How effective are minimax methods in mitigating sample selection bias?

Zeeshan Khan

Supervisor(s): Joana Gonçalves, Yasin Tepeli

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Zeeshan Khan
Final project course: CSE3000 Research Project
Thesis committee: Joana Gonçalves, Yasin Tepeli, Julian Urbano Merino

Abstract

Sample selection bias is a well-known problem in machine learning, where the source and target data distributions differ, leading to biased predictions and difficulties in generalization. This bias presents significant challenges for modern machine learning algorithms. To tackle this problem, researchers have focused on developing domain adaptation techniques that aim to create robust methods against sample selection bias. One approach is the use of minimax estimation techniques, which belong to the category of inference-based techniques. Despite the extensive research in developing these domain adaptation methods, there remains a critical need to evaluate their performance. This thesis explores the performance differences of various minimax estimation techniques in the presence of sample selection bias, providing insights into their effectiveness in mitigating the challenges posed by biased data. By understanding and evaluating the performance of these techniques, this research contributes to the advancement of domain adaptation methods and their application in real-world machine learning scenarios.

1 Introduction

The field of machine learning has witnessed tremendous advancements in recent years, with applications ranging from image recognition to natural language processing. However, one persistent challenge that hampers the performance and generalization of machine learning algorithms is sample selection bias. This phenomenon occurs when the training and test data come from different distributions, leading to biased predictions and limitations in the algorithm's ability to adapt to new domains. Addressing sample selection bias is crucial for ensuring the robustness and reliability of machine learning models in real-world scenarios.

Sample selection bias poses unique challenges in machine learning. The biased data distribution can result in models that are trained on incomplete or distorted information, leading to suboptimal performance when deployed in different contexts. This issue is particularly relevant in domains where collecting comprehensive and unbiased training data is challenging or expensive. Therefore, it is imperative to develop techniques that can effectively mitigate the impact of sample selection bias on machine learning models.

Numerous research efforts have been devoted to understanding and mitigating the effects of sample selection bias in machine learning. Existing literature has explored various approaches, including domain adaptation techniques that aim to create models robust to biased data [3][4][6][7][8]. These techniques focus on adapting models to perform well on target domains that differ from the source domain. Notably, minimax estimation techniques, such as the Robust Bias Aware classifier (RBA)[8] and the Target Contrastive Pessimistic Risk Classifier (TCPR)[6], have shown promise in addressing sample selection bias.

While considerable progress has been made in developing domain adaptive models and minimax estimation techniques, several unanswered questions remain. It is essential to critically evaluate the effectiveness of these techniques in the presence of sample selection bias. Previous studies have primarily focused on theoretical aspects and algorithmic design, but the practical evaluation of these models in real-world scenarios is limited. Therefore, there is a pressing need to bridge this gap by empirically testing the performance of minimax estimation techniques and assessing their suitability for mitigating sample selection bias.

The main research question of this thesis is: "How effective are minimax estimation techniques in mitigating sample selection bias?" To answer this question, we will investigate several subquestions, including the comparison of minimax estimation techniques with traditional supervised learning methods in the presence of sample selection bias, the analysis of how different types of sample selection bias impact the performance of minimax estimation techniques, and the examination of how hyperparameters influence the effectiveness of these techniques in mitigating sample selection bias.

The primary contribution of this research is a comprehensive understanding of the effectiveness of minimax estimation techniques, particularly RBA and TCPR, in mitigating sample selection bias and the factors that impact their performance in domain adaptation. By empirically evaluating these techniques and addressing the unanswered questions, we aim to provide insights that can enhance the practical applicability of minimax estimation methods in real-world machine learning scenarios.

The remainder of this paper is structured as follows: In Section 2, we provide a detailed description of the research methodology, including data generation, model selection, and evaluation framework. Section 3 presents the experimental setup and provides pseudocode explanations. Next, we present our findings and discuss their implications in Section 4. Section 5 focuses on Responsible Research practices. Finally, we conclude by summarizing our contributions, discussing the results, and providing recommendations for future research in Section 6.

2 Research Methodology

In this section, we present the research methodology employed in our study, which encompasses various aspects such as data generation, model selection, and the evaluation framework. A robust and well-designed methodology is crucial for conducting reliable and informative experiments, ensuring the validity and integrity of our findings.

2.1 Data generation

Two different datasets were utilized for experimentation: the *make_moons* dataset from the *sklearn.datasets* library and the *breast cancer* dataset from the UCI machine learning repository

2.1.1 Make moons dataset

The *make_moons* dataset served as the primary dataset for most of our experiments. It is a two-dimensional dataset that

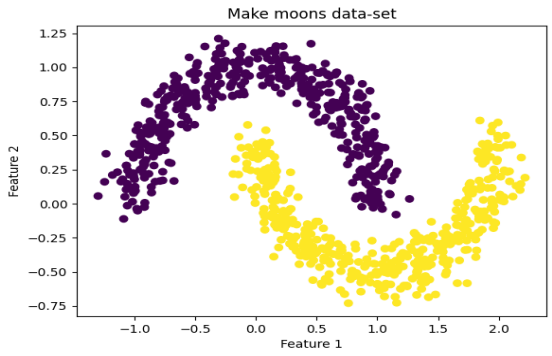


Figure 1: Visualisation of the *make_moons* dataset with *n_samples=1000*, *noise=0.1*

