Applying Positive Unlabeled Learning Techniques and Using the Kullback-Leibler Divergence to Improve Geothermal Surveying Assessments

by

Martín Thomas Rodriguez

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science
in
Electrical and Computer Engineering

Thesis Committee:
John Lipor, Chair
James McNames
Christof Teuscher
Erick Burns

Portland State University
2024

Abstract

As we face the current climate crisis, the discovery of geothermal energy resources has the potential to greatly reduce our dependence on fossil fuels worldwide. However, the development of any new energy infrastructure is expensive and depends on the willingness of energy agencies and developers to make initial investments based on calculated risk measures. One such measure, called geothermal favorability, is the likelihood that a site has conditions favorable for geothermal systems containing recoverable energy potential. Its prediction from existing geophysical datasets proves to be a nontrivial task. The prediction of geothermal favorability can be framed as a binary classification problem, the data for which consists entirely of samples which are either positive (location contains a conventional hydrothermal system) or unlabeled (location does not contain any known geothermal system but should not be considered strictly negative). Data that comprises only positive and unlabeled examples is called positive unlabeled data, or PU data. PU learning is the branch of semi-supervised learning concerned with learning effectively from PU data. In addition to containing no negative examples, the dataset this thesis uses is heavily imbalanced, with many unlabeled examples and very few positive examples.

Previously, resource assessments have required experts to be directly involved in the assessment process, which can be time-consuming, expensive, and incurs human bias. Recent work in geothermal favorabililty prediction has used undersampling to mitigate class imbalance when training linear and nonlinear machine learning models, but more sophisticated techniques exist in the larger body of literature to specifically target the deficiencies in PU data. Furthermore, the metrics most common to the classification literature, e.g., the F_1 score and area under the receiver operating characteristic curve (ROC-AUC), do not adequately reflect the performance of models trained on PU data with such severe class imbalance. This thesis introduces the Kullback-Leibler divergence (D_{KL}) as a means of model evaluation for PU data and explores two PU learning techniques: Selected-at-Random Expectation-Maximization

(SAR-EM) and Difference-of-Estimated-Densities-based PU Learning (DEDPUL), on U.S. Geological Survey (USGS) data from 2008. It then compares these two most current PU learning techniques against previous naïve methods using logistic regression (LR) and XGBoost. We demonstrate that, when used as a scoring function to tune hyperparameters for linear and nonlinear machine learning models, the D_{KL} has an intrinsic ability to separate the unlabeled from predicted positive distributions. It has the weakness of being a highly variable scoring function, however: the strategy with the highest D_{KL} score has a standard deviation that is 38% larger than its mean.

In the PU learning context, a nontraditional classifier (NTC) is a classifier that is trained on PU data to separate positive from unlabeled examples as if it were performing traditional binary classification. NTCs are often an integral part of PU learning algorithms, where they function to reduce data dimensionality, provide a lower bound for class prior prediction, and act as a baseline for determining PU learning algorithm performance. We show that when LR is used to train an NTC, SAR-EM achieves a more accurate estimate of the class prior over other methods, with a mean absolute error (MAE) 185% lower than the closest naïve approach. SAR-EM also produces a slight improvement of 18% in F_1 score compared to all other methods. The favorability maps resulting from SAR-EM resemble those given by naïve LR methods in that the transition between regions of favorability is much more gradual and the lower favorability regions much smaller than the other methods explored. When XGBoost is used to train an NTC, DEDPUL enhances the model's bias toward heavily penalizing likely negative samples/regions with steep boundaries between favorability regions. When comparing ridge plots, naïve XGBoost is better at separating the unlabeled distribution from predicted positive results, even over more advanced PU methods. It remains that these simpler models such as LR and XGBoost, which rely on undersampling and the appropriate tuning of class weights, and treat all unlabeled examples as negative, can provide performance that is on par with the current state-of-the-art in PU learning.

To my mother, Lisa.

Acknowledgments

First and foremost, I'd like to thank my advisor, John Lipor, who has continued to show me patience and guidance throughout my complicated journey. Being taught by him feels like learning alongside a dear friend. John has seen me both fail and eventually succeed and I am truly grateful for his support.

I would also like to thank the rest of my thesis committee. Thank you to James McNames for keeping me on my toes with the hard questions and making me wonder if I really do understand what I am talking about. I knew after taking one of his signal processing courses that I wanted James to be on my committee and I'm glad that we could make it happen. Thank you to Christof Teuscher for his advice on reference-wrangling and making the difficult task of writing a thesis appear surmountable. Thanks to Erick Burns for his valuable expertise in geology, helping me craft my introduction, and filling in the gaps of my knowledge.

Thank you to Doug Hall for teaching me the five-minute rule, which I still use to this day. Although, sometimes for me, five minutes can last a few days. On that note, I'd like to thank all the other professors at Portland State who have pushed me to try new things and expand my comfort zone.

Finally, I'd like to thank my mom, Lisa, for all her love and support. To my grandmother, Gerry, for teaching me how to live a life of creativity. And to my cat, Spaghetti, for being my rock these past few years and reminding me that play is just as important as work.

Table of Contents

Abstract i

Dedication iii

Acknowledgements iv

List of Tables vi

List of Figures viii

List of Symbols xii

1 Introduction 1

1.1 Motivation . . . 1

1.2 Background and Related Work . . . 4

1.2.1 Geothermal Favorability Assessment . . . 4

1.2.2 Binary Classification . . . 10

1.2.3 Positive Unlabeled Classification . . . 12

1.2.4 Assumptions to Enable PU Learning . . . 14

1.2.5 Class Prior and Propensity Estimation . . . 20

1.2.6 PU Learning Techniques . . . 22

1.2.7 A Note on Propensity Score Estimation . . . 31

1.2.8 Evaluating PU classifiers . . . 33

1.3 Contributions . . . 40

1.3.1 The Use of the D_{KL} as an Evaluation Metric . . . 40

1.3.2 Comparison of PU Learning Approaches . . . 40

2 Methodology 42

2.1 Experimental Design . . . 42

2.1.1 Data Features and Labels 45
2.1.2 Applying the D_{KL} to Select Hyperparameters 45
2.1.3 Comparing PU Learning Methods 48
2.2 Model Evaluation 48

3 Results **52**
3.1 Identifying the Class Prior 52
3.2 Optimal Hyperparameters 52
3.3 Model Performance 54
3.4 Favorability Maps 61
3.5 Single-Feature Maps 66
3.6 Model Predictions 68

4 Discussion **73**
4.1 The Use of the D_{KL} for Model Evaluation 74
4.2 PU Learning Methods for Geothermal Favorability Prediction 75

5 Conclusion & Future Work **77**
5.1 The Use of the D_{KL} for Model Evaluation 77
5.2 PU Learning Methods for Geothermal Favorability Prediction 78

References **80**

List of Tables

1.1 Comparison of attributes between state-of-the-art PU learning approaches. As of the writing of this thesis, SAR-EM [1] is the only general PU learning algorithm based on the SAR assumption rather than the more restrictive SCAR assumption. KM2 [2] is the algorithm with the least support for large datasets, as explained in Section 1.2.6.3. nnPU [3] and AlphaMax [4] are not evaluated in this thesis, but are included in this table to give the reader a more general overview of the PU learning landscape. 32

3.1 Estimates for label frequency c and class prior α for SAR-EM using logistic regression (LR) with F_1-tuned (F1) and D_{KL}-tuned (KL) and DEDPUL using XGB classifier models. Ground truth for class prior estimation is the expert opinion-based class prior of ≈ 0.0014 (1.40×10^{-3}) [5, 6]. Each case is abbreviated within the table as [*NTC model*]+[*source of hyperparameters*], e.g. LR+F1 is logistic regression tuned via F_1 score. Bold text indicates the best result. 53

3.2 Optimal hyperparameters for logistic regression per tuning metric used. . . . 54

3.3 Optimal hyperparameters for XGBoost per tuning metric used. 54

3.4 Mean and std. dev. test scores over all 120 train/test splits. Model trained for inductive inference. Each case is abbreviated within the table as [*NTC model*]+[*source of hyperparameters*], e.g. LR+F1 is logistic regression tuned via F_1 score. Bold text indicates the highest score for each metric. 56

3.5 Out-of-sample test scores (inductive inference) over all data for k=5 folds. Each case is abbreviated within the table as [*NTC model*]+[*source of hyperparameters*], e.g. LR+F1 is logistic regression tuned via F_1 score. Bold text indicates the highest score for each metric. 57

3.6 Results of PU metrics, where PUF [7], $\widehat{TPR}$, $\widehat{FPR_{PU}}$, $\widehat{FPR}$, $\widehat{Precision}$, and $\widehat{AUC}$ [8] are outlined in Section 1.2.8.2. Scores result from models trained for inductive inference. Each case is abbreviated within the table as [*NTC model*]+[*source of hyperparameters*], e.g. LR+F1 is logistic regression tuned via F_1 score. Bold text indicates the highest score for each metric. 58

List of Figures

1.1 Favorability maps for logistic regression using (a) 12 models from 2008 assessment [9] created using different feature sets, averaged [6] and (b) five features used in [5] and [6]. Dots show the locations of known geothermal sites. . . . 6
1.2 Favorability maps created by [6] using (a) logistic regression and (b) XGBoost. Dots show the locations of known geothermal sites. 7
1.3 Venn diagram showing the set formulation for the PU learning problem with class imbalance. One only has access to the subset of labeled positive examples and the subset of unlabeled examples. The goal is to uncover the hidden boundary such that all true positive examples can be reliably separated from the negative examples. 14
1.4 Illustration highlighting the difference between the single-training-set and case-control scenarios. In the single-training-set scenario, the training data is an i.i.d. sample from the overall distribution and the true y is unknown to the learner. In the case-control scenario two independent datasets are used for training: one containing only positives called the cases, and another being an i.i.d. sample from the overall distribution called the background sample. The positive samples in both scenarios are selected according to their propensity scores, a compact way to represent some probabilistic labeling mechanism. . 16
1.5 Illustration of D_{KL} between two univariate Gaussian distributions. As the mean of q (the estimated PDF of the positive instances) increases and/or its variance decreases, the KL divergence from p to q increases. 39

2.1 Overview of train-all-predict-all and out-of-sample prediction schemes. Both methods use the same strategy to arrive at the optimal hyperparameters; subsampling of the validation and test sets is carried out at this step to reflect the expert-opinion positive-negative ratio of 1:700. At the final step in the out-of-sample prediction scheme, no such subsampling is performed. This step is illustrated in Figure 2.2. 43
2.2 Obtaining out-of-sample test results via standard stratified k-fold cross-validation for $k = 5$. Stratification ensures coverage over the entire dataset. Figure adapted from scikit-learn documentation [10, 11]. 44
2.3 Diagram showing the normal score transform pipeline for prediction scores and how this transform affects the resulting PDF ridge plot. Top of diagram shows the PDF for the combined distribution (including positive and unlabeled data), and bottom of diagram shows the PDFs for the separated positive and unlabeled sets. 49
2.4 Example ridge (top) and CDF (bottom) plots from Mordensky et al. [6]. . . 50

3.1 Box and whisker plots showing evaluation metrics for logistic regression (LR) and XGBoost (XGB), with hyperparameters tuned using the F1 score (F1), recall, ROC-AUC, or D_{KL} (KL). Scores result from averaging over 120 train/test splits in an inductive inference scheme. This figure included to show a comparison of the various naïve methods. 59
3.2 Box and whisker plots showing evaluation metrics for logistic regression (LR), XGBoost (XGB), and SAR-EM, with hyperparameters tuned using the F1 score (F1) or D_{KL} (KL). Fewer tuning methods are included in this figure in order to show a comparison between select naïve strategies and one PU strategy. For SAR-EM, tuning refers to the tuning of the NTC under the SCAR assumption. A SAR-EM experiment involving no tuning is included, with the default parameters of $C = 1.0$ and equal class weighting. Scores result from averaging over 120 train/test splits in an inductive inference scheme. . . 60
3.3 Favorability maps for logistic regression (LR) tuned using (a) F_1 scores (F1), (b) Recall, (c) ROC-AUC, or (d) D_{KL} (KL). Circles show the locations of known geothermal sites. Models are trained for transductive inference. . . . 62
3.4 Favorability maps for XGBoost (XGB) tuned using (a) F_1 scores (F1), (b) Recall, (c) ROC-AUC, or (d) D_{KL} (KL). Circles show the locations of known geothermal sites. Models are trained for transductive inference. 63

3.5 Favorability maps for SAR-EM using logistic regression (LR) (a) with F_1-tuned (F1), (b) D_{KL}-tuned (KL), and (c) default (no tuning) classifier models. Circles show the locations of known geothermal sites. Models are trained for transductive inference. 64
3.6 Favorability maps for DEDPUL with XGBoost (XGB) as NTC, tuned using (a) F_1 scores (F1), (b) Recall, (c) ROC-AUC, or (d) D_{KL} (KL). Circles show the locations of known geothermal sites. Models are trained for transductive inference. 65
3.7 Single-feature maps for (a) heat flow (hf_raw) and (b) seismic density (eqptd_raw). 66
3.8 Single-feature maps for (a) magma distance (mag_raw), (b) fault distance (flt_raw), and (c) stress (str_raw). 67
3.9 Ridge (top) and CDF (bottom) plots for logistic regression (LR) and XGBoost (XGB) prediction results, tuned using either their F_1 scores (F1), Recall, ROC-AUC, or D_{KL} (KL). Models are trained on all data and used to predict on all data. This is an indication of each model's performance when used for transductive inference. 69
3.10 Ridge (top) and CDF (bottom) plots for logistic regression (LR) and XGBoost (XGB) prediction results, tuned using either their F_1 scores (F1), Recall, ROC-AUC, or D_{KL} (KL). Stratified cross-validation is used to obtain out-of-sample predictions across the entire dataset. This shows the relative model performance when used for inductive inference. 70
3.11 Ridge (top) and CDF (bottom) plots showing comparison of non-PU methods with SAR-EM methods. Similar to Figure 3.9, models are trained on all data and used to predict on all data in a transductive manner. 71
3.12 Ridge (top) and CDF (bottom) plots for DEDPUL with XGBoost (XGB) as NTC, tuned using either the F_1 scores (F1), Recall, ROC-AUC, or D_{KL} (KL). Models are trained on all data and used to predict on all data in a transductive manner. 72

List of Symbols

Symbol	Description
$\mathbf{x}$	Example vector of attributes
$\mathbf{X}$	Set of example vectors comprising a data matrix
y	Example target variable
$\mathbf{y}$	Target vector
s	Example label indicator
$\mathbf{s}$	Label indicator vector
α	Class prior, $\alpha = \Pr(y = 1)$
c	Label frequency, $c = \Pr(s = 1 \mid y = 1)$
e	Propensity score function, $e(x) = \Pr(s = 1 \mid y = 1, x)$
$f(\mathbf{x})$	Classifier output for one example
$f_x(x)$	Probability density function (PDF) of the entire instance space
$f_{x_p}(x)$	PDF of the positive instance space
$f_{x_n}(x)$	PDF of the negative, unobserved instance space
$f_{x_l}(x)$	PDF of the labeled instance space
$f_{x_u}(x)$	PDF of the unlabeled instance space
$\hat{\bullet}$	An estimate of $\bullet$

1 Introduction

1.1 Motivation

Geothermal energy is a renewable resource that has proven to be an asset in helping our society move further away from a dependence on fossil fuels [12–16]. Although the energy produced from geothermal systems worldwide saves just under half a billion barrels of oil annually, it only accounts for about one percent of energy generated from all renewable sources, which in turn accounts for less than 10 percent of the total global energy supply [17–19]. A 2008 geothermal resource assessment [20] conducted by the U.S. Geological Survey (USGS) estimated that within the U.S. there is the mean potential for a three-fold increase in power generation if currently identified and undiscovered conventional geothermal systems are incorporated into the national energy portfolio. This indicates a large margin for further geothermal energy production predicated on the discovery of previously undiscovered geothermal energy sites, which can both reduce our dependence on fossil fuels and help to balance fluctuating renewables like wind and solar.

The earliest U.S. geothermal survey assessments, starting in 1965 and continuing through 1973, varied widely in their concluding estimates. Subsequent expert-driven national assessments [21, 22] completed from 1975–1978 were aimed at reconciling the variability in these estimates and better quantifying the national geothermal potential. They also succeeded in establishing concrete guidelines and definitions related to the potential extraction of geothermal resources.

Adapting machine learning methods to this problem of geothermal favorability was the

aim of two assessments [9,23] conducted by the USGS on potential moderate- ($90°C$ to $150°C$) to high-temperature ($> 150°C$) geothermal energy sources in the western U.S. Although they were successful in applying logistic regression (LR) and weight-of-evidence (WoE) models to existing data to achieve results comparable with previous methods, they relied on expert analysis for several key aspects of the model building process such as binning and thresholding of features, which remains time-consuming and incurs human bias. A more data-driven approach was taken recently by Mordensky et al. [5,6], who, through applying LR, eXtreme Gradient Boosting (XGBoost) [24], Support Vector Machines (SVMs) [25], and multilayer perceptron artificial neural networks (ANNs) in single-classifier and ensemble approaches (for all except the ANN) to the five most successful features from [9, 23], were able to achieve comparable results to these expert-driven assessments.

The labeled data these most recent assessments [5, 6] used suffers from severe class imbalance of one positive for every 2600 nonpositive examples, and the subset of data with an observed target variable consists solely of positive examples. The nonpositive examples must be considered unlabeled; treating them as explicitly negative rather than unlabeled would possibly result in a classifier with poor performance, since its training dataset could contain undiscovered positive examples which are incorrectly labeled as negative. Similarly problematic or noisy data is the norm among geothermal datasets collected worldwide, which are often reported differently by different organizations, may originate from inconsistent methods of data collection, contain many outliers, or be missing some data from one or more classes [18]. Data that consists of a positive set and unlabeled mixture set composed of unidentified positive and negative examples is called positive unlabeled (PU) data. PU learning, or learning from PU data, is a branch of the semi-supervised learning problem, since rather than having access to some labels from each domain as in the generic semi-supervised scenario, only a subset of the positive examples are observed [26, 27]. Generally, semi-supervised machine learning models are improved when unlabeled data is included in

training rather than removing it altogether [28].

The two main problems central to PU learning are, firstly, how to train PU learning models and, secondly, how to evaluate trained PU learning models. Techniques may be borrowed from other domains where PU data naturally emerge, like natural language processing and knowledge base building, to assist in mitigating the issue of missing data and to build more robust models [29]. Where common cardinality-based metrics such as the F_1 score fall short of estimating true PU classifier performance [8], the Kullback-Leibler divergence (D_{KL}) [30] is a possible alternative measure of model robustness. The D_{KL} measures the difference between two continuous or discrete distributions and has previously been used successfully in problems like anomaly detection [31, 32] and class prior estimation in unlabeled test data [33]. Here we apply the D_{KL} in the process of hyperparameter optimization, where the hyperparameters that are selected for training are those which maximize the D_{KL} between the unlabeled data and the model's predicted positive data after normal score transformation. By using the D_{KL} in this manner, hyperparameters which tend to push these two distributions farther apart are favored.

As datasets grow, both in terms of number of geophysical features and in sample size, the need for data-driven methods increases. The question arises of whether it is possible to create a solely data-driven model from PU data that can correctly determine the viability of potential geothermal energy sources with equal or better performance to methods which rely on human decisions, thus saving time on future surveys and improving the likelihood of success. Further, can recent advances in PU learning be incorporated to add to the effectiveness of these data-driven models?

This thesis examines the Kullback-Leibler divergence as a method of model evaluation for positive unlabeled data and studies the application of current techniques in positive unlabeled learning to the estimation of geothermal favorability in the western United States.

1.2 Background and Related Work

1.2.1 Geothermal Favorability Assessment

The first national geothermal resource assessment conducted in [21] defined geothermal resources as "stored heat, both identified and undiscovered, that is recoverable using current or near-current technology, regardless of cost." White et al. [21] found that many of the currently identified systems were likely to result in new geothermal discovery, for the reason that the volume or temperature contained within the systems may be higher than previous measurements and estimates indicated. The systems they examined were also grouped in terms of the type of recoverable geothermal resource; only two of the four categories they identify occur with any frequency in the region of interest for this thesis (western U.S.). These are:

- Regional conduction-dominated areas, and
- High-temperature ($> 150°C$) and moderate-temperature ($90°C$ to $150°C$) hydrothermal convection systems [21].

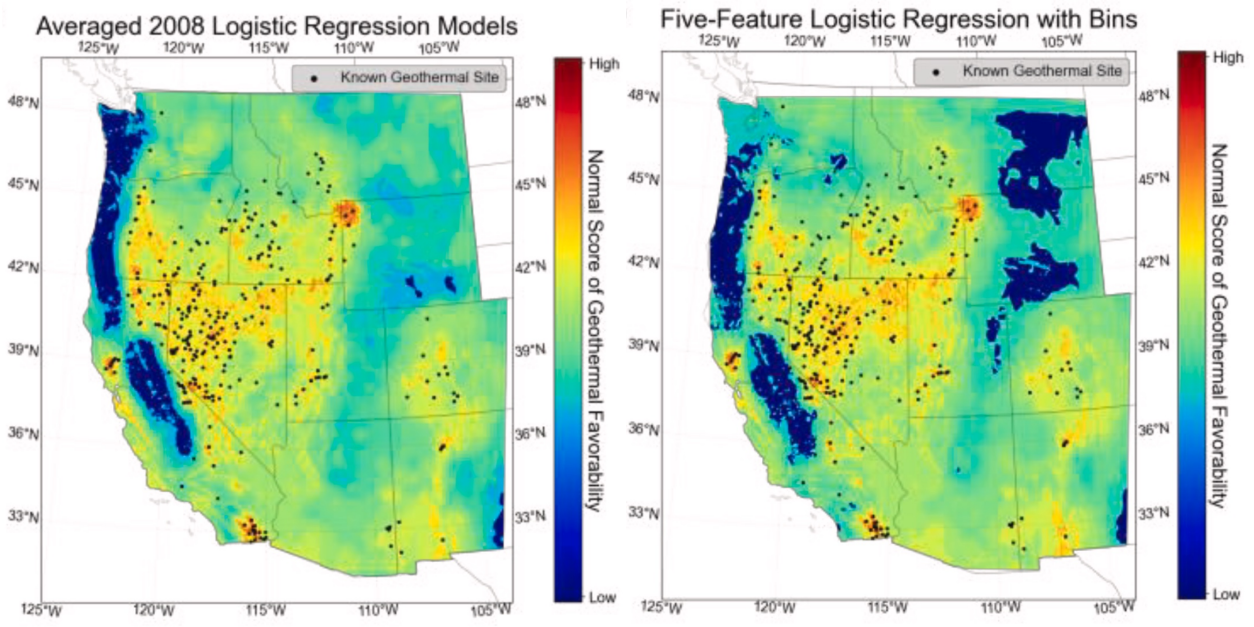
White et al. estimated that within the discovered high-temperature hydrothermal convection systems, there exists the potential for a five-fold increase in the geothermal resources waiting for future discovery [21]. Within the intermediate-temperature systems, they estimated the potential presence of more than three times the recognized geothermal resources resulting from hydrothermal convection systems [21].

