Phenotype Prediction with Semi-supervised Classification Trees

Jurica Levatić[1,2(✉)], Maria Brbić[3], Tomaž Stepišnik Perdih[1,2], Dragi Kocev[1,2], Vedrana Vidulin[1,3,4], Tomislav Šmuc[3], Fran Supek[3,5], and Sašo Džeroski[1,2]

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
jurica.levatic@ijs.si
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[3] Division of Electronics, Ruder Boskovic Institute, Zagreb, Croatia
[4] Faculty of Information Studies, Novo Mesto, Slovenia
[5] Center for Genomic Regulation, Barcelona, Spain

Abstract. In this work, we address the task of phenotypic traits prediction using methods for semi-supervised learning. More specifically, we propose to use supervised and semi-supervised classification trees as well as supervised and semi-supervised random forests of classification trees. We consider 114 datasets for different phenotypic traits referring to 997 microbial species. These datasets present a challenge for the existing machine learning methods: they are not labelled/annotated entirely and their distribution is typically imbalanced. We investigate whether approaching the task of phenotype prediction as a semi-supervised learning task can yield improved predictive performance. The results suggest that the semi-supervised methodology considered here is especially helpful when using single trees, especially when the amount of labeled data ranges from 20 to 40%. Similar improvements can be seen when the presence of the phenotype is very imbalanced.

Keywords: Semi-supervised learning · Phenotype · Decision trees
Predictive clustering trees · Random forests · Binary classification

1 Introduction

The most common task in machine learning is supervised learning, where the goal is to predict the value of a target attribute of an example by using the values of descriptive attributes. Supervised methods often need a large amount of labeled data to learn a predictive model with a satisfying predictive performance. However, in many real-life problems, such as phonetic annotation of human speech, protein 3D structure prediction, and spam filtering, only a few labeled examples are available to learn from because of the expensive and/or time-consuming annotation procedures. Contrary to labeled examples, unlabeled examples are often freely available in vast amounts. For example, human speech can be recorded from radio broadcasts, while DNA sequences of proteins can

A. Appice et al. (Eds.): NFMCP 2017, LNAI 10785, pp. 138–150, 2018.
https://doi.org/10.1007/978-3-319-78680-3_10

be extracted from gene databases. Semi-supervised learning (SSL) emerged as an answer to the problem of labeled data scarcity [1], with an idea to exploit freely/easily available unlabeled examples to get better predictive performance than the one achieved using labeled data alone.

In this work, we are concerned with the task of microbial phenotype prediction. Phenotypes are defined as variations in observable characteristics of an organism. Microbial organisms display a large diversity of possible phenotypic traits, such as ability to inhabit different environments, adaptation to extreme conditions and association to different hosts. The annotation of organisms with phenotypes is important for understanding the genetic basis of phenotypes. It often requires expensive experimental measurements and time-consuming manual curation, hence there is a huge amount of unlabeled organisms. On the other hand, phenotypes can be efficiently predicted from genome [2–5] and metagenome data [6].

Thanks to the emergence of DNA sequencing technology, the number of sequenced genomes is rapidly increasing, making unlabeled data easily available. This makes the problem of phenotype prediction well suited for semi-supervised learning. In this work, we explore whether better predictive performance can be achieved with semi-supervised machine learning methods than with supervised methods that have been used for this task in [7][1], namely classification trees and random forests. To the best of our knowledge, this is the first application of semi-supervised learning for microbial phenotype prediction.

In this work, we compare the predictive performance of supervised and semi-supervised classification trees and random forests thereof [8] to predict 114 phenotypes of 997 microbial organisms. These datasets pose interesting challenges for existing machine learning methods because the annotations are not complete and the available datasets are imbalanced. To this end, we investigate whether we can benefit from using semi-supervised learning under these difficult conditions. In a nutshell, the results reveal that the semi-supervised classification trees can improve the predictive performance over supervised classification trees in cases where the amount of labeled data is in the range 20–40% and for phenotypic traits that are not extremely rare.

The rest of this paper is organized as follows. Section 2 describes the semi-supervised methods used in this study, while Sect. 3 describes the data used for phenotype prediction. Section 4 specifies the experimental design. The results of the empirical investigations are presented and discussed in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Methods

In this work, we consider semi-supervised classification trees and semi-supervised random forests [8], which are based on the predictive clustering trees (PCTs) [9] and ensembles thereof [10]. PCTs view a decision tree as a hierarchy of clusters,

[1] Phenotype predictions from [7] are available at protraits.irb.hr.

where the top-node corresponds to one cluster containing all the data. This cluster is then recursively partitioned into smaller clusters while moving down the tree. Semi-supervised PCTs are implemented in the CLUS system [11] (implementation available at http://kt.ijs.si/jurica_levatic/). In this section, we briefly describe semi-supervised trees and random forests, while for more details we refer the reader to the work of Levatić et al. [8].