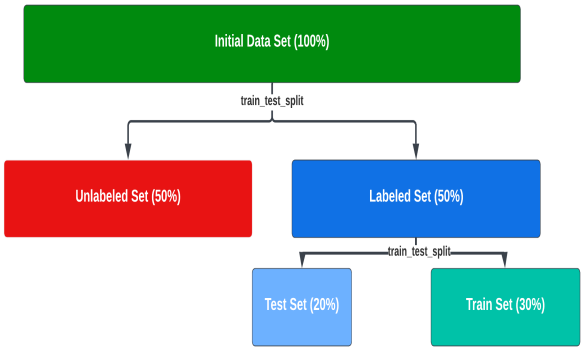


Figure 2: Splitting data into test, training, and unlabeled set. The data is split into unlabeled (50%), train (30%), and test (30%) sets by making use of the *train_test_split* function from *sklearn* library

enables easy visualization of the created bias. The dataset has two important parameters:

- **Noise:** This parameter controls the amount of random noise added to the data points. It allows us to introduce variability into the dataset, making it more realistic and challenging.
- **n_samples:** This parameter determines the total number of samples generated in the dataset. It provides flexibility in controlling the dataset size.

The *make_moons* dataset offers several benefits for our experimentation:

- **Two-dimensional nature:** The dataset's simplicity allows us to clearly observe and analyze the biases introduced.
- **Control over noise and sample size:** By adjusting the *noise* and *n_samples* parameters, we can create various scenarios to study the impact of bias on classification performance.

Figure 1 illustrates the "make moons" data-set.

2.1.2 UCI breast cancer dataset

The breast cancer dataset consists of 30 features that describe various characteristics of breast cancer tumors. It is a high-dimensional dataset widely used in machine learning research.

2.1.3 Data Partitioning

To conduct our experiments, we partitioned the data into three sets:

- **Unlabeled set:** This set represents the global domain, which comprises 50% of the overall data.
- **Train set:** This set represents the source domain, comprising 30% of the overall data. The experiments conducted introduced a bias on this dataset. The biased dataset was then used to train the selected models.
- **Test set:** This set represents the target domain, consisting of 20% of the initial data. We utilized this set to evaluate the performance of the classifiers trained on the biased source domain data. The train and test sets have different distributions since a bias was induced in the train set.

Figure 2 illustrates the data split, providing a visual representation of the partitioning process.

2.2 Model Selection

In the context of model selection, we will work on two domain adaptation models for addressing sample selection bias: the Robust Bias Aware classifier (RBA) [8] and the Target Contrastive Pessimistic Risk estimator (TCPR) [6]. In the following subsections we will discuss these classifiers in detail along with their parameters.

2.2.1 Robust Bias Aware Classifier

The Robust Bias-Aware (RBA) classifier is a minimax estimator designed to address sample selection bias in domain adaptation. It operates within a minimax estimation framework, aiming to minimize the risk for one classifier while an adversary maximizes the risk using another classifier. However, to ensure convergence and avoid divergent behavior, the adversary is constrained to select posteriors that align with the moments of the source domain's feature statistics. This constraint encourages the optimization process to capture the underlying distribution of the source domain.

The RBA classifier is controlled by several hyperparameters that influence its behavior. These hyperparameters include:

- **L2 Regularization** (`l2`): Controls regularization strength for preventing overfitting.
- **Order of Feature Statistics** (`order`): Determines the feature moment order used by the classifier.
- **Decaying Learning Rate** (`gamma`): Adjusts the step size during optimization for refined adjustments.
- **Convergence Threshold** (`tau`): Defines the minimum gradient change required for convergence.
- **Maximum Iterations** (`max_iter`): Sets the maximum number of optimization iterations.
- **Weight Clipping** (`clip`): Limits the range of importance weights to stabilize training.

These hyperparameters enable customization and fine-tuning of the RBA classifier according to the specific domain adaptation task and dataset. By appropriately adjusting these parameters, practitioners can control the regularization, optimization process, and convergence behavior of the RBA classifier, leading to improved performance and adaptability.

2.2.2 Target Contrastive Pessimistic Risk (TCPR)

The Target Contrastive Pessimistic Risk (TCPR) estimator is a minimax estimator specifically designed for domain adaptation tasks. Its primary focus is on improving the performance of the target classifier compared to the source classifier by contrasting their empirical target risks. By considering the difference in risks between the source and target classifiers, the TCPR estimator effectively excludes parameter settings that are known to produce worse risks than those of the source classifier. Formally, the TCPR estimator is defined as follows:

$$\hat{h}_T = \arg\min_{h \in H} \max_{q} \frac{1}{m} \sum_{j=1}^{m} \left(\ell(h(z_j), q_j) - \ell(\hat{h}_S(z_j), q_j) \right) \tag{1}$$

The performance and effectiveness of the TCR estimator can be influenced by the selection of its parameters. In the context of TCR, some of the important parameters to consider are:

- **Loss Function** (`str 'loss'`): Determines how the classifier's performance is measured and optimized.
- **L2-Regularization Parameter** (`float 'l2'`): : Controls the model's complexity and generalization capability.
- **Maximum Number of Iterations** (`int 'max_iter'`): Influences convergence and computational time.
- **Convergence Tolerance** (`float 'tolerance'`): Sets the criterion for stopping the optimization process.
- **Learning Rate** (`float 'learning_rate'`):Determines the step size of parameter updates during optimization.
- **Learning Rate Decay** (`str 'rate_decay'`): Defines the adjustment of the learning rate over time.

