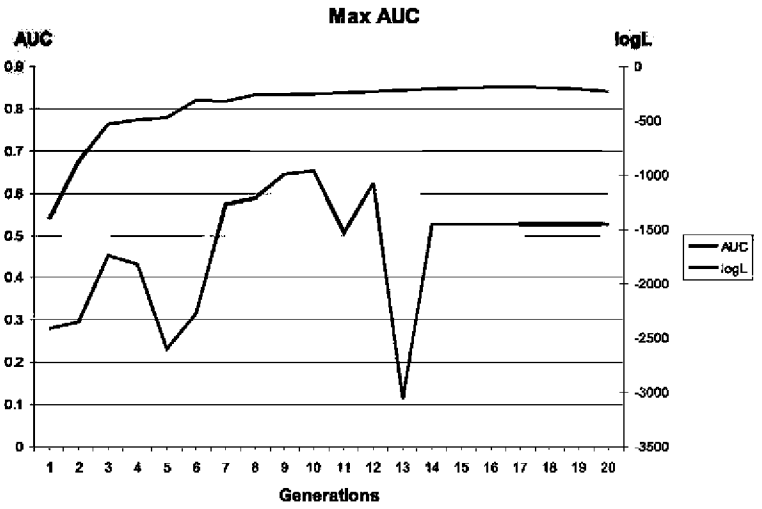


**Fig. 4** Evolution of $\log \mathcal{L}$ and AUC in the UMDA$_c^G$-AUC algorithm for the `Diabetes` data set

**Table 3** UMDA$_c^G$-AUC algorithm final results of the five executions for `Diabetes`

| Execution | Final $\log \mathcal{L}$ | Final AUC |
|---|---|---|
| 1 | −1440.478 | 0.8400 |
| 2 | −1413.735 | 0.8397 |
| 3 | −1395.572 | 0.8334 |
| 4 | −1096.662 | 0.8398 |
| 5 | −1407.625 | 0.8332 |

variance in the final $\log \mathcal{L}$ values (mean −1526.52 with standard deviation 241.56) because each of the executions reaches quite different search space points.

Table 3 reports the final $\log \mathcal{L}$ and AUC values of the logistic regression models that have been achieved in the five executions of the UMDA$_c^G$-AUC algorithm for the `Diabetes` data set.

An explanation for this lies in the different shapes of the optimization functions. The $\log \mathcal{L}$ is a concave function of the $\boldsymbol{\beta}$ coefficient, and the free variation of $\boldsymbol{\beta}$ in a convex set guarantees that there are no local maxima on the log-likelihood surface of a logistic regression model. That implies an easy-to-reach optimization, which also brings incidentally good values in other performance measures like AUC, as we have shown.

Nonetheless, the AUC surface has many local maxima, each one with similar AUC values but different $\log \mathcal{L}$ values. This implies a harder optimization problem, with the added difficulty of having to be cautious with the (perhaps bad) $\log \mathcal{L}$ value associated to a good AUC value obtained when maximizing the AUC function.

This is also observed in Fig. 5 that depicts the search space points that the EDA algorithms visited in the whole optimization process of the `Diabetes` data set. For

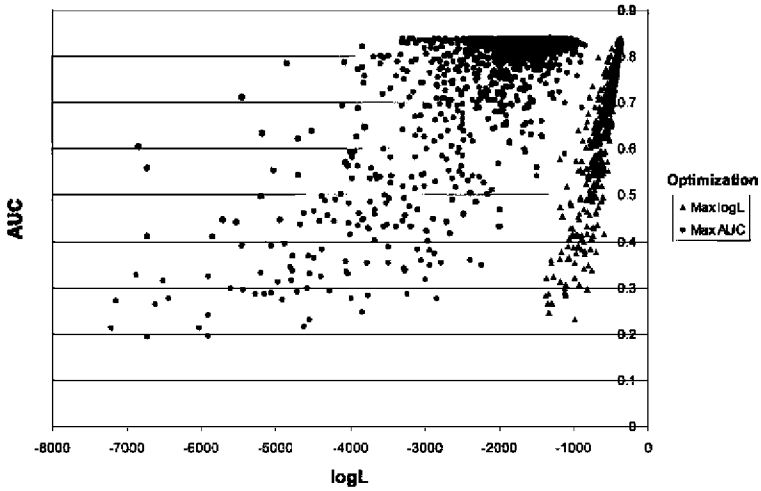


**Fig. 5** Performance measures, $\log \mathcal{L}$ vs. AUC, of all visited logistic regression models during the two EDA-based optimization processes of the `Diabetes` data set

