Evaluating the Effectiveness of Importance Weighting Techniques in Mitigating Sample Selection Bias

Andrei Camil Tociu[1]

Supervisor(s): Joana de Pinho Gonçalves[1]**, Yasin Tepeli**[1]

[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: Andrei Camil Tociu
Final project course: CSE3000 Research Project
Thesis committee: Joana de Pihno Gonçalves, Yasin Tepeli, Julian Urbano Merino

Abstract

Importance weighting is a class of domain adaptation techniques for machine learning, which aims to correct the discrepancy in distribution between the train and test datasets, often caused by sample selection bias. In doing so, it frequently uses unlabeled data from the test set. However, this approach has certain drawbacks: it requires retraining for each new test set and fails when the number of test samples is very small. Therefore, we seek to study the performance of importance weighting techniques when the unlabeled data comes from an underlying domain, instead of one specific test set. We propose an evaluation framework inspired from scenarios traditionally known for posing difficulties to importance weighting and apply it to two popular algorithms, KMM and KLIEP. Our results reveal that both algorithms produce statistically significant classification improvements in most experiments. However, their performance is highly dependent on the characteristics of the dataset and the sampling bias. In particular, class overlap seems to influence adaptation ability in the case of unequal conditional probabilities of the source and target domains, while the "intensity" of the sampling bias is an important confounding factor when the train set size is small.

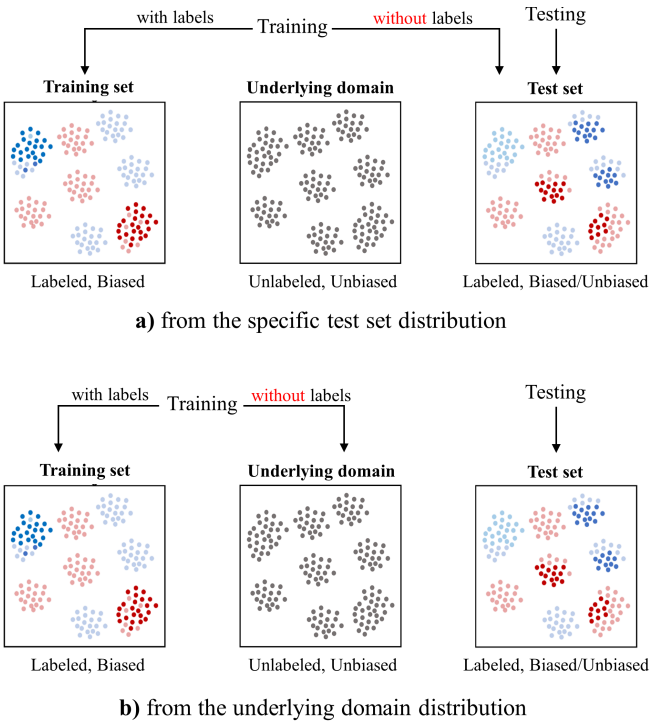


Figure 1: Learning approaches for domain adaptation based on the provenance of the unlabeled train data. Example shown for a binary classification task (red and blue classes), where darker-colored samples are the ones picked through sample selection bias and the gray-colored samples are unlabeled. The approach used in this paper is that from (b).

1 Introduction

A common assumption in supervised machine learning is that the train and test sample points are drawn independently and identically according to the same probability distribution. However, *sample selection bias* causes this assumption to fail in many practical situations, either due to limitations in uniformly collecting data from the entire domain or because the domain from which the available data originates is unknown [10]. In turn, the discrepancy in distributions causes the generalisation ability of many popular classifiers to degrade [5]. Some noteworthy fields affected by this issue are summarised in [10] and include econometrics [7], clinical trials [8], and gene sequencing in bioinformatics [22].

Importance weighting represents a popular *domain adaptation* technique for correcting this discrepancy in distributions by assigning a weight to the cost of error of each train point, where a large weight indicates that the sample is deemed highly relevant for the test set distribution [10]. Research effort focused so far on matching the distribution of the train set to that of a specific test set by using unlabeled data points from the test set [3, 10]. However, this approach presents several shortcomings, the most important being its lack of generalisability to different test sets, since it needs retraining for each of them. Another hard constraint is that the test set must be known beforehand and contain an adequately large number of samples. A largely unexplored alternative that could potentially avoid these issues is to use unlabeled samples from the underlying domain distribution of the data. Adapting once to the underlying domain would intuitively imply adaptation to all its "subdomains" too, represented by the differently sampled test sets. Moreover, using a large enough underlying domain would remove constraints on the size of the test sets. We use the term *source* for the domain that importance weighting adapts and *target* for the domain that it adapts to (and from which the unlabeled data is sourced). Therefore, in the classic approach the target domain consists of the sampled data points in the test set, whereas in our approach it consists of the underlying domain from which the data points originate (Figure 1).

The aim of this paper is to determine ***how effective is importance weighting in mitigating sample selection bias when the unlabeled data is sourced from the underlying domain instead of a particular test set***. To that end, we answer three sub-questions on the effectiveness of importance weighting in different scenarios:

1. *(SQ1) unequal conditional probabilities of the source and target distributions*
2. *(SQ2) small train sample size*
3. *(SQ3) high-dimensional data*

Our contribution to the state of the knowledge is two-fold. First, we introduce an empirical evaluation framework for determining how well importance weighting performs in scenarios traditionally known for posing difficulties. Its methodology is generic in nature so it can be applied to any importance weighting technique. Second, we apply the framework to two popular importance weighting methods, Kernel Mean Matching (KMM) [9] and the Kullback-Leibler Estimation Procedure (KLIEP) [17], comparing their performance and identifying both strengths and weaknesses.

The remainder of this paper is structured as follows. In Section 2 we analyse related research, including the main cases in which importance weighting underperforms. The evaluation framework is introduced in Section 3. Section 4 presents the results of applying the framework to two importance weighting techniques, while Section 5 places them into context. Then, Section 6 briefs on the ethical aspects of the experiments and their reproducibility. Finally, Section 7 concludes by summarising the paper and discussing some potential directions for further research.

2 Related Work

Mitigation of sample selection bias represents a rich field of study, impacted by both the type of distribution discrepancy studied and the particularities of the specific domain adaptation approach used, in our case importance weighting.

2.1 Sample Selection Bias

Sample selection bias is classified by [23] in three types, depending on the source of bias: (1) bias only depends on the feature vector x, (2) bias only depends on the label y, and (3) bias depends on both x and y. It has been argued that type (1) is "the most important sample selection bias case in the practice of classifier learning" [23, p. 2] and this reflects in the body of literature. In particular, importance weighting techniques make the fundamental, simplifying assumption that the conditional probability distributions of both the source (S) and target (T) domains are equal, $P_S(y|x) = P_T(y|x)$. Therefore, they infer the weight w of a sample based solely on its feature vector, as the ratio of the target and source marginal distributions, $w(x) = P_T(x)/P_S(x)$ [9, 10].

Even within the scope of type (1), further particularities in the dataset structure determine the impact of bias on the performance of a given classifier [5]. Of great importance is whether all features in x are used for biasing, or just a subset. Most research evaluates performance by biasing only one of the features [3, 16, 23], but some cases exist in which the entire feature vector was used [9].

There have been some limited attempts at studying the performance of importance weighting under sampling bias type (2) as well. For example, KMM was shown to still improve test error given imbalanced train sets [9]; however, the study probed only one sampling scheme (10-90% class sampling proportions) on a single dataset.

Lastly, the impact of sampling bias depends on the classifier and its learning equation. Linear classifiers are asymptotically immune to sampling bias type (1) when the data points in the source domain are "linearly" separable [5] mainly because their prediction is not based on a distribution over the entire input space (i.e. $P(x)$) [23]. Moreover, the impact of sampling bias as the train set size increases to infinity is best visible when the model is mis-specified (e.g. using a linear classifier for a non-linear classification task); otherwise, the unadapted classifier converges to the importance-weighted one [10, 21].

2.2 Shortcomings of Importance Weighting

Unequal conditional probabilities As discussed earlier, importance weighting makes the simplifying assumption that the conditional probabilities remain unchanged between the source and target distributions, which corresponds to a type (1) bias. As explained above, to our knowledge, the only research done into the performance of importance weighting when this assumption is violated is that from [9] for KMM.

High-dimensional data One approach for calculating $w(x)$ is to estimate the source and target densities separately and subsequently compute their ratio. However, this solution is known to underperform in the case of high-dimensional data, when measuring the degree of alignment of the two distributions is hard [10, 17]. An alternative is to directly infer the weights through an optimisation procedure for minimising different metrics of distribution discrepancy [10]. Both KMM and KLIEP belong to this latter category and use as metrics the Maximum Mean Discrepancy and the Kullback-Leibler divergence, respectively. However, [17] shows empirically that these approaches are also prone to the curse of dimensionality if tuned incorrectly.

Performance bounds The performance difference between an optimal classifier, trained on the target domain, and an importance-weighted one was bounded by [2] (Theorem 3) with a certain probability to a value which depends, among others, on the train sample size and the divergence between the source and target distributions. The bound indicates that the more divergent the distributions are, the larger the train set required by importance weighting to maintain the same difference level [10]. Simultaneously, increasing the train sample size while maintaining the divergence constant should theoretically decrease the difference to the optimum [10]. Concerning this latter case, experiments in [17] showed indeed a decrease in the error of the importance-weighted classifier, but no comparison to the optimum was done. Because the performance bound discussed above depends on a probability factor, it is interesting to explore also the empirical impact that a small train set size has on the adaptation performance.

High weight variance The importance weights can be used as explanatory factors of how domain adaptation is achieved, indicating both how much bias correction and where in the dataset is applied [10]. Therefore, their

values can be compared between typical success and failure cases to interpret how the method behaves [10]. For a large weight variance, it was observed that a few samples are assigned extremely high weights and end up dominating the learning process [2, 3]. In turn, this means that "the effective sample size drops" [10, p. 14], causing poor classification performance. This phenomenon was shown to appear especially when the regions with high data density in the target domain are not contained within the ones in the source domain [2]. To achieve successful adaptation, [10] suggested that weights should vary smoothly around value 1, but weight values are known to be unbounded in many real-life situations [2, 3].

3 Evaluation Framework

The proposed evaluation framework contains three test cases, each evaluating the effectiveness of importance weighting techniques in one of the scenarios:

- unequal conditional probabilities of the source and target distributions;
- varying train sample sizes;
- high-dimensional data.

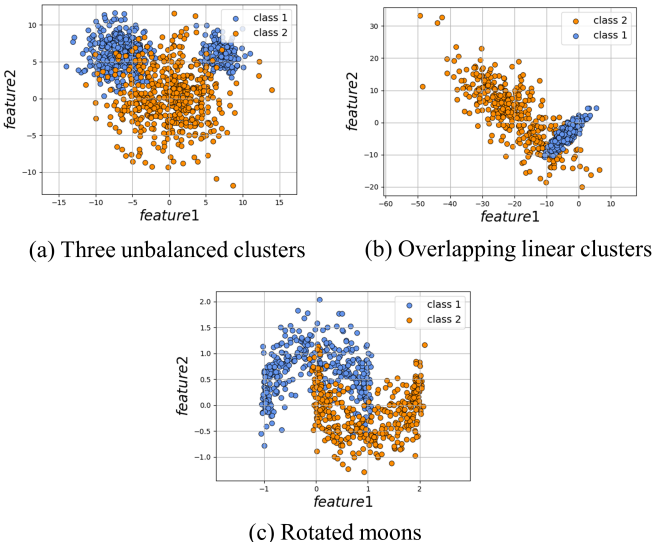
We focus on a classification scenario in which the impact of sample selection bias is known to be significant: a linear learner for a non-linear classification task (see discussion in Section 2.1). Therefore, we use datasets that have partially overlapping classes and do not present an intuitively linear decision boundary. For the choice of linear classifier we select logistic regression due to its simplicity. Lastly, to create the domain adaptation scenario, samples of each dataset are split up randomly into the underlying domain (50%), the train set (40%) and the test set (10%).

