AdaSampling for Positive-Unlabeled and Label Noise Learning with Bioinformatics Applications

Pengyi Yang, John T. Ormerod, Wei Liu, Chendong Ma, Albert Y. Zomaya, *Fellow, IEEE* and Jean Y.H. Yang

***Abstract*—Class labels are required for supervised learning but may be corrupted or missing in various applications. In binary classification, for example, when only a subset of positive instances is labeled whereas the remaining are unlabeled, positive-unlabeled learning is required to model from both positive and unlabeled data. Similarly, when class labels are corrupted by mislabeled instances, methods are needed for learning in the presence of class label noise. Here we propose AdaSampling, a framework for both positive-unlabeled learning and learning with class label noise. By iteratively estimating the class mislabeling probability with an adaptive sampling procedure, the proposed method progressively reduces the risk of selecting mislabeled instances for model training and subsequently constructs highly generalisable models even when a large proportion of mislabeled instances is present in the data. We demonstrate the utilities of proposed methods using simulation and benchmark data, and compare them to alternative approaches that are commonly used for positive-unlabeled learning and/or learning with label noise. We then introduce two novel bioinformatics applications where AdaSampling is used to (1) identify kinase-substrates from mass spectrometry-based phosphoproteomics data and (2) predict transcription factor target genes by integrating various next-generation sequencing data.**

***Index Terms*—Positive-unlabeled learning, Class label noise, Adaptive sampling, Bioinformatics, Kinase substrate prediction, Transcription factor target gene prediction.**

I. Introduction

SUPERVISED learning algorithms traditionally assume perfectly and fully specified class labels to build models that generalise to unseen data. In various real-world applications, however, the generalisation of classification models is often negatively affected by partial and/or mislabeled class labels [1], [2]. In bioinformatics applications, for example, defining genes that are unrelated to a genetic disease can be difficult as there could be unknown association between genes and the disease [3]. Similarly, characterising target genes of a transcription factor (TF) by its genome-wide binding profile is often complicated by false positive/negative binding sites used as class labels [4]. In binary classification, when only positive instances are available but negative instances are absent due to a lack of domain knowledge and/or definition [5], [6], positive-unlabeled learning techniques are often employed to model from positively labeled instances augmented with unlabeled instances that comprise of both unknown positive and negative instances [7], [8], [9]. When both positive and negative instances are available, but corrupted by labelling errors (i.e., mislabeling occurs), methods that implicitly or explicitly handle label noise should be utilised [10], [11].

While positive-unlabeled learning and learning with the presence of label noise are usually formulated and treated differently, they are nonetheless similar in the sense that both can be viewed as learning with ambiguity in the observed class labels [12]. Here, we extend our recently proposed adaptive sampling (AdaSampling) approach to handle both scenarios in a unified framework [13]. Akin to wrapper-based feature selection procedure [14], AdaSampling can be coupled with any probabilistic classifier to iteratively estimates the probability that each initial label is mislabeled. At each iteration, resampling from the initial data set is performed where the probability of exclusion of an instance into the new data set is equal to the estimated probability of the instance being mislabeled. This process continues with AdaSampling "wraps" around the classification model and prediction uncertainties for each instance are incorporated for each subsequent iterations of sampling to gradually reduce the probability of selecting instances that have a higher chance to be mislabeled for training the classifier (Figure 1).

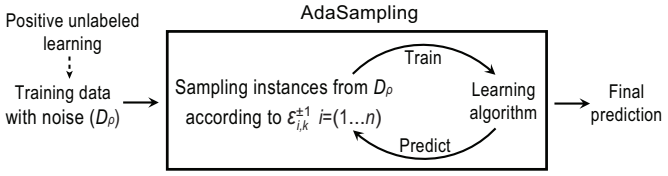


Fig. 1. A schematic illustration of AdaSampling framework for handling both positive unlabeled learning and learning with noisy class labels.

AdaSampling is noise-robust in that it adheres to a bootstrap-like sampling procedure. Yet, unlike common robust methods, AdaSampling explicitly handles unlabeled instances and/or class label noise but does not require bias estimation, nor does it impose on a pre-specified threshold for negative instance identification or noisy instance filtering as are often required in other approaches.

The rest of the paper is organised as follows. In Section II, we review related work in positive-unlabeled learning and

Pengyi Yang, Chendong Ma, and Jean Y.H. Yang are with the Charles Perkins Centre, School of Mathematics and Statistics, University of Sydney, NSW 2006, Australia.

John T. Ormerod is with School of Mathematics and Statistics, University of Sydney, NSW 2006, and the ARC Centre of Excellence for Mathematical and Statistical Frontiers, The University of Melbourne, Parkville, VIC, Australia.

Wei Liu is with Advanced Analytics Institute, University of Technology Sydney, NSW 2007, Australia.

Albert Y. Zomaya is with School of Information Technologies, University of Sydney, NSW 2006, Australia.

E-mail: pengyi.yang@sydney.edu.au

Manuscript received xx xx, xxxx; revised xx xx, xxxx.

learning in the presence of class label noise. In Section III, we introduce the AdaSampling framework and formulate AdaSampling-based approach with a single model and an ensemble of models for positive-unlabeled learning and learning with noisy labels. In Section IV, we describe the experimental designs and setups for simulation studies and performance comparison with other generic approaches that can be applied in conjunction with a wide range of classification algorithms using real-world benchmark datasets. Section V introduces and formulates two bioinformatics problems into positive-unlabeled learning and learning with class label noise, respectively. Section VI presents experimental results from simulation studies, performance comparisons, and applications in the above two bioinformatics problems. Section VII concludes and outlines future work.

II. Related Work

Positive-unlabeled learning methods can be roughly categorised into (i) heuristic, (ii) bias-based, (iii) one-class, and (iv) robust approaches.

Heuristic approaches often rely on partitioning the learning process into two steps where in the first step negative instances are identified using heuristics such as information retrieval techniques [15], Bayesian models [16], Expectation-Maximization [17], [18], or domain-specific knowledge [3], and then in the second step a final classification model is trained using labeled positive instances and unlabeled negative instances identified by the heuristic step. A key disadvantage of most heuristic approaches is the requirement of a predefined threshold to determine either to include or exclude a potential negative instance obtained from unlabeled data for final model training. The lack of formality and the need to find an optimal threshold in many heuristic methods for negative instance selection could greatly affect whether the final model will generalise to unseen data.

In contrast, bias-based approaches treat all unlabeled data as negative instances and employ a traditional learning algorithm but introduce a "bias to weight the classification and/or cost function towards the positive class [19]. Here, bias-based models use a reduced penalty for positive predictions made in unlabeled data. This approach was utilised for biased SVM [16], logistic model [20], naïve Bayes classification [21], ensemble learning [22], and have been subsequently formulated into a general framework that can be applied using a large selection of classification models [1]. Nevertheless, bias-based approaches often rely on training data for estimating the "bias" to be applied for model correction. Hence, part of the training data needs to be reserved for bias estimation and cannot be directly used for training the classification model. This is unattractive especially when the amount of training data is limited. Furthermore, the "bias estimation" assumes that a reasonably informative model is trained in the first place without accounting for unlabeled positive instances. This may not be the case when large number of unknown positive instances are present which could significantly affect on the "bias estimation", causing under- or over-correction and hence poor generalisation to unseen data.

Positive-unlabeled learning can also be treated as a one-class classification problem where only positive labels are used for training a classification model [23]. This has given rise to a set of approaches that adhere to the same principle of one-class classification but tuned for positive-unlabeled learning [24], [6], [25]. The drawback of adjusting one-class learning methods for positive-unlabeled learning is that they generally rely on generative classification models and ignore unlabeled data, and therefore need more labeled positive instances to achieve comparable performance in comparison to methods that effectively utilise both labeled and unlabeled instances.

Recently, methods based on bootstrap sampling were proposed to create robust models for positive-unlabeled learning [26], [27]. In such settings, unlabeled instances are treated as negative instances and bootstrap sampling is performed on unlabeled instances to create random subsets that are subsequently concatenated with labeled positive instances to train base classifiers and form a robust ensemble. These approaches exploit the property of bagging-like procedures [28] by taking advantage of instability caused by the random inclusion of unlabeled positive instances. However, unlabeled instances are not formally treated, and therefore still carry noise which propagates and affects the performance of the ensemble model.

Methods for classification with class label noise [29] are generally categorised into (i) noise-robust methods, (ii) data cleansing approaches, and (iii) noise modelling approaches [2].

Many noise-robust methods in noisy label learning make use of modified bagging [30] or boosting procedures [31], [32] and are conceptually similar to robust methods used for positive-unlabeled learning. Classification models that utilise certain loss functions, e.g. least-squares loss, can be robust to random classification noise (RCN) [33]. However, these methods do not explicitly handle the class label noise. Class label noise that propagate through classification models may distort the decision boundary especially when they violate the RCN assumption such as in asymmetric RCN where the prevalence of random label noise differs in each class [34].

Data cleaning approach formulates the noise label learning problem similarly as heuristic approach in positive-unlabeled learning in that an initial step is performed to identify, remove or relabel mislabeled instances, and a final classification model is trained using cleansed data [35]. For instance, outlier detection methods could be used for label noise identification [36], a classification model can be trained to identify mislabeled instances [37], [38], a clustering model can be used for weighting instances for classification [39], or a combination of unsupervised and supervised models can be used to find inconsistency in data labels and distributions for detecting mislabeled instances [40]. Since using a single model for data filtering may remove a large number of correctly labeled instances that are particularly close to the decision boundary, and therefore critical for training a highly generalisable model, ensemble approaches were proposed to improve the robustness for data filtering by majority or consensus voting [35], [41]. Similar to the heuristic approach used in positive-unlabeled learning, data cleansing methods in noisy label learning either require a threshold to be determined for instance filtering or rely on the prediction threshold of a classification model. This

is especially unattractive when there is a large number of mislabeled instances and a model trained on the noisy data need to make a one-time decision on which instances to retain for subsequent training of a final model.

An alternative approach to learning with label noise is to model the distribution of noisy instances and construct modified kernels or surrogate loss functions [42], [43], [44]. These approaches have strong theoretical supports and perform better when the underlying assumptions of class label noise hold [45]. But they require modifying the implementation of a classification algorithm and may not be generically applicable to all types of classification algorithms.

Finally, positive-unlabeled learning and learning with class label noise are related to partial label learning and learning with class imbalanced data. In partial label learning, each instance is supplied with multiple labels where only one of them is correct [46]. The problem of identifying the correct label can be viewed as removing incorrect labels associated with each instance, thus error-correcting [47]. In imbalanced data classification [48], where instances from one class significantly outnumber the other class, cost-sensitive metrics are often introduced to balance the instances from both majority and minority classes [49]. Alternatively, over-sampling approach can be applied to generate synthetic instances from the minority class [50] or under-sampling approach can be used to identify most representative instances from the majority class while filtering noisy instances [51], [52].