each visited point the performance estimates of $\log \mathcal{L}$ and AUC are plotted. The behavior on the other five data sets is quite similar.

### 6.5 Pareto front in the calibration vs. discrimination space

Our previous experiments with EDAs have shown that optimizing $\log \mathcal{L}$ achieves good results in AUC. This provides the basis to develop an ad hoc method to explore promising regions of the bi-objective space of calibration ($\log \mathcal{L}$) vs. discrimination (AUC). The method proceeds as follows. The EDA algorithm is run many times, each one with the same value of parameter $\mu_{i0}$ and different $\sigma_{i0}$ ($i = 0, 1, \ldots, k$), both determining the normal density function of the $\text{UMDA}_c^G$ model at the initialization step, see Sect. 5.2. Parameter $\mu_{i0}$ is fixed as the component $i$ of the solution achieved by the Newton–Raphson method. Parameter $\sigma_{i0}$ is progressively increased to enlarge the space to be explored. Since $\log \mathcal{L}$ is the objective function for the EDA, this naïve procedure leads to points with hopefully good values not only in $\log \mathcal{L}$, but in AUC.

In the bi-objective space, non-dominated points are of interest as optimal solutions in the sense of not having any other point that is equal or better with respect to all the objective functions. The resulting set is the non-dominated or efficient set, also called Pareto front (Steuer 1986). Circles (magenta color) of Fig. 6 show the approximate Pareto front obtained by the ad hoc method described above.

Alternative ways to find the non-dominated points would consist of undertaking bi-objective optimizations. A recent research by Zhang et al. (2008) proposes a regularity model based multiobjective EDA (RM-MEDA) for continuous multiobjective optimization problems. Under mild smoothness conditions, it holds that the Pareto set is a piecewise continuous manifold of dimension $r - 1$, where $r$ is the number of objectives. The idea is to exploit explicitly this regularity property of the Pareto set in building the probabilistic model the EDA needs. EDA constructs the probability