2.3 Evaluation

To answer the research question "How effective are minimax estimation techniques in the presence of sample selection bias?", we designed a comprehensive evaluation framework consisting of three subquestions. In this section, we discuss our approach to answering these subquestions and evaluating the performance of machine learning classifiers under sample selection bias.

2.3.1 Comparison with traditional ML models

To answer the first subquestion, we compared the performance of our chosen minimax estimation techniques, Robust Bias Aware classifier (RBA) and Target Contrastive Pessimistic Risk Classifier (TCPR), with traditional supervised learning methods. Specifically, we selected logistic regression as a linear classifier and support vector machine (SVM) with the radial basis function (RBF) kernel as a non-linear classifier. Initially, all classifiers were trained on unbiased data and tested using a test set which had the same distribution as the train set. Subsequently, we introduced sample selection bias into the training data and retrained the classifiers. By comparing their performances on the test set, which this time had a different distribution to that of the train set due to the biasing step on the train data, we assessed the impact of sample selection bias on these models.

2.3.2 Evaluation under Different Types of Bias

To answer the second subquestion, we conducted experiments to assess the performance of RBA and TCPR under different types of biases. We considered four types of biases:

1. **Survivorship bias**: Survivorship bias arises when the dataset only includes certain instances that meet specific criteria, leading to biased representation. We incorporated survivorship bias into the data and analyzed its impact on the classifiers.
2. **Covariate shift**: Covariate shift refers to a change in the input feature distribution between the source and target domains. We simulated covariate shift by modifying the distribution of input features, introducing a discrepancy between the training and test data. We assessed the models' performance under this type of bias.
3. **Class imbalance**: Class imbalance bias occurs when the data points are sampled in a way that creates an unequal distribution of classes, with one class having significantly fewer instances compared to the other. We introduced class imbalance into the data and examined its effect on the classifiers' performance.

By systematically evaluating the classifiers under these different types of biases, we gained insights into the robustness and effectiveness of RBA and TCPR in handling various real-world scenarios. The analysis allowed us to understand the impact of different biases on the classifiers' performance and identify any specific challenges associated with each bias type.

2.3.3 Evaluation Metrics

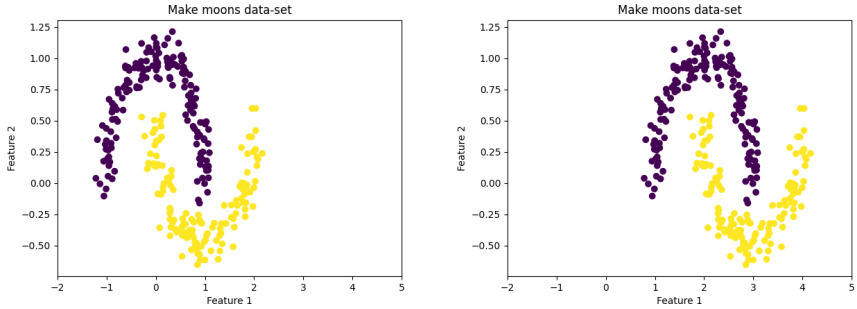
To ensure a comprehensive assessment of the classifiers' performance, we employed three evaluation metrics: F1-score, log loss, and Area Under ROC curve (AUC).

The F1-score provides a balanced measure of precision and recall, capturing the classifier's accuracy on both positive and negative instances.

Log loss assesses the probability estimates generated by the classifier, penalizing inaccurate predictions.

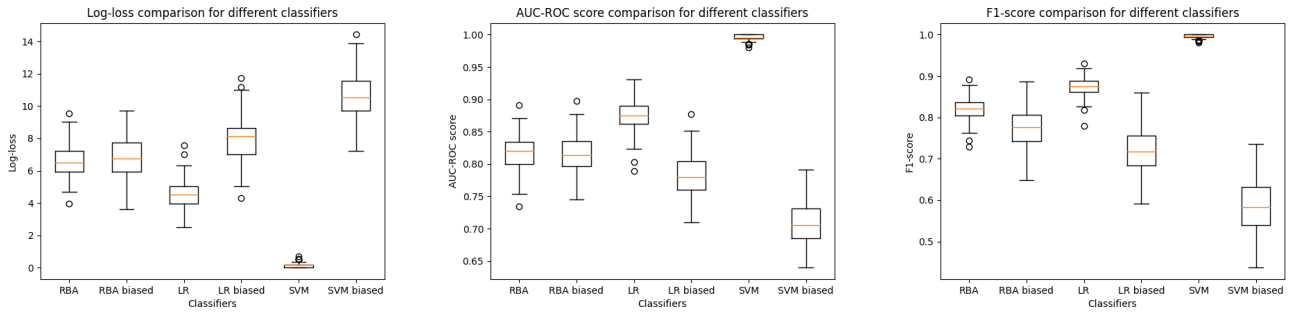
AUC measures the classifier's ability to distinguish between positive and negative instances across different threshold settings. By using multiple evaluation metrics, we obtained a holistic view of the classifiers' performance under sample selection bias.