In general we estimate classifier performance by the proportion of test samples classified correctly (i.e. accuracy). Results are the average over 30 train-test splits in the form of random sub-sampling. We benchmark the importance-weighted classifiers against two others: one trained on the underlying domain (i.e. optimal) and another trained on the biased train set (i.e. unweighted).

The specifics of each test case are described below.

3.1 Test Case 1: Unequal Conditional Probabilities

This test aims to evaluate how the classification performance varies depending on the difference between the source and target conditional probabilities, so on the amount of bias introduced in the sample labels. For this, we generate three synthetic binary classification balanced datasets, each consisting of 3000 samples with two features, shown in Figure 2 (see Table A.1 for a formal description). We induce bias in the train labels by randomly sub-sampling each of the two classes at varying complementary ratios, ranging from 50-50% to 2-98%, at steps of 2%. Because the original set from which we subsample is balanced, the train sample size remains constant for all class imbalance ratios and ensures protection against confounding learning factors. We compute accuracy for each class sampling ratio in part.



(a) Three unbalanced clusters (b) Overlapping linear clusters

(c) Rotated moons

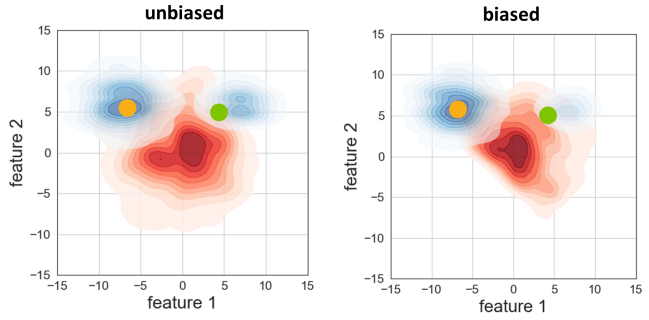
Figure 2: A visualisation of the the synthetic classification datasets used in the experiments. Class 1 is marked in blue and class 2 in red.

3.2 Test Case 2: Varying Train Sample Sizes

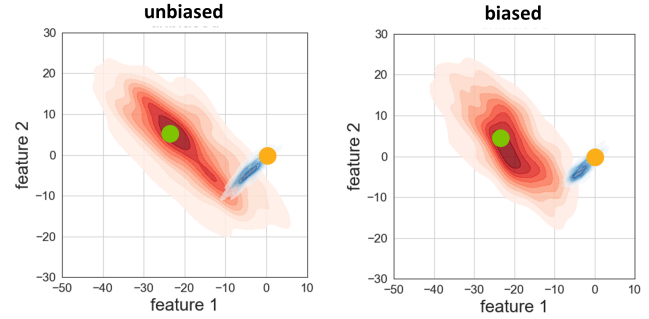
We aim to evaluate the robustness of importance weighting when the number of samples in the biased train set is scarce. We consider a biasing scheme for the datasets in Figure 2 that operates on the entire feature vector $x = (x_1, x_2)$ of the train samples. To achieve this, we first define for each class a point with coordinates $(\Delta_{x_1}, \Delta_{x_2})$ in the 2D plane described by the feature vector. We pick samples from each class depending on how close they are to $(\Delta_{x_1}, \Delta_{x_2})$: the probability of a sample with feature vector $x = (x_1, x_2)$ to be selected (i.e. $s = 1$) decreases exponentially with the Manhattan distance to $(\Delta_{x_1}, \Delta_{x_2})$ [19]. Lastly, we multiply the Manhattan distance with a factor $b \in \mathbb{R}$ in order to better control its intensity. The formula of the sampling probabilities is then $P(s = 1 \mid x) = e^{-b*(|x_1 - \Delta_{x_1}| + |x_2 - \Delta_{x_2}|)}$. Figure 3 shows the probability density function of the datasets pre and post biasing: for set (a) we prefer the left cluster of class 1 and the upper-right points in class 2; for set (b) we favour points generally closer to the center of the clusters; for the rotated moons in set (c) we pick more points from the central, overlapping regions. The values of $b, \Delta_{x_1}, \Delta_{x_2}$ used in the experiments are available in Table A.2.

We compute classification accuracy for models trained on various proportions of the train set, from 100% to 2%, at steps of 2%. When diminishing the sample size we ensure that the class proportions remain balanced, in other words that feature bias does not infer label bias as well. We also introduce an additional benchmark for the scores, namely the performance of a classifier trained on

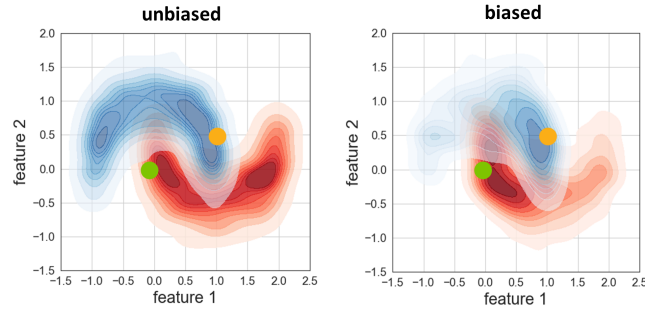
the same diminishing train set size as the other models, but unsubjected to any type of sampling bias.



(a) Three unbalanced clusters



(b) Overlapping linear clusters



(c) Rotated moons

Figure 3: Probability density function of the synthetic datasets, before and after applying the biasing scheme. The points with coordinates $(\Delta_{x_1}, \Delta_{x_2})$ used in the biasing scheme are marked in yellow for class 1 (blue) and in green for class 2 (red).

3.3 Test Case 3: High-Dimensional Data

The last test case is concerned with whether an increase in the size of the feature space leads to a decrease in adaptation performance. We generate random binary classification balanced datasets[1], having 70% of the features informative and 30% redundant. It is important to ensure a sufficient amount of train data for each of the different feature sizes in our experiment. There is no consensus on how to determine the adequate number of samples s based on the number of features f [20], reason why we generate datasets for multiple sample-feature functions. The first function we use is Events Per Variable (EPV) because of its widespread adoption [12, 20], for which we try two variants: EPV=50 (i.e. $s = f * 50$) and EPV=100 (i.e. $s = f * 100$). We also use a quadratic function $s = f^2 * 5$ because EPV was shown to underestimate sometimes the number of samples [12].

Three different biasing schemes for the train set are used in this experiment: (a) on the most important feature only; (b) on all features while maintaining class balance; (c) on all features without maintained class balance. For scheme (a) we identify the most important feature via the impurity decrease in a Random Forest classifier [14]. This approach for feature selection was shown to generally produce sensible results [13] and outperform alternatives [1]. For schemes (b) and (c) we wish to reduce the arbitrarily large feature space to a single feature, similarly to (a), on which we can then apply the sampling bias. To achieve this we perform PCA on the train data and select the projection on the first principal component as our feature [9]. We compute the minimum (m) and average ($\bar{m}$) values of the selected feature in all three cases, based on which we apply a sampling scheme in the form of a normal distribution with mean $m + (\bar{m} - m)/3$ and variance $(\bar{m} - m)/4$ [9].

For this test case we average accuracy over 10 datasets for each feature dimension in part and perform five train-test splits for each of the generated datasets. We seek an alternative to visualising results in relation to the benchmarks without having to plot the accuracy curves for each dataset. Therefore, we develop a new metric called *percentual domain adaptation*, which quantifies the adaptation performance of an importance weighting method as the proportion (%) of the accuracy "gap" between the optimal and unweighted classifiers that it cancels out. Using Acc_{IW} for the accuracy of the importance-weighted classifier, $Acc_{unweighted}$ for the unweighted and Acc_{opt} for the optimal, its formula is:

$$100 * (Acc_{IW} - Acc_{unweighted})/(Acc_{opt} - Acc_{unweighted})$$

4 Experiments and Results

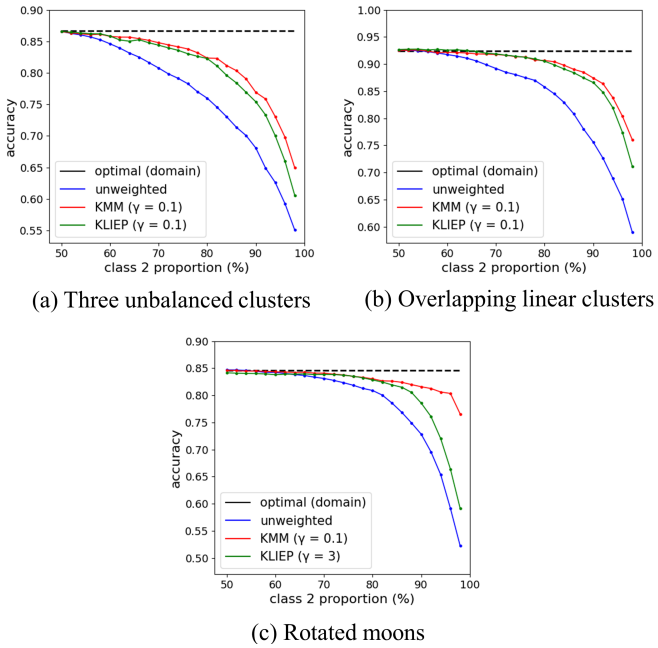
In this section we apply the evaluation framework to two importance weighting methods, KMM [9] and KLIEP [17], to compare and better understand their behaviour. We use the implementations provided by the ADAPT framework for Python [4].

The performance of both KMM and KLIEP is subject to the choice of hyper-parameters. In all our experiments we used a Gaussian kernel $K(x, y) = exp(-\gamma||x - y||^2)$. Multiple γ values have been tried for each experiment and results are presented for the best performing one; the chosen γ is specified for each result. In plus, for KMM we bound the maximum weight value to $B = 1000$ and set the constraint parameter to $\epsilon = (\sqrt{n_{tr}} - 1)/\sqrt{n_{tr}}$ (n_{tr} is the train set size) following the paper [9].

[1] We use function `sklearn.datasets.make_classification` from Python scikit-learn library.

4.1 Unequal Conditional Probabilities

Overall, the results indicate that the performance of the importance-weighted classifiers is always either on par with the optimum or considerably better than that of the unweighted method (Figure 4). The performance of the unweighted classifier seems to degrade quite fast, but both KMM and KLIEP remain significantly close to the optimum for most of the experiment, until the class 2 proportion reaches on average 78% (*Mann-Whitney test* at significance level 5%). Furthermore, starting from a class 2 proportion of 64% for datasets (a) and (b) and 76% for dataset (c), both KMM and KLIEP significantly outperform the unweighted classifier. Results are based on the *Wilcoxon signed-rank test* for the normally distributed data and the *corrected resampled t-test* [11] for the non-normally distributed data, at significance level 5%. The full results of the statistical analysis are available in Appendix B. The performance of KMM and KLIEP is surprising given that we violate the key assumption used by importance weighting, that of equal conditional probabilities of the source and target distributions. However, the fact that KMM and KLIEP follow a trend close in shape to the unweighted classifier is a sign that their robustness is still limited.