III. AdaSampling

Let us denote the noisy training data as D_ρ and comprises of instances $\{\mathbf{x}_i, y_i\} \in \mathcal{X} \times \{-1, +1\}$ and $(i = 1, \ldots, n)$. In a binary classification setting with asymmetric random classification noise (RCN) [11], specific noise rates $\rho^+ = P(Y' = -1|Y = +1)$ and $\rho^- = P(Y' = +1|Y = -1)$ are assumed to be associated with instances observed from negative and positive classes, respectively. It is assumed that $\rho^+, \rho^- \in [0, 1)$ and $\rho^+ + \rho^- < 1$ so that there are less incorrectly labeled instances than correctly labeled instances. Positive unlabeled learning can be viewed as discriminating positive and negative instances from a dataset with negative instances been contaminated by hidden positive instances, and therefore reduces to a special case of learning with class label noise where $\rho^+ \in [0, 1), \rho^- = 0$. Note that $\rho^\pm$ are the aggregated statistics of $\varepsilon_i^\pm = P(y_i' = \mp 1|y_i = \pm 1)$ which are the key quantities to be estimated in both problems.

A. Problem Formulation

It is not hard to see that the prediction uncertainties of a probabilistic model for each instance $1 - P(\widehat{y}_i|\mathbf{x}_i, D_\rho)$ is an estimator of $\varepsilon_i^\pm$ in that:

$$\begin{aligned}\widehat{\varepsilon_i^+} &= P(\widehat{y}_i = -1|\mathbf{x}_i, y_i = +1, D_\rho) \\ &= 1 - P(\widehat{y}_i = +1|\mathbf{x}_i, y_i = +1, D_\rho) \quad (1)\\ \widehat{\varepsilon_i^-} &= P(\widehat{y}_i = +1|\mathbf{x}_i, y_i = -1, D_\rho) \\ &= 1 - P(\widehat{y}_i = -1|\mathbf{x}_i, y_i = -1, D_\rho). \quad (2)\end{aligned}$$

Training a probabilistic classification model on the noisy data allows class label prediction for each instance to be made as a posterior probability $P(\widehat{y}_i|\mathbf{x}_i, D_\rho) = \text{predict}(h_\theta(D_\rho), \mathbf{x}_i)$. Classification-based data cleaning methods filter instances based on the estimation of $\varepsilon_i^\pm$ by $1 - P(\widehat{y}_i|\mathbf{x}_i, D_\rho)$ and a threshold defined for $\varepsilon_i^\pm$ on whether to retain each instance i. Due to potentially large amount of noisy labels may present in training data D_ρ, the estimation of $\varepsilon_i^\pm$ may not be accurate and filtering instances based on their $\varepsilon_i^\pm$ suffers from a high risk of removing instances that are particularly close to the decision boundary and therefore reduce the generalisation power of the classification model trained on the cleansed data.

AdaSampling mitigates this by resampling from D_ρ initially with uniform probability $1/n$ and trains a given classification model $h_{\theta,0}(D_{\rho,0})$ that iteratively updates the training dataset $D_{\rho,k}$ (k is the index of iterations) by weighted sampling from D_ρ with updated probability of mislabeling $\varepsilon_{i,k}^\pm$ for each instance. Note that sampling probabilities for instances in positive and negative classes are normalised so that $\sum_{i=1}^{n^+} \varepsilon_{i,k}^- / n^+ = \sum_{i=1}^{n^-} \varepsilon_{i,k}^+ / n^- = 1/2$ where n^+ and n^- are the numbers of positive and negative instances in D_ρ, respectively. The probability normalisation procedure allows similar number of positive and negative instances to be selected and therefore robust to datasets with imbalanced class distribution [48]. The weighted sampling with respect to the normalised probabilities reduces the risk of selecting potentially mislabeled instances without instance filtering.

B. AdaSampling-Based Learning Models

AdaSampling can be utilised in conjunction with various probabilistic classification algorithms for positive-unlabeled learning and/or learning with class label noise. As illustrated in the schematic diagram (Figure 1), the AdaSampling procedure wraps around a classification model to iteratively estimate $\varepsilon_{i,k}^\pm$ and subsequently update $D_{\rho,k}$. A criterion of:

$$\frac{1}{n}\sum_{i=1}^{n}\left|p_k(\widehat{y}_i|\mathbf{x}_i, D_{\rho,k}) - p_{k-1}(\widehat{y}_i|\mathbf{x}_i, D_{\rho,k-1})\right| < \delta \quad (3)$$

can be utilised to summarise the predictions for all instances in D_ρ in iteration k compared to $k-1$ and terminate if overall change is smaller than δ. We set δ to be 0.01, requiring smaller than 1% change in mean predicted probabilities of all instances for the process to terminate. The model from the final iteration is used to classify each instance in D_ρ or any unseen data drawn from the same distribution. Algorithm 1 summarises AdaSampling-based single classification model in pseudocode.

AdaSampling can also be extended for ensemble learning, in which the estimates $\varepsilon_{i,k}^\pm$ from the last iteration k can be used to sample from D_ρ multiple times to create multiple training datasets $D_{\rho,k}^\ell$, $(\ell = 1, \ldots, L)$. This allows for creating L base models $h_{\theta,k}^l(D_{\rho,k}^\ell)$ each trained on a different sample of the original dataset for ensemble prediction. The key advantage of this procedure is to make effective use of instances in D_ρ and avoid potential high variance introduced by training a single classification model. Algorithm 2 summarises AdaSampling-based ensemble of models in pseudocode.

Algorithm 1: AdaSampling-based single model

Data: Training data D_ρ
Result: Output $\widehat{\mathbf{y}}$

$\varepsilon^{\pm}_{i,0} \leftarrow \mathbf{0}$; // initialise likelihood of label flip
$D_{\rho,0} \leftarrow$ sampling(D_ρ, $\varepsilon^{\pm}_{i,0}$); // sampling w.r.t to $\varepsilon^{\pm}_{i,0}$
$P(\widehat{\mathbf{y}}|\mathbf{X}) \leftarrow 0$;
$k \leftarrow 0$;
do
  // train a model and classify all instances
  $P(\widehat{y}_1|\mathbf{x}_1, D_{\rho,k}), \ldots, P(\widehat{y}_n|\mathbf{x}_n, D_{\rho,k}) \leftarrow$ predict($h_{\theta,k}(D_{\rho,k}), \mathbf{X}$);
  // update likelihood of label flip using Eq. 1 and 2
  $\varepsilon^{\pm}_{i,k} \leftarrow 1 - P(\widehat{y}_i|\mathbf{x}_i)$;
  $D_{\rho,k} \leftarrow$ sampling(D_ρ, $\varepsilon^{\pm}_{i,k}$);
  if Eq. 3 $\geqslant \delta$ **then**
    $k \leftarrow k + 1$;
  end
while Eq. 3 $\geqslant \delta$;
$\widehat{\mathbf{y}} \leftarrow$ classify($h_{\theta,k}(D_{\rho,k}), \mathbf{X}$);

Algorithm 2: AdaSampling-based ensemble of models

Data: Training data D_ρ
Result: Output $\widehat{\mathbf{y}}$

$\varepsilon^{\pm}_{i,0} \leftarrow \mathbf{0}$; $D_{\rho,0} \leftarrow$ sampling(D_ρ, $\varepsilon^{\pm}_{i,0}$);
$P(\widehat{\mathbf{y}}|\mathbf{X}) \leftarrow 0$; $k \leftarrow 0$;
do
  $P(\widehat{y}_1|\mathbf{x}_1, D_{\rho,k}), ..., P(\widehat{y}_n|\mathbf{x}_n, D_{\rho,k}) \leftarrow$ predict($h_{\theta,k}(D_{\rho,k}), \mathbf{X}$);
  $\varepsilon^{\pm}_{i,k} \leftarrow 1 - P(\widehat{y}_i|\mathbf{x}_i, D_{\rho,k})$;
  $D_{\rho,k} \leftarrow$ sampling(D_ρ, $\varepsilon^{\pm}_{i,k}$);
  if Eq. 3 $\geqslant \delta$ **then**
    $k \leftarrow k + 1$;
  end
while Eq. 3 $\geqslant \delta$;
// create an ensemble of models $h^E_\theta \leftarrow \emptyset$;
for $l \in 1...L$ **do**
  $D^l_{\rho,k} \leftarrow$ sampling(D_ρ, $\varepsilon^{\pm}_{i,k}$);
  $h^E_\theta \leftarrow h^E_\theta \cup h^l_{\theta,k}(D^l_{\rho,k})$;
end
$\widehat{\mathbf{y}} \leftarrow$ classify($h^E_\theta, \mathbf{X}$);

C. Classification Models

AdaSampling is a generic framework and can be applied with a probabilistic model that measures the posterior probability $P(y|x, D)$. Here we selected four types of commonly used classification algorithms including radial kernel support vector machine (SVM), k-nearest neighbour (kNN), logistic regression (Logit), and linear discriminant analysis (LDA) and thus covers both linear and nonlinear classifiers as well as eager and lazy learning models. Specifically, the nonlinear lazy model kNN estimates the posterior probability as follows:

$$P(\widehat{y} = 1|\mathbf{x}, D) = \frac{1}{K} \sum_{\tau \in \mathcal{N}} I(y_\tau = 1) \tag{4}$$

where y_τ is the class label of an instance that is within a neighbourhood $\mathcal{N}$ of $\mathbf{x}$ (defined by Euclidean distance in this study), and K is a pre-defined size of $\mathcal{N}$.

For radial kernel SVM, a nonlinear eager learning model, the posterior probability is estimated by Platt's method [53]:

$$P(\widehat{y} = 1|\mathbf{x}, D) = \frac{1}{1 + \exp(A \times f(\mathbf{x}) + B)}$$
$$f(\mathbf{x}) = \beta + \sum_{\tau \in \mathcal{S}} \alpha_\tau \exp(-\gamma||\mathbf{x} - \mathbf{x}_\tau||^2_2) \tag{5}$$

where $\mathcal{S}$ is the support vector set and A and B are parameters (estimated by maximum likelihood) of a Sigmoid link function that converts the output $f(\mathbf{x})$ from the SVM into a probability.

