Complex & Intelligent Systems (2021) 7:41–59
https://doi.org/10.1007/s40747-020-00169-w

ORIGINAL ARTICLE

Optimal subset selection for causal inference using machine learning ensembles and particle swarm optimization

Dhruv Sharma[1] · **Christopher Willy**[2] · **John Bischoff**[3]

Received: 29 July 2018 / Accepted: 16 June 2020 / Published online: 2 July 2020

Abstract

We suggest and evaluate a method for optimal construction of synthetic treatment and control samples for the purpose of drawing causal inference. The balance optimization subset selection problem, which formulates minimization of aggregate imbalance in covariate distributions to reduce bias in data, is a new area of study in operations research. We investigate a novel metric, cross-validated area under the receiver operating characteristic curve (AUC) as a measure of balance between treatment and control groups. The proposed approach provides direct and automatic balancing of covariate distributions. In addition, the AUC-based approach is able to detect subtler distributional differences than existing measures, such as simple empirical mean/variance and count-based metrics. Thus, optimizing AUCs achieves a greater balance than the existing methods. Using 5 widely used real data sets and 7 synthetic data sets, we show that optimization of samples using existing methods (Chi-square, mean variance differences, Kolmogorov–Smirnov, and Mahalanobis) results in samples containing imbalance that is detectable using machine learning ensembles. We minimize covariate imbalance by minimizing the absolute value of the distance of the maximum cross-validated AUC on M folds from 0.50, using evolutionary optimization. We demonstrate that particle swarm optimization (PSO) outperforms modified cuckoo swarm (MCS) for a gradient-free, non-linear noisy cost function. To compute AUCs, we use supervised binary classification approaches from the machine learning and credit scoring literature. Using superscore ensembles adds to the classifier-based two-sample testing literature. If the mean cross-validated AUC based on machine learning is 0.50, the two groups are indistinguishable and suitable for causal inference.

Keywords Analytics · Evolutionary computing · Swarm optimization · Machine learning

Introduction

There is a lack of experimental data in many domains but there is widely available non-experimental observational data. Observational data (e.g., from surveys, internet, big data, etc.) is non-experimental and lacks the benefit of balance on observables and unobservables due to randomization. Moving beyond correlation to causal inference is an important area of study especially given the use of data science to develop unbiased algorithms and decision analytics using observational data. Biased algorithms can result in harm/costs for society. Using observation data without balance can lead to bias in inference and estimation.

Problem statement Often engineering managers are faced with given non-experimental data which can contain imbalance that can lead to bias in assessing impact of decisions/treatments.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s40747-020-00169-w) contains supplementary material, which is available to authorized users.

✉ Dhruv Sharma
dhruvsharma@gwu.edu

Christopher Willy
cwilly@gwmail.gwu.edu

John Bischoff
jeb@email.gwu.edu

[1] George Washington University, 2023 N. Cleveland St, Arlington, VA 22201, USA

[2] George Washington University, 21610 South Essex Drive, Lexington Park, MD 20653, USA

[3] George Washington University, 203 Greenhow Ct Se, Leesburg, VA 20175, USA

Thesis statement Machine learning can be used to detect imbalance in samples, which existing methods (Chi-square, mean/variance differences, Mahalanobis distance, Kolmogorov–Smirnov distance) can miss. Optimizing balance using machine learning and particle swarm optimization can result in data sets with less imbalance.

The objective of this study is to take a non-experimental data set, resample it, introduce balance, and make accurate causal inferences from the data. Samples collected without randomized procedures are observational samples. These include surveys, internet behavioral data, and any data recorded without the benefit of an experimental design. Such observational data suffers from confounding due to selection bias. The social cost and prevalence of bias in readily available data are large. Selection bias is prevalent in a variety of data sets, including gender discrimination in text data, inaccurate teacher evaluations, racial discrimination in models predicting crime, and criminal recidivism used to make legal decisions [42]. Using biased data to build algorithms can impose social costs by magnifying and amplifying social inequities [42]. To perform causal inference using observational data, where there is no experimental treated or control group, the group of observations exposed to a treatment of interest and a sample of most similar untreated observations are used as a synthetic experiment.

If covariates in the treatment and control sample are imbalanced, the researcher can observe differences due to covariate imbalance instead of the treatment and confuse the two effects. For example, if older males are more likely to select a certain treatment, then comparing older males who took a treatment to the entire population would show an effect related to being older and male instead of the actual treatment. This phenomenon is known as confounding. Balancing the data across covariates yields a control sample and treatment sample that would be similar enough to isolate the treatment effect.

We focus on the problem of detecting and reducing bias in data by improving balance in covariates in the data. This problem of selecting subsets of data, which are balanced along covariates, has been recently formulated in the operations research literature by Nikolaev et al. [41] and has contributed to the literature by re-formulating the balancing problem by measuring success at the sample or aggregate level. Prior to this work, methods to process samples for causal inference consisted of matching individual treated units to similar individual control units and was approached primarily by the statistics community. Recently, sample construction and machine learning have received more interest from the operations research community. The advantage of formulating balance at the sample level is that matching individual observations or units can result in infeasible matches, and focusing on the end goal of balance permits more feasible solutions to be identified. The problem of selecting balanced subsets from data has been shown to be NP-hard, which means that the problem cannot be solved in polynomial time, and the use of heuristic and nature inspired computing methods are appropriate to achieve solutions. However, finding the global best solution is computationally prohibitive given the large search space of possible solutions.

Scientists and operations researchers can move beyond correlational studies of observational data to study causal relationships using optimal sample selection techniques. The aim of optimal sample selection is to find 'an experimental data set hidden inside a non-experimental' biased data set [30]. The importance of the problem of isolating causal effects is fundamental to science. This problem can benefit from recent advances in optimization and machine learning. Traditional distance between samples has been measured via empirical differences based on assumptions in the data, such as the mean, variance, Mahalanobis-based measures, or discrete binning of variables and minimizing the differences in counts in categorical bins. Machine learning methods allow for automatic detection of patterns in data that classical tests may miss.

Unlike past efforts in this problem area, we focus on stochastic balance using binary classification approaches from machine learning and credit scoring. These fields have had extensive success in classification of two groups. We extend the literature by approaching the problem using tools from machine learning and nature inspired warm algorithms. We build on the success of the binary classification problem, which has been extensively studied in the credit scoring and machine learning communities, and propose to use the machine learning ensemble classification accuracy as a measure of sample balance. The central hypothesis of this work is that if machine learning-based ensembles can detect differences between treatment and control samples, then the samples are not balanced. Conversely, if both treatment and control samples are balanced, then machine learning algorithms for binary classification cannot distinguish them from one another. In this case, machine learning algorithms would not perform better than random chance or a fair coin toss between predicting treatment and control samples.

The research objectives of this study are as follows:

- To assess whether a machine learning can be used to detect imbalance in data sets
- To assess if machine learning can detect imbalance in data sets that are balanced using existing methods
  - i.e., to ascertain if existing balance methods are sub-optimal
- To determine if machine learning can be used to improve balance in data sets over existing balance optimization methods

The benefits of a machine learning-based measure can include transparency, objectivity, and automation only if the underlying data is free of bias. In addition, this measure provides useful information to the researcher about balance and bias within samples. Existing matching techniques for balance are subject to the decisions and assumptions of the researcher regarding appropriate metrics; they also assist in making judgment calls regarding acceptable levels of balance. In multivariate settings, a researcher cannot typically anticipate relationships between variables once they are permuted or interacted. The combinations of potential interactions between variables can grow exponentially. In addition, simple mean/variance empirical tests may be effective given assumptions on the functional form of the relationship. However, the method proposed here does not rely on a distribution.

Overview of the study

The outline of our paper is as follows: (1) we review the literature on causal inference and sample balancing, and (2) we propose a novel machine learning-based metric for sample balance based on the binary classification literature and the area under the receiver operating characteristic curves (AUCs). To demonstrate the effectiveness of the metric, we evaluate real and synthetic data sets. We optimize using popular balance metrics: discretized bin counts via Chi-square statistic, Mahalanobis distance, Kolmogorov–Smirnov statistic, and the sum of differences in standardized mean and variance for the K covariates between samples and compare how well machine learning can separate the treatment and control samples. Our results show that optimizing using these traditional measures does not guarantee balanced samples because machine learning successfully classifies the treatment and control units. Then, we attempt to directly optimize the machine learning-based AUC measure to minimize imbalance and show that this method results in a greater balance than that provided by existing methods. Given the non-linear, gradient-free metric of cross-validated AUCs from machine learning ensembles, we use swarm optimization algorithms to optimize the measures and show that particle swarm optimization outperforms modified cuckoo swarm optimization for the AUC-based measures.

This approach is novel because prior studies have optimized the distance between covariate distributions [9, 26, 56, 80]. Using machine learning methods, we show that minimizing the distance between covariate distributions to achieve balance using existing methods does not guarantee that treatment and control samples are 'statistically indistinguishable' [9]. The balance optimization subset selection (BOSS) problem is to choose a control sample S_c, which is a subset of the control sample C, to minimize imbalance from the treatment sample T. The objective of minimizing imbalance is evaluated at the sample level. There is no universal agreement on what level of balance is acceptable or what the metric for balance should be because these aspects are left to the judgment of data analysts [9, 26]. Matching can be seen as a non-parametric pre-processing process that makes 'it possible to greatly reduce the dependence of causal inferences on hard-to-justify, but commonly made, statistical modeling assumptions' Ho et al. [26]. Recent formulations of minimizing balance using means, variances, and discretizing variables and applying a Chi-square test statistic has been found to be effective [9]. The problem with this approach is that balancing using the mean and variance or cell counts in the multivariate space does not guarantee that the two samples are identical. An example of how this difference can occur is due to the differences in joint distributions of variables and interaction effects of variables.

