Tri-Testing: A Novel Semi-Supervised Learning Method based on Ensemble Learning and Active Learning

Shota Kubo[1], Masahiro Terabe[1], Kazuo Hashimoto[1], and Ryuta Ono[2]

[1] G.S.I.S., Tohoku University,
6-6-11-304 Aramaki-Aza-Aoba, Aoba-Ku, Sendai, 980-8579, Japan
{kubo, terabe, kh}@aiet.ecei.tohoku.ac.jp
[2] NEC Corporation
r-ono@bk.jp.nec.com

Abstract. In many practical problems such as web page classification, it is highly expensive to obtain labeled training data set. Semi-supervised learning is a method to form a training data set by labeling abundant unlabeled data with classifiers learned from a small number of labeled data. It has a remarkable advantage to reduce the cost of labeling. We propose Tri-Testing, a novel semi-supervised learning method. We improved Tri-Training, a semi-supervised learning method based on ensemble learning, by introducing active learning. Tri-testing is defined as a combination of an ensemble learning process with three different classifiers, and an active learning process which selectively choose unlabeled data set to be labeled. The experiment shows that the proposed method derives more accurate classifier than existing other method, especially in the case where only small labeled data is available as training data set.

1 Introduction

In supervised learning scheme, classifiers are trained by a set of labeled data, and its accuracy is improved as the size of training data set increases. However, providing a sufficient size of training data set is sometimes difficult because it requires human expertise in the field under consideration for accurate labeling. There are cases where a large number of unlabeled instances are easy to obtain but labeled instances are hard. Web page classification is such a case. Semi-supervised learning increases the number of labeled data by labeling the unlabeled data with classifiers by themselves, and shows a better performance than supervised learning using only the original set of training data.

The concept of semi-supervised learning started to be studied in the mid '60s[4] as a probabilistic approach, such as EM algorithm, generative model, discriminative model, and so forth. These approaches are all parametric methods, and require sufficient supply of training data.

Blum et al. have proposed a PAC learning based algorithm, Co-Training[2]. Since PAC learning is not sensitive to the distribution of cases, Co-Training

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 175–184

Received 17/07/07
Accepted 31/08/07
Final version 23/09/07

works well with a small number of labeled data, and is noise tolerant. But Co-Training works under the strong assumption that cases are to be divided into two sufficient and redundant *view*. Which means attribute set can split into two independent subsets. This constraint sometimes prevent from Co-Training being applied to real problem.

Several methods are derived from Co-Training. Co-Testing[5] introduces an active-learning technique into Co-Training and Co-EM uses EM with Co-Training. Zhou et al.[8] proposes Tri-Training, an ensemble learning with three classifiers, each of which is trained by bootstrap sampling of labeled instance set. Tri-Training avoids constraint on Co-Training.

In this paper, we propose Tri-Testing, a novel semi-supervised learning method inspired by Tri-Training. Tri-Testing is an active learning process which selects unlabeled data to improve the accuracy of classification using three classifiers.

The following section is composed as follows. Section 2 discusses the conventional methods and Section 3 presents the proposed method. Section 4 and 5 shows our experiment that compare with other methods. Finally, Section 6 is the conclusion and future work.

2 Semi-supervised Learning

Semi-supervised learning trains classifiers using a set labeled data, as supervised learning does. Classifiers predict a label of each unlabeled data, selectively add predicted data, which is expected to improve the performance the most, to the original training data set, then updates classifiers with an updated training data set. The process iterates until the performance saturates.

Algorithms are different in according to the followings;

- How to generate classifier
- How to select unlabeled instance to label
- How to make final prediction

2.1 Co-Training

Co-Training[2] splits an attribute set of labeled data into two independent sets A_1 and A_2 which authors called as *"view(s)"*, and divide the labeled data into S_1 and S_2 according to the views. The original label y is given to splitted instances. Then, two classifiers h_1 and h_2 from each labeled data sets S_1 and S_2 are induced. Each classifier h_1 and h_2 predicts a label of any instance x in unlabeled dataset U, and allows h_i to label instances selectively where high confidence is obtained. Finally, classifiers are updated with both S_1 and S_2 and self-labeled instances. The whole process iterates k (k is an optional parameter) times.

Co-Training is theoretically proven that unlabeled instance supports classify ability. However, there are problem about applicability in real data because it is sometimes difficult to define view appropriately even though the view definition is a key performance factor of Co-Training. Additionally, some of the data set does not include sufficient number of attributes to construct view.