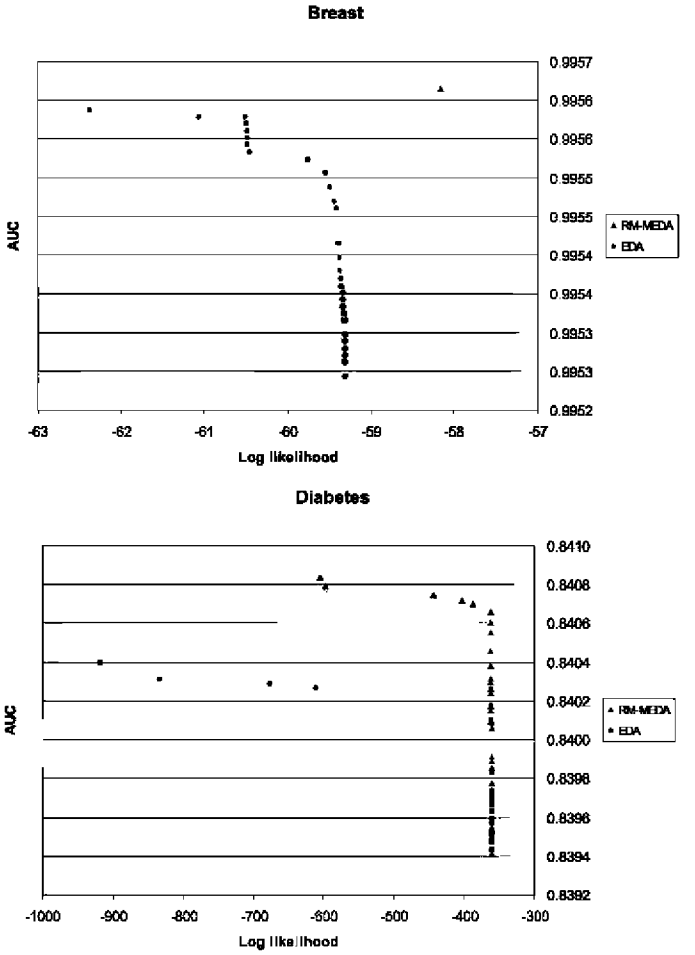


**Fig. 6** (Color online) Pareto fronts found by the ad hoc procedure (*magenta circles*) and by RM-MEDA (*blue triangles*) of all data sets

model whose centroid is a piecewise continuous manifold, via a local PCA algorithm. Experimental results show a global superiority of RM-MEDA against recent competitive multiobjective metaheuristics, including NSGA-II (Deb et al. 2006). Triangles (blue color) of Fig. 6 show the approximate Pareto front obtained by RM-MEDA.

It is remarkable that for `Breast`, the Pareto front includes an isolated point (located at the right upper corner). This point is only found by RM-MEDA. The behavior of RM-MEDA is also better for the rest of data sets as compared with our ad hoc method, specially as regards their AUC values. For `ICU` and `Adult-r`, all points found by our procedure are dominated by some point obtained by RM-MEDA. However, for `Diabetes`, `Prostate` and `UIS`, both methods find a similar number of points that are non-dominated between them. The chosen scale of Fig. 6, in particu-

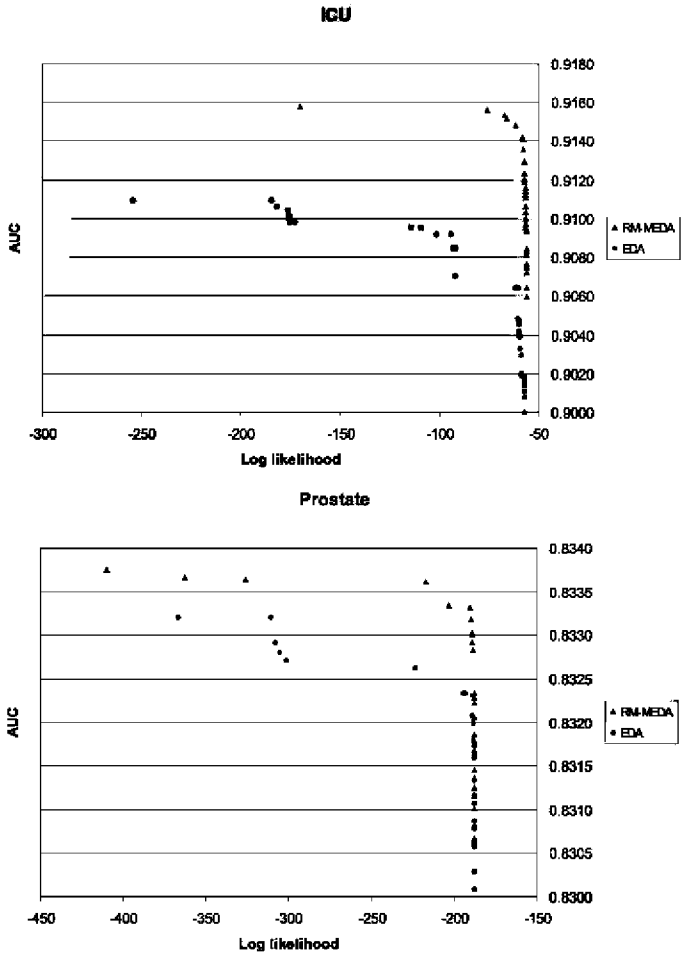


**Fig. 6** (*Continued*)

lar for the $\log \mathcal{L}$, avoids clearly visualizing this fact, since we tried to emphasize the differences between the AUC objective values for the two methods.