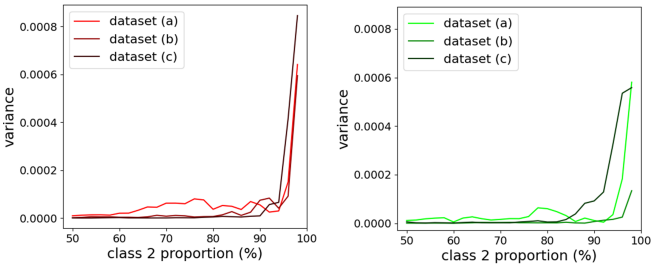
(a) Three unbalanced clusters (b) Overlapping linear clusters

(c) Rotated moons

Figure 4: Classification performance on a label-based sampling scheme.

We use the importance weights assigned to the train samples to understand what factors limit the performance of KMM and KLIEP. When class imbalance increases, due to overlapping clusters, more samples of the majority class are sampled in the vicinity of the minority class and are consequently assigned high weights (see Figures D.1 and D.2 for a visualisation). This phenomenon occurs because importance weighting does not account for class labels when aligning distributions. The train data points in the minority class are gradually outweighed by the ones in the majority class, which produces a skewed decision boundary. We analyse how two characteristics of the train set, namely the sample size and the distance between clusters, influence the overshadowing effect described above. We expect that having more data available in the minority class and less class overlap, respectively, will both improve adaptation performance.

Contrary to expectations, varying the number of samples does not generate any considerable performance difference. The accuracy score variance registered by KMM and KLIEP over five proportions $(3, 2, 1, 1/2, 1/4)$ of the original train set size does not exceed 0.00084 and 0.0006, respectively (Figure 5). The performance curves based on which the variance is computed are shown in Figure E.1. This result can be explained by the fact that sampled data in both minority and majority classes increase directly proportionally with the overall number of train samples, therefore maintaining the outweighing effect.



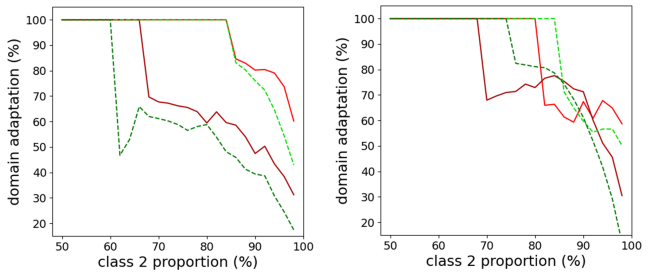
(a) KMM (b) KLIEP

Figure 5: Variance of classification accuracy on a label-based sampling scheme computed over different train set sizes.

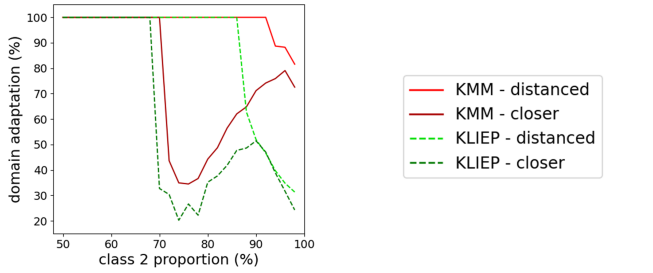
Decreasing the regions of class overlap does improve in general the adaptation performance of KMM and KLIEP (Figure 6). Fewer samples in the majority class are assigned overly large weights because they are further away from the minority class. Nevertheless, the effect is not uniform across the datasets and tends to depend on their shape; dataset (a) displays the most visible improvement and dataset (b) almost no clear-cut advantage.

Lastly, we study if the common approach of assigning class weights to correct class imbalance solves the overshadowing problem faced by importance weighting as well. We assign weights to the two classes inversely proportional to their frequency in the train set[2]. The results show that both the weighted and unweighted classifiers improve up to the point where they perform on par with the optimum irrespective of the bias intensity (Figure 7). This suggests that class weights not only successfully correct the overshadowing problem, but they might be a much better alternative to importance weighting for class imbalance correction altogether.

[2]We set parameter `class-weight='balanced'` of class `sklearn.linear_model.LogisticRegression`.

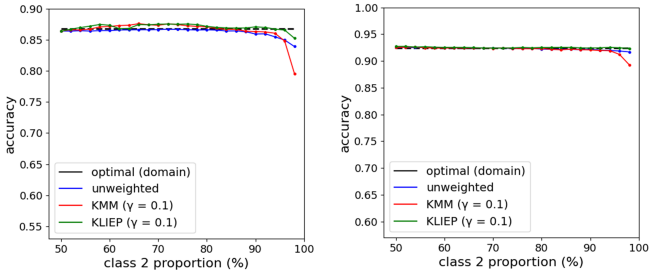


(a) Three unbalanced clusters (b) Overlapping linear clusters

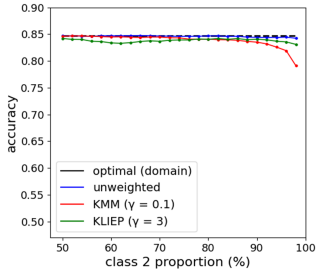


(c) Rotated moons

Figure 6: Domain adaptation performance for more versus less overlapping classes.



(a) Three unbalanced clusters (b) Overlapping linear clusters

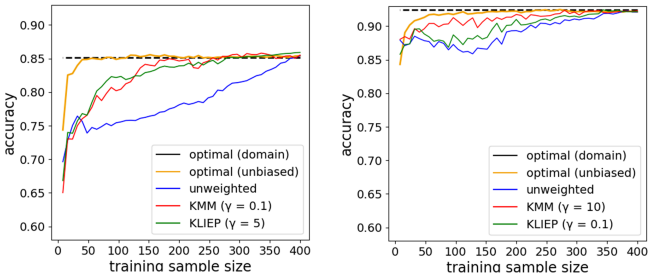


(c) Rotated moons

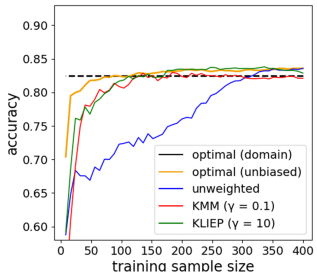
Figure 7: Classification performance on a label-based sampling scheme when the classes are weighted.

4.2 Varying Train Sample Sizes

As expected, having fewer train points negatively influences the adaptation performance on all three datasets (Figure 8). In spite of this, both KMM and KLIEP significantly outperform the unweighted classifier throughout much of the experiment. Results are based on the *Wilcoxon signed-rank test* for the normally distributed data and the *corrected resampled t-test* [11] for the non-normality distributed data, at significance level 5%. Astonishingly, KMM and KLIEP even significantly match the accuracy of the optimal (domain) classifier for a considerable reduction in the number of train samples (*Mann-Whitney test* at significance level 5%). The full results of the statistical analysis are available in Appendix C. We note however that the accuracy score of both adaptation methods drops steeply and can even underperform that of the unweighted classifier for extremely low sample sizes (under 50). This is unsurprising because the optimal (unbiased) classifier registers a similar behaviour, sign that the number of train samples is simply insufficient for any appropriate prediction.



(a) Three unbalanced clusters (b) Overlapping linear clusters



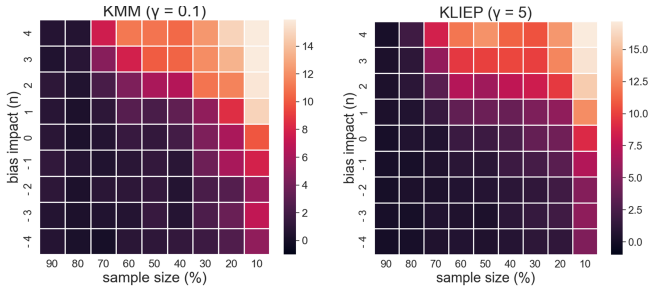
(c) Rotated moons

Figure 8: Classification performance on a feature-based sampling scheme for varying train sample sizes.

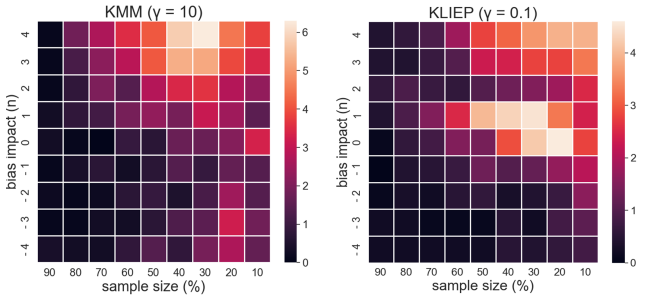
The shape of the performance curves for KMM and KLIEP (Figure 8) shows a non-linear effect of sample size on their accuracy, which indicates that other contributing factors might exist. Instead of a proportional decrease, domain-adapted accuracy registers roughly three phases: initially remains close to the optimum, then slowly declines and finally sharply decreases. We hypothesise that the observed effect is the combined result of both a reduced sample size and the applied sampling bias because the "sizes" of the three phases differ among the datasets, which also use different biasing schemes. To verify our assumption, we repeat the experiment from test case 2 for various values of the intensity factor b and compute in each case the difference in accuracy between the optimal (domain) classifier and the importance-weighted one (Figure 9).

It is easy to observe that both sample size and bias intensity influence considerably the depreciation in per-

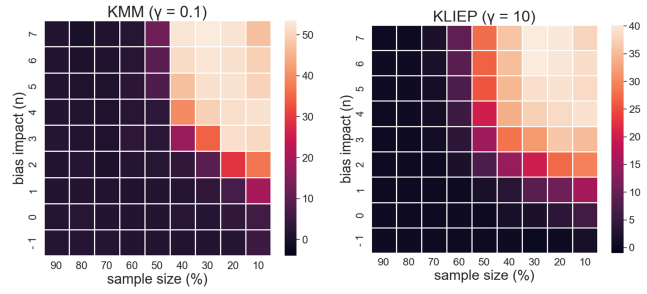
formance of KMM and KLIEP because the heat maps exhibit a diagonal-shaped pattern for all three datasets (Figure 9). We quantify the contribution of each of the two factors by computing the normalised marginal standard deviations of the accuracy differences in each heat map (Table 1). The values tend to vary depending on the dataset, from a minimum of 47.5 to a maximum of 60.5 on the *sample size* column. This indicates that the proportion in which the small sample size and the bias intensity each affect the importance weighting method is not fixed, but rather an inherent attribute of the dataset and the sampling scheme applied to it.



(a) Three unbalanced clusters



(b) Overlapping linear clusters



(c) Rotated moons

Figure 9: Accuracy difference between the optimal (domain) and importance-weighted classifiers as a function of the bias impact factor n (used for varying the values of b) and the proportion (%) of the original train set size (400 samples). The original values of b used in the sampling scheme are varied as follows: for set (a) $b * 1.5^n$ for class 1 and $b * 1.3^n$ for class 2; for (b) $b * 2.3^n$ for both classes; for (c) $b * 2^n$ for both classes. Lighter colors in the heat map indicate higher differences, so poorer adaptation performance.

