A Robust Ensemble Approach to Learn From Positive and Unlabeled Data Using SVM Base Models

Marc Claesen\textsuperscript{a,\ast}, Frank De Smet\textsuperscript{b,1}, Johan A. K. Suykens\textsuperscript{a}, Bart De Moor\textsuperscript{a}

\textsuperscript{a}KU Leuven, ESAT – STADIUS/iMinds Medical IT
Kasteelpark Arenberg 10, box 2446
3001 Leuven, Belgium

\textsuperscript{b}KU Leuven, Department of Public Health and Primary Care, Environment and Health
Kapucijnenvoer 35 blok d, box 7001
3000 Leuven, Belgium

Abstract

We present a novel approach to learn binary classifiers when only positive and unlabeled instances are available (PU learning). This problem is routinely cast as a supervised task with label noise in the negative set. We use an ensemble of SVM models trained on bootstrap subsamples of the training data for increased robustness against label noise. The approach can be considered in a bagging framework which provides an intuitive explanation for its mechanics in a semi-supervised setting. We compared our method to state-of-the-art approaches in a simulation based on the MNIST digit recognition data set. The included benchmark comprises three settings with increasing label noise: (i) fully supervised, (ii) PU learning and (iii) PU learning with false positives. Our approach shows a marginal improvement over existing methods in the second setting and a significant improvement in the third.

Keywords: classification, semi-supervised learning, ensemble learning, support vector machine

\ast Corresponding author

Email addresses: marc.claesen@esat.kuleuven.be (Marc Claesen), frank.desmet@cm.be (Frank De Smet), johan.suykens@esat.kuleuven.be (Johan A. K. Suykens), bart.demoor@esat.kuleuven.be (Bart De Moor)

1 Frank De Smet is a member of the medical management department of the National Alliance of Christian Mutualities.

Preprint submitted to Neurocomputing February 14, 2014
1. Introduction

Training binary classifiers on positive and unlabeled data is referred to as PU learning (Liu et al., 2003). The absence of known negative training instances warrants appropriate learning methods. Inaccurate label information can be more problematic than attribute noise (Zhu and Wu, 2004). Specialised PU learning approaches are recommended when (i) negative labels cannot be acquired, (ii) the training data contains a large amount of false negatives or (iii) the positive set has many outliers.

Practical applications of PU learning typically feature large, imbalanced training sets with a small amount of labeled (positive) and a large amount of unlabeled training instances. The PU learning problem arises in various settings, including web page classification (Yu et al., 2002), intrusion detection (Lazarevic et al., 2003) and bioinformatics tasks such as variant prioritization (Sifrim et al., 2013), gene prioritization (Aerts et al., 2006; Mordelet and Vert, 2011) and virtual screening of drug compounds (Shoichet, 2004).

Though these applications share a common underlying learning problem, the final evaluation criteria may be fundamentally different. For instance, in prioritization one wishes to obtain high precision since highly ranked targets may be subjected to further biological analysis. Intrusion detection, on the other hand, necessitates high recall to ensure that no anomalies go unnoticed.

Following Mordelet and Vert (2010), we will use the term contamination to refer to the fraction of mislabeled instances in a given set. We will denote the positive and unlabeled training instances by $P$ and $U$, respectively. Contamination in $P$ refers to false positives while contamination in $U$ refers to the presence of positives in $U$. Usually $U$ contains mostly true negative instances (e.g. contamination below 0.5) and $P$ is assumed to be uncontaminated.

The distributions of the positive and a contaminated unlabeled set overlap even when those of the positive and underlying negative sets do not, which makes classification more difficult compared to a traditional supervised setting. Elkan and Noto (2003) and Blanchard et al. (2010) report statistical approaches to estimate the contamination of the unlabeled set and additionally show that distinguishing positives from unlabeled instances is a valid proxy for distinguishing positives from negatives.

The assumption in PU learning that $P$ is uncontaminated may be violated in applications due to various reasons (Frénay and Verleysen, in press). Additionally, outliers in the positive set may have a similar effect on classification performance (Pechenizkiy et al., 2006). We propose a novel PU learning
method that is less vulnerable to potential contamination in $P$ called the robust ensemble of support vector machines (RESVM). RESVM is compared to other methods in a benchmark based on the MNIST data set (LeCun et al., 1998).

2. Related work

PU learning approaches can be split into two main conceptual categories: (i) approaches that account for the contamination of the unlabeled set explicitly by modeling the label noise and (ii) approaches that try to infer an uncontaminated (negative) subset $\mathcal{\hat{N}}$ from $\mathcal{U}$ and then train supervised algorithms to distinguish $P$ from $\mathcal{\hat{N}}$. When very few labeled examples are available, the structure within the data is the main source of information which can be exploited by semi-supervised clustering techniques (Alzate and Suykens, 2012).

*Accounting for the contamination of $\mathcal{U}$ in the modeling process.* A first way to achieve this is by weighting individual data points, such as in weighted logistic regression (Elkan and Noto, 2008; Lee and Liu, 2003). Another approach is by changing the penalties on misclassification during training, as is done in class-weighted SVM (Liu et al., 2003), bagging SVM (Mordelet and Vert, 2010) and RT-SVM (Liu et al., 2005).