In Section 3, we provide detailed descriptions of the experimental methodology, including the data generation process, model training procedures, and the specific evaluation framework employed to address the research questions outlined in this section.



(a) (b)

Figure 3: Visualization of the source domain before (a) and after (b) inducing covariate shift. The source and target domains are the same in (a), while (b) represents the source domain with introduced covariate shift.



(a) (b) (c)

Figure 4: Box Plots illustrating the impact of training on unbiased and covariate-shifted source domain on the prediction performance of RBA, LR, and SVM classifiers. Evaluation metrics include Log-loss (a), AUC-ROC score (b), and F1-score (c).

3 Experimental Setup and Results

In this section, we discuss the setup of the experiments conducted to test the performance of RBA and TCPR compared to Logistic Regression and SVM. We provide the pseudocode for different parts of the experiments and visualize the effect of inducing different biases on the data. Finally, we compare the results of different classifiers.

The data used in all experiments was partitioned following the procedure described in section 2.1.3. The data was split into unlabeled, train, and test sets, where the unlabeled set represents the global domain, the train set represents the source domain, and the test set represents the target domain. Bias was induced only on the train set, creating different distributions between the source and target domains.

3.1 Covarite Shift

The first experiment focuses on inducing a covariate shift in the data using the *make moons* dataset with n_samples = 1000 and noise = 0.1.

In the experiment, we followed a specific procedure to investigate the impact of covariate shift on the performance of different classifiers. Initially, the dataset was split into unlabeled, train, and test sets using the predefined splitting scheme. The classifiers were then trained on the unbiased source data, where no bias was induced, and their performance was evaluated on the target data. Since the source and target domains had the same distribution at this stage, the evaluation provided a baseline measure of classifier performance.

To induce bias, we introduced a shift in the source domain by altering the values of the first feature for all data points. By applying a fixed shift value, we created a discrepancy between the source and target domains, resulting in different distributions. Next, the classifiers were retrained using the shifted source domain, and their performance was once again evaluated on the target data. This allowed us to assess how the introduced covariate shift affected the classifiers' ability to generalize to the target domain.

To account for the variability in results, the entire process was repeated 100 times with random splits of the data. The performance metrics for each classifier on biased and unbiased data were stored separately in lists. Finally, we visualized the results using box plots, enabling a comprehensive comparison of the classifier performance on biased and unbiased data.

Algorithm 1 provides a clear outline of the steps taken to induce covariate shift by shifting the source data. This algorithm serves as a reference to understand the specific proce-

dure employed for introducing bias in the source domain.

Figure 3 visually illustrates the effect of covariate shift on the source data. As depicted in the figure, the shift causes the source data to shift towards the right, thereby creating a noticeable difference in the distribution compared to the original data.

By referring to Algorithm 1 and examining Figure 3, one can gain a comprehensive understanding of the method used to induce covariate shift and observe its impact on the source data.

Experiments were conducted using three different evaluation metrics and the performance of RBA, LR, and SVM on both biased and unbiased data was analyzed.

Initially, when trained on the source domain with the same distribution as the target domain, RBA exhibited the worst performance compared to Logistic Regression and SVM. However, when we introduced covariate shift in the source data and retrained the classifiers, the results were striking. RBA demonstrated remarkable adaptability to the domain shift, outperforming both LR and SVM. In fact, it transformed from the worst performing classifier on unbiased data to the best performing classifier when trained on source data with a different distribution than the target data.

These findings highlight the robustness and effectiveness of RBA in handling covariate shift scenarios. The detailed results are presented in Figure 4, which showcases box plots illustrating the performance of the classifiers across different evaluation metrics.

Algorithm 1 Covariate Shift

Require: X_{train}
$shift\_value \leftarrow 2.0$
$X_{train\_biased}[:, 0] \leftarrow X_{train}[:, 0] + shift\_value$
return X_{train_biased}

It is worthy to note that one of the minimax classifiers under investigation, TCPR, is omitted from the results. This is becasue the implementation of TCPR we used for our experiments continue to throw an Assertion error during the training.

3.2 Survivorship bias

This section provides a detailed overview of the experimental setup and procedure used to evaluate the performance of minimax methods in the presence of survivorship bias. The aim of this study was to investigate the impact of sample size and bias on the classification performance, employing the make moons dataset with a noise level of 0.1.

To investigate the relationship between sample size and classification performance, the *sample_size* parameter of the make moons dataset was systematically varied from 100 to 3000, with a step size of 100. The noise level was set to 0.1 to introduce some variability in the data.

For each sample size, the following steps were performed 10 times to ensure the reliability of the results:

1. The *make moons* dataset was split into unlabeled, train, and test sets as described in figure 2. This procedure ensured that the distribution of the data remained consistent throughout the experiments.
2. Four types of classifiers were employed in this study: minimax classifiers (TCPR and RBA) and traditional classifiers (Logistic Regression and SVM). Initially, these classifiers were trained on the unbiased source data and evaluated on the target data, which shared the same distribution as the source data since no bias was introduced. The resulting scores were recorded in separate lists for each classifier.
3. To introduce survivorship bias, the classifiers were subsequently trained on biased source data. Samples from the train set were only chosen if the value for feature 0 was less than -0.5 or greater than 1.5 in order to obtain a biased train set. The biasing procedure is detailed further as pseudocode in Algorithm2. After training, the classifiers were tested on the target data, which now exhibited a different distribution due to the introduced bias. The resulting scores were once again stored in separate lists for each classifier.