## 7 Discussion on benefits of the EDA approach

The advantages of using our EDA framework rather than the traditional numerical methods may be enumerated as follows:

- EDA is flexible enough to cope with any optimization function. EDA does not require derivative information nor matrix inversions. EDA does not need a function with an explicit formula, which is the case of the AUC function. On the contrary, numerical methods are only designed for optimizing the $\log \mathcal{L}$ function demanding matrix inversions.

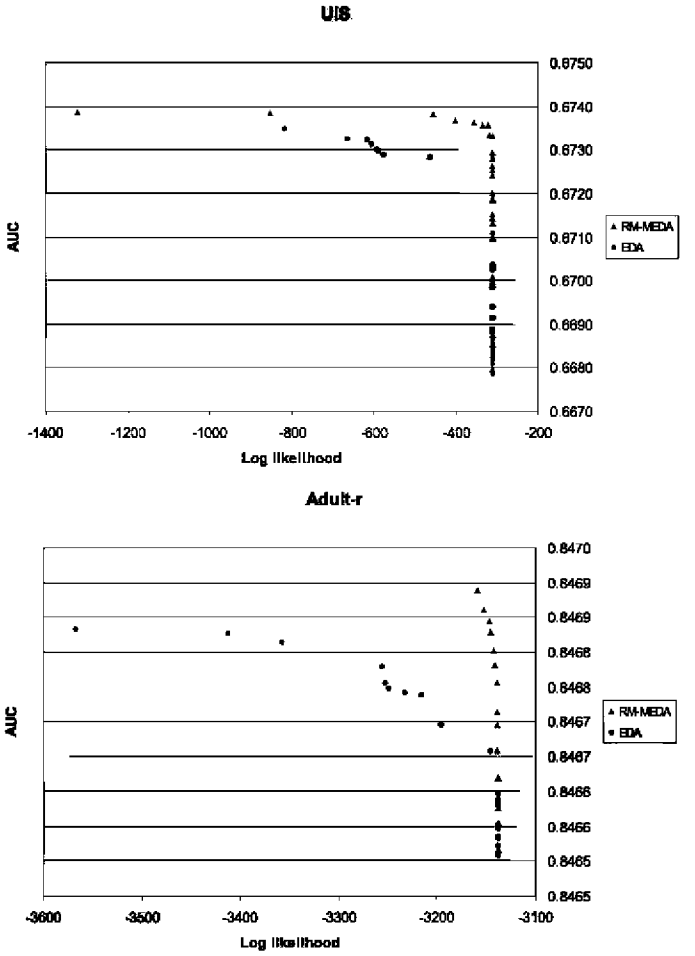


**Fig. 6** (*Continued*)

- EDA may use any performance measure. We have used log $\mathcal{L}$ and AUC but other performance measures may be used. In fact, we tried with the Hosmer–Lemeshow statistic calibration measure (Hosmer and Lemeshow 2000). The results were not shown, since they produced very high correlations (greater than 0.90) with the log likelihood function. We also tried with the Brier score (Brier 1950), with worse results than the current ones.
- EDA is a parallel and an inherently global optimal search technique that simultaneously evaluates many points in the parameter space and is more likely to converge toward the global solution of the optimization problem. Thus, it avoids being trapped at local optima. We are aware of local optima with the AUC function but not with log $\mathcal{L}$, which is concave.
- Moreover, numerical procedures like Newton–Raphson usually converge, but overshooting can occur (McLachlan 1992). Also, they exhibit some dependence on the

initial starting conditions for convergence to be guaranteed, although we do not have experimented this perhaps due to the refined implementation of the R program or due to the chosen data sets. Being a population-based search method, the EDA approach is unlikely to suffer from these drawbacks.
- EDA is not influenced by situations when the number of covariates is relatively large compared to the number of observations. Traditional numerical methods do not work in this scenario, having problems in estimating parameters properly.
- EDAs create a framework where we can study calibration and discrimination measures. We could investigate the behavior of EDAs when maximizing calibration or discrimination as compared to R-glm, with the only disadvantage of having bad $\log \mathcal{L}$ values when the AUC is optimized. The joint evolution of both measures has been analyzed, where a constructive interaction between calibration and discrimination was found when optimizing calibration, and more independence between both was found when optimizing discrimination.
- Furthermore, in the space of the two objectives of calibration vs. discrimination, the Pareto front was found with a competitive and sophisticated multiobjective EDA, RM-MEDA. In contrast, a simple uni-objective EDA procedure guided by the $\log \mathcal{L}$ yielded a slightly worse approximation to the Pareto front.

