Regularized Empirical EMP Maximization Framework for Profit-Driven Model Building

Eugen Stripling*
Department of Decision Sciences and Information Management, KU Leuven
Leuven, Belgium
eugen.stripling@kuleuven.be

Bart Baesens
Department of Decision Sciences and Information Management, KU Leuven
Leuven, Belgium
bart.baesens@kuleuven.be

Seppe vanden Broucke
Department of Decision Sciences and Information Management, KU Leuven
Leuven, Belgium
seppe.vandenbroucke@kuleuven.be

ABSTRACT

Value-centric decision making is vital for a corporate organization to stay competitive. Typically, this crucial business requirement is expressed as *profit maximization*. Consequently, it is necessary to also integrate this business requirement into the data science application. Profit-driven analytics acknowledges the need for the value-centric approach and aims to take both the costs and the benefits that result from a business action into consideration. A few profit-driven methods were proposed in the literature but they lack a general formulation of the essential parts that make a successful profit-driven model.

Hence, we propose the general *Regularized Empirical EMP Maximization* (*REEM*) framework for profit-driven model building. It centers around the *Expected Maximum Profit* (*EMP*) measure and formally describes the building blocks required for constructing a profit-driven model. We demonstrate *REEM*'s applicability for customer churn prediction. This entails finding the best-performing derivative-free optimization (DFO) method. Through the state-of-the-art Bayesian hierarchical correlated t test, we find that the real-coded genetic algorithm is an adequate DFO method for solving the *REEM* problem. The goal of *REEM* is to drive profit-oriented analytics in a forward direction by generalizing the previously proposed methods and serving as a formal guide for new profit-driven classifiers to come.

KEYWORDS

Profit-driven model building, general framework, derivative-free optimization

ACM Reference Format:
Eugen Stripling, Bart Baesens, and Seppe vanden Broucke. 2018. Regularized Empirical EMP Maximization Framework for Profit-Driven Model Building. In *Proceedings of 2018 Workshop on Utility-Driven Mining in conjunction with the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (UDM workshop).* ACM, New York, NY, USA, Article 6, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Corresponding author.

UDM workshop, August 20, 2018, London, UK

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Profit maximization is a crucial and permanent business requirement of a corporate organization in order to secure the future of the company in an ever-increasing competitive market. For example, consider the saturated market of the telecommunications services sector, it is vital to the business of a telephone company (telco) to prevent their customers from switching to a competitor (i.e., from churning). In doing so, the telco operator has to rely on predictive models to identify potential churners. Accurately predicting a customer as a churner does not suffice, however, since the company aims to first retain those customers that are the *most valuable* to the business.

Profit-driven business analytics is a branch of research [19, 40–45] that acknowledges the need for the *value-centric* approach to provide support for businesses in their day-to-day decision making. It actively aims to not only take the costs (cf. cost-sensitive learning) but also to take the *benefits* that result from a business action into consideration when building the data science application.

In previous works, Verbraken et al. [45] established the *Expected Maximum Profit* (*EMP*) measure, a profit-based classification performance metric. Taking a *probabilistic* perspective, the *EMP* measure is inherently suitable for real-world applications, because it accounts for the uncertainty that naturally emerges from trying to pinpoint the exact values of the costs and benefits of a business action. Nevertheless, the *EMP* measure is only applicable for *profit-based model selection*. Hence, there is an appetency to move forward to *profit-driven model building* [42, 44] in which a predictive model is built such that it directly accounts for the main business requirement of profit maximization [41, 42, 45]. A few works by [19, 39, 40] explore this research direction but still lack a general formulation of the essential parts required for constructing a profit-driven model.

In this work, we introduce the general *Regularized Empirical EMP Maximization* (*REEM*) framework for profit-driven model building. The *REEM* framework aims to drive profit-oriented business analytics in a forward direction by generally outlining the components necessary for successfully building a profit-driven classification model. Thus, *REEM* is a *generalization* of previously established techniques (i.e., *ProfLogit* and *ProfTree*) [19, 39, 40] and serves as a guide for new profit-driven classifiers to come. In particular, we demonstrate the applicability of the *REEM* framework for customer churn prediction through *ProfLogit*. Due to the inherently complex nature of the *REEM* problem, there is a strong reliance on derivative-free optimization (DFO) methods.

Our contributions can be summarized as follows:

- We introduce the general framework called *Regularized Empirical EMP Maximization* (*REEM*) for profit-driven model building. It is a formal description of the necessary components for designing a profit-driven classifier that can be used in a specific business context (e.g., churn prediction).
- We perform a comparative study of derivative-free optimization (DFO) methods on multiple real-world churn data sets with adequate statistical tests. More specifically, we make use of the Bayesian hierarchical correlated t test to determine the best DFO solver for the *REEM* problem.

Throughout the text, it should be kept firmly in mind that this work solely focuses on the *application* of a *selection* of DFO methods – as a means to an end – for solving a difficult optimization problem. The vast amount of available DFO methods and their numerous variants makes it sheer impossible to survey all techniques. A extensive comparative study of DFO methods is therefore *out of scope*.

The remainder of this paper is structured as follows. The next section concisely provides background information on the *EMP* measure, outlines the global optimization problem and selected DFO methods for the comparative study, as well as discusses related work. In §3, we introduce our proposed *REEM* framework. The empirical study in which we compare various DFO methods to solve the *REEM* problem is presented in §4. Finally, we summarize the conclusions of this research in §5.

2 BACKGROUND AND RELATED WORK

2.1 The EMP Framework

2.1.1 General EMP Formulation. In previous works [42, 43, 45], the *EMP* measure was introduced for profit-based binary classification performance evaluation, which takes into consideration the costs and benefits that emerge due to an action undertaken toward *predicted* cases. In the *EMP* framework, a *case* is an instance with a class label 0, representing an event of interest (e.g., churn), whereas a *non-case* is an instance with a class label 1. This non-conventional encoding is used because it results in a simplified mathematical notation [42, 43, 45] and is required for the empirical *EMP* calculation [42, see §3.4.2 and §4.4]. This notation was also used, for example, by [11]. Consequently, the *EMP* framework assumes that the instances from class 0 have a lower score than those from class 1.

The benefit of a correctly predicted case is expressed as b_0, and the cost of an incorrectly predicted case as c_1. The cost associated with an action undertaken toward a predicted case is denoted as c^*, assuming $c^* < b_0$. At the core of the *EMP* measure is the *average classification profit* function that we explicitly specify in terms of a classifier g for our purposes.

Definition 2.1 (Average Classification Profit, adapted from [42]). Let $b_0, c_1, c^* \in \mathbb{R}_{\geq 0}$, $b_0 > c^*$, be the incremental classification benefits and costs corresponding to predicted cases. The average classification profit of a classifier g, $P_g(t;\cdot)$, is the profit generated by the employment of the classifier, and is expressed by:

$$P_g(t; b_0, c_1, c^*) = (b_0 - c^*)\pi_0 F_0^g(t) - (c_1 + c^*)\pi_1 F_1^g(t) \quad (1)$$

with $t \in \mathbb{R}$, $\pi_k \in [0,1]$, and $F_k^g(s)$, $k \in \{0,1\}$, respectively being the classification threshold, the prior class probability of class k, and the cumulative distribution function of the scores generated by classifier g for class k instances.