2.2 Co-Testing

Co-Testing[5] introduces an active learning approach to Co-Training based on the idea that active learning works better with multiple views than a single. Co-Testing selects instances with higher confidence from a set of instances whose predicted labels are not consistently given by two classifiers, and asks human experts to label and teach them.

Co-Testing improves effectively by collaborate with human expert. However Co-Testing also requires many attributes, as Co-Training does.

2.3 Tri-Training

Co-Training requires many attributes to define views appropriately. To make free to this constraints, Tri-Training[8] employs an ensemble learning approach like Bagging[3] with three classifier.

First, Tri-Training prepares three different and same size data sets S_1, S_2, and S_3 by bootstrap resampling the labeled data set L, and generates classifiers h_1,h_2, and h_3 respectively. In labeling phase, each classifier predicts a label for unlabeled instance u if two classifiers predict the same label. For instance, h_1 predicts a label y for an unlabeled instance u where $h_2(u) = h_3(u) = y$. Then classifiers are retrained with using both self-labeled and originally labeled instances.

Tri-Training shows a better classification performance than Co-Training if dataset has insufficient number of attributes. Tri-Training creates a labeled instance and add to the labeled instance set if two other classifiers, h_2, h_3, agree, regardless of the prediction by the classifier h_1. It is expected that the probability for the classifier h_1 to agree with h_2 and h_3 is not negligibly small, the performance improvement will saturate.

3 Tri-Testing

We propose Tri-Testing, a novel semi-supervised learning method by introducing an active learning approach to Tri-Training. As Tri-Training does, Tri-Testing generates three classifiers h_1,h_2, and h_3 from S_1, S_2, and S_3 which are sampled from labeled data set L by bootstrap resampling.

Tri-Testing asks human expert to label some of the instances that prediction disagrees among classifiers, and retrain classifiers with updated labeled instance set. Tri-Testing is designed to reduce this labeling cost of human expert by selecting as few unlabeled instances as possible to improve the overall accuracy in classification.

3.1 Concept of Tri-Testing

For instance, think about learning from dataset which has two class, "YES" or "NO". After initial learning, the number of combination of prediction from three classifier h_1,h_2, and h_3 is 2^3, as shown by table 1.

Table 1. Combination of label prediction by three classifiers.

Region	h_1	h_2	h_3
R_1	YES	YES	YES
R_2	YES	YES	NO
R_3	YES	NO	YES
R_4	YES	NO	NO
R_5	NO	YES	YES
R_6	NO	YES	NO
R_7	NO	NO	YES
R_8	NO	NO	NO

By predictions of three classifier h_1,h_2, and h_3, instances are classified into 8 region, R_1 to R_8. R_1 and R_8 regard as classify instance correctly because all classifiers predict the same label. The classifiers have already learned the instance label correctly so that labeling instance in R_1 or R_8 and retraining classifier may not boot classify accuracy significantly. The other hand, label prediction for instance in R_2 to R_7 is not agreed among all three classifiers. In this case, we can assume that some of the classifiers have not learned the instance label efficiently. Therefore, if human teach the label to such instance, at least one of classifier's performance probably refine.

3.2 Tri-Testing Algorithm

In this section, we explain how Tri-Testing works referring to the pseudo code of Tri-Testing algorithm shown in table 2.

Let x be the instance, y the label by human expert, l the true label, L the labeled data set, U the unlabeled data set, *Learn* the learning algorithm, S_i the bootstrap resampled L, h_i the classifier induced from S_i, $ContentionPoints[i]$ the list of candidate for label, L_i the labeled instance, and *saveU* the buffer for contain unlabeled instance.

At first, Tri-Testing prepares an initial training data set with different class distribution S_1, S_2, and S_3 by bootstrap resampling from initial labeled data L. Three classifiers h_1, h_2, and h_3 are generated from data set S_1, S_2, and S_3. The classifiers h_1, h_2, and h_3 are slightly different with each other because the training data set S_1, S_2, and S_3 has different instance composition.

The next step is a selection of instance for human labeling, which is the most important phase in Tri-Testing. Each classifier h_i predicts the label of every instance x in unlabeled data set U. If only one classifier h_i predicts the label of instance x differently among three classifiers, the instance x is added to the $ContentionPoints[i]$. The instances in $ContentionPoints[i]$ are candidates of asking the label to human expert.