Supervised classification trees evaluate the quality of splits on the basis of the class labels, by using, for example information gain or gini impurity as a quality measure. Consequently, the resulting clusters (i.e., groups of examples defined by splits in the tree) are homogeneous only with respect to the class label. Semi-supervised PCTs [8], on the other hand, measure the quality of splits considering both the class labels and descriptive attributes. Therefore, the resulting clusters are homogeneous with respect to both the descriptive attributes and the class labels. Note that, only the descriptive attributes are known for unlabeled examples, thus, such semi-supervised trees can exploit them during the tree construction - contrary to supervised trees. The rationale behind the described semi-supervised classification trees is the semi-supervised cluster assumption [1]: *If examples are in the same cluster, then they are likely of the same class.*

The semi-supervised PCTs are based on the standard *top-down induction of decision trees* (TDIDT) algorithm (see Table 1), which takes as input a set of examples E and outputs a tree. The heuristic score (h) that is used for selecting the tests (t) to put in the internal tree nodes is reduction of impurity caused by partitioning ($\mathcal{P}$, Table 1, line 3 of the BestTest procedure) the examples according to the tests.

In supervised PCTs, the impurity for each set of examples E is calculated as the gini impurity (Table 1, line 5 of the *BestTest* procedure):

$$Impurity(E) = Gini(E, Y). \tag{1}$$

Table 1. The top-down induction algorithm for decision trees construction.

procedure InduceTree
Input: A dataset E
Output: A predictive clustering tree
1: $(t^*, h^*, \mathcal{P}^*) = \text{BestTest}(E)$
2: **if** $t^* \neq none$ **then**
3: **for each** $E_i \in \mathcal{P}^*$ **do**
4: $tree_i =$
5: InduceTree(E_i)
6: **return**
7: node(t^*, $\bigcup_i \{tree_i\}$)
8: **else**
9: **return**
10: leaf(Prototype(E))

procedure BestTest
Input: A dataset E
Output: the best test (t^*), its heuristic score (h^*) and the partition ($\mathcal{P}^*$) it induces on the dataset (E)
1: $(t^*, h^*, \mathcal{P}^*) = (none, 0, \emptyset)$
2: **for each** possible test t **do**
3: $\mathcal{P} =$ partition induced by t on E
4: $h = Impurity(E) -$
5: $\sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Impurity(E_i)$
6: **if** $(h > h^*) \wedge \text{Acceptable}(t, \mathcal{P})$ **then**
7: $(t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
8: **return** $(t^*, h^*, \mathcal{P}^*)$

As mentioned before, to identify the best splits, the impurity function of semi-supervised PCTs takes into account both the target attribute (i.e., the class labels) and the descriptive attributes. This is achieved by changing the equation for the calculation of impurity for supervised PCTs (Eq. 1). Impurity of a set of examples E (which may contain labeled and unlabeled examples) is calculated as a weighted sum of impurities over the target attribute (Y) and impurities over the descriptive attributes (X_i):

$$Impurity_{SSL}(E) = w \cdot Impurity(E_l, Y) + \frac{1-w}{D} \cdot \sum_{i=1}^{D} Impurity(E, X_i), \quad (2)$$

where $E = E_l \cup E_u$ is the dataset available at a node of the tree, D is the number of descriptive attributes, X_i is the i^{th} descriptive attribute, and $w \in [0, 1]$ is a weight parameter.

The impurity of the target attribute Y is calculated as gini impurity over a set of labeled examples E_l. Differently from the target attribute, which is nominal, the descriptive attributes can be either nominal or numeric, therefore, the two cases are considered separately: if the attribute is nominal as a measure of impurity gini impurity is used, whereas, if the attribute is numeric, as a measure of impurity variance is used.

The weight parameter w in (2) controls how much the target side or the descriptive side contribute to the calculation of the impurity. Consequently, this controls how much the unlabeled examples affect the learning of semi-supervised PCTs. Namely, depending on the values of the w parameter, semi-supervised PCTs can range from fully supervised trees (i.e., $w = 1$) to completely unsupervised trees (i.e., $w = 0$). This aspect is important since unlabeled examples can sometimes cause semi-supervised algorithms to perform worse than their supervised counterparts [12–14]. The w parameter acts as a safety mechanism of semi-supervised PCTs, enabling them to control the influence of unlabeled examples and adapt to a given dataset.

