Project Report

Self-labeled techniques for semi-supervised learning

Mark Laane

Abstract. The aim of this project is to reimplement a selection of semi-supervised learning techniques surveyed by Isaac Triguero et al. in paper [1] and to independently reproduce the reported results. For this, two self-labeled techniques were chosen and implemented in Python programming language: Standard self-training and Tri-Training. Two well-known UCI datasets were chosen for testing the implementations: abalone and dermatology. The transductive and inductive classification accuracy of algorithms was measured on the datasets. The classification results achieved in this project are much lower than the results achieved by Triguero et al. indicating an inferior implementation of the algorithms.

1 Background

Semi-Supervised learning is a learning paradigm that joins unsupervised and supervised learning paradigms. It can be useful on data that has small amount of labeled data and a large amount of unlabeled data. In supervised learning only labeled data can be used, but in semi-supervised learning the unlabeled data is leveraged to provide better models. It is useful in situations where there is large amount of unlabeled data and labeling all samples is unviable. [1]

Self-labeled techniques are a subset of semi-supervised classification techniques that grow the labeled dataset by iteratively labeling some samples from the unlabeled dataset. [1]

2 Base classifiers

Some Self-labeled techniques build on supervised learning classifiers by using them as base classifiers. The base classifiers are used on labeled data to predict the labels of the unlabeled samples. By incorporating those predictions to the labeled dataset, the size of the labeled dataset is made larger and base classifier is retrained with this larger dataset. [1]

In the paper, 4 base classifiers were used with self-labeled methods: K-nearest neighbor (KNN), C4.5, Naive Bayes (NB) and Support vector machines (SVM). They all are classic and well known classifiers. In this project KNN and NB was used and the C4.5 decision tree classifier was substituted with CART as it is offered in Scikit-learn library and is similar to C4.5. [2]

3 Implemented algorithms

In this project two self-labeled techniques were implemented: Standard self-training and Tri-Training. Per classification proposed by Isaac Triguero et al. both are single-view, single learning and incremental learning algorithms. This means that the feature space does not have to be split into independent views (single-view), the labeled dataset is only incrementally enlarged from unlabeled dataset (incremental) and identical learners or base classifiers are used (single-learning). The algorithms differ in the number of classifiers used: Self training is a Single-classifier algorithm and uses only one instance of base classifier. Tri-Training is a multi-classifier and uses three instances of a given base classifier.

3.1 Standard Self-Training

The implementation of Standard Self-Training is based on description of the algorithm in paper [3]. Training a Standard Self-Training classifier is an iterative process where the base classifier retrained multiple times. The base classifier is initially trained with initial labeled samples. Then it is used for labelling the unlabeled samples and the most confident predictions are added to the labeled set. Then the classifier is retrained with this extended labeled set. The training and labeling process is repeated until all the training samples are labeled or maximum number of iterations is reached.

The algorithm can be summarized as follows:

Notation: C – base classifier used D – initial training dataset L – labeled training samples, $L \subseteq D$ U – unlabeled training samples, $U \subseteq D$
1. Train C on L 2. Use C to predict labels on U and select most confident predictions 3. Remove the most confident predictions from U and add them to L with the predicted labels 4. Repeat steps 1-4 until U is empty or the maximum number of iterations is reached. (40 in our implementation)

The final C can then be used to classify unseen data.

3.2 Tri-Training

The implementation of Tri-Training is based on description of the algorithm in paper [4]. In Tri-Training, three base classifiers are trained on randomly subsampled sets of the labeled data. Then each of them will be iteratively trained by also considering labels that two other base classifiers have predicted. The predictions from the other two base classifiers inherently introduce noise by mislabeling some samples. To set an upper bound for classification noise rate during the training process a restriction is set in place. Chosen base classifier will only be retrained if following equation holds

$$0 < \frac{e^t}{e^{t-1}} < \frac{|L^{t-1}|}{|L^t|} < 1 \quad (1)$$

where $|L^t|$ is the number of samples that can be added in the t-th round (agreed by the other two base classifiers) and et is the upper bound of classification error rate of those labeled samples L^t in t-th round. If $|L^t|$ is too large to satisfy equation 1, then the number of samples can be reduced by randomly subsampling the set so it would contain s samples, where $s = \left\lceil \frac{e^{t-1}|L^{t-1}|}{e^t} - 1 \right\rceil$. The algorithm continues as long there is at least one Ci, where equation 1 can be satisfied.