Classifier h_i can improve its prediction accuracy when a correct label is given to the instance for which h_i predicts wrongly with high confidence. Tri-Testing select an instance has most confidence by h_i from $ContentionPoints[i]$ list for

Table 2. Pseudo code describing the Tri-Testing Algorithm

Input: L, U, $Learn$
for $i \in \{1..3\}$ **do**
  $S_i \leftarrow BootstrapSample(L)$ //bootstrap resample
  $h_i \leftarrow Learn(S_i)$ //generate h_i
end of for
for k iterations //k is optional number
  for $i \in \{1..3\}$ **do**
    $ContentionPoints[i] \leftarrow \phi$
  end of for
  for $i \in \{1..3\}$ **do**
    for every $x \in U$ **do** $(j, k \neq i)$
      if $h_i(x) \neq h_j(x)$ and $h_j(x) = h_k(x)$
      then $ContentionPoints[i] \leftarrow x \ \ ContentionPoints[i]$
    end of for
    $x \leftarrow SelectQuery(ContentionPoints[i])$
    $y \leftarrow Label(x)$ //hand-ladel by human
    $delete(x, ContentionPoints[i])$
    if $y \neq h_i(x)$
    then $L_i \leftarrow L_i\{(x, y)\}$
    else then $L_j \leftarrow L_j\{(x, y)\}$
      $L_k \leftarrow L_k\{(x, y)\}$
  end of for
  for $i \in \{1..3\}$ **do**
    $saveU \leftarrow saveUContentionPoints[i]$
    $h_i \leftarrow Learn(LL_i)$
  end of for
  $U \leftarrow saveU$
end of for k iterations
Output: $h(x) \leftarrow \arg\max_{l \in label} \sum_{i:h_i(x)=y} 1$ //majority vote

retraining. An instance which has high confidence may locate in far from class bound.

In the case shown in table 1, this method has the following assumptions;

- h_1 makes wrong predictions for instances in R_4 and R_5,
- h_2 makes wrong predictions for instances in R_3 and R_6,
- h_3 makes wrong predictions for instances in R_2 and R_7.

An instance x in $ConjunctionPoint[i]$ with highest confidence of h_i is picked out. The instance x is labeled by human expert. If the label of instance x predicted by $h_i(x)$ and human expert y is different, the instance x is added to the training set L_i. If the $h_i(x)$ and y is the same label, it indicates that h_j and h_k predicts wrong class. In this case, the instance x is added to L_j, L_k for retraining.

Finally, three classifiers retrained with labeled data set L_i and original instance set L. Tri-Testing iterates this process specified times, and generates final

classifiers. The iteration also stops in the case all *ContentionPoints* are empty, because it assume there is no instance efficient for improving the classifier.

4 Experiment 1: Unlabeled Instance Rate and Prediction Accuracy

4.1 Exprimental Setting

To confirm the efficiency of Tri-Testing, we conducted the following experiments. We used dataset from UCI Machine Learning Repository[1]. All data sets are binary class, and are used in the experiment of [8]. Table 4 indicates specifications of each dataset. "Pos." and "Neg." shows proportions of positive and negative instances in the data set respectively.

Table 3. The specifications of data sets for experiment.

Dataset	# of inst.	# of attr.	# of cls.	Pos./Neg. (%)	Cont. Attr. (%)	Missing Value
AUSTRALIAN	690	14	2	55.5/44.5	42.8	no
BUPA	345	6	2	42.0/58.0	100.0	no
COLIC	368	22	2	63.0/37.0	31.8	yes
DIABETES	768	8	2	65.1/34.9	100.0	no
GERMAN	1000	20	2	70.0/30.0	35.0	no
IONOSPHERE	351	34	2	35.9/64.1	100.0	no
KR-VS-KP	3196	36	2	52.2/47.8	0.0	no
SICK	3772	39	2	6.1/93.9	19.4	yes
TIC-TAC-TOE	958	9	2	65.3/34.7	0.0	no
VOTE	435	16	2	61.4/38.6	0.0	yes

We compare the prediction accuracy of following three learners.

- **NB (Naive Bayes):** Single classifier induced by Naive Bayes.
- **Tri-Training:** Ensemble of three classifiers. Each classifiers induced by Naive Bayes.
- **Tri-Testing:** Ensemble of three classifiers. Each classifiers induced by Naive Bayes. Maximum iteration times $k = 5$.

In the first experiment, we investigate the effect of unlabeled instance rate on prediction accuracy. In order to take into account the influence of the sampling in Tri-Training and Tri-Testing ten fold cross validation was repeated 10 times and the error rate was evaluated by taking the average over the traials.

4.2 Results

Table 4 indicates error rate of each learners. In all of the data set, the error rate is not so different among all learners when the unlabeled instance rate is under the 90%. However, in most of the cases, Tri-Testing shows better performance than the other learners when the unlabeled instance rate is over the 90%.