*Inferring an uncontaminated subset from $\mathcal{U}$.* Alternative approaches exist which try to infer a negative set $\mathcal{\hat{N}}$. After the inferential step, binary classifiers are trained to distinguish $P$ from $\mathcal{\hat{N}}$ in a supervised fashion. Examples of such two-step approaches include S-EM (Liu et al., 2002), mapping convergence (MC) (Yu, 2005) and ROC-SVM (Li and Liu, 2003).

*Class-weighted SVM and related approaches.* The approach we suggest belongs to the first class of methods and is closely related to class-weighted SVM and bagging SVM (which uses class-weighted SVM internally). We will discuss both of these approaches in more detail before moving on to the proposed method. We evaluated our method compared to both class-weighted SVM and bagging SVM.

2.1. Class-weighted SVM

Class-weighted SVM (CWSVM) is a supervised technique in which the penalty for misclassification differs per class. Liu et al. (2003) first applied
class-weighted SVM for PU learning by considering the unlabeled set to be negative with noise on its labels. CWSVM is trained to distinguish $P$ from $U$. During training, misclassification of positive instances is penalized more than misclassification of unlabeled instances to emphasize the higher degree of certainty on positive labels. In the context of PU learning, the optimization problem for training CWSVM can be written as:

$$
\min_{\alpha, \xi, b} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \kappa(x_i, x_j) + C_P \sum_{i \in P} \xi_i + C_U \sum_{i \in U} \xi_i, \quad (1)
$$

subject to

$$
y_i \left( \sum_{j=1}^{N} \alpha_j y_j \kappa(x_i, x_j) + b \right) \geq 1 - \xi_i, \quad i = 1, \ldots, N,
$$

$$
\xi_i \geq 0, \quad i = 1, \ldots, N,
$$

with $\alpha \in \mathbb{R}^N$ the support values, $y \in \{-1, +1\}^N$ the label vector, $\kappa(\cdot, \cdot)$ the kernel function, $b$ the bias term and $\xi \in \mathbb{R}^N$ the slack variables. The misclassification penalties $C_P$ and $C_U$ require tuning ($C_P > C_U$ to emphasize known labels). SVM formulations with unequal penalties across classes have been used previously to tackle imbalanced data sets (Osuna et al., 1997).

2.2. Bagging SVM

Mordelet and Vert introduce bagging SVM as a meta-algorithm which consists of aggregating classifiers trained to discriminate $P$ from a small, random subsample of $U$ (Mordelet and Vert, 2010). They posit that PU learning problems have a particular structure that leads to instability of classifiers, namely the sensitivity of classifiers to the contamination of the unlabeled set. Bagging is a common technique used to improve the performance of unstable classifiers (Breiman, 1996).

In bagging SVM, random subsamples of $U$ are drawn and CWSVM classifiers are trained to discriminate $P$ from each subsample. By subsampling $U$, the contamination is varied. This induces variability in the classifiers which the aggregation procedure can then exploit. The size of the bootstrap subsample of $U$ is a tuning parameter in bagging SVM. The ratio $C_P / C_U$ is fixed so that the following holds:

$$
|P| \times C_P = n_U \times C_U, \quad (2)
$$

with $|P|$ the size of the positive set and $n_U$ the size of subsamples from the unlabeled set. This choice of weights is common in unbalanced settings.
All base models in bagging SVM classify the full set of positives against a subset of unlabeled instances and use a high misclassification penalty on the positives similar to CWSVM.

3. Robust Ensemble of SVMs

Breiman (1996) introduced bagging as a technique to construct strong ensembles from a set of base models. In bagging, the ensemble models use majority voting to aggregate decisions of base models which are trained on bootstrap resamples of the training set. From a Bayesian point of view, bagging can be interpreted as a Monte Carlo integration over an approximated posterior distribution (Rao and Tibshirani, 1997). The essential problem in combining classifiers is growing a suitably diverse ensemble of base classifiers which can be done in various ways (Breiman, 2000).

We propose a new technique called the robust ensemble of SVMs (RESVM). RESVM is an ensemble method using CWSVM base models as discussed in Section 2.1 with voting aggregation. RESVM constructs base model training sets by bootstrap resampling both positive and unlabeled training instances separately, both of which may be contaminated.

Resampling contaminated sets with replacement induces variability in contamination across resamples. The variability in contamination between resamples increases for increasing contamination of the original set. Due to the law of large numbers the contamination in bootstrap resamples of increasing size converges to the expected contamination, which equals that of the original set that is being resampled. As a result, the variability in contamination decreases for increasing resample size. Figure 1 illustrates this property empirically based on 20,000 repeated measurements for each resample size: the expected value (mean) equals the original contamination, but the variability in resample contamination is a function of resample size.