Observe that for a fixed classification threshold t, $F_0^g(t)$ and $F_1^g(t)$ express the *true positive rate* and *false positive rate* of classifier g, respectively [37, 42]. For the *EMP* measure, a probability density function h is assigned to the classification costs and benefits, reflecting our uncertainty about the precise values of these parameters. This leads to the definition of the *EMP*, which is a *probabilistic* profit-based performance measure, measuring the *expected* maximum profit of a classifier g.

Definition 2.2 (Expected Maximum Profit, adapted from [42]). Let $P_g(t; b_0, c_1, c^*)$ be the average classification profit (1) of a classifier g and $h(b_0, c_1, c^*)$ be the joint probability density of the classification costs and benefits: $b_0, c_1, c^* \in \mathbb{R}_{\geq 0}$, $b_0 > c^*$. The expected maximum profit, *EMP*, is the expectation of the maximal profit of a classifier g with respect to the distribution of classification costs, and is equal to:

$$EMP(g) = \int_{b_0}\int_{c_1}\int_{c^*} P_g(T(\theta); b_0, c_1, c^*) h(b_0, c_1, c^*)\, \mathrm{d}c^*\, \mathrm{d}c_1\, \mathrm{d}b_0, \quad (2)$$

where T is the optimal classification threshold such that the average classification profit in (1) is maximized, which in turn solely depends on the ratio of the given cost benefit parameters: $\theta = (c_1 + c^*)/(b_0 - c^*) \in [0, +\infty)$.

Taking a *probabilistic* perspective makes the *EMP* measure more adequate for real-world business applications than other common classification performance measures (e.g., the area under the ROC curve), as it accounts for the uncertainty in costs and benefits.

2.1.2 Specific EMP Formulation for Customer Churn Prediction. For the *EMP* framework to be applicable in practice, it is required to specify reasonable values for the classification costs and benefits, as well as finding a sensible probability density for the uncertain parameters involved. In previous works [42, 45], the following specifications were proposed for customer churn prediction (CCP) within the context of a churn management campaign:

$$b_0 = \gamma(CLV - d) = \gamma(1-\delta)CLV,\ c_1 = d = \delta CLV,\ c^* = f = \phi CLV, \quad (3)$$

where γ is the probability of acceptance, CLV is the customer lifetime value (200 €), $\delta = d/CLV$ corresponds to the cost of an offer ($d = 10$ €), and $\phi = f/CLV$ corresponds to the cost of contact ($f = 1$ €). Uncertainty surrounds the probability of acceptance, therefore a Beta($\alpha = 6, \beta = 14$) distribution is assigned to γ. Note that the values in parentheses were extensively researched and are recommended as default values for the telco sector [45]. Furthermore, all parameters must be strictly positive, where it must hold that CLV $> d$ and $\alpha, \beta > 1$ for the Beta distribution. By plugging (3) into (1), we obtain the average classification profit function for CCP.

Definition 2.3 (Average Classification Profit for Customer Churn Prediction, adapted from [42]). The average classification profit of a classifier g for customer churn prediction, $P_g^{ccp}(t;\cdot)$, is the

interpretation of Def. 2.1 specifically for customer churn:

$$P_g^{ccp}(t;\gamma,CLV,\delta,\phi) = CLV(\gamma(1-\delta)-\phi)\pi_0 F_0^g(t) - CLV(\delta+\phi)\pi_1 F_1^g(t). \tag{4}$$

Once a profit function is defined, it is straightforward to derive the respective *EMP* measure.[1]

Definition 2.4 (Expected Maximum Profit for Customer Churn Prediction, adapted from [42]). The expected maximum profit of a classifier g for customer churn prediction, EMP^{ccp}, is the interpretation of Def. 2.2 in a customer churn setting:

$$EMP^{ccp}(g) = \int_\gamma P_g^{ccp}(T(\gamma);\gamma,CLV,\delta,\phi)h(\gamma)\,\mathrm{d}\gamma \tag{5}$$

with T being the optimal classification threshold for a given γ and $h(\gamma)$ being the probability density function assigned to γ.

2.2 Derivative-Free Optimization Methods for Finding Globally Optimal Solutions

Derivative-free optimization (DFO) methods are auspicious techniques for the optimization problem we introduce in §3. A DFO method allows optimizing complex functions of which the derivatives are either expensive to compute, difficult to derive, or do no exist; even the analytical formula of the function itself does not have to be known. In this sense, DFO is also referred to as *black-box optimization* in which function values of evaluated search points are the only accessible information on the objective function [13]. Additionally, most DFO methods are capable of performing *global optimization* in which the real-valued function to be optimized might also be non-convex, non-linear, noisy, or multimodal.[2]

Definition 2.5 (Global Optimization [10, 33]). Considering a maximization problem, the general *global optimization problem* is formulated as

$$\underset{x\in\Theta}{\text{maximize}}\ f(x)$$

with $\Theta \subset \mathbb{R}^d$, $d \in \mathbb{N}_1$, being a compact set and f being a continuous function on Θ. The goal is to find a point x in the set $\Theta^\star = \{x \in \Theta \mid f(x) = f^\star\}$, where $f^\star = \max_{x\in\Theta} f(x)$. This problem is inherently unsolvable in a finite number of steps without additional assumptions. Thus, the global optimization problem is usually considered as solved if a point is found in $B_\epsilon^\star = \{x \in \Theta \mid \|x^\star - x\| \le \epsilon \text{ for some } x^\star \in \Theta^\star\}$ or in the level set $\Theta_\epsilon = \{x \in \Theta \mid f(x) \ge f^\star - \epsilon\}$ for some $\epsilon > 0$. A point $x \in B_\epsilon^\star$ (or $x \in \Theta_\epsilon$) that is in the vicinity of the global optimum is regarded as a *near-optimal* solution.

In Def. 2.5, we formulate the optimization task as a maximization problem. If the formulation of a minimization problem is preferred, one can simply apply a reversion of the inequality signs and a multiplication with minus one of the function (i.e., $-f$).

[1] An R implementation is provided in the `EMP` package [5].

[2] In mathematics, a real-valued function f is said to be *convex* if $f(ax + (1-a)y) \le af(x) + (1-a)f(y)$ for any x and y in an interval, a convex set in a real vector space, and for any $a \in [0, 1]$. If f is twice-differentiable, $f(x)$ is convex if and only if $\partial^2 f(x)/\partial x^2 \ge 0$. The negation of a convex function is a *concave* function (i.e., reversing the inequality sign). Hence, minimizing a convex function f is equivalent to maximizing the function $-f$, which is concave. In this text, we use the term 'convex (upwards)' instead of 'concave' in order to be consistent with the *EMP* framework. From the context, however, it is clear whether 'convex' refers to a minimization or maximization problem, where for the latter the term 'upwards' is omitted.

Over the past decades, a myriad of DFO methods has been proposed in the academic literature (see, e.g., [23, 24, 26, 32]). To this day, DFO is an active research field and shows no signs of a slowdown. To name a few methods, DFO can be performed through Bayesian optimization, coordinate descent, differential evolution, evolution strategy, genetic algorithm, hit-and-run algorithms, Nelder-Mead simplex algorithm, particle swarm optimization, Hooke-Jeeves method (or pattern search), random-search algorithms, and simulated annealing.