Key contributions

We review the problem domain of causal inference and balanced sample selection and build on the machine learning and credit scoring literature to build a binary classification-based two-sample metric for optimization. We are the first to tackle this problem using cross-validated AUCs, which provides a method that is robust to overfitting, and the first to approach this subset selection problem using nature-inspired swarm methods. To use supervised machine learning, as a test, we develop an approach using ensembles so that the method is robust to different data sets, and we demonstrate the method on five real data sets and seven synthetic data sets. In addition to showing samples generated from existing methods to be suboptimal in terms of AUCs, we show that the AUC metric can be directly optimized using particle swarms, and particle swarms outperform modified cuckoo swarms for this objective function. Optimizing the AUC metric to 0.50 achieves statistical or stochastic balance. When this objective is not achieved or if there are empirical differences in distributions that persist even when the AUC metric is 0.50, bias can remain in the data. We show that coupling AUC optimization with a double robust inference, which involves a regression estimation using all covariates, eliminates all remaining bias. Overall, the AUC metric serves as a useful balance metric and diagnostic, and further insights can be gleaned by modeling the differences between the optimized sample and the original sample to obtain insight into the nature of bias.

Scope of study

The focus of the study is to construct optimally balanced samples for causal inference; thus, the data sets are of moderate size in relation to experimental research, psychology, bio/stats and economics/social science. To make our

approach scalable is a topic of future study. However, the scope of this study is to establish a viable solution. In addition, we focus on the causal inference, focused on bias minimization, as opposed to the covariate shift literature, where the focus is on generalizing classifiers built on biased samples by reweighting data to a target population [48, 60]. The problem is focused on causal inference and not on covariate shifts in sample populations. In addition, samples of less than 100 observations are beyond the scope of this work because the sample size is insufficient for reliable machine learning results. The scope of this work fits with data sets ranging from 100 to thousands of observations. The data sets used both real and synthetic are commonly used data sets with citations and representative of the array of problems that can benefit from such approaches.

The aim of the study is to operationalize a general approach in which machine learning is used to detect bias in data and in a general form automated technique that can be used across data sets without tuning. The focus is not on each individual classifier or tuning of performance. To improve and refine the classification accuracy further, we acknowledge can be done by tuning classifiers to data set or empirically testing to determine which classifier performs best in a particular data set at hand. The use of machine learning ensemble serves only to guard against poor model fit and use most common machine learning algorithms: logistic regression, support vector machines, stochastic gradient boosting and random forests. Past uses of classifier based two-sample tests suffer from relying on only one class of machine learning which may be poor on a given data set (for example Clemencon et al. [10] only using decision trees). The approach outlined here is an improvement to that literature of classification based two-sample testing.

Literature review

The literature review for the optimal sample selection problem cuts across the disciplines of causal inference, machine learning, and operations research as depicted in Fig. 1.

Balance subset selection optimization

Recently, the problem of optimal sample balancing was approached as an optimization problem of balancing sample-level covariate information [9, 55]. This problem area is important given the wide availability of observational data sets collected outside the experimental design. In addition, there has been a strong interest in machine learning and novel operational methods for improving causal inference [1, 5, 34]. BOSS minimizes the differences between the treatment and control samples for a data set D, which is a union of a treatment group, where $T = 1$, and a control sample C, where $T = 0$ [9, 41]. K is the number of covariates available for matching and excludes the treatment variable T and outcome variable Y.

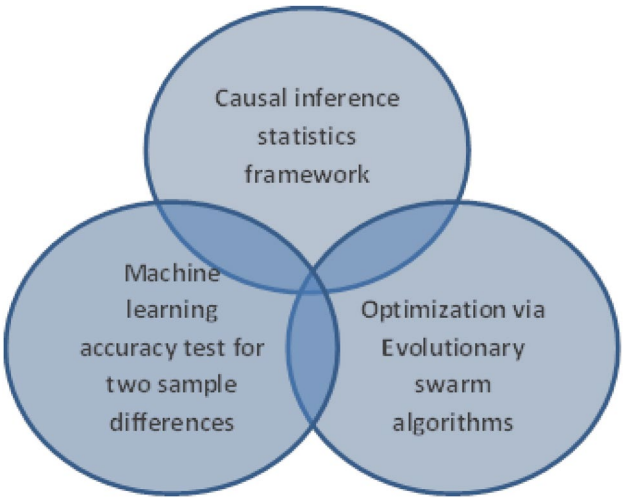

Fig. 1 Combined approach using operations research, machine learning, and statistics

Outcome values should not be used in the matching process to avoid forcing the samples to a 'desired result' [62]. The goal of matching procedures is to achieve samples that are 'statistically indistinguishable' [11, 49–52, 62]. Thus, the samples should be indistinguishable by ignoring the treatment variable T and the outcome variable Y. The sample S_t contains observation units with exposure to the treatment of interest, i.e., $T = 1$.

The causal effect of the treatment variable T on an outcome variable Y can be computed using the Rubin–Neyman causal model [51, 62]. The causal effect in this framework is estimated by computing the average treatment effect for T [51, 62]. One never knows $Y_i(S_t)$ and $Y_i(S_c)$ for a given T and for a given observational unit i. The essential problem of causal inference is that for each unit, only the treated or untreated outcomes are observed. Thus, the causal effect of T is computed as the average treatment effect, i.e., $E\left[Y_i(S_t) - Y_i(S_c)\right] = E\left[Y_i(S_t)\right] - E[Y_i(S_c)]$. [51, 62].

Achieving covariate balance reduces bias in the treatment effect of estimation because applying linear regression without a sufficient balance can lead to inaccurate estimates, especially if covariates have a non-linear relationship [62]. For regression to be reliable, the 'absolute standardized differences of means should be less than 0.25, and the variance ratios should be between 0.5 and 2' [62]. Most recently, Sauppe and Jacobson [54] showed that bias can be removed under certain functional forms within reasonable assumptions when 'empirical differences are removed' [54] using the balanced subset optimization (BOSS) approach. Their work indicates a greater balance

is needed for more complex functional forms of data, and there is a distinct trade-off between degree of balance and bias in certain functional forms [54]. Currently, the BOSS approach is being applied to empirical differences to optimize samples.

Appropriateness of swarm algorithms for the studied problem

Sauppe showed that balanced subset selection reduces to the NP-hard decision problem via the reduction to the 3-dimensional matching problem [53]. Swarm methods have also been found to be competitive in both accuracy and computation performance for discrete optimization [67]. Given the computational complexity of the balance subset problem, the use of evolutionary algorithms is appropriate and is an actively researched area for feature selection [8, 37, 38]. Among metaheuristics, a recent examination of the makespan problem found that swarm algorithms are the most frequently used metaheuristic [20, 63].

The state of the art and history of particle swarms was reviewed recently over the last quarter century showing a variety of applications and success [7]. Variants of PSO range from global vs local search balance, population diversity, hybridization of PSO and other swarms, multiple swarms and efficient learning. PSO has been used for single objective, multi-objective and multimodal multiobjective problems as well [7]. Most applications of PSO are hard problems of scheduling and data mining. Recent review by Bonyadi and Michalewicz found few applications of SPSO for constrained optimization [5]. Ab Wahab et al. [1, 7] conducted the most comprehensive review and benchmarking of swarm algorithms and found particle swarms of be one of the best performing optimization second only to differential evolution. Per the literature review of recent citations PSO is considered state of the art still.

Swarm algorithms are a form of nature inspired search. These algorithms are modeled after nature and use stochastic search to optimize global and local optimal solutions at the individual particle and global swarm levels [28, 29]. MCS has been shown to perform well against particle swarms, cuckoo algorithms, and differential evolution for high-dimensional real-world engineering problems [69]. MCS improves upon the original cuckoo algorithm modeled after cuckoo breeding behavior. Both variants of cuckoo algorithms use Levy flight distributions to make steps in the search space using stochastic global optimization; given enough computation time, this approach is guaranteed to converge to the global optimum [74, 75]. Swarm algorithms have been found to be effective for NP-hard problems, combinatorial problems, integer programming, and complex gradient-free non-linear objective functions [33, 63, 73].

AUC metric for the two-sample problem

The AUC metric has been used in machine learning and signal processing to assess how well statistical models can separate two groups. The AUC metric is an important measure for binary discrimination and includes the Kolmogorov–Smirnov (KS) statistic and D-concordance statistic because these statistics can be derived from the AUC metric [64]. Credit scoring is a domain where operations research methods and binary classification have been used to great success for the classification of groups where the receiver operating characteristic curve is widely used [64]. For comparing two samples, the null hypothesis is associated with an AUC metric of 0.5 [10]. Clemencon et al. [10] showed that the AUC metric is an efficient method to compare two samples using a training sample and a holdout set compared with the Mann–Whitney Wilcoxon test. Beling et al. [3] showed that dominance on the ROC space leads to dominance in profit maximization and can be used for multiobjective trade-offs. Lopes et al. [36] used ROCs to demonstrate the power of high-dimensional two-sample tests [36]. Lopes and Thulin's test for high-dimensional data with a small sample size are worth further investigation for optimization of small data sets, which is beyond the scope of this study [36, 65] because a sufficient sample size is necessary to train machine learning-based classifiers.

To use the AUC metric requires computing a model to classify the treatment and control samples [10]. Machine learning algorithms minimize error rate and maximize classification accuracy for binary classification. Cortes and Mohri showed that the 'AUC is a monotonically increasing function of classification accuracy' and is equivalent to the Wilcoxon–Mann–Whitney statistic [14]. The AUC metric is 'a measure based on pairwise comparisons between classifications of the two classes', which are samples in our context [14]. Thus, the AUC metric is a probability from 0 to 1, in which a 'classifier ranks a randomly chosen positive example higher than a negative example' [14]. The AUC metric is equal to the Wilcoxon-ranked sum and Mann–Whitney U statistic divided by the product of number of observations in the two classes being compared [14]. In our context, the samples are treatment vs. control subsets.