The key difference between RESVM and bagging SVM is that the former resamples \( P \) in addition to \( U \) to increase variability between base models. RESVM additionally features an extra degree of freedom to control the relative misclassification penalty between positive and unlabeled instances, which is fixed in bagging SVM. Mordelet and Vert (2010) report no significant changes when varying the relative penalty in bagging SVM, though our experiments show that it is important in RESVM (see \( w_{\text{pos}} \) in Table 3).

Varying the contamination between base model training sets induces variability between base models without increasing bias. This observation en-
ables us to create a diverse set of base models by resampling both $\mathcal{P}$ and $\mathcal{U}$. The variance reduction of bagging is an excellent mechanism to exploit the variability of base models based on resampling (Bauer and Kohavi, 1999; Breiman, 2000). Breiman (1996) emphasizes that base model instability is an important factor in the success of bagging. Bagging schemes typically use instable base models like decision trees or neural networks to obtain variability whereas base model variability in RESVM is a direct result of resampling training sets from a contaminated set.

Grandvalet (2004) explains that base model instability is not related to the intrinsic variability of a predictor but rather to the presence of influential instances in a data set for a given predictor (base model). The effect of bagging is explained as equalizing the influence of instances, which is beneficial when highly influential instances are harmful for the predictor’s accuracy. When resampling contaminated sets, the mislabeled instances are influential on CWSVM base models and deteriorate their performance. Following Grandvalet (2004), bagging lowers the influence of mislabeled instances and
thus improves robustness against contamination in the context of RESVM.

3.1. Training parameters for RESVM

RESVM takes possible contamination of the positive training set into account by design. Because the contamination between $\mathcal{P}$ and $\mathcal{U}$ can vary, the ability to vary the size of resamples from $\mathcal{P}$ and $\mathcal{U}$ separately is required. This results in two tuning parameters: $n_{\text{pos}}$ and $n_{\text{unl}}$.

RESVM additionally inherits two tuning parameters from its CWSVM base models, namely misclassification penalties for both classes and potential parameters of the kernel function $\kappa(\cdot, \cdot)$. We define the penalties (see Eq. (1)) based on two additional tuning parameters $C_{\mathcal{U}}$ and $w_{\text{pos}}$:

$$C_{\mathcal{P}} = C_{\mathcal{U}} \times w_{\text{pos}} \times \frac{n_{\text{unl}}}{n_{\text{pos}}}. \tag{3}$$

The final tuning parameter $n_{\text{models}}$ is the number of base models to include in an ensemble. Using more base models improves the stability of the ensemble at a linear increase in computational cost for training. In practice, this parameter can be determined during training by monitoring the predictive performance of the ensemble while iteratively adding more base models instead of doing a grid search.

The RESVM training approach has been summarised in Algorithm 1. The algorithm uses 5 hyperparameters plus additional kernel parameters.

3.2. Ensemble decision values

RESVM uses majority voting to aggregate base model predictions. By default, the returned label is the one predicted by most base models. The fraction of positive votes for a test instance $x$ can be written as:

$$v(x) = \frac{n_{\text{models}} + \sum_{i=1}^{n_{\text{models}}} \text{sgn}(\psi^{(i)}(x))}{2n_{\text{models}}}, \tag{4}$$

where $\text{sgn}(\cdot)$ is the sign function and $\psi^{(i)}$ denotes the decision function of SVM base model $i$ with codomain $\mathbb{R}$. $v(\cdot)$ has the interval $[0, 1]$ as codomain.

The RESVM decision value for a test instance $x$ is defined as the fraction of votes in favor of the positive class $v(x)$ unless the result is unanimous. In the case of a unanimous vote, the ensemble decision value is based on the decision values of its base models to increase the model’s ability to differentiate. In case of a unanimous negative vote, the sum of the decision values
**Algorithm 1**: Training procedure for RESVM.

**Data**: \( P \): the set of positive instances.

\( U \): the set of unlabeled instances.

**Input**:
- \( n_{\text{models}} \): number of base models to include in the ensemble.
- \( n_{\text{unl}} \): size of bootstrap resamples of \( U \).
- \( n_{\text{pos}} \): size of bootstrap resamples of \( P \).
- \( C_U \): misclassification penalty for \( U \) in class-weighted SVM.
- \( w_{\text{pos}} \): positive misclassification penalty coefficient.
- \( \kappa(\cdot, \cdot) \): kernel function to be used by base models.

**Output**: \( \Omega \): RESVM with \( n_{\text{models}} \) base models.

\[ \begin{align*}
\Omega & \leftarrow \emptyset; \\
C_P & \leftarrow M \times w_{\text{pos}} \times \frac{n_{\text{unl}}}{n_{\text{pos}}}; \\
\text{for } i & \leftarrow 1 \text{ to } n_{\text{models}} \text{ do} \\
& \quad \text{// create base model training set from } P \text{ and } U. \\
& \quad P^{(i)} \leftarrow \text{sample } n_{\text{pos}} \text{ instances from } P \text{ with replacement}; \\
& \quad U^{(i)} \leftarrow \text{sample } n_{\text{unl}} \text{ instances from } U \text{ with replacement}; \\
& \quad \text{// train CWSVM base model } \psi^{(i)} \text{ and add to ensemble } \Omega. \\
& \quad \psi^{(i)} \leftarrow \text{train BSVM for } P^{(i)} \text{ vs. } U^{(i)} \text{ (parameters } C_P, C_U, \kappa); \\
& \quad \Omega \leftarrow \{ \Omega, \psi^{(i)} \}; \\
\end{align*} \]