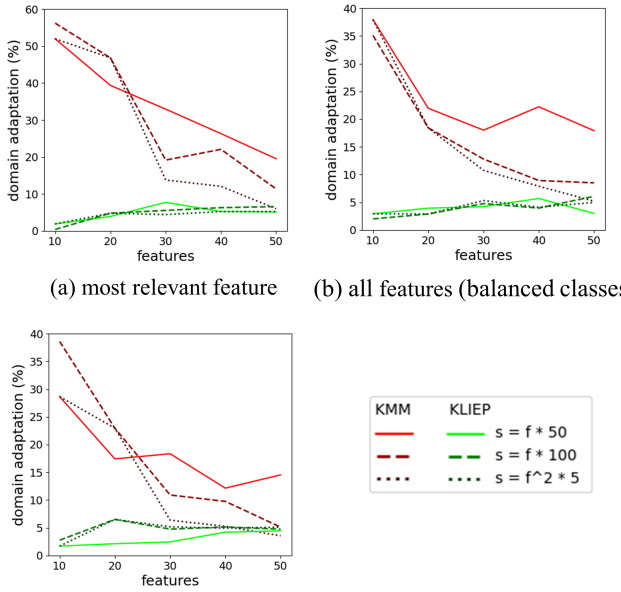
Table 1: Normalised marginal standard deviations of the scores recorded in each heat map of Figure 9.

Dataset	**KMM**		**KLIEP**	
	Sample size	Bias impact	Sample size	Bias impact
(a)	52.7	47.3	47.8	52.2
(b)	48.4	51.6	47.5	52.5
(c)	60.5	39.5	57.1	42.9

4.3 High-Dimensional Data

A general remark on Figure 10 is that the results follow a similar trend for all three sample-feature functions. This indicates that the scores obtained are reliable and do not suffer from a lack of sufficient train data.

The adaptation performance of KMM is greatly affected by high-dimensional data irrespective of the type of bias applied (Figure 10). The average score differences of KMM between the minimum (10) and the maximum (50) number of features are 41.1 (a), 26.5 (b), 24.3 (c). The values for the two biasing schemes using all features are smaller than that for the scheme using only one feature, which shows that the impact of high-dimensionality is not necessarily correlated with the "strength" of the biasing scheme. Moreover, the closeness of the values in cases (b) and (c) indicates that a bias including labels does not result in a more impactful curse of dimensionality than a bias using solely the feature vector. Unlike KMM, the scores for KLIEP do not indicate fluctuations under high-dimensional datasets. However, the adaptation performance for KLIEP is in general poor throughout the experiment, ranging between 0.39 and 7.74.



(a) most relevant feature

(b) all features (balanced classes)

(c) all features (imbalanced classes)

Figure 10: Domain adaptation performance for different feature dimensions. The sampling scheme involves either (a) the most important feature only or (b)(c) all features.

5 Discussion

A key liability of importance weighting is that it does not universally, or even uniformly, improve classification performance across datasets or types of sampling bias. As [10, p. 14] also highlights, "in domain adaptation there is no free lunch". Our experiments indicate that the performance of KMM and KLIEP is largely context-dependant. For example, adaptation to unequal conditional probabilities is largely influenced by the degree of class overlap, while robustness to small train sample sizes has as confounding factor the intensity of the sampling bias. Our observation that importance weighting can indeed fail in certain particular scenarios has been reiterated in multiple prior studies as well [2, 3, 16]. Unfortunately, as [10] also notices, verifying assumptions about the dataset is not a trivial task. We purposefully use datasets with only two features to be able to visualise them and identify performance patterns, but real-life scenarios are usually much more complex.

It is commonly agreed in literature that assigning weights and learning based on them are two independent processes, reason why performance bounds of importance weighting are often studied independently of any learning algorithm [2,3,16]. Consequently, we expect that repeating our experiments with other linear classifiers will produce similar results. In fact, most empirical studies employ only one classifier like we did, typically either SVM [9] or regression [16]. As [5, 23] shows, both these learners are affected by sample selection bias and can produce meaningful experimental results.

5.1 Unequal Conditional Probabilities

Our results solidify previous claims in [9] that KMM improves classification accuracy for unequal conditional probabilities and extends them to cover KLIEP as well. While [9] used a single dataset and one class imbalance ratio, we show the trend maintains for three datasets with different patterns and over the entire span of label-based sampling schemes. We also performed a statistical analysis on our results to prove their significance, given that scores shown in [9] differ by a low margin (≤ 0.03) and no information is available on their variance. However, an important remark is that our study is empirical in nature, meaning our conclusion can not be generalised to other methods, datasets or sampling schemes.

It is hard to determine which factors influence the performance of importance weighting when the original class proportions in the train set are changed because all theoretical bounds proposed in literature assume the constraint of equal conditional probabilities. For example, our results show that sample size does not play a role anymore, contrary to what Theorem 3 from [2] suggests. In spite of this, we tried to explain our results by the degree of class overlap in the datasets and obtained performance improvements when we distanced the clusters. We believe that measuring the overlap between the probability densities of the two classes would be even more insightful over simply distancing clusters. However, we resorted to this approach over more fine-grained alternatives because computing them over a multi-dimensional feature space, even two-dimensional, is non-trivial [6].

5.2 Varying Train Sample Sizes

Our results mirror to a large extent prior empirical findings and fall in line with the theoretical bound introduced in [2](Theorem 3) and further interpreted in [10] on the role of sample size in importance weighting adaptation. It was also shown in [17] that both KMM and KLIEP degrade when the train set size decreases; however, they tried a range of values much smaller than ours ($[50, 150]$ vs $[8, 400]$ samples) and did not provide a benchmark for comparison. In addition, we also performed a statistical analysis on our results to prove their significance. Interestingly, while Figure 2(b) in [17] shows that a fine-tuned KMM should visibly outperform KLIEP, they perform mostly similarly in our experiments. This can be attributed to the difference in datasets used in the two studies. Lastly, the inability of the adaptation methods to improve accuracy anymore for very small train set sizes (≤ 50 in our experiments) was also noticed in [9], who tested KMM on multiple real-life datasets.

The particular shape of our performance curves, concave down, is explained by the value of the Rényi divergence between the source and target domain distributions [2]. Our sampling bias is sufficiently "mild" such that the divergence results in a fast convergence of the accuracy error for an increase in sample size [2]. By increasing bias intensity we heighten divergence, which makes convergence to the optimum slow. This explains the existence of bias intensity as confounding factor.

We also aimed to empirically quantify the individual impact of bias intensity and sample size by computing the marginal standard deviations in the heat maps. However, our approach is limited in that results can be easily influenced by the chosen step size. A theoretical bound based on the Rényi divergence and the sample size was proposed in [2], but it contains a probabilistic factor that makes precise quantification hard unfortunately.

5.3 High-Dimensional Data

KMM and KLIEP show very different performance trends, which is surprising given that both of them infer weights in a similar way, through optimisation. Our results disagree with previous claims that importance weighting as a whole suffers from the curse of dimensionality [10] because KLIEP maintained a stable performance in our experiment. The same was found to be true by [17] under the condition that the hyper-parameters are tuned correctly.

Our results concerning KMM are particularly different from those of prior studies. Whereas in [17] KMM tends to improve performance as the number of features increases, in our case it degrades. This disagreement could be due to [17] using a constant sample size across all feature vector dimensions and not providing a benchmark. This last aspect makes comparing results partic-

ularly hard, since we quantify adaptation in relation to the unweighted and optimal classifiers. The difference in datasets and sampling schemes between the two studies can also impact the results. After all, dataset configuration and distribution divergence played a major role for the two-dimensional experiments, so we expect the same to hold true in the case of high-dimensional data.

6 Responsible Research

A key ethical aspect of machine learning experiments is the provenance of data. While most research concerned with combating sample selection bias uses at least partially, if not fully, real-world data [3, 9, 16, 23], we base our experiments exclusively on synthetic datasets. Not only does our approach allow for better control over the biasing schemes, but completely eliminates the risks associated with, for example, consent, copyright, re-identification, and data storage and manipulation.

The fact that "importance weighting is most often used in applications involving clinical or social science" [10, p. 5], both highly critical fields, emphasises the ethical implications of this research and its practical use. Our results in the case of class imbalance (test case 1) demonstrate that importance weighting techniques have the potential to alleviate the effects of underrepresentation of certain population categories in decision-making situations. This makes us hopeful that our research can be further used for alleviating at least certain types of social bias in real-world data science applications.

The source code for generating the datasets and running the experiments presented in this paper is made publicly available on GitLab [18] in order to ensure full reproducibility of the research. We also include Appendix A containing all parameter settings for recreating the identical datasets and biasing schemes to the ones described in our test cases. To ensure that experimental results are not influenced by confounding factors, all classification scores were averaged over a considerable number (30) of experimental runs, in the form of train-test splits. Lastly, all pseudo-random number generators used in the codebase of the experimental setup employ seeds, reason why we expect that future runs of our experiments will generate identical results.

7 Conclusions and Future Work

This paper aimed to evaluate the effectiveness of importance weighting techniques in mitigating sample selection bias. More specifically, we studied the lesser-known scenario when the domain adaptation uses unlabeled samples from the underlying domain of the data, instead of a particular test set. The proposed evaluation framework contains three scenarios, traditionally known for posing difficulties to importance weighting: (1) unequal conditional probabilities of the source and target distributions, (2) small train sample size, and (3) high-dimensional data. We applied the framework to two popular techniques, KMM and KLIEP, to analyse their performance. We substantiate our results with a full statistical significance analysis and augment them with extra experiments for a better understanding.

Results matched to a large extent prior findings on adaptation performance, but produced some novel insights too. Overall, importance weighting proved to be no one-size-fits-all solution, its success being largely dependent on the characteristics of the dataset and the sampling bias. We showed that theoretical performance bounds, particularly involving the train set size, fail to hold anymore when the fundamental assumption of equal conditional probabilities is violated. Results for test case 1 also showed that KMM and KLIEP can still, surprisingly, significantly improve classification in this scenario. We hypothesized their behaviour is influenced by the degree of overlap between the classes. In test case 2, a decreasing train sample size showed to negatively influence the adaptation ability. Even though KMM and KLIEP remained largely performant, for an extremely small size this ceased to be the case anymore. Furthermore, we observed that success was grossly subject to the "intensity" of the sampling bias. Lastly, we showed that high-dimensional data (test case 3) does not necessarily negatively affect all importance weighting techniques as previously thought. Surprisingly, techniques from the same class of importance weighting algorithms displayed very different behaviours in this scenario.

Our work represented a first step in using importance weighting with unlabeled data that is not sourced from a particular test set. However, the underlying domain and the test set still follow the same distribution in our research. Future efforts can focus on using the evaluation framework when the test set is subject to sampling bias as well, which would offer better insight into the ability to generalise simultaneously to different test sets. Furthermore, researchers can also analyse our hypothesis on the impact of class overlap on the adaptation performance when the conditional probabilities are unequal. This would be especially relevant considering how easily this key assumption made by importance weighting can be violated in practice. For the case of high-dimensional data, future efforts can continue exploring its impact on more importance weighting techniques with multiple biasing schemes, since our empirical results proved to mismatch expectations set in literature.

Acknowledgements

This thesis would not have been possible without the contribution of my two supervisors, whose interest, patience and feedback made this project enjoyable and elevated the quality of my work.

I would like to express my deepest gratitude to my mother (Dr. Ing. Carmen Tociu), uncle (Romanian Academy Professor Dr. Ing. Gheorghe Maria), and aunt (Dr. Ing. Cristina Maria), whose multifaceted support was invaluable throughout my university years. They continue to motivate and inspire me at every step of my personal and academic journey.