For Logit, a parametric linear model, the posterior probability is estimated using a logistic function:

$$P(\widehat{y} = 1|\mathbf{x}, D) = \frac{e^{\beta_0 + \beta_1 \mathbf{x}}}{1 + e^{\beta_0 + \beta_1 \mathbf{x}}} \tag{6}$$

and when predictive variables in the data are approximately normal, an LDA can be used to better capture this information by estimating the posterior probability as follows:

$$P(\widehat{y} = 1|\mathbf{x}, D) = \frac{n^+ f^+(\mathbf{x})}{n^+ f^+(\mathbf{x}) + n^- f^-(\mathbf{x})} \tag{7}$$

where $f^+(\mathbf{x})$ and $f^-(\mathbf{x})$ are the density functions for positive and negative classes, respectively, defined as follows:

$$f^{\pm}(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}^{\pm})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}^{\pm})\right) \tag{8}$$

IV. Evaluation and Benchmark

A. Synthetic Datasets

Synthetic datasets were created to simulate the scenarios of positive-unlabeled learning and learning with class label noise. These were used to investigate the behaviour of the AdaSampling procedure. Specifically, for positive-unlabeled learning, we randomly generated 100 positively labeled instances and 400 unlabeled instances comprising 100 unlabeled positive instances and 300 unlabeled negative instances ($\rho^+ = 0.5$; $\rho^- = 0$). Each sample is represented by two features and for positive instances the two features were created from $\mathcal{N}(\mu = 6, \sigma^2 = 1)$ whereas for negative instances they are from $\mathcal{N}(\mu = 4, \sigma^2 = 1)$. The goal is to create a classification model that is capable of classifying data consisting of labeled and unlabeled positive instances and unlabeled negative instances.

In the case of learning with class label noise, we generated 100 positive instances each described by two features drawn independently from $\mathcal{N}(\mu = 6, \sigma^2 = 1)$ and 300 negative instances each described by two features drawn independently from $\mathcal{N}(\mu = 4, \sigma^2 = 1)$. We subsequently introduced 50 positive instances with their class label flipped to the negative class and 100 negative instances with their class label flipped to the positive class ($\rho^+ = 1/3$; $\rho^- = 1/4$). The aim is to recover the true class labels for mislabeled instances and correctly classify both positive and negatives instances in the dataset. Table I summarises synthetic data used for assessing AdaSampling in these two scenarios.

TABLE I
SYNTHETIC DATA FOR ADASAMPLING IN POSITIVE-UNLABELED LEARNING (PU) AND LEARNING WITH CLASS LABEL NOISE (LN).

Dataset	n_o^+, n_o^-, m	Setup	Task
Synthetic I	200, 300, 2	ρ^+=1/2; ρ^-=0	PU
Synthetic II	150, 400, 2	ρ^+=1/3; ρ^-=1/4	LN

B. Benchmark Datasets

To compare the proposed AdaSampling-based approach for positive-unlabeled and label noise learning with other commonly used alternative approaches, we obtained six benchmark datasets from UC Irvine Machine Learning Repository [54] that cover a range of sample and feature sizes. These include breast cancer diagnosis (referred to as "Breast"), predicting free electrons in the ionosphere ("Ionosphere"), sonar prediction of mines vs. rocks ("Sonar"), the Wisconsin database of breast cancer ("WDBC"), the Pima Indians diabetes dataset ("Pima"), and the spam e-mail database ("Spam").

To simulate positive-unlabeled learning scenarios, we randomly flipped 1/2, 2/3 or 3/4 of instances in the positive class to the negative class and treated them with the rest of negative instances as unlabeled instances in each dataset. This gives three experimental setups that we refer to as "easy", "median" and "hard" for each dataset on which the evaluation experiments were performed. To simulate learning with class label noise, we also created three setups where 35% and 15% (easy); 40% and 20% (median); and 50% and 30% (hard) of positive and negative instances were flipped to their opposite class, respectively. Table II shows the details of datasets and configurations used for benchmark comparison. The notation n_o^+ and n_o^- refer to the sizes of positive and negative instances with respect to the true labels.

TABLE II
SUMMARY OF DATASETS AND CONFIGURATIONS USED FOR POSITIVE-UNLABELED (PU) AND CLASS LABEL NOISE (LN) LEARNING.

Dataset	n_o^+, n_o^-, m		ρ^+	ρ^-	Task	Level
Breast	239, 444, 9		0.5	0	PU	easy
Inonsphere	126, 225, 34		2/3	0	PU	median
Sonar	97, 111, 60	×	0.75	0	PU	hard
WDBC	212, 357, 32		0.35	0.15	LN	easy
Pima	268, 500, 8		0.4	0.2	LN	median
Spam	1813, 2788, 57		0.5	0.3	LN	hard

C. Performance Comparison

We compared AdaSampling-based single model ("AdaSingle") and ensemble of models ("AdaEnsemble") with other commonly used generic approaches that can be applied to a wide range of classification algorithms. Specifically, for comparison in positive-unlabeled learning settings, we implemented bias-based approach (denoted as "BiasModel") described in [1] and bagging-like approach ("BagModel") described in [26]. For BiasModel, aligned with [1], we used 20% of the training dataset for estimating a bias factor that was used to correct for the final predictions. For BagModel, we implemented the procedure described in [26] where a bootstrap sample from unlabeled instances were drawn and combined with labeled positive instances for prediction.

For comparison in learning with class label noise we implemented the filtering model ("FilterModel") [37] and a bagging-like subsampling ("SubsampleBag") as described in [30]. The FilterModel used all instances and their provided class labels to train an initial classifier. Instances that are labeled as positive but predicted as negative and vice versa by the initial classifier were filtered to remove potentially mislabeled instances. The remaining data were used to train a final model for prediction. SubsampleBag approach is similar to Bagging but subsampling randomly 60% of the original data. AdaEnsemble, BagModel, and SubsampleBag are ensemble-based approaches and for all ensemble-based approaches the number of base classifiers were set to 50.

In all comparisons, the classification results from each original dataset without introducing label-flip were used as gold standard ("Original"), and the classification results after introducing label-flip but without applying any form of correction were used as a baseline ("Baseline") for comparisons.

A multi-layered repeated 5-fold cross-validation (CV) procedure was used for performance comparison. Specifically, class labels of instances were repeatedly (5 times) randomly removed in positive-unlabeled learning settings or flipped in class label noise settings to account for variability in class label perturbations. Data were subsequently split using 5-fold CV in such a way as to maintain the proportion of class label changes applied to each fold in each setting. To account for randomised data split in 5-fold CV, this procedure was repeated 10 times. Each time of data split in the 5-fold CV, the same partition of data was used for training or testing each of all methods to reduce variability in randomised data split and class label changes for model comparisons.

The performance of each method is the average of each trial plus and minus the mean standard error with respect to a given evaluation metric. Evaluation metrics included are sensitivity (Se), specificity (Sp), and F_1 score that combines the both. Area under the curve (AUC) is not included as a comparison metric as it is not effective for evaluating bias-based methods where the ranks of the prediction often remain the same [1], leading to adjusted thresholds but result in the same ROC curve. Since the study focuses on data with roughly balanced class distribution, the F_1 score provides a good trade-off of summarising sensitivity with specificity for the purpose of method comparison. The definition of each metric is included in Supplementary Materials.

The non-parametric Wilcoxon signed ranks test was applied for performance comparison on final predictions (w.r.t. F_1) over multiple datasets [55]. The test was two-sided, paired across each dataset, and p-values were reported.

V. BIOINFORMATICS APPLICATIONS

Partial and inaccurate class labelling are common in many bioinformatics applications [3], [56]. Here we formulate two novel biological problems into positive-unlabeled learning and learning with class label noise, respectively.

A. Kinase Substrate Prediction

Protein phosphorylation is one of the most common types of post-translational modifications in that a protein kinase

alters its substrates between activate and inactive forms for signal transduction [57]. The development of mass spectrometry (MS)-based phosphoproteomics has enabled global protein phosphorylation profiling at unprecedented scale and resolution [58]. A key challenge in phosphoproteomics data analysis is to identify novel substrates of kinases that are intimately involved in signal transduction, as these kinase-substrate relationships can subsequently be used to reconstruct signalling networks in the cell. Kinase substrate prediction can be formulated as a positive-unlabeled learning problem because often only a handful of known substrates (a.k.a. positive examples) of a given kinase are annotated in current protein phosphorylation databases [59], whereas the kinases that regulate the rest of MS quantified phosphorylation sites are unknown (i.e. unlabeled). Given a kinase of interest, the task is therefore to model from both its known substrates as well as the rest of phosphorylation sites to discover novel substrates that are phosphorylated by the given kinase.

Here we processed a time-course phosphoproteomics data of insulin treated adipocytes [58] and aim to predict novel substrates for Akt and mTOR kinases which are known to be the key nodes in insulin signalling networks [60]. The data contains 9 time points that can be used as learning features and there are 22 and 26 known substrates for Akt and mTOR, respectively, with another ~10,000 MS-quantified phosphorylation sites.

B. Transcription Factor Target Gene Prediction

Transcription factors (TFs) are key regulators that bind to specific genes to control their expressions [61]. Predicting TF target genes is critical for characterising transcriptional networks. The binding modes of TFs to their target genes are diverse and can cause significant challenges in transcription factor target gene prediction. For example, TFs can bind closely to their target genes (refer to as proximal targets) or to distal regions of their target genes (distal targets) [62]. Using chromatin immunoprecipitation followed by sequencing (ChIP-seq) [63], many of the TF proximal target genes can be identified with relatively high accuracy. While experimental techniques such as genome-wide chromatin interaction analysis with paired-end-tag (ChIA-PET) [64] can be used to identify distal genes by measuring interaction between distal TF binding sites and candidate gene promoters, calling any genes that have ChIA-PET interactions as TF distal targets will give high false positive rates because not all ChIA-PET measured interactions are functional. Since most of the proximal target genes are regulated by direct binding, and therefore are mostly positive instances whereas distal binding contains much more negative instances, we propose to formulate TF target gene prediction as learning with class label noise by treating proximal target genes as positive and distal ones as negative, and recover from each class "mislabeled" instances.

To apply AdaSampling for TF target gene prediction, we processed RNA-sequencing (RNA-seq) data from embryonic stem cell (ESC) differentiation [65], ChIP-seq data of two master TFs, Nanog and cMyc, that are known to regulate ESC identity and their differentiation [66], [67], and ChIA-PET data that profile interactions genome-wide in ESC [68]. By integrating these "multi-omics" data, we compiled a putative target list that comprises of both proximal and distal genes for Nanog and cMyc, respectively. The RNA-seq data contains 4 time points each with two biological replicates that can be used as learning features, and the putative positive and negative instances for Nanog are (339, 598) and cMyc are (2785, 1281).