of the base models is taken (each SVM base model decision value is negative in this case). In case of a unanimous positive vote, the sum of the decision values of the base models (all positive) plus one is taken. The decision value \( d(\cdot) \) has codomain \( \mathbb{R} \) and is computed as follows:

\[ d(x) = \begin{cases} 
  v(x) & \text{if } 0 < v(x) < 1, \\
  \sum_{i=1}^{n_{\text{models}}} \psi^{(i)}(x) & \text{if } v(x) = 0, \\
  1 + \sum_{i=1}^{n_{\text{models}}} \psi^{(i)}(x) & \text{if } v(x) = 1.
\end{cases} \quad (5) \]

The resulting label for a given decision threshold \( T \) can be written as follows:

\[ l(x) = \text{sgn}(d(x) - T). \quad (6) \]

The default decision value threshold for positive classification is \( T = 0.5 \) (this is majority voting, e.g. positive iff more than half of all base models predict positive).
4. Benchmark setup

RESVM has been compared to class-weighted SVM (CWSVM) and bagging SVM (BAG). All SVM models we used in our experiments are linear (standalone CWSVM and all base models in ensembles). The ensemble models are implicitly nonlinear due the use of a majority voting aggregation scheme as shown in Equations (1) to (6).

Data set. The MNIST digit recognition data set was used as a basis for the benchmark (LeCun et al., 1998). This data set contains 10 classes (one for each digit), 780 features, 60,000 training instances and 10,000 test instances. We used a version of the data that is made available by the LIBSVM authors without further preprocessing.\(^2\)

Benchmark approach. The multiclass problem was cast into 10 binary classification problems by doing one-vs-all classification for every digit. In each binary problem we dispose of \(\pm6.000\) true positive and \(\pm54.000\) true negative training instances. From these sets of true positives and true negatives we sampled the training sets used in the benchmark without replacement so a given instance does not occur multiple times in a single training set.

Every binary learning task was repeated 10 times for each digit to get reliable assessments of all methods. The repetitions include resampling of training sets from the true positives and true negatives. Contamination was introduced at random where applicable. Figure 2 illustrates all steps in a single benchmark iteration for one digit in a particular learning setting.

Training set construction. All training sets we constructed contain 50 positively labeled instances and 2,000 unlabeled instances. These training sets resemble practical applications which typically feature few positive and many unlabeled instances. The following learning settings were considered:

1. **Fully supervised**: no contamination in \(P\) and \(U\).
2. **PU learning**: no contamination in \(P\), 10\% contamination in \(U\).
3. **noisy PU learning**: 10\% contamination in \(P\) and \(U\).

\(^2\)http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#mnist
Hyperparameter selection. In every iteration, hyperparameters of each method were tuned using 10-fold cross-validation over a grid of parameter tuples. To ensure a fair comparison, one set of folds is generated in each iteration and used by all methods. Since negative labels are unavailable in PU learning, we used the following score function in all learning settings which only requires positive labels for hyperparameter selection [Lee and Liu, 2003]:

\[
\text{pu score} = \frac{\text{precision} \times \text{recall}}{Pr(y = 1)} = \frac{\text{recall}^2}{Pr(\hat{y} = 1)},
\]

where \(Pr(y = 1)\) is the fraction of known positive labels in the predicted set and \(Pr(\hat{y} = 1)\) is the fraction of positive predictions made by the classifier. Note that this score function is not ideal when \(P\) is contaminated, though we obtained good results even in that setting.

During cross-validation we used 15 base models for both bagging SVM and RESVM. The following parameters were tuned per method: (CWSVM) \(C_P\) and \(C_U\), (BAG) \(C_U\) and \(n_U\) and (RESVM) \(C_U\), \(w_{pos}\), \(n_{pos}\) and \(n_{unl}\).

Performance assessment. Models are trained with the optimal hyperparameters on the full training set and subsequently tested on the independent test set with 10,000 instances. The final bagging SVM and RESVM models contained 50 base models. Note that the test set does not change between experiments. We use the known test labels to compute the area under the Precision-Recall curve (AUC) for each model. We opted to use PR curves because they capture the performance of interest of models over their entire operating range and work well for unbalanced data [Davis and Goadrich, 2006].
The results of our experiments are reported in Tables 1 and 2. We additionally report area under the ROC curve for all experiments in Appendix A.

Implementation details. We used the class-weighted SVM implementation available in LIBLINEAR (Fan et al., 2008). Bagging SVM and RESVM were implemented using the EnsembleSVM library (Claesen et al., in press). The decision values of bagging SVM used to compute PR and ROC curves were defined in the same way as for RESVM (see Section 3.2).