A given AUC value, called A, for a classifier is defined as follows: $A = \sum_{i=1}^{i=t} \sum_{j=1}^{j=c} 1_{x_t > x_c} / (tc)$, where x_t are the outputs from the classifier for the treatment sample, S_t and x_c are the output predictions for the control sample, and S_c and tc are the product of the number of observations in the treated sample and the number of observations in the

control sample, respectively. The numerator in the AUC calculation is the U statistic, i.e., the Wilcoxon rank-sum.

If the AUC metric of a classifier is 0.5, the samples are homogeneous, as shown by Clemencon et al. [10]. Under this condition, the model has no predictive power in separating the two groups based on the covariates and is not different from random guessing. If two samples are statistically indistinguishable, then the AUC metric of the optimal prediction algorithms should be 0.5 because there would no detectable patterns or differences in the samples for the algorithm to learn. The AUC metric for small samples can be highly variable, and the maximum cross-validated AUC is used as a target for the optimization to provide more stable results. The AUC metric also has the advantage of being non-parametric; thus, it is a distribution-free statistic that does not require assumptions of normality and is effective for small samples. Clemencon et al. [10] showed that the AUC metric, which was derived from machine learning, outperformed state-of-the-art multivariate tests, such as the reproducing kernel Hilbert space maximum mean discrepancy. Matsuoka showed that the U statistic is a kernel [39]. The AUC metric based on machine learning has been shown to be consistent and powerful for multidimensional two-sample testing [10].

The relationship between bias and balance metrics requires further study. Recent work has linked AUC improvements with bias reduction. Sauppe and Jacobson [54] have shown a trade-off between level balance and bias reduction based on appropriate functional forms and assumptions. Reduction in bias for estimates has been shown to be associated with improvements in the AUC metric [22, 79]. Moreover, the root-mean-square error has been show to increase as the AUC metric increases from 0.6 to 0.9 and the rate of change in error increases from 0.6 to 0.7 and above [79]. Franklin et al. [22] performed exhaustive tests of potential metrics for covariate balance in the context of causal matching using propensity scores and found that the AUC metric, referred to as the c-statistic in some literature, followed by the difference in standardized means were the most effective in reducing bias in causal estimates. The c-statistic is the AUC metric for a logistic regression model and has the best performance in terms of bias reduction.

Review of machine learning classifiers for binary classification

Logistic regression is the most popular binary classification method used in credit scoring [64]. A random forest is an ensemble of recursive partitioning decision trees trained on bootstrapped samples that use different random subsets of variables in each run, and the final prediction is then the aggregate result of the individual trees [68]. Random forest methods can detect interaction effects that otherwise would have to be specified as constraints on an ad hoc basis [59]. Random forests tend to outperform deep nets for binary classification, while deep neural nets tend to outperform for multi-label outcomes, such as image detection and object recognition [27, 31]. Simply modeling interactions and balancing across all interactions is likely to result in overfitting [23]. Support vector machines (SVMs) have been successful in various domains and maximize the distance from the margin separating classes, provided such a hyperplane exists for classification. SVMs search for the maximum margin, where the margin is defined by the 'shortest distance between the closest points in the data to a hyperplane', which yields the best generalization ability for the support vectors [15, 73].

In addition, based on Wolpert's no-free lunch theorem, no classifier can be optimal in all data sets [19]. For example, under some circumstances, SVMs perform better than random forests [67]. Finlay's study of multiple classifier architectures showed that bagging and boosting algorithms outperform other multiclassifier systems [21]. Combining heterogeneous classifiers is an effective technique for achieving optimal predictive performance [21, 35]. The stochastic gradient boosting algorithm (SGB) improves on the original adaptive bootstrapping method using a 'random permutation sampling strategy'. In SGB, weak learners are iteratively built and weighted toward observations misclassified by prior learners. In addition, SGB uses regularization and performs a combination of bagging and boosting [16].

The advantage of using a machine learning-based sample comparison is that machine learning can detect subtle patterns in data that simple mean, variance, or moment-matching methods may miss. For example, consider the case of a covariate in one sample that is a univariate random variable ranging from -1 to 1 and another sample drawn from a normal distribution with a mean of 0 and a variance of 1/3 [39]. For both variables, the mean is 0 and the variance is 1/3. The use of classical mean and variance tests would make the samples appear indistinguishable despite important differences [21]. Another example that classical tests would not detect is that of conditional relationships between variables. If a conditional relationship or dependency exists between the variables or within certain ranges of variables, this relationship would not be uncovered by analyzing the overall mean, variances, and correlations that can still be similar.

Potential interactions between covariates can grow exponentially and are difficult to infer using human judgment. Fortunately, machine learning and pattern recognition techniques can automatically detect these interactions [19]. To make machine learning a convincing tool for two-sample comparisons, we propose an ensemble of different algorithms because no one machine learning method works best on all data sets [19].

Novel balance metric: the cross-validated AUC metric

The aim of this study is to consider a new balance metric for the optimization and to consider direct optimization on this novel metric. We propose the use of the well-studied AUC metric to measure whether candidate samples S_t and S_c are statistically indistinguishable.

Machine learning ensembles are 'well established … method(s) for obtaining a highly accurate classification' when learners are diverse and uncorrelated [18]. The predictions of all classification models, including ensembles, can be combined using a super scorecard ensemble model, which uses the product of predicted probabilities that result from individual classifiers [24]. This ensemble of ensembles, called a super scorecard ensemble by Hand and Kelly, is used to maximize the classification accuracy and the AUC metric; it is then evaluated using cross-validation for M folds. The data sample is split into M mutually exclusive samples called folds, and each fold is used as a test data set while keeping $M-1$ folds for model training purposes. The proposed approach to study the problem is an extension of the balance optimization subset selection (BOSS) approach [41].

The proposed cross-validated AUC metric for machine learning ensembles extends on the BOSS approach and shares the benefit of 'automatically evaluating balance on all covariates simultaneously and interactions among covariates' [22]. Unlike prior work on optimizing empirical differences by Sauppe and Jacobson [54], we optimize stochastic balance in expectation using machine learning. Optimizing the AUC metric yields a local average treatment effect because the treatment and control samples that are imbalanced are dropped [47].

Proposed approach: how BOSS is extended

Figure 2 shows the original BOSS approach, and Fig. 3 highlights how our work extends the BOSS literature. Our approach results in an estimation of a local average treatment effect because the proposed approach can drop the treated samples that are not balanced with the control samples. In addition, we use stochastic optimization to construct samples and produce treatment effects using double robust inference methods on the constructed samples.

To compute the AUC metric, the following state-of-the-art algorithms are used in this study: random forest, stochastic gradient boosting, logistic regression, and support vector machines. These algorithms have been recently studied in Lessmann et al. [35]. The ensemble is combined using Hand and Kelly's [24] super scorecard approach.

The novel idea of using machine learning to maximize the AUC metric given a data set, which is then minimized by subsetting the data, allows for automated balancing of the samples. Automated balancing is desirable because searching for interaction effects involves a difficult search space, and the current practice is to allow researchers to develop models with ad hoc interaction effects. The lack of agreement on acceptable levels of distributional differences and constraints on matching using judgment leads the state of models to be ad hoc. The AUC performance on the samples is an objective measure that simultaneously captures many distinguishable differences across the samples and makes a better choice for measuring balance. Allowing machine learning to build maximum predictive models given the data also avoids the human element in the modeling step.

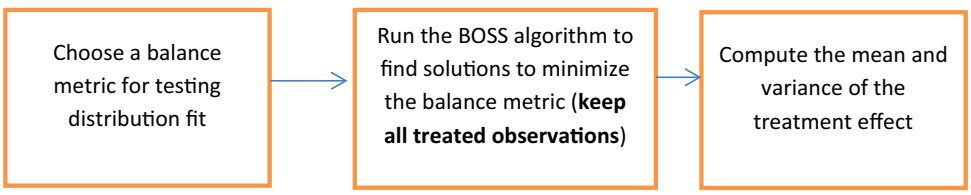

Fig. 2 BOSS framework [41]

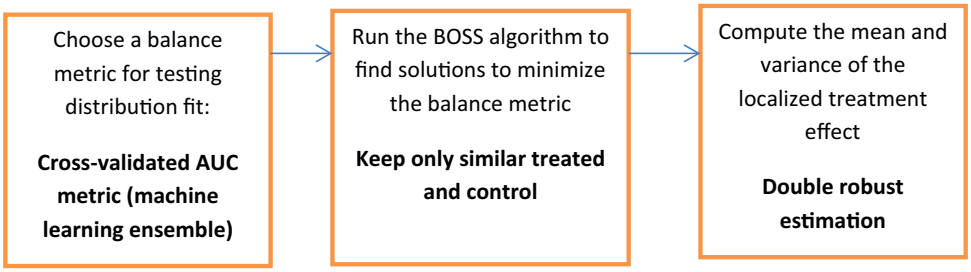

Fig. 3 Extension of BOSS using machine learning ensembles and evolutionary optimization

Optimization approach using particle swarm optimization

Both PSO and the recent MCS were used to optimize the proposed AUC function. The swarms operate on a decision vector representing the observations to keep in the sample, candidate samples. The swarm agents generate candidate samples using continuous variables between 0 and 1 which are thresholded above 0.50 to generate the sample to be kept. As the swarms evolve solutions the candidate samples are evaluated via cost fitness functions including a machine learning based cost metric and other traditional metrics described later.

The results showing PSO to dominate MCS are in Supplementary Appendix C. The cross-validated AUC metric produced by stacked ensembles is a complex non-linear gradient-free function. Therefore, the use of a metaheuristic optimization method is practical, especially given that this is an NP-hard problem [41, 55].