A subsequent circular [22] expanded on this first assessment, including in their discussion regions with low-temperature ($< 90°C$) hydrothermal convection systems and updating the results based on moderate- to high-temperature hydrothermal convection systems by incorporating statistical methods, which, depending on the system, resulted in a significant

increase or decrease of estimated thermal reservoir areas or temperatures. Overall, [22] reported 24 percent fewer hydrothermal convection systems than those reported by [21].

The most recent moderate- to high-temperature geothermal resource assessments [9, 23] conducted in 2008 by the USGS used WoE and LR to create 28 models with the goal of predicting regions which are likely to have high geothermal favorability by mapping the ratio of posterior to prior probability over a varying combination of feature sets within the data. In general, geothermal favorability is a measure of likelihood that, within a region, favorable conditions for geothermal systems exist [5]. Although these linear models (WoE and LR) help to mitigate the bias inherent in expert-based decision making, they maintain a heavy reliance on expert opinion in several key aspects of the model-building process like feature selection and preprocessing, binning, and thresholding [5].

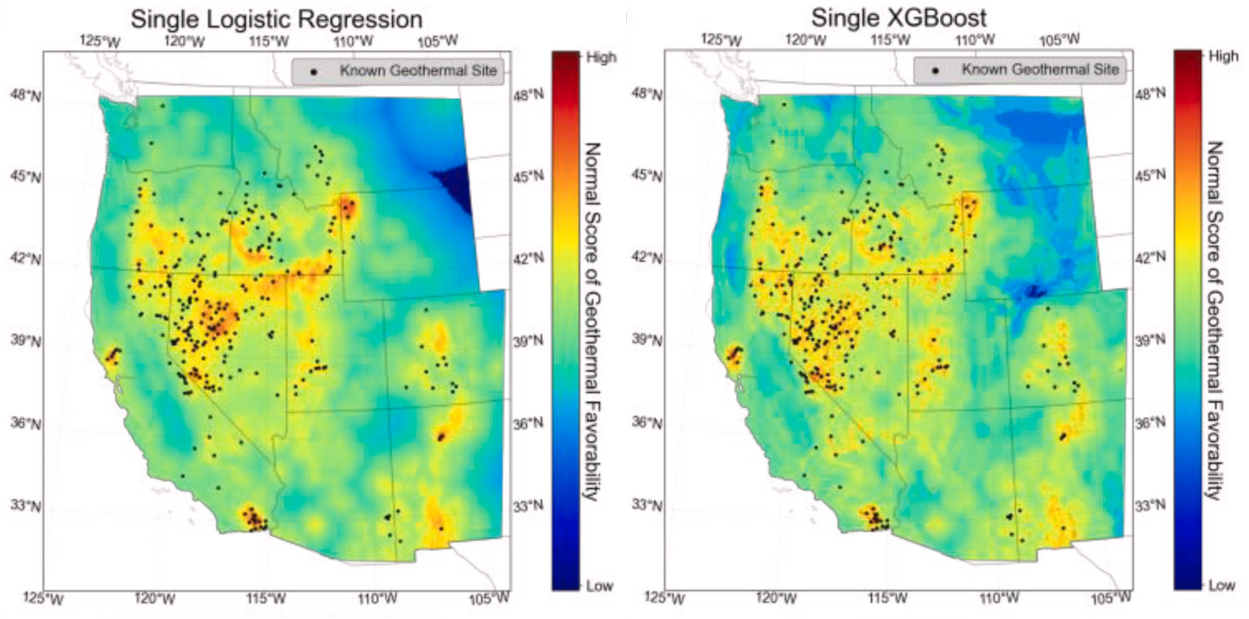
The dataset used in [9] is the result of gridding the western U.S. into cells, 2 km-by-2 km, and assigning each cell a positive label if it contains a known conventional hydrothermal system. Out of 725,442 total cells, only 278 belong to the positive class (contain a known conventional hydrothermal system). This dataset, then, suffers from severe class imbalance at a ratio of about 1:2600 positive:unlabeled data points. Many machine learning algorithms perform best when classes are balanced, and often when a dataset has moderate class imbalance (from 1:$>$1 to 1:$<$100), strategies like oversampling, undersampling, or sample weighting are employed to mitigate the imbalance [34–36]. Unfortunately when class imbalance is severe (1:$<$100), if one simply predicts the majority class for all examples, one can achieve a highly accurate model even though this model provides no information about the minority distribution of interest. Another strategy, then, might be the use of different evaluation metrics or learning algorithms better suited to datasets with class imbalance or missing data. Williams and DeAngelo [9] assumed that all non-positive data points were negative; better performance might be achieved by, for example, assigning some importance to finding data points which were unlabeled but true positives.



(a) **(b)**

Figure 1.1: Favorability maps for logistic regression using (a) 12 models from 2008 assessment [9] created using different feature sets, averaged [6] and (b) five features used in [5] and [6]. Dots show the locations of known geothermal sites.

Recent work in [5] applied LR, XGBoost, and SVM models in both single-classifier and ensemble approaches to a subset of the data used in the assessment in [9]. A subsequent study in [6] also compared these methods to a multilayer perceptron ANN. The five raw features used in these most recent studies were: estimated conductive heat flow in milliwatts per square meter ($\mathrm{mW/m^2}$), distance to nearest Quaternary fault in meters, distance to nearest magma body in meters, seismic event density for all events greater than magnitude (M) 3 within a 4 km radius in number of M3 events per square kilometer ($\mathrm{km^2}$), and maximum horizontal stress in megapascals (mPa). These features are abbreviated "heat flow," "fault distance," "magma distance," "seismic density," and "stress," respectively. Mordensky et al. [5, 6] found that on average, the best performance (using the average F_1 score) was achieved using the single-classifier LR model, even over the more complex nonlinear and ensemble approaches. Yet, the nonlinear models (SVMs, XGBoost, and ANNs) produced results which were in better agreement with [9], indicating a need for a more revealing method of model evaluation



(a) **(b)**

Figure 1.2: Favorability maps created by [6] using (a) logistic regression and (b) XGBoost. Dots show the locations of known geothermal sites.

and the possibility that the expert decisions were in fact adding some degree of nonlinearity to the linear models from the previous assessments [5].

Figures 1.1 and 1.2 show the favorability maps created by [5] to compare the expert decision-based results in [9] with those created by more data-driven models (the single LR and XGBoost strategies). The four maps are generally in agreement in areas of high favorability but differ most in areas of lower favorability. Notably, the data-driven models in Figure 1.2 show a smoother transition from high- to low-favorability regions than the expert decision-based models in Figure 1.1, which have steeper transitions and more concentrated regions of low favorability. This indicates that, in effect, adding expert decisions to the model building process (through data selection, thresholding, and binning) added nonlinearities to these linear models [6].

Although it is desirable to incorporate domain knowledge into the model building process, one must be wary of the side effect of incurring human bias through thresholding or binning

of a problem which is potentially unnecessary. To this end, [23] identified three potential characteristics of undiscovered geothermal systems:

- They might include flow from thermal springs,
- They may have no surface/near-surface discharge but significant thermal anomalies, and
- They may be deep systems with modest or undetectable thermal anomalies.

Furthermore, a process similar to that observed in petroleum resource assessments may be influencing the separation between discovered and undiscovered geothermal systems. Petroleum reserves often consist of larger, more easily discoverable systems whereas the smaller, less historically discoverable systems outnumber them [23]. This suggests the potential to model a certain *propensity* function which determines by what mechanism examples are chosen to be labeled [37]. "Chosen to be labeled" implies the example is also from the positive class. Accurately estimating the propensity score, either by domain knowledge or through learning, has the potential to improve the performance of existing resource assessment models.

Mordensky et al. [5, 6] used random undersampling of the validation and test set to simulate the estimated natural distribution of the target variable. This natural distribution estimate was created by taking an estimate of the mean undiscovered power potential from [9], estimating the mean power generation from identified systems, and using Equations 1.1 through 1.3 to arrive at an estimate of the total number of geothermal systems, both discovered and undiscovered [6].

$$\text{Avg. System Power Generation} = \frac{\text{Power Generation of Identified Systems}}{\#\text{ of Identified Systems}} \tag{1.1}$$

$$\#\text{ of Undiscovered Systems} = \frac{\text{Total Undiscovered Power Potential}}{\text{Avg. System Power Generation}} \tag{1.2}$$

$$\text{Total \# of Geothermal Systems} = \text{Identified Systems} + \text{Undiscovered Systems} \tag{1.3}$$

Using the mean power production estimates from [9], [6] arrived at an estimated natural class ratio of around 1:700 positive:negative data points. The validation and test sets were then undersampled to obtain this balance and allow for better estimation of classifier performance. Despite undersampling, the 2008 USGS dataset contains very few positive examples, which far surpasses the threshold for what is considered an extremely imbalanced dataset and makes learning difficult. Both [5] and [6] report mean F_1 scores of less than 0.10 for all classifiers.

Although undersampling can help mitigate modest class imbalance, it is likely that better estimator performance might be achieved through other means. Mordensky et al. [6] arrived at several key insights to inform future models, which are summarized here:

- Metrics other than the F_1 score may be better suited to the evaluation of models created from PU data, particularly when the data suffers from severe class imbalance.
- Better models might incorporate techniques that specifically address the positive-unlabeled nature of the data.
- It may be the case that geothermal systems are better explained as a multiclass classification problem, that is, positive examples should be separated into classes based on their specific properties, e.g., separating "large" and "small" geothermal systems, or systems which are recoverable with current technology and those which may one day be recoverable, into distinct classes.
- The data from the 2008 USGS assessment is inherently limited in its utility and an

effort should be made to explore better methods of data collection, standardization, and feature engineering in order to gain more information relevant to building a representative geophysical model.

This thesis examines the first and seconds points above.

Poor classifier performance using traditional metrics, and agreement between models incorporating expert-led decisions in [9] and those using data-driven methods in [5], indicate that improvement over these initial findings is possible.

Out of a necessity for dealing with weakly supervised datasets that are similar to the 2008 USGS dataset, in domains such as text classification, knowledge base construction, and gene identification, a subset of semi-supervised machine learning algorithms have emerged under the name PU learning [27, 29, 38, 39]. The diversity in PU learning approaches is immense, so this thesis considers only a few of these approaches for application to geothermal favorability prediction on the subset of the 2008 USGS dataset used in [5, 6]. A motivation for the PU learning approach for binary classification follows.

1.2.2 Binary Classification

A binary classifier is a one-to-one function which maps a real-valued matrix $\mathbf{X}^{m \times n}$ consisting of m sample vectors $\{\mathbf{x}^1, \ldots, \mathbf{x}^m\}$, each with n features, to a target vector $\mathbf{y}^{m \times 1}$ consisting of binary target variables $\{y^1, \ldots, y^m\}$. Formally, given a feature vector $\mathbf{x} \in \mathbb{R}^n$, the classifier $f : \mathbb{R}^n \to [0, 1]$ outputs the class label $y \in \{0, 1\}$ where $y = 1$ is a positive outcome and $y = 0$ is a negative outcome. A classifier is a probabilistic classifier if it outputs the probability that an example belongs to each class, where the probability of all classes sums to one [27].

For classifier strategies that output a probability or probability-like score and do not directly output a binary target variable, a threshold (generally 0.5) is chosen to select between a positive and negative result. Other threshold values may result in better performance but this must be tuned per-algorithm and per-dataset.

The classifier output can be related to Bayes' theorem, given in Equation 1.4

$$\Pr(y = 1|\mathbf{x}) = \frac{\Pr(\mathbf{x}|y = 1)\Pr(y = 1)}{\Pr(\mathbf{x})}. \tag{1.4}$$

Assuming that the classifier gives a probabilistic output, which is true for LR, XGBoost, and others (excluding SVMs), the output of $f(\mathbf{x})$ can be viewed as the posterior probability $\Pr(y = 1|\mathbf{x})$. For algorithms that do not output a probability-like score, their outputs can be easily transformed into such via, for instance, Platt scaling or isotonic regression [27].

Given the likelihood $\Pr(\mathbf{x}|y = 1)$, the prior probability (class prior) $\Pr(y = 1)$, and the marginal probability $\Pr(\mathbf{x})$, it is possible to exactly recover the posterior. In practice, however, the likelihood and class prior must be estimated from the data. The marginal probability is usually ignored since the existence of the data itself does not depend on y and it only affects the end result by a constant value [27].

A training example in a traditional supervised binary classification problem is represented by the tuple $(\mathbf{x}, y)$, where $\mathbf{x}$ is a n-length feature vector, y is the target variable, and it is assumed that every training example is correctly labeled. A classifier is then trained on each training example such that given a new, unseen example $\hat{\mathbf{x}}$, it can produce the prediction $\hat{y} : f(\hat{\mathbf{x}}) = \hat{y}$. This forms the principle of *inductive* inference, which attempts to learn a general rule that can be applied to new data. In contrast, *transductive* inference attempts to learn a rule for a specific dataset by training and testing on the same data. This means that models trained by transductive learning cannot be as easily generalized. Inductive and transductive inference are discussed further in Section 2.2.

Another assumption made in traditional binary classification concerns the origin of the training set. It is assumed that the training set $\mathbf{X}$ composed of samples $\{\mathbf{x}^1, \ldots, \mathbf{x}^m\}$ is an

i.i.d. (independent and identically distributed) sample of the total population $f_X(x)$

$$\begin{aligned} \mathbf{X} &\sim f_x(x) \\ &\sim \alpha f_{x_p}(x) + (1-\alpha) f_{x_n}(x), \end{aligned} \tag{1.5}$$

where $\alpha = \Pr(y = 1)$ is the class prior and $f_{x_p}(x)$ and $f_{x_p}(x)$ are the probability density functions (PDFs) of the positive and negative examples respectively. Written another way, the distribution $\Pr(\mathbf{x})$ of a single example $\mathbf{x}$ is

$$\Pr(\mathbf{x}) = \alpha \Pr(\mathbf{x}|y = 1) + (1 - \alpha) \Pr(\mathbf{x}|y = 0). \tag{1.6}$$

So far the fully-supervised scenario has been assumed. However, data originating from many domains might contain label noise, partially labeled data (weakly-supervised learning), be missing data from one or more classes, or in the extreme, be completely unlabeled (unsupervised learning) [29]. These scenarios arise quite often in real-world datasets and when all the possibilities of label noise are considered, perfect, fully labeled data is a rare phenomenon [40]. It is therefore worthwhile to explore methods of using data in the PU setting to train new models and evaluate the performance of existing models in order to better generalize these models to real-world scenarios.

1.2.3 Positive Unlabeled Classification

A traditional probabilistic binary classifier is a classifier that has been trained on a set of positive and negative examples selected randomly from the overall distribution $\Pr(x, y)$, to learn a function $f(x)$ that approximates the posterior probability of an example $\mathbf{x}$ belonging to the positive class, $\Pr(y = 1|\mathbf{x})$ [38]. When training data contains only labels from the positive class, this is a special case of weakly-supervised learning known as PU learning. PU learning is closely related to other efforts to study learning in less-than-ideal conditions, such

as learning with noisy labels [40, 41]. In areas of machine learning research such as medical diagnoses, knowledge completion, spam filtering, social media sentiments, recommendation algorithms, and many others, negative examples are either sparse or do not occur naturally, so it is worth studying how to learn effectively from PU data [29]. Unfortunately, it is impossible to train a traditional classifier directly from PU data without making assumptions about the origins of the data [27, 29].

To formalize the PU learning scenario, a new random variable $s \in \{0, 1\}$, called the label indicator, was defined in [38]. If an example is selected to be labeled, this is indicated by $s = 1$. When y is unknown and the example could be of the positive or the negative class, it is indicated by $s = 0$. This PU conditional probability relationship is summarized by Equations 1.7 and 1.8

$$\Pr(y = 1|x, s = 1) = 1, \tag{1.7}$$

and

$$\Pr(s = 1|x, y = 0) = 0. \tag{1.8}$$

An example, drawn from the distribution $\Pr(x, y, s)$, becomes $(\mathbf{x}, y, s)$. The *propensity score* $e(x) = \Pr(s = 1|y = 1, x)$ accounts for the probability of each positive example from $f_{x_p}(x)$ to be selected for labeling by some probabilistic labeling mechanism. Without making any further assumptions about such a labeling mechanism, we can say that the PDF of the distribution of labeled examples $f_{x_l}(x)$ is a biased form of the PDF of the distribution of positive examples $f_{x_p}(x)$, the proof for which is given in [29]

$$f_{x_l}(x) = \frac{e(x)}{c} f_{x_p}(x), \tag{1.9}$$

where $c = \mathbb{E}_x[e(x)] = \Pr(s = 1|y = 1)$ is the fraction of positive examples selected to be labeled, the *label frequency*.

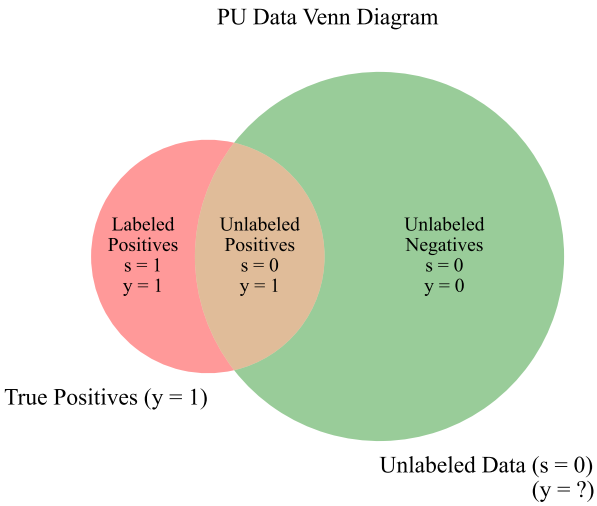


Figure 1.3: Venn diagram showing the set formulation for the PU learning problem with class imbalance. One only has access to the subset of labeled positive examples and the subset of unlabeled examples. The goal is to uncover the hidden boundary such that all true positive examples can be reliably separated from the negative examples.

In the context of PU learning, it is necessary to distinguish between a traditional binary classifier, which has been trained on a traditional positive-negative training set and whose goal is to separate positive from negative examples, and a nontraditional classifier (NTC), which has been trained on a PU dataset to separate positive from unlabeled examples. The NTC only has access to $\mathbf{s}$; $\mathbf{y}$ is unobserved [38].

The unavailability of labeled examples from both positive and negative distributions in PU learning require that some new assumptions be made about the origin of the data in order to train a successful classifier [29]. The next section explores those assumptions.

1.2.4 Assumptions to Enable PU Learning

Generally, the literature differentiates between assumptions made concerning the collection of the data, those made concerning the labeling mechanism, and general assumptions made about the data in order to facilitate PU learning [27, 29, 39]. These are outlined in the following three subsections.

1.2.4.1 Data Collection Assumptions

Prior work has identified two possible scenarios for the origin of the PU data collected: the *single-training-set scenario* and the *case-control scenario* [29, 38].

Single-training-set scenario Also sometimes called the censoring scenario [42], in the single-training-set scenario all training data are drawn randomly from the overall distribution $\Pr(\mathbf{X}, \mathbf{y}, \mathbf{s})$, but only $(\mathbf{X}, \mathbf{s})$ are saved for training; $\mathbf{y}$ is effectively discarded [38]. An easy-to-remember example of this is provided by [29]: data from surveys on topics that are often underreported, like smoking, can lead to the single-training set scenario. Social pressures might cause some people to report themselves as nonsmokers when the opposite is true, in effect censoring their own data. The single-training-set scenario is summarized by

$$\begin{aligned} \mathbf{X} &\sim f_x(x) \\ &\sim \alpha f_{x_p}(x) + (1-\alpha) f_{x_n} \\ &\sim \alpha e(x) f_{x_l}(x) + (1 - \alpha e(x)) f_{x_u}(x). \end{aligned} \tag{1.10}$$

Case-control scenario The case-control scenario assumes the training data is derived from two independent datasets, drawn independently from $\Pr(\mathbf{X}, \mathbf{y}, \mathbf{s})$. For the first dataset in the case-control scenario, only the examples with $s = 1$ are saved. This is the positive set and consists of "cases" or "presences" (leftmost set excluding the intersection in Figure 1.3) [38]. The second set in the case control scenario is the unlabeled set, also called the background sample, contaminated controls, or pseudo-absences, for which $s = 0$ (rightmost set including the intersection in Figure 1.3) [38]. The case control scenario can result from data being collected in different methods or from different populations [29]. A simple example of this scenario could arise if one were interested in classifying whether an audio clip contains a specific invasive bird call. Suppose one also has access to one dataset containing positive examples

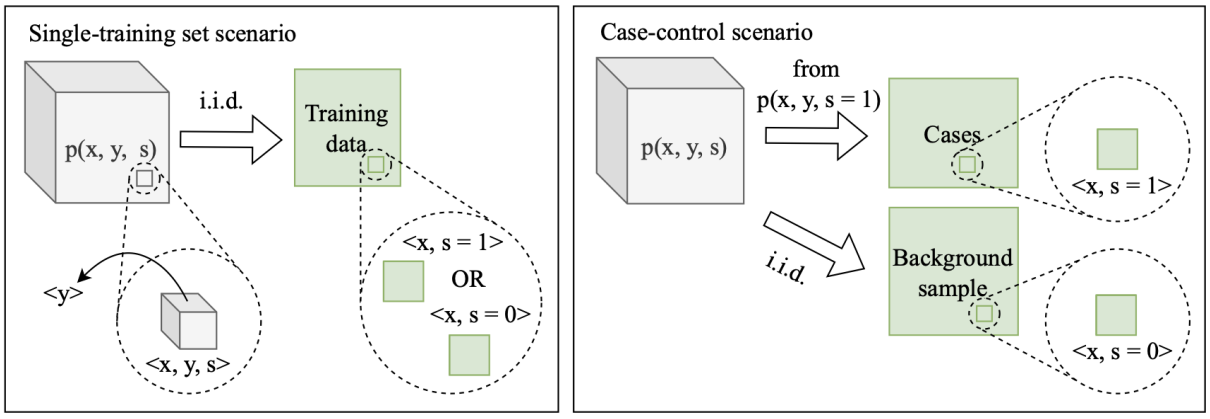


Figure 1.4: Illustration highlighting the difference between the single-training-set and case-control scenarios. In the single-training-set scenario, the training data is an i.i.d. sample from the overall distribution and the true y is unknown to the learner. In the case-control scenario two independent datasets are used for training: one containing only positives called the cases, and another being an i.i.d. sample from the overall distribution called the background sample. The positive samples in both scenarios are selected according to their propensity scores, a compact way to represent some probabilistic labeling mechanism.

collected from the bird's native habitat, and another dataset with unlabeled recordings taken in random locations. In the case-control scenario, the unlabeled dataset is described by

$$\begin{aligned} \mathbf{X}|\mathbf{s} = 0 &\sim f_u(x) \\ &\sim f_x(x) \\ &\sim \alpha f_{x_p}(x) + (1-\alpha) f_{x_n}(x). \end{aligned} \tag{1.11}$$

If one can control any duplicate data between the distributions and there is no covariate shift between the two sets of data, the case-control scenario can be reduced to the single-training set scenario [27]. In this thesis, because we are working with one dataset drawn i.i.d. from the overall distribution, we will assume the single-training-set scenario; we do not have two sets of data drawn from different populations or using distinct methods. An added bonus is that the single-training-set scenario is easier to work with and is the scenario most often assumed in the literature

1.2.4.2 Data Labeling Assumptions

Three assumptions have been proposed for the nature of the labeling mechanism: the *Selected Completely At Random* (SCAR) assumption, the *Selected At Random* (SAR) assumption, and *Probabilistic Gap*, also related to the *Invariance of Order* assumption [27, 29]. Elkan and Noto [38] showed that most PU strategies involve the SCAR assumption, which assumes that the labeled positive examples are selected uniformly at random from all positive data points. This simplifies the problem of learning from PU data to estimating one scalar value to account for the label frequency or class prior, which is equal for all positive data points, and then training a model to learn the labeling mechanism [1].