VI. Results

A. Synthetic Data

We first assessed the AdaSampling on positive-unlabeled learning and learning with class label noise using simulations. Figure 2a shows the decision boundary of each of the four tested classification algorithms without (yellow strap; Baseline) or with (green strap) AdaSampling on a simulated positive-unlabeled learning scenario (see Section IV-A). Firstly, all four classification algorithms tested are susceptible to misclassifying positive instances (both labeled and unlabeled) presumably due to a great amount of unknown positive instances treated as negative instances in model fitting. In comparison, the decision boundary in all cases are largely corrected when AdaSampling is applied with each of the four classification algorithms, recovering most of the positive instances that are misclassified in Baseline cases. Importantly, while AdaSampling is a generic procedure it retains the characteristics of the classification algorithm-specific decision boundary in that it remains to be linear or nonlinear based solely on the classification algorithm utilised.

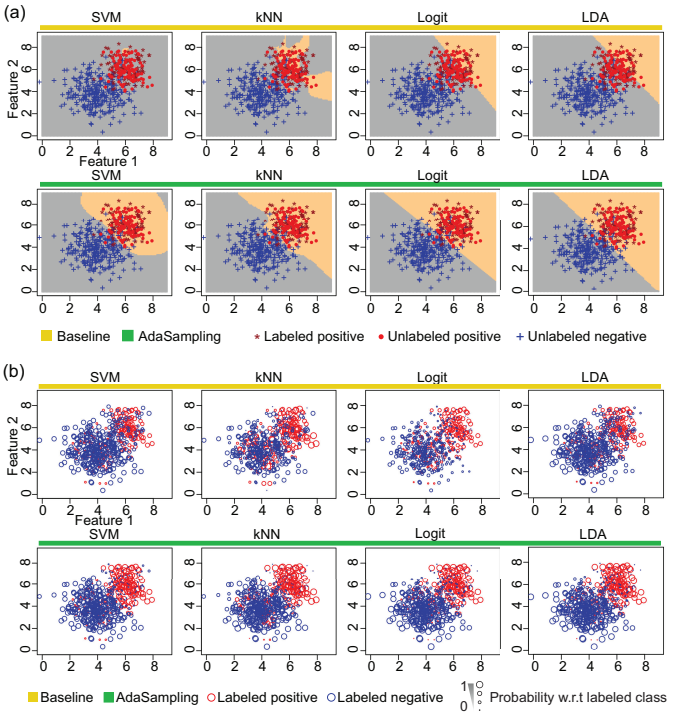


Fig. 2. Comparison of "Baseline" (yellow strap) and AdaSampling-based classification (green strap) in (a) positive-unlabeled learning and (b) learning with class label noise, using synthetic data.

Similarly, when applying AdaSampling for learning with class label noise (Figure 2b; green strap), the prediction

TABLE III
POSITIVE-UNLABELED LEARNING. FOR EACH DATASET AND CLASSIFIER, THE HIGHEST F_1 SCORES GIVEN BY EACH METHOD ARE IN BOLD.

	SVM	kNN	Logit	LDA	SVM	kNN	Logit	LDA	SVM	kNN	Logit	LDA
	Breast; easy (ρ^+=0.5; ρ^-=0)				Breast; median (ρ^+=2/3; ρ^-=0)				Breast; hard (ρ^+=0.75; ρ^-=0)			
Original	95.5±0	95.5±0	95.1±0	94.1±0	95.5±0	95.5±0	95.1±0	94.1±0	95.5±0	95.5±0	95.1±0	94.1±0
Baseline	57.4±0.7	64.3±0.6	60.1±0.4	70.9±0.6	11.4±1.2	40±0.5	31.2±0.9	47.2±0.8	3.4±0.8	25.3±0.6	15.1±1.1	33.3±0.8
BiasModel	78.5±0.3	88±0.2	83.4±0.3	81.4±0.3	67.4±1	79±0.4	76.8±1.1	72±1.1	58.8±1.3	69.4±0.7	72.8±1.4	67.7±1.3
BagModel	72.1±0.4	77.1±0.4	73.3±0.4	78.7±0.4	31.4±0.8	54.7±0.6	48.8±0.7	59±1	14.4±1.2	37.8±0.9	31.4±0.8	45.2±0.7
AdaSingle	95.1±0.1	95.7±0.1	92.6±0.2	**93.8±0.1**	95±0.1	95.1±0.1	91.5±0.3	93.3±0.2	95±0.1	94.8±0.1	90.8±0.3	93.1±0.2
AdaEnsemble	**95.2±0**	**95.7±0**	**93.2±0.2**	93.7±0.1	**95.2±0.1**	**95.3±0.1**	**92±0.3**	**93.4±0.2**	**95.1±0**	**94.9±0.1**	**91.9±0.3**	**93.2±0.2**
	Ionosphere; easy (ρ^+=0.5; ρ^-=0)				Ionosphere; median (ρ^+=2/3; ρ^-=0)				Ionosphere; hard (ρ^+=0.75; ρ^-=0)			
Original	91.3±0.1	73.2±0.2	75.9±0.2	75.7±0.3	91.3±0.1	73.2±0.2	75.9±0.2	75.7±0.3	91.3±0.1	73.2±0.2	75.9±0.2	75.7±0.3
Baseline	25.5±1.4	45.7±1.8	51.5±0.7	53.4±0.6	1.3±0.2	27.5±2.3	40.1±0.6	40.2±0.6	0.2±0.1	16.7±1.7	37.3±0.5	37±0.6
BiasModel	84.5±0.4	71.9±0.9	**66.7±0.5**	67.1±0.6	75.3±0.9	58.6±1.7	56.4±0.6	58±0.7	71.2±1.2	50.8±1.5	52.1±0.7	53.2±0.9
BagModel	68.8±0.9	55.9±2	62.1±0.5	60.8±0.5	41.5±1.3	36.2±2.8	50±0.6	47.9±0.7	23.5±1.7	21.5±2.2	46.3±0.6	42.6±0.6
AdaSingle	88.2±0.3	73.5±1.1	64.6±0.4	69.5±0.4	85.1±0.3	**61.9±1.8**	62.2±0.5	66.8±0.4	80.7±0.5	52.5±2.2	58.5±0.7	62.2±0.6
AdaEnsemble	**89.3±0.2**	**74.6±1.1**	65.5±0.4	**69.9±0.4**	**86.6±0.4**	61.2±2.2	**64.1±0.4**	**68.2±0.4**	**83.4±0.5**	**53.8±2.5**	**61.8±0.6**	**64.7±0.5**
	Sonar; easy (ρ^+=0.5; ρ^-=0)				Sonar; median (ρ^+=2/3; ρ^-=0)				Sonar; hard (ρ^+=0.75; ρ^-=0)			
Original	80.9±0.2	77.9±0.2	67±0.3	71.7±0.4	80.9±0.2	77.9±0.2	67±0.3	71.7±0.4	80.9±0.2	77.9±0.2	67±0.3	71.7±0.4
Baseline	36.8±1.9	53±2.2	51.8±1	50.2±1.3	13.4±1.8	33.5±2.5	44.3±1	37±1.3	3.9±0.8	25.4±2.3	36.9±1	30.8±1.2
BiasModel	64.7±0.5	**63.6±1**	52.6±0.8	58±0.8	52±0.9	49.7±1	45.7±1	46.6±0.9	48.9±1.2	45.2±1.3	39.9±0.9	41.4±1
BagModel	55.5±0.9	59.4±1.9	56.1±1	56.5±1.2	31.3±2.2	39±2.5	45.8±1.2	43.8±1.2	19.3±2	29.5±2.4	39.9±1.2	37.7±1.4
AdaSingle	67±0.4	60.4±0.8	55.8±0.7	56.8±0.7	59.5±0.6	**53.9±0.7**	49.6±0.6	56.1±0.8	56.9±1	52.5±0.9	50.4±0.7	57.9±0.6
AdaEnsemble	**68.5±0.4**	61.1±0.8	**59.1±0.9**	**59.2±0.8**	**60.5±0.5**	52.9±0.6	**54.3±0.8**	**60.7±0.8**	**58.2±1**	**52.6±0.8**	**54.5±0.6**	**61.3±0.6**
	WDBC; easy (ρ^+=0.5; ρ^-=0)				WDBC; median (ρ^+=2/3; ρ^-=0)				WDBC; hard (ρ^+=0.75; ρ^-=0)			
Original	96.6±0	90.1±0.1	92.2±0.1	93.7±0.1	96.6±0	90.1±0.1	92.2±0.1	93.7±0.1	96.6±0	90.1±0.1	92.2±0.1	93.7±0.1
Baseline	42.7±1.6	57.3±0.6	57±0.5	60.4±0.4	5.9±0.8	37.7±0.8	34.6±0.7	39.6±0.8	0.5±0.2	23.4±0.8	22.8±0.6	28.3±0.6
BiasModel	81.4±0.4	79.9±0.4	81.9±0.4	79.4±0.6	70.3±0.9	70.3±0.4	72.4±0.7	70.1±0.9	64.5±1	61.7±0.5	68.5±1.1	67±1
BagModel	67.8±0.3	69.1±0.5	70.8±0.4	71.8±0.3	25.8±1.5	49.6±0.7	50.6±0.5	52±0.5	8.7±1	34±0.8	37.9±0.6	40.7±0.6
AdaSingle	93.2±0.2	**88.6±0.1**	85.8±0.3	92.5±0.2	91.9±0.1	87.6±0.2	82.1±0.5	88.9±0.3	**91.4±0.2**	87±0.2	78.6±0.5	85.6±0.5
AdaEnsemble	**93.2±0.1**	88.4±0.1	**87.6±0.3**	**93.1±0.1**	**92.3±0.1**	**87.9±0.2**	**85.1±0.4**	**90.2±0.3**	**91.4±0.2**	**87.7±0.2**	**83±0.4**	**87.3±0.4**
	Pima; easy (ρ^+=0.5; ρ^-=0)				Pima; median (ρ^+=2/3; ρ^-=0)				Pima; hard (ρ^+=0.75; ρ^-=0)			
Original	61.5±0.1	55.2±0.1	63.7±0.1	63.3±0.1	61.5±0.1	55.2±0.1	63.7±0.1	63.3±0.1	61.5±0.1	55.2±0.1	63.7±0.1	63.3±0.1
Baseline	2.3±0.2	28.9±0.3	17.8±0.4	21±0.4	0.3±0.1	14.8±0.4	4.4±0.2	6.4±0.2	0±0	9.6±0.4	1.9±0.2	3.4±0.3
BiasModel	52.1±0.1	53.6±0.2	66±0.2	66.1±0.2	51.7±0.1	46.6±0.3	65.1±0.2	64.3±0.2	51.7±0	42±0.5	64.1±0.2	63.2±0.3
BagModel	16.5±0.5	39±0.4	37.2±0.4	38.3±0.5	2.6±0.1	23±0.5	14.1±0.2	16.7±0.2	0.7±0.1	15.8±0.5	6.9±0.4	8.9±0.5
AdaSingle	65.5±0.2	61.8±0.2	65.9±0.2	67±0.1	64.4±0.1	60.5±0.3	65.2±0.2	65.9±0.2	63.7±0.2	59.4±0.3	64.3±0.2	64.6±0.3
AdaEnsemble	**66±0.1**	**63±0.2**	**66.4±0.1**	**67.2±0.1**	**65.3±0.1**	**61.5±0.3**	**65.5±0.2**	**66.1±0.1**	**64.9±0.2**	**61.3±0.3**	**64.6±0.2**	**65.4±0.2**
	Spam; easy (ρ^+=0.5; ρ^-=0)				Spam; median (ρ^+=2/3; ρ^-=0)				Spam; hard (ρ^+=0.75; ρ^-=0)			
Original	91.2±0	88.2±0	90.1±0.2	84.7±0.1	91.2±0	88.2±0	90.1±0.2	84.7±0.1	91.2±0	88.2±0	90.1±0.2	84.7±0.1
Baseline	27.6±1.3	55.7±0.7	41.7±0.4	47±0.5	1±0.1	31±0.7	7.6±0.2	21.1±0.2	0.5±0.1	19.7±0.8	3.1±0.1	12.1±0.9
BiasModel	80.3±0.4	80.8±0.2	**88.5±0.1**	81.8±0.2	57.9±0.3	71.1±0.4	85.6±0.2	79.2±0.3	56.5±0	61.8±0.4	84±0.3	75.7±0.4
BagModel	66.3±0.2	68.2±0.6	67.5±0.2	61.2±0.3	11.9±0.5	42.9±0.6	26.4±0.4	34.6±0.2	3.8±0.2	29.2±0.8	9.5±0.2	21.2±0.5
AdaSingle	89.5±0.1	84.2±0.1	87±0.5	86.3±0.1	88.6±0.1	82.5±0.2	84.4±0.5	86.4±0.1	87.9±0.1	80.5±0.3	82.7±0.3	85.5±0.1
AdaEnsemble	**89.6±0.1**	**85.2±0.1**	88.3±0.2	**86.4±0.1**	**89±0.1**	**84±0.2**	**86.3±0.5**	**86.6±0.1**	**88.5±0.1**	**82.7±0.2**	**84.6±0.3**	**85.9±0.1**