PSO evolves particle position and velocity over time toward each particle's best solution thus far and the global swarm best solution. The particle positions and velocities are updated iteratively via equations based on the following velocity equation, and random variation is based on uniform random variables between 0 and 1. Where $v_{i,k}$ is the velocity of particle i on iteration k,w is the inertia factor, α is a self-confidence learning constant and β is the swarm influence learning constant, r_1 and r_2 are random numbers between zero and one, PB is the best position ever obtained by particle i, and GB is the best position ever obtained by any member of the population. $x_{i,k}$ is the particle position. The velocity and position are updated using the following to equations:

$$v_{i,k+1} = w * v_{i,k} + \alpha * r_1 * \left(\mathrm{PB} - x_{i,k}\right) + \beta * r_2 * \left(\mathrm{GB} - x_{i,k}\right), \tag{1}$$

$$x_{i,k+1} = x_{i,k} + v_{i,k}. \tag{2}$$

PSO encoding and search for subset selection using machine learning

In this study, the decision variables are indicator variables for each observation in the data set, which indicate whether the observation should be kept in the sample. Based on these decisions, variable candidate solution samples are generated and evaluated using the swarm encoded sample. The PSO algorithm makes the decision of which observations to keep in the sample and decision variables are the inclusion binary variable for each observation. Based on the decisions of the swarm the resulting solution of which observations to keep are then evaluated by fitness function which is the objective function measuring how balanced the samples are for statistical analysis. The fitness functions takes in the input vector of continuous values ranging between 0–1 and these values are converted to integer of 1 if the value is above 0.5. PSO takes the inputs are of length n of the number of observations in the data D and are the continuous decision variables D_n. The PSO uses this vector D_n to encode the solution of which observations to keep in the subset by only keeping observations with decision variable values > 0.50 are kept in the subset S based on $D_n \geq 0.50$ which is a union of the treatment sample S_t. and control sample S_c. The PSO then calculates cost functions on candidate solutions of the particles, encoding the sample to be selected, used to guide search is computed on this subset to evaluate fitness along the five cost functions studied in this paper. The PSO makes the decision on which observations to keep in the sample based on the cost function. This allows combinatorial problem of subset selection to be optimized by swarm algorithms. The observations with a value of 1 are then used to subset the data and fitness is evaluated. Five different fitness cost functions are explored in this study discussed in the experiment setup and are operated on the subset selected by the particle swarms which will be described in the next section. The proposed fitness function based on machine learning ensembles is the maximum cross-validated AUC over M folds $F = \max\left(\mathrm{abs}\left(\mathrm{AUC}_i - 0.5\right)\right) + L$, where is a penalty for violating minimum sample size restrictions. This specification solves a constrained combinatorial optimization using continuous decision variables which are converted to binary values for subsect selection of data for cost calculation.

Machine learning is applied within the balanced subset selection optimization problem in the following way.

1. Swarm algorithm selects disjoint (but not necessarily collectively exhaustive) subsets, S_t, from data D, where the treatment indicator variable $T = 1$, and S_c from the control sample C, where $T = 0$;
2. Determine AUC_i, where $i = 1 \ldots M$ (across all M folds), using selected machine learning algorithms which use T indicator variable as target of classification, considering only the K covariates (treatment and outcome variables excluded);
3. Calculate $F = \max\left(\mathrm{abs}\left(\mathrm{AUC}_i - 0.5\right)\right)$ over $i = 1 \ldots M$; and
4. Using particle swarm optimization (PSO) and modified cuckoo swarm (MCS) methods (separately), repeat (1) thru (3) to minimize $F + L$ (over subsets S_t and S_c), where L is a penalty function used to enforce minimum sample size constraints for S_t and S_c.

The optimization step chooses which observations to keep in the data to minimize the objective function for imbalance. Optimizing AUC and other metrics results in different

samples with different sets of observations retained in the data for inference. Using the penalty of L for sample allows for constrained optimization to enforce statistically sound sample size for a combinatorial problem. The decision variables used for optimization are 0–1 continuous variables for each observation in the data which are then thresholded to be 0 or 1 using 0.5 as a cut off decision boundary to include observations in the sample, which are then evaluated by machine learning ensemble based cost function.

Experimental setup

Both real-world and synthetic data sets are used to evaluate how existing objective functions for optimization perform against the AUC metric. The following approach is used to investigate how well traditional methods perform in terms of the AUC metric. Each objective function is minimized using swarm computations. During optimization of the AUC, due to the computational intensity of the objective function evaluations, threefold cross-validation was used. The computations were performed at George Washington University's high-performance computing cluster, Colonial One. In each objective function evaluation, the machine learning ensemble was built using logistic regression, random forest, stochastic gradient boosting, and support vector machines. All coding was performed in R [46]. The product of the model predictions was computed and used as a scoring function to obtain a super predictor, which was used to compute the AUC metric. Given the variability of cross-validation, a robust objective function was used to minimize the absolute value of the maximum difference from 0.5 for M folds. An AUC metric below 0.5 can be viewed as predictive because the classifier can be used in reverse to yield a predictive performance of 1-AUC; thus, the relative difference from 0.5 is what matters in assessing the discrimination power of a model and test.

For testing the results of the solutions, the cross-validated 10-fold 10-trial AUC metric was computed for each of the 30 trials. Each of the 30 trials consisted of optimizing metrics using PSO for a solution. Thus optimization is performed 30 times resulting in 30 data subsets each of which undergo 10-fold 10 trial cross validation. Repeating tenfold cross validation with 10 trials for each of the 30 optimization results yields 3000 observations. The resulting performance of 3000 AUC observations was compared against alternative methods/objective functions using a t test with Holm's adjustment for multiple comparisons. The 10-fold approach with 10 trials was considered preferable because it has been demonstrated to provide higher power and unbiased estimates with high variance [6, 17]. The cross-validated AUC metric and balance optimization across 5 objective functions were tested on 5 real data sets and 7 synthetic benchmark data sets developed by Setoguchi et al. [57]. The experimental setup was composed of 30 trials of optimizations of the data sets using 5 objective functions. Objective functions 1, 2, and 5 were optimized using PSO and MCS algorithms, while Sekhon's metrics were optimized using a weighted genetic algorithm called GenMatch [56].

Traditional objective functions

Four existing methods and objective functions were evaluated in this study. These methods have been the state-of-the-art methods in the covariate balancing literature [9, 26, 56]. The 4 objective functions included the (1) Chi-square statistics, (2) mean variance differences, (3) KS, and (4) generalized Mahalanobis distance. Objective functions 1 and 2 were studied by Cho et al. [9]. Objective functions 3 and 4 have been developed and optimized using Sekhon's [56] weighted genetic algorithm GenMatch.

Existing objective functions (1–4)

Objective function 1 Chi-square statistics for $B(S_t)$ and $B(S_c)$, where B is a post-processing function to discretize X_i, where $i = 1 \ldots K$, were computed for 10 equal-frequency bins.

Objective function 2 This function is computed as the sum of abs $\sum_{i=1}^{i=K} \left|U(S_{ti}) - U(S_{ci})\right| + \left|\text{var}(S_{ti}) - \text{var}(S_{ci})\right|$, where $U(S_{ti})$ and $U(S_{ci})$ are the means of covariates $i = 1 \ldots k$ in the treatment and control samples, respectively, and $\text{var}(S_{ti})$ and $\text{var}(S_{ci})$ are the variances of the covariates in the treatment and control samples, respectively [9]. This is the benchmark of BOSS methodology as optimizing mean/variance was found to perform best in Cho et al. [9].

Objective function 3 This function is the default GenMatch function of the KS statistics for X_i, where $i = 1 \ldots K$, for S_t and S_c [56].

Objective function 4 This function is Sekhon's [56] generalized Mahalanobis distance of X_i, $i = 1 \ldots K$, for S_t and S_c.

Proposed objective function

To minimize balance using the AUC metric for machine learning, a measure that is robust for sampling error is necessary so that the resulting AUC does not depend on a poor training sample selected by chance. To ensure that the $\max\left|\text{AUC}_{i=1\ldots M} - 0.5\right|$ difference is minimized, a max difference from the absolute value of 0.5 for m-fold cross-validations is proposed. Using a robust optimization metric, given uncertainty in the objective function, minimizes worst case outcomes. This minimization provides the added benefit

of ensuring a sufficient sample size and providing performance that is not simply due to the random selection of a poor training sample for the machine learning algorithm. The proposed minimization of the maximum distance on the cross-validated AUC metric searches for robust solutions by reducing the worst case AUC difference from 0.5.

New objective function

Objective function 5 This function minimizes $\min\left(\max\left|\text{AUC}_{i=1\ldots M}-0.5\right|\right)+L$, where L is a penalty function used to enforce minimum sample size constraints. If the sample size is below the threshold for the minimum sample size, then L is large; otherwise, it is 0. For robustness results on AUC and sample size see Supplementary Appendix A. In cases of large number of features the minimum sample size and regularization can be set experimentally and use of regularization or dimensionality reduction should be considered. For the data sets in hand minimum samples size were not binding but depending on the data set, the researcher should set an appropriate minimum sample size.

The following null research hypotheses will be evaluated:

H1o Optimizing Chi-square distributional difference will result in identical groups which cannot be differentiated by machine learning prediction (AUC = 0.50)

H2o Optimizing covariate mean and variance difference will result in non-identical groups which cannot be differentiated by machine learning prediction (AUC = 0.50)

H3o Optimizing GenMatch KS balance will result in non-identical groups which cannot be differentiated by machine learning prediction (AUC = 0.50)

H4o Optimizing GenMatch Mahalanobis balance will result in non-identical groups which cannot differentiated by machine learning prediction (AUC = 0.50)

H5o Using machine learning for optimization will not lead to an Area under the curve closer to 0.50 than existing methods (Chi-square, mean/variance, KS, Mahalanobis)

Optimization setup

For trials 1 to 30, the objective functions were optimized for each data set D for 2000 iterations with a max function call limit of 3500. The PSO algorithm was derived from the PSO package [4]. The Bendtsen [4] PSO package is an implementation of the standard PSO 2011 algorithm by Maurice Clerc. To perform the MCS optimization, a custom R implementation of the MCS algorithm was developed.