Selected Completely At Random Under the SCAR assumption, every positive example has exactly the same probability to be selected for labeling regardless of its attributes. It is often used in the literature because it is the simplest to work with and it works well for many datasets [1, 29, 38]. Because the labeling mechanism is necessarily biased in some way (it depends somewhat on the data itself or some external factors), a weaker assumption will be necessary to generalize PU techniques to a broader array of datasets [1, 27].

Selected At Random Instead of assuming a constant probability for selection among all positive examples, [1] considered the situation in which the probability for labeling is some unknown function of a subset of attributes used to train the classifier. This is referred to as the SAR assumption. In order to estimate the labeling mechanism from SAR PU data, [1] borrowed a value from work on causal inference [37, 43] called the *propensity score*, which indicates the probability a positive example is selected for labeling. The propensity score $e(\mathbf{x})$ is assigned to each positive example

$$e(\mathbf{x}) = \Pr(s = 1 | y = 1, \mathbf{x}). \tag{1.12}$$

Under the SCAR assumption, $e(\mathbf{x})$ is the same for all positive examples and is equal to the label frequency c. Bekker et al. [1] identified another possible assumption: the probability for an example to be labeled could depend directly on the probability that the example is positive. This third possibility is called the *Selected Not At Random* (SNAR) assumption.

If the propensity function can be learned or accurately represented by domain knowledge, this is equivalent to partially matching the positive distribution [27, 29, 44]. Partial matching in a PU learning context begins by making density estimates on two distributions: the first being the positive distribution resulting from estimation on the labeled examples, and the second being the overall distribution by estimating on all the data. One can then partially match the estimated overall distribution by minimizing some divergence criteria with a scaled version of the estimated positive distribution, where the class prior is the scaling factor [29, 44]. Of course, there can still exist some unlabeled positive examples which would not be captured by the partially matched positive distribution.

This thesis instead seeks to explore a method of maximizing the KL divergence between the predicted positive distribution and the unlabeled distribution to arrive at better predictions for geothermal favorability in a PU learning context.

Probabilistic Gap The probabilistic gap (PG) assumption is also known as the invariance of order assumption as used in [42, 45], and is closely related to the general smoothness assumption [27]. PG is a specific case of SAR that supposes that positive examples which are more similar to negative examples are generally less likely to be labeled. The PG for a dataset can be viewed as the difficulty of labeling a specific example. Furthermore, the propensity function is a non-negative, monotone increasing function of the PG [29]. One can use this assumption to select reliable negative examples, which one might do when applying a two-step technique described in Section 1.2.6.4, by selecting the unlabeled examples which have a smaller PG than the smallest PG among the labeled examples [45]. The scenario

examined in this thesis most probably falls under the SAR or PG assumptions, and could be modeled using either.

1.2.4.3 General Data Assumptions

The three general data assumptions used when searching for PU learning classifiers are negativity, separability, and smoothness, and are outlined in the following three paragraphs [27, 29].

Negativity Negativity supposes that all unlabeled examples are negative and trains a classifier as if this were true. This is the approach taken in [5] and [6], and is often used simply because it allows for the use of standard supervised binary classification algorithms [29]. This assumption becomes less reliable if more unlabeled positive examples are expected to be present in the dataset.

Separability Separability implies there is a decision boundary that optimally separates positives from negatives. This boundary may be a complex shape given by a function, where a parameter may be defined that optimally determines the position of the decision boundary [29]. This can be stated succinctly: the two classes of interest are non-overlapping or are naturally separated.

Smoothness Smoothness assumes examples that are close to each other in the feature space will have similar posterior probabilities, and are likely to have the same label [29].

1.2.5 Class Prior and Propensity Estimation

For the single-training-set scenario, relating class prior α to label frequency c under the SCAR assumption may be performed by

$$\begin{aligned} \alpha &= \Pr(y = 1) \\ c &= \Pr(s = 1)/\alpha. \end{aligned} \tag{1.13}$$

A similar relationship allows one to use the label frequency under the SCAR assumption to transform the output of a classifier trained to model the label mechanism $\Pr(s = 1|\mathbf{x})$ to the posterior probability of y

$$\Pr(y = 1|\mathbf{x}) = \Pr(s = 1|\mathbf{x})/c. \tag{1.14}$$

Still, the true value of the label frequency depends on a piece of information that can not be definitively known: whether the number of unlabeled examples is small due to a small positive class prior probability or a low label frequency [1, 27]. Estimating this value, then, is difficult without making further assumptions.

The following assumptions are made within the literature in order to facilitate identifying the class prior [1, 29]:

- *Separability*: The positive and negative classes are non-overlapping [38, 40, 46]. If all unlabeled positives could be somehow identified, an estimate for the class prior would naturally follow [29].

- *Positive subdomain/anchor set*: Define a subdomain A as any subset of the data determined by certain attribute values (partial variable assignment) [47]. Under the SCAR assumption, within any A, the label frequency remains constant and is equal to

the overall label frequency of the dataset, derived in [47]

$$\Pr(s = 1|x \in A, y = 1) = \Pr(s = 1|y = 1) = c, \tag{1.15}$$

and

$$c = \frac{\Pr(s = 1|x \in A)}{\Pr(y = 1|x \in A)}. \tag{1.16}$$

If A is a positive subdomain (anchor set) where $\Pr(y = 1|x \in A) = 1$, the probability of being labeled within A is the proportion of labeled examples among all examples within A and is exactly the label frequency of the dataset. In general, for all subdomains within the instance space, this probability is actually a lower bound on the label frequency [47],

$$c \geq \Pr(s = 1|x \in A). \tag{1.17}$$

To summarize, if such a purely positive anchor set can be found, the label frequency is calculated as the ratio of labeled examples to positive examples in this subdomain [44, 47–49].

- *Positive function*: This is a relaxation of the positive subdomain assumption in that it is not required to define a positive subdomain as resulting from partial variable assignment, but any function can define such a purely positive subdomain [2].

- *Irreducibility*: The negative distribution is irreducible and cannot contain any examples from the positive distribution [50, 51]. Irreducibility is implied by the three assumptions above.

The validity of these assumptions can only be tested on synthetic PU data for which the true distributions are known. For instance, we cannot assume irreducibility because it is highly likely that our unlabeled distribution contains positives and we have no known

examples from the negative distribution. It can be possible to identify very likely negative data, which is the goal of many PU learning techniques.

1.2.6 PU Learning Techniques

1.2.6.1 Elkan-Noto (EN) Method

An NTC is a classifier which has been trained on PU data to separate positive from unlabeled data in a naïve manner, that is, *as if the data were fully-supervised* [38]. Formally, it is a function $g(x)$ that maps the positive and unlabeled distributions, $f_{x_p}(x)$ and $f_{x_u}(x)$, to the posterior probability that an example is selected to be labeled, $\Pr(s = 1|x)$

$$g(x) = \Pr(s = 1|x) \equiv \frac{f_{x_p}(x)}{f_{x_p}(x) + f_{x_u}(x)}. \tag{1.18}$$

Taking $f(x)$ to be a traditional classifier and $g(x)$ to be its associated NTC, because f is a monotonically increasing function of g, if the goal is merely to provide optimal ranking of examples then g is equivalent to f [38]. If one requires more detailed estimates of PU classifier performance, obtaining an accurate estimate of the class prior can allow for a reduction in the bias resulting from the use of performance measures intended for models trained on supervised data to evaluate models trained on PU data [8].

Overall, [38] propose two methods for obtaining the traditional classifier $f(x)$ from the NTC $g(x)$. Under the SCAR assumption,

$$\Pr(s = 1|y = 1, \mathbf{x}) = \Pr(s = 1|y = 1) = c. \tag{1.19}$$

Replacing the posteriors in Equation 1.14 with their respective classifier outputs, we have

$$\Pr(y = 1|x) = \frac{\Pr(s = 1|x)}{\Pr(s = 1|y = 1)} = \frac{g(x)}{c} = f(x). \tag{1.20}$$

Therefore, under the SCAR assumption, an NTC $g(x)$ trained on the labeled positives and unlabeled data from a sample population produces conditional probabilities that differ by a constant factor c from those produced by a model $f(x)$ trained from the same sample population that has access to the fully labeled positive and negative examples.

With Equation 1.20 and one of the following three estimators for c, an estimate for $f(x)$ is obtained

$$\hat{c}_{e_1} = \frac{1}{|P|} \sum_{x \in P} g(x), \tag{1.21}$$

$$\hat{c}_{e_2} = \frac{\sum_{x \in P} g(x)}{\sum_{x \in V} g(x)}, \tag{1.22}$$

or

$$\hat{c}_{e_3} = \max_{x \in V} g(x), \tag{1.23}$$

where V is a validation set which is sampled from the entire dataset in the same manner as the training set and P is the subset of labeled positive examples within V. This is the first method given by [38], whereas the second method weights each individual training example. Positive examples get unit weight and unlabeled examples are duplicated such that each copy gets a complementary weight. A weight of $\Pr(y = 1|\mathbf{x}, s = 0)$ gets assigned to one copy and the other gets assigned the weight of $1 - \Pr(y = 1|\mathbf{x}, s = 0)$. An estimate for $\alpha = \Pr(y = 1)$ can be obtained from the following

$$\hat{\alpha} = \Pr(y = 1) = \frac{n^2}{m \sum_{x \in P} g(x)}, \tag{1.24}$$

where n is the number of labeled examples in the training set, and m is the total number of examples in the training set.

1.2.6.2 TIcE: Tree Induction for Label Frequency Estimation

In the *Tree Induction for Label Frequency (c) Estimation* (TIcE) strategy, decision trees are used to find a strong likely-positive subset within a sampled dataset [47]. This strategy relies on the assumption that such a positive-only subdomain exists within the dataset that depends partially on the dataset's attributes. This is a relaxed assumption compared to the separability assumption, which states that the positive and negative examples are separable without overlap. In TIcE, the dataset is randomly split into two parts. The first is used to create the decision trees and the second is used to estimate the label frequency. In order to find the pure-positive subsets, the *maximum biased-to-zero estimate for the proportion of positives* (*max-bepp*) score from [52] is used to score the splits [47]. A *bepp* score at each node is calculated by dividing the number of positive examples P by the total number of examples T seen by the node

$$\text{bepp} = \frac{P}{T+k}, \tag{1.25}$$

where k is the "to-zero" bias which prefers larger subsets; larger values of k brings the estimate closer to zero. Define the children of a node to be those nodes which immediately follow in the tree. For *max-bepp*, the split at each node whose children contain the maximum overall *bepp* score is selected. The label frequency c is converted to the class prior with $\hat{\alpha} = \frac{L}{(\hat{c}T)}$, where L is the number of labeled examples and T is the total. Algorithm performance is measured by the absolute error between $\hat{\alpha}$ and α. The three hyperparameters available for tuning this strategy are k (the max-bepp parameter), M (the maximum number of splits), and f (the number of tree/estimation folds).

1.2.6.3 KM2

If one examines the form of Equation 1.6, it becomes clear that the distribution $f_x = \Pr(x)$ resembles a mixture model with two components, where α is also known as the mixing

proportion within the mixture proportion estimation (MPE) literature [2]. The goal of MPE is to determine the ratio or weight of a component distribution contained within a mixture distribution by learning from a number of samples originating from the mixture and some number originating from the component [2, 49]. It uses the positive function assumption, which is a relaxed version of the positive subdomain assumption, see Subsection 1.2.5. MPE is used for anomaly detection and crowdsourcing, among other applications, and is the basis for the KM2 strategy [2]. It has good theoretical guarantees for convergence on the condition that the component distributions satisfy the positive function assumption, although its major weakness is that it is quite slow compared to, say, TIcE, and can only be run on a small subset of larger datasets [29, 47]. Indeed, the authors of the KM2 paper used a maximum of 3200 samples for evaluating its performance against other strategies [2].

Given a mixture distribution f_x comprised of component distributions f_{x_p} and f_{x_n}, we have

$$f_x = \alpha f_{x_p} + (1 - \alpha) f_{x_n}, \tag{1.26}$$

and

$$f_{x_n} = \lambda f_x + (1 - \lambda) f_{x_p}, \tag{1.27}$$

where $\lambda = \frac{1}{1-\alpha}$.

KM2, which notably does not require the computation of an estimate for the conditional probability, works by embedding the distributions into a reproducing kernel Hilbert space (RKHS) in order to produce an accurate estimate for the mixture proportion (class prior) α [2]. The RHKS is a simpler functional space which allows for the evaluation of functions through the use of its reproducing kernel without having to explicitly compute high- to infinite-dimensional calculations in the feature space.

1.2.6.4 Two-Step Techniques

Two-step techniques are so-called because they generally consist of two distinct steps and an optional third step:

(1) Identify reliable likely negatives (and positives) within the unlabeled distribution. The goal is to augment the known positives with a set of highly likely negatives.

(2) Use some (semi-) supervised learning technique such as SVMs, Naive Bayes, or expectation-maximization (EM), to learn the classifier to separate the positive and negative distributions.

(3) Select the best classifier based on some criteria.

Two-step techniques rely on the separability and smoothness assumptions on the data. A bit more detail on each step is given below. For an overview and comparison of many two-step techniques, see [29].

Step 1: Identify reliable negatives (and positives) The first step in the two-step approach is to identify reliable negatives within the unlabeled distribution by using some distance measure to select examples which are very different from the labeled positive examples. This is achieved by methods such as k-means clustering [53], k-nearest neighbors [54], probabilistic gap [45], and embedding *spies* from the positive distribution into the negative examples [55], along with other techniques, many of which originated in the text classification literature [29].

Step 2: Use some (semi-) supervised learning technique Once the reliable negatives are extracted from the dataset, a supervised or semi-supervised classifier is trained using the labeled positives and the reliable negatives. Any supervised classifier can be used, and the examples given in [29] include SVMs or Naive Bayes classifiers in addition to custom

iterative algorithms or distance-based approaches. EM can be used to find the best classifier in a semi-supervised learning scheme [29].

Step 3: Select the best classifier if applicable If an iterative or semi-supervised EM approach is taken, a third step is needed to select the best classifier using some criteria such as voting [56], change in F_1 score [57] or probability of error [55].

1.2.6.5 SAR-EM

When the assumptions on the selection or labeling mechanism are relaxed from the SCAR to the SAR assumption, EM can be applied to learn the classifier model and the propensity model in tandem by jointly maximizing their expected log likelihood [1].

In SAR-EM, the propensity scores are used to reweight the data in a negative weighting scheme, so an estimator which can appropriately handle negative sample weights must be used for propensity score learning unless this scheme or the estimator is modified to account for this. Unfortunately, XGBoost is not well-suited to deal with negative sample weights, so for the results in this thesis, SAR-EM is only applied to the output of of an LR model. In summary, given the estimated propensity score e_i for each data point x_i, every labeled positive example receives a weight $\frac{1}{e_i}$, and negative examples with weights $1 - \frac{1}{e_i}$ are added in tandem to match the labeled example count. From the definition of SAR, the propensity score model is learned from a subset of attributes called the *propensity attributes*.

The EM equations are given in 1.28 and 1.29. We are interested in finding the correct prediction for the expected probability $\hat{y}_i$ that example x_i and label s_i belong to the positive class. To do so, we apply the expected classification and propensity score models $\hat{f}$ and $\hat{e}$

Expectation step

$$\begin{aligned}\hat{y}_i &= \Pr(y_i = 1 | s_i, x_i, \hat{f}, \hat{e}) \\ &= s_i + (1 - s_i) \frac{\hat{f}(x_i)(1 - \hat{e}(x_i))}{1 - \hat{f}(x_i)\hat{e}(x_i)}.\end{aligned} \tag{1.28}$$

Next, the expected probabilities $\hat{y}_i$ are used to jointly optimize the expected log likelihood of f and e, which are the models to be learned and to be fed back into the expectation step

Maximization step

$$\begin{aligned}&\text{argmax}_{f,e} \sum_{i=1}^{n} \mathbb{E}_{y_i|x_i,s_i,\hat{f},\hat{e}} \ln \Pr(x_i, s_i, y_i | f, e) \\ &\quad = \text{argmax}_f \sum_{i=1}^{n} \left[\hat{y}_i \ln f(x_i) + (1 - \hat{y}_i) \ln(1 - f(x_i))\right], \\ &\quad\quad \text{argmax}_e \sum_{i=1}^{n} \hat{y}_i \left[s_i \ln e(x_i) + (1 - s_i) \ln(1 - e(x_i))\right].\end{aligned} \tag{1.29}$$

Taking a closer look at Equation 1.29, we optimize the classification model f via the following

$$\text{argmax}_f \sum_{i=1}^{n} \left[\hat{y}_i \ln f(x_i) + (1 - \hat{y}_i) \ln(1 - f(x_i))\right],$$

where f receives two weighted versions of each example x_i. The first term, $\hat{y}_i \ln f(x_i)$ indicates the positively-weighted example, with $\hat{y}_i$ being the expected probability of x_i being positive. The second term, $(1-\hat{y}_i)\ln(1-f(x_i))$ indicates the negatively-weighted example, with $(1-\hat{y}_i)$ being the expected probability of x_i being negative.

The propensity model e is optimized via

$$\text{argmax}_e \sum_{i=1}^{n} \hat{y}_i \left[s_i \ln e(x_i) + (1 - s_i) \ln(1 - e(x_i))\right],$$

where e receives one weighted example with $\hat{y}_i$ as its weight. Inside the brackets, $s_i \ln e(x_i) + (1 - s_i) \ln(1 - e(x_i))$ reduces to $\ln e(x_i)$ in the positive case where $s = 1$, and $\ln(1 - e(x_i))$ otherwise.

Unfortunately, if the propensity score depends on the same attributes as the classifier model, it is unidentifiable whether an unlabeled example is unlabeled due to either the propensity score or the class probability being low without making further assumptions on the propensity score. This is the main weakness of this approach and, as [58] shows, in this scenario many possible propensity score/posterior class probability combinations can yield the same observed data.

1.2.6.6 DEDPUL: Difference-of-Estimated-Densities-based Positive-Unlabeled Learning

The main insight gained from [59] is that the process of obtaining predictions via an NTC $g(x)$ is both a prior and a posterior-preserving transformation. The relationship between unlabeled and positive distributions and the posterior via the NTC is formally defined in Equation 1.18 [38]. To cope with the unidentifiability of α, DEDPUL [59] predicts α^*, which is the upper limit on α, given by $\alpha^* \equiv \inf_{x \sim f_{y_u}(y)} \frac{f_{y_u}(y)}{f_{y_p}(y)}$

(Biased) estimate of prior from NTC

$$\alpha^* = \inf_{x \sim f_{x_u}(x)} \frac{1 - g(x)}{g(x)}. \tag{1.30}$$

(Biased) estimate of posterior from NTC

$$p_p^*(x) = \alpha^* \frac{g(x)}{1 - g(x)}. \tag{1.31}$$

These estimates can be obtained directly from the NTC predictions as in [38]. Instead, [59] uses these predictions as a biased estimate of the posterior probability that a sample is positive. From this biased estimate, the PDFs for the positive and unlabeled samples are estimated. Bayes' rule is then applied and EM is used to find the unbiased estimates for the prior and posterior probabilities

Expectation step: unbiased estimate of posterior

$$\tilde{p}_p(y) = \min\left(\tilde{\alpha}\frac{\tilde{f}_{y_p}(y)}{\tilde{f}_{y_u}(y)}, 1\right). \tag{1.32}$$

Maximization step: unbiased estimate of prior

$$\tilde{\alpha} = \frac{1}{|Y_u|}\sum_{y\in Y_u}(\tilde{p}_p(y)). \tag{1.33}$$

An alternate method of estimating α^* is used if the EM algorithm converges trivially. This alternate method uses the maximum slope of the difference $D(\tilde{\alpha})$, which is defined as

$$D(\tilde{\alpha}) = \tilde{\alpha} - \frac{1}{|Y_u|}\sum_{y\in Y_u}(\tilde{p}_p(y)), \tag{1.34}$$

where $\tilde{p}_p$ is obtained from

$$\tilde{p}_p(y) = \min(\tilde{\alpha} * \tilde{r}(y), 1), \tag{1.35}$$

and $\tilde{r}(y)$ is the sorted array of density ratios after the application of smoothing heuristics (monotonization and rolling median),

$$\tilde{r}(y) = \left\{\frac{f_{y_p}(y)}{f_{y_u}(y)}, y \in \tilde{Y_u}\right\}. \tag{1.36}$$

For the experiment in [59], the mixing proportion (class prior) α is varied in $\alpha \in$

$\{0.05, 0.25, 0.50, 0.75, 0.95\}$, while $|X_p|$ is held constant by adding unlabeled examples to increase $|X_u|$. This lowest tested class prior of 0.05 is far from the known alpha of the USGS 2008 dataset, which is ≈ 0.0014 [5].