The collected scores from each experiment were utilized to calculate the mean and standard deviation for each classifier at each sample size, considering both the biased and unbiased scenarios. These statistical measures provided insights into the average performance and the variability of the classifiers under different conditions.

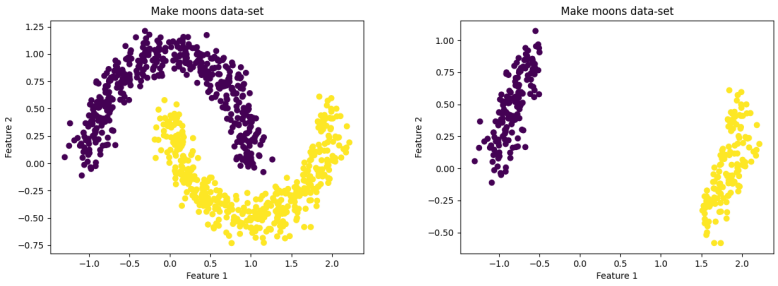
The mean scores and standard deviations were plotted, as depicted in Figure 6, allowing for a visual examination of how the classifiers' performance was affected by sample size and survivorship bias. The results provided insights into the behavior and effectiveness of the minimax methods and traditional classifiers in the presence of survivorship bias and shed light on the influence of varying sample sizes.

Based on the analysis of Figure 6, the performance of the RBA, SVM, and LR classifiers in the experiment described above is examined with respect to the change in performance as sample size increases, both when trained on biased and unbiased source data.

Algorithm 2 Survivorship bias

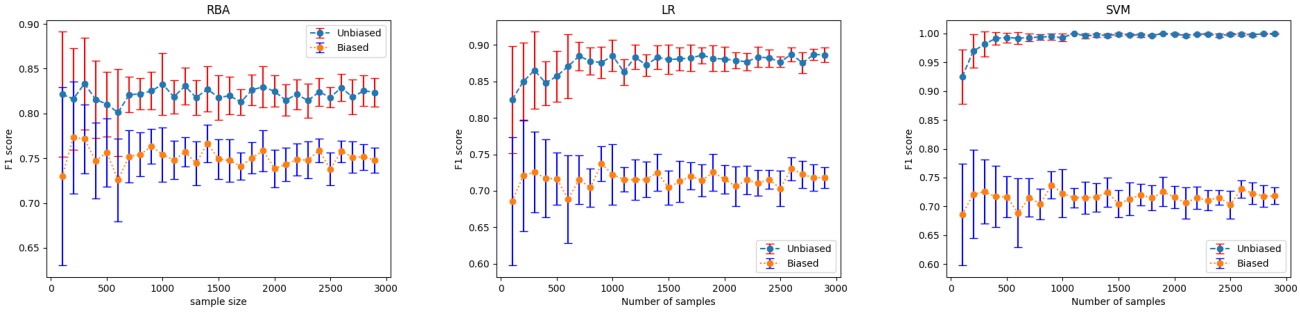
Require: X_{train}, Y_{train}
$indices \leftarrow emptylist$
$n \leftarrow size(X_{train})$
for $i = 0$ *to* $n - 1$ **do**
  if $X_{train}[i, 0] < -0.5\ OR\ X_{train}[i, 0] > 1.5$ **then**
    append i to the end of $indices$
  end if
end for
$X_{train\_biased} \leftarrow X_{train}[indices]$
$Y_{train\_biased} \leftarrow Y_{train}[indices]$
return $X_{train\_biased}, Y_{train\_biased}$

The results show a common trend among all three classifiers: as the sample size increases, the variance (standard deviation) of their performances decreases. This is evident from the decreasing size of the vertical bars in the graph. Initially, when the sample size is relatively small, the vertical bars are



(a) (b)

Figure 5: Visualization of the source domain before (a) and after (b) inducing survivorship bias.



(a) (b) (c)

Figure 6: Impact of increasing sample size and survivorship bias on the performance of classifiers (F1-score). (a) presents the performance of RBA, (b) displays the performance of LR, and (c) illustrates the performance of SVM.

larger, indicating higher variability in the classifier's performance. However, as the sample size increases, the vertical bars become smaller, indicating reduced variability.

This decrease in variance with increasing sample size is expected since larger sample sizes reduce the sensitivity to individual samples and lower the chances of overfitting. It suggests that the classifiers become more robust and stable as more data becomes available for training.

One noteworthy observation is the performance of the RBA classifier. On average, as the sample size increases, its performance remains relatively consistent. This suggests that the RBA classifier can achieve comparable performance even with smaller training data. However, when trained on biased data, the RBA classifier shows a drop in its prediction score. Notably, the magnitude of this decrease is the smallest among the three classifiers.

Comparatively, the SVM classifier exhibits a significant drop in its prediction ability when trained on biased data, while the LR classifier also shows a considerable decrease in prediction F1 score.

From this information, we can conclude that the RBA classifier demonstrates a relatively consistent performance as the sample size increases, indicating its robustness to limited training data. Additionally, when faced with survivorship bias, the RBA classifier shows the smallest decrease in prediction score compared to the SVM and LR classifiers. This suggests that the RBA classifier is more resilient to the introduced bias and could be a preferred choice when dealing with survivorship bias in classification tasks.