Many review papers exist that provide a comprehensive overview of a specific DFO method, as well as for their various extensions (see, e.g., [8, 9, 29, 31, 48]). Most of those methods typically have parameters that are critical for the performance, which themselves may be researched intensively (see, e.g., [28]). Clearly, each DFO method has its own merits, but the *No Free Lunch Theorem* [47] dictates that there is no single best optimization method. It is therefore of interest to investigate which of the considered DFO methods performs best—"best" meaning *optimal* in terms of a performance measure of interest (e.g., EMP^{ccp})—on our optimization problem given a fixed budget or maximum allowable number of function evaluations.

As mentioned in the introduction, we concentrate on a selection of DFO solvers we view as being "suitable for the task." Following the criteria for our selection of DFO methods: It must (i) be a well-established method in academic literature; (ii) be a global optimizer that does *not* require the real-valued objective function to be convex, linear, unimodal, or differentiable; (iii) be a solver for real-parameter single objective optimization problems; (iv) have a publicly available software implementation.

Table 1 provides an overview of selected DFO methods that satisfy our criteria for the empirical study in §4. Our selection mostly contains evolutionary algorithms (EAs) as they are proven to find the global optimum given limitless search time, as long as the EA is elitist [30].

2.3 Related Work

Stripling et al. [39, 40] introduced the *ProfLogit* method which maximizes the EMP^{ccp} when building the logistic regression model through a RGA. Using real-world churn data sets, they showed that their proposed method achieves the overall highest EMP^{ccp} performance on independent test sets. An interesting observation of their study is the *profit-accuracy trade-off*, meaning that their method scores high on the profit-based measures but low on the non-profit-based measures (e.g., the area under the ROC curve). This trade-off pattern was also observed by Verbraken et al. [45]. In the last installment of *ProfLogit* [40], a lasso penalty [17] was incorporated which drastically improved the model performance.

In similar fashion, Höppner et al. [19] proposed the *ProfTree* method that maximizes the EMP^{ccp} measure through an evolutionary decision tree induction algorithm. The authors also incorporated a penalty term for model complexity (i.e., number of terminal nodes) into *ProfTree*'s EMP^{ccp}-based objective function. In their empirical study, the authors showed that their proposed method achieves an overall higher EMP^{ccp} performance than other common decision tree induction algorithms, and they also confirmed the profit-accuracy trade-off.

Table 1: Overview of Selected Derivative-Free Optimization (DFO) Methods.

DFO method	Abbreviation	Description	Population-based	Implementation	References
Covariance matrix adaptation evolution strategy	CMAES	Uses a multivariate normal distribution to generate candidate solutions. The corresponding covariance matrix is updated as the algorithm progresses to better adapt to the search requirements (cf. exploration-exploitation trade-off).	✓	`cma` [12]	[12–15]
Differential evolution	DE	Generates new candidate solutions by combining distinct population members, most noticeable through the use of differential mutation. This key feature allows the algorithm to automatically adapt to the function surface.	✓	`SciPy` [20, 27]	[30, 38]
Particle swarm optimization	PSO	Mimics collective behavior that is comparable to a flock of birds. The particles (or candidate solutions) influence the movement of other swarm members, thereby steering the swarm toward the optimal solution.	✓	`pyswarm`	[21, 36]
Basin-hopping	BH	Operates in a similar fashion as simulated annealing, but additionally applies a local numeric optimization at each iteration.		`SciPy` [20, 27]	[22, 46]
Real-coded genetic algorithm	RGA	Mimics the biological process of evolution, where it induces random perturbations in the genetic pool and applies the "Survival of the Fittest" principle to steer the search toward the optimal solution. The algorithm is adapted to solve real-parameter single objective optimization problems.	✓	on GitHub [a]	[18, 40]
Pure random search	PRS	Randomly samples candidate solutions from a uniform distribution over the search space. The candidate solutions are independent and identically distributed.		`NumPy` [27]	[2, 6, 33]

[a] https://github.com/estripling/proflogit-python

Despite the successful practical application of these profit-driven classification models, both works lack to establish a rigorous and more formal description for profit-driven model building. Therefore, the presented work in this paper aims to fill this gap and to formally establish the framework for profit-driven model building on the basis of the *EMP* measure described in §2.1.

3 METHODOLOGY: PROFIT-DRIVEN MODEL BUILDING

3.1 Foundation for Profit-Driven Model Building

Taking a probabilistic perspective on binary classification [4], let (X, Y) be a pair of random variables, modeling an *instance* $x \in \mathcal{X}$ and its corresponding class *label* $y \in \mathcal{Y}$, where $\mathcal{X}$ represents the *instance domain*, some measurable space equipped with a σ-algebra, and $\mathcal{Y}$ is the *label domain*, a two-element set of possible class labels. We assume that the random pair (X, Y) follows the true (but unknown) probability distribution $\mathcal{P}$. The realization after measurement of, e.g., the random variable X is denoted by $X = x$.

The goal is to learn function or (soft) classifier $g\colon \mathcal{X} \to \mathbb{R}$, which is a mapping expressing the confidence (a continuous score) in the guess of Y given X. The hard classifier is easily derived by $g_c(X, t) = \mathbb{1}_{[g(X)\geq t]}$, where $\mathbb{1}_{[\cdot]}$ denotes the indicator function and $t \in \mathbb{R}$ is the classification threshold.

In the model building phase, the standard notation is applied in which *cases* and *non-cases* are encoded as 1s and 0s, respectively. However, when computing the *EMP*, the label and the score generated by g of a given instance-label pair are respectively inverted by the expressions $s_{\mathcal{Y}}(Y) = \mathbb{1}_{[Y=0]}$ and $s_g(X) = 1 - g(X)$ in order to comply with the *EMP* framework (§2.1). In this way, the optimization problem at the core of the classification model is solved in a familiar fashion (same holds for the model interpretation if applicable).

Further, let S_k^g be the random variable (RV) that takes values in

$$\left\{ s_g(x) \in \mathbb{R} \;\middle|\; (\forall (x, y) \in \mathcal{X} \times \mathcal{Y})[s_{\mathcal{Y}}(y) = k] \right\}, \tag{6}$$

which is the set of scores generated by a classifier g for class $k \in \mathcal{Y}$. Then, the cumulative distribution function (CDF) of S_k^g is defined by

$$F_k^g\colon \mathbb{R} \to [0, 1] \qquad F_k^g(s) = \Pr(S_k^g \leq s) = \int_{-\infty}^{s} f_k^g(r)\,\mathrm{d}r, \tag{7}$$

where f_k^g is the probability density function (PDF) computed based upon the score set in (6). Given (6) and (7), the *EMP* formulation in (2) is now fully specified in terms of classifier g.

Since the classification model is built based on a sample of instance-label pairs, there is a minimal non-zero error for any classifier.