1.2.6.7 Confident Learning

A method that relies more on cleaning up existing data rather than modifying classification strategies is confident learning [40]. In confident learning, it is assumed that the target variable is noisy and results from a class-conditional process which gives the joint distribution of positive and unlabeled examples. A technique called rank pruning is used which employs k-nearest neighbors to find likely positives and negatives, and uses probabilistic thresholds and ranking to create a sense of confidence in the labels available for training and testing. A preliminary application of confident learning techniques to the 2008 USGS dataset found (out of the total 725,442 data points) 88,695 near-duplicate data points, 42,031 outlier data points (data points which may be out-of-distribution or anomalous), and from 125 to 170 potentially mislabeled data points. Duplicate and near-duplicate data points may indicate inconsistencies in the dataset and can potentially adversely impact the performance of classifiers trained on this dataset. More investigation is needed to determine the effect of pruning these potentially problematic data points, but is beyond the scope of this thesis.

1.2.7 A Note on Propensity Score Estimation

As presented in [58], an alternate formulation for learning the propensity score involves maximizing the posterior probability of the label indicator in one of two scenarios:

- The propensity score can follow any arbitrary function but the positive and negative distributions cannot overlap (*local certainty scenario*) or
- The propensity score must be a monotonic decreasing function of $\Pr(y = 1|\mathbf{x})$ (*probabilistic gap scenario*).

Table 1.1: Comparison of attributes between state-of-the-art PU learning approaches. As of the writing of this thesis, SAR-EM [1] is the only general PU learning algorithm based on the SAR assumption rather than the more restrictive SCAR assumption. KM2 [2] is the algorithm with the least support for large datasets, as explained in Section 1.2.6.3. nnPU [3] and AlphaMax [4] are not evaluated in this thesis, but are included in this table to give the reader a more general overview of the PU learning landscape.

Algorithm	SCAR/SAR	MPE	Estimates α	Handles large datasets
EN [38]	SCAR	Yes	Yes	Yes
TIcE [47]	SCAR	Yes	Yes	Yes
KM2 [2]	SCAR	Yes	Yes	No
SAR-EM [1]	SAR	Yes	Yes	Yes
nnPU [3]	SCAR	No	No	Yes
AlphaMax [4]	SCAR	Yes	Yes	Yes
DEDPUL [59]	SCAR	Yes	Yes	Yes

For the local certainty scenario, the propensity score e can be described by the following relationship

$$e = \frac{\Pr(s=1)\Pr(\mathbf{x}|s=1)}{\Pr(y=1)\Pr(\mathbf{x}|y=1)}. \tag{1.37}$$

However, since one cannot sample from $\Pr(\mathbf{x}|y=1)$ and an estimate for $\Pr(y=1)$ cannot be obtained, an approximate substitution must be made [58].

Let e^* be a modified estimate for the true propensity score, e. Since e is undefined for negative examples, a substitute estimate e^* is

$$e^* = \frac{\Pr(s=1)\Pr(\mathbf{x}|s=1)}{\Pr(\mathbf{x})}, \tag{1.38}$$

where e^* is equal to 0 for all negative examples. $\Pr(s=1)$ is calculated as the ratio of labeled to unlabeled data. The ratio $\frac{\Pr(\mathbf{x}|s=1)}{\Pr(\mathbf{x})}$ can be estimated by training an NTC to separate labeled vs. unlabeled data and using the output [58].

For the probabilistic gap scenario, an NTC can be trained in the same manner to distinguish labeled vs. unlabeled data. Then the propensity score can be related to the output

of this classifier $h(x)$

$$e = \sqrt{\sup_{\mathbf{x} \sim \mathcal{X}} [h(x)] h(x)}. \tag{1.39}$$

Both of these methods result in propensity score estimations which are identifiable and close to the true propensity score, claims which [1] does not make [58].

1.2.8 Evaluating PU classifiers

The most popular metrics in the general machine learning literature for evaluating classifiers in the supervised scenario include the accuracy, error rate, precision, recall, and F_1 score [27, 29]. These are calculated from the number of true positive predictions (TP), true negative predictions (TN), false positive predictions (FP), and false negative predictions (FN) when comparing the classifier's predictions to the ground truth labels.

Traditional evaluation of a classifier requires a fully labeled sample from the true distribution. It is possible to simulate PU data for which the true labels are known, either in a fully-synthetic manner by drawing from known distributions to generate the data, or in a semi-synthetic manner by adapting existing fully labeled datasets and hiding some proportion of the labels [27]. One can then use this synthetic data to evaluate PU classification algorithms. Despite the utility of synthetic datasets in comparing PU learning algorithms, such a fully labeled sample is not available in real world PU learning scenarios, so an NTC must be evaluated in a non-traditional manner, that is, by using the unlabeled data as a surrogate for labeled negative examples.

The standard cardinal metrics such as TN and FN cannot be determined. Since these are necessary to calculate the accuracy and precision, neither of these metrics, nor derivatives of them can be calculated directly. If the classifier outputs a positive classification for an unlabeled data point, it is unknown whether this is due to a poorly-fit model or to mislabeled data (an unlabeled data point that should actually be considered positive). These most popular supervised machine learning metrics are summarized in Section 1.2.8.1.

For some optimality criteria such as the *receiver operating characteristic* (ROC) curve, *precision-recall* (PR) curve, or *balanced error* (BER), under traditional evaluation, an NTC can achieve similar performance as an optimal traditional classifier [50, 60]. This is true because in many scenarios, these criteria rely on the optimal ranking of data points rather than the prediction scores themselves [4, 8, 51]. As long as the ranking is unchanged, the NTC optimized on these criteria is equivalent to the traditional classifier optimized on the same criteria with a fully labeled sample.

Despite this surprising result, evaluating a classifier non-traditionally does add bias to metrics such as the ROC *area under the curve* (ROC-AUC), precision, recall, accuracy, and F_1 score. Because unlabeled examples are likely to contain unidentified positives, the ROC-AUC estimated in a non-traditional manner by assuming all unlabeled examples are negative is necessarily lower than the true ROC-AUC resulting from classification in a traditional setting. This bias can be removed by incorporating knowledge about the class prior and noise proportion, where the noise proportion is the ratio of mislabeled positives to correctly-labeled positives within the training set [8]. The assumption made for the dataset used in this thesis is that its noise proportion is zero.

Standard supervised learning uses a train-validate-test pipeline to train a classifier on labeled data which can then be used in an inductive manner to predict on unlabeled data. With PU learning, a classifier can be trained in either an inductive or transductive manner. An inductive PU classifier is trained with labeled positive and unlabeled data and then generalized to new unlabeled data. A transductive PU classifier is trained with labeled and unlabeled data but cannot be used to generalize to new data; the data to be labeled is included in the training set.

Jaskie and Spanias [27] differentiate between evaluating PU algorithms and evaluating PU models. Evaluating PU algorithms involves comparing algorithms to one another via benchmarking on simulated PU data. Evaluating PU models is concerned with a PU

algorithm's performance on a particular, real PU dataset and must be done differently since the traditional metrics cannot be calculated. This thesis is mainly concerned with evaluation of PU models on the 2008 USGS dataset used in [5] and [6].

1.2.8.1 Standard evaluation metrics

Accuracy

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.40}$$

Error rate

$$error\,rate = 1 - accuracy = \frac{FP + FN}{TP + FP + TN + FN} \tag{1.41}$$

Precision

$$precision = \frac{TP}{TP + FP} \tag{1.42}$$

Recall

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{1.43}$$

F_1 score

$$F_1 = 2 \times \frac{(precision \times recall)}{(precision + recall)} = \frac{2\,TP}{2\,TP + FP + FN} \tag{1.44}$$

The F_1 score is the most widely-used metric in the PU learning literature, in addition to being the most appropriate metric for imbalanced PU learning problems out of the standard four (the other three being precision, recall, accuracy) [29, 61]. Mordensky et al. [5] use the models' F_1 scores for hyperparameter tuning and model evaluation. When used with PU data, the F_1 score will incorrectly penalize unlabeled samples if they are predicted positive when indeed they should be considered positive. Thus, the F_1 score will always underestimate the classifier's true performance and may fail to differentiate a classifier which does a better job at predicting unlabeled positive samples.

ROC curve and ROC-AUC score The ROC curve shows a model's true positive rate versus false positive rate as the decision boundary is varied. The ROC-AUC score reduces the ROC curve to a single number, where a higher score indicates a better classifier, but does not account for imbalanced data. Alternatives for when the classes are severely imbalanced are the PR curve and its associated area under the curve (PR-AUC) and *average precision score* (an alternative formulation to the PR-AUC) [62].

Precision-recall (PR) curve PR curves show a model's precision as a function of its recall and, compared to ROC curves, allow for better comparison when class sizes are heavily imbalanced [63].

Matthews correlation coefficient (MCC)

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{((TP + FP)(TP + FN)(TN + FP)(TN + FN))}} \tag{1.45}$$

The Matthews correlation coefficient (MCC) is symmetric in the way it treats positive and negative classes with equal importance. It is also robust to imbalanced class sizes, making it a useful metric for many datasets [27]. Ramola et al. [64] derive an unbiased estimate for the MCC to be used on PU data when one does not have access to the true labels for all positives and negatives.

1.2.8.2 Metrics for evaluating PU learning algorithms

Because the F_1 score grows as both the precision and recall increase, [7] recognized that an analog to this score can be constructed from just the recall and the estimated class prior taken as the percentage of labeled positive examples in $\mathbf{X}$, $\Pr(\hat{y} = 1)$

PU-estimated F_1 score (PUF-score)

$$PUF\text{-}score = \frac{recall^2}{\Pr(\hat{y} = 1)}, \tag{1.46}$$

where the recall can be estimated from the known labeled positives P_L and the number of correct positive predictions TP_L

$$\widehat{recall} = \frac{TP_L}{P_L} = \frac{\sum_{s=1} \hat{y}}{\sum_{s=1} s}. \tag{1.47}$$

Jain et al. [8] derive unbiased estimates of the true positive rate γ, false positive rate η, precision ρ and ROC-AUC. They begin from definitions for γ and η that are given by the expectation $\mathbb{E}$ with respect to the positive and negative distributions P, N over some classifier $h : \mathcal{X} \to \mathcal{Y}$

True positive rate/recall

$$\gamma = \mathbb{E}_P[f(\mathbf{x})], \tag{1.48}$$

False positive rate

$$\eta = \mathbb{E}_N[f(\mathbf{x})]. \tag{1.49}$$

The unbiased estimate of γ is simply the empirical mean of the classifier $h(x)$ over P_L, which is the sample of labeled positives

Estimate of true positive rate, γ

$$\widehat{\gamma} = \frac{1}{|P_L|} \sum_{x \in P_L} f(\mathbf{x}). \tag{1.50}$$

Similarly, an intermediate estimate for η, $\hat{\eta}_{PU}$, can be taken by considering the naïve

empirical mean over the unlabeled samples U

Estimate of PU false positive rate, η_{PU}

$$\hat{\eta}_{PU} = \frac{1}{|U|} \sum_{x \in U} f(\mathbf{x}). \tag{1.51}$$

With $\hat{\eta}_{PU}$, $\hat{\gamma}$, and an estimate $\hat{\alpha}$ for the class prior, Jain et al. [8] arrive at the unbiased estimate for η

Estimate of false positive rate, η

$$\hat{\eta} = \frac{\hat{\eta}_{PU} - \hat{\alpha}\hat{\gamma}}{1 - \hat{\alpha}}. \tag{1.52}$$

Using these variables, they also give an unbiased estimate for the precision ρ

Estimate of precision, ρ

$$\hat{\rho} = \frac{\hat{\alpha}\hat{\gamma}}{\hat{\eta}_{PU}}. \tag{1.53}$$

Finally, Jain et al. [8] provide the unbiased estimate for the ROC-AUC, given $\hat{\alpha}$ and the biased ROC-AUC, calculated from training a classifier to predict positive or unlabeled

Estimate of area under the ROC curve, AUC

$$\widehat{AUC} = \frac{AUC^{PU} - \frac{\hat{\alpha}}{2}}{1 - \hat{\alpha}}. \tag{1.54}$$

The results of these PU metrics on the naïve strategies examined in this thesis are included in Table 3.6.

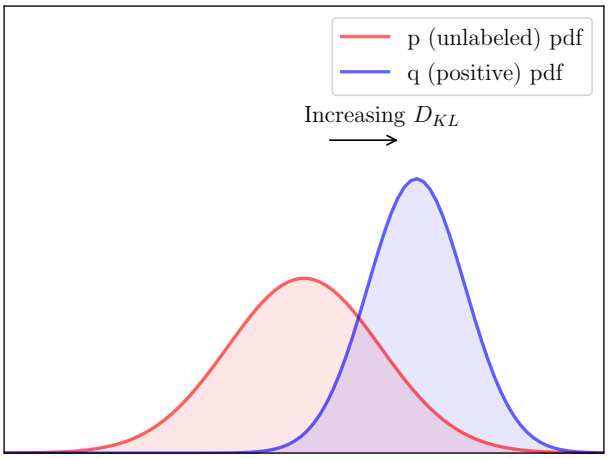


Figure 1.5: Illustration of D_{KL} between two univariate Gaussian distributions. As the mean of q (the estimated PDF of the positive instances) increases and/or its variance decreases, the KL divergence from p to q increases.

1.2.8.3 Alternative Evaluation Metrics

Kullback-Leibler divergence The Kullback-Leibler (KL) divergence, also called the relative entropy or I-divergence [65], allows for comparison between two arbitrary distributions P and Q without knowing the true posterior distribution. In the continuous case, for random variables P and Q, the KL divergence from P to Q is

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx. \tag{1.55}$$

The $D_{KL}(P||Q)$ has a lower bound at zero, which indicates that the two distributions are identical. In the positive direction it is unbounded and increases as the mean of Q increases and its variance decreases. It is not symmetric, i.e. $D_{KL}(P||Q) \neq D_{KL}(Q||P)$, and it does not satisfy the triangle equality [30]. For univariate Gaussian distributions, once maximum likelihood estimates are obtained for each distribution's mean and variance, the closed-form solution of Equation 1.55 exists using these estimates for $p \sim \mathcal{N}(\mu_p, \sigma_p)$ and $q \sim \mathcal{N}(\mu_q, \sigma_q)$ [66]

$$D_{KL}(p, q) = \log \frac{\sigma_q}{\sigma_p} + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2}. \tag{1.56}$$

We will henceforth use the abbreviation D_{KL} to refer to the KL divergence taken from P to Q, where P is the normal score transformed prior distribution of unlabeled samples and Q is the normal score transformed estimated posterior distribution of positive predictions. We apply the D_{KL} in the selection of hyperparameters by favoring classifiers which predict the Q that gives the maximum D_{KL}. By maximizing the D_{KL} instead of traditional supervised classification metrics such as the F_1 score or ROC-AUC, the goal is to select more appropriate hyperparameters when little is known about the true distribution from which we are sampling.

1.3 Contributions

1.3.1 The Use of the D_{KL} as an Evaluation Metric

Although the D_{KL} is often used in anomaly detection [31,32], partial distribution matching [44], and class prior estimation in unlabeled test data [33], where the goal is to minimize divergence between the PDFs of the training and test data, the maximization of the D_{KL} between the unlabeled distribution and the predicted positive distribution as a method of model evaluation for PU learning problems, presented in Chapter 2, is a novel application. This thesis demonstrates that the D_{KL} can make known positive predictions more positive. Despite this, without full knowledge of the true distribution, it is ultimately unknown whether the effect of the D_{KL} pushing the distributions apart is due to the predicted area containing true positive examples or if it is predicting a larger positive area than is appropriate.

1.3.2 Comparison of PU Learning Approaches

A preliminary comparison of PU learning approaches shows that the best classifier performance (F_1 score) is achieved by SAR-EM when its favorability classifier uses the optimal

hyperparameters found through F_1 score tuning. This method also produces a class prior estimate that is closest to the estimated natural class balance as reported in [5, 6]. On the other hand, using the optimized class weight hyperparameters both for LR and XGBoost, we show that the true class prior may be considerably underestimated by the average system power calculation provided by [5, 6]; the true positive class may contain many more data points than are labeled in the dataset.

2 Methodology

2.1 Experimental Design

The aim of this thesis is two-fold: to apply the D_{KL} as an evaluation metric in selecting hyperparameters for machine learning models training with PU data and to compare PU learning strategies, contrasting them with approaches that simply assume all unlabeled data as negative. One thing to keep in mind is that we are restricting our scope to evaluation on the data at hand; we do not evaluate each model on synthetic data that provides access to true labels, so we are forced to make conclusions that may be specific to our particular dataset.

In this thesis, we firstly train two classes of machine learning models consisting of one linear method and one nonlinear method: LR and XGBoost, respectively. The optimal hyperparameters for each are selected via grid search optimized on four separate measures of model success over test data which is withheld via subsampled stratified k-fold cross validation over 120 folds with a ratio of 80:20 train:test as per [5, 6]. The evaluation metrics used for optimization in each case are the F_1 score, recall, ROC-AUC, and the D_{KL}. These two classes of models, eight cases total, are trained as NTCs by assuming all unlabeled data points are negative; they will be referred to as the "naïve" methods and, in addition to reproducing the results from [5, 6], are intended to show the potential advantage of using the D_{KL} as a measure of model ability with PU data.

Secondly, we apply two state-of-the-art PU learning algorithms to the data, SAR-EM and DEDPUL, and include results from two other PU learning algorithms, TIcE and KM2,

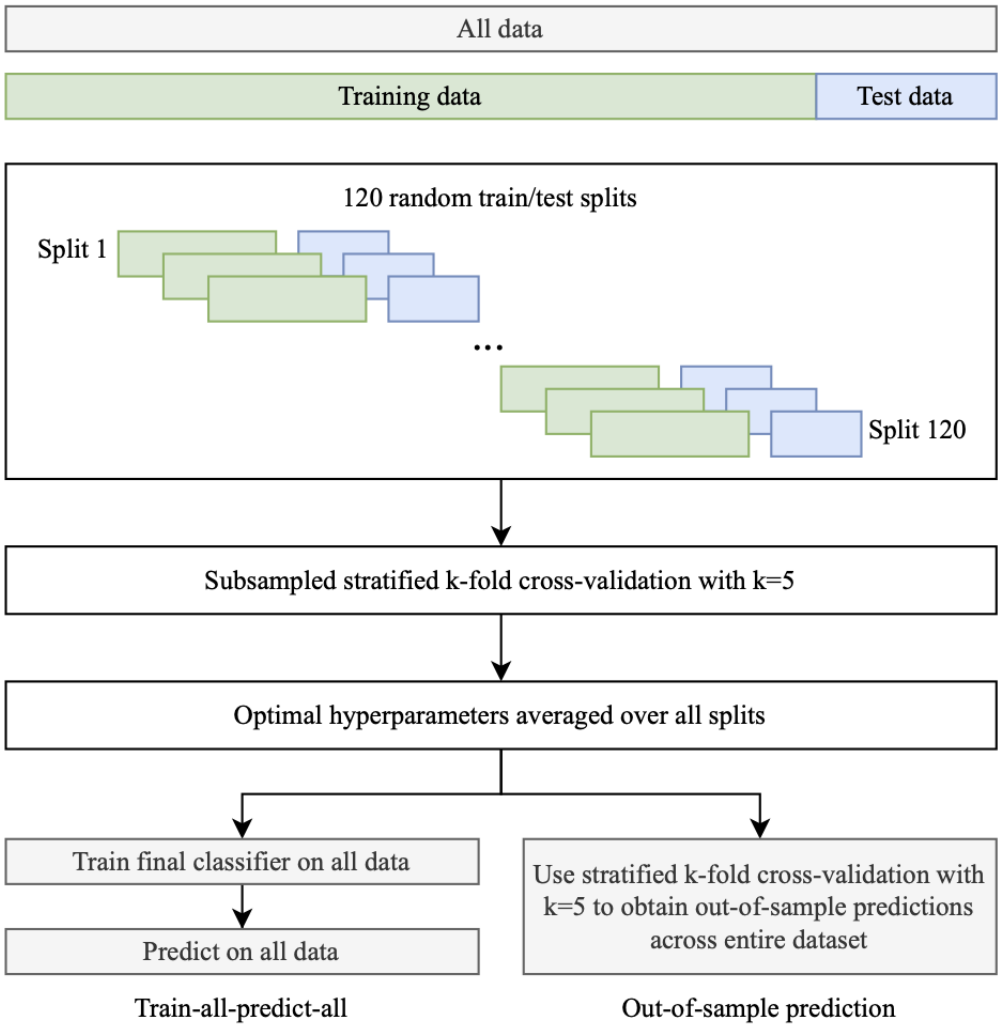


Figure 2.1: Overview of train-all-predict-all and out-of-sample prediction schemes. Both methods use the same strategy to arrive at the optimal hyperparameters; subsampling of the validation and test sets is carried out at this step to reflect the expert-opinion positive-negative ratio of 1:700. At the final step in the out-of-sample prediction scheme, no such subsampling is performed. This step is illustrated in Figure 2.2.

where TIcE is used for class label prediction and class prior estimation and KM2 is used only for class label prediction. Because all four of these PU learning strategies require the training of an NTC in their formulation, either LR or XGBoost are used, depending on their compatibility with that specific PU learning strategy. In particular, because SAR-EM uses negative example weights, and XGBoost does not support the use of negative example

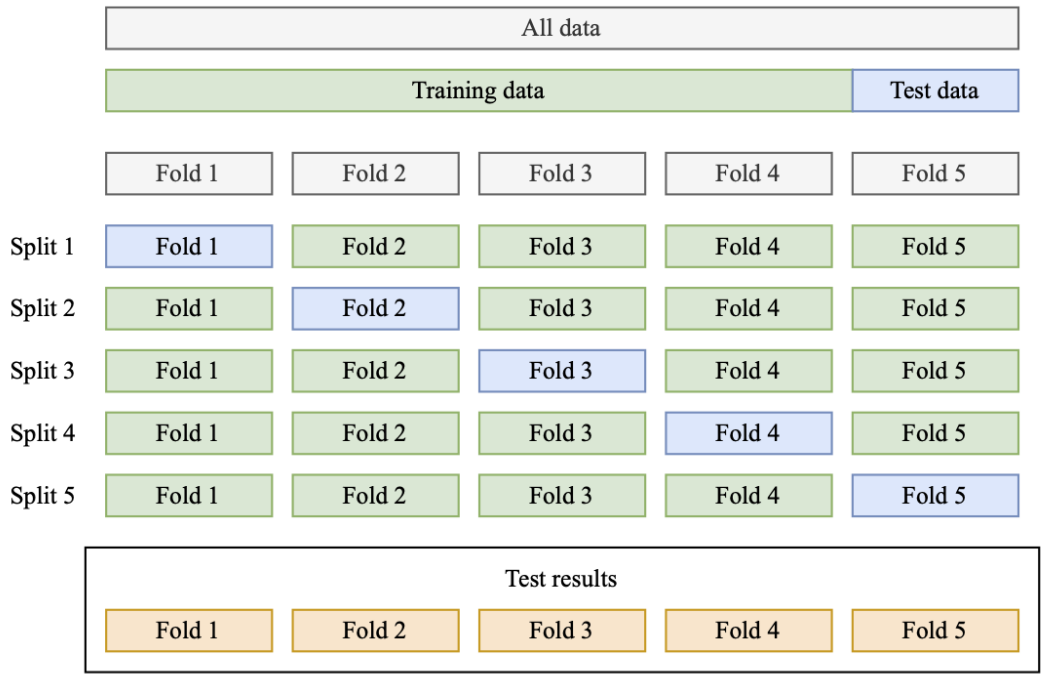


Figure 2.2: Obtaining out-of-sample test results via standard stratified k-fold cross-validation for $k = 5$. Stratification ensures coverage over the entire dataset. Figure adapted from scikit-learn documentation [10,11].

weights, LR is used as the sole class of NTC for SAR-EM. Three LR models are trained: one with default hyperparameters, and the other two using the hyperparameters optimized for either the F_1 score or D_{KL}.