The algorithm can be summarized as follows:

1. Randomly subsample L to form three sets of labeled data L_1, L_2 and L_3
2. Train three instances of C, each Ci on Li
3. For each Ci:
   i. From U take samples where other two classifiers agree L_{agree} (two other classifiers predict the same label)
   ii. If restrictions on the added classification noise are met, retrain Ci with the $L_i \cup L_{agree}$
4. Repeat step 3 until none of the classifiers are retrained any more.

The three trained classifiers can then be used for classification by using majority voting.

4 Experiments

10-fold cross-validation procedure was used to validate the classification algorithms. The 10% in each split was kept for testing and the rest used for training. To simulate dataset with unlabeled samples, each training dataset was further split in two: unlabeled and labeled samples. From unlabeled samples, the labels were removed. Four labelling-ratios were tested: 10%, 20%, 30% and 40%

Two accuracy scores were measured in the experiments: accuracy of transductive learning and accuracy of inductive learning. Transductive accuracy describes the classifier's ability to correctly predict the labels of unlabeled samples used during the training process. Inductive accuracy describes the classifiers ability to correctly predict labels of unseen data – test data that is withhold and not been used for training.

5 Results

Each self-labeled algorithm was tested with each base classifier the validation was performed on each dataset. The results can be seen in the table in Appendix 2 and a selection of confusion matrixes in Appendix 1.

As expected, raising labelling rate also raises the transductive and test accuracy when dermatology datset is used. Abalone dataset proves to be more difficult to classify and extra labels do not always improve the results.. There is also an anomalous result with

unknown source, where of Tri-Training used with Naïve Bayes has the highest accuracy on 10% labelling ratio. Both algorithms perform worst with Naïve Bayes as a base classifier and best with CART base classifier. Triguero et al also reported good performance with another similar decision tree algorithm C4.5. On dermatology dataset, both Self-Training with CART and Tri-Training with CART achieved best results reaching towards 90% of accuracy.

Generally the classification accuracy results outputted by the implemented algorithms are lower than the results achieved by Triguero et al. indicating a poor performance of the implementations. Also, high standard deviation indicates poor performance, as the results seems to strongly depend on the fold that the data is being tested on.

6 Conclusions

In this project two self-labeled techniques - Standard self-training and Tri-Training were implemented and tested on two datasets: abalone and dermatology. The classification accuracy of the implemented algorithms is lower than the results achieved by Triguero et al. This could be explained by poor implementation. Therefore, further work should ensue to locate problematic parts of the implementations and fix them to improve prediction accuracy.

References

[1] I. Triguero, S. García and F. Herrera, "Self-labeled techniques for semi-supervised learning," *Knowledge and Information Systems,* vol. 42, no. 2, pp. 245--284, 2015.

[2] "Scikit-learn Decision Trees," [Online]. Available: http://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart.

[3] Y. D, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd annual meeting of the association for computational linguistics*, 1995.

[4] L. M. Zhou ZH, "Tri-training: exploiting unlabeled data using three classifiers," *IEEE,* 2005).