Definition 3.1 (Bayes Expected Maximum Profit and Classifier). Given a distribution $\mathcal{P}$ over $\mathcal{X} \times \mathcal{Y}$, the Bayes expected maximum profit, $EMP^\star$, is defined as the supremum of the *EMP*s achieved by measurable functions $g\colon \mathcal{X} \to \mathbb{R}$:

$$EMP^\star = \sup_{\substack{g \\ g \text{ measurable}}} EMP(g). \tag{8}$$

The *Bayes classifier* $g^\star$ is then the classifier with $EMP(g) = EMP^\star$.

By definition, it holds that $EMP(g^\star) \geq EMP(g)$ for any classifier g. Thus, the Bayes classifier achieves the highest attainable *EMP* performance, but it is not accessible in practice since the true form of $\mathcal{P}$ is unknown.

Typically, we only have access to a collection of instance-label pairs: $D^n = (X_1, Y_1), \ldots, (X_n, Y_n)$, assuming that (X_i, Y_i), $i \in [n]$,[3]

[3] The notation $[a]$ is shorthand for the set $\{1, 2, \ldots, a\} \subset \mathbb{N}_1$.

are independent and identically distributed (i.i.d.) random pairs, having the same distribution as (X, Y). Denote by g_n the classifier built based on D^n.

Next, let $S_{k,n}^{g_n}$ denote a sequence of RV's $\left\{ S_{k,1}^{g_n}, S_{k,2}^{g_n}, \dots \right\}$ with its CDF being $F_{k,n}^{g_n}$. Then, the sequence converges in distribution with increasing n:

$$S_{k,n}^{g_n} \xrightarrow{\text{D}} S_k^g \tag{9}$$

or equivalently

$$\lim_{n \to +\infty} F_{k,n}^{g_n}(s) = F_k^g(s) \tag{10}$$

for all s where F_k^g is continuous. Hence,

$$\lim_{n \to +\infty} EMP(g_n) = EMP(g). \tag{11}$$

Typically, we consider a class or family of classifiers $C : \mathcal{X} \to \mathbb{R}$, and we are interested in finding a classifier $g \in C$ built based on D^n that has an *EMP* performance closest to $EMP^\star$.

3.2 Regularized Empirical EMP Maximization

In the model building phase, the *EMP* of a binary classifier is merely computed based on the available data $D^n = (x_1, y_1), \dots, (x_n, y_n)$ with $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, i \in [n]$, sampled i.i.d. from distribution $\mathcal{P}$. Informally, the goal is to find the classifier g from a given class of classifiers $C : \mathcal{X} \to \mathbb{R}$ that exhibits the highest empirical *EMP* that can be achieved on the basis of D^n. Note that, in this section, we omit n in the subscript of g which indicated that g is built on D^n to lighten the notation.

Definition 3.2 (Empirical Expected Maximum Profit). Given a classifier $g \in C : \mathcal{X} \to \mathbb{R}$ and a collection of observed instance-label pairs $D^n = (x_1, y_1), \dots, (x_n, y_n)$, the empirical expected maximum profit of classifier g, which is the empirical interpretation of Def. 2.2, is denoted by $EMP_n(g)$, and it is computed based upon the empirical score sets

$$\left\{ s_g(x) \in \mathbb{R} \;\middle|\; (\forall (x, y) \in D^n)[s_{\mathcal{Y}}(y) = k] \right\} \tag{12}$$

for class $k \in \{0, 1\}$.

Observe that the empirical *EMP* is the mapping $EMP_n : C \to \mathbb{R}_{\geq 0}$. For the specific *EMP* formulations established in the research literature (i.e., for churn and credit scoring), the empirical calculation of the *EMP* involves computing the convex hull of the empirical ROC curve on the basis of the empirical score sets in (12).

Commonly, a machine learning model involves a *regularization term* in order to penalize model complexity, leading to a better generalization performance [16, 17, 25, 34, 35]. For a classifier g, it is generally expressed by

$$performance(g) + \lambda \cdot complexity(g),$$

where $\lambda \in \mathbb{R}_{\geq 0}$ is the regularization parameter, controlling the influence of the regularization that penalizes for the model complexity of g. To this end, the general objective function of regularized empirical *EMP* maximization (*REEM*) is defined by

$$Q^{REEM}(g) \stackrel{\text{def}}{=} EMP_n(g) - \lambda R(g), \tag{13}$$

where $EMP_n(g)$ is defined in Def. 3.2, $\lambda \in \mathbb{R}_{\geq 0}$ is the regularization parameter, and $R : C \to \mathbb{R}_{\geq 0}$ is the regularizer. The latter depends strongly on the classifier class considered. Moreover, it has been shown [19, 40] that the inclusion of a regularization term in the objective function of a profit-driven model indeed immensely helps to improve the model performance.

Definition 3.3 (Regularized Empirical EMP Maximizer). Given a classifier class $C : \mathcal{X} \to \mathbb{R}$ and a sample of observed instance-label pairs $D^n = (x_1, y_1), \dots, (x_n, y_n)$, the *REEM* classifier maximizes the objective function (13):

$$\begin{aligned} g^{REEM} &= \arg\max_{g \in C} Q^{REEM}(g) \\ &= \arg\max_{g \in C} EMP_n(g) - \lambda R(g) \end{aligned} \tag{14}$$

with $EMP_n : C \to \mathbb{R}_{\geq 0}$, $\lambda \in \mathbb{R}_{\geq 0}$, and $R : C \to \mathbb{R}_{\geq 0}$ being the empirical *EMP* of g measured on D^n, the regularization parameter, and the regularizer, respectively.

Now, for the practical application of Def. 3.3, it becomes a matter of configuration, requiring to make choices for the following:

(1) The *EMP* measure for the business application of interest. For example, Eq. (5) for customer churn prediction.
(2) The family of classifiers $C : \mathcal{X} \to \mathbb{R}$.
(3) The regularizer $R(g)$, applicable to any $g \in C$.
(4) The DFO method that can operate on C.

3.3 Regularized Empirical EMP Maximization via Logistic Regression

In §3.2, we established the *REEM* framework that finds a classifier g from a classifier class $C : \mathcal{X} \to \mathbb{R}$ such that a chosen profit-based performance measure is maximized through the classifier on the basis of a sample D^n. One of the *REEM* requirements is to define C. Here, we consider the logistic regression (LR) model, a common choice for binary classification, that fits the specified model requirements well. For this purpose, assume a data sample of the form $D^n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is available, where the instance i is described by a d-dimensional feature vector $\mathbf{x}_i$ and label $y_i \in \{0, 1\}$. Now, to provide a formal LR definition, we proceed as follows.

First, let the class of affine functions $C_{AF} : \mathcal{X} \to \mathbb{R}$ be

$$C_{AF} = \left\{ \mathbf{x} \mapsto \beta_0 + \langle \boldsymbol{\beta}, \mathbf{x} \rangle \;\middle|\; \mathbf{x} \in \mathcal{X}, \boldsymbol{\theta} \in \Theta \right\}, \tag{15}$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is the instance domain, $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\beta}^\intercal)^\intercal$ is the parameter vector associated with the parameter space $\Theta \subseteq \mathbb{R}^{d+1}$, and

$$\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x} \rangle = \beta_0 + \sum_{j=1}^{d} \beta_j x_j. \tag{16}$$

Each function in C_{AF} takes as input a vector $\mathbf{x} \in \mathbb{R}^d$ and returns as output the scalar $\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x} \rangle$, where $\beta_0 \in \mathbb{R}$ is a constant and $\boldsymbol{\beta} \in \mathbb{R}^d$ a vector.