For the AUC metric, in objective function 5, the penalty L was set to obtain the minimal viable sample size based on a literature review. A minimal sample size constraint value of 100 was chosen based on the literature of a minimum sample size needed to validate classification algorithms [2, 58]. In addition, simulations and sensitivity analyses were conducted to assess and confirm the power of the machine learning-based AUC metric using the synthetic scenarios of Thulin [65]. A penalty was used when the sample size was less than 100 or if either the treatment or control group size was less than 50. The value of the penalty L was set to 999, although a value of at least 1 was sufficient to ensure that solutions that failed the constraint were eliminated based on a sensitivity analysis. To ensure that the AUC metric for the classifier was not due to imbalance or insufficient data, an additional constraint on the treatment of each class and control of a minimum of 50 observations were applied to ensure that both the treatment and control samples were sufficient. During cross-validation, a stratified sample for each training and test fold ensured that both the treatment and control samples had sufficient observations in each fold. Because the cross-validation metric was used, the solutions were found to be robust in terms of overfitting and generated more samples that were robust to imbalance. During the analysis, sensitivity analysis was performed by removing the sample size penalty; the results were similar to the optimization with penalty. Thus, given the variability of the data sets, it is reasonable to keep the enforced minimal sample size constraint to obtain valid optimization and statistical results.

Wright and Ziegler's [72] ranger package in R was used for the random forest algorithm. David Myer's e1071 R package was used for the SVM component implementation with a default radial basis kernel setting [40]. Culp, Johnson, and Michailidis' ada package was used for the stochastic boosting component. For all algorithms, the default settings were used in R, (500 trees for random forest; for SVM with radial basis functions, gamma of $1/K$ and cost of 1; for stochastic boosting, the iterations were 50 with a learning rate $nu = 1$). Genetic and generalized Mahalanobis matching followed Sekhon's GenMatch and Ho, Imai, King & Stuart's MatchIt packages in R. The ROCR package was used for the ROC calculations [61].

Description of the real-world data sets

The real data sets included the widely used the Lalonde data set, daughters' data, and right heart catheterization data [32, 43, 69]. The LaLonde [32] data set has been extensively studied for matching and isolates the causal effect of the treatment of workforce training. The right heart catheterization (rhc) data have been studied for matching using assignment methods and is related to the effect of receiving a Swan–Ganz catheter treatment on patient survival time [13, 25, 43]. The third real-world data set, the daughters' data set, is related to the impact of having daughters as a treatment on the outcome of legislators' voting behavior toward women's

issues [71]. The Lalonde data set included 10 covariates and 445 observations; the rhc data set contained 40 covariates (based on converting factors into categories). The rhc data were down-sampled to 1124 observations from 5728 sample for performance reasons. The daughters' data set had 33 covariates and 1735 observations. The number of variables only included the K covariates; the treatment variable and outcome variable were excluded. The fourth real-world data set is voter data from the Gerber Green Imai get-out-the-vote data set, contained in the Matching package in R with 10,089 observations and 8 features in which the treatment was phone call urging response in local voting election [56]. The fifth real data set is a 1000 observation sample from the recent Propublic's recidivism data on likelihood to commit crime with 7 features where the treatment was set to be either being female or a minority [44, 45].

Description of synthetic data sets

The synthetic data sets were obtained from the Setoguchi setup for simulated scenarios A, B, C, D, E, F, and G with 1000 observations and $K = 10$ covariates, where the treatment assignment was set as a function of various scenarios relating to non-linearity and non-additivity. The variables $W_{i=1\ldots10}$ are normal random variables with 0 mean and unit variance [34, 57]. Setoguchi's E model treatment is a function of three two-way interaction terms and one quadratic term. Scenario F models 10 two-way interaction terms, and scenario G contains 10 two-way interaction terms and three quadratic terms. These scenarios highlight cases where matching on distributions and single moments would miss important differences in the samples that machine learning can detect.

The correlation structure between the 10 covariates $W_{i=1\ldots10}$ is that W_9 and W_4 have a 0.9 correlation coefficient, W_3 and W_8 have a 0.2 correlation, W_1 and W_5 have a 0.2 correlation, and W_2 and W_6 have a 0.9 correlation [57].

Scenario A (a model with additivity and linearity)
Scenario B (a model with mild non-linearity)
Scenario C (a model with moderate non-linearity)
Scenario D (a model with mild non-additivity)
Scenario E (a model with mild non-additivity and non-linearity)
Scenario F (a model with moderate non-additivity)
Scenario G (a model with moderate non-additivity and non-linearity) [57].

Table 1 shows the pre-optimization cross-validated AUC metric for the data sets using $M = 10$ folds with 10 trials. The target variable was set as the indicator if the observation was from the treatment sample, where $T = 1$, or control sample, where $T = 0$.

Table 1 Pre-optimization cross-validated AUC metric for 10 folds with 10 trials based on machine learning for classifying treatment from control

Data type	Data set	Mean AUC	St. dev. AUC
Real data	Lalonde	0.57	0.08
Real data	rhc	0.76	0.04
Real data	Daughters	0.84	0.05
Real data	Voter	0.8	0.03
Real data	Propub	0.64	0.03
Simulated	Data Setoguchi A	0.78	0.03
Simulated	Data Setoguchi B	0.75	0.03
Simulated	Data Setoguchi C	0.77	0.05
Simulated	Data Setoguchi D	0.78	0.05
Simulated	Data Setoguchi E	0.78	0.04
Simulated	Data Setoguchi F	0.77	0.04
Simulated	Data Setoguchi G	0.79	0.06

Results and discussion

Comparing direct optimization of the AUC metric vs. alternate objective functions

Because PSO performed as well as or better than the MCS algorithm, PSO was used to compare the performance of direct optimization of Chi-square, mean and variance, and AUC metric objectives. These results were also compared against the existing genetic matching and generalized Mahalanobis matching methods. AUC metric optimization using PSO outperformed the other methods and objective functions for all data sets and was closely followed by the mean and variance optimization method. Figure 4 summarizes the AUC metric results for all optimized objective functions using PSO for all data sets. Figure 4 shows that AUC metric optimization was closest to the 0.5 AUC metric target across all data sets. Data on individual classifier's performance information for each data set are in Supplementary Appendix B.

Comparing optimization results across methods using other empirical difference metrics

Table 2 shows that the optimization of the AUC metric for the real-world data set using PSO resulted in a performance near the 0.5 benchmark; it outperformed the other methods in 2 out of 5 real-world data sets. In Table 2, the optimized solutions are shown in terms of mean/variance differences and Chi-square statistics. The AUC-optimized solution was only slightly worse in terms of Chi-square metrics. For the Lalonde data set, the AUC-optimized solution had a 23% higher Chi-square statistic than the Chi-square-optimized solution, a 10% lower Chi-square solution for rhc, and a

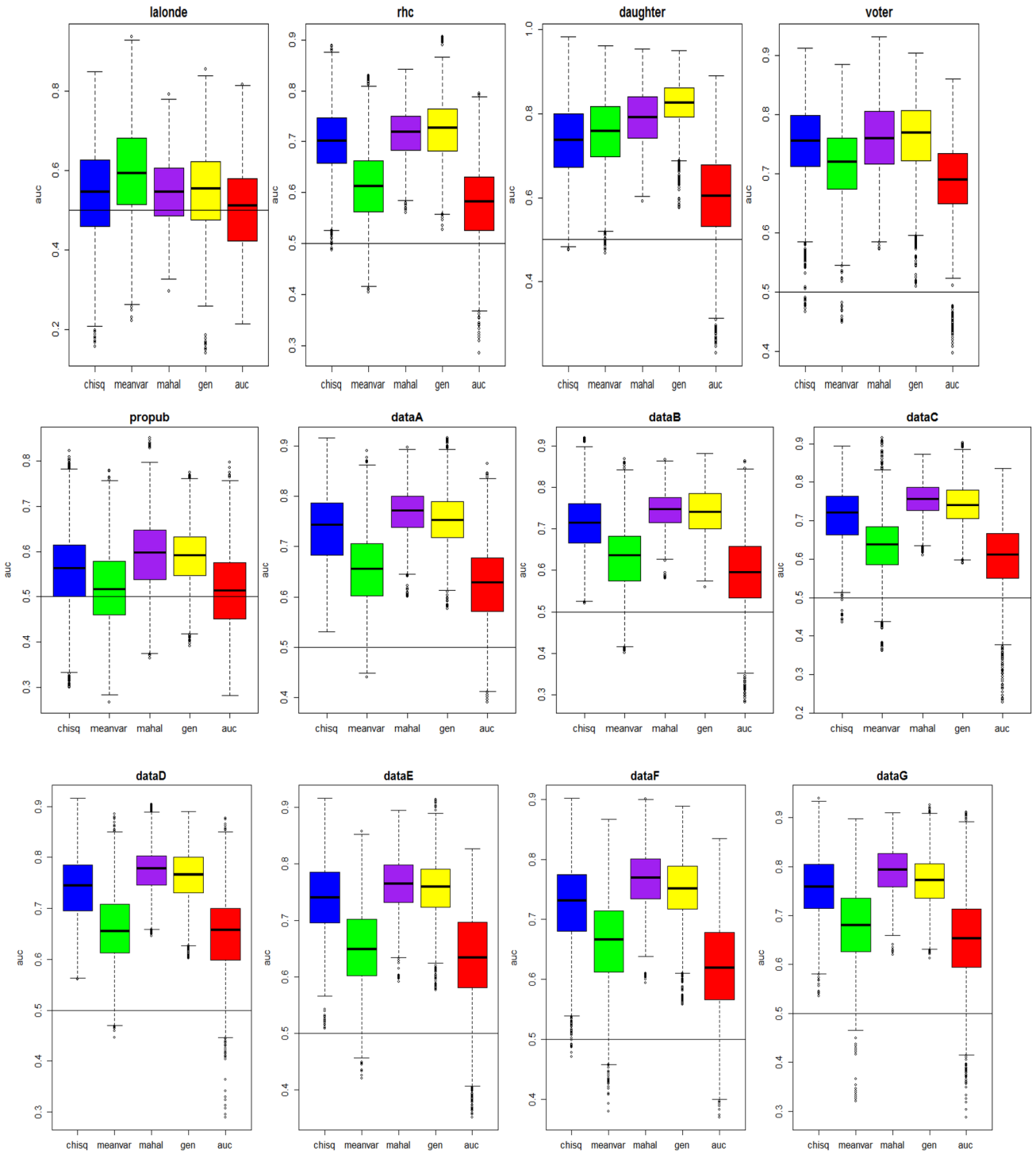

Fig. 4 Box plots of the cross-validated AUC metric for different objective functions

5% higher Chi-square statistic for the daughters' data. In terms of the sum of the absolute values of the standardized means and variances, the AUC-optimized solution was higher, although the differences across the empirical distributions were small because the overall AUC metric was near 0.5. When averaged over the number of covariates, the differences were small. For example, $K = 10$ for the Lalonde data, $K = 40$ for the rhc data, $K = 10$ for the voter data and $K = 33$ for the daughters' data.