If no acceptable test is found because some stopping criteria is met (e.g., minimum number of examples in the leaf has reached the user predefined value, the variance reduction is not relevant etc.) then the algorithm places a leaf node at that position. In each leaf node, the prototype for the examples belonging to that leaf node is calculated (by using the function $Prototype(E)$ from the InduceTree procedure in Table 1 at line 10) and stored.

By using semi-supervised PCTs, it is possible to build semi-supervised random forests. A random forest [15] is an ensemble of trees, where diversity among the trees is obtained by making bootstrap replicates of the training set, and additionally by randomly selecting the subset of descriptive attributes used to evaluate the splits. Random forests often substantially improve the predictive performance of single trees, however, the interpretability aspect of trees is lost. Semi-supervised random forests of PCTs are built by using semi-supervised PCTs as members of the ensemble, instead of using supervised PCTs. In semi-supervised random forests, the bootstrap sampling procedure is modified to perform

stratified bootstrap sampling (considering the proportions of labeled and unlabeled examples) to avoid having bootstrap samples consisting only of unlabeled examples.

We next analyze the computational complexity of semi-supervised PCTs. We first recall the procedures that contribute to the computational complexity of *supervised* PCTs. These are as follows: sorting the values of D descriptive attributes ($O(DN \log N)$), calculating the best split for T target variables ($O(TDN)$), and applying the split to the N (labeled) training examples ($O(N)$). Assuming that the depth of the tree is in the order of $O(\log N)$ [16], the total computational complexity of constructing a single PCT is $O(DN \log^2 N) + O(TDN \log N) + O(N \log N)$.

We then consider what changes from supervised PCTs to semi-supervised PCTs. This is, first, the value of N: In the case of semi-supervised PCTs, the number of training examples is equal to the number of labeled and unlabeled examples combined, i.e., $N = N_l + N_u$, instead of $N = N_l$. Second, SSL-PCTs consider both D descriptive attributes and T target variables when the split is calculated, thus the complexity of this step is $O((T + D)DN)$. The total computational complexity of learning a single SSL-PCT is thus $O(DN \log^2 N) + O((T+D)DN \log N) + O(N \log N)$. This cost is then linearly extended to random forests of PCTs similarly as in [10]. Additionally, one should also consider the cost for obtaining the optimal value for the w parameter, which is usually performed using an inner cross-validation procedure.

3 Data Description

Prokaryotic genome sequences and gene annotations were downloaded from the NCBI Genomes database and COG/NOG gene families were downloaded from eggNOG 3 [17]. In our analysis, we considered species that have a genome quality score greater or equal to 0.9 (out of 1) [22]. Higher score corresponds to the higher level of completeness of sequenced genome data, where scores of 0.8 or higher indicate that a genome can be safely used for standard comparative genomics analysis. Phenotype annotations are NCBI+Bacmap labels as in [7], collected from the NCBI microbial genome projects list ('lproks0' table) and from the BacMap database [18], in total 114 different phenotypic traits. We considered only species having at least one assigned phenotype label, resulting in 997 species. Each example corresponds to one species labeled with a set of available phenotypic traits. For each species, the labels correspond to presence or absence of traits, thus, the task of phenotype prediction corresponds to a binary classification problem.

The labelling is, however, not exhaustive: For most of the phenotypes, only 30% of species are labeled (Fig. 1a). Hence, the dataset at hand contains unlabeled data, which can be exploited with semi-supervised methods. The class distribution of most of the phenotypes is unbalanced (Fig. 1b): Many traits appear at less than 10% of species, e.g. radiation-resistance phenotype and ability to withstand extremely high (hyperthermophilic organisms) or extremely low temperature (psychrophilic organisms).

In all experiments, we used the gene repertoire representation [2]. The features describing the species were encoded as the presence/absence of the clusters of orthologous (COG) and non-supervised orthologous (NOG) groups of proteins, resulting in the 80576 binary valued features. In order to reduce the dimensionality of the feature set we applied principal component analysis (PCA) as a preprocessing step and retained principal components explaining 90% of the variance. This resulted in 526 features, i.e., principal components.