Besides the test scores and predicted class prior for each model (except test scores for DEDPUL and KM2), we have two main tools for analysis at our disposal. The first are favorability maps generated by fusing location data with the models' normal score transformed predicted favorability for each cell. Circles within the favorability maps indicate cells with known geothermal energy sites. Secondly we have ridge and CDF plots of the predicted positive distribution shown alongside the unlabeled distribution for each model; these are also normal score transformed in each case. The ridge and CDF plots show the success of each model at separating the two distributions, even when the true class labels are unknown.

2.1.1 Data Features and Labels

For a comparison of the models created in [5] and [6], this thesis uses one of the highest-performing dataset combinations from 2008 which consists of the following five geophysical features:

- Heat flow: interpolated map of estimated vertical heat flow in milliwatts per square meter (mW/m^2)
- Fault distance: distance in meters to the nearest Quaternary fault
- Magma distance: distance in meters to the nearest magmatic activity
- Seismic density: density of epicenters for seismic events $\geq$ M3 within a 4 km radius in number of events per square kilometer (km^2)
- Stress: maximum horizontal stress in megapascals (MPa)

Coordinates were removed from the data prior to training and the data was shuffled so that trained models would not explicitly depend on location.

2.1.2 Applying the D_{KL} to Select Hyperparameters

The main test results in Table 3.4 were obtained by first generating 120 random train/test splits at a ratio of 80:20, using a random number generator with the same sequence of seeds as used by [5] and [6] for reproducibility. The test sets were subsampled to maintain the sample's original estimated target variable distribution: a positive-to-negative ratio of about 1:700 [5,6]. A hyperparameter grid search was performed using stratified k-fold cross-validation with five folds. The validation sets were subsampled similarly to the test sets, maintaining a 1:700 ratio. Subsampling the validation and test sets allows the ratio of positives to negatives to more accurately reflect the naturally occurring ratio. Because we are limited to traditional

evaluation of algorithms trained on PU data, subsampling at this stage allows us to remove a bit of the resulting bias seen in the test scores.

In order to investigate using the D_{KL} as an evaluation metric, the naïve models are evaluated by taking the maximum F_1 score, recall, ROC-AUC, or D_{KL} when scored on the validation set. The hyperparameters of the best-performing models are then averaged over all 120 train/test splits for each metric as reported in Table 3.4. Test scores are generated by training the final classifiers using the optimal average hyperparameters on all training data, then predicting on the test sets. The test scores are then averaged to give the final reported values.

The label frequency is estimated using the ratio of the labeled positives to the total predicted positive examples. The naïve estimate for the class prior is taken as the ratio of predicted positives to total examples. Experimental details specific to each algorithm follow.

2.1.2.1 Logistic Regression

Since its introduction in 1944 in [67], LR has become ubiquitous in the machine learning literature as a simple and effective solution to some linearly separable classification problems, in cases where more complexity is unnecessary, and as a baseline for comparing more complex machine learning algorithms. LR applies a logit transformation to the dependent variable Y in order to facilitate modeling a sigmoidal decision function. The natural log of the odds of y (ratio of probability of event y occurring to the event y not occurring) is taken:

$$\text{logit}(y) = \ln\left(\frac{\Pr(y=1|\mathbf{x})}{1-\Pr(y=1|\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n \tag{2.1}$$

where $x_1 \ldots x_n$ are the input features, and $\beta_0 \ldots \beta_n$ are the regression coefficients, fit either through maximum likelihood estimation or weighted least squares. Taking the antilog of both sides results in the probability $\Pr(y = 1|x_0, \ldots, x_n)$:

$$\Pr(y = 1|x_0, \ldots, x_n) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_n x_n}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_n x_n}} \tag{2.2}$$

The probability output of Equation 2.2 is thresholded to result in the classifier output. A hyperparameter search for LR is performed over the two most relevant hyperparameters in alignment with [5] and [6]: the class weight and the inverse regularization strength. These hyperparameters are reported in Table 3.2. The same threshold of 0.5 was also used. The test results for LR models are found in Tables 3.4, 3.5, and 3.1.

2.1.2.2 XGBoost

XGBoost is a learning algorithm that was first introduced in 2016 by [24]. It has since become a popular first choice for many data scientists for exploring the application of nonlinear models to more complex problems. It relies on *boosting* a weak learner (an archetypal simple learning model such as a decision tree). Many trees are arranged into an ensemble configuration which is learned through training. Each training round iteration passes the residual from the previous tree to the next tree, which can then improve upon the previous result based on some objective function. Once the model begins to overfit and no longer improves upon the previous iteration, the training is finished. The terminal node in each branch outputs a probability value and the average probability at every terminal node gives the final output.

Again, a similar setup to that used in [5] and [6] is used: a hyperparameter search over the maximum depth of estimators, number of estimators, class weight, and learning rate is performed. The resulting optimal hyperparameters are recorded in Table 3.3. The test results for XGBoost are also found in Tables 3.4, 3.5, and 3.1.

2.1.3 Comparing PU Learning Methods

2.1.3.1 SAR-EM

SAR-EM uses two LR classifiers: one to model the function separating positive/negative examples, and one to model the propensity score function. As implemented in [1], no hyperparameter optimization is performed on these final two classifiers. For comparison, three models are trained: one using the optimal hyperparameters from maximizing the F_1 score, one using the hyperparameters from maximizing the D_{KL}, and the results from running SAR-EM without any optimization. The test results for SAR-EM are found in Tables 3.4, 3.5, and 3.1.

2.1.3.2 DEDPUL

The DEDPUL algorithm is designed to obtain transductive predictions of the unlabeled set only. Inductive predictions over new data can be obtained from a trained model, however, through interpolation. The NTC can be applied to new data, the likelihood ratio can be predicted from mapping the new $y(x)$ to the interpolated $r(y)$, then predictions are found by multiplying by α^* and clipping to $[0, 1]$. For the purposes of this thesis, DEDPUL is simply modified such that $\tilde{p}_p(y)$ is obtained for all points rather than just the unlabeled set. DEDPUL test results are recorded in Table 3.1.

2.2 Model Evaluation

In the creation of supervised classification models, an inductive learning approach is taken [27]. In inductive learning, the goal is to create a generalized model by training on fully labeled data that can then be expected to achieve similar performance on unlabeled data. An effective inductive learning pipeline splits the available labeled data into randomized training, validation, and test sets without leakage between the sets. Increasing performance on the

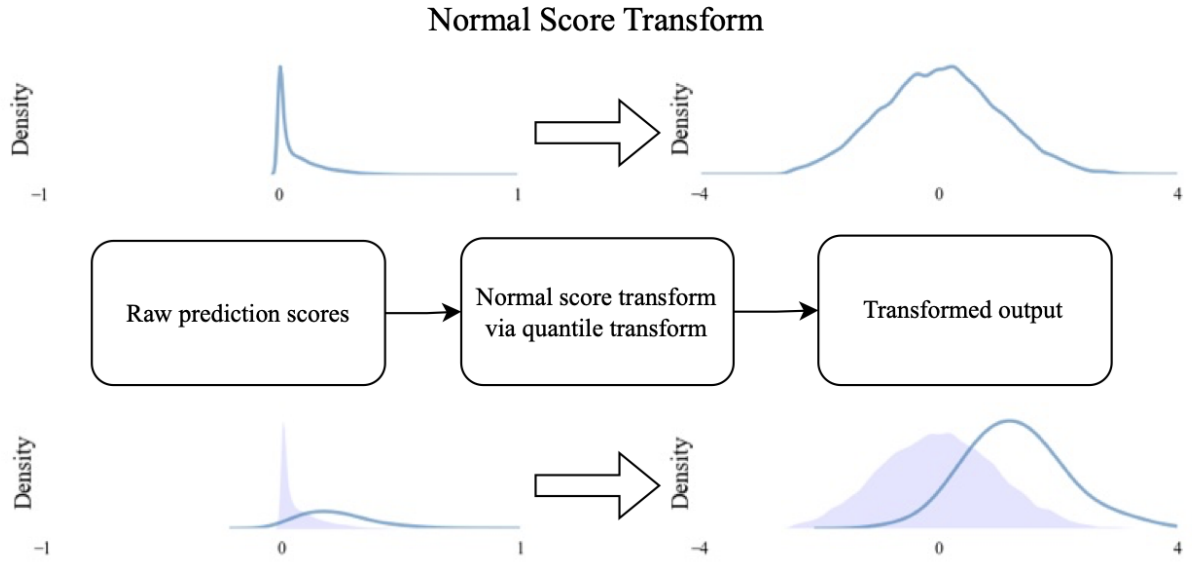


Figure 2.3: Diagram showing the normal score transform pipeline for prediction scores and how this transform affects the resulting PDF ridge plot. Top of diagram shows the PDF for the combined distribution (including positive and unlabeled data), and bottom of diagram shows the PDFs for the separated positive and unlabeled sets.

test set then makes it reasonably likely that the model will perform well on new, unseen data. This contrasts with the goal of transductive learning, which is interested in optimizing performance on a particular dataset. Models created through transductive learning take the same data as their training and test sets and do not generalize well to new data. Under the structural risk minimization (SRM) principle, inductive learning tries to estimate a function over its entire domain and transductive learning tries to estimate only values of a function at specific points in a discrete space [68, 69].

As is the case for other semi-supervised learning models, some PU learning models can be used inductively and some are purely transductive. DEDPUL is an example of an algorithm that is designed to be used for transductive inference, but can be applied in an inductive manner through interpolation on new data [59]. In this thesis, hyperparameters are optimized in a pipeline designed to perform inductive inference, that is, they are tested on new data which is unobserved at training. Once the optimal hyperparameters are found via averaging

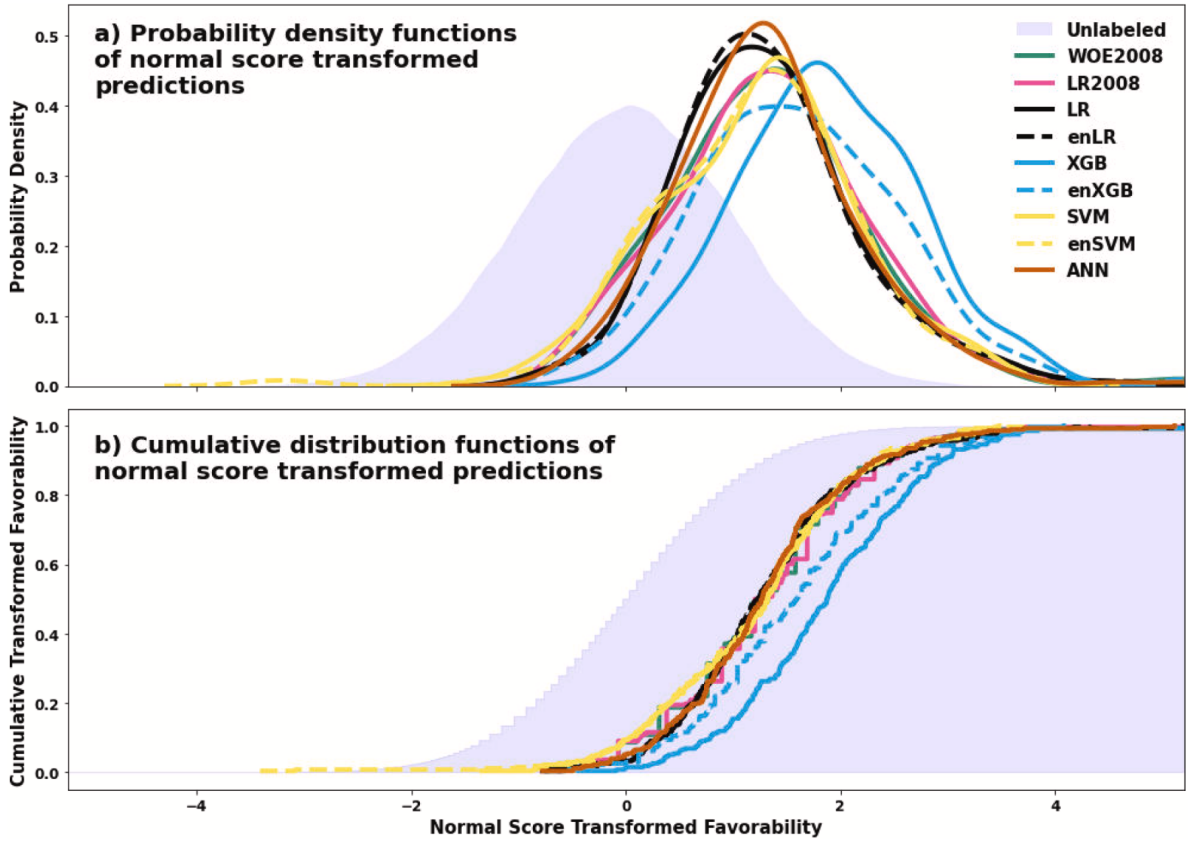


Figure 2.4: Example ridge (top) and CDF (bottom) plots from Mordensky et al. [6].

over all splits, they are used to train a new model on all training data (combined training and validation sets) and test scores are averaged from the results of predicting on each test set of the 120 total train/test splits.

Ridge plots such as that in Figure 2.4, taken from [6], show an estimate of the PDF and cumulative distribution function (CDF) of the output of each algorithm. These estimates are made via kernel density estimation methods on the normal score transformed output. An illustration of the normal score transformation is shown in Figure 2.3. For Figures 3.9 and 3.10, the two prediction schemes, train-all-predict-all (transductive) and out-of-sample prediction (inductive), were employed to observe the behavior of the naïve models in each scenario respectively. For the rest of the ridge plots and favorability maps, the train-all-predict-all

scheme was used.

Raw predictions are normal score transformed via quantile transform, with the quantile function for the normal distribution defined as the inverse of the CDF

$$Q(\mathbf{y}) = \Phi^{-1}(\mathbf{y}), \tag{2.3}$$

where Φ is the CDF of the normal distribution

$$\Phi(\mathbf{y}) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{\mathbf{y} - \mu}{\sigma\sqrt{2}}\right)\right], \tag{2.4}$$

with a mean of μ and variance of σ^2. This has the following PDF

$$\phi(\mathbf{y}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mathbf{y}-\mu)^2}{2\sigma^2}}. \tag{2.5}$$

3 Results

3.1 Identifying the Class Prior

Many PU learning methods rely on accurately identifying the class prior. Table 3.1 shows the test results for class prior prediction. Compared to the estimated expert opinion-based positive class prior of 1:700 or about 1.43e-3 positives per every one negative example, SAR-EM achieves the closest class prior prediction with a MAE of 1.01e-4.

3.2 Optimal Hyperparameters

The optimal hyperparameters found through a grid search for each algorithm are provided in Tables 3.2 and 3.3. To facilitate the grid search, 120 train/test splits are taken, from which five subsampled stratified folds are used to create the train and validation sets for each split. Each hyperparameter is taken as the average across all splits. Each real number is rounded to two decimal places after the zero and hyperparameters that accept natural numbers are rounded up. When XGBoost is tuned with the D_{KL}, it benefits from a slightly faster learning rate, but requires more branches and more estimators to achieve the maximum score, as seen in Table 3.3.

Table 3.1: Estimates for label frequency c and class prior α for SAR-EM using logistic regression (LR) with F_1-tuned (F1) and D_{KL}-tuned (KL) and DEDPUL using XGB classifier models. Ground truth for class prior estimation is the expert opinion-based class prior of ≈ 0.0014 (1.40×10^{-3}) [5, 6]. Each case is abbreviated within the table as [*NTC model*]+[*source of hyperparameters*], e.g. LR+F1 is logistic regression tuned via F_1 score. Bold text indicates the best result.

Strategy	Est. label frequency	Est. class prior	Avg. class prior MAE
Naive/negative approach			
LR+F1	9.56×10^{-2}	4.01×10^{-3}	2.57×10^{-3}
LR+KL	3.91×10^{-3}	9.81×10^{-2}	9.67×10^{-2}
XGB+F1	7.9×10^{-2}	4.85×10^{-3}	3.42×10^{-3}
XGB+KL	7.24×10^{-2}	5.29×10^{-3}	3.86×10^{-3}
SCAR-TIcE			
LR+Default	3.23×10^{-4}	3.93×10^{-6}	1.43×10^{-3}
LR+F1	3.23×10^{-4}	1.00	9.99×10^{-1}
LR+KL	3.23×10^{-4}	1.00	9.99×10^{-1}
SCAR-KM2			
LR+Default	8.95×10^{-2}	2.18×10^{-2}	2.03×10^{-2}
SAR-EM			
LR+F1	1.04×10^{-3}	1.53×10^{-3}	$\mathbf{1.01 \times 10^{-4}}$
LR+KL	1.78×10^{-3}	1.39×10^{-5}	1.41×10^{-3}
LR+Default	9.66×10^{-4}	2.39×10^{-2}	2.25×10^{-2}
DEDPUL			
XGB+F1	3.91×10^{-4}	2.00×10^{-2}	1.86×10^{-2}
XGB+Recall	3.91×10^{-4}	2.00×10^{-2}	1.86×10^{-2}
XGB+ROC-AUC	3.95×10^{-4}	2.92×10^{-2}	2.78×10^{-2}
XGB+KL	3.91×10^{-4}	2.00×10^{-2}	1.86×10^{-2}

Table 3.2: Optimal hyperparameters for logistic regression per tuning metric used.

Metric	C	Class weight
F1	6000.05	248.32
Recall	14375.00	995.62
ROC-AUC	770.78	217.93
KL	5181.27	10.10

Table 3.3: Optimal hyperparameters for XGBoost per tuning metric used.

Metric	Learning rate	Max. depth	Num. estimators	Scale pos. weight
F1	0.23	3	61	205.92
Recall	0.55	3	59	224.00
ROC-AUC	0.17	2	66	192.08
KL	0.58	4	77	189.67

3.3 Model Performance

The results from comparing LR and XGBoost, tuned by maximizing the F_1 score, recall, ROC-AUC, or D_{KL} are summarized in Table 3.4. Also in this table are the test results for SCAR-TIcE and SAR-EM, tuned by maximizing either the F_1 score or D_{KL}. The best F_1 score performance is achieved by the SAR-EM method using a classifier to predict $\hat{y}$ that is tuned by maximizing the F_1 score. The classifier to predict $\hat{e}$ uses the default hyperparameters.

Table 3.5 shows the test results when results over all data are obtained in a stratified out-of-sample prediction scheme. This is meant to mimic the classifier's performance in an inductive inference scheme. The test set is not subsampled to maintain the natural class balance of 1:700 in this configuration. Instead, five models are trained, each on a different

subset of the data and used to make predictions on the remaining data so that out-of-sample predictions are obtained across the entire dataset. Interestingly, LR tuned to the ROC-AUC achieves the highest F_1 score in this scheme. Also a bit surprisingly, there is much less variation between the different strategies when the test predictions are created out-of-sample. It is much more difficult to pick a clear winner based on the out-of-sample predictions.

Table 3.4: Mean and std. dev. test scores over all 120 train/test splits. Model trained for inductive inference. Each case is abbreviated within the table as [*NTC model*]+[*source of hyperparameters*], e.g. LR+F1 is logistic regression tuned via F_1 score. Bold text indicates the highest score for each metric.

	F1		Recall		ROC-AUC		KL	
Strategy	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Naïve/negative approach								
LR+F1	0.035	0.022	0.036	0.027	0.517	0.013	2.125	1.227
LR+Recall	0.014	0.002	0.438	0.057	0.677	0.028	2.037	1.165
LR+ROC-AUC	0.019	0.023	0.074	0.142	0.531	0.058	2.766	1.845
LR+KL	0.005	0.012	0.002	0.006	0.501	0.003	3.981	2.405
XGB+F1	0.025	0.016	0.034	0.023	0.516	0.011	2.810	1.604
XGB+Recall	0.024	0.011	0.068	0.031	0.531	0.016	18.234	24.830
XGB+ROC-AUC	0.024	0.019	0.028	0.025	0.513	0.012	2.888	1.755
XGB+KL	0.015	0.010	0.037	0.027	0.516	0.013	**28.051**	41.127
SCAR-TIcE								
LR+Default	0.008	0.000	**0.840**	0.050	**0.846**	0.019	2.0247	1.147
LR+F1	0.028	0.024	0.032	0.027	0.815	0.028	2.708	1.780
LR+KL	0.007	0.014	0.008	0.028	0.792	0.024	8.23	8.94
SAR-EM								
LR+Default	0.020	0.006	0.174	0.060	0.575	0.029	1.846	1.176
LR+F1	**0.042**	0.023	0.042	0.024	0.520	0.012	2.149	1.273
LR+KL	0.005	0.012	0.002	0.006	0.501	0.003	3.840	2.356

Table 3.5: Out-of-sample test scores (inductive inference) over all data for k=5 folds. Each case is abbreviated within the table as [*NTC model*]+[*source of hyperparameters*], e.g. LR+F1 is logistic regression tuned via F_1 score. Bold text indicates the highest score for each metric.