Second, the logistic function $p : \mathbb{R} \to (0, 1)$, which is a special sigmoid curve, is given by

$$p(z) = \left(1 + e^{-z}\right)^{-1}. \tag{17}$$

Then, the class of logistic regression models $C_{LR} : \mathcal{X} \to (0, 1)$ is defined to be the composition of the logistic function over class of affine functions:

$$C_{LR} = p \circ C_{AF} = \left\{ \mathbf{x} \mapsto p\big(\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x} \rangle\big) \;\middle|\; \mathbf{x} \in \mathcal{X}, \boldsymbol{\theta} \in \Theta \right\}. \tag{18}$$

Hence, a LR model $g_{\boldsymbol{\theta}} \in C_{LR}$ is the function expressed as

$$g_{\boldsymbol{\theta}}(\mathbf{x}) \stackrel{\text{def}}{=} \left(1 + e^{-(\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x} \rangle)}\right)^{-1}. \tag{19}$$

As the LR model outputs a continuous value between zero and one, it allows for the effective modeling of the (conditional) *probability* that $\mathbf{x} \in \mathcal{X}$ is a class 1 instance. In this sense, observe that Eq. (16) expresses the logit or log-odds of the probability, which is the logarithm of the odds that $\mathbf{x}$ belongs to class 1. Therefore, the logistic model is often also referred to as logit model.

Now, this allows defining the *ProfLogit* classifier in its most general form.

Definition 3.4 (ProfLogit: Interpretation of Def. 3.3). Given the classifier class C_{LR} (18) and data set $D^n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, *ProfLogit* is the LR model $g_{\boldsymbol{\theta}} \in C_{LR}$ with parameter vector $\boldsymbol{\theta} \in \Theta$ that maximizes the *REEM* objective function:

$$ProfLogit = \underset{g_{\boldsymbol{\theta}} \in C_{LR}}{\arg\max}\ EMP_n(g_{\boldsymbol{\theta}}) - \lambda R(g_{\boldsymbol{\theta}}). \tag{20}$$

With Def. 2.4 and Def. 3.4, we obtained the *ProfLogit* classifier for churn.

Definition 3.5 (ProfLogit for Churn). *ProfLogit* for customer churn prediction (CCP) is the interpretation of Def. 3.4 with the *EMP* measure defined in Eq. (5):

$$ProfLogit^{ccp} = \underset{g_{\boldsymbol{\theta}} \in C_{LR}}{\arg\max}\ EMP_n^{ccp}(g_{\boldsymbol{\theta}}) - \lambda R(g_{\boldsymbol{\theta}}) \tag{21}$$

with $EMP_n^{ccp}: C_{LR} \to \mathbb{R}_{\geq 0}$ being the empirical *EMP* for CCP.

In [40], the authors demonstrated the practical application of $ProfLogit^{ccp}$ in which the regularizer $R(g_{\boldsymbol{\theta}})$ is specified to be the lasso penalty $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^d |\beta_j|$, accompanied by the soft-thresholding operator [17]: $S_\lambda(\beta) = \text{sign}(\beta)(|\beta| - \lambda)_+, \forall \beta \in \boldsymbol{\beta}$. They applied a RGA to solve the $ProfLogit^{ccp}$ optimization problem, for which the predictors in D^n were standardized to have zero mean and unit variance.

4 EMPIRICAL STUDY

In this section, we demonstrate the applicability of the *REEM* framework for customer churn prediction through *ProfLogit*. In §3.2, we list the four general requirements for *REEM*. Three out of the four requirements are specified by Def. 3.5. The final choice to make is with regard to the DFO method.

4.1 Experimental Setup

The goal of this comparative study is to determine the DFO method from Table 1 that yields the highest EMP^{ccp} performance. We thereby execute the DFO methods to solve the *REEM* problem on $q = 6$ real-world churn data sets that are available to our research group (Table 2). These data sets were also used for the empirical studies in [19, 39, 40]. A detailed description of the data preprocessing steps is given in [40]. Table 2 also provides an overview of the data-dependent settings for the study. That is, for all population-based DFO methods, the size of the population is determined through the expression $10(1 + d)$, where $d \in \mathbb{N}_1$ is the number of predictor variables added by one to account for the intercept. The budget, or number of allowed function evaluations, is also set based on the dimensionality of the data set through the formula $100(1 + d)^2$, as also done in [1]. The quadratic term is used because it is expected that problems with higher dimension are more difficult to solve than those with lower dimension [1].

As for the DFO methods, we apply the default values of the respective software implementations, as it is typically done for such comparative studies (see, e.g., [26]). The only exception is with regard to the DE algorithm, where we switch on the "classic DE" through the `DE/rand/1/bin` strategy. That is because we deem it to be more appropriate to use the classic DE as it has less greedy search properties than the default `DE/best/1/bin` strategy. For the two DFO methods (i.e., CMAES and BH) that require a starting point, we use the zero vector with the adequate number of elements. The search space for all DFO methods is set to be $\Theta = [-3, 3]^{d+1}$. Due to regularization in *REEM*, it is expected that the optimal solution is located close to the zero vector so that a bound of $[-3, 3]$ in each dimension is considered appropriate. The initial standard deviation required for CMAES is set such that 99.7% (i.e., 3σ's from the mean) of the one-dimensional normal distribution covers the interval $[-3, 3]$.

A stratified 5×2 cross-validation procedure is performed, as this procedure allows a reliable estimation of the average classification performance, producing $n_{cv} = 10$ performance estimates per DFO method and per data set.

The optimal value for the regularization parameter, λ, in Def. 3.5 is determined via grid search, where the grid is generated in the exact same way as described in [40]. That is, the grid of 15 equidistant candidate values is defined by

$$\Lambda \stackrel{\text{def}}{=} \left\{ \lambda \;\middle|\; \lambda_{min} < \lambda < \lambda_{max} \right\}, \tag{22}$$

where $\lambda_{max} = \max_j \left|\frac{1}{n}\langle \mathbf{x}_j, \mathbf{y}\rangle\right|$, $\lambda_{min} = \epsilon \lambda_{max}$, and $\epsilon = 0.1$. Note that for the calculation, the j-th predictor variable, which is the vector $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^\intercal$, is standardized to have zero mean and unit variance, $\forall j \in [d]$, and $\mathbf{y} = (y_1, \ldots, y_n)^\intercal$ denotes the response vector. The optimal value for the regularization parameter corresponds to the $\lambda \in \Lambda$ that exhibits the highest, average EMP^{ccp} performance obtained through the stratified 5×2 cross-validation.