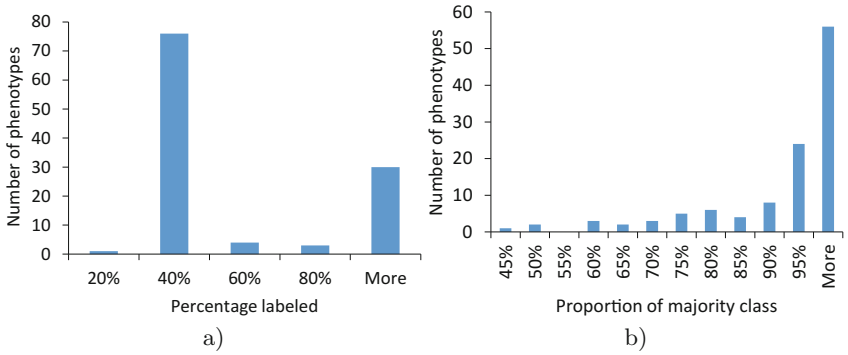

Fig. 1. (a) Histogram of the amount of labeled (relative to unlabeled) examples for each phenotype. (b) Histogram of the majority class distributions of phenotypes.

4 Experimental Design

We learn a separate model for each phenotype, transforming the problem of phenotype prediction into 114 binary classification tasks. We then approach these tasks with two learning paradigms: supervised and semi-supervised learning. In other words, we learn predictive models in the form of supervised classification trees (PCTs) and semi-supervised classification trees (SSL-PCTs) as well as supervised random forests and semi-supervised random forests. Performance was estimated with 10-fold cross validation procedure. The predictive performance reported in the results is the average of the performance values obtained from the 10 folds.

In the experiments, both supervised and semi-supervised trees are pruned with the procedure used in C4.5 classification trees [19]. The weight parameter w of semi-supervised algorithms was chosen from the set $\{0, 0.1, \ldots, 0.9, 1\}$ by using internal 3-fold cross validation on the labeled part of the training set. We construct random forests consisting of 100 trees. The trees in random forests are not pruned and the number of random features at each internal node is set to the square root of the number of features, which in our case amounted to 23.

Next, we compare the performance of semi-supervised PCTs and semi-supervised random forests to their supervised counterparts. For every phenotype, examples with unknown labels were used as unlabeled data for learning the semi-supervised PCTs and ensembles thereof.

Furthermore, we investigate the influence of the amount of annotated phenotypes on the performance of the semi-supervised methods. More specifically, we analyze the performance of the predictive models across the different percentages of annotated phenotypes. Moreover, we juxtapose this influence with the influence of the imbalance of the class labels.

We also investigate how the value of the w parameter affects semi-supervised methods. To do this, we randomly select 4 phenotypes and learn semi-supervised models (both single PCTs and random forests) to predict them for all w in $\{0, 0.1, \ldots, 0.9, 1\}$ the resulting performances. Additionally, we analyze the performance of predictive models for different values of the w parameter selected by the internal cross validation.

Finally, in our evaluation scenario we use truly unlabeled data, and not unlabeled data that is obtained by removing the labels as it is usually done in most SSL studies. Therefore, for each phenotype we use all the available unlabeled data. We have performed an analysis of the influence of the amount of the unlabelled data in [8]. The study revealed that the advantage of semi-supervised classification trees over supervised trees is dependant more on the dataset at hand, rather than on the amount of unlabeled data used, i.e., if the SSL algorithm wins, it is likely to win for different amounts of unlabeled data (on that dataset).

5 Results and Discussion

5.1 Predictive Performance

The performances of predicting 114 microbial phenotypic traits with supervised and semi-supervised trees and random forests are presented in Fig. 2. Because class distribution was very imbalanced for some phenotypes, we used F1 score (harmonic mean of precision and recall) in addition to accuracy to measure the performance. We can observe that for many of the traits, semi-supervised algorithms outperform their supervised counterparts, suggesting that semi-supervised methods can successfully exploit unlabeled data and more accurately predict microbial phenotypes. The advantage of semi-supervised methods is, however, not observed for all phenotypes. This is expected, since several researchers found that the success of semi-supervised methods is, in general, dataset dependent [20]. In other words, it cannot be expected that semi-supervised methods will win against supervised ones for all cases. Furthermore, several researchers have found that semi-supervised learning may sometimes perform worse than supervised learning [12–14]. The numbers of wins, ties and losses of semi-supervised algorithms compared to their supervised counterparts in accuracy and F1 score can be seen in Table 2. Ties were results where the difference in performance was smaller than 0.01.

Our results also suggest that improving (with unlabeled data) a supervised random forest is a harder task than improving over a supervised tree: The number of wins of semi-supervised random forests is lower than the number of wins of semi-supervised PCTs. This observation complies with previous findings [8].