5. Results and discussion

In this Section we summarize all results of the benchmark comparing class-weighted SVM (CWSVM), bagging SVM (BAG) and the robust ensemble of SVMs (RESVM). To illustrate the implicit mechanisms of RESVM we have included the optimal hyperparameters that were found using cross-validation for each setting in Table 3.

5.1. Results for supervised classification

In the first setting we compare the methods in a fully supervised setting. In these experiments both \(\mathcal{P}\) and \(\mathcal{U}\) are uncontaminated. The results in Table 1 show that all methods obtain comparable results for all digits except 8 (where CWSVM performs very poorly). Bagging SVM and RESVM show a marginal improvement over class-weighted SVM. The performance difference could be caused by the fact we used linear class-weighted SVM while both ensemble models implicitly yield nonlinear decision boundaries.

The overall good results confirm that the score function in Equation (7) is a good choice for tuning. In these supervised experiments we could have used a traditional score like accuracy or F-measure, but these would no longer be useful in the next experiments. The performance in these supervised experiments can be considered an objective baseline for comparison in the PU learning setting since only levels of contamination are varied.

5.2. Results for PU learning

The results of our experiments in PU learning settings are shown in Table 2. In the pure PU learning setting, \(\mathcal{P}\) is uncontaminated but the contamination in \(\mathcal{U}\) is set to 10%. Class-weighted SVM exhibits the largest loss in performance between supervised learning and pure PU learning based on PR curves. Bagging SVM and RESVM maintain very good performance in
Table 1: 95% confidence intervals for the mean area under the Precision-Recall curve in percent on MNIST test set for each digit in a fully supervised one-vs-all setup. The results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $AUC_{RESVM} > AUC_{BAG}$ is included using the following result coding: • $p < 0.05$, •• $p < 0.01$ and ••• $p < 0.001$.

the pure PU learning setting compared to supervised. RESVM consistently exhibits the best performance, though the improvement over bagging SVM is marginal. The results additionally show that the performance drop between supervised and pure PU learning is much lower for RESVM than for both other approaches.
Table 2: 95% confidence intervals for the mean area under the Precision-Recall curve in percent on MNIST test set for each digit in a one-vs-all setup. Models are trained in the following settings PU learning settings: (i) classic PU learning setting (10% false negatives, no false positives) and (ii) a noisy PU learning setting (10% false positives and 10% false negatives). In each setting the results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $AUC_{RESVM} > AUC_{BAG}$ is included using the following result coding: • $p < 0.05$, •• $p < 0.01$ and • • • $p < 0.001$.

| digit | PU learning |       |       |       | noisy PU learning |       |       |       |
|-------|-------------|-------|-------|-------|-------------------|-------|-------|-------|
|       | CWSVM       |  BAG  | RESVM | $p$   | CWSVM             |  BAG  | RESVM | $p$   |
| 0     | 74.2–78.6   | 89.8–94.1 | 94.2–95.6 | • | 57.5–63.5 | 70.5–84.2 | 90.7–93.2 | • • • |
| 1     | 88.1–91.4   | 94.8–97.0 | 96.2–97.4 | •• | 80.2–84.0 | 91.5–94.9 | 96.4–97.8 | • • • |
| 2     | 52.9–60.3   | 74.1–80.8 | 84.1–86.7 | • • • | 41.6–49.9 | 58.5–68.8 | 77.8–84.0 | • • • |
| 3     | 54.2–61.0   | 74.2–81.5 | 81.8–86.1 | • | 43.0–49.1 | 56.3–66.6 | 78.4–82.6 | • • • |
| 4     | 55.6–61.2   | 70.9–79.4 | 83.3–85.6 | •• | 51.0–57.9 | 66.3–75.6 | 78.5–83.4 | • • • |
| 5     | 52.9–57.8   | 61.8–71.2 | 70.4–73.3 | • | 39.0–47.2 | 52.9–61.7 | 69.0–71.7 | • • • |
| 6     | 65.8–68.2   | 86.1–90.0 | 89.4–92.4 | • | 50.7–58.4 | 71.2–81.8 | 87.1–91.2 | • • • |
| 7     | 71.8–76.2   | 85.2–89.7 | 90.7–92.1 | • • • | 56.9–60.0 | 65.5–78.7 | 88.1–90.1 | • • • |
| 8     | 32.8–39.3   | 62.1–68.1 | 72.3–75.3 | • • | 29.7–36.1 | 49.4–59.4 | 70.0–73.7 | • • • |
| 9     | 49.1–51.9   | 63.6–71.3 | 73.6–76.8 | • • • | 41.2–44.8 | 50.8–59.3 | 67.2–71.7 | • • • |