Tables 3 and 4 show that the PSO AUC metric outperforms or performs as well as the mean and variance for the synthetic data sets. For the simulated data sets, scenarios

Table 2 Cross-validated AUC metric for 30 trials of 10 folds with 10 trials for real data

PSO Objective functions	Objective function 1 Chi-square (PSO)	Objective function 2 MeanVar (PSO)	Objective function 3 Mahal (GenMatch)	Objective function 4 KS (GenMatch)	Objective function 5 AUC (PSO)
Data: Lalonde					
Mean AUC	0.54	0.6	0.55	0.55	0.51
St. dev. AUC	0.12	0.12	0.09	0.1	0.12
MeanVar difference	5.2	1.39	3.43	3.79	4.55
Chi-square statistic	1257.18	1440.04	2348.66	1932.85	1451.49
Number of observations	201	224	370	314	228
Number of treated observations	84	93	185	185	95
Data: rhc					
Mean AUC	0.7	0.61	0.72	0.72	0.58
St. dev. AUC	0.06	0.08	0.05	0.06	0.08
MeanVar difference	14.56	6.78	14.66	15.56	11.61
Chi-square statistic	308.39	384.6	623.85	558.1	385.95
Number of observations	587	577	906	767	514
Number of treated observations	225	216	453	453	190
Data: Daughters					
Mean AUC	0.73	0.76	0.79	0.82	0.6
St. dev. AUC	0.09	0.09	0.07	0.06	0.11
MeanVar difference	6.78	2.44	6.77	6.3	6.19
Chi-square statistic	199.91	217.62	276	332.28	190.21
Number of observations	409	409	450	814	361
Number of treated observations	299	294	225	603	270
Data: Voter					
Mean AUC	0.75	0.72	0.76	0.76	0.69
St. dev. AUC	0.07	0.06	0.06	0.06	0.07
MeanVar difference	3.7	2.11	3.87	3.89	3.11
Chi-square statistic	1567.68	1737.85	1827.68	1816	1671.43
Number of observations	5393	5280	494	491	4963
Number of treated observations	108	111	247	247	102
Data: Propublica					
Mean AUC	0.56	0.52	0.60	0.59	0.51
St. dev. AUC	0.09	0.09	0.08	0.07	0.09
MeanVar difference	4.67	1.27	4.53	3.78	4.43
Chi-square statistic	41.88	59.89	72.54	70.64	57.32
Number of observations	506	500	512	985	502
Number of treated observations	376	368	256	744	375

p values for all comparisons were derived using the Holm correction for multiple comparisons < 0.01

C and G, which have moderate to high non-linearity, the AUC metric optimization using PSO resulted in a value of approximately 0.5. For the real-world data sets, i.e., the Lalonde and daughters' data, the method resulted in an AUC metric of nearly 0.5 as well. For Setoguchi scenarios A–G, optimizing the AUC metric with PSO provided the best performance. However, the resulting AUC metric was not 0.5, where AUC = 0.5 indicates that the ensemble of algorithms had no predictive power in separating the two sets (S_t and S_c) using the K covariates X_i, where $i = 1 \ldots K$. For Setoguchi scenarios A–G, the AUC metrics ranged from 0.59 to 0.65. Figure 4 shows that the performance of the PSO-based AUC metric optimization outperformed the other methods and was no worse than the mean and variance optimization method. The results for the various objective functions resulted in samples with a similar size of approximately half the original data sets, which was well above the minimum 100 observations needed to conduct the analysis.

In Tables 3 and 4, the optimized solutions are shown in terms of mean/variance differences and Chi-square statistics,

Table 3 The cross-validated AUC metric for simulated Setoguchi scenarios A–C ($N = 3000$; 30 trials of 10 folds with 10 trials)

PSO	Objective function 1	Objective function 2	Objective function 3	Objective function 4	Objective function 5
Data: Setoguchi A: Additivity and linearity (main effects only)					
Mean AUC	0.74	0.65	0.77	0.75	0.62
St. dev. AUC	0.07	0.08	0.04	0.05	0.08
Data: Setoguchi B: mild non-linearity (one quadratic term)					
Mean AUC	0.71	0.63	0.74	0.74	0.59
St. dev. AUC	0.07	0.08	0.05	0.06	0.09
Data: Setoguchi C: moderate non-linearity (three quadratic terms)					
Mean AUC	0.71	0.64	0.75	0.74	0.61
St. dev. AUC	0.07	0.08	0.05	0.05	0.09

p values for all comparisons were derived using the Holm correction for multiple comparisons < 0.01

Table 4 The cross-validated results for the AUC metric for simulated Setoguchi scenarios D–G ($N = 3000$; 30 trials of 10 folds with 10 trials)

PSO	Objective function 1 Chi-square (PSO)	Objective function 2 MeanVar (PSO)	Objective function 3 Mahal (GenMatch)	Objective function 4 KS (GenMatch)	Objective function 5 AUC (PSO)
Data: Setoguchi D: mild non-additivity (three two-way interaction terms)					
Mean AUC	0.74	0.66	0.78	0.77	0.65
St. dev. AUC	0.06	0.07	0.04	0.05	0.08
Data: Setoguchi E: moderate non-additivity (ten two-way interaction terms)					
Mean AUC	0.74	0.65	0.76	0.76	0.63
St. dev. AUC	0.07	0.07	0.05	0.05	0.09
Data: Setoguchi F: moderate non-additivity (ten two-way interaction terms)					
Mean AUC	0.73	0.66	0.77	0.75	0.62
St. dev. AUC	0.07	0.07	0.05	0.06	0.08
Data: Setoguchi G: moderate non-additivity and non-linearity (ten two-way interaction terms and three quadratic terms)					
Mean AUC	0.76	0.68	0.79	0.77	0.65
St. dev. AUC	0.06	0.08	0.05	0.05	0.09

p values for all comparisons were derived using the Holm correction for multiple comparisons < 0.01

and the AUC-optimized solution was only slightly worse in terms of these metrics. In terms of the Chi-square statistic, the AUC-optimized results were within 10–15% of the Chi-square value of direct optimization of the Chi-square statistic. In terms of the sum of the absolute values of the standardized means and variances, the AUC-optimized solution was higher, although differences across the empirical distributions were small when averaged over the number of covariates, i.e., $K = 10$.

Analysis of bias in AUC-optimized samples

Once balanced samples are constructed, they can be analyzed using mean differences or double robust methods, which simply include the treatment variable and all standard covariates K in the regression model to estimate the causal effect. Looking at the AUC metric, only the Lalonde data set was sufficiently balanced to appear generated by random chance. When the samples are not balanced, i.e., an AUC metric not equal to 0.50, double robust methods should be used for analyzing the resulting data. In addition, even when the AUC metric is near 0.50, the mean/variance can be higher between samples, and the use of double robust methods can effectively handle the remaining imbalance. The results show that although an AUC of 0.50 guarantees statistical balance, small empirical differences can persist and may affect accuracy. Tables 5 and 6 show that for the Lalonde data set, the empirical differences do not affect bias because estimating the causal effect using only the treatment variable or all covariates yields the same effect. However, for the rhc, voter, daughters' and Setoguchi scenario A-G data sets, double robust methods were needed with AUC metric optimization to achieve unbiased results. The results of the PSO-based AUC-optimized samples were analyzed to isolate the treatment effect. For the 30 samples resulting from the optimization trials, as shown in Table 5, the mean treatment effect and p values matched the

Table 5 Treatment effects for real data sets using PSO AUC metric optimization (double robust regression controlling for all covariates; 30 trials)

Result type	Data	Mean
Beta	Lalonde	***1782***
Beta	rhc	***− 0.02***
Beta	Daughters	3.6
Beta	Voter	*0.09*
Beta	Propub	*0.00*

Cells in italicized indicate estimates for the benchmark matched in the literature [13, 62, 69]. Bold cells indicate results with p value ≤ 0.05 from regression

Table 6 Treatment effects for real data sets using regressions with only a treatment variable (30 trials)

Result type	Data	Mean
Beta	Lalonde	*1866*
Beta	rhc	−0.036
Beta	Daughters	6.3
Beta	Voter	**0.16**
Beta	Propub	0.05

Cells in italics indicate estimates for the benchmark matched in the literature [13, 62, 69]. Bold cells indicate results with p value ≤ 0.05 from regression

benchmarks for the data sets using a regression control for all covariates. This result highlights the importance of using double robust inference procedures after the optimization step to obtain accurate treatment effects. Table 6 shows that only regressing the treatment variable results in estimates close to published benchmarks for Lalonde and the estimates are not as precise as the double robust effects listed in Table 5 [13, 69]. The results from the double robust method are the estimated treatment effect using linear regression and controlling for all covariates as opposed to the unconditional mean difference between treatment and control observations presented below. Of the data sets only Lalonde, Voter and Setoguchi data have actual experiment to compare to and for these data sets the double robust analysis shows benchmark estimates are achieved of $1782 for Lalonde as the effect of work training on income, for Voter no significant impact of calling on voter turnout and − 0.4 effect for the synthetic data.