confidence for correctly labeled instances (with respect to their labeled class) are much higher than those without (Figure 2b; yellow strap). Whereas for the mislabeled instances, the prediction probability to their corresponding class labels dropped significantly. This can be seen from the much less colour overlap in predictions for each of the four classification algorithms with AdaSampling compared to those without (green strap vs yellow strap in Figure 2b).

B. Evaluation on Benchmark Data

The comparison on six UCI benchmark datasets in positive-unlabeled learning are shown in Table III and Supplementary Table 1-3 in Supplementary Materials. The results from Original and Baseline comparison demonstrate consistently, i.e., across all datasets, that prediction sensitivity suffers most severely when we directly apply any of the four tested classification algorithms (SVM, kNN, Logit, and LDA) in positive-unlabeled learning settings. Furthermore, the decrease in sensitivity is clearly associated to the increase in percentage of unlabeled positive instances as demonstrated by the "easy", "median", and "hard" settings. BiasModel improves prediction sensitivity in most cases but could over-correct towards positive class such as those in Pima dataset when using SVM, suggesting a potential problem on correcting a poorly fitted initial classifier. BagModel appears to improve the prediction sensitivity moderately compared to Baseline but the improvement is lower than other alternative methods and therefore leads to an overall lower F_1 score in most cases. This is expected because though the bootstrap sampling on unlabeled instances can incorporate model diversity [69], this procedure does not enforce unlabeled positive instances to be treated as negative examples in base classifier training since the sampling is completely random. In comparison, AdaSampling-based single model (AdaSingle) and ensemble of models (AdaEnsemble) show higher overall F_1 score in three positive-unlabeled settings across all datasets. While AdaSingle and AdaEnsemble both sacrifice specificity when improving sensitivity, the improvement in sensitivity outweigh the relatively small decline in specificity, and hence leads to the most competitive performance in terms of F_1.

We observed similar results in learning with class label noise (Table IV and Supplementary Table 4-6 in Supplementary Materials). Specifically, FilterModel appears to offer limited improvement on F_1 in both "easy", "median", and "hard" settings potentially due to the removal of correctly labeled training instances that are close to the decision boundary since these instances are likely to be excluded in a single pass by a pre-defined filtering threshold. SubSampleBag in comparison

TABLE IV
LEARNING WITH CLASS LABEL NOISE. FOR EACH DATASET AND CLASSIFIER, THE HIGHEST F_1 SCORES GIVEN BY EACH METHOD ARE IN BOLD.

	SVM	kNN	Logit	LDA	SVM	kNN	Logit	LDA	SVM	kNN	Logit	LDA
	Breast; easy (ρ^+=0.35; ρ^-=0.15)				Breast; median (ρ^+=0.4; ρ^-=0.2)				Breast; hard (ρ^+=0.5; ρ^-=0.3)			
Original	95.5±0	95.5±0	95.1±0	94.1±0	95.5±0	95.5±0	95.1±0	94.1±0	95.5±0	95.5±0	95.1±0	94.1±0
Baseline	90.6±0.2	76.9±0.3	84.1±0.2	86.4±0.2	84.2±0.4	72.6±0.3	79.2±0.5	81.3±0.5	61.5±0.5	56.6±0.3	62±0.6	63.7±0.6
FilterModel	90.6±0.2	82.9±0.4	84.4±0.2	90.2±0.2	84.9±0.4	79.2±0.4	79.4±0.5	86±0.6	61.5±0.6	58.3±0.5	61.9±0.6	69.9±0.6
SubsampleBag	92±0.2	81.6±0.3	84.2±0.2	86.3±0.2	85.8±0.4	77.5±0.3	79.2±0.5	81.3±0.5	58.7±0.8	58.2±0.4	61.7±0.7	63.6±0.6
AdaSingle	**95.2±0.1**	94.2±0.2	91.3±0.2	**93.3±0.1**	**95.2±0.1**	90.1±0.7	89.8±0.2	**93.1±0.1**	90.9±0.3	76.8±0.8	78.6±0.6	83.7±0.4
AdaEnsemble	95±0.1	**94.8±0.2**	**92±0.2**	93.1±0.1	**95.2±0.1**	**91.8±0.5**	**90.6±0.2**	93±0.1	**92.5±0.2**	**80.3±0.9**	**80.7±0.5**	**85.3±0.4**
	Ionosphere; easy (ρ^+=0.35; ρ^-=0.15)				Ionosphere; median (ρ^+=0.4; ρ^-=0.2)				Ionosphere; hard (ρ^+=0.5; ρ^-=0.3)			
Original	91.3±0.1	73.2±0.2	75.9±0.2	75.7±0.3	91.3±0.1	73.2±0.2	75.9±0.2	75.7±0.3	91.3±0.1	73.2±0.2	75.9±0.2	75.7±0.3
Baseline	78.7±0.4	57.5±0.7	65.1±0.6	65.3±0.5	71.8±0.8	56.3±1	63±0.9	62.9±0.9	33.8±1.1	45.5±0.6	56.6±0.8	56.4±0.8
FilterModel	75.4±0.5	56.1±0.8	**66.5±0.5**	63.5±0.5	69±0.8	55.4±1.5	62.7±0.8	60.6±0.9	47.8±0.8	45.1±1.4	55.8±0.8	52.7±0.8
SubsampleBag	74.7±0.5	58.5±1	65.5±0.6	65±0.6	66.4±1.1	57.3±1.3	**63.5±0.9**	63±1	22.5±1.1	45.1±1.1	**56.8±0.9**	56.1±0.8
AdaSingle	87±0.4	68.9±0.5	59.1±0.6	**68.5±0.5**	86±0.3	70.5±0.6	58±0.5	66.7±0.5	69.9±1.8	**54.9±1.7**	50.4±0.6	56.2±0.7
AdaEnsemble	**87.6±0.4**	**70.7±0.4**	60±0.6	68.3±0.5	**87±0.3**	**72.8±0.6**	58.2±0.5	**67.6±0.5**	**71.6±1.2**	54.9±2.1	50.9±0.6	**57.9±0.7**
	Sonar; easy (ρ^+=0.35; ρ^-=0.15)				Sonar; median (ρ^+=0.4; ρ^-=0.2)				Sonar; hard (ρ^+=0.5; ρ^-=0.3)			
Original	80.9±0.2	77.9±0.2	67±0.3	71.7±0.4	80.9±0.2	77.9±0.2	67±0.3	71.7±0.4	80.9±0.2	77.9±0.2	67±0.3	71.7±0.4
Baseline	51.9±1.3	**59.4±0.8**	56.1±0.5	56.6±0.5	46.3±1.5	**56.6±0.6**	52.4±0.6	53.1±0.7	20.9±1.5	48.5±0.7	46.4±0.6	45.9±0.6
FilterModel	55.2±1.2	57.6±0.8	55±0.6	54.3±0.7	**50.8±1.3**	55.6±0.7	52.6±0.6	51.6±0.8	22.6±1.5	46.2±0.9	46.1±0.5	46.3±0.6
SubsampleBag	42±1.2	54.1±0.8	**56.3±0.7**	**56.8±0.6**	38.8±1.7	52.4±0.7	52.9±0.7	52.6±0.7	12.7±1.3	46.2±0.6	47.4±0.7	46.4±0.7
AdaSingle	64.5±1.4	55.1±0.5	54.3±0.6	53.7±0.5	50.2±2.3	55.2±0.6	52.4±0.5	53.6±0.6	**29.5±1.4**	**52.1±0.8**	50.6±0.6	**50.9±0.6**
AdaEnsemble	**65.2±1.3**	54±0.5	55±0.7	54.1±0.5	42.9±2.9	54.2±0.7	**53.4±0.6**	**54.1±0.6**	15.8±1	50.9±0.8	**51±0.6**	50.8±0.6
	WDBC; easy (ρ^+=0.35; ρ^-=0.15)				WDBC; median (ρ^+=0.4; ρ^-=0.2)				WDBC; hard (ρ^+=0.5; ρ^-=0.3)			
Original	96.6±0	90.1±0.1	92.2±0.1	93.7±0.1	96.6±0	90.1±0.1	92.2±0.1	93.7±0.1	96.6±0	90.1±0.1	92.2±0.1	93.7±0.1
Baseline	84±0.4	70.5±0.4	76.6±0.2	78.1±0.2	77.5±0.5	63.4±0.3	71.9±0.4	73±0.4	47.7±1.4	49±0.4	53.8±0.7	54.2±0.8
FilterModel	85.4±0.3	73.1±0.5	76.9±0.3	80.2±0.2	78.6±0.6	66.8±0.4	71.3±0.4	74.6±0.5	47.5±1.3	50.9±0.5	53.2±0.7	53±0.7
SubsampleBag	83.2±0.4	76±0.5	77.1±0.3	78.2±0.2	76.8±0.4	69.8±0.4	72.3±0.4	73±0.4	43.1±1.5	51.5±0.5	54.7±0.6	54.8±0.8
AdaSingle	92.9±0.2	86.8±0.3	76.4±0.5	89.4±0.2	91.9±0.2	82.9±0.3	72.2±0.3	86.6±0.4	75.9±1.6	64.9±0.5	57.8±0.5	64.1±0.7
AdaEnsemble	**93.6±0.2**	**87.5±0.3**	**77.5±0.4**	**90.1±0.2**	**92.5±0.3**	**84.4±0.2**	**73±0.3**	**87.3±0.4**	**80±1.4**	**67.5±0.5**	**58.2±0.4**	**65.3±0.7**
	Pima; easy (ρ^+=0.35; ρ^-=0.15)				Pima; median (ρ^+=0.4; ρ^-=0.2)				Pima; hard (ρ^+=0.5; ρ^-=0.3)			
Original	61.5±0.1	55.2±0.1	63.7±0.1	63.3±0.1	61.5±0.1	55.2±0.1	63.7±0.1	63.3±0.1	61.5±0.1	55.2±0.1	63.7±0.1	63.3±0.1
Baseline	34.9±0.3	42.8±0.3	44±0.8	44.7±0.8	26.8±0.9	42.6±0.2	36±0.9	36.4±0.9	20±0.9	39±0.3	24.3±0.7	24.3±0.7
FilterModel	36.8±0.3	42.3±0.4	44±0.8	42.4±1	27.2±1.1	41.6±0.4	36.6±0.8	33.3±0.9	21.4±0.8	37.9±0.4	23.8±0.8	18.8±0.8
SubsampleBag	30.6±0.4	43.1±0.3	44.1±0.8	44.5±0.8	21.1±1	41.3±0.3	36.4±0.9	36.6±0.8	15.2±1	38.5±0.3	24.4±0.7	24.3±0.7
AdaSingle	62.2±0.2	56±0.3	63.9±0.2	64.1±0.2	59.5±0.4	53.6±0.3	63.9±0.3	64.1±0.3	35.2±1.5	48.4±0.3	55.5±0.7	55.6±0.7
AdaEnsemble	**63.1±0.2**	**57.1±0.3**	**64.3±0.2**	**64.2±0.2**	**60.4±0.5**	**55.2±0.2**	**64.2±0.3**	**64.2±0.3**	**35.3±1.7**	**49.7±0.4**	**56.5±0.7**	**57±0.7**
	Spam; easy (ρ^+=0.35; ρ^-=0.15)				Spam; median (ρ^+=0.4; ρ^-=0.2)				Spam; hard (ρ^+=0.5; ρ^-=0.3)			
Original	91.2±0	88.2±0	90.1±0.2	84.7±0.1	91.2±0	88.2±0	90.1±0.2	84.7±0.1	91.2±0	88.2±0	90.1±0.2	84.7±0.1
Baseline	78.4±0.1	69±0.3	68.7±0.1	68.4±0.1	71.2±0.1	62.2±0.2	59.7±0.4	60.2±0.6	32.5±1.4	49.9±0.4	38.7±0.3	39.1±0.2
FilterModel	79.6±0.2	70.9±0.3	69.3±0.2	68.8±0.1	72.5±0.2	65.2±0.2	59.7±0.6	59.8±0.8	32.4±1.3	50.5±0.5	38.3±0.3	39.1±0.4
SubsampleBag	77.6±0.2	72.1±0.2	69.1±0.2	68.5±0.1	69.4±0.2	65.4±0.2	60.3±0.4	60.4±0.5	26.8±1.4	50.4±0.5	39.2±0.4	39.4±0.3
AdaSingle	89.2±0.1	82±0.3	87.3±0.2	82.7±0.1	88.1±0.1	78.8±0.3	85.1±0.2	83.3±0.3	80.8±0.5	63.6±0.4	75±0.3	76.4±0.3
AdaEnsemble	**89.4±0.1**	**83.8±0.2**	**87.9±0.1**	**83±0.1**	**88.4±0.1**	**81±0.3**	**85.9±0.2**	**83.6±0.3**	**82±0.5**	**66±0.3**	**76.1±0.4**	**77.1±0.4**