We also run a paired t-test between Tri-Training and Tri-Testing. The prediction accuracy of Tri-Testing performs better in the significant level over the 95% is bolded. In AUSTRALIAN, COLLIC, IONOSPHERE, and KR-VS-KP, Tri-Testing performs better than other algorithms with statistically significant level over 95%.when the unlabeled instance rate is over 95%. Additionally, in DIABETES, GERMAN, TIC-TAC-TOE, and VOTE, Tri-Testing outperforms at confidence level 95% when unlabeled instance rate is over 99%.

4.3 Discussion

In most of the cases, Tri-Testing performs better than Tri-Training when the unlabeled instance rate high. This results shows that active learning works well to improve the prediction accuracy of each classifiers and ensemble classifier.

In the case of BUPA, IONOSPHERE, Tri-Testing and Tri-Training perform worse than Naive Bayes when the unbalanced instance rate is 99%. This is because that the number of training data for weak learners in ensemble. The weak learners learn from the training data which includes only 3 or 4 instances. This effect is investigated in our former research[7].

On the other hands, in the case of SICK, Tri-Testing can not improve the prediction accuracy, and Tri-Training also can not. SICK's class distribution is extremely biased. In this data set, the ratio of positive class is only 10%. We assume that this biased class distribution makes it hard to learn for ensemble learner such as Tri-Training and Tri-Testing.

5 Experiment 2: Class Distribution and Prediction Accuracy

5.1 Experimental Setting

As confirmed in experiment 1, Tri-Testing does not perform well when the class distribution is biased as SICK data set. If the class distribution bias is canceled, the performance of Tri-Testing would be improved. Gao et al.[9] demonstrated the effect of class distribution bias canceling by sampling. In this experiment, we investigate the effect of class distribution bias to the prediction accuracy of proposed method.

We use the same data set as experiment 1. We compare two cases;

- **Case1 (Uniform Class Distribution):** The training data for weak learners in ensemble is sampled to be the class distribution uniformly.

Table 4. The classification error rate of NaiveBayes, Tri-Training and Tri-Testing.

Dataset	Method	Unlabeled Instance Rate(%)							
		20	40	60	80	90	95	97	99
AUSTRALIAN	NB	22.8	22.9	22.6	22.3	21.2	22.8	24.7	31.5
	Tri-Training	22.8	22.9	22.6	22.3	21.0	23.1	24.0	28.0
	Tri-Testing	22.6	22.7	22.5	21.7	20.7	**21.4**	**22.3**	**23.6**
BUPA	NB	44.7	44.3	43.5	46.7	46.1	45.5	46.6	48.5
	Tri-Training	45.9	45.5	44.3	47.1	46.7	46.3	46.7	42.7
	Tri-Testing	44.3	44.1	43.3	46.2	46.2	45.3	45.5	46.3
COLIC	NB	21.5	21.8	21.7	23.3	25.3	29.5	33.2	49.5
	Tri-Training	21.6	21.9	21.8	22.9	24.9	29.5	30.8	54.6
	Tri-Testing	21.4	21.6	21.3	22.2	23.0	**25.7**	**27.7**	**39.8**
DIABETES	NB	24.8	25.0	25.1	26.2	27.4	29.4	31.4	34.9
	Tri-Training	24.7	25.0	25.1	26.2	27.5	29.3	31.5	35.0
	Tri-Testing	24.8	24.9	24.8	26.0	27.0	28.3	30.2	32.4
GERMAN	NB	25.1	25.3	25.8	26.8	29.3	31.0	32.5	37.0
	Tri-Training	25.2	25.3	25.9	26.9	29.2	30.9	32.7	35.2
	Tri-Testing	25.0	25.3	25.8	26.7	28.9	30.7	32.3	**33.8**
IONOSPHERE	NB	17.9	17.8	18.1	18.1	18.8	19.0	26.9	36.3
	Tri-Training	18.0	18.1	18.8	19.0	21.1	22.3	30.5	35.4
	Tri-Testing	17.3	17.2	17.4	16.3	**16.2**	**16.3**	**21.4**	37.0
KR-VS-KP	NB	12.4	12.8	13.0	13.7	16.5	18.5	20.3	28.0
	Tri-Training	12.4	12.8	13.1	13.8	16.6	18.8	20.9	31.0
	Tri-Testing	12.4	12.7	12.9	13.6	15.9	**17.6**	**19.0**	**25.5**
SICK	NB	7.3	7.3	7.7	7.8	7.6	6.4	6.3	6.8
	Tri-Training	7.3	7.3	7.8	7.9	7.7	6.6	6.4	7.0
	Tri-Testing	7.3	7.4	8.0	8.2	8.5	7.5	7.7	7.2
TIC-TAC-TOE	NB	29.7	29.6	29.8	30.5	32.4	34.6	36.4	40.7
	Tri-Training	9.8	29.6	29.8	30.5	32.6	34.8	36.7	41.1
	Tri-Testing	29.6	29.6	29.9	30.4	32.3	33.9	35.7	**37.4**
VOTE	NB	9.7	9.8	9.9	9.9	10.0	10.4	10.2	11.9
	Tri-Training	29.7	9.7	9.8	10.0	10.2	10.6	10.6	11.7
	Tri-Testing	9.9	9.8	9.7	9.7	9.6	9.7	9.3	**8.8**