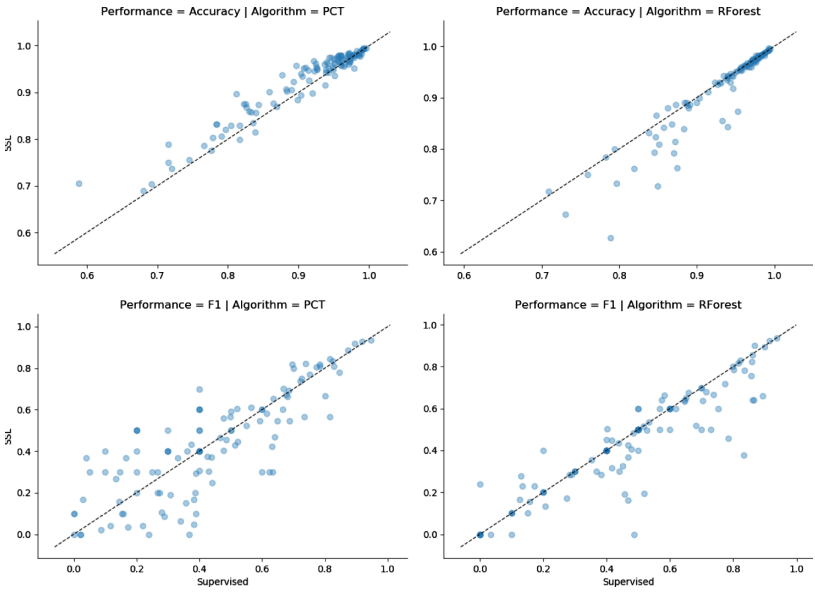

Fig. 2. Each dot represents the performance on one phenotype of supervised and semi-supervised methods. Values above the diagonal (dashed line) denote that the semi-supervised algorithm performed better. Darker color means greater density of dots. (Color figure online)

Table 2. Numbers of wins, ties and losses of semi-supervised algorithms compared to their supervised counterparts.

	PCT:Acc	PCT:F1	RForest:Acc	RForest:F1
Wins	62	50	3	16
Ties	44	15	91	58
Losses	8	49	20	40

We consider that this is due to the fact that ensembles are very powerful predictive models, which are able to exploit all the information in a given (labeled) dataset and approach the learnability borders of a given domain closer than a single predictive model. Thus, arguably, random forests do not benefit so much from additional information that unlabeled data bring, as compared to single trees.

We further analyze the results with the goal to identify phenotypes that are suitable for prediction with semi-supervised methods. The amount of available labeled data (relative to unlabeled) is an important factor for the performance of semi-supervised methods [8]. We therefore analyze the results from that aspect (Fig. 3). We can observe that semi-supervised single trees perform better with smaller amounts of labeled data according to accuracy and F1 score,

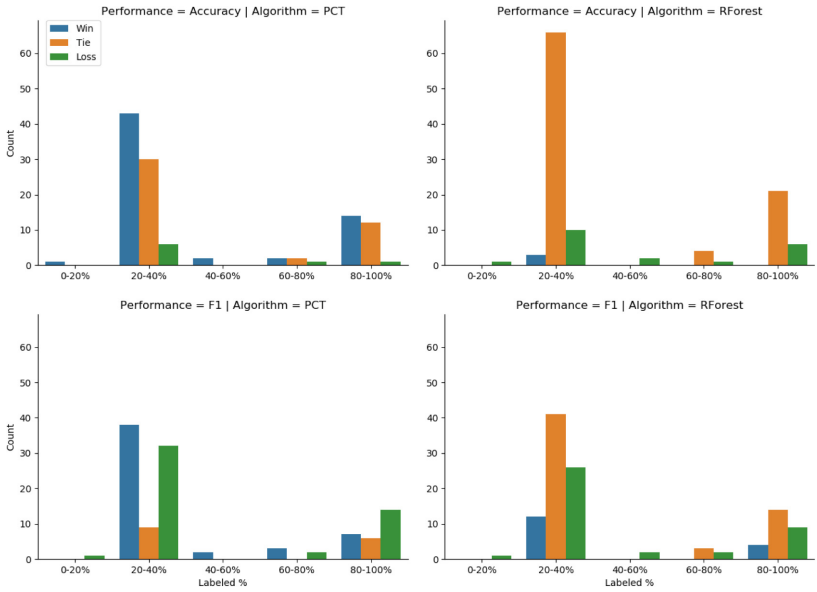

Fig. 3. The numbers of wins, ties and losses of semi-supervised PCTs and random forests versus their supervised counterparts, achieved for phenotypes with different amounts of labeled examples.

while supervised single trees have better F1 scores on phenotypes with a lot of labeled examples (over 80%). Supervised and semi-supervised Random forests are mostly tied, especially in terms of accuracy, however losses are more common than wins.