Table 2: 95% confidence intervals for the mean area under the Precision-Recall curve in percent on MNIST test set for each digit in a one-vs-all setup. Models are trained in the following settings PU learning settings: (i) classic PU learning setting (10% false negatives, no false positives) and (ii) a noisy PU learning setting (10% false positives and 10% false negatives). In each setting the results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $AUC_{RESVM} > AUC_{BAG}$ is included using the following result coding: • $p < 0.05$, •• $p < 0.01$ and • • • $p < 0.001$. 
In the noisy PU learning setting we violated the assumption of an uncontaminated positive training set on purpose and created 10% contamination in both $P$ and $U$. The results are as expected: both class-weighted and bagging SVM are vulnerable to contamination in $P$ and show very large performance drops. We believe this is due to the use of high misclassification penalties for training instances in $P$ without any resampling to account for potential false positives. In bagging SVM this leads to a systematic bias in all base models. RESVM is affected much less because resampling $P$ removes systematic bias in base models. The results clearly show that RESVM is more robust to false positives, yielding a much lower drop in predictive performance. The performance difference between bagging SVM and RESVM is statistically significant for all digits. RESVM is the clear winner in this benchmark.

The confidence intervals of AUC for RESVM are consistently narrower than those of both other approaches in both PU learning settings (often less than half as wide). The confidence interval widths for RESVM also have fewer outliers than the others. Even though RESVM base models have more variability compared to bagging SVM base models, the performance of RESVM is more reliable. The increased reliability of RESVM in every experiment is an important practical advantage since assessing different models is not trivial outside of simulation studies (e.g. when no negative labels are available). The consistency in results leads us to hypothesize this behaviour can be generalized beyond our experimental setup but further investigation is required.

As an illustration, we have plotted the performance curves of one iteration with digit 7 as positive in the noisy PU learning setting in Figure 3. Even though the improvement of RESVM compared to both other approaches is smallest for this digit (see Table 2), the PR curve of RESVM completely dominates both others. In this experiment, RESVM is best regardless of design priorities (high recall or high precision). Because the PR curve of RESVM completely dominates the others we know that its ROC curve does too (Davis and Goadrich, 2006).

The PR curves in Figure 3a show that the RESVM model maintains perfect precision at more than twice the recall of BAG and both ensemble approaches completely outperform CWSVM. Maximizing precision at a given level of recall is crucial in ranking applications such as gene and variant prioritization (Aerts et al., 2006; Sifrim et al., 2013).
5.3. RESVM optimal parameters

As an illustration of the implicit mechanism of RESVM we show some of the optimal tuning parameters for every setting in Table 3. These parameters were obtained by performing 10-fold cross-validation on the training set.

An interesting observation is that the size of the training sets that are being used decreases for increasing contamination. Increasing label noise induces RESVM to favor smaller base model training sets for which the variability in contamination is larger (see Figure 1). Though this may appear counterintuitive, bagging approaches are known to exhibit a bias-variance tradeoff (Bauer and Kohavi, 1999) for which using weaker base models with increased variability may yield better ensembles (Keijzer and Babovic, 2000).

The optimal value of the misclassification penalty for positive training instances relative to unlabeled instances, \( w_{\text{pos}} \), changes between learning settings (see Equation (3)). It exhibits expected behaviour: the maximum value is obtained when the certainty on \( P \) relative to \( U \) is largest (e.g. the pure PU learning setting). This parameter implicitly balances empirical certainty on \( P \) and \( U \) and is an important degree of freedom in RESVM. In bagging SVM, this parameter is implicitly fixed to 1 via Equation (2) (Mordelet and Vert, 2010).
|          | supervised | PU learning | noisy PU |
|----------|------------|-------------|----------|
| $n_{\text{pos}}$ | 10 20 20 20 10 20 20 20 10 16.0 | 10 10 10 10 5 10 10 10 10 9.5 | 10 8 10 5 10 15 5 10 10 9.2 |
| $n_{\text{unl}}/n_{\text{pos}}$ | 10 10 10 10 10 10 10 10 10 10.0 | 5 5 5 5 5 10 8 10 10 6.8 | 5 5 5 8 5 8 5 8 5 6.0 |
| $w_{\text{pos}}$ | 1.6 1.6 1.6 4.8 3.2 3.2 3.2 1.6 4.8 3.2 | 6.4 6.4 3.2 12.8 6.4 3.2 2.4 3.2 3.2 6.4 | 4.8 6.4 3.2 6.4 3.2 4.3 4.8 1.6 2.4 2.1 |

Table 3: Medians of optimal hyperparameters per digit obtained via cross-validation and mean of all medians per setting. The normalized relative weight on positives versus unlabeled instances ($w_{\text{pos}}$) is associated with the relative size and contamination of the positive and unlabeled training sets.

### 6. Conclusion

The robust ensemble of SVMs works well in all settings we have analyzed. RESVM promotes base model variability more than existing approaches due to an improved resampling scheme. In a pure PU learning setting the average improvement over existing methods is marginal though RESVM classifiers exhibit lower variance in performance making it more reliable. We pose that this improvement is due to an improvement in bagging efficiency when the variability between base models is increased.

RESVM shows very strong improvements over other approaches when the positive set is contaminated. Our empirical results show that RESVM has a tendency to favor smaller base models with increased variability when label noise increases. The effect of label noise is reduced by aggregation through majority voting resulting in robust models.