References

[1] R. Chen, C. Dewi, S. Huang, and R. E. Caraka. Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, 7(1):52–78, 2020.

[2] C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[3] C. Cortes and M. Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

[4] A. de Mathelin, F. Deheeger, G. Richard, M. Mougeot, and N. Vayatis. Adapt: Awesome domain adaptation python toolbox. *arXiv preprint arXiv:2107.03049*, 2021.

[5] W. Fan, I. Davidson, B. Zadrozny, and P. S. Yu. An improved categorization of classifier's sensitivity on sample selection bias. *Data Mining, IEEE International Conference on Data Mining (ICDM'05)*, 0:605–608, 12 2005.

[6] E. Gutiérrez-Peña and S. G. Walker. An efficient method to determine the degree of overlap of two multivariate distributions. In *Selected Contributions on Statistics and Data Science in Latin America*, pages 59–68. Springer International Publishing, 2019.

[7] J. Heckman. Varieties of selection bias. *American Economic Review*, 80(2):313–318, 1990.

[8] M. A. Hernan and J. M. Robins. Estimating causal effects from epidemiological data. *Journal of Epidemiology Community Health*, 60(7):578–586, 2006.

[9] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

[10] W. M. Kouw and M. Loog. A review of domain adaptation without target labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):766–785, 3 2021.

[11] C. Nadeau and Y. Bengio. Inference for the generalization error. *Machine Learning*, 52:239–281, 2003.

[12] R. Riley, J. Ensor, K. Snell, F. Harrell, G. Martin, J. Reitsma, K. Moons, G. Collins, and M. van Smeden. Calculating the sample size required for developing a clinical prediction model. *BMJ*, 368, 2020.

[13] M. Saarela and S. Jauhiainen. Comparison of feature importance measures as explanations for classification models. *Applied Sciences*, 3(2):272–284, 2021.

[14] scikit learn. Feature importances with a forest of trees. https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html, 2023.

[15] K. Stapor. Evaluating and comparing classifiers: Review, some recommendations and limitations. pages 12–21, 5 2017.

[16] M. Sugiyama and K. R. Müller. Model selection under covariate shift. In W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, editors, *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, pages 235–240. Springer Berlin Heidelberg, 2005.

[17] M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[18] A. C. Tociu. Framework for evaluating the effectiveness of importance weighting to sample selection bias (v1.0) [source code]. https://gitlab.ewi.tudelft.nl/goncalveslab/bachelor-projects/bsc-rp-2223-andrei-tociu.git, 2023.

[19] ADAPT: Awesome Domain Adaptation Python Toolbox. Correcting sample bias with transfer learning: Diabetes dataset. https://adapt-python.github.io/adapt/examples/Sample_bias_example.html#Sample-Bias-on-the-diabetes-dataset, 2020.

[20] M. van Smeden, K. G. M. Moons, J. A. H. de Groot, G. S. Collins, D. G. Altman, M. J. C. Eijkemans, and J. B. Reitsma. Sample size for binary logistic prediction models: Beyond events per variable criteria. *Statistical Methods in Medical Research*, 28(8):2455–2474, 2019.

[21] J. Wen, C. Yu, and R. Greiner. Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 631–639. PMLR, 6 2014.

[22] Q. Xu and Q. Yang. A survey of transfer and multitask learning in bioinformatics. *Journal of Computing Science and Engineering*, 5(3):257–268, 2011.

[23] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine Learning*. Association for Computing Machinery, 2004.

A Formal Description of the Synthetic Datasets

This appendix contains the parameter settings used for constructing the synthetic datasets used in our experiments and for inducing sample selection bias in them.

Table A.1: Parameters of the synthetic datasets used in the experiments. Datasets (a) and (b) use multivariate normally-distributed clusters based on means and covariance matrices. Dataset (c) is generated using Python method `sklearn.datasets.make_moons`[3] based on the noise parameter.

Dataset	**Test case**	**Class 1**			**Class 2**		
		Samples	Means	Cov matrices / Noise	Samples	Means	Cov. matrices / Noise
(a)	1	1100	(-7, 6)	$\begin{pmatrix} 4.5 & 0 \\ 0 & 4.5 \end{pmatrix}$	1500	(0, 0)	$\begin{pmatrix} 15 & 0 \\ 0 & 15 \end{pmatrix}$
		400	(7, 6)	$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$			
	2	350			500		
		150					
(b)	1	1500	(-4, -4)	$\begin{pmatrix} 8 & 7 \\ 7 & 8 \end{pmatrix}$	1500	(-20, 2)	$\begin{pmatrix} 80 & -60 \\ -60 & 80 \end{pmatrix}$
	2	500			500		
(c)	1	1500	(0, 1)	(0.05, 0.35)	1500	(1, -0.5)	(0.05, 0.35)
	2	500			500		

Table A.2: Parameters of the sample selection biasing function $f(x = (x_1, x_2)) = e^{-b(|x_1 - \Delta_{x_1}| + |x_2 - \Delta_{x_2}|)}$ applied during test case 2 on the datasets.

Dataset	**Class 1**			**Class 2**		
	Δ_{x_1}	Δ_{x_2}	b	Δ_{x_1}	Δ_{x_2}	b
(a)	-7	6	0.1	5	5	5
(b)	0	0	0.5	-25	0	0.1
(c)	1	0.5	1.5	0	0	1.5

[3] Method is available at https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html .

B Statistical Analysis of Results of Test Case 1

This appendix contains a statistical analysis of the results obtained during test case 1 to determine if the importance-weighted classifiers significantly outperform the unweighted one.

Firstly, we perform a paired samples analysis on the mean accuracy differences between the weighted and unweighted classifiers (i.e. $d_{KMM} = Acc_{KMM} - Acc_{unweighted}$, $d_{KLIEP} = Acc_{KLIEP} - Acc_{unweighted}$) recorded in the 30 train-test splits for each of the 25 class imbalance ratios (50-50% - 2-98%). The null hypothesis is $H_0 : d_{KMM}, d_{KLIEP} \leq 0$ and the alternative hypothesis is $H_1 : d_{KMM}, d_{KLIEP} > 0$. We begin by using a box plot to identify and eliminate potential outliers. Subsequently, we perform the *Shapiro-Wilk test* at significance level 5% and conclude that the majority of data is normally distributed (Table B.1). On the normally distributed data we perform a one-tailed *corrected resampled t-test* [11,15] and on the non-normally distributed data we use the one-tailed *Wilcoxon signed-rank test*, both at significance level 5%. The results of both tests are shown jointly in Table B.2.

Secondly, we perform an independent samples analysis to compare the mean accuracies of the weighted classifiers and the optimal one trained on the underlying domain, again for each of the 25 class imbalance ratios. This covers the edge case when the weighted classifier is in the vicinity of the unweighted one, so it does not significantly outperform it, yet it is on par with the optimum, in which case we consider the domain adaptation successful as well. Analysing data with the *Shapiro-Wilk test* at significance level 5% yields a large number of non-normal distributions, reason why we decide to perform the non-parametric *Mann-Whitney test* on all data points. The u statistics and p values are shown in Table B.3.

Table B.1: p values for the *Shapiro-Wilk test*. Significant values ($p \leq 0.05$) are marked in red.

	Dataset (a)		Dataset (b)		Dataset (c)	
	KMM	KLIEP	KMM	KLIEP	KMM	KLIEP
1	.516	.847	.072	.456	.328	.056
2	.4	.215	.05	.064	.144	.328
3	.81	.562	.166	.056	.019	.24
4	.484	.085	.279	.209	.018	.082
5	.466	.621	.183	.032	.103	.357
6	.853	.16	.383	.218	.405	.49
7	.132	.144	.279	.42	.34	.421
8	.196	.924	.142	.375	.084	.825
9	.312	.779	.12	.582	.261	.359
10	.711	.063	.069	.94	.026	.529
11	.479	.781	.895	.515	.202	.255
12	.079	.954	.457	.435	.074	.431
13	.388	.94	.027	.069	.179	.062
14	.807	.764	.704	.056	.055	.917
15	.798	.195	.185	.26	.556	.691
16	.883	.667	.189	.607	.7	.534
17	.245	.847	.281	.275	.777	.561
18	.978	.824	.074	.112	.425	.499
19	.37	.114	.252	.335	.9	.775
20	.683	.068	.919	.895	.344	.182
21	.114	.074	.467	.99	.294	.857
22	.631	.276	.229	.889	.587	.462
23	.392	.929	.449	.838	.052	.07
24	.588	.783	.565	.971	.311	.116
25	.858	.531	.447	.591	.622	.165

Table B.2: Degrees of freedom (df), t statistic and p value for the paired samples tests on the mean accuracy differences between the weighted and unweighted classifiers. Results for the non-normally distributed data (*Wilcoxon signed-rank test*) are marked in red, while the rest represent the *corrected resampled t-test*. Significant values ($p \leq 0.05$) are highlighted grey.

	Dataset (a)						Dataset (b)						Dataset (c)					
	KMM			KLIEP			KMM			KLIEP			KMM			KLIEP		
	df	t	p	df	t	p	df	t	p	df	t	p	df	t	p	df	t	p
1	27	0.0	.5	29	0.0	.5	27	0.144	.443	29	0.27	.394	29	-0.133	.553	27	-0.921	.817
2	27	0.167	.434	29	0.223	.413	28	171.0	.157	29	0.547	.294	29	-0.277	.608	27	-1.107	.861
3	28	0.313	.378	29	0.37	.357	27	0.524	.302	29	0.634	.265	27	84.0	.951	29	-0.723	.762
4	25	0.829	.208	29	0.578	.284	29	0.124	.451	29	0.523	.302	24	86.5	.016	29	-0.532	.701
5	29	1.095	.141	29	1.428	.082	27	0.098	.461	29	320.5	.001	28	0.575	.285	28	-0.205	.58
6	28	1.261	.109	28	1.619	.058	28	0.446	.329	29	1.395	.087	28	0.07	.472	29	-0.458	.675
7	29	2.019	.026	29	1.094	.142	25	0.925	.182	28	1.819	.04	28	0.44	.332	29	-0.147	.558
8	29	2.334	.013	29	1.741	.046	28	1.026	.157	28	2.43	.011	29	0.569	.287	29	0.081	.468
9	29	2.544	.008	29	2.519	.009	25	2.216	.018	29	2.484	.01	27	0.963	.172	26	0.599	.277
10	29	2.372	.012	29	2.652	.006	28	2.261	.016	28	2.77	.005	28	229.0	.001	28	0.407	.344
11	26	2.894	.004	29	2.963	.003	28	2.404	.012	28	3.296	.001	29	1.472	.076	29	0.881	.193
12	29	2.345	.013	29	3.115	.002	29	2.546	.008	28	3.956	.001	29	1.65	.055	29	1.202	.12
13	28	2.87	.004	29	3.415	.001	29	465.0	.001	28	4.153	.001	29	1.8	.041	29	1.467	.077
14	29	3.189	.002	29	3.678	.001	29	3.46	.001	29	3.952	.001	29	1.833	.039	29	1.617	.058
15	27	3.525	.001	27	5.234	.001	29	3.254	.001	29	4.59	.001	28	2.243	.016	29	1.931	.032
16	28	3.595	.001	29	4.077	.001	28	4.39	.001	28	4.799	.001	28	2.268	.016	28	1.78	.043
17	28	4.015	.001	29	3.621	.001	29	4.144	.001	29	3.726	.001	28	2.25	.016	28	2.066	.024
18	29	2.951	.003	29	3.321	.001	27	5.403	.001	29	4.548	.001	29	2.453	.01	29	2.259	.016
19	29	2.932	.003	29	3.22	.002	29	5.202	.001	29	4.72	.001	28	3.261	.001	28	3.664	.001
20	29	2.651	.006	29	2.927	.003	28	4.864	.001	29	5.518	.001	29	3.477	.001	29	3.621	.001
21	29	2.737	.005	29	2.911	.003	28	5.467	.001	29	7.295	.001	29	3.361	.001	28	3.549	.001
22	29	2.685	.006	29	3.789	.001	29	5.778	.001	29	7.681	.001	29	4.168	.001	29	3.852	.001
23	28	4.771	.001	27	5.273	.001	29	4.774	.001	29	7.812	.001	29	4.821	.001	29	4.686	.001
24	29	2.966	.003	28	3.391	.001	29	4.469	.001	28	6.906	.001	29	5.178	.001	29	5.239	.001
25	29	2.363	.013	28	3.894	.001	29	4.957	.001	29	6.265	.001	29	5.568	.001	29	2.783	.005

Table B.3: u statistic and p value for the independent samples tests on the mean accuracies of the weighted and optimal (domain) classifiers (*Mann-Whitney test*). Significant values ($p \leq 0.05$) are highlighted grey.