Strategy	F1	Recall	ROC-AUC	KL
LR+F1	0.013	0.054	0.843	1.422
LR+Recall	0.003	**0.507**	0.843	1.442
LR+ROC-AUC	**0.016**	0.043	0.843	1.493
LR+KL	0.007	0.004	0.843	2.341
XGB+F1	0.010	0.068	0.848	1.923
XGB+Recall	0.003	**0.507**	0.843	1.418
XGB+ROC-AUC	0.012	0.050	**0.854**	2.328
XGB+KL	0.004	0.040	0.803	**9.158**

The adapted (unbiased) PU scores from [7,8] for the naïve methods are shown in Table 3.6. The equations for these scores are given in 1.2.8.2. The purpose of including this table is to show that using these unbiased PU metrics can result in a different learning strategy appearing more attractive if, for instance, the recall were favored. Interestingly, LR tuned by maximizing the recall is ranked the best by four of the six PU metrics included. LR tuned by maximizing the D_{KL} achieves a $\widehat{Precision}$ that is an order of magnitude greater than the other naïve methods.

Table 3.6: Results of PU metrics, where PUF [7], $\widehat{TPR}$, $\widehat{FPR_{PU}}$, $\widehat{FPR}$, $\widehat{Precision}$, and $\widehat{AUC}$ [8] are outlined in Section 1.2.8.2. Scores result from models trained for inductive inference. Each case is abbreviated within the table as [*NTC model*]+[*source of hyperparameters*], e.g. LR+F1 is logistic regression tuned via F_1 score. Bold text indicates the highest score for each metric.

Strategy	PUF	$\widehat{TPR}$	$\widehat{FPR_{PU}}$	$\widehat{FPR}$	$\widehat{Precision}$	$\widehat{AUC}$
LR+F1	7.597	0.054	0.003	0.003	0.007	0.842
LR+Recall	**700.154**	**0.518**	**0.111**	**0.110**	0.002	0.842
LR+ROC-AUC	5.706	0.047	0.002	0.002	0.010	0.842
LR+KL	0.034	0.004	0.000	0.000	**0.125**	0.843
XGB+F1	8.644	0.058	0.005	0.005	0.005	0.842
XGB+Recall	22.825	0.094	0.009	0.009	0.004	0.827
XGB+ROC-AUC	4.862	0.043	0.003	0.003	0.006	**0.849**
XGB+KL	4.862	0.043	0.007	0.007	0.002	0.812

Box and whisker plots are provided in Figures 3.1 and 3.2. The median is shown with a line and notch on each box. The box extends from the first to third quartiles and the whiskers show the extent of 1.5 times the interquartile range. Outliers are depicted by the hollow points. Figure 3.1 shows a comparison between the different naïve methods of LR and XGBoost, trained by maximizing the F_1 score, recall, ROC-AUC, or D_{KL}. Figure 3.2 differs in that it compares a few of the naïve strategies with one PU strategy, SAR-EM, which requires the training of an NTC. This NTC is varied between tuning by F1 score, D_{KL}, or without tuning the default settings of $C = 1.0$ and using equal class weighting.

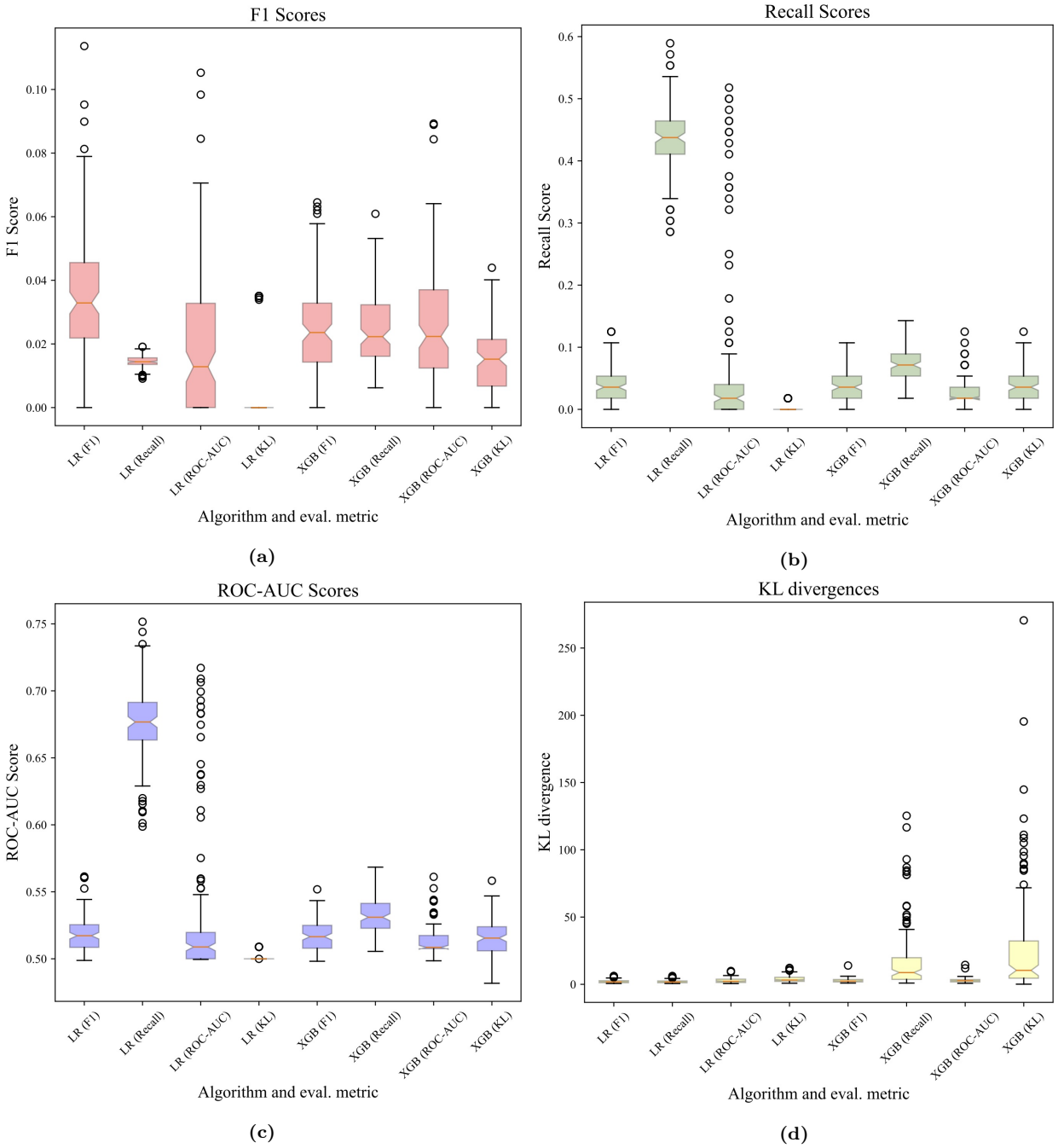


Figure 3.1: Box and whisker plots showing evaluation metrics for logistic regression (LR) and XGBoost (XGB), with hyperparameters tuned using the F1 score (F1), recall, ROC-AUC, or D_{KL} (KL). Scores result from averaging over 120 train/test splits in an inductive inference scheme. This figure included to show a comparison of the various naïve methods.

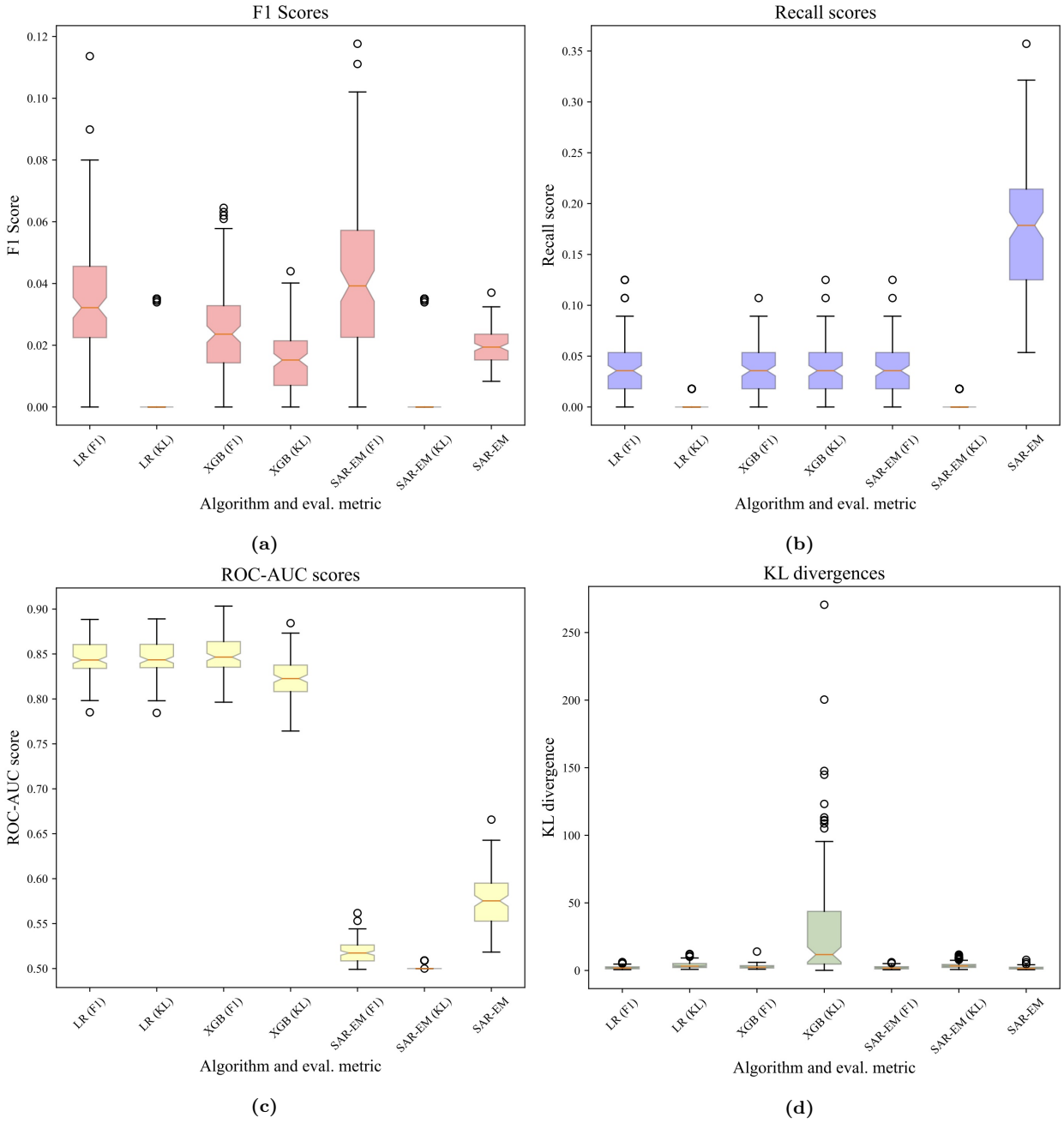


Figure 3.2: Box and whisker plots showing evaluation metrics for logistic regression (LR), XGBoost (XGB), and SAR-EM, with hyperparameters tuned using the F1 score (F1) or D_{KL} (KL). Fewer tuning methods are included in this figure in order to show a comparison between select naïve strategies and one PU strategy. For SAR-EM, tuning refers to the tuning of the NTC under the SCAR assumption. A SAR-EM experiment involving no tuning is included, with the default parameters of $C = 1.0$ and equal class weighting. Scores result from averaging over 120 train/test splits in an inductive inference scheme.

3.4 Favorability Maps

The non-PU favorability maps are displayed in Figures 3.3 and 3.4. The favorability maps from SAR-EM are provided in Figure 3.5 and those from DEDPUL are shown in Figure 3.6. These favorability maps are generated from training each method on all data then predicting on all data. Probability outputs from all methods are normal score transformed. For reference, four favorability maps from [6] are provided in Section 1.2.1. These include two maps created from WoE and LR using the methods from [9] and two created using LR and XGBoost in the single-classifier strategy.

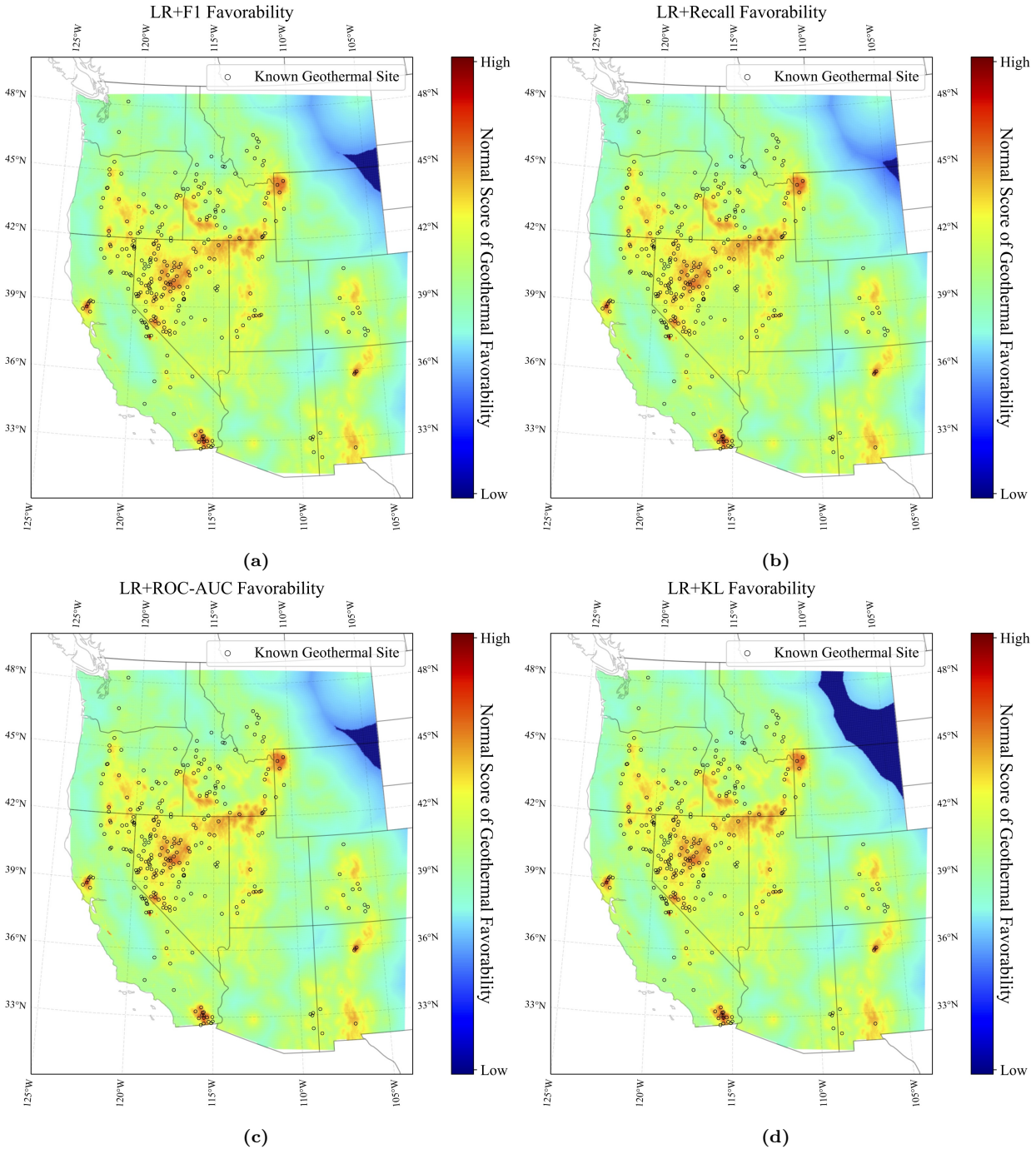


Figure 3.3: Favorability maps for logistic regression (LR) tuned using (a) F_1 scores (F1), (b) Recall, (c) ROC-AUC, or (d) D_{KL} (KL). Circles show the locations of known geothermal sites. Models are trained for transductive inference.

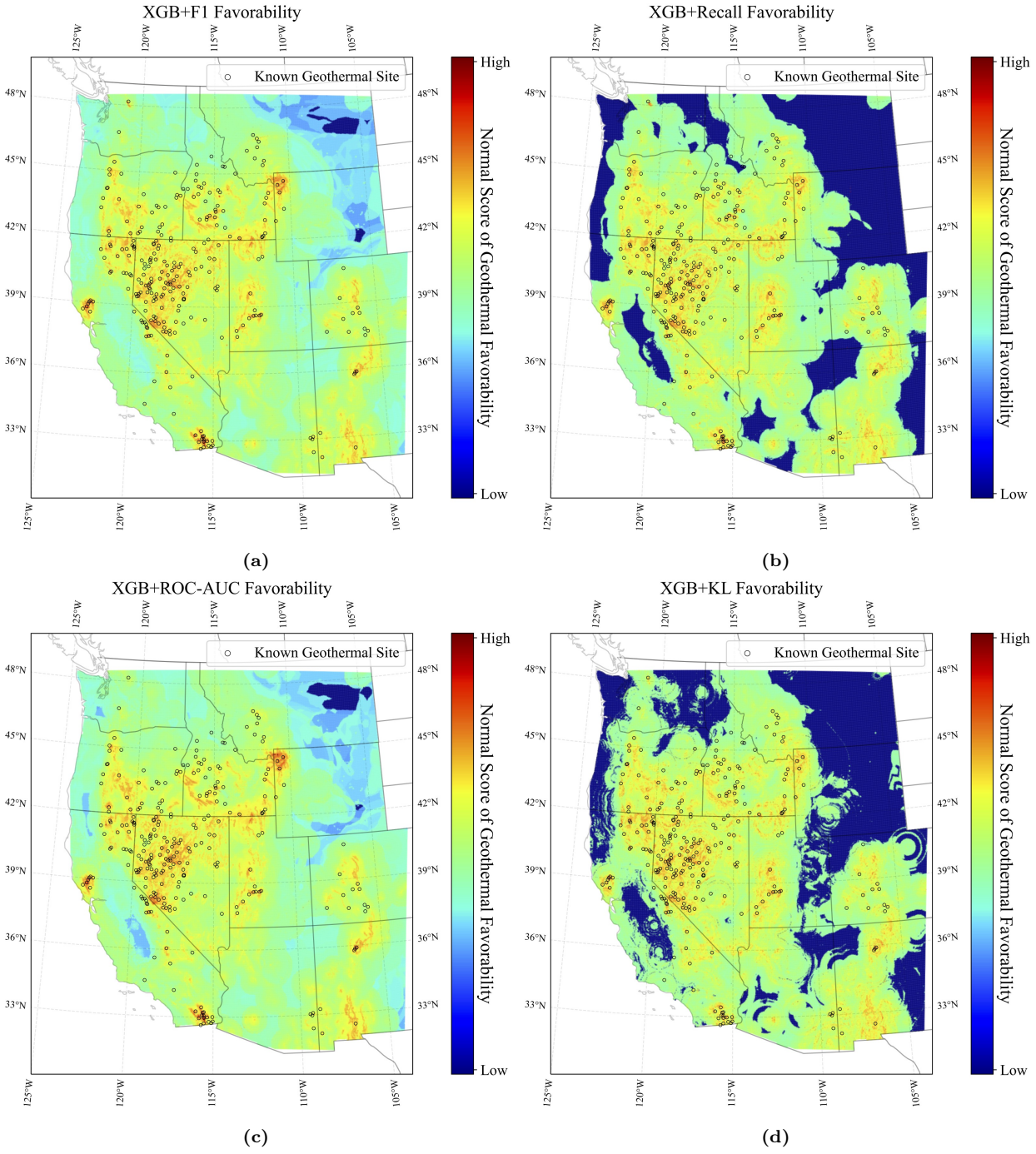


Figure 3.4: Favorability maps for XGBoost (XGB) tuned using (a) F_1 scores (F1), (b) Recall, (c) ROC-AUC, or (d) D_{KL} (KL). Circles show the locations of known geothermal sites. Models are trained for transductive inference.

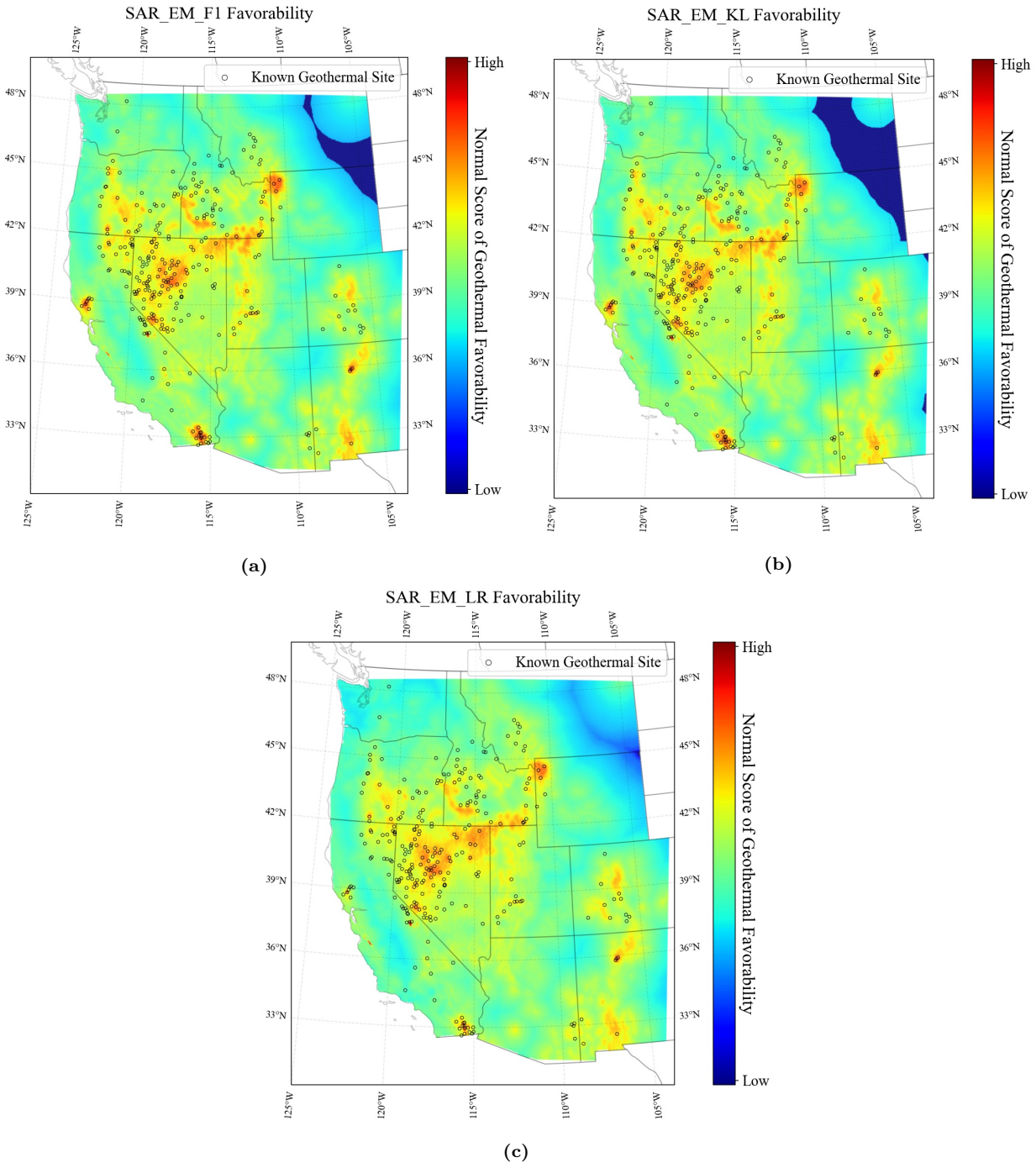


Figure 3.5: Favorability maps for SAR-EM using logistic regression (LR) (a) with F_1-tuned (F1), (b) D_{KL}-tuned (KL), and (c) default (no tuning) classifier models. Circles show the locations of known geothermal sites. Models are trained for transductive inference.