For the synthetic data sets, only Setoguchi scenario G resulted in the same estimate for the treatment using only the treatment and using treatment plus standard covariates, as shown in Tables 7 and 8. For scenarios A–F, double robust methods are needed given that empirical differences in the data exist; the AUC metric was not 0.50.

Table 7 Treatment effects for Setoguchi et al. [57] data using PSO AUC metric optimization (double robust regression controlling for all covariates and treatment variable; 30 trials)

Result type	Data	Mean
Beta	Setoguchi A-G	***− 0.40***

Cells in italics indicate estimates for the matched benchmark. Bold cells indicate p value <=0.05

Table 8 Treatment effect for Setoguchi et al. [57] data, regression with treatment variable (30 trials)

Result type	Data	Mean
Beta	Setoguchi A	**− 0.28**
Beta	Setoguchi B	**− 0.33**
Beta	Setoguchi C	**− 0.26**
Beta	Setoguchi D	**− 0.29**
Beta	Setoguchi E	**− 0.27**
Beta	Setoguchi F	**− 0.30**
Beta	Setoguchi G	**− 0.28**

Cells in yellow indicate estimates for the matched benchmark. Bold cells indicate p value ≤ 0.05

Conclusion

Combining operations research approaches with machine learning is a fruitful area of study that can provide insights for both fields and help improve solution optimality. Using the proposed cross-validated AUC metric for balance, we show that existing matching optimizing approaches result in matching that is suboptimal. The machine learning ensembles were able to detect differences in the optimized samples in 12 data sets comprising of 5 real-world data sets and 7 synthetic data sets. After identifying this gap, we showed a significant improvement in balance using direct optimization of the AUC metric via nature inspired stochastic optimization algorithms. Nature inspired stochastic search methods, such as particle swarm and modified cuckoo algorithms, appear to be effective in optimizing the cross-validated AUC metric. PSO significantly outperformed MCS optimization in the context of the cross-validated AUC metric. The PSO-based AUC optimization approach achieved statistically significant reduction in imbalance, in terms of distance from AUC 0.5, relative to other methods in all 12 data sets.

Direct optimization of the AUC metric can improve results for many data sets. Given the NP-hard nature of the search space, even this approach can still result in samples that are not identical. For example, except for the Lalonde data, no balancing approach was able to achieve a perfect balance of AUC = 0.5. Using double robust inference using

the AUC optimized sample yielded unbiased estimates in 11 of the 12 data sets. In practice, sufficient balance may or may not be achieved but having the diagnostic information regarding how much balance in AUC can be achieved is valuable to researchers and analysts to know the limitations of the data and bias potentially present in the data. Where a sufficient match balance was not achieved, other methods may be needed to improve performance. One such approach is that of double robust methods, which is composed of regression analysis control for K covariates. The method has been proposed as a way to control the remaining imbalance after matching [12, 62]. Using double robust methods with optimized samples is crucial for achieving unbiased estimates based on our results. PSO-based AUC optimization can yield better samples than past metrics and serves as a useful diagnostic for balance. The null hypotheses 1-5 were rejected in favor of the following alternative hypotheses that machine learning could detect differences in samples optimized using existing methods (Chi-square, mean/variance differences, Mahalanobis distance and KS) and optimizing AUC lead to solutions with less imbalance as the direct optimization sample had an AUC closer to 0.5 than existing methods (Chi-square, mean/variance, Mahalanobis and KS).

This research makes the following contributions to the literature:

- Novel application of machine learning used to detect balance to improve causal inference
  - Optimized data selection to improve balance and resulting inference
- Improved state of practice of causal analysis of data
  - Showed optimizing existing balance metrics/objective functions can still result in imbalance detectable by machine learning
- Proposed metric provides an important diagnostic to know if imbalance remains
- Showed particle swarm optimization is effective in optimizing machine learning based AUC metric.

Scaling-up approach and future work

Scaling the algorithms up to a parallelized implementation is an interesting topic for future research. High-performance computing was primarily used to speed up the repeated trials. Most optimization methods ran within 6–8 h. Measuring the cross-validated AUC metric using machine learning is efficient because highly optimized algorithms are available. The rise of inexpensive computing power makes the computationally intensive methods discussed here practical for researchers.

To scale the approach to 'big data' would simply involve breaking larger data into randomly sampled chunks to be processed in parallel and then combining via parallel approaches, such as map-reduce. Given the proposed approach, it seems reasonable to parallelize different random samples and combine using test measures of similarity to scale the approach to larger data sets. However, we leave this approach to future study. Another way to scale the approach would be to perform mean and variance optimization on samples, which runs quickly, and then run the AUC-based optimization as a second stage on the smaller data set from the first stage to combine the benefits of mean and variance distance metrics and the AUC metric. Another extension of the optimal samples created using the AUC metric is to analyze the pattern of imbalance by studying how different the optimized sample is from the original data set. This analysis may provide insight into the biasing process. The improvement in the AUC metric via machine learning ensemble methods suggests that there is value in the use of interaction effects of the treatment variable with the other covariates in the regression step. This work suggests further avenues of research for both operations research and machine learning. Researching and developing improved swarms and studying whether PSO has a general advantage for noisy objective functions and for machine learning is another area for future research.

Improving classification components of approach

In most cases the super scorecard ensemble recommended by Hand and Kelly [24] performed as well as the best individual classifier but not in all cases (examples Supplementary Appendix B Setoguchi A, D, E, F). In all cases it had better AUC or equivalent AUC as averaging the individual classifier AUCs. As such one might achieve better results by minimizing the maximum AUC difference from 0.5 of all individual classifiers and ensemble. In addition, individual classifiers can be improved further via tuning and regularization. Other ways to enhance the approach are setting sample size constraints based on experimentation for data sets with higher number of features. Reducing dimensionality by projecting the data into a lower dimensional space might also be another way to balance competing sample size and more efficiency in the machine learning classification iterations.

Improving optimization future work

Any step to reduce the number of observations can make the optimization more efficient computationally. Given the difficult search space of 2^N, where N is the number of observations, approaches to reduce the search space via preprocessing the data could be fruitful. For example by dropping points using a first pass based on classifier predictions like dropping observations in the highest or lowest predicted probability prediction deciles might be effective in reducing

the search space. For the PSO optimization step currently the approach does not make use of machine learning classifier predictions to guide steps in the search space directly. Using the results of machine learning predictions by learning from PSO results iteratively to suggest potential points to drop would be interesting lines of further inquiry.

The large decision space also warrants future study and application of recent swarm optimizers for large-scale optimization. Large search spaces can have poor convergence and many local minimums, recent large-scale optimization methods try to overcome these issues by either using more diverse solutions developed in parallel to overcome local minima or divide and conquer to reduce dimensionality of problem space. Large scale multi-objective competitive swarm optimizer also has promise with a more efficient search algorithm and can be paired with decision variable analysis or grouping to reduce search space as well [66, 76]. The current problem can naturally lend itself to divide and conquer into sub-samples which could be optimized in parallel thus benefitting from recent large-scale optimization swarms [70, 78]. Given the computational expense of the problem can also benefit from application of recent distributed algorithms that offer the benefit of more efficient parallel search and more efficient particle updates [77].

Acknowledgements Thanks to Jason Sauppe for generous input and feedback especially on BOSS problem and relationship of balance and bias. Also thanks to my brother Paras Sharma and Stephanie Jones for various rounds of feedback and review to improve the writing and readability.

References

1. Ab Wahab MN, Nefti-Meziani S, Atyabi A (2015) A comprehensive review of swarm optimization algorithms. PloS One 10(5):e0122827
2. Beleites C, Neugebauer U, Bocklitz T, Krafft C, Popp J (2013) Sample size planning for classification models. Anal Chim Acta 760:25–33
3. Beling P, Covaliu Z, Oliver RM (2005) Optimal scoring cutoff policies and efficient frontiers. J Oper Res Soc 56:1016–1029
4. Bendtsen C (2012) PSO: particle swarm optimization. DIALOG. https://cran.r-project.org/web/packages/pso/index.html. Accessed 1 Aug 2016
5. Bonyadi MR, Michalewicz Z (2017) Particle swarm optimization for single objective continuous space problems: a review
6. Bouckaert RR (2003) Choosing between two learning algorithms based on calibrated tests. In: Fawcett T, Mishra N (eds) Proceedings of 20th international conference on machine learning. AAAI Press, Washington, DC, pp 51–58
7. Cheng S, Lu H, Lei X, Shi Y (2018) A quarter century of particle swarm optimization. Complex Intell Syst 1–13
8. Cho WKT, Liu YY (2016) A parallel evolutionary algorithm for subset selection in causal inference models. In: Proceedings of the XSEDE16 conference on diversity, big data, and science at scale. ACM, Miami, pp 1–8
9. Cho WKT, Sauppe JJ, Nikolaev AG, Jacobson SH, Sewell EC (2013) An optimization approach for making causal inferences. Stat Neerl 67:211–226
10. Clemencon S, Depecker M, Vayatis N (2009) AUC optimization and the two-sample problem. Adv Neural Inf Process Syst 22:360–368
11. Cochran WG, Moses LE, Mosteller F (1983) Planning and analysis of observational studies. Wiley, New York
12. Colson KE, Rudolph KE, Zimmerman SC, Goin DE, Stuart EA, Laan MVD, Ahern J (2016) Optimizing matching and analysis combinations for estimating causal effects. Sci Rep 6:23222
13. Connors AF Jr, Speroff T, Dawson NV et al (1996) The effectiveness of right heart catheterization in the initial care of critically ill patients. SUPPORT investigators. JAMA 276:889–897
14. Cortes C, Mohri M (2003) AUC optimization vs. error rate minimization. In: Proceedings of the 16th international conference on neural information processing systems. MIT Press, Canada, pp 313–320
15. Cristianini N, Shawe-Taylor J (1999) An introduction to support vector machines. Cambridge University Press, Cambridge
16. Culp M, Johnson K, Michailidis G (2006) ada: an R package for stochastic boosting. J Stat Softw 17:1–27
17. Dietterich TG (1998) Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput 10:1895–1923
18. Dietterich TG (2000) Ensemble methods in machine learning. In: Kittler J, Roli F (eds) Multiple classifier systems. First international workshop, MCS 2000, Cagliari, Italy, vol 1857 of Lecture Notes in Computer Science. Springer, Berlin, pp 1–15
19. Duda RO, Hart PE (2000) Pattern classification and scene analysis. Wiley, New York
20. Fernandez-Viagas V, Ruiz R, Framinan JM (2017) A new vision of approximate methods for the permutation flowshop to minimise makespan: state-of-the-art and computational evaluation. Eur J Oper Res 257:707–721
21. Finlay S (2011) Multiple classifier architectures and their application to credit risk assessment. Eur J Oper Res 210:368–378
22. Franklin JM, Rassen JA, Ackermann D, Bartels DB, Schneeweiss S (2013) Metrics for covariate balance in cohort studies of causal effects. Stat Med 33:1685–1699
23. Gayler R (1995) Is the wholesale modeling of interactions worthwhile? In: Proceedings of the credit scoring and credit control conference. University of Edinburgh Management School, Edinburgh
24. Hand D, Kelly MG (2002) Superscorecards. IMA J Manag Math 13:273–281
25. Harrell F (2002) Right heart catheterization data set. Available via DIALOG. http://biostat.mc.vanderbilt.edu/wiki/pub/Main/Datasets/rhc.html. Accessed 1 Aug 2016