	Dataset (a)				Dataset (b)				Dataset (c)			
	KMM		KLIEP		KMM		KLIEP		KMM		KLIEP	
	u	p	u	p	u	p	u	p	u	p	u	p
1	900.0	.001	900.0	.001	900.0	.001	900.0	.001	827.0	.001	900.0	.001
2	900.0	.001	900.0	.001	900.0	.001	900.0	.001	788.5	.001	900.0	.001
3	896.0	.001	900.0	.001	898.5	.001	900.0	.001	811.0	.001	900.0	.001
4	882.5	.001	900.0	.001	871.0	.001	900.0	.001	803.5	.001	899.5	.001
5	880.5	.001	900.0	.001	846.0	.001	894.0	.001	779.5	.001	886.5	.001
6	855.5	.001	897.0	.001	822.5	.001	886.0	.001	756.0	.001	829.5	.001
7	847.5	.001	889.0	.001	794.0	.001	861.5	.001	731.5	.001	789.0	.001
8	832.0	.001	877.5	.001	770.5	.001	838.0	.001	722.0	.001	755.5	.001
9	784.0	.001	849.0	.001	704.0	.001	783.0	.001	714.5	.001	723.5	.001
10	771.0	.001	811.5	.001	698.0	.001	734.0	.001	662.0	.002	694.0	.001
11	724.5	.001	805.0	.001	680.5	.001	680.0	.001	638.5	.005	637.5	.006
12	714.5	.001	787.5	.001	647.0	.004	637.0	.006	618.5	.013	609.5	.018
13	680.5	.001	761.0	.001	627.5	.009	622.0	.011	580.0	.055	589.0	.04
14	662.5	.002	731.5	.001	585.5	.045	590.5	.038	550.5	.138	578.0	.059
15	614.0	.015	706.5	.001	578.5	.057	561.0	.101	536.0	.205	571.5	.073
16	592.0	.036	660.0	.002	549.5	.142	547.0	.152	532.0	.227	566.5	.086
17	561.0	.101	614.0	.015	542.0	.174	481.5	.645	515.5	.335	538.0	.195
18	538.5	.192	625.5	.01	502.0	.444	432.5	.8	523.5	.279	541.5	.178
19	548.0	.148	587.5	.043	495.0	.509	410.5	.562	515.5	.335	556.0	.118
20	539.5	.188	535.5	.208	493.5	.523	421.0	.671	505.0	.419	564.0	.093
21	499.0	.472	511.5	.366	510.0	.378	389.5	.373	487.5	.583	536.5	.202
22	517.0	.325	511.0	.37	454.0	.959	409.5	.552	470.0	.772	539.5	.187
23	507.5	.398	495.0	.509	425.0	.716	381.0	.309	482.0	.641	552.5	.13
24	498.5	.477	478.5	.678	426.5	.733	399.0	.453	457.0	.923	539.5	.187
25	471.5	.756	453.5	.964	432.5	.801	406.5	.523	447.5	.976	532.0	.227

C Statistical Analysis of Results of Test Case 2

This appendix contains a statistical analysis of the results obtained during test case 2 to determine if the importance-weighted classifiers significantly outperform the unweighted one.

We perform two analyses identical to those outlined in Appendix B. The procedure is repeated this time for each of the 50 training sample size ratios (100% - 2%). Table C.1 shows the results of the *Shapiro-Wilk test* on the normal distribution of data. The joint results of the *corrected resampled t-test* and the *Wilcoxon signed-rank test* for the paired samples analysis are available in Table C.2. The results of the *Mann-Whitney test* for the second analysis, to determine proximity to the optimal (domain) classifier, are available in Table C.3.

Table C.1: p values for the *Shapiro-Wilk test*. Significant values ($p \leq 0.05$) are marked in red.

	Dataset (a)		Dataset (b)		Dataset (c)	
	KMM	KLIEP	KMM	KLIEP	KMM	KLIEP
1	.025	.001	.001	1.0	.004	.001
2	.073	.037	.013	.001	.002	.005
3	.216	.161	.098	.001	.006	.079
4	.288	.127	.183	1.0	.085	.015
5	.058	.079	.165	.001	.004	.03
6	.835	.311	.202	.001	.012	.019
7	.268	.691	.637	.001	.029	.08
8	.058	.167	.208	.002	.002	.046
9	.287	.475	.256	.117	.071	.597
10	.291	.055	.697	.005	.662	.437
11	.229	.191	.099	.001	.577	.254
12	.088	.407	.033	.117	.715	.719
13	.049	.297	.312	.064	.932	.752
14	.379	.883	.404	.003	.106	.109
15	.416	.612	.384	.003	.048	.291
16	.48	.19	.957	.079	.381	.097
17	.009	.144	.284	.154	.308	.299
18	.176	.14	.316	.116	.785	.522
19	.646	.482	.586	.244	.589	.171
20	.054	.411	.037	.208	.786	.375
21	.191	.055	.453	.105	.338	.252
22	.235	.372	.49	.469	.586	.83
23	.261	.528	.354	.102	.318	.189
24	.054	.283	.19	.195	.346	.107
25	.102	.242	.181	.063	.06	.802
26	.334	.115	.09	.349	.717	.386
27	.18	.132	.273	.047	.036	.172
28	.162	.143	.168	.03	.257	.605
29	.204	.082	.574	.505	.973	.217
30	.246	.654	.151	.014	.667	.47
31	.18	.606	.111	.008	.292	.797
32	.368	.405	.869	.042	.011	.009
33	.934	.101	.248	.026	.092	.019
34	.138	.118	.525	.001	.368	.613
35	.553	.054	.25	.022	.478	.323
36	.215	.192	.34	.017	.275	.387
37	.445	.031	.239	.019	.971	.924
38	.566	.144	.618	.041	.89	.505
39	.581	.07	.12	.006	.821	.461
40	.534	.453	.034	.12	.852	.383
41	.832	.349	.017	.202	.085	.063
42	.566	.184	.111	.056	.607	.145
43	.495	.145	.139	.363	.16	.193
44	.384	.787	.025	.502	.257	.28
45	.699	.081	.035	.02	.998	.249
46	.694	.83	.265	.383	.824	.315
47	.153	.457	.127	.122	.908	.526
48	.49	.572	.002	.026	.211	.255
49	.698	.351	.001	.192	.621	.913
50	.434	.672	1.0	.019	.574	.635

Table C.2: Degrees of freedom (df), t statistic and p value for the paired samples tests on the mean accuracy differences between the weighted and unweighted classifiers. Results for the non-normally distributed data (*Wilcoxon signed-rank test*) are marked in red, while the rest represent the *corrected resampled t-test*. Significant values ($p \leq 0.05$) are highlighted grey. Values of t and p are marked *nan* where df was too low to produce reliable results.