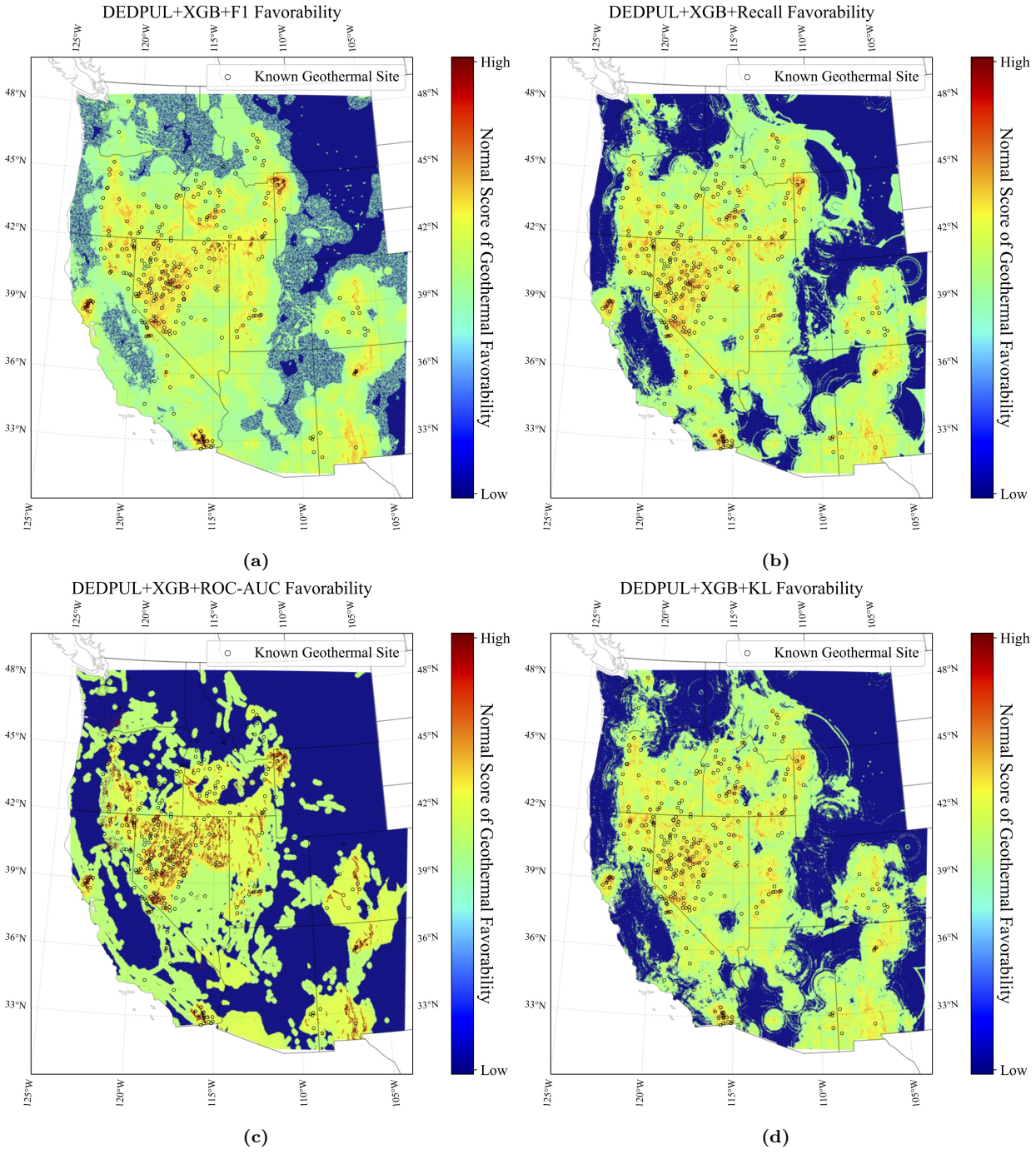


Figure 3.6: Favorability maps for DEDPUL with XGBoost (XGB) as NTC, tuned using (a) F_1 scores (F1), (b) Recall, (c) ROC-AUC, or (d) D_{KL} (KL). Circles show the locations of known geothermal sites. Models are trained for transductive inference.

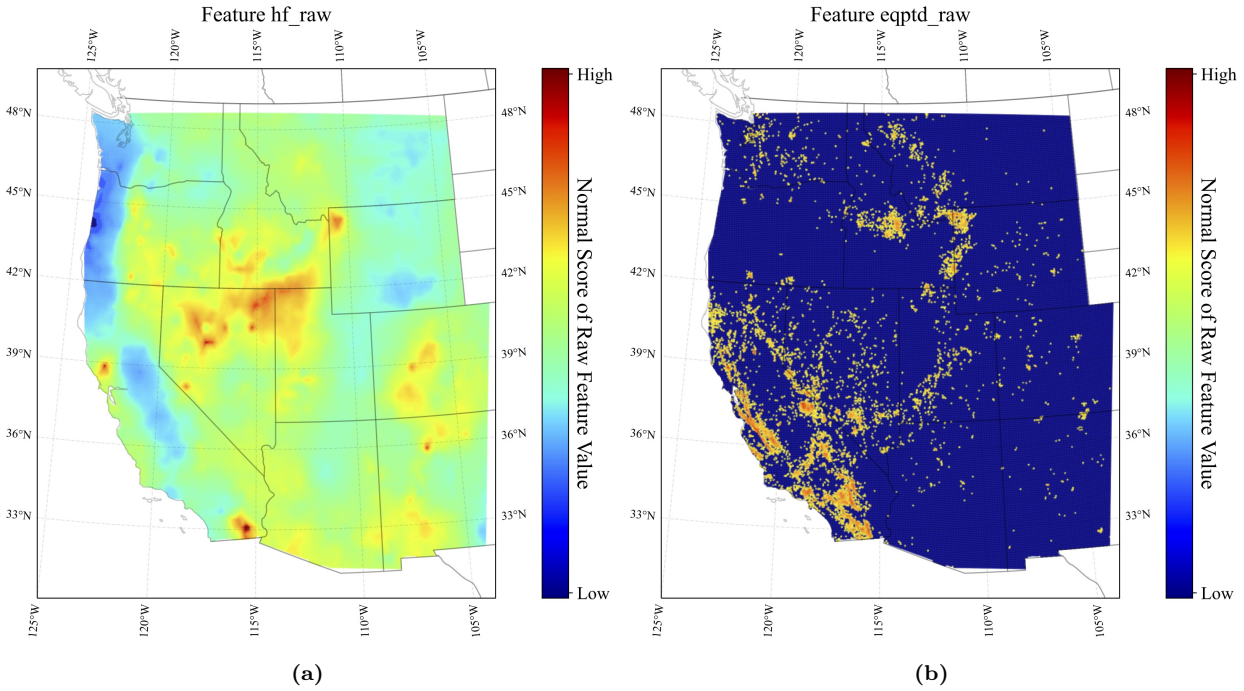


Figure 3.7: Single-feature maps for (a) heat flow (hf_raw) and (b) seismic density (eqptd_raw).

3.5 Single-Feature Maps

The five features (heat flow, fault distance, magma distance, seismic density, and stress) are each normal score transformed and displayed according to their coordinates in Figures 3.7 and 3.8. Single-feature maps such as these have the potential to show where each feature is "bleeding through," or showing more influence in the biases in each model.

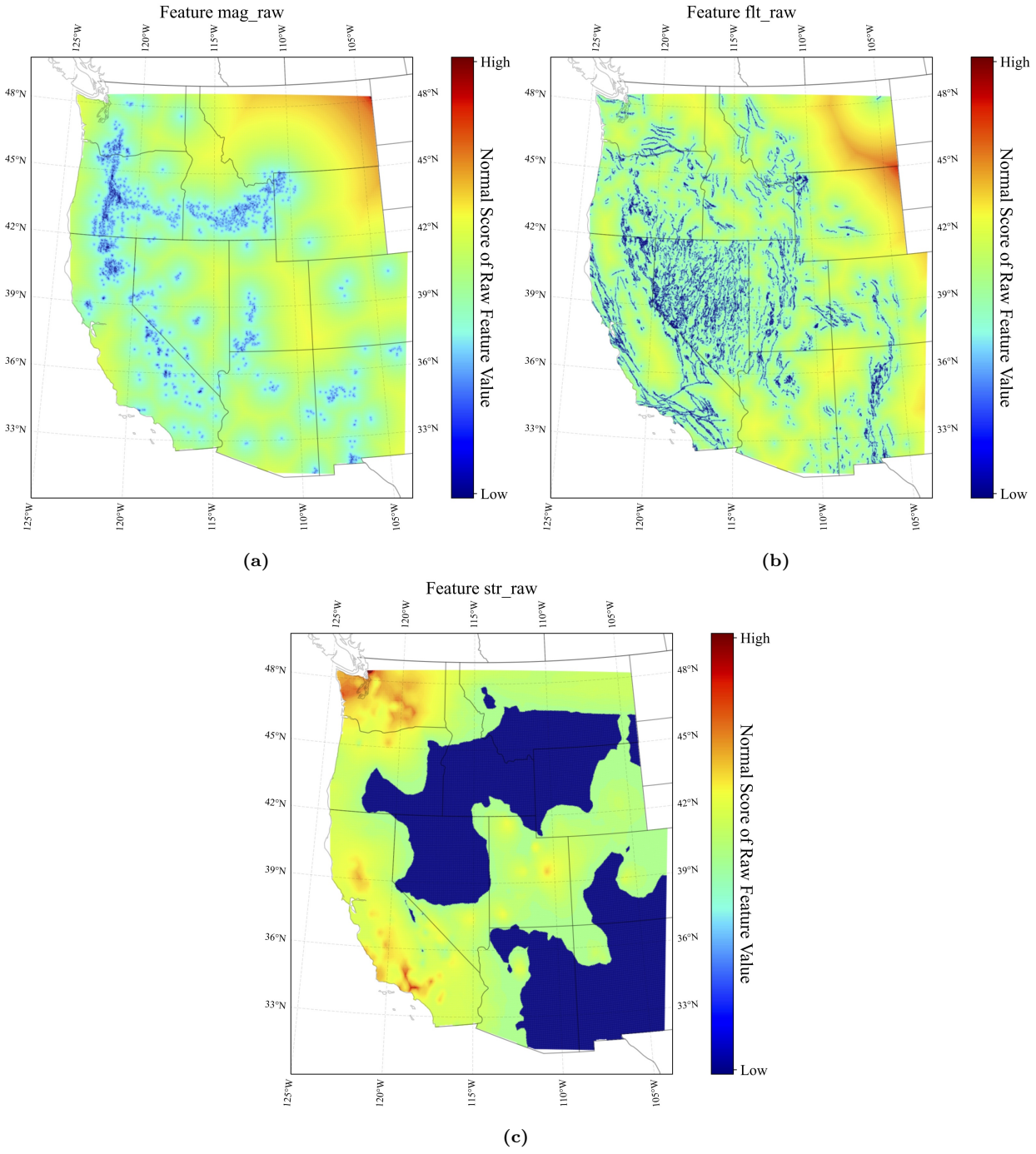


Figure 3.8: Single-feature maps for (a) magma distance (mag_raw), (b) fault distance (flt_raw), and (c) stress (str_raw).

3.6 Model Predictions

The ridge and CDF plots in Figures 3.9, 3.11, and 3.12 were created in a transductive manner by training one model on the entire dataset using the averaged optimal hyperparameters obtained through grid search optimization for each algorithm and evaluation metric. Predictions were then made on the entire dataset and normal-score transformed, following the same method as for the favorability maps. Despite being normal score transformed, the unlabeled distribution appears bimodal, particularly in Figure 3.12, because an overwhelming number of predictions produced by the model are extremely close to zero. For Figure 3.10, stratified out-of-sample predictions were used in an inductive manner to cover the entire dataset before normal-score transforming them. The difference between these two training and prediction strategies is that in the first (Figures 3.9, 3.11, and 3.12), cross-validation is used only for hyperparameter tuning. The final classifier for each model is trained on all data and then used to make predictions on all data, which could lead to an overfit model that performs well on in-sample predictions but does not translate as well to unseen data. In the second strategy (Figure 3.10), cross-validation is also used to train on subsets of the data in a stratified manner. Five separate final models are trained, each on a new random subset of data, and test predictions are made on the remaining unseen data by each of these five models. Obtaining predictions out-of-sample should prevent overfitting and better indicate the relative performance of each model when generalized to unseen data.

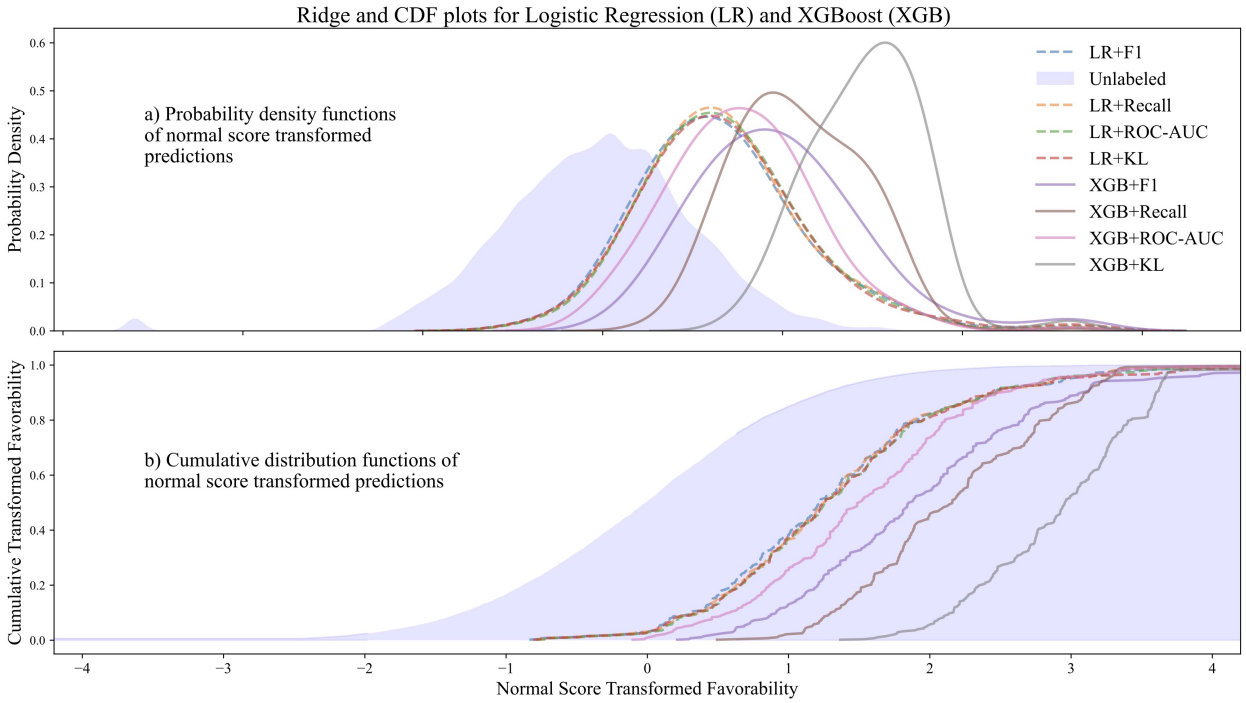


Figure 3.9: Ridge (top) and CDF (bottom) plots for logistic regression (LR) and XGBoost (XGB) prediction results, tuned using either their F_1 scores (F1), Recall, ROC-AUC, or D_{KL} (KL). Models are trained on all data and used to predict on all data. This is an indication of each model's performance when used for transductive inference.

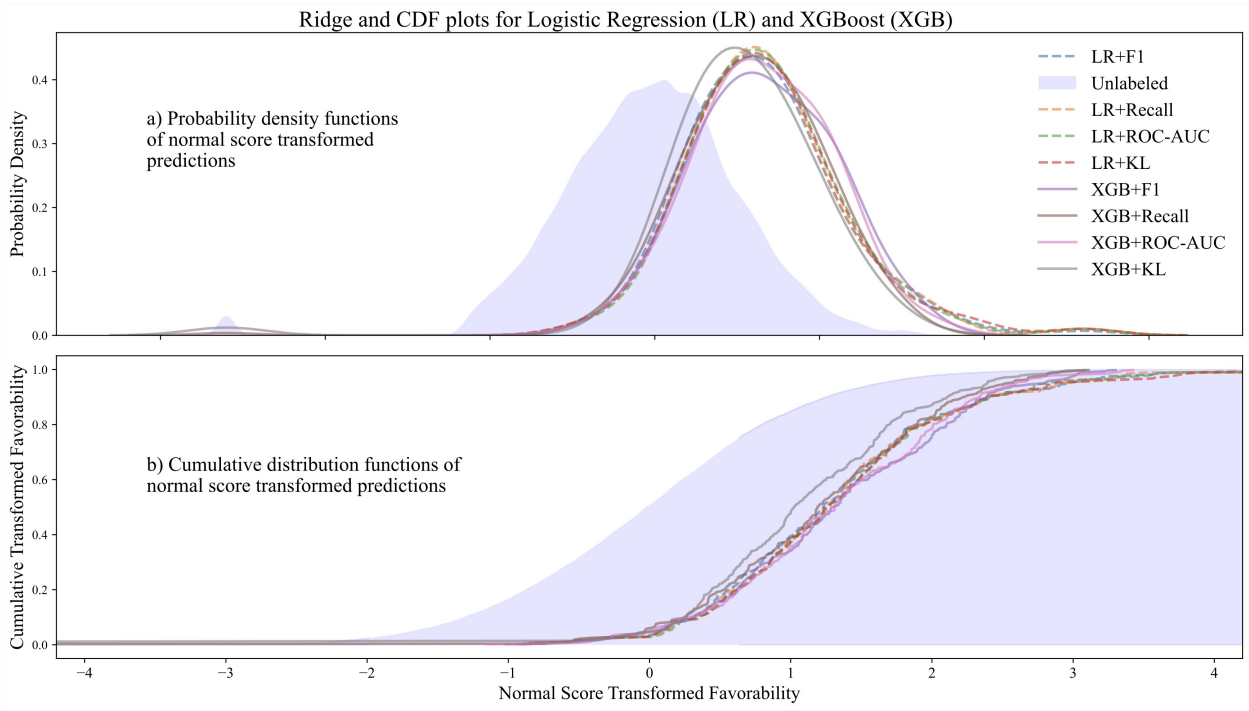


Figure 3.10: Ridge (top) and CDF (bottom) plots for logistic regression (LR) and XGBoost (XGB) prediction results, tuned using either their F_1 scores (F1), Recall, ROC-AUC, or D_{KL} (KL). Stratified cross-validation is used to obtain out-of-sample predictions across the entire dataset. This shows the relative model performance when used for inductive inference.

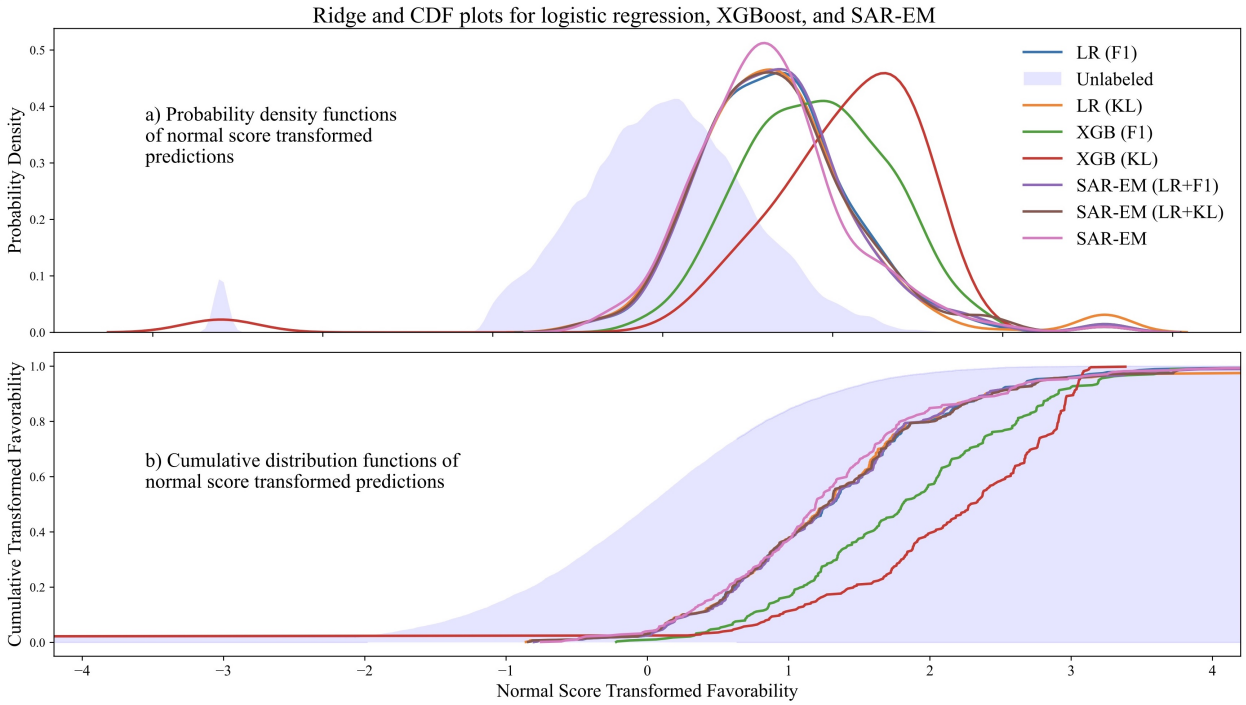


Figure 3.11: Ridge (top) and CDF (bottom) plots showing comparison of non-PU methods with SAR-EM methods. Similar to Figure 3.9, models are trained on all data and used to predict on all data in a transductive manner.