  These last two benefits of EDAs are not obtained with traditional numerical methods which only search for the point that maximizes the log-likelihood.

On the other hand, metaheuristics such as GAs, tabu search, ant colony, scatter search, etc., hold all the properties above. However, EDAs stand out against other metaheuristics and traditional numerical methods because of the following characteristics:

- EDAs avoid the tuning of the values of many parameters.
- EDAs capture explicitly the probabilistic dependencies among parameters $\beta_i$'s what is a useful information in logistic regression models. Different graphical structures may show chains, trees, polytrees, and general acyclic structures.

Some disadvantages follow:

- In a way the stochastic nature of EDA algorithms may be considered a disadvantage, since different executions may lead to slightly different results.
- All this comes at the price of having a higher computational cost, which in our case was alleviated by the powerful machines available to us (see Acknowledgments).

## 8 Conclusions and future research

To our knowledge, this was the first description of utilizing EDAs to estimate regression parameters, as well as the first one to compare different optimization functions, using maximum log-likelihood and the Newton–Raphson method implemented in R as a benchmark. Although our results did not differ dramatically from those of the benchmark, it is important to emphasize some points. First, MLE estimation requires the inversion of a matrix and will simply not work if the number of variables exceeds the number of observations. This is often the case in contemporary data sets.

Although dimensionality reduction and feature selection are active areas of research and may help remedy this situation, the use of EDAs can offer an attractive alternative, as the algorithms do not offer this limitation. Furthermore, we have only utilized a very simple form of EDAs, and it is expected that more complex forms will yield better results. Finally, our initial exploration of alternative optimization functions suggests that it may be better to favor functions that take into account calibration more strongly than discrimination, and provides initial empirical support for the use of these functions.

Our results support the need for further exploration of how to better estimate parameters for logistic regression. However, the use of EDAs is by no means limited to this type of models and further exploration in terms of their use in the context of more complex models is also warranted.

## References


Baumgartner C, Böhm C, Baumgartner D, Marini G, Weinberger K, Olgemöller B, Liebl B, Roscher AA (2004) Supervised machine learning techniques for the classification of metabolic disorders in newborns. Bioinformatics 20(17):2985–2996

Blanco R, Inza I, Larrañaga P (2003) Learning Bayesian networks in the space of structures by estimation of distribution algorithms. Int J Intell Syst 18:205–220

Bouckaert R, Frank E (2004) Evaluating the replicability of significance tests for comparing learning algorithms. In: Dai H, Srikant R, Zhang C (eds) PAKDD. LNAI, vol 3056. Springer, Berlin, pp 3–12

Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145–1159

Brier G (1950) Verification of forecasts expressed in terms of probabilities. Monthly Weather Rev 78:1–3

Deb K, Sinha A, Kukkonen S (2006) Multi-objective test problems, linkages, and evolutionary methodologies. In: GECCO-2006, Genetic and evolutionary computation conference, vol 2. ACM Press, New York, pp 1141–1148

Fawcett T (2003) ROC graphs: Notes and practical considerations for data mining researchers. Technical report, HPL 2003-4, HP Labs

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading

Hajek J, Zidak ZB, Sen PK (1999) Theory of rank tests, 2nd edn. Academic Press, San Diego

Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143:29–36

Harrell FE, Lee KL, Califf R, Pryor D, Rosati R (1984) Regression modelling strategies for improved prognostic prediction. Stat Med 3:143–152

Harrell FE, Lee KL, Mark DB (1996) Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Stat Med 15:361–387

Hilden J (1991) The area under the ROC curve and its competitors. Med Decis Mak 11(2):95–101