	Dataset (a)						Dataset (b)						Dataset (c)					
	KMM			KLIEP			KMM			KLIEP			KMM			KLIEP		
	df	t	p	df	t	p	df	t	p	df	t	p	df	t	p	df	t	p
1	28	65.5	.553	27	60.0	.044	26	65.5	.077	17	nan	nan	27	0.0	1.0	26	5.5	.999
2	28	0.165	.435	29	142.0	.006	29	148.0	.378	27	55.5	.421	24	0.0	1.0	26	37.5	.952
3	29	0.374	.356	28	0.708	.242	28	-0.267	.604	24	14.5	.957	29	20.0	.999	29	-0.29	.613
4	28	0.755	.228	26	0.863	.198	29	0.071	.472	18	nan	nan	29	-0.995	.836	29	140.0	.475
5	28	0.994	.164	26	0.92	.183	28	0.0	.5	27	57.0	.061	29	12.0	1.0	29	105.5	.492
6	29	0.951	.175	29	1.353	.093	28	0.108	.457	25	68.0	.01	29	30.0	1.0	29	155.5	.292
7	28	1.417	.084	28	1.362	.092	29	0.06	.476	28	61.0	.133	29	46.5	1.0	29	0.041	.484
8	26	1.649	.056	28	1.396	.087	29	0.156	.439	27	120.0	.003	27	39.5	1.0	29	143.5	.163
9	28	2.072	.024	28	1.749	.046	29	0.905	.187	28	0.366	.358	28	-0.899	.812	29	0.49	.314
10	28	1.952	.031	29	2.118	.021	29	0.823	.209	25	104.5	.005	27	-0.793	.783	28	0.609	.274
11	28	2.006	.027	28	1.861	.037	28	1.149	.13	25	23.0	.057	29	-0.366	.642	29	0.656	.259
12	28	2.714	.006	29	2.321	.014	28	192.5	.047	28	0.354	.363	29	-0.125	.549	29	0.666	.255
13	28	397.0	.001	28	2.2	.018	29	0.59	.28	29	0.544	.295	29	0.0	.5	29	0.772	.223
14	28	2.312	.014	28	2.09	.023	29	0.859	.199	28	126.0	.031	29	0.411	.342	29	1.117	.137
15	28	2.213	.018	29	2.083	.023	28	0.982	.167	25	107.0	.002	29	203.5	.063	29	1.072	.146
16	28	2.625	.007	29	2.213	.017	28	1.047	.152	28	0.782	.22	29	0.512	.306	29	1.34	.095
17	28	435.0	.001	29	2.097	.022	29	0.942	.177	29	1.181	.124	29	0.753	.229	28	1.264	.108
18	29	2.11	.022	29	1.967	.029	25	1.119	.137	29	0.711	.242	29	0.885	.192	29	1.399	.086
19	28	2.059	.024	29	2.09	.023	29	0.725	.237	29	0.493	.313	29	1.017	.159	29	1.711	.049
20	29	1.776	.043	29	2.205	.018	27	209.0	.001	29	0.646	.262	29	1.126	.135	29	1.882	.035
21	27	2.209	.018	29	2.145	.02	29	0.658	.258	29	0.398	.347	29	1.383	.089	29	1.905	.033
22	29	2.196	.018	29	2.255	.016	27	0.747	.231	28	0.465	.323	29	1.36	.092	29	1.829	.039
23	29	1.916	.033	26	2.415	.012	29	0.734	.234	27	0.191	.425	29	1.597	.06	26	3.482	.001
24	29	1.882	.035	29	2.15	.02	29	0.565	.288	29	0.453	.327	29	1.945	.031	28	2.511	.009
25	29	1.907	.033	27	2.15	.02	28	0.971	.17	26	0.915	.184	28	1.577	.063	27	2.617	.007
26	29	1.879	.035	28	2.369	.012	27	0.712	.241	27	0.872	.195	29	1.574	.063	28	2.161	.02
27	29	2.144	.02	26	2.505	.009	29	0.868	.196	26	182.0	.009	29	465.0	.001	29	2.517	.009
28	29	2.077	.023	29	2.538	.008	29	0.812	.212	28	198.0	.002	28	1.954	.03	28	2.611	.007
29	29	2.142	.02	29	2.103	.022	28	1.18	.124	28	0.811	.212	29	1.777	.043	28	2.527	.009
30	29	2.207	.018	27	2.849	.004	27	0.935	.179	26	249.0	.002	29	2.135	.021	29	2.519	.009
31	28	2.752	.005	28	2.405	.012	28	0.728	.236	25	113.0	.038	28	1.814	.04	29	2.571	.008
32	28	2.192	.018	29	2.249	.016	29	1.054	.15	26	186.5	.065	29	465.0	.001	29	465.0	.001
33	29	1.992	.028	29	2.092	.023	29	1.006	.161	26	220.0	.058	29	1.735	.047	29	465.0	.001
34	29	1.901	.034	29	1.612	.059	29	0.626	.268	25	113.5	.009	29	2.138	.021	29	2.575	.008
35	28	1.307	.101	29	1.444	.08	28	1.17	.126	27	195.0	.001	29	1.735	.047	29	2.417	.011
36	27	1.289	.104	29	1.543	.067	28	0.856	.2	26	213.5	.01	27	2.343	.013	27	3.025	.003
37	29	0.921	.182	29	431.0	.001	29	0.886	.191	28	172.0	.024	28	1.393	.087	28	2.221	.017
38	27	1.187	.123	29	1.547	.066	29	0.736	.234	26	158.5	.005	24	1.694	.052	29	2.717	.005
39	29	0.748	.23	29	1.341	.095	29	1.101	.14	27	211.5	.001	29	1.35	.094	29	2.768	.005
40	29	0.851	.201	28	1.545	.067	28	350.0	.001	27	-0.083	.533	27	1.538	.068	27	2.553	.008
41	29	0.639	.264	28	1.252	.11	29	289.5	.001	27	-0.09	.536	29	1.2	.12	27	2.154	.02
42	29	0.517	.304	29	1.107	.139	28	0.57	.287	27	0.12	.453	27	1.065	.148	28	2.135	.021
43	28	0.603	.276	29	0.885	.192	28	0.672	.254	29	0.285	.389	27	1.314	.1	29	1.656	.054
44	28	0.444	.33	29	0.745	.231	29	316.0	.001	28	0.097	.462	27	1.236	.114	27	1.898	.034
45	29	0.265	.396	26	0.392	.349	29	292.0	.007	27	141.5	.312	28	0.669	.255	29	1.315	.099
46	28	0.122	.452	28	0.129	.449	29	0.243	.405	26	0.078	.469	24	1.439	.082	29	1.516	.07
47	29	-0.141	.556	28	-0.133	.552	28	0.231	.41	27	0.021	.492	29	0.402	.345	29	0.977	.168
48	29	-0.16	.563	27	-0.123	.548	25	58.5	.169	25	61.0	.921	29	0.025	.49	29	0.743	.232
49	28	-0.037	.515	24	0.137	.446	29	111.0	.002	26	-0.044	.517	29	-0.122	.548	27	0.378	.354
50	25	-0.222	.587	27	-0.122	.548	19	nan	nan	28	29.0	.998	28	-0.338	.631	28	0.061	.476

Table C.3: u statistic and p value for the independent samples tests on the mean accuracies of the weighted and optimal (domain) classifiers (*Mann-Whitney test*). Significant values ($p \leq 0.05$) are highlighted grey.

	Dataset (a)				Dataset (b)				Dataset (c)			
	KMM		KLIEP		KMM		KLIEP		KMM		KLIEP	
	u	p	u	p	u	p	u	p	u	p	u	p
1	880.0	.001	845.0	.001	649.0	.003	704.0	.001	880.0	.001	840.0	.001
2	793.0	.001	793.0	.001	668.5	.001	694.0	.001	777.0	.001	739.5	.001
3	812.0	.001	822.5	.001	661.0	.002	670.5	.001	750.5	.001	650.0	.003
4	786.5	.001	810.0	.001	619.0	.012	614.0	.015	687.5	.001	676.0	.001
5	760.0	.001	781.5	.001	658.5	.002	593.0	.034	643.0	.004	673.5	.001
6	732.5	.001	799.0	.001	611.0	.017	618.5	.012	556.0	.118	650.0	.003
7	695.5	.001	773.5	.001	609.0	.019	668.0	.001	608.0	.019	627.0	.009
8	681.0	.001	723.5	.001	576.5	.061	693.0	.001	589.5	.039	607.5	.02
9	702.0	.001	706.0	.001	613.5	.015	688.5	.001	559.5	.105	574.0	.067
10	677.5	.001	676.0	.001	586.5	.043	674.0	.001	583.0	.049	549.0	.144
11	652.5	.003	642.0	.004	580.5	.053	731.0	.001	540.5	.181	496.0	.499
12	677.0	.001	640.5	.005	515.5	.333	674.0	.001	530.0	.237	446.5	.964
13	687.0	.001	640.5	.005	571.5	.072	699.0	.001	528.0	.249	448.5	.988
14	653.5	.003	652.5	.003	637.0	.005	721.0	.001	537.0	.199	437.5	.858
15	625.0	.01	632.5	.007	592.0	.035	649.0	.003	460.5	.882	450.5	1.0
16	565.0	.089	602.5	.024	566.5	.084	718.5	.001	461.5	.87	420.5	.666
17	523.0	.281	625.0	.009	645.0	.004	667.5	.001	417.0	.629	456.0	.935
18	504.0	.425	622.5	.011	631.0	.007	702.0	.001	426.5	.732	424.0	.704
19	524.5	.271	601.0	.025	552.0	.13	707.0	.001	429.0	.76	457.0	.923
20	526.5	.258	590.5	.037	603.5	.023	645.5	.004	485.0	.608	447.5	.976
21	481.5	.644	563.5	.093	547.0	.151	593.5	.034	445.5	.953	444.0	.935
22	481.0	.65	558.0	.109	583.0	.049	691.5	.001	462.5	.858	383.5	.325
23	465.5	.823	561.0	.099	537.5	.196	597.0	.029	406.0	.517	346.0	.123
24	497.0	.488	583.5	.048	552.0	.131	620.0	.012	413.5	.593	361.0	.188
25	496.0	.498	553.0	.127	517.5	.318	582.5	.05	455.5	.941	360.5	.186
26	477.5	.687	542.0	.173	501.0	.452	603.0	.023	465.5	.824	360.0	.184
27	490.5	.55	532.0	.225	562.5	.096	623.5	.01	424.5	.71	353.0	.152
28	563.5	.093	566.5	.084	498.5	.475	602.5	.024	433.0	.806	358.0	.174
29	582.0	.05	518.0	.315	535.0	.208	589.5	.039	421.0	.672	367.0	.219
30	539.5	.185	551.5	.133	548.5	.143	584.5	.046	443.5	.929	364.5	.205
31	503.5	.43	501.5	.447	545.0	.16	559.5	.105	442.0	.911	338.0	.097
32	458.5	.905	486.0	.596	541.5	.175	604.5	.022	455.5	.941	339.0	.1
33	485.5	.602	507.0	.4	511.0	.368	570.0	.076	462.0	.864	358.0	.172
34	444.5	.941	484.5	.613	517.0	.323	558.5	.108	431.5	.789	351.5	.144
35	449.0	.994	456.0	.934	468.5	.789	540.5	.18	461.0	.876	351.0	.143
36	403.0	.487	455.0	.946	510.0	.376	557.0	.113	445.5	.953	349.0	.134
37	412.0	.576	460.5	.882	485.5	.602	531.0	.231	449.5	1.0	349.5	.137
38	462.5	.858	460.5	.881	537.5	.196	563.5	.093	467.5	.801	351.5	.145
39	409.5	.551	438.5	.87	500.5	.456	541.0	.178	470.5	.766	357.5	.171
40	412.0	.575	448.5	.988	476.0	.704	524.0	.274	477.0	.694	349.0	.135
41	412.5	.581	439.5	.882	487.0	.587	519.0	.308	477.0	.694	346.0	.124
42	398.5	.446	441.0	.899	460.0	.887	511.5	.364	468.0	.795	337.0	.094
43	414.5	.601	428.0	.748	501.0	.452	453.0	.97	482.0	.64	354.0	.155
44	422.0	.682	428.5	.754	502.0	.443	475.5	.709	466.0	.818	352.0	.146
45	466.0	.817	411.0	.565	485.5	.602	468.5	.789	467.0	.806	363.5	.201
46	444.5	.941	399.5	.455	491.0	.547	460.0	.888	466.0	.818	374.5	.264
47	427.5	.743	396.0	.425	473.5	.732	478.0	.682	446.5	.964	379.5	.298
48	426.0	.726	392.0	.391	475.5	.71	481.0	.65	451.0	.994	377.5	.284
49	449.0	.994	384.5	.332	477.0	.693	485.5	.602	473.0	.738	391.0	.384
50	422.0	.682	382.0	.314	471.0	.76	479.5	.666	474.0	.727	417.0	.629

D Visualisation of Importance Weighting for a Label-Based Sampling Scheme

This appendix illustrates the weight values assigned by the importance weighting techniques to the train samples in the synthetic datasets when a label-based bias of different intensities is used. Results are shown for KMM and KLIEP.