Recall that many of the phenotypes have very unbalanced classes (Fig. 1b). We next analyze whether the imbalance of the classes affects the performance of semi-supervised methods (Fig. 4). Interestingly, we can see that the semi-supervised methods achieve most wins in F1 score on phenotypes with the highest class imbalance. This holds for both single trees, where losses are more common than wins when the proportion of the majority class is less than 95%, and random forests, where the number of wins on the most imbalanced targets is almost the same as the number of losses, even though losses are far more common overall. Less surprisingly, we can also see that the vast majority of ties comes from phenotypes with the highest class imbalance.

5.2 Influence of the w Parameter

Figure 5 shows the accuracy of semi-supervised methods with different values of the w parameter for 4 randomly selected targets (phenotypes). We can see that in some cases its influence is minimal (e.g., random forests on target 2) but more often we need to select the right value for w to improve the performance over supervised methods. The best performance is achieved with different values

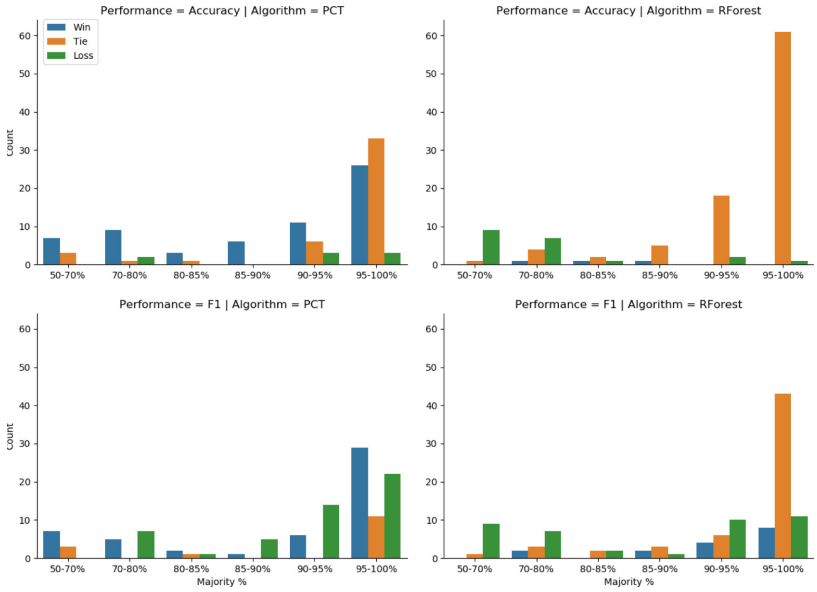

Fig. 4. The numbers of wins, ties and losses of semi-supervised PCTs and random forests versus their supervised counterparts, achieved for phenotypes with different proportion of the majority class.

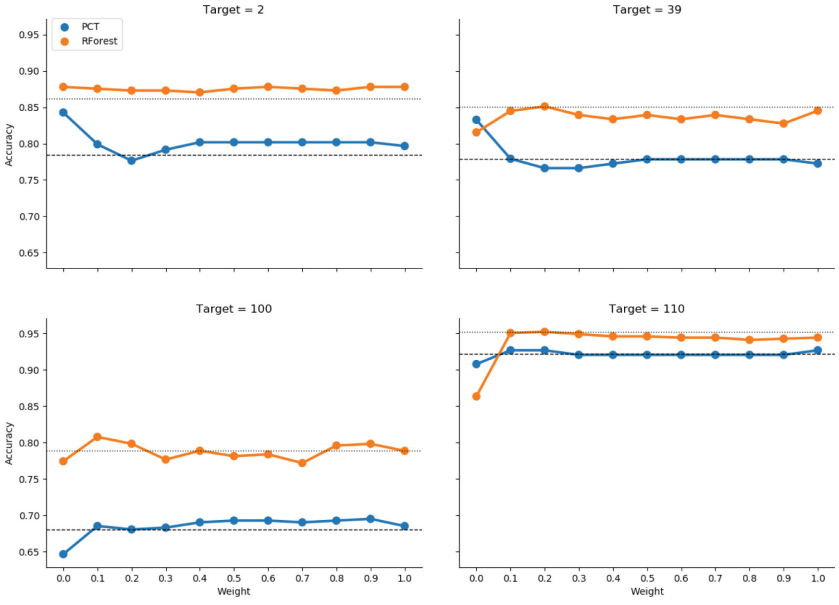

Fig. 5. The performance of semi-supervised methods for various values of w for 4 random phenotypes. The dashed lines represents the performance of supervised PCTs while the dotted line shows the performance of supervised random forests.

of the parameter, indicating that it should be tuned for every dataset. This is consistent with previous results on this topic [8] and the reason why we used internal cross validation to select it.