Appendix 1: Confusion matrixes

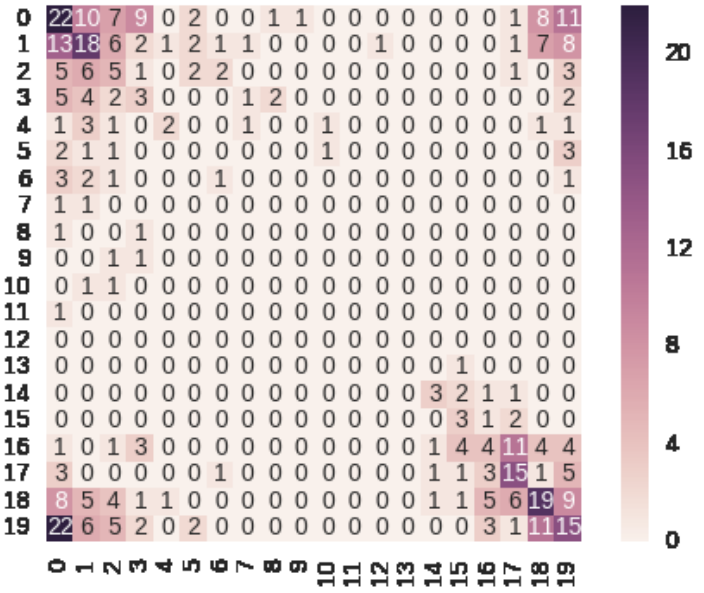

Fig. 1. Self-Training (CART), abalone, 40% labeling ratio

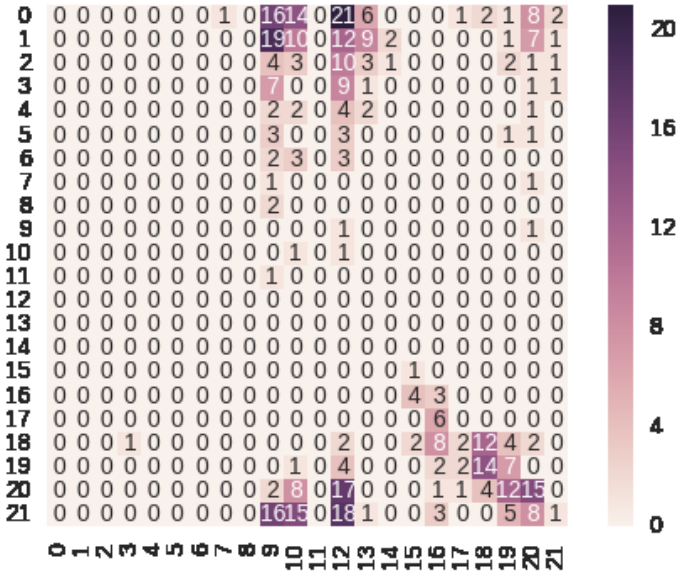

Fig. 2. Self-Training (NB), abalone, 40% labeling ratio

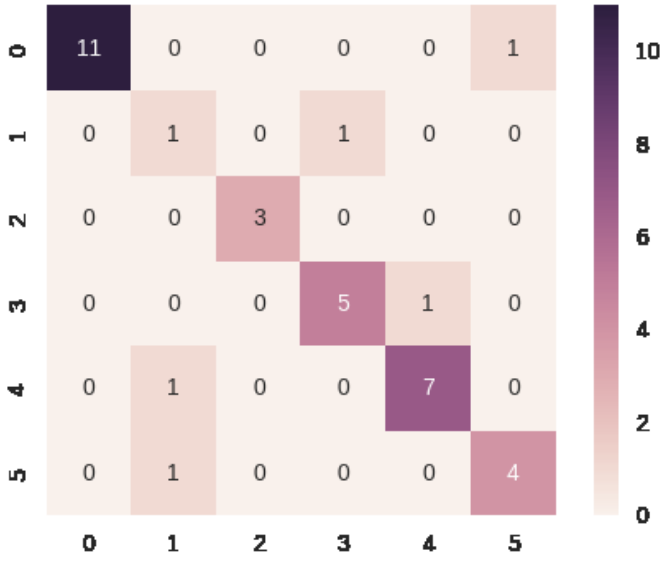

Fig. 3. Self-Training (KNN), dermatology, 40% labeling ratio

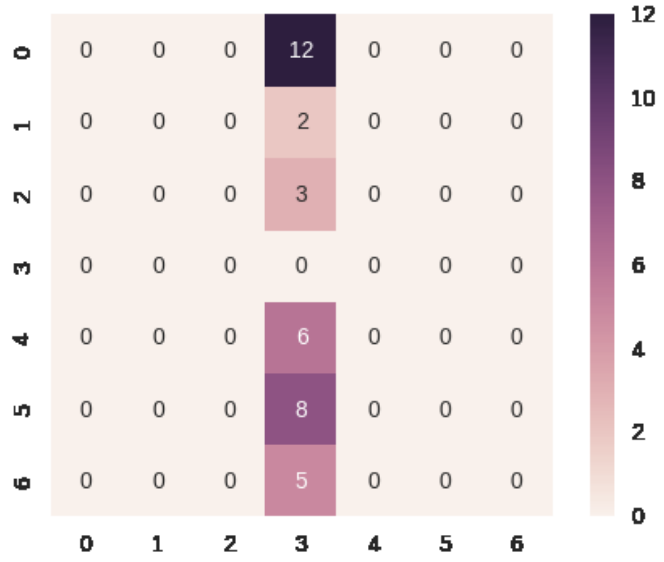

Fig. 4. TriTraining (NB), dermatology, 40% labeling ratio

Appendix 2: Testing results

	classifier	dataset	Labeling rate	Test mean	Test std	Trans mean	Trans std
0	TriTraining (KNN)	abalone	0.1	0.159610	0.093613	0.186119	0.063314
1	TriTraining (KNN)	abalone	0.2	0.185034	0.084115	0.167961	0.084159
2	TriTraining (KNN)	abalone	0.3	0.152725	0.092253	0.186408	0.063170
3	TriTraining (KNN)	abalone	0.4	0.146791	0.085206	0.182644	0.061573
4	TriTraining (KNN)	dermatology	0.1	0.478078	0.061992	0.443366	0.035022
5	TriTraining (KNN)	dermatology	0.2	0.647748	0.070364	0.668955	0.020381
6	TriTraining (KNN)	dermatology	0.3	0.737913	0.080060	0.744788	0.015202
7	TriTraining (KNN)	dermatology	0.4	0.773348	0.083499	0.800413	0.033052
8	TriTraining (NB)	abalone	0.1	0.090718	0.035779	0.044423	0.044664
9	TriTraining (NB)	abalone	0.2	0.085693	0.053914	0.043761	0.053617
10	TriTraining (NB)	abalone	0.3	0.090472	0.058631	0.057842	0.047577
11	TriTraining (NB)	abalone	0.4	0.088085	0.057578	0.082292	0.041538
12	TriTraining (NB)	dermatology	0.1	0.401126	0.300181	0.406665	0.275658
13	TriTraining (NB)	dermatology	0.2	0.278153	0.210628	0.251790	0.211221
14	TriTraining (NB)	dermatology	0.3	0.166366	0.116880	0.196640	0.110576
15	TriTraining (NB)	dermatology	0.4	0.155030	0.097225	0.180885	0.097111
16	TriTraining (CART)	abalone	0.1	0.167789	0.080768	0.181449	0.061071
17	TriTraining (CART)	abalone	0.2	0.197977	0.063281	0.178936	0.060470
18	TriTraining (CART)	abalone	0.3	0.192957	0.041478	0.197925	0.011228
19	TriTraining (CART)	abalone	0.4	0.169758	0.070042	0.198067	0.009981
20	TriTraining (CART)	dermatology	0.1	0.715766	0.152987	0.769580	0.041982