Due to its unpredictable behavior and consistent Assertion errors during random data splits, the TCPR classifier was excluded from the analysis and results presented in Figure 6.

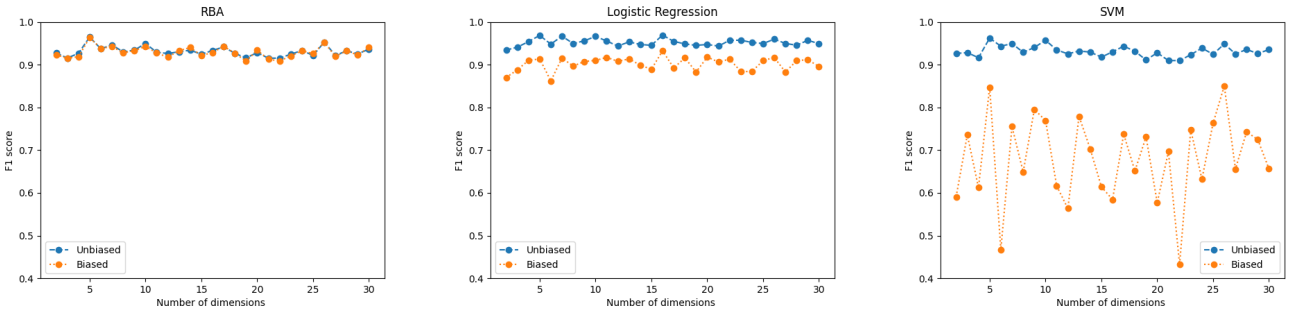
3.3 Class imbalance

In this section, we aim to evaluate the effectiveness of minimax estimation techniques in the presence of class imbalance and explore the impact of increasing dimensions on the classifiers' performance. For our experiments, we utilize the well-known breast cancer dataset from the UCI library.

The breast cancer dataset consists of 30 features, and to examine the classifiers' behavior in lower dimensions, we employed a dimensionality reduction technique called Principal Component Analysis (PCA). PCA is a method that transforms high-dimensional data into a lower-dimensional representation while preserving the most important information and capturing the underlying structure of the data [5].

To assess the influence of class imbalance and the number of dimensions on the classifiers, we follow these steps:

1. We split the data into unlabeled, source, and target sets using the splitting technique depicted in Figure 2. By performing the data split before the dimensionality reduction step, we ensure that no information leaks from



(a) (b) (c)

Figure 7: Effect of increasing number of dimensions and class imbalance on the performance of classifiers. (a) RBA, (b) Logistic Regression, and (c) SVM.

the target set during the reduction process.

2. After the data is split, we apply PCA to reduce the dimensions of all three sets: unlabeled, source, and target.
3. Subsequently, we train the classifiers on both the biased source data, characterized by class imbalance, and the unbiased dataset with balanced classes. The class imbalance is induced by sampling the benign class (label 0) nine times more frequently than malignant class data points. This setup enables us to examine how the classifiers cope with imbalanced class distributions.
4. To ensure robustness of our findings, we repeat the training process 10 times for each dimension, maintaining the same class biasing probabilities. We then average the results of the classifiers' performance.

The performance plots in Figure 7 illustrate the impact of class imbalance and increasing dimensions on the classifiers.

SVM performs well when the classes are balanced, exhibiting consistent scores as the number of dimensions increases. However, its performance significantly deteriorates when faced with imbalanced classes. The classifier struggles to maintain consistent performance, as shown by the fluctuating average score with increasing dimensions.

LR demonstrates good performance in both the balanced and unbalanced cases, with only a slight drop in f1-score observed in the presence of class imbalance. Furthermore, LR exhibits consistent scores as the number of dimensions increases, indicating its ability to handle high-dimensional data effectively.

RBA stands out by delivering remarkable performance in both the unbalanced and balanced cases. It showcases minimal performance drop even when trained on imbalanced source data. Notably, RBA maintains consistent performance as the number of dimensions increases. This robust behavior of the RBA classifier underscores its effectiveness in handling class imbalance and increasing dimensions.

3.4 Parameter Analysis of minimax estimators

In this section, we will outline the experimental design used to test the effect of parameters in RBA in the presence of sample selection bias. Specifically, we focused on two parameters: the *max_iter* parameter and the *learning_rate*.

3.4.1 Dataset and Biasing Procedure

For this experiment, we utilized the *breast cancer dataset*. To introduce sample selection bias, we biased the values of the third feature in the data by adding a constant value of 30. This biasing procedure was applied to each data point.

3.4.2 Parameter Exploration

We conducted experiments to investigate the impact of different values of the *learning_rate* parameter on the classifier's performance. We tested four learning rate values: 0.0001, 0.001, 0.005, and 1.0. Each learning rate value was evaluated separately, and the resulting scores were plotted on the same graph for comparison.

For the *max_iter* parameter, we tested the classifier's performance using values ranging from 20 iterations to 1000 iterations with a step size of 20 iterations. The experiment was repeated 10 times for each *max_iter* value.