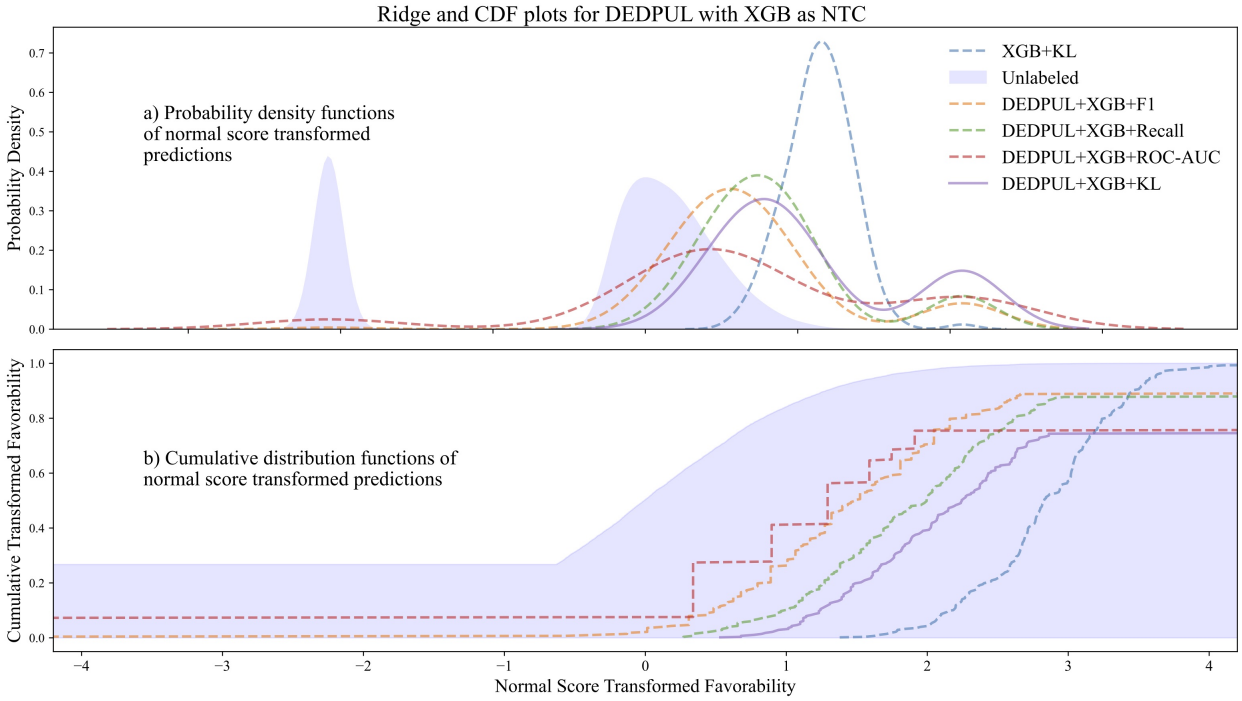


Figure 3.12: Ridge (top) and CDF (bottom) plots for DEDPUL with XGBoost (XGB) as NTC, tuned using either the F_1 scores (F1), Recall, ROC-AUC, or D_{KL} (KL). Models are trained on all data and used to predict on all data in a transductive manner.

4 Discussion

The hyperparameters found for XGBoost both through F_1 score and D_{KL} tuning agree with those found in [6]. For LR, however, only the class weight found through F_1 score tuning was similar in range to that reported in [6]. The optimal inverse regularization strength is a little less than three orders of magnitude larger (for LR tuned by F_1 score: 2023.1 vs. 3 as reported in [6]). In addition, the optimal class weight for LR tuned by D_{KL} is one order of magnitude lower than the natural class balance of 1:700 estimated by [5] and [6]. It is closer to the optimal class weights found by [6] for the single and ensemble classifier strategies when using the 95th percentile ratio of 1:955 for the natural class balance as estimated from [9].

The single-feature maps in Figures 3.7 and 3.8 allow us to view some of the influence each feature has on the final model predictions. We can see the areas of high heat flow in Figure 3.7 (a) contribute directly to the regions of high geothermal favorability, especially in Northern Nevada and centered around the Idaho-Nevada-Utah border. As another example, the additional confluence of high seismic density and proximity to bodies of magma as seen in Figure 3.7 (b) and (c), as well as proximity to faults and areas of low stress as depicted in Figure 3.8 (a) and (b), may result in the pocket of high favorability around the Idaho-Montana-Wyoming border. Interestingly, SAR-EM and LR seem to reproduce the region of low fault proximity in the northeast corner of the map in Figure 3.8 (a) in their favorability maps, Figures 3.5 and 3.3 respectively.

4.1 The Use of the D_{KL} for Model Evaluation

Figures 3.9, 3.10, and 3.11 show ridge and CDF plots to compare several different training and prediction pipelines. Figure 3.9 compares strategies when trained on all data using optimal hyperparameters and then used to make predictions on all data, in a transductive inference scheme. Finally, the predictions were normal-score transformed and separated into the posterior positive and prior unlabeled distributions. This is the same method used to generate the ridge and CDF plots provided in [6]. Mordensky et al. [5] showed that, even though the favorability maps produced by the XGBoost model more closely aligned with those created by expert opinion, the LR models consistently scored the best according to the F_1 score. Because predicting on the training set can be optimistic and not reflect true classifier performance on new data, Figure 3.10 compares the same strategies when cross-validation is used to obtain out-of-sample predictions across the entire dataset, in an inductive inference scheme. Indeed, when out-of-sample predictions are taken, the strategies perform more similarly to each other and it is harder to discern a clear winner. Interestingly, in this pipeline XGB+F1 and XGB+ROC-AUC perform a bit better than the other six strategies. Also of note is that XGB+KL performs the worst when out-of-sample predictions are taken. This differs greatly from the outcome of the "in-sample" predictions in Figure 3.9, which would suggest that the best performance is achieved by the XGB+KL strategy. Figure 3.11 compares a subset of the strategies in the previous two figures against the performance of the SAR-EM strategy. From this figure it is clear that SAR-EM achieves a similar performance to the control strategies taken from [5] and [6].

One could conclude from Figure 3.9 that XGBoost is capable of pushing the known positive distribution further away from the unlabeled distribution when optimizing the D_{KL}. However, one cannot determine just by examining the D_{KL} if this is due to the algorithm finding more true positive examples or if it is due to the D_{KL}'s tendency to make a larger

area positive, past what is likely for the underlying distribution. This uncertainty makes the D_{KL} an imperfect method of model evaluation but one that still may be helpful in identifying possible positive samples within the unlabeled distribution.

The favorability maps in Figures 3.3 and 3.4 show that XGBoost tuned to the D_{KL} has a sharper contrast between regions of high and low favorability. These results indicate an algorithm which is much more certain of its negative predictions. All favorability maps generated are in general agreement among regions of high favorability and most of the difference is observed in regions of lower favorability. As postulated in [6], this may be due to the inherent nature of the positive unlabeled data from the 2008 assessment [9] to train classifiers on positive examples which are similar to each other; positive examples all share similar qualities which make them more likely to be selected for labeling, whereas true negative examples may be considered negative for a number of different reasons. These results are in general agreement with the findings in [5] and [6], which indicate an inherent limitation of the data used in this study, and the need for a more comprehensive model of geological conditions in addition to a more standardized data collection procedure.

4.2 PU Learning Methods for Geothermal Favorability Prediction

As seen in Table 3.1, the SAR-EM method achieves a MAE of 1.01e-4 from the estimated expert opinion-based positive class prior of 1:700 or about 1.43e-3 positives per every one negative example. However, due to the findings in [6] and the optimal class weight hyperparameters found for LR and XGBoost, it is likely that the natural class balance contains more positive samples than this ratio indicates. The favorability maps generated from SAR-EM also very closely resemble those generated from the LR models. Interestingly, as seen in Table 3.4, the ROC-AUC scores for SAR-EM are much lower than the other methods (both the F_1- and D_{KL}-tuned models score just above random guessing, with the default model scoring marginally better). SAR-EM achieves the best Recall against the other non-PU strategies.

An interesting aspect of the ridge plots produced by DEDPUL, seen in Figure 3.12, is that the positive prediction probability density functions (PDFs) resemble bimodal Gaussians. The right peaks of the PDFs have lower maxima than the left peaks, but also have means that are greater than the mean of the positive prediction PDF resulting from XGB+KL. This may indicate the ability of DEDPUL to separate the positive predictions into two distinct regions based on some latent feature of the data.

5 Conclusion & Future Work

This thesis has presented several ways of incorporating techniques from PU learning into the problem of predicting geothermal favorability from a severely class-imbalanced PU dataset. In addition, it has shown the utility of using the D_{KL} between estimated positive and unlabeled distributions as a method of model evaluation when training machine learning models from PU data. The key contributions of this thesis are presented below, along with possible directions for future study.

5.1 The Use of the D_{KL} for Model Evaluation

The D_{KL} has been used previously for anomaly detection [31, 32], and for class prior estimation [33]. This thesis has presented maximizing the D_{KL} between prior unlabeled and predicted positive posterior distributions for model evaluation in PU learning problems as a novel application and provided insight into the possibilities of using other methods of model evaluation. The D_{KL} is an unbounded measure which does not depend on knowing the ground truth for training data. When used for hyperparameter optimization, it results in a model which more harshly penalizes likely negative samples. However, because the D_{KL} only considers the statistical properties of distributions, it does not directly allow for insight into whether a model correctly identifies an unlabeled sample as positive. Still, when used with other techniques it can be a valuable tool for identifying possible positive examples within the unlabeled distribution.

5.2 PU Learning Methods for Geothermal Favorability Prediction

This thesis has examined the classification results from three PU learning techniques: TIcE [47], SAR-EM [1], and DEDPUL [59], and the class prior estimation results from four PU learning algorithms: KM2 [2], TIcE [47], SAR-EM, and DEDPUL. For comparison with these state-of-the-art PU learning algorithms, the results from [5] and [6], with LR and XGBoost classifiers trained naïvely, were reproduced. Results from these two studies were reinforced by those obtained in this thesis, which show that the predictions from the nonlinear XGBoost model more closely resemble the results from the previous 2008 assessment [9]. The favorability maps generated by DEDPUL are similar in appearance to those produced using the D_{KL} for hyperparameter optimization, indicating it also has an affinity for giving likely negatives a lower favorability. The bimodality of the positive prediction PDFs in the DEDPUL ridge plots show that it may be able to identify distinct regions within the positive samples based on some latent features in the data. This aspect of the predictions made by DEDPUL would benefit from future study.

Because the body of work relating to positive unlabeled learning is rich yet fairly new, it is possible that other techniques may be better suited to the problem of geothermal favorability assessment which have not yet been considered. Clustering may be used to identify subclasses within the positive examples, possibly based on differing geological processes which may not be explicitly distinct when performing binary classification, to further enhance the performance of classification algorithms. It is possible that further utility can be gained from multiclass classification to separate currently accessible resources (those we are able to extract with current technology, e.g. steam and hot-water hydrothermal processes) from hopefully accessible resources (those we may have the ability to extract in the future through the continued development of new technology, e.g. magma).

Another avenue for future work is the incorporation of newer, more detailed datasets

which may provide deeper insight and allow for comparison of methods on similar but more consistent data. Short of using an entirely new dataset, the dataset from the 2008 USGS assessment [9] might also be improved through the use of rank pruning methods, as indicated by preliminary application of confident learning techniques to identify near-duplicates, outliers, and potentially mislabeled data [40]. Mordensky et al. [6] mention that a quantile-to-quantile transformation might be beneficial to remove outliers and further normalize the data. It would also be interesting to explore feature importance as demonstrated in [6], and how it changes relative to each PU method. Finally, classifiers may benefit from feature engineering (e.g. combining depth of magmatic activity, permeability, vertical connectivity, etc.) to better reflect the geophysical conditions required for favorable geothermal energy production.

References

[1] J. Bekker, P. Robberechts, and J. Davis, "Beyond the selected completely at random assumption for learning from positive and unlabeled data," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019*. Springer International Publishing, 2019, pp. 71–85.

[2] H. Ramaswamy, C. Scott, and A. Tewari, "Mixture proportion estimation via kernel embeddings of distributions," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, Jun. 2016, pp. 2052–2060.

[3] R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017, pp. 1674–1684.

[4] S. Jain, M. White, M. W. Trosset, and P. Radivojac, "Nonparametric semi-supervised learning of class proportions," *arXiv preprint arXiv: 1601.01944*, 2016.

[5] S. P. Mordensky, J. J. Lipor, J. Deangelo, E. R. Burns, and C. R. Lindsey, "Predicting geothermal favorability in the western United States by using machine learning: addressing challenges and developing solutions," in *47th Workshop on Geothermal Reservoir Engineering*, Feb. 2022, pp. 1–18.

[6] S. P. Mordensky, J. J. Lipor, J. DeAngelo, E. R. Burns, and C. R. Lindsey, "When less is more: How increasing the complexity of machine learning strategies for geothermal energy assessments may not lead toward better estimates," *Geothermics*, vol. 110, no. 102662, 2023.

[7] W. S. Lee and B. Liu, "Learning with positive and unlabeled examples using weighted logistic regression," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003, pp. 448–455.

[8] S. Jain, M. White, and P. Radivojac, "Recovering true classifier performance in positive-unlabeled learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 2017, pp. 2066–2072.

[9] C. F. Williams and J. DeAngelo, "Mapping geothermal potential in the western United States," *Geothermal Resources Council Transactions*, vol. 32, pp. 181–188, 2008.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[12] E. Barbier, "Geothermal energy technology and current status: an overview," *Renewable and Sustainable Energy Reviews*, vol. 6, no. 1, pp. 3–65, 2002.

[13] R. Bertani, "Geothermal energy: an overview on resources and potential," in *Proceedings of the International Conference on National Development of Geothermal Energy Use*, Slovakia, 2009.

[14] J. W. Tester, B. J. Anderson, A. S. Batchelor, D. D. Blackwell, R. DiPippo, E. M. Drake, J. Garnish, B. Livesay, M. Moore, K. Nichols *et al.*, "The future of geothermal energy," in *Oversight Hearing on Renewable Energy Opportunities and Issues on Federal Lands: Review of Title II, Subtitle B- Geothermal Energy of EPAct; and Other Renewable Programs and Proposal for Public Resources*, 2006.

[15] L. Rybach, "The future of geothermal energy and its challenges," in *Proceedings of the World Geothermal Congress*, vol. 29, Bali, Indonesia, 2010.

[16] J. W. Tester, K. F. Beckers, A. J. Hawkins, and M. Z. Lukawski, "The evolving role of geothermal energy for decarbonizing the United States," *Energy Environ. Sci.*, vol. 14, pp. 6211–6241, 2021.

[17] J. W. Lund and A. N. Toth, "Direct utilization of geothermal energy 2020 worldwide review," *Geothermics*, vol. 90, no. 101915, 2020.

[18] M. Krieger, K. A. Kurek, and M. Brommer, "Global geothermal industry data collection: A systematic review," *Geothermics*, vol. 104, no. 102457, 2022.

[19] D. Timmons, J. M. Harris, and B. Roach, "The economics of renewable energy," Global Development and Environment Institute, Tufts University, 2014.

[20] C. F. Williams, M. J. Reed, and R. H. Mariner, "A review of methods applied by the U.S. Geological Survey in the assessment of identified geothermal resources: U.S. Geological Survey Open-File Report 2008-1296," USGS, Tech. Rep., 2008.

[21] D. E. White and D. L. Williams, "Assessment of geothermal resources of the United States, 1975," U. S. Geological Survey, Tech. Rep., 1975. [Online]. Available: https://pubs.usgs.gov/publication/cir726

[22] L. J. P. Muffler, "Assessment of geothermal resources of the United States: 1978," U. S. Geological Survey, Tech. Rep., 1978. [Online]. Available: https://pubs.usgs.gov/publication/cir790

[23] C. F. Williams, M. J. Reed, J. DeAngelo, and S. P. G. Jr., "Quantifying the undiscovered geothermal resources of the United States," *Geothermal Resources Council Transactions*, vol. 33, pp. 995–1004, 2009.

[24] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794.

[25] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[26] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer Cham, 2009.

[27] K. Jaskie and A. Spanias, *Positive unlabeled learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. United States: Springer Nature, 2022.

[28] F. De Comité, F. Denis, R. Gilleron, and F. Letouzey, "Positive and unlabeled examples help learning," in *Algorithmic Learning Theory*, 1999, pp. 219–230.

[29] J. Bekker and J. Davis, "Learning from positive and unlabeled data: a survey," *Machine Learning*, vol. 109, no. 4, pp. 719–760, Apr. 2020.

[30] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[31] M. Afgani, S. Sinanovic, and H. Haas, "Anomaly detection using the Kullback-Leibler divergence metric," in *2008 First International Symposium on Applied Sciences on Biomedical and Communication Technologies*, 2008, pp. 1–5.

[32] L. Zhang, D. Veitch, and K. Ramamohanarao, "The role of KL divergence in anomaly detection," in *SIGMETRICS Perform. Eval. Rev.*, vol. 39, no. 1. Association for Computing Machinery, 2011, pp. 315–316.

[33] M. C. Du Plessis and M. Sugiyama, "Semi-supervised learning of class balance under class-prior change by distribution matching," *Neural Netw.*, vol. 50, pp. 110–119, 2014.

[34] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–50, Aug. 2016.

[35] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, pp. 221–232, 2016.

[36] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Springer, 2018, vol. 10.

[37] G. W. Imbens and D. B. Rubin, *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015.

[38] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 213–220.

[39] K. Jaskie and A. Spanias, "Positive and unlabeled learning algorithms and applications: a survey," in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2019, pp. 1–8.

[40] C. G. Northcutt, T. Wu, and I. L. Chuang, "Learning with confident examples: Rank pruning for robust classification with noisy labels," in *Uncertainty in Artificial Intelligence (UAI) 2017*, 2017.

[41] E. Sansone, F. G. D. Natale, and Z. H. Zhou, "Efficient training for positive unlabeled learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2584–2598, Nov. 2019.

[42] M. Kato, T. Teshima, and J. Honda, "Learning from positive and unlabeled data with a selection bias," in *International Conference on Learning Representations*, 2019.

[43] P. R. Rosenbaum and D. B. Rubin, *The Central Role of the Propensity Score in Observational Studies for Causal Effects*. Cambridge University Press, 2006, pp. 170–184.

[44] M. C. du Plessis, G. Niu, and M. Sugiyama, "Class-prior estimation for learning from positive and unlabeled data," *Machine Learning*, vol. 106, pp. 463–492, 2017.

[45] F. He, T. Liu, G. I. Webb, and D. Tao, "Instance-dependent PU learning by Bayesian optimal relabeling," arXiv pre-print, 2020.

[46] M. C. du Plessis and M. Sugiyama, "Class prior estimation from positive and unlabeled data," *IEICE Transactions on Information and Systems*, vol. 97, no. 5, pp. 1358–1362, 2014.

[47] J. Bekker and J. Davis, "Estimating the class prior in positive and unlabeled data through decision tree induction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018, pp. 2712–2719.

[48] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 03, pp. 447–461, Mar. 2016.

[49] C. Scott, "A rate of convergence for mixture proportion estimation, with application to learning from noisy labels," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Lebanon and S. Vishwanathan, Eds., vol. 38. San Diego, California, USA: PMLR, May 2015, pp. 838–846.

[50] G. Blanchard, G. Lee, and C. Scott, "Semi-supervised novelty detection," *Journal of Machine Learning Research*, vol. 11, pp. 2973–3009, Dec. 2010.

[51] S. Jain, M. White, and P. Radivojac, "Estimating the class prior and posterior from noisy positives and unlabeled data," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 2693–2701.

[52] H. Blockeel, D. Page, and A. Srinivasan, "Multi-instance tree learning," in *Proceedings of the 22nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: Association for Computing Machinery, 2005, pp. 57–64.

[53] S. Chaudhari and S. Shevade, "Learning from positive and unlabelled examples using maximum margin clustering," in *Proceedings of the 19th International Conference on Neural Information Processing - Volume Part III*, ser. ICONIP'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 465–473.

[54] B. Zhang and W. Zuo, "Reliable negative extracting based on kNN for learning from positive and unlabeled examples," *Journal of Computers*, vol. 4, no. 1, pp. 94–101, 2009.

[55] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, 2002, pp. 387–394.

[56] T. Peng, W. Zuo, and F. He, "SVM based adaptive learning method for text classification from positive and unlabeled documents," *Knowledge and Information Systems*, vol. 16, no. 3, pp. 281–301, 2008.

[57] X.-L. Li and B. Liu, "Learning from positive and unlabeled examples with different data distributions," in *Machine Learning: ECML 2005*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 218–229.

[58] W. Gerych, T. Hartvigsen, L. Buiquicchio, E. Agu, and E. Rundensteiner, "Recovering the propensity score from biased positive unlabeled data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, 2022, pp. 6694–6702.

[59] D. Ivanov, "DEDPUL: Difference-of-estimated-densities-based positive-unlabeled learning," in *International Conference on Machine Learning and Applications (ICMLA)*, 2020.

[60] A. Menon, B. V. Rooyen, C. S. Ong, and B. Williamson, "Learning from corrupted binary labels via class-probability estimation," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, 2015, pp. 125–134.

[61] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou, "On the class imbalance problem," in *2008 Fourth International Conference on Natural Computation*, vol. 4, 2008, pp. 192–201.

[62] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 233–240.

[63] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, no. 3, pp. 1–21, Mar. 2015.

[64] R. Ramola, S. Jain, and P. Radivojac, "Estimating classification accuracy in positive-unlabeled learning characterization and correction strategies," in *Pac. Syp. Biocomput.*, vol. 24, 2019, pp. 124–135.

[65] I. Csiszar, "I-Divergence Geometry of Probability Distributions and Minimization Problems," *The Annals of Probability*, vol. 3, no. 1, pp. 146–158, 1975.

[66] J. Hershey and P. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, 2007.

[67] J. Berkson, "Application of the logistic function to bio-assay," *Journal of the American Statistical Association*, vol. 39, pp. 357–365, 1944.

[68] V. Vapnik, *Estimation of Dependences Based on Empirical Data*. New York, NY: Springer, 2006.

[69] ——, "Transductive inference and semi-supervised learning," in *Semi-Supervised Learning*, O. Chapelle, B. Schölkopf, and A. Zien, Eds. Cambridge, MA: MIT Press, 2006, ch. 24, pp. 453–472.