– **Case2 (Original Class Distribution):** The training data for weak learners in ensemble is sampled randomly. The class distribution of training data is not different from that of labeled data set.

We run Tri-Testing in this two cases and evaluate the classification error rate in various unlabeled instance rate.

5.2 Results

We calculate the difference of error rate between case 1 and 2. If the difference is negative, the prediction accuracy is improved by canceling the class distribution bias by sampling. The experimental result is shown in table 5.

Table 5. Effect of canceling the class distribution bias.

Dataset	Unlabeled Instance Rate(%)			
	90%	95%	97%	99%
AUSTRALIAN	-0.1	1.37	1.10	9.42
BUPA	0.41	-0.88	-0.44	8.94
COLIC	0.86	1.36	2.22	-5.06
DIABETES	0.05	0.86	0.65	2.08
GERMAN	-0.17	-0.24	0.58	-0.88
IONOSPHERE	0.18	2.26	2.32	5.29
KR-VS-KP	0.11	0.43	0.55	1.08
SICK	-0.32	-0.58	-0.56	-0.17
TIC-TAC-TOE	0.02	-0.19	0.06	1.11
VOTE	0.10	0.07	0.63	0.98

The effect of prediction accuracy improvement is different among data set. However, in case of SICK which includes biased class distribution, the prediction accuracy is improved when the bias is canceled by sampling. This result shows the possibility that the prediction accuracy is improved by canceling the class distribution bias in the case the class distribution of original data set is biased.

6 Conclusion

Some of the interesting issues to be investigated remain. One of the issues is improving the active learning process to minimize the labeling cost of human expert and to maximize the performance. Using the confidence value of each classifier to select the label request instance is one of the promising solutions.

The other issue is design of ensemble classifier which includes adequate variance. In Co-Training, independent view is pre-defined to induce two variant classifiers. In the other hands, Tri-Training and Tri-Testing introduce bootstrap

resampling to generate slightly variant three training data set and classifiers. Introducing feature selection and construction mechanism [10] to generate adequate view automatically is interesting and to be investigated.

References

1. C. Blake, E. Keogh, C. J. Merz, "UCI repository of machine learning databases", *Department of Information and Computer Science, University of California, Irvine, CA*, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`
2. A. Blum, T. Mitchell, "Combining labeled and unlabeled data with co-training", *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp.92–100, 1998.
3. L. Breiman, "Bagging Predictors", *Machine Learning*, Vol.24, No.2, pp.123–140, 1996.
4. H. J. Scudder, "Probability of error of some adaptive pattern-recognition machines", *IEEE Transactions on Information Theory*, Vol.11, pp.363–371, 1965.
5. I. Muslea, S. Minton, C. A. Knoblock, "Selective Sampling With Naive Co-Testing: Preliminary Result", *The ECAI-2000 workshop on Machine Learning for information extraction*, 2000.
6. K. Nigam, R. Ghani, "Analyzing the Effectiveness and Applicability of Co-Training", *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pp.86–93, 2000.
7. M. Terabe, T. Washio, H. Motoda, "S^3*Bagging*: Fast Classifier Induction Method with Subsampling and Bagging. *Proceedings of Fourth International Symposium of Intelligent Data Analysis*, pp. 177-186, 2001.
8. Z. H. Zhou, M. Li, "Tri-training: exploiting unlabeled data using three classifiers", *IEEE Transactions on Knowledge and Data Engineering*, vol.17, pp.1529–1541, 2005.
9. J. Gao, W. Fan, J. Han and P. S. Yu, "A General Framework for Mining Concept-Drifting Data Streams with Skewed Distributionsh, *Proceedings of the 2007 SIAM International Conference on Data Mining*, 2007.
10. H. Liu, H. Motoda, "Feature Selection for Knowledge Discovery and Data Miningh, Kluwer, 1998.