21	TriTraining (CART)	dermatology	0.2	0.896321	0.074975	0.918764	0.020228
22	TriTraining (CART)	dermatology	0.3	0.901727	0.068320	0.916243	0.027497
23	TriTraining (CART)	dermatology	0.4	0.907132	0.064047	0.946829	0.014985
24	Self-Training (KNN)	abalone	0.1	0.242754	0.084157	0.237461	0.007264
25	Self-Training (KNN)	abalone	0.2	0.230311	0.083290	0.233099	0.011785

	classifier	dataset	Labeling rate	Test mean	Test std	Trans mean	Trans std
26	Self-Training (KNN)	abalone	0.3	0.229596	0.073373	0.232699	0.008617
27	Self-Training (KNN)	abalone	0.4	0.227203	0.074212	0.226223	0.008845
28	Self-Training (KNN)	dermatology	0.1	0.549399	0.072864	0.506809	0.052777
29	Self-Training (KNN)	dermatology	0.2	0.696772	0.045663	0.692879	0.026361
30	Self-Training (KNN)	dermatology	0.3	0.748574	0.053008	0.737414	0.025927
31	Self-Training (KNN)	dermatology	0.4	0.781381	0.057703	0.785715	0.030130
32	Self-Training (NB)	abalone	0.1	0.075651	0.023860	0.065025	0.004766
33	Self-Training (NB)	abalone	0.2	0.086187	0.033949	0.085060	0.010414
34	Self-Training (NB)	abalone	0.3	0.091458	0.036784	0.084710	0.006705
35	Self-Training (NB)	abalone	0.4	0.091689	0.040028	0.083315	0.005651
36	Self-Training (NB)	dermatology	0.1	0.676877	0.250911	0.645830	0.256829
37	Self-Training (NB)	dermatology	0.2	0.196396	0.069305	0.194363	0.012955
38	Self-Training (NB)	dermatology	0.3	0.196396	0.069305	0.215726	0.040900
39	Self-Training (NB)	dermatology	0.4	0.196396	0.069305	0.223930	0.035261
40	Self-Training (CART)	abalone	0.1	0.208044	0.057041	0.210919	0.012029
41	Self-Training (CART)	abalone	0.2	0.212577	0.058447	0.210920	0.010081
42	Self-Training (CART)	abalone	0.3	0.209482	0.045617	0.211227	0.010464
43	Self-Training (CART)	abalone	0.4	0.212609	0.051943	0.210881	0.015029
44	Self-Training (CART)	dermatology	0.1	0.778153	0.107618	0.790551	0.068355
45	Self-Training (CART)	dermatology	0.2	0.879730	0.089891	0.911953	0.027151
46	Self-Training (CART)	dermatology	0.3	0.877102	0.071579	0.901929	0.026938
47	Self-Training (CART)	dermatology	0.4	0.909835	0.068335	0.935674	0.015822