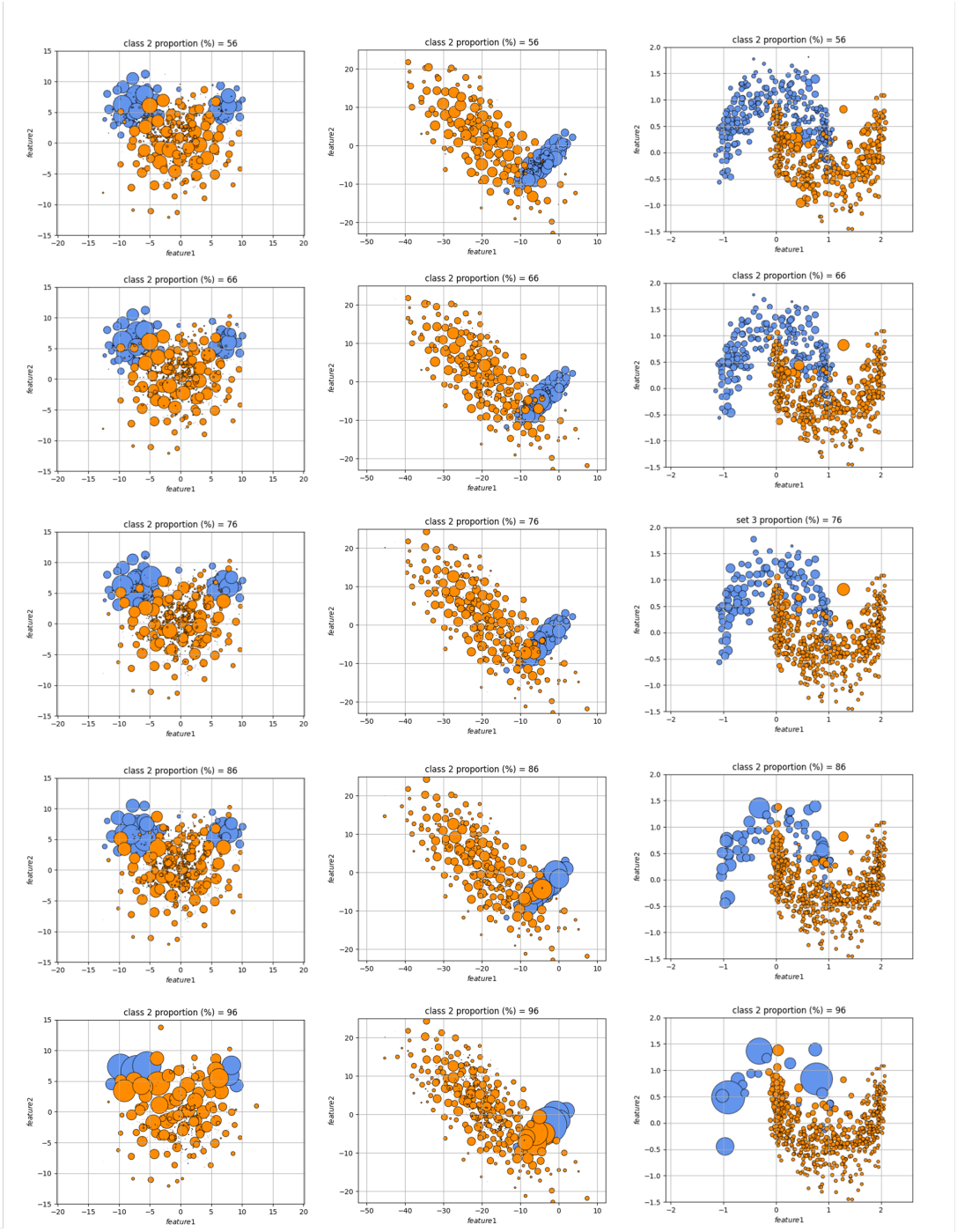
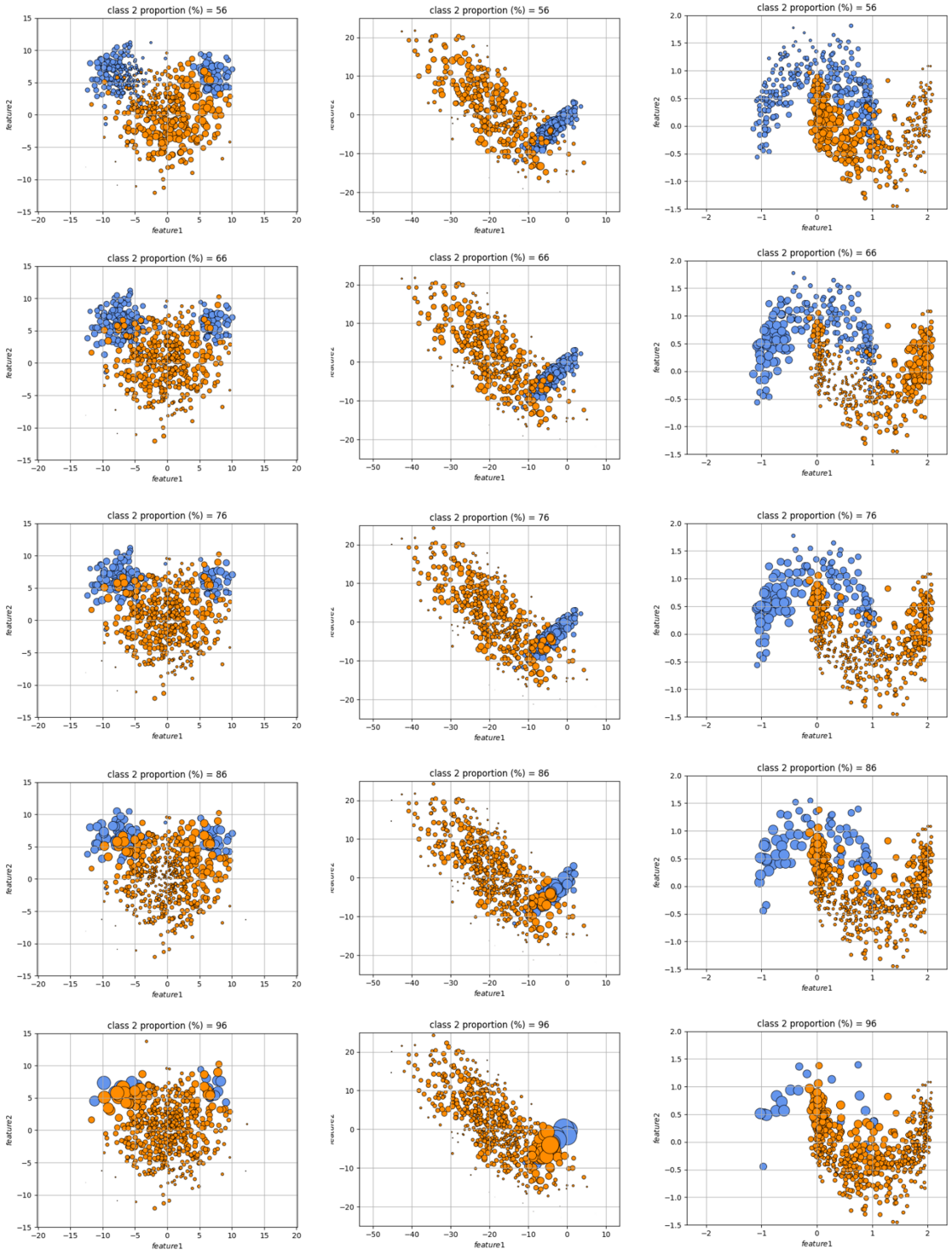


Figure D.1: Visualisation of the weights assigned by KMM to the train samples for various class imbalance ratios. Class 2 proportion (%) takes on values 56, 66, 76, 86, 96. Larger sizes indicate larger weights.

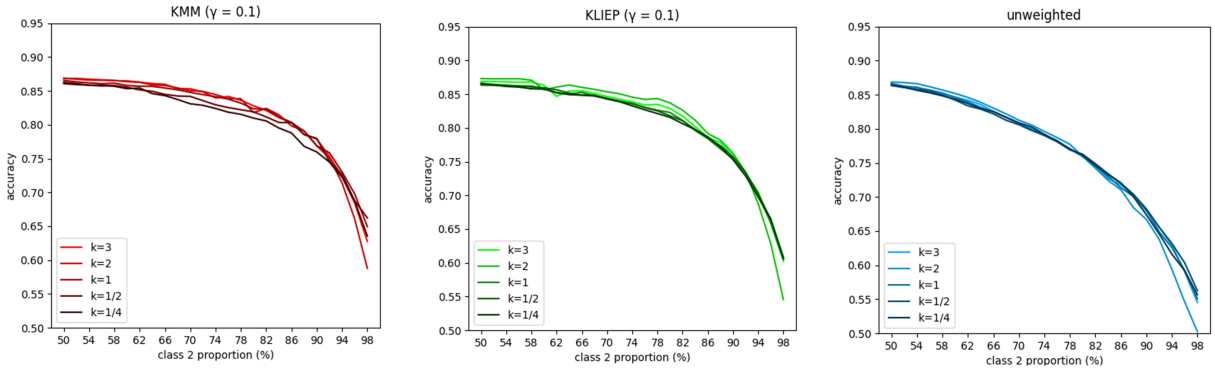


(a) Three unbalanced clusters (b) Overlapping linear clusters (c) Rotated moons

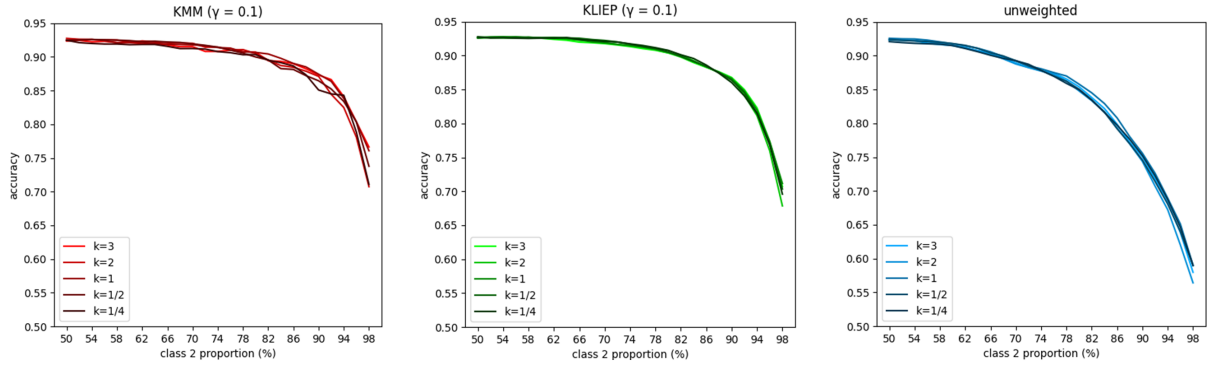
Figure D.2: Visualisation of the weights assigned by KLIEP to the train samples for various class imbalance ratios. Class 2 proportion (%) takes on values 56, 66, 76, 86, 96. Larger sizes indicate larger weights.

E Classification Accuracy for a Label-Based Sampling Scheme for Various Train Sample Sizes

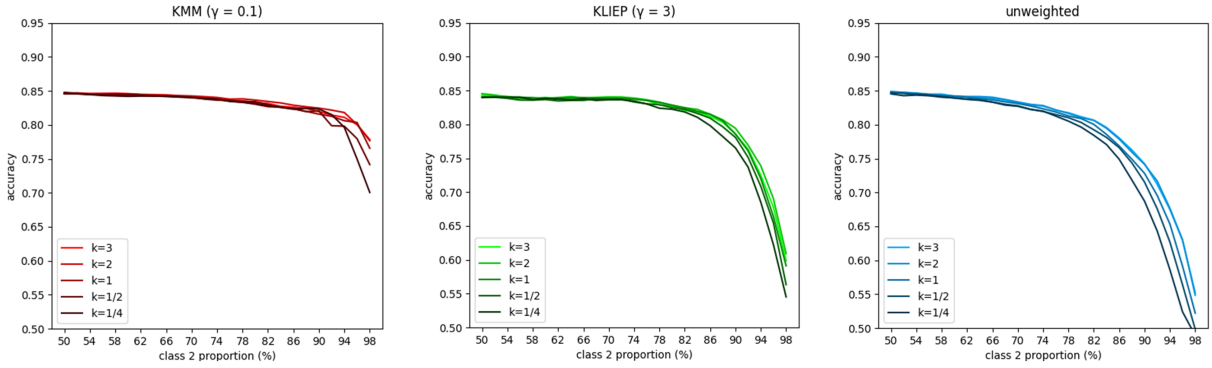
This appendix contains a visualisation of the performance curves of both the weighted and unweighted classifiers on a label-based sampling scheme when the train sample size is varied.



(a) Three unbalanced clusters



(b) Overlapping linear clusters



(c) Rotated moons

Figure E.1: Classification performance on a label-based sampling scheme for various proportions K of the original train sample size used in test case 1.