3.4.3 Experimental Procedure

For each *max_iter* value, the following steps were repeated 10 times:

1. Random Data Split: The dataset was randomly split using the procedure described in Figure 2. This ensured the variability of the data samples in each iteration.
2. Classifier Training: We trained the RBA classifier on the biased data using the biasing procedure mentioned earlier, with the corresponding parameter combination (one of the four *learning_rate* values and the current *max_iter* value).
3. Performance Evaluation: The trained classifier was then evaluated on target data, which belonged to a different domain. This evaluation assessed the classifier's ability to generalize and classify unseen samples accurately.

The scores obtained from the 10 repetitions were averaged to provide a representative performance measure for each parameter combination.

The average scores obtained were plotted against the*max_iter* values, with four distinct plot lines representing the four different *learning_rate* values (Figure 8).

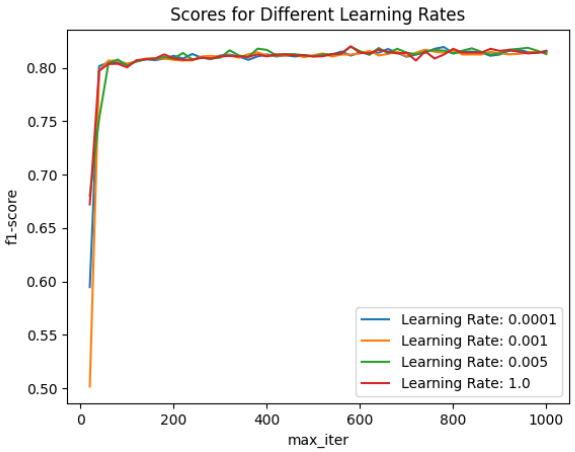


Figure 8: Impact of increasing *max_iter* parameter and different *learning_rates* (0.0001, 0.001, 0.005, 1.0) on the performance (F1 score) of RBA when trained on source data with covariate shift.

3.4.4 Results

The RBA classifier demonstrated rapid adaptation across all four learning rate values, as evidenced by a sharp increase in accuracy from *max_iter* = 20 to *max_iter* = 40 for each learning rate.

After reaching a certain point, increasing the *max_iter* value did not yield significant improvements in classification performance. The classifier's accuracy appeared to plateau, suggesting diminishing returns beyond a certain number of iterations.

The *learning_rate* parameter did not exhibit a substantial impact on the classifier's performance. The classification accuracy remained relatively consistent across the four *learning_rate* values, indicating that the *learning_rate* parameter did not significantly affect the classifier's ability to adapt and generalize.

Based on these observations, we can conclude that the RBA classifier demonstrates efficient adaptation to the presence of sample selection bias. However, the performance improvement plateaus beyond a certain number of iterations, and the choice of learning rate does not have a substantial impact on the classifier's performance in this experimental setup.

4 Responsible Research

This study has been committed to upholding the principles of responsible research and ethical conduct throughout its duration. This section aims to provide an overview of the ethical considerations and measures taken to ensure the reproducibility of the methods employed.

4.1 Ethical Considerations

During the course of this research, ethical considerations have been paramount in guiding the design, implementation, and reporting of the study. The principles outlined in the [1] have served as a fundamental framework for upholding ethical standards in this research project. Adhering to the principles of honesty, reliability, and respect for research subjects and participants has been of utmost importance.

Moreover, the case of Diederik Stapel [2] serves as a stark reminder of the consequences of research misconduct. Stapel's fraudulent practices, where he fabricated data for multiple publications, emphasize the need for vigilance in maintaining scientific integrity and ethical conduct. By learning from such cases, we are reminded of the significance of adhering to rigorous research practices and the responsibility we bear as researchers to maintain the highest ethical standards.

4.2 Reproducibility of Methods

Ensuring the reproducibility of research methods is a fundamental aspect of scientific inquiry. Reproducibility allows for the validation and verification of findings, promotes transparency, and facilitates the advancement of knowledge through cumulative research efforts. As emphasized by the Yale Law School Roundtable on Data and Code Sharing [9], the ability to reproduce computational results is crucial in the face of technological advancements and evolving scientific landscapes.

4.3 Open Science and Data Sharing

In line with the principles of open science, I believe in the importance of making research findings accessible to the scientific community and the public. Whenever possible and appropriate, I have shared my datasets, code, and analysis scripts to facilitate reproducibility and promote further exploration and collaboration. By sharing these resources, I aim to contribute to the collective knowledge and encourage the robustness of scientific inquiry.

Conducting responsible research involves a commitment to ethical principles, meticulous data management, methodological rigor, and the promotion of reproducibility. By adhering to these principles and practices, I have strived to ensure the integrity, transparency, and reliability of the research presented in this thesis.

5 Discussion

The results obtained from these experiments highlight the strengths and limitations of RBA as a minimax classifier. RBA exhibited robustness and adaptability in the presence of covariate shift, survivorship bias, and class imbalance. Its ability to effectively handle domain shifts and adapt to changes in the underlying data distribution makes it a promising choice for real-world applications where data distributions may vary over time or across different domains.

Furthermore, RBA demonstrated resilience to survivorship bias, which is a common issue in many classification tasks where certain samples are missing due to various factors. The classifier's performance remained relatively stable even when trained on biased data, suggesting its capability to generalize well and make accurate predictions on unseen instances. This characteristic is particularly valuable in domains such as healthcare, finance, and social sciences, where data collection processes often introduce biases and missing information.