Horton NJ, Brown ER, Qian L (2004) Use of R as a toolbox for mathematical statistics exploration. Am Stat 58(4):343–357

Hosmer DW, Lemeshow S (2000) Applied logistic regression, 2nd edn. Wiley, New York

Ihaka R, Gentleman R (1996) R: A language for data analysis and graphics. J Comput Graph Stat 5:229–314

Inza I, Larrañaga P, Etxeberria R, Sierra B (2000) Feature subset selection by Bayesian network-based optimization. J Artif Intell Res 123(1–2):157–184
Kiang MY (2003) A comparative assessment of classification methods. Decis Support Syst 35:441–454
Larrañaga P, Lozano JA (2002) Estimation of distribution algorithms. A new tool for evolutionary computation. Kluwer Academic, Dordrecht
Larrañaga P, Etxeberria R, Lozano JA, Peña JM (2000) Optimization in continuous domains by learning and simulation of Gaussian networks. In: Workshop in optimization by building and using probabilistic models within the 2000 genetic and evolutionary computation conference, GECCO 2000, pp 201–204
Lasko TA, Bhagwat JG, Zou KH, Ohno-Machado L (2005) The use of ROC curves in biomedical informatics. J Biomed Inform 38:404–415
Lozano JA, Larrañaga P, Inza I, Bengoetxea E (2006) Towards a new evolutionary computation. Advances in estimation of distribution algorithms. Springer, New York
McLachlan G (1992) Discriminant analysis and statistical pattern recognition. Wiley, New York
Minka T (2003) A comparison of numerical optimizers for logistic regression. Technical report. 758, Carnegie Mellon University
Nakamichi R, Imoto S, Miyano S (2004) Case-control study of binary disease trait considering interactions between SNPs and environmental effects using logistic regression. In: Fourth IEEE symposium on bioinformatics and bioengineering, vol 21, pp 73–78
Newman D, Hettich S, Blake C, Merz C (1998) UCI repository of machine learning databases
Ng A, Jordan M (2001) On discriminative versus generative classifiers: A comparison of logistic regression and naive Bayes. In: Proceedings of NIPS, vol 14, pp 841–848
Pepe MS (2003) The statistical evaluation of medical tests for classification and prediction. Oxford University Press, Oxford
Provost F, Fawcett T, Kohavi R (1998) The case against accuracy estimation for comparing induction algorithms. In: Proceedings 15th international conference on machine learning. Morgan Kaufmann, San Mateo, pp 445–453
R Development Core Team (2004). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0
Romero T, Larrañaga P, Sierra B (2004) Learning Bayesian networks in the space of orderings with estimation of distribution algorithms. Int J Pattern Recogn Artif Intell 4(18):607–625
Ryan TP (1997) Modern regression methods. Wiley, New York
Steuer RE (1986) Multiple criteria optimization: Theory, computation, and application. Wiley, New York
Steyerberg E, Borsboom G, van Houwelingen H, Eijkemans M, Habbema J (2004) Validation and updating of predictive logistic regression models: a study on sample size and shrinkage. Stat Med 23(10):2567–2586
Stone M (1974) Cross-validatory choice and assessment of statistical predictions. J R Stat Soc Ser B 36:111–147
Thisted RA (1988) Elements of statistical computing. Chapman and Hall, London
van den Hout WB (2003) The area under an ROC curve with limited information. Med Decis Mak 23:160–166
Vinterbo S, Ohno-Machado L (1999a) A genetic algorithm to select variables in logistic regression: Example in the domain of myocardial infarctio. J Am Med Inform Assoc 6:984–988
Vinterbo S, Ohno-Machado L (1999b). A recalibration method for predictive models with dichotomous outcomes. In: Predictive models in medicine: Some methods for construction and adaptation. PhD thesis, Norwegian University of Science and Technology
Winker P, Gilli M (2004) Applications of optimization heuristics to estimation and modelling problems. Computat Stat Data Anal 47:211–223
Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: A regularity model based multiobjective estimation of distribution algorithms. IEEE Trans Evol Comput 12(1):41–63