Visual inspection of the PR curves shows that in the majority of experiments the curve for RESVM not only has higher AUC but completely dominates the other curves. As such RESVM models are a good approach regardless of design priorities (high recall versus high precision).
The robustness of our approach to potential contamination in both $\mathcal{P}$ and $\mathcal{U}$ can be attributed to the synergy between our resampling scheme and voting aggregation. The resampling itself strongly resembles a typical bootstrap approach. RESVM uses class-weighted SVM base models though the resampling scheme is likely to work well with other types of base models.

A weakness of RESVM is its amount of hyperparameters (5 plus potential kernel parameters), though RESVM models are less sensitive to accurate tuning of these parameters than standard SVM. An interesting question is whether prior knowledge regarding contamination of $\mathcal{P}$ and $\mathcal{U}$ can help in limiting the search scope for some of the hyperparameters ($n_{pos}$, $n_{unl}$ and $w_{pos}$ specifically).

7. Acknowledgements

Marc Claesen is a doctoral researcher at the Katholieke Universiteit Leuven. Bart De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium. Johan Suykens is a professor at the Katholieke Universiteit Leuven, Belgium.

Research supported by: IWT grant number 111065 (Marc Claesen), Research Council KU Leuven: GOA/10/09 MaNet, KUL PFV/10/016 SymBioSys, several PhD/postdoc and fellow grants Industrial Research fund (IOF): IOF/HB/13/027 Logic Insulin, IOF: HB/12/022 Endometriosis; Flemish Government: FWO: PhD/postdoc grants, projects: G.0871.12N (Neural circuits), research community MLDM; IWT: PhD Grants; TBM-Logic Insulin, TBM Rectal Cancer, TBM IEETA, O&O ExaScience Life Pharma Hercules Stichting: Hercules III PacBio RS iMinds: SBO 2013; Art&D Instance; ICON b-SLIM IMEC: phd grant VLK Stichting E. van der Schueren: rectal cancer VSC Tier 1: exome sequencing Federal Government: FOD: Cancer Plan 2012-2015 KPC-29-023 (prostate) COST: Action BM1104: Mass Spectrometry Imaging, Action BM1006: NGS Data analysis network; EU: ERC AdG A-DATADRIVE-B.

8. References

References

Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Van Loo, Bert Coessens, Frederik De Smet, Leon-Charles Tranchevent, Bart De Moor, Peter Mary-
nen, Bassem Hassan, Peter Carmeliet, and Yves Moreau. Gene prioritization through genomic data fusion. *Nature Biotechnology*, 24(5):537–544, May 2006. ISSN 1087-0156.

Carlos Alzate and Johan A. K. Suykens. A semi-supervised formulation to binary kernel spectral clustering. In *2012 IEEE World Congress on Computational Intelligence (IEEE WCCI/IJCNN 2012)*, Brisbane, Australia, June 2012.

Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139, 1999.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11:2973–3009, 2010.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. ISSN 0885-6125.

Leo Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40(3):229–242, 2000.

Gavin C Cawley. Leave-one-out cross-validation based model selection criteria for weighted ls-svms. In *Neural Networks, 2006. IJCNN’06. International Joint Conference on*, pages 1661–1668. IEEE, 2006.

Marc Claesens, Frank De Smet, Johan A. K. Suykens, and Bart De Moor. EnsembleSVM: A library for ensemble learning using support vector machines. *Journal of Machine Learning Research*, in press.

Anneleen Daemen, Olivier Gevaert, Fabian Ojeda, Annelies Debuquoy, Johan AK Suykens, Christine Sempoux, Jean-Pascal Machiels, Karin Haustermans, and Bart De Moor. A kernel-based integration of genome-wide data for clinical decision support. *Genome Medicine*, 1(4):39, 2009.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, ICML ’06, pages 233–240, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.
Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’08, pages 213–220, New York, NY, USA, 2008. ACM.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research, 9:1871–1874, 2008.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. IEEE Transactions on Neural Networks and Learning Systems, in press. doi: 10.1109/TNNLS.2013.2292894.

Yves Grandvalet. Bagging equalizes influence. Machine Learning, 55(3):251–270, 2004.

Maarten Keijzer and Vladan Babovic. Genetic programming, ensemble methods and the bias/variance tradeoﬀ introductory investigations. In Riccardo Poli, Wolfgang Banzhaf, William B. Langdon, Julian Miller, Peter Nordin, and Terence C. Fogarty, editors, Genetic Programming, volume 1802 of Lecture Notes in Computer Science, pages 76–90. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-67339-2. doi: 10.1007/978-3-540-46239-2_6. URL http://dx.doi.org/10.1007/978-3-540-46239-2_6.

Ar Lazarevic, Aysel Ozgur, Levent Ertoz, Jaideep Srivastava, and Vipin Kumar. A comparative study of anomaly detection schemes in network intrusion detection. In Proceedings of the Third SIAM International Conference on Data Mining, 2003.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In Proceedings of the IEEE, pages 2278–2324, 1998.

Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In Proceedings of the Twentieth International Conference on Machine Learning (ICML), page 2003, 2003.

Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In IJCAI’03: Proceedings of the 18th international joint conference on Artificial intelligence, pages 587–592, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.
Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 387–394, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1-55860-873-7.

Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM ’03, pages 179–186, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1978-4.

Zhigang Liu, Wenzhong Shi, Deren Li, and Qianqing Qin. Partially supervised classification – based on weighted unlabeled samples support vector machine. In *Proceedings of the First international conference on Advanced Data Mining and Applications*, ADMA’05, pages 118–129, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-27894-X, 978-3-540-27894-8.

Fantine Mordelet and Jean-Philippe Vert. A bagging SVM to learn from positive and unlabeled examples. *arXiv preprint arXiv:1010.0772*, 2010.

Fantine Mordelet and Jean-Philippe Vert. ProDiGe: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC bioinformatics*, 12(1):389, 2011.

Edgar Osuna, Robert Freund, and Federico Girosi. Support Vector Machines: Training and Applications. Technical Report AIM-1602, 1997.

Mykola Pechenizkiy, Alexey Tsymbal, Seppo Puuronen, and Oleksandr Pechenizkiy. Class noise and supervised learning in medical domains: The effect of feature extraction. In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pages 708–713. IEEE, 2006.

J Sunil Rao and Robert Tibshirani. The out-of-bootstrap method for model averaging and selection. *University of Toronto*, 1997.

Brian K. Shoichet. Virtual screening of chemical libraries. *Nature*, 432(7019):862–865, December 2004. ISSN 0028-0836.

Alejandro Sifrim, Dusan Popovic, Léon-Charles Tranchevent, Amin Ardervschirdavani, Ryo Sakai, Peter Konings, Joris Vermeesch, Jan Aerts, Bart
De Moor, and Yves Moreau. eXtasy: Variant prioritization by genomic data fusion. *Nature Methods*, 10:1083–1084, 2013.

Hwanjo Yu. Single-class classification with mapping convergence. *Machine Learning*, 61(1-3):49–69, November 2005. ISSN 0885-6125.

Hwanjo Yu, Jiawei Han, and Kevin C. Chang. PEBL: positive example based learning for web page classification using SVM. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–248, New York, NY, USA, 2002. ACM Press. ISBN 158113567X.

Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2004.

9. Vitae

This Section will be completed later.

Appendix A. Receiver Operating Characteristic curves

Tables A.4 and A.5 show area under the ROC curve (AUROC) for all experiments. These are based on the same experiments as Tables 1 and 2 which contain area under the PR curve (AUPR). We include these tables for readers that are less familiar with PR curves. Naturally, all main conclusions we have drawn from PR curves directly translate to ROC curves (Davis and Goadrich, 2006).

Table A.5 shows that the difference in AUROC is much smaller than the difference in area under the PR curves. This is because AUROC is fairly insensitive to class imbalance and thus poorly captures the difference between models. Based on AUROC, RESVM still consistently yields the best models for both PU learning settings.

Results in Table 2 showed that RESVM models were more consistent in terms of AUPR (narrower confidence intervals). This effect is much smaller for confidence intervals of AUROC in Table A.5. This confirms that using PR curves for performance assessment is a better choice in this setting.
Table A.4: 95% confidence intervals for the mean area under the ROC curve in percent on MNIST test set for each digit in a fully supervised one-vs-all setup. The results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $AUC^{RESVM} > AUC^{BAG}$ is included using the following result coding: • $p < 0.05$, •• $p < 0.01$ and ••• $p < 0.001$.

| digit | CWSVM   | BAG    | RESVM  | $p$ |
|-------|---------|--------|--------|-----|
| 0     | 99.4–99.5 | 99.5–99.6 | 99.4–99.6 |     |
| 1     | 99.6–99.7 | 99.7–99.7 | 99.7–99.7 |     |
| 2     | 95.5–96.1 | 96.1–96.6 | 96.5–97.1 | •• |
| 3     | 94.9–95.7 | 96.2–97.0 | 97.1–97.5 | ••• |
| 4     | 97.5–98.0 | 98.0–98.4 | 98.1–98.6 |     |
| 5     | 93.6–94.5 | 94.0–94.8 | 94.9–96.4 | ••• |
| 6     | 98.2–98.7 | 98.7–99.0 | 98.7–98.9 |     |
| 7     | 97.7–98.1 | 97.8–98.2 | 97.9–98.3 |     |
| 8     | 86.1–87.6 | 92.8–93.8 | 95.0–95.2 | ••• |
| 9     | 93.6–94.5 | 94.7–95.5 | 95.7–96.1 | ••• |
Table A.5: 95% confidence intervals for the mean area under the ROC curve in percent on MNIST test set for each digit in a one-vs-all setup. Models are trained in the following settings PU learning settings: (i) classic PU learning setting (10% false negatives, no false positives) and (ii) a noisy PU learning setting (10% false positives and 10% false negatives). In each setting the results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $AUC_{RESVM} > AUC_{BAG}$ is included using the following result coding: • $p < 0.05$, •• $p < 0.01$ and • • • $p < 0.001$.