The repeated cross-validation is also a premise to carry out the Bayesian hierarchical correlated t test [3, 7]. The Bayesian hierarchical correlated t test is an adequate, state-of-the-art statistical test that jointly analyzes $q \times n_{cv} = 60$ cross-validated performance differences of two classifiers evaluated on multiple data sets. Thanks to the Bayesian hierarchical model applied in the test, the average performance difference between the two classifiers is estimated more accurately due to the shrinkage effect than with a traditional maximum likelihood estimation approach. More importantly, the test returns three posterior probabilities which can be interpreted in a Bayesian manner, where two of them are the estimated probabilities that one classifier is superior over the other, and the remaining one is the probability of both classifiers being practically equivalent. Concrete examples of the test are given in the next section.

4.2 Results of the Comparative Study

Given our experimental setup, fifteen Bayesian hierarchical correlated t tests are performed in total, one test for each pairwise comparison of the six DFO methods considered. As a result, 45 posterior probabilities are available for interpretation. Note that

Table 2: Overview of Real-World Churn Data Sets and Data-Dependent Settings.

ID	Source	#Predictors	#Observations	Churn rate	Population size	Budget
		d	n	[%]	$10(1+d)$	$100(1+d)^2$
D1	Duke	11	12,499	39.32	120	14,400
O1	Operator	37	7,056	29.14	380	144,400
O2	Operator	8	889	31.16	90	8,100
O3	Operator	11	13,601	22.59	120	14,400
O4	Operator	9	3,698	13.28	100	10,000
UCI	UCI [a]	11	5,000	14.14	120	14,400

[a] http://www.sgi.com/tech/mlc/db

with the Bayesian approach we are not required to apply any sort of correction for multiple comparisons. We first describe three distinct outcomes of the statistical test before presenting all 45 probabilities.

The outcome of the Bayesian hierarchical correlated t test can be visualized via a probability simplex (Figure 1). Each simplex shows the posterior distribution obtained from the Bayesian hierarchical model that estimates the mean difference in EMP^{ccp} performance between two DFO methods evaluated on $q = 6$ real-world churn data sets. The posterior provides us with information about the estimation uncertainty, magnitude of effect size, practical difference and equivalence.

The corresponding three posterior probabilities returned by the test are shown next to the simplexes. From top to bottom, the top value is the posterior probability that the left DFO method is practically better than the right DFO method. The middle value is the posterior probability of the *region of practical equivalence* (ROPE), indicating the likelihood that the difference in EMP^{ccp} performance between the two DFO methods has no practical impact. In other words, if the majority of the posterior is in the ROPE, the performance difference between two considered DFO methods is said to be negligibly small and has no practical implication. Note that a ROPE value of 0.1, instead of the default 0.01, is regarded as more appropriate for the EMP^{ccp} measure. The bottom value is the posterior probability that the right DFO method is practically better than the left DFO method. As a hard rule, a statistically significant result is obtained if one of the three probabilities exceeds 95%.

The test result in Figure 1a reveals that *RGA is practically better than PRS* with a probability of 99.6%, a significant outcome since it exceeds the 95% threshold. Figure 1b shows that the majority of the posterior (90%) is in the region in favor of RGA, although not a significant result according to the hard rule. In this case, we compute the *posterior odds* as follows: $o(\text{left}, \text{right}) = p(\text{left})/p(\text{right})$, where $p(\text{left})$ and $p(\text{right})$ correspond to the posterior probability of the left and right DFO method, respectively. Corani et al. [7] proposed an interpretation for the posterior odds which is shown in Table 3. Thus, for the test in Figure 1b, the posterior odds is $o(\text{RGA}, \text{DE}) = 0.9005/0.0915 = 9.8$, indicating *positive evidence for RGA*. The test result in Figure 1c shows that most of the posterior mass lies in the ROPE. In other words, with a probability of 87%, there is *no practical difference between RGA and CMAES* in terms of achieved EMP^{ccp} performance.

The posterior probabilities of all fifteen Bayesian hierarchical correlated t tests are summarized in a matrix (Figure 2). Values on the diagonal of the matrix are the average ranks of the respective DFO method computed based on the EMP^{ccp} performance over the six data sets. The white boxes in each off-diagonal cell of the matrix contain three values: the upper value is the posterior probability than the DFO method mentioned on the left performs practically better than the DFO method mentioned on the top, the lower left value is the posterior probability of the ROPE, and the lower right value is the posterior odds if none of the three probabilities exceeds 95%, otherwise there is a dash.

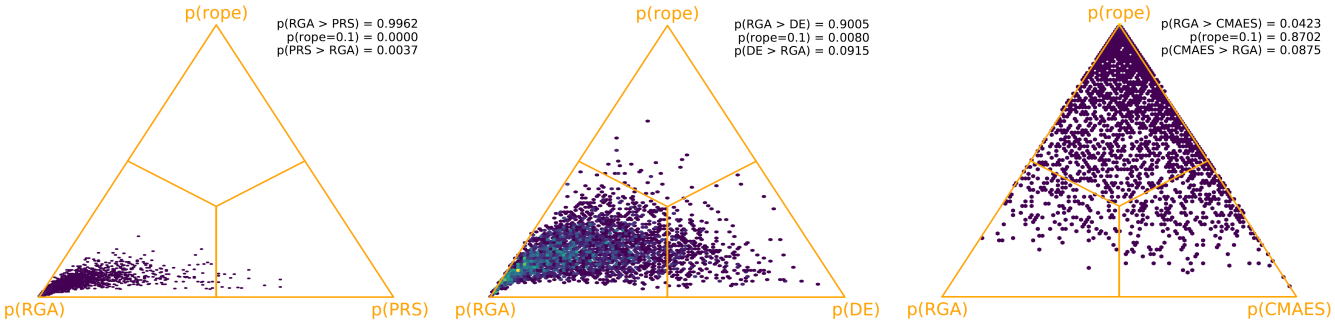
Table 3: Grades of Evidence for Posterior Odds [7].

Posterior odds	Evidence
1-3	*weak*
3-20	*positive*
> 20	*strong*

Evidently, RGA and CMAES share both the top position, dominating the other DFO methods either with high probability (i.e., with at least 85% over DE, PRS, and BH) or with positive evidence (i.e., with at least 6.4 over PSO). The difference between DE and PSO is less conclusive, yet there is positive evidence of 3.8 in favor of PSO. PRS and BH are the clearly inferior DFO methods for solving the *REEM* problem. There is even no practical difference between PRS and BH, since the probability of the ROPE (37%) is the largest.

4.3 Discussion of the Results

Despite its relatively simple design, RGA is capable of competing with the more sophisticated evolutionary algorithm CMAES. Hence, the analysis performed in §4.2 not only confirms but also provides statistical evidence that the RGA choice by the authors in [40] is appropriate for solving the *REEM* problem in Def. 3.5. The Bayesian hierarchical correlated t test is a statistical tool that helps users to draw meaningful conclusions when comparing two methods on *multiple* data sets. Here, statistical evidence supports that RGA and CMAES are *practically equivalent in general*. Thus, given our experimental setup and a high ROPE probability of 87%, it is—generally speaking—irrelevant whether RGA or CMAES is applied to build $ProfLogit^{ccp}$. However, the final choice between the DFO method depends on the data, as it may also be evident by the equal average ranks. That is because of the practical equivalence, users are encouraged to try out both DFO methods on their data. It can occur—although with low probability—that on a given data set one



(a) Definite statistical significance for RGA. **(b) Positive evidence for RGA.** **(c) RGA practically equivalent to CMAES.**

Figure 1: Three distinct results of the Bayesian hierarchical correlated t test based on EMP^{ccp} performance with ROPE = 0.1.