In terms of class imbalance, RBA showed promising results by achieving high performance in both balanced and imbalanced scenarios. Class imbalance is a common challenge in classification problems where the number of instances in one class significantly outweighs the other(s). RBA's ability to maintain consistent performance in the presence of imbalanced classes highlights its effectiveness in handling such scenarios and its potential to provide reliable predictions even when the data is heavily skewed.

Additionally, the parameter analysis revealed the impact of max iter and learning rate on RBA's performance. However, further exploration and fine-tuning of these parameters are necessary to fully understand their effects and identify optimal values for different datasets and biasing procedures. Additionally, other hyperparameters of RBA, such as regularization strength and learning rate decay, could be investigated to further enhance its performance and generalizability.

Although RBA has demonstrated superior performance in various experimental settings, it is important to note some limitations and areas for improvement. The analysis excluded TCPR, a minimax classifier, due to instability and assertion errors during training. Further attempts to evaluate TCPR were made using multiple datasets, but the classifier implementation showed unpredictable behavior.

One experiment successfully tested TCPR's ability to adapt when trained on biased data with covariate shift using the breast cancer dataset(results shown in 9. However, it should be noted that the TCPR implementation threw exceptions when other features were biased, indicating its unpredictability. Therefore, the results of this experiment should be interpreted with caution.

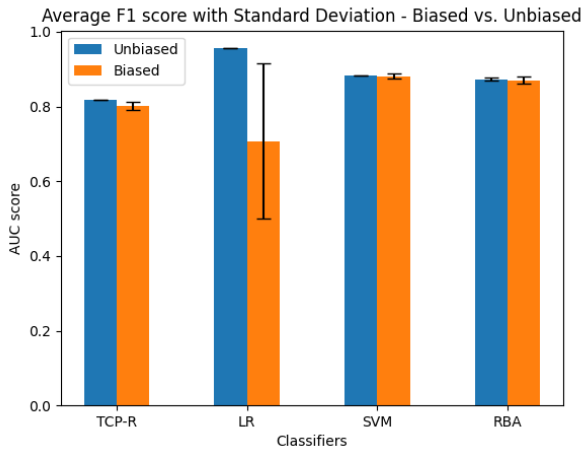


Figure 9: Comparison of TCPR, LR, SVM, and RBA classifiers on unbiased and biased source data with covariate shift.

These limitations highlight the need for addressing the instability and assertion errors in the TCPR implementation and improving its overall predictability. By overcoming these issues, a comprehensive evaluation of TCPR's performance and potential benefits can be achieved, providing a more accurate assessment of its effectiveness in real-world classification tasks.

In conclusion, the experiments conducted in this study demonstrate the effectiveness of RBA as a minimax classifier in challenging scenarios, including covariate shift, survivorship bias, and class imbalance. RBA exhibited adaptability, resilience, and consistent performance, making it a promising choice for real-world classification tasks. Further research and refinement of RBA, along with exploration of other minimax classifiers, could provide valuable insights and advancements in the field of robust and bias-resistant classification algorithms.

6 Conclusion

In this thesis, we investigated the performance of minimax estimation techniques, specifically the Robust Bias Aware Classifier under sample selection bias. Our findings indicate that RBA outperformed traditional supervised learning algorithms LR and SVM, demonstrating robustness in handling these challenges.

Specifically, RBA showed superior performance when trained on source data with covariate shift. However, further exploration of the TCPR classifier is recommended to address its instability and assertion errors during training. Additionally, future work should involve testing the effectiveness of minimax estimation techniques with increasing distance between the source and target domains.

We also examined the impact of max_iter parameter and learning rates on RBA's performance, revealing that increasing max_iter beyond a threshold did not significantly improve results, while a learning rate of 0.001 yielded the best performance.

Class imbalance negatively affected all classifiers, highlighting the need for techniques to address this issue and improve overall performance.

For future research, it is recommended to explore the other classifiers, investigate effective techniques to handle covariate shift, study feature selection and dimensionality reduction methods, consider additional evaluation metrics, and conduct experiments on larger datasets. These efforts will contribute to a more comprehensive evaluation of classifier performance and provide insights into enhancing their capabilities.

In conclusion, this thesis provides valuable insights into the performance of RBA, highlighting RBA's effectiveness and suggesting directions for further research. By testing TCPR more extensively and exploring minimax estimation techniques, researchers can expand the understanding of classifier behavior and improve their adaptability to varying source and target domains.

References

[1] K Algra, L Bouter, A Hol, J van Kreveld, D Andriessen, C Bijleveld, R D'Alessandro, J Dankelman, and P Werkhoven. Netherlands code of conduct for research integrity 2018, 2018.

[2] Ewen Callaway et al. Report finds massive fraud at dutch universities. *Nature*, 479(7371):15, 2011.

[3] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation.

Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020, pages 877–894, 2021.

[4] Baochen Huang, Yi Dai, and Xiaolong Wang. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, pages 609–617, 2016.

[5] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[6] Wouter M. Kouw and Marco Loog. Target contrastive pessimistic risk for robust domain adaptation. 2017.

[7] Wouter M Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019.

[8] Anqi Liu and Brian Ziebart. Robust classification under sample selection bias. *Advances in neural information processing systems*, 27, 2014.

[9] Yale. Law school roundtable on data and code sharing reproducible research. *Computer Science and Engineering*, 12, 2010.