We also look at the numbers of wins, ties and losses according to the w selected (Fig. 6). Because the performance was measured with 10-fold cross validation and a different w was selected for each fold, we here compare the performances on each fold and not aggregated as before.

First, we note that for single trees $w = 0$ and $w = 1$ were the most common selections. Interestingly, when $w = 0$ is selected, accuracy is improved in most cases while the F1 score is close to even. Ties are most common for $w = 1$, which is to be expected. For random forests $w = 1$ is selected almost always, which contributes to the high number of ties in performance observed in the results previously.

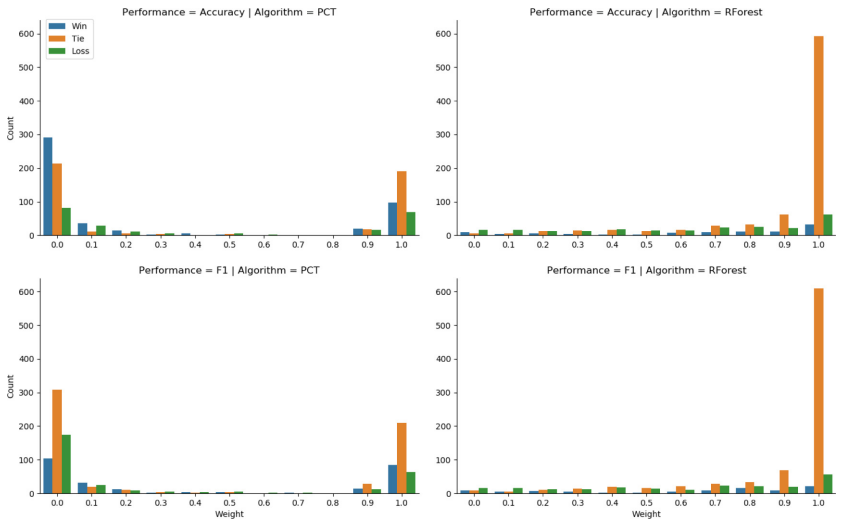

Fig. 6. Numbers of wins, ties and losses for different values of the w parameter selected by the internal cross validation.

6 Conclusions

In this work, we approach the task of phenotypic traits prediction using methods for semi-supervised learning. This task is important to understand the genetic basis for appearance of specific phenotypes. More specifically, we consider 114 datasets with different phenotypic traits referring to 997 microbial species. The datasets are not completely labelled and different amount of annotation is available for the different traits.

We investigate whether approaching the task of phenotype prediction as a semi-supervised learning task can yield improved predictive performance.

More specifically, we learn supervised and semi-supervised classification trees as well as supervised and semi-supervised random forests of classification trees. We then compare the performance of predictive models learned using supervised and semi-supervised methods.

The result suggest that the semi-supervised methodology considered here improves the accuracy of single trees and also their F1 score when the amount of labeled data ranges from 20 to 40%. Similar improvement can be seen when the presence of a phenotype is very imbalanced (proportion of the majority class over 95%). Improvement of random forests was rarer but also more common on previously mentioned groups of phenotypes. In applications where interpretable models are needed, semi-supervised classification trees should be favored over the supervised classification trees. We also showed that the performance of semi-supervised methods is sensitive to the value of the w parameter and that it should be tuned to each dataset.

We plan to further extend this work along several dimensions. To begin with, we plan to use phenotypes from other sources, specifically phenotypes from GOLD database [21] and especially biochemical phenotypes from [7] where the labeled examples are extremely scarce. Furthermore, we plan to consider other feature spaces, namely the proteome composition, gene neighborhoods and translation efficiency representations [7]. Next, we will compare the approaches presented here with other methods used for phenotype prediction including, but not limited to, SVMs and semi-supervised SVMs. Note that, considering the number of datasets considered here, such experiments will require massive computational power. Finally, we can treat the problem as a multi-label classification problem and obtain a partially labelled dataset that can be then approached from this perspective.