TABLE V
PAIRED WILCOXON SIGNED RANKS TEST FOR PERFORMANCE COMPARISON (F_1) OF ADAENSEMBLE TO OTHER METHODS IN POSITIVE-UNLABELED (PU) LEARNING AND LEARNING WITH CLASS LABEL NOISE (LN). TEST IS TWO-SIDED AND REPORTED ARE p-VALUES.

	PU; easy (ρ^+=0.5; ρ^-=0)				PU; median (ρ^+=2/3; ρ^-=0)				PU; hard (ρ^+=0.75; ρ^-=0)			
	Baseline	BiasModel	BagModel	AdaSingle	Baseline	BiasModel	BagModel	AdaSingle	Baseline	BiasModel	BagModel	AdaSingle
AdaEnsemble	1.2×10^{-7}	8.0×10^{-5}	1.2×10^{-7}	1.0×10^{-4}	1.2×10^{-7}	1.2×10^{-7}	1.2×10^{-7}	3.7×10^{-4}	1.2×10^{-7}	1.2×10^{-7}	1.2×10^{-7}	2.9×10^{-5}
	LN; easy (ρ^+=0.35; ρ^-=0.15)				LN; median (ρ^+=0.4; ρ^-=0.2)				LN; hard (ρ^+=0.5; ρ^-=0.3)			
	Baseline	FilterModel	SubsampleBag	AdaSingle	Baseline	FilterModel	SubsampleBag	AdaSingle	Baseline	FilterModel	SubsampleBag	AdaSingle
AdaEnsemble	3.0×10^{-5}	1.4×10^{-4}	1.6×10^{-5}	1.2×10^{-3}	1.4×10^{-4}	1.6×10^{-5}	4.7×10^{-5}	3.0×10^{-3}	1.0×10^{-5}	7.1×10^{-5}	2.3×10^{-6}	1.8×10^{-3}

is more conservative in that all instances will be treated equally. However, similarly to BagModel in positive-unlabeled learning, SubSampleBag does not directly estimate for mislabeled instances but relies on model diversity to improve the final classification accuracy. Therefore, it suffers from the same problem as all mislabeled instances can pollute the training subsets in SubSampleBag, and therefore propagate to the final prediction. Both AdaSingle and AdaEsnemble show notably better performance than other alternative methods across all datasets and classification algorithms tested.

Using the Wilcoxon signed ranks test for performance comparison of the proposed approach and other alternatives (Table V), we found that the performance of AdaEnsemble in terms of F_1 score is statistically significantly better than all other methods including AdaSingle in both positive-unlabeled learning and learning with class label noise. While the numeric improvement of AdaEnsemble on AdaSingle is comparatively small, such improvement appears to be consistent across most datasets, settings, and classification algorithms.

C. Application in Bioinformatics

1) Kinase Substrate Prediction: In kinase substrate prediction using phosphoproteomics data, often only a handful of substrates are known for a given kinase [58]. This problem can easily be formulated into a positive-unlabeled learning task where the goal is to uncover novel kinase substrates that are embedded in the unlabeled phosphorylation sites that are profiled in phosphoproteomics data. We demonstrate this by using a time-course phosphoproteomics dataset profiled in insulin stimulated adipocytes [58]. It is known that kinases such as Akt and mTOR are activated after insulin stimulation

and therefore the aim is to identify novel substrates for Akt and mTOR by learning the typical response of previously known substrates for each kinase in the phosphoproteomics dataset.

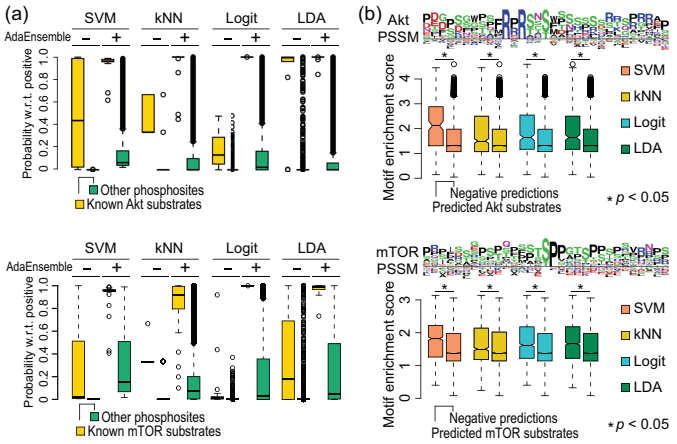


Fig. 3. Positive-unlabeled learning for predicting novel kinase substrates. (a) Comparison of prediction probability with respect to positive class for known Akt and mTOR substrates and other phosphorylation sites with ("+") or without ("-") using AdaEnsemble. (b) Comparison of motif enrichment in terms of position-specific scoring matrix score for predicted Akt and mTOR substrates and their respective negative predictions. Statistical significance was calculated using Wilcoxon signed ranks test.

For Akt, the prediction probability of being a substrate for either known Akt substrates or other unknown phosphorylation sites (or phosphosites) with or without using AdaSampling are shown in Figure 3(a), upper panel. As can be seen, directly applying a classification model (columns marked by "-") on this dataset gives relatively low prediction score (in terms of probability; yellow boxes) for known Akt substrates and discover almost no new substrates from other phosphosites (green boxes). In contrast, using AdaEnsemble for positive-unlabeled learning (columns marked by "+") gives much higher prediction score for known substrates while also uncover many potential novel substrates. Similar results are observed for mTOR (Figure 3(a), lower panel). Together, these results suggest that AdaEnsemble can accurately classify known Akt and mTOR substrates while may also uncover many previously unknown substrates for each kinase.

To validate novel substrate predictions, we obtained known Akt and mTOR kinase recognition motifs from public database [59] and compared the motif enrichment score (PSSM) for those that are predicted to be substrates of either Akt (Figure 3(b), upper panel) or mTOR (Figure 3(b), lower panel) with those that are predicted to be negative. We found that positive predictions for Akt and mTOR are significantly enriched for their respective kinase recognition motifs compared to negative predictions, suggesting these predicted novel substrates indeed possess structures that are more likely to be recognised and bind by Akt or mTOR.

