Reply letter

Marc Claesen Frank De Smet Johan A.K. Suykens Bart De Moor

July 9, 2014

We would like to thank all reviewers and the editors for their valuable comments and suggestions related to our submitted manuscript. In this document we will briefly summarize the modifications and extensions we made to the manuscript to address the comments we received.

The main improvements to the manuscript are as follows:

- We expanded and clarified the theoretical justification of our approach (Sections 3.1 to 3.3).
- We have added references to out-of-bag estimates that can be used for some of the hyperparameters associated to RESVM (namely n_{models}, n_{pos} and n_{unl}) (Section 3.4).
- We performed extra experiments on new data sets with various properties (Sections 4 and 5).
- We report the simulation results in more details, namely we now include the number of times each method had best test set performance in simulations (Tables 2 to 5).

1 Guest editor's comments

The following papers also treat how the sampling size in bagging can improve the performance ... We have added a reference to the technique described in [1] in Section 3.4. We were unable to retrieve the second reference that was provided. As noted, the technique in [1] could provide more efficient ways to obtain suitable values for the n_{pos} and n_{unl} hyperparameters. An additional difficulty in the context of learning with label noise concerns how to estimate out-of-bag performance properly (e.g. the scoring problem resurfaces). The focus of our manuscript is on the properties of the new approach itself. While relevant, we opt not to elaborate on potential efficiency gains through clever means of hyperparameter search.

2 Reviewer 1

The method is only tested in one dataset that splits into ten classification problems. In any case, this is clearly not enough to validate the proposed technique. The experiments are repeated only 10 times, 100 would be better in order to obtain more stable results. We have added experiments on several new data sets, namely: a `synthetic` data set, Wisconsin breast `cancer` (biological data, binary), `ijcnn1` (synthetic data set used in a data mining competition, binary) and `sensit` (vehicle classification, three classes). The statistical tests we used already account for the variability in outcomes. Statistically significant outcomes in small samples imply a large effective difference (e.g. a large performance improvement of RESVM compared to bagging SVM). Having larger samples (more iterations) would likely favor RESVM even more in comparisons by yielding even more statistically significant outcomes (since then smaller improvements can become statistically significant).

Only one label noise level is used (10%) and no noise. Given that this Special Issue is about label noise, a further analysis of the proposed method under different noise conditions should be analyzed. How the method performs with higher contamination in the unlabeled set? What happens when the unlabeled set has more positive instances than negative? The proposed method was analyzed under three different noise conditions for each data set (no label noise, label noise in $\mathcal{U}$ and label noise in $\mathcal{U}$ and $\mathcal{P}$). The noise levels were chosen per data set based on when the different methods' performance started to diverge. We have added an empirical analysis of the effect of contamination in $\mathcal{P}$ and $\mathcal{U}$ in Section 5.5 on a synthetic data set.

The train set is composed of 50+ and 2000- instances, Why these numbers considering that 6000+ and 54000- are available. The variability of the training set can be huge with such a reduced samples. This is intended to illustrate that using (some of) these methods good models can be obtained even when little data is available. Additionally, the resulting variability allows us to assess the consistency of each method. We can see that RESVM is typically more consistent than the other approaches in the PU learning and semi-supervised settings, illustrated by narrow confidence intervals on generalization performance.

The test set is composed of 1000+ and 9000-. Why not using all instances for testing (except the ones used for training)? In the interest of clarity and reproducibility we decided to use prespecified test sets when they are included in a data set.

How is the noise injected in the training sets? Flipping the class labels? substituting negative for positive instances? This is an important point to understand how everything works. The noise was injected to simulate our definition of contamination[1], e.g. by substituting negatives for positives. For example $\mathcal{P}$ is a mixture of true positives and true negatives in the semi-supervised setting.

The grid of parameter value tuples used in the in-train cross-validation is not specified. The same search grids were used in each setting (different grids for different data sets). Since we obtained good models for each approach in a supervised setting, the grids themselves are well chosen. When different methods have comparable hyperparameters (e.g. the RBF kernel parameter), all methods shared the same grid points in these hyperparameters to allow a fair comparison. Finally, we ensured that the optimal parameters that were found were never on the edge of a method's search grid (if they were, the experiment was repeated with an extended grid). We have clarified this in the manuscript in Section 4.1.

Using a different size for the ensembles in the parameter estimation (15) and in the final execution (50) might produce some errors in the estimation of the optimum parameters. We now consistently use 50 base models in every ensemble (bagging SVM and RESVM), both during parameter estimation and final execution. We did not witness any significant changes in the resulting models compared to our previous experiments on `mnist`.

One of the most interesting things of ensemble leaning is that they are almost parameterless. It would be interesting if some experiments could be done to analyze if some parameter configuration could be valid in general for the proposed method. Unfortunately the ensembles discussed here are not parameterless. No generally applicable default values exist, especially for the base model hyperparameters (particularly kernel parameters). That said, we did observe that both ensemble approaches are less sensitive to parameter changes than traditional SVM, which can also be considered a form of robustness. Good ensemble models can be obtained using a much coarser search grid than a single SVM. We have provided references to estimation procedures for the parameters associated with the meta-classifier in Section 3.4 (n_{models}, n_{pos} and n_{unl}).

[1] Contamination is the fraction of mislabeled instances in a given set.