Acknowledgments. We acknowledge the financial support of the Slovenian Research Agency, via the grant P2-0103 and a young researcher grant to TSP, Croatian Science Foundation grants HRZZ-9623 (DescriptiveInduction), as well as the European Commission, via the grants ICT-2013-612944 MAESTRA and ICT-2013-604102 HBP. We would also like to acknowledge the joint support of the Republic of Slovenia and the European Union under the European Regional Development Fund (grant "Raziskovalci-2.0-FIŠ-52900", implementation of the operation no. C3330-17-529008).

References

1. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised Learning, vol. 2. MIT Press, Cambridge (2006)
2. MacDonald, N.J., Beiko, R.G.: Efficient learning of microbial genotype-phenotype association rules. Bioinformatics **26**(15), 1834 (2010)
3. Smole, Z., Nikolic, N., Supek, F., Šmuc, T., Sbalzarini, I.F., Krisko, A.: Proteome sequence features carry signatures of the environmental niche of prokaryotes. BMC Evol. Biol. **11**(1), 26 (2011)
4. Feldbauer, R., Schulz, F., Horn, M., Rattei, T.: Prediction of microbial phenotypes based on comparative genomics. BMC Bioinform. **16**(14), S1 (2015)

5. Brbić, M., Warnecke, T., Kriško, A., Supek, F.: Global shifts in genome and proteome composition are very tightly coupled. Genome Biol. Evol. **7**(6), 1519 (2015)
6. Chaffron, S., Rehrauer, H., Pernthaler, J., von Mering, C.: A global network of coexisting microbes from environmental and whole-genome sequence data. Genome Res. **20**(7), 947–959 (2010)
7. Brbić, M., Piškorec, M., Vidulin, V., Kriško, A., Šmuc, T., Supek, F.: The landscape of microbial phenotypic traits and associated genes. Nucleic Acids Res. **44**(21), 10074 (2016)
8. Levatić, J., Ceci, M., Kocev, D., Džeroski, S.: Semi-supervised classification trees. J. Intell. Inf. Syst. **49**(3), 461–486 (2017)
9. Blockeel, H., De Raedt, L., Ramon, J.: Top-down induction of clustering trees. In: Proceedings of the 15th International Conference on Machine learning, pp. 55–63 (1998)
10. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recogn. **46**(3), 817–833 (2013)
11. Blockeel, H., Struyf, J.: Efficient algorithms for decision tree cross-validation. J. Mach. Learn. Res. **3**, 621–650 (2002)
12. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Mach. Learn. **39**(2–3), 103–134 (2000)
13. Cozman, F., Cohen, I., Cirelo, M.: Unlabeled data can degrade classification performance of generative classifiers. In: Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference, pp. 327–331 (2002)
14. Guo, Y., Niu, X., Zhang, H.: An extensive empirical study on semi-supervised learning. In: Proceedings of the 10th International Conference on Data Mining, pp. 186–195 (2010)
15. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
16. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Cambridge (2005)
17. Powell, S., Szklarczyk, D., Trachana, K., Roth, A., Kuhn, M., Muller, J., Arnold, R., Rattei, T., Letunic, I., Doerks, T., Jensen, L.J., von Mering, C., Bork, P.: eggNOG v3.0: orthologous groups covering 1133 organisms at 41 different taxonomic ranges. Nucleic Acids Res. **40**(D1), D284 (2012)
18. Stothard, P., Van Domselaar, G., Shrivastava, S., Guo, A., O'Neill, B., Cruz, J., Ellison, M., Wishart, D.S.: BacMap: an interactive picture atlas of annotated bacterial genomes. Nucleic Acids Res. **33**(suppl. 1), D317–D320 (2005)
19. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
20. Chawla, N., Karakoulas, G.: Learning from labeled and unlabeled data: an empirical study across techniques and domains. J. Artif. Intell. Res. **23**(1), 331–366 (2005)
21. Reddy, T., Thomas, A.D., Stamatis, D., Bertsch, J., Isbandi, M., Jansson, J., Mallajosyula, J., Pagani, I., Lobos, E.A., Kyrpides, N.C.: The genomes online database (GOLD) v.5: a metadata management system based on a four level (meta)genome project classification. Nucleic Acids Res. **43**(D1), D1099 (2015)
22. Land, M.L., Hyatt, D., Jun, S.R., Kora, G.H., Hauser, L.J., Lukjancenko, O., Ussery, D.W.: Quality scores for 32,000 genomes. Stand. genomic sci. **9**(1), 20 (2014)