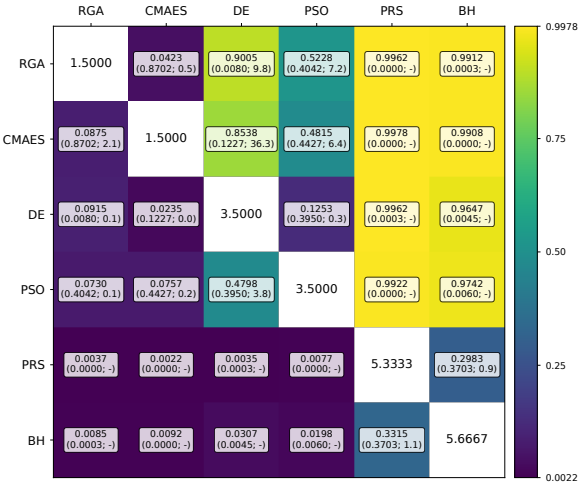


Figure 2: Posterior probability matrix of the 15 Bayesian hierarchical correlated t tests performed on basis of the EMP^{ccp} with ROPE = 0.1 on six real-world churn data sets.

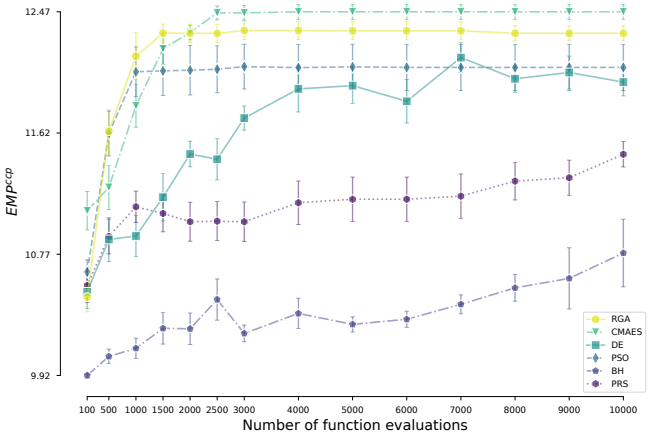


Figure 3: EMP^{ccp} performance as a function of allowable number of function evaluations on the O2 data set.

DFO method significantly outperforms the other, an outcome we observed in our analysis as well (results not shown).

To gain deeper insights, we compute the EMP^{ccp} performances as a function of the budget on the O2 data set (Figure 3). The y-coordinates of the points at each number of function evaluations are the means obtained from the 5×2 cross-validation accompanied by error bars (standard error of the mean). Evidently, RGA raises very quickly to a top performance, which is as good as CMAES's. In fact, when only considering the O2 data set and the experimental setup in §4.1, the ROPE exhibits the largest posterior probability (56%), indicating that RGA and CMAES are practical equivalent on O2. However, when the budget is low (i.e., 100 function evaluations), observe that CMAES yields a higher average performance than RGA. Thus, if a budget of $100(1+d)^2$ is acceptable, we recommend RGA as the default choice as it is has a simpler EA design than CMAES and still is an adequate DFO method for solving the considered optimization problem.

Statistical evidence that BH is practically equivalent to PRS with a probability of 37% makes it clear that the *REEM* problem in Def. 3.5 is an inherently complex optimization problem. Its average rank of 5.67 indicates that it is more often positioned behind PRS (average rank: 5.33), our de facto simplest DFO method under consideration. This indicates that the objective function possesses many local optima in which the BH gets trapped, and it is difficult for the BH to jump to a position that is in the vicinity of the optimal solution, returning solutions worse than PRS. This is also evident from Figure 3.

5 CONCLUSIONS

In this paper, we introduced the general framework called *Regularized Empirical EMP Maximization* (*REEM*) for profit-driven model building (Def. 3.3), formally defining the necessary components for a profit-driven classifier. We demonstrated the applicability of the *REEM* framework for customer churn prediction (Def. 3.5), which also entailed finding the best DFO method in terms of the profit-based measure EMP^{ccp} in (5). To do so, we found through the use of Bayesian hierarchical correlated t tests that RGA is a suitable choice. The objective behind the *REEM* framework is to be a generalization of previously proposed profit-driven classifiers [19, 39, 40] and be a formal guide for new profit-driven classification models we plan to develop in the future.

ACKNOWLEDGMENTS

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] M. Montaz Ali, Charoenchai Khompatraporn, and Zelda B. Zabinsky. 2005. A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems. *Journal of Global Optimization* 31, 4 (2005), 635–672. https://doi.org/10.1007/s10898-004-9972-2

[2] R. S. Anderssen. 1972. Global Optimization. *R. S. Anderssen, L. S. Jennings, D. M. Ryan, Optimization, University of Queensland Press* (1972).

[3] Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. 2017. Time for a Change: a Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis. *arXiv preprint arXiv:1606.04316* (2017).

[4] Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. 2005. Theory of Classification: a Survey of Some Recent Advances. *ESAIM: Probability and Statistics* 9 (2005), 323–375. https://doi.org/10.1051/ps:2005018

[5] Cristian Bravo, Seppe vanden Broucke, and Thomas Verbraken. 2017. *EMP: Expected Maximum Profit Classification Performance Measure.* R package version 2.0.2.

[6] Samuel H. Brooks. 1958. A Discussion of Random Methods for Seeking Maxima. *Operations Research* 6, 2 (1958), 244–251. https://doi.org/10.1287/opre.6.2.244

[7] Giorgio Corani, Alessio Benavoli, Janez Demšar, Francesca Mangili, and Marco Zaffalon. 2017. Statistical Comparison of Classifiers Through Bayesian Hierarchical Modelling. *Machine Learning* (2017), 1–21. https://doi.org/10.1007/s10994-017-5641-9

[8] Swagatam Das, Sankha Subhra Mullick, and Ponnuthurai N. Suganthan. 2016. Recent Advances in Differential Evolution–An Updated Survey. *Swarm and Evolutionary Computation* 27 (2016), 1–30. https://doi.org/10.1016/j.swevo.2016.01.004

[9] Swagatam Das and Ponnuthurai N. Suganthan. 2011. Differential Evolution: a Survey of the State-Of-The-Art. *IEEE Transactions on Evolutionary Computation* 15, 1 (2011), 4–31. https://doi.org/10.1109/TEVC.2010.2059031

[10] L. C. W. Dixon. 1978. *Global Optima Without Convexity.* Technical Report. Numerical Optimisation Centre, Hatfield Polytechnic.

[11] David J. Hand. 2009. Measuring Classifier Performance: a Coherent Alternative to the Area Under the ROC Curve. *Machine Learning* 77, 1 (2009), 103–123.