2) Prediction of Transcription Factor Target Genes: A key task in characterising cell type-specific transcriptional networks is to accurately identify genes that are directly targeted and regulated by cell type-specific TFs. By integrating RNA-seq, ChIP-seq, and ChIA-PET data generated from ultrafast sequencing techniques, we created a multi-omic dataset comprising all putative target genes that are potentially regulated by proximal and/or distal TF bindings in ESCs (see V-B for details). Nevertheless, we expect that a subset of genes labeled as positive instances are not truly regulated by TF bindings and vice versa. Therefore, the task is to remove target genes that are falsely labeled as target genes and recover genes that are true targets but mislabeled as negative.

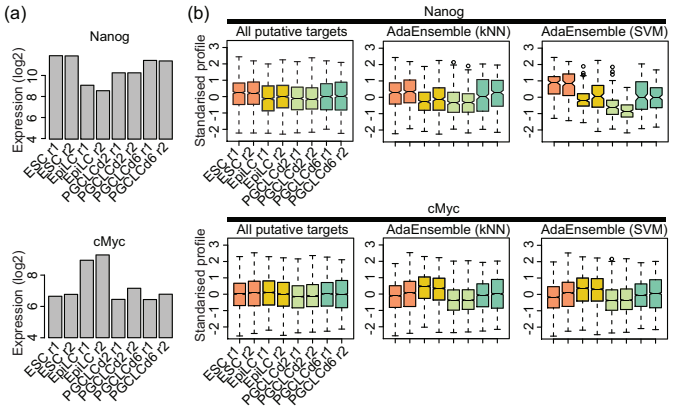


Fig. 4. Learning with noisy labels for predicting transcription factor target genes using multi-omics dataset. (a) Gene expression (log2) of Nanog and cMyc in ESC, Epiblast-like cells (EpiLC), primordial germ cell-like cells at day 2 (PGCLCd2) and day 6 (PGCLCd6) in two biological replicates (r1 and r2). (b) Standardised expression profiles of all putative target genes or target genes identified by AdaEnsemble for Nanog and cMyc, respectively.

Figure 4(a) shows the gene expression of two master TFs, Nanog and cMyc, during ESC differentiation to Epiblast-like cells (EpiLC) and subsequently to primordial germ cell-like cells at day 2 (PGCLCd2) and then day 6 (PGCLCd6) in two biological replicates, i.e. r1 and r2. The expression of Nanog appears to be down-regulated from ESC to EpiLC and PGCLC2d, and up-regulated from PGCLC2d to PGCLC6d to its original level (Figure 4(a), upper panel). In contrast, the expression of cMyc is up-regulated from ESC to EpiLC, and down-regulated to its original level from EpiLC to PGCLCd2 and PGCLCd6 (Figure 4(a), lower panel). The standardised expression profile of all putative Nanog and cMyc target genes are shown in the first column of Figure 4(b) whereas the AdaEnsemble predicted ones are shown in the second and third columns. Compared to the expression profiles of all putative target genes, genes identified by AdaEnsemble for Nanog and cMyc resemble much better the expression of themselves (Figure 4(a)), suggesting functional relevance of these genes that are more likely to be genuinely regulated by Nanog or cMyc during the ESC differentiation process. These results illustrate the formulation of TF target gene prediction as a class label noise learning problem and the utility of AdaSampling in identifying true target genes from all putative candidate genes.

VII. Conclusion and Future Work

In this work, we extended an adaptive sampling (AdaSampling) approach to handle positive-unlabeled learning and learning with class label noise in the same framework. We evaluated the characteristics of AdaSampling using simulation studies and demonstrated its effectiveness in learning a single

or an ensemble of classifiers using multiple real-world benchmark datasets, and a range of base classification algorithms. Empirical results on synthetic and benchmark datasets verified the effectiveness and robustness of our proposed framework. We subsequently demonstrated the utility of the proposed framework in two novel bioinformatics applications in which we (1) identified novel kinase substrates, and (2) TF target genes from diverse biological systems using various high-throughput sequencing and mass spectrometry data.

Here, we considered the scenarios that noise are random within each class but asymmetric between classes. It is of interest in our future work to consider the scenarios that noise not at random (NNRA). Secondly, while it is clear that the performance of AdaSampling-based classification decrease with increasing data noise, in our future work, we will evaluate how different amounts and ratios (within and between classes) of mislabeled instances impact on the model quality. Thirdly, our future work will also look to incorporate imbalanced data classification and partial label learning into the proposed framework. Lastly, many more bioinformatics problems are confounded by the incomplete knowledge due to our increasing capacity to generate data but limited resource to annotate them. To the end of applications, our future work will be to extend and apply AdaSampling to address bioinformatics problems that can be formulated as positive-unlabeled learning and/or learning with class label noise.

ACKNOWLEDGMENT

This work is funded by an ARC/DECRA (DE170100759) to Pengyi Yang; an ARC/DP (DP170100654) to Jean Y.H. Yang, John T. Ormerod, and Pengyi Yang; an ARC/DP (DP130104591) to Albert Y. Zomaya; and a NHMRC/CDF (1111338) to Jean Y.H. Yang.

REFERENCES

[1] 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. ACM, 2008, pp. 213–220.

[2] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," IEEE transactions on neural networks and learning systems, vol. 25, no. 5, pp. 845–869, 2014.

[3] P. Yang, X.-L. Li, J.-P. Mei, C.-K. Kwoh, and S.-K. Ng, "Positive-unlabeled learning for disease gene identification," Bioinformatics, vol. 28, no. 20, pp. 2640–2647, 2012.

[4] S. Roy, A. F. Siahpirani, D. Chasman, S. Knaack, F. Ay, R. Stewart, M. Wilson, and R. Sridharan, "A predictive modeling approach for cell line-specific long-range regulatory interactions," Nucleic Acids Research, vol. 43, no. 18, pp. 8694–8712, 2015.

[5] F. Letouzey, F. Denis, and R. Gilleron, "Learning from positive and unlabeled examples," in Algorithmic Learning Theory. Springer, 2000, pp. 71–85.

[6] B. Calvo, P. Larrañaga, and J. A. Lozano, "Learning bayesian classifiers from positive and unlabeled examples," Pattern Recognition Letters, vol. 28, no. 16, pp. 2375–2384, 2007.

[7] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in Proceedings of the eleventh annual conference on Computational learning theory. ACM, 1998, pp. 92–100.

[8] F. Denis, R. Gilleron, and F. Letouzey, "Learning from positive and unlabeled examples," Theoretical Computer Science, vol. 348, no. 1, pp. 70–83, 2005.

[9] X. Li, S. Y. Philip, B. Liu, and S.-K. Ng, "Positive unlabeled learning for data stream classification." in SDM, vol. 9. SIAM, 2009, pp. 257–268.

[10] D. Angluin and P. Laird, "Learning from noisy examples," Machine Learning, vol. 2, no. 4, pp. 343–370, 1988.

[11] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," Artificial Intelligence Review, vol. 33, no. 4, pp. 275–306, 2010.

[12] 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 (ICML-15), 2015, pp. 125–134.

[13] P. Yang, W. Liu, and J. Yang, "Positive unlabeled learning via wrapper-based adaptive sampling," in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 3273–3279. [Online]. Available: https://doi.org/10.24963/ijcai.2017/457

[14] R. Kohavi and G. H. John, "Wrappers for feature subset selection," Artificial Intelligence, vol. 97, no. 1, pp. 273–324, 1997.

[15] X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in Proceedings of the 18th international joint conference on Artificial intelligence. Morgan Kaufmann Publishers Inc., 2003, pp. 587–592.

[16] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," in Data Mining, 2003. ICDM 2003. Third IEEE International Conference on. IEEE, 2003, pp. 179–186.

[17] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Learning to classify text from labeled and unlabeled documents," in Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence. American Association for Artificial Intelligence, 1998, pp. 792–799.

[18] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in ICML, vol. 2. Citeseer, 2002, pp. 387–394.

[19] M. C. du Plessis, G. Niu, and M. Sugiyama, "Analysis of learning from positive and unlabeled data," in Advances in Neural Information Processing Systems, 2014, pp. 703–711.

[20] W. S. Lee and B. Liu, "Learning with positive and unlabeled examples using weighted logistic regression," in Proceedings of the 20th International Conference on Machine Learning (ICML-03), 2003, pp. 448–455.

[21] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," Machine learning, vol. 39, no. 2-3, pp. 103–134, 2000.

[22] P. Yang, S. J. Humphrey, D. E. James, Y. H. Yang, and R. Jothi, "Positive-unlabeled ensemble learning for kinase substrate prediction from dynamic phosphoproteomics data," Bioinformatics, vol. 32, no. 2, pp. 252–259, 2016.

[23] W. Li, Q. Guo, and C. Elkan, "A positive and unlabeled learning algorithm for one-class classification of remote-sensing data," IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 2, pp. 717–725, 2011.

[24] F. Denis, R. Gilleron, and M. Tommasi, "Text classification from positive and unlabeled examples," in Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'02, 2002, pp. 1927–1934.

[25] S. S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques," The Knowledge Engineering Review, vol. 29, no. 03, pp. 345–374, 2014.

[26] F. Mordelet and J.-P. Vert, "A bagging svm to learn from positive and unlabeled examples," Pattern Recognition Letters, vol. 37, pp. 201–209, 2014.

[27] M. Claesen, F. De Smet, J. A. Suykens, and B. De Moor, "A robust ensemble approach to learn from positive and unlabeled data using svm base models," Neurocomputing, vol. 160, pp. 73–84, 2015.

[28] L. Breiman, "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.

[29] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," Artificial Intelligence Review, vol. 22, no. 3, pp. 177–210, 2004.

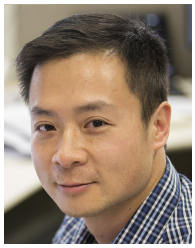
[30] M. Sabzevari, G. Martínez-Muñoz, and A. Suárez, "Small margin ensembles can be robust to class-label noise," Neurocomputing, vol. 160, pp. 18–33, 2015.

[31] J. Friedman, T. Hastie, R. Tibshirani et al., "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," The Annals of Statistics, vol. 28, no. 2, pp. 337–407, 2000.

[32] Y. Freund, "An adaptive version of the boost by majority algorithm," Machine Learning, vol. 43, no. 3, pp. 293–318, 2001.

[33] N. Manwani and P. Sastry, "Noise tolerance under risk minimization," IEEE Transactions on Cybernetics, vol. 43, no. 3, pp. 1146–1151, 2013.