26. Ho DE, Imai K, King G, Stuart EA (2011) MatchIt: nonparametric preprocessing for parametric causal inference. J Stat Softw 42:1–28
27. Jaques N, Nutini J (2016) A comparison of random forests and dropout nets for sign language recognition with the Kinect. Available via DIALOG. http://www.cs.ubc.ca/~jaquesn/MachineLearningProject.pdf. Accessed 1 Aug 2016
28. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. IEEE, Piscataway, NJ, pp 1942–1948
29. Kennedy J, Eberhart RC, Shi Y (2001) Swarm intelligence. Morgan Kaufmann Publishers, San Francisco
30. King G, Nielson R (2016) Why propensity scores should not be used for matching. Available via DIALOG. http://gking.harvard.edu/files/gking/files/psnot.pdf. Accessed 1 Aug 2016
31. Krauss C, Do XA, Huck N (2017) Deep neural networks, gradient-boosted trees, random forests: statistical arbitrage on the S&P 500. Eur J Oper Res 259:689–702
32. LaLonde RJ (1986) Evaluating the econometric evaluations of training programs with experimental data. Am Econ Rev 76:604–620
33. Laskari EC, Parsopoulos KE, Vrahatis MN (2002) Particle swarm optimization for integer programming. In: Proceedings of the IEEE congress on evolutionary computation. IEEE, Honolulu, pp 1582–1587
34. Lee BK, Lessler J, Stuart EA (2010) Improving propensity score weighting using machine learning. Stat Med 29:337–346
35. Lessmann S, Baesens B, Seow H-V, Thomas LC (2015) Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. Eur J Oper Res 247:124–136
36. Lopes ME, Jacob L, Wainwright MJ (2011) A more powerful two-sample test in high dimensions using random projection. In: Proceedings of the 24th international conference on neural information processing systems. Curran Associates Inc., Granada, pp 1206–1214
37. López FGA, Torres MGA, Batista BM, Pérez JAM, Moreno-Vega JM (2006) Solving feature subset selection problem by a parallel scatter search. Eur J Oper Res 169:477–489
38. Marqués AI, García V, Sánchez JS (2013) A literature review on the application of evolutionary computing to credit scoring. J Oper Res Soc 64:1384–1399
39. Matsuoka Y (2016) Forefront of the two sample problem: from classical to state of the art methods. Available via DIALOG. http://yuchimatsuoka.github.io/seminar/201612.pdf. Accessed 1 Sep 2017
40. Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F, Chang C, Lin C (2017) Package 'e1071'. Available via DIALOG. https://cran.r-project.org/web/packages/e1071/index.html. Accessed 1 Aug 2016
41. Nikolaev AG, Jacobson SH, Cho WKT, Sauppe JJ, Sewell EC (2013) Balance optimization subset selection (BOSS): an alternative approach for causal inference with observational data. Oper Res 61:398–412
42. O'Neil C (2017) Weapons of math destruction: how big data increases inequality and threatens democracy. Broadway Books, New York
43. Pimentel SD (2016) Large, sparse optimal matching with R package rcbalance. Obs Stud 2:4–23
44. ProPublica (2016) Machine bias. Available via DIALOG. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. Accessed 1 Feb 2017
45. ProPublica (2017) COMPASS analysis and data. Available via DIALOG. https://github.com/propublica/compas-analysis
46. Development Core Team R (2006) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna
47. Ratkovic M (2014) Balancing within the margin: causal effect estimation with support vector machines. Princeton University, Princeton **(Unpublished Manuscript)**
48. Reddi SJ, Poczos B, Smola AJ (2015) Doubly robust covariate shift correction. In: Proceedings of the twenty-ninth AAAI conference on artificial intelligence. AAAI Press, Austin, pp 2949–2955
49. Rosenbaum PR (2002) Observational studies. Springer, New York
50. Rosenbaum PR (2005) An exact distribution-free test comparing two multivariate distributions based on adjacency. J R Stat Soc Ser B Stat Methodol 67:515–530
51. Rosenbaum PR, Rubin DB (1985) Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. Am Stat 39:33–38
52. Rubin DB (2005) Causal inference using potential outcomes. J Am Stat Assoc 100:322–331
53. Sauppe JJ (2015) Balance optimization subset selection: a framework for causal inference with observational data. Ph.D. Thesis. University of Illinois at Urbana-Champaign, Urbana, IL
54. Sauppe JJ, Jacobson SH (2017) The role of covariate balance in observational studies. NRL 64:323–344
55. Sauppe JJ, Jacobson SH, Sewell EC (2014) Complexity and approximation results for the balance optimization subset selection model for causal inference in observational studies. INFORMS J Comput 26:547–566
56. Sekhon JS (2011) Multivariate and propensity score matching software with automated balance optimization: the matching package for R. J Stat Softw 42:7
57. Setoguchi S, Schneeweiss S, Brookhart MA, Glynn RJ, Cook EF (2008) Evaluating uses of data mining techniques in propensity score estimation: a simulation study. Pharmacoepidemiol Drug Saf 17:546–555
58. Shao L, Fan X, Cheng N, Wu L, Cheng Y (2013) Determination of minimum training sample size for microarray-based cancer outcome prediction–an empirical assessment. PLoS One 8:e68579
59. Sharma D (2012) Improving the art, craft and science of economic credit risk scorecards using random forests: why credit scorers and economists should use random forests. Acad Bank Stud J 11:93–116
60. Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. J Stat Plan Inference 90:227–244
61. Sing T, Sander O, Beerenwinkel N, Lengauer T (2005) ROCR: visualizing classifier performance in R. Bioinformatics 21:3940–3941
62. Stuart EA (2010) Matching methods for causal inference: a review and a look forward. Stat Sci 25:1–21
63. Tasgetiren MF, Liang Y-C, Sevkli M, Gencyilmaz G (2007) A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. Eur J Oper Res 177:1930–1947
64. Thomas LC (2009) Consumer credit models: pricing, profit and portfolios. OUP Oxford, New York
65. Thulin M (2014) A high-dimensional two-sample test for the mean using random subspaces. Comput Stat Data Anal 74:26–38
66. Tian Y, Zheng X, Zhang X, Jin Y (2019) Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. IEEE Trans Cybern
67. Unler A, Murat A (2010) A discrete particle swarm optimization method for feature selection in binary classification problems. Eur J Oper Res 206:528–539
68. Verikas A, Gelzinis A, Bacauskiene M (2011) Mining data with random forests: a survey and results of new tests. Pattern Recognit 44:330–349

69. Walton S, Hassan O, Morgan K, Brown MR (2011) Modified cuckoo search: a new gradient free optimisation algorithm. Chaos Solitons Fractals 44:710–718
70. Wang X, Wang GG, Song B, Wang P, Wang Y (2019) A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. IEEE Trans Evol Comput 23:815–827
71. Washington EL (2008) Female socialization: how daughters affect their legislator fathers' voting on women's issues. Am Econ Rev 98:311–332
72. Wright MN, Ziegler A (2017) ranger: a fast implementation of random forests for high dimensional data in C++ and R. J Stat Softw 77:1–17
73. Wu X, Kumar V, Quinlan JR et al (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14:1–37
74. Yang XS, Deb S (2010) Engineering optimisation by cuckoo search. Int J Math Model Numer Optim 1:330–343
75. Yang XS, Deb S (2013) Cuckoo search: recent advances and applications. Neural Comput Appl 24:169–174
76. Yang Q, Chen WN, Da Deng J, Li Y, Gu T, Zhang J (2017) A level-based learning swarm optimizer for large-scale optimization. IEEE Trans Evol Comput 22:578–594
77. Yang Q, Chen WN, Gu T, Zhang H, Yuan H, Kwong S, Zhang, J (2019) A distributed swarm optimizer with adaptive communication for large-scale optimization. IEEE Trans Cybern
78. Yang P, Tang K, Yao X (2019) A parallel divide-and-conquer-based evolutionary algorithm for large-scal optimization. IEEE Access 7:163105–163118
79. Zhang Z (2007) Use of area under the curve (AUC) from propensity model to estimate accuracy of the estimated effect of exposure. Master's Thesis. University of Pittsburgh, Pittsburgh
80. Zubizarreta JR (2012) Using mixed integer programming for matching in an observational study of kidney failure after surgery. J Am Stat Assoc 107:1360–1371