[12] Nikolaus Hansen. 2006. *The CMA Evolution Strategy: a Comparing Review.* Springer-Verlag Berlin Heidelberg, 75–102. https://doi.org/10.1007/3-540-32494-1_4

[13] Nikolaus Hansen. 2016. The CMA Evolution Strategy: a Tutorial. *CoRR* abs/1604.00772 (2016). http://arxiv.org/abs/1604.00772

[14] Nikolaus Hansen and Andreas Ostermeier. 1996. Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: the Covariance Matrix Adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation.* IEEE, 312–317. https://doi.org/10.1109/ICEC.1996.542381

[15] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9, 2 (2001), 159–195. https://doi.org/10.1162/106365601750190398

[16] T. Hastie, R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer New York. https://web.stanford.edu/~hastie/ElemStatLearn/

[17] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. 2015. *Statistical Learning with Sparsity: The Lasso and Generalizations.* CRC Press.

[18] John H. Holland. 1975. Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. *Ann Arbor, MI: University of Michigan Press* (1975).

[19] Sebastiaan Höppner, Eugen Stripling, Bart Baesens, Seppe vanden Broucke, and Tim Verdonck. 2017. Profit-Driven Decision Trees for Churn Prediction. *European Journal of Operational Research* (2017). Under Review.

[20] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–. SciPy: Open Source Scientific Tools for Python. http://www.scipy.org/ [Online; accessed December 2017].

[21] James Kennedy and Russell Eberhart. 1995. Particle Swarm Optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4. https://doi.org/10.1109/ICNN.1995.488968

[22] Zhenqin Li and Harold A. Scheraga. 1987. Monte Carlo-Minimization Approach to the Multiple-Minima Problem in Protein Folding. *Proceedings of the National Academy of Sciences* 84, 19 (1987), 6611–6615.

[23] Michael A. Lones. 2014. Metaheuristics in Nature-Inspired Algorithms. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation.* ACM, 1419–1422. https://doi.org/10.1145/2598394.2609841

[24] Jose A. Lozano, Pedro Larrañaga, Iñaki Inza, and Endika Bengoetxea (Eds.). 2006. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms* (1 ed.). Vol. 192. Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/3-540-32494-1

[25] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of Machine Learning.* MIT Press.

[26] Katharine Mullen. 2014. Continuous Global Optimization in R. *Journal of Statistical Software* 60, 6 (2014), 1–45. https://doi.org/10.18637/jss.v060.i06

[27] Travis E. Oliphant. 2007. Python for Scientific Computing. *Computing in Science & Engineering* 9, 3 (2007), 10–20. https://doi.org/10.1109/MCSE.2007.58

[28] Adam P. Piotrowski. 2017. Review of Differential Evolution Population Size. *Swarm and Evolutionary Computation* 32 (2017), 1–24. https://doi.org/10.1016/j.swevo.2016.05.003

[29] Riccardo Poli, James Kennedy, and Tim Blackwell. 2007. Particle Swarm Optimization: an Overview. *Swarm Intelligence* 1, 1 (2007), 33–57. https://doi.org/10.1007/s11721-007-0002-0

[30] Kenneth Price, Rainer M. Storn, and Jouni A. Lampinen. 2006. *Differential Evolution: a Practical Approach to Global Optimization.* Springer Berlin Heidelberg.

[31] Anguluri Rajasekhar, Nandar Lynn, Swagatam Das, and Ponnuthurai N. Suganthan. 2017. Computing with the Collective Intelligence of Honey Bees–A Survey. *Swarm and Evolutionary Computation* 32 (2017), 25–48. https://doi.org/10.1016/j.swevo.2016.06.001

[32] Luis Miguel Rios and Nikolaos V. Sahinidis. 2013. Derivative-Free Optimization: a Review of Algorithms and Comparison of Software Implementations. *Journal of Global Optimization* 56, 3 (2013), 1247–1293. https://doi.org/10.1007/s10898-012-9951-y

[33] H. Edwin Romeijn. 2009. *Random Search Methods.* Springer US, Boston, MA, 3245–3251. https://doi.org/10.1007/978-0-387-74759-0_556

[34] Bernhard Schölkopf and Alexander J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT press.

[35] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: from Theory to Algorithms.* Cambridge University Press.

[36] Yuhui Shi and Russell Eberhart. 1998. A Modified Particle Swarm Optimizer. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation.* IEEE, 69–73. https://doi.org/10.1109/ICEC.1998.699146

[37] Marina Sokolova and Guy Lapalme. 2009. A Systematic Analysis of Performance Measures for Classification Tasks. *Information Processing & Management* 45, 4 (2009), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

[38] Rainer Storn and Kenneth Price. 1997. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11, 4 (1997), 341–359. https://doi.org/10.1023/A:1008202821328

[39] Eugen Stripling, Seppe vanden Broucke, Katrien Antonio, Bart Baesens, and Monique Snoeck. 2015. Profit Maximizing Logistic Regression Modeling for Customer Churn Prediction. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (19-21 October 2015). IEEE, Paris, France, 1–10.

[40] Eugen Stripling, Seppe vanden Broucke, Katrien Antonio, Bart Baesens, and Monique Snoeck. 2017. Profit Maximizing Logistic Model for Customer Churn Prediction Using Genetic Algorithms. *Swarm and Evolutionary Computation* (2017). https://doi.org/10.1016/j.swevo.2017.10.010 In Press.

[41] Wouter Verbeke, Karel Dejaeger, David Martens, Joon Hur, and Bart Baesens. 2012. New Insights into Churn Prediction in the Telecommunication Sector: a Profit Driven Data Mining Approach. *European Journal of Operational Research* 218, 1 (2012), 211–229. Issue 1. https://doi.org/10.1016/j.ejor.2011.09.031

[42] Tomas Verbraken. 2013. *Business-Oriented Data Analytics: Theory and Case Studies.* Ph.D. Dissertation. Department of Decision Sciences and Information Management, Faculty of Economics and Business, KU Leuven, Leuven, Belgium. Ph.D. Dissertation.

[43] Thomas Verbraken, Cristián Bravo, Richard Weber, and Bart Baesens. 2014. Development and Application of Consumer Credit Scoring Models Using Profit-Based Classification Measures. *European Journal of Operational Research* 238, 2 (2014), 505–513. https://doi.org/10.1016/j.ejor.2014.04.001

[44] Thomas Verbraken, Stefan Lessmann, and Bart Baesens. 2012. Toward Profit-Driven Churn Modeling with Predictive Marketing Analytics. In *Cloud Computing and Analytics: Innovations in E-business Services. Workshop on E-Business (WEB2012).*

[45] Thomas Verbraken, Wouter Verbeke, and Bart Baesens. 2013. A Novel Profit Maximizing Metric for Measuring Classification Performance of Customer Churn Prediction Models. *IEEE Transactions on Knowledge and Data Engineering* 25, 5 (2013), 961–973.

[46] David J. Wales and Jonathan P. K. Doye. 1997. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms. *The Journal of Physical Chemistry A* 101, 28 (1997), 5111–5116. https://doi.org/10.1021/jp970984n

[47] David H. Wolpert and William G. Macready. 1997. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation* 1, 1 (1997), 67–82. https://doi.org/10.1109/4235.585893

[48] Yudong Zhang, Shuihua Wang, and Genlin Ji. 2015. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Mathematical Problems in Engineering* 2015 (2015). https://doi.org/10.1155/2015/931256