[34] C. Scott, G. Blanchard, and G. Handy, "Classification with asymmetric label noise: Consistency and maximal denoising," in Conference On Learning Theory, 2013, pp. 489–511.
[35] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," Journal of Artificial Intelligence Research, vol. 11, pp. 131–167, 1999.
[36] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, "Enhancing data analysis with noise removal," IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 3, pp. 304–319, 2006.
[37] P. Jeatrakul, K. W. Wong, and C. C. Fung, "Data cleaning for classification using misclassification analysis," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 14, no. 3, pp. 297–302, 2010.
[38] A. L. Miranda, L. P. F. Garcia, A. C. Carvalho, and A. C. Lorena, "Use of classification algorithms in noise detection and elimination," in International Conference on Hybrid Artificial Intelligence Systems. Springer, 2009, pp. 417–424.
[39] U. Rebbapragada and C. E. Brodley, "Class noise mitigation through instance weighting," in European Conference on Machine Learning. Springer, 2007, pp. 708–715.
[40] C. Bouveyron and S. Girard, "Robust supervised classification with mixture models: Learning from data with uncertain labels," Pattern Recognition, vol. 42, no. 11, pp. 2649–2658, 2009.
[41] S. Verbaeten and A. Van Assche, "Ensemble methods for noise elimination in classification problems," in International Workshop on Multiple Classifier Systems. Springer, 2003, pp. 317–325.
[42] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise." ACML, vol. 20, pp. 97–112, 2011.
[43] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in Advances in Neural Information Processing Systems, 2013, pp. 1196–1204.
[44] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," IEEE Transactions on pattern analysis and machine intelligence, vol. 38, no. 3, pp. 447–461, 2016.
[45] E. Côme, L. Oukhellou, T. Denoeux, and P. Aknin, "Learning from partially supervised data using mixture models and belief functions," Pattern Recognition, vol. 42, no. 3, pp. 334–348, 2009.
[46] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," Journal of Machine Learning Research, vol. 12, no. 4, pp. 1501–1536, 2011.
[47] M.-L. Zhang, F. Yu, and C.-Z. Tang, "Disambiguation-free partial label learning," IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 10, pp. 2155–2167, 2017.
[48] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.
[49] L. Jiang, C. Li, and S. Wang, "Cost-sensitive bayesian network classifiers," Pattern Recognition Letters, vol. 45, no. 1, pp. 211–216, 2014.
[50] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, no. 1, pp. 321–357, 2002.
[51] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," IEEE Transactions on Cybernetics, vol. 47, no. 12, pp. 4263–4274, 2017.
[52] P. Yang, P. D. Yoo, J. Fernando, B. B. Zhou, Z. Zhang, and A. Y. Zomaya, "Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications," IEEE Transactions on Cybernetics, vol. 44, no. 3, p. 445, 2014.
[53] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," Advances in Large Margin Classifiers, vol. 10, no. 3, pp. 61–74, 1999.
[54] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml
[55] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," Journal of Machine learning research, vol. 7, no. Jan, pp. 1–30, 2006.
[56] J. Bootkrajang and A. Kabán, "Classification of mislabelled microarrays using robust sparse logistic regression," Bioinformatics, vol. 29, no. 7, pp. 870–877, 2013.
[57] P. Yang, X. Zheng, V. Jayaswal, G. Hu, J. Y. H. Yang, and R. Jothi, "Knowledge-based analysis for detecting key signaling events from time-series phosphoproteomics data," PLoS Comput Biol, vol. 11, no. 8, p. e1004403, 2015.
[58] S. J. Humphrey, G. Yang, P. Yang, D. J. Fazakerley, J. Stöckli, J. Y. Yang, and D. E. James, "Dynamic adipocyte phosphoproteome reveals that akt directly regulates mtorc2," Cell Metabolism, vol. 17, no. 6, pp. 1009–1020, 2013.
[59] P. V. Hornbeck, B. Zhang, B. Murray, J. M. Kornhauser, V. Latham, and E. Skrzypek, "Phosphositeplus, 2014: mutations, ptms and recalibrations," Nucleic Acids Research, vol. 43, no. D1, pp. D512–D520, 2015.
[60] S. J. Humphrey, S. B. Azimifar, and M. Mann, "High-throughput phosphoproteomics reveals in vivo insulin signaling dynamics," Nature Biotechnology, vol. 33, no. 9, pp. 990–995, 2015.
[61] A. J. Oldfield, P. Yang, A. E. Conway, S. Cinghu, J. M. Freudenberg, S. Yellaboina, and R. Jothi, "Histone-fold domain protein nf-y promotes chromatin accessibility for cell type-specific master transcription factors," Molecular Cell, vol. 55, no. 5, pp. 708–722, 2014.
[62] P. Yang, A. Oldfield, T. Kim, A. Yang, J. Y. H. Yang, and J. W. Ho, "Integrative analysis identifies co-dependent gene expression regulation of brg1 and chd7 at distal regulatory sites in embryonic stem cells," Bioinformatics, vol. 33, no. 13, pp. 1916–1920, 2017.
[63] V. A. Spencer, J.-M. Sun, L. Li, and J. R. Davie, "Chromatin immunoprecipitation: a tool for studying histone acetylation and transcription factor binding," Methods, vol. 31, no. 1, pp. 67–75, 2003.
[64] J. Zhang, H. M. Poh, S. Q. Peh, Y. Y. Sia, G. Li, F. H. Mulawadi, Y. Goh, M. J. Fullwood, W.-K. Sung, X. Ruan et al., "Chia-pet analysis of transcriptional chromatin interactions," Methods, vol. 58, no. 3, pp. 289–299, 2012.
[65] K. Kurimoto, Y. Yabuta, K. Hayashi, H. Ohta, H. Kiyonari, T. Mitani, Y. Moritoki, K. Kohri, H. Kimura, T. Yamamoto et al., "Quantitative dynamics of chromatin remodeling during germ cell specification from mouse embryonic stem cells," Cell Stem Cell, vol. 16, no. 5, pp. 517–532, 2015.
[66] X. Chen, H. Xu, P. Yuan, F. Fang, M. Huss, V. B. Vega, E. Wong, Y. L. Orlov, W. Zhang, J. Jiang et al., "Integration of external signaling pathways with the core transcriptional network in embryonic stem cells," Cell, vol. 133, no. 6, pp. 1106–1117, 2008.
[67] W. A. Whyte, D. A. Orlando, D. Hnisz, B. J. Abraham, C. Y. Lin, M. H. Kagey, P. B. Rahl, T. I. Lee, and R. A. Young, "Master transcription factors and mediator establish super-enhancers at key cell identity genes," Cell, vol. 153, no. 2, pp. 307–319, 2013.
[68] Y. Zhang, C.-H. Wong, R. Y. Birnbaum, G. Li, R. Favaro, C. Y. Ngan, J. Lim, E. Tai, H. M. Poh, E. Wong et al., "Chromatin connectivity maps reveal dynamic promoter-enhancer long-range associations," Nature, vol. 504, no. 7479, pp. 306–310, 2013.
[69] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," Machine Learning, vol. 51, no. 2, pp. 181–207, 2003.



Pengyi Yang is a Senior Lecturer and an Australian Research Council (ARC)/Discovery Early Career Researcher Award (DECRA) Fellow at the School of Mathematics and Statistics, University of Sydney. He received his Ph.D. from the School of Information Technologies, University of Sydney, in 2012. His was a Research Fellow at National Institutes of Health (NIH), USA, from 2012 to 2015 before joining the School of Mathematics and Statistics at University of Sydney.

His reseach is at the interface of machine learning and data mining, and their applications to bioinformatics and computational biology. He has over 40 publications in the broad area of bioinformatics and systems biology. For his scientific contribution to the interdisciplinary research in bioinformatics, he was awarded the J G Russell Award from Australian Academy of Science.

John T. Ormerod received the Ph.D. degree in Statistics from the University of New South Wales, Sydney, Australia, in 2008. He is currently a Senior Lecturer at the School of Mathematics and Statistics at the University of Sydney. He is also an associate investigator with the ARC Centre of Excellence for Mathematical and Statistical Frontiers at the University of Melbourne. He is a former ARC DECRA Fellow. His current research interests include model selection, variational Bayes, semiparametric regression, and statistical bioinformatics.

Wei Liu is a Senior Lecturer at the Advanced Analytics Institute, School of Software, Faculty of Engineering and IT, the University of Technology Sydney, Australia. He obtained his PhD in Data Mining research from the University of Sydney. Before joining UTS, he was a Research Fellow at the University of Melbourne, and then a Machine Learning Researcher at NICTA working with the transportation industry. He works in broad areas of data mining and machine learning, and has published over 50 papers on topics of deep learning, game theory, tensor factorization, graph mining, causal inference, and anomaly detection.

Chendong Ma received her Bachelor of Science with Honours from the School of Mathematics and Statistics, University of Sydney, in 2017. Her Honours project is related to statistical learning and its application to biology and bioinformatics. Her research interests are in applied statistics and data analysis, and their application in machine learning and data visualisation.

Albert Y. Zomaya is the Chair Professor of High Performance Computing & Networking in the School of Information Technologies, University of Sydney, and he also serves as the Director of the Centre for Distributed and High Performance Computing. Professor Zomaya published more than 600 scientific papers and articles and is author, co-author or editor of more than 20 books. He is the Founding Editor in Chief of the IEEE Transactions on Sustainable Computing and serves as an associate editor for more than 20 leading journals. Professor Zomaya served as an Editor in Chief for the IEEE Transactions on Computers (2011-2014).

Professor Zomaya is the recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award (2011), the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing (2011), and the IEEE Computer Society Technical Achievement Award (2014). He is a Chartered Engineer, a Fellow of AAAS, IEEE, and IET. Professor ZomayaâĂŹs research interests are in the areas of parallel and distributed computing and complex systems.

Jean Y.H. Yang is a Professor in School of Mathematics and Statistics, University of Sydney, and a NHMRC CDF Fellow. She is an applied statistician with expertise in statistical bioinformatics. She was awarded the 2015 Moran Medal in statistics from the Australian Academy of Science in recognition of her work on developing methods for molecular data arising in cutting edge biomedical research. Her research stands at the interface between medicine and methodology development and has centered on the development of methods and the application of statistics to problems in -omics and biomedical research. In particular, her focus is on developing methods for integrating omics and clinical data to answer a variety of scientific questions. As a statistician who works in the bioinformatics area, she enjoys research in a collaborative environment, working closely with scientific investigators from diverse backgrounds.

Her research interests and expertise include statistical bioinformatics; applied statistics; statistical machine learning; complex data analytics; integrative analysis of omics data.