I do not see the point in eq. (5) since the decision is the same when using $v(x)$ or $d(x)$ since $sgn(d(x)-T) = sgn(v(x)-T)$. The use of the parameter w_{pos} is also not very clear. The difference between $d(x)$ and $v(x)$ is relevant for ranking, e.g. when computing PR curves: using $v(x)$ it is impossible to rank within groups of instances that received unanymous votes, whereas $d(x)$ can account for base model confidence to rank within such groups. We clarified this in Section 3.5. w_{pos} allows reweighting the relative misclassification penalty on positives. In bagging SVM, w_{pos} is implicitly fixed to 1 while we observe that allowing this ratio to change improves performance for RESVM (see also Table 6 in Section 5.6).

"Bagging schemes typically use instable base models...like decision trees...variability in RESVM is a direct result of resampling...". This is not clear at all. Resampling is the basic idea behind bagging to introduce variability in the base models. However, not all models are subject to this variability. For instance, standard bagging nearest neighbors is equivalent to single nearest neighbors [Breiman, 1996] We expanded on this aspect in Sections 3.1 to 3.3. Indeed, not all models are subject to the variability induced by resampling (in fact, SVM models are considered to be fairly insensitive to such effects in normal circumstances). In the context of using SVMs with label noise, resampling does introduce variability between base models. This is because resampling leads to variability in contamination levels across training sets, which do have a large influence on the resulting base models.

The [number of base learners] can be determined during training by monitoring the predictive performance...while adding more base models...". How could this be done? Cites required. A reference was added [2], which explains how out-of-bag estimates can be used to assess when an ensemble is of sufficient size.

variability in contamination... increases for increasing contamination...". This is true up to 50% contamination. For higher values the variability starts to decrease. This is correct. We have added that we assume contamination levels below 50% in the given paragraph.

The confidence interval...have fewer outliers." I do not understand this sentence. What kind of outliers are you referring to? label-noise? What does it mean that a confidence interval has fewer outliers? We agree that this was phrased poorly. We meant to say that there are few cases where the confidence interval for RESVM is very large (e.g. due to one or two very bad models in the simulations). The outliers referred to the width of the confidence intervals, not the intervals themselves. This has been rephrased entirely in the revision.

In the conclusions it is said that RESVM favor smaller base models. This has not been treated in the experimental section and what is the sense of smaller models in the context of SVM? Smaller models have higher variability, based on the reasoning in Sections 3.1 and 3.3. In supervised contexts, more data typically leads to better SVM models. A tradeoff must be made between base model accuracy (larger models) and bagging efficacy (e.g. more base model variability (smaller models)). When learning with label noise, more data does not necessarily imply better models. This situation tends to favor improving the efficacy of bagging, which is typically achieved by introducing more base model variability (e.g. smaller models).

3 Reviewer 2

The drawback of this algorithm is that it has at least five hyper parameters. Searching these hyper parameters requests the higher time complexity. This is indeed a drawback of RESVM. In the revision, we have added references to efficient ways to estimate the meta-classifier's hyperparameters in Section 3.4 (e.g. n_{models}, n_{pos} and n_{unl}). This simplifies the hyperparameter search.

In this paper the period is used as the thousand separator, e.g. 10.000. However, the comma is a common character of the thousand separator rathe than the period, e.g. 10,000. We now use commas as thousand separators.

4 Reviewer 3

My main reservation, and I suspect that others will also say this, is that empirical results on a single dataset are never particularly convincing. I would like to see several datasets, with various characteristics, and the main comparison methods applied to each of them. We have included experiments on new data sets with varying properties in the manuscript.

Halfway down para 3 you state that contamination in bootstrap samples converges to the contamination in the dataset, but since you seem to sample from P and U in varying proportions that isn't true generally. If you take uniform bootstrap samples of any fixed size over $P \cup U$, then indeed the *average* contamination over the bootstrap samples converges to its expectation - the contamination in the dataset. The resampling argument we make in Section 3.1 applies to resampling $\mathcal{P}$ and $\mathcal{U}$ separately (see also Algorithm 1 which formalises the resampling). The expected contamination of a resample from $\mathcal{P}$, say $\mathcal{P}^{(i)}$, equals the contamination of $\mathcal{P}$. It is indeed correct that the contamination of the new training set $\mathcal{P}^{(i)} \cup \mathcal{U}^{(i)}$ depends on the sizes of the resamples $|\mathcal{P}^{(i)}|$ and $|\mathcal{U}^{(i)}|$ as well as the contamination in the original sets $\mathcal{P}$ and $\mathcal{U}$.

Since I suspect that biasing the expected contamination (i.e. to make it smaller) is probably a good thing to do here, I don't think that is a problem for your approach however. This might indeed be a subtle mechanism that improves the efficacy of our approach.

At the top of para 3.1 you say the contamination between P and U can vary, so the ability to vary the size of resamples is important. I'm not really clear on how, for a fixed training set, the contamination can vary? For a given training set the contamination is indeed fixed. What we meant here is that the relative contamination of $\mathcal{P}$ and $\mathcal{U}$ depends on the problem at hand, so a tuning parameter is useful to account for such differences. When training a RESVM, the same resample sizes n_{pos} and n_{unl} are used by all base models (see also Algorithm 1), but these sizes must be configurable to be able to address different data sets.

References

[1] Gonzalo Martínez-Muñoz and Alberto Suárez. Out-of-bag estimation of the optimal sample size in bagging. *Pattern Recognition*, 43(1):143–152, 2010.

[2] Robert E Banfield, Lawrence O Hall, Kevin W Bowyer, and W Philip Kegelmeyer. A comparison of decision tree ensemble creation techniques. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):173–180, 2007.