An Ensemble Approach to Combining Expert Opinions

Hua Zhang[1], Evgueni Smirnov[1], Nikolay Nikolaev[2], Georgi Nalbantov[3], and Ralf Peeters[1]

[1] Department of Knowledge Engineering, Maastricht University,
P.O.BOX 616, 6200 MD Maastricht, The Netherlands
{hua.zhang,smirnov,ralf.peeters}@maastrichtuniversity.nl
[2] Department of Computing, Goldsmiths College, University of London,
London SE14 6NW, United Kingdom
n.nikolaev@gold.ac.uk
[3] Faculty of Health, Medicine and Life Sciences, Maastricht University,
P.O.BOX 616, 6200 MD Maastricht, The Netherlands
g.nalbantov@maastrichtuniversity.nl

Abstract. This paper introduces a new classification problem in the context of human computation. Given training data annotated by m human experts s.t. for each training instance the true class is provided, the task is to estimate the true class of a new test instance. To solve the problem we propose to apply a well-known ensemble approach, namely the stacked-generalization approach. The key idea is to view each human expert as a base classifier and to learn a meta classifier that combines the votes of the experts into a final vote. We experimented with the stacked-generalization approach on a classification problem that involved 12 human experts. The experiments showed that the approach can outperform significantly the best expert and the majority vote of the experts in terms of classification accuracy.

1 Introduction

Human computation is an interdisciplinary field involving systems of humans and computers capable of solving problems that neither party can solve better separately [4]. This paper introduces a new classification problem in the context of human computation and proposes an ensemble-related approach to that task.

The classification problem we define is essentially a single-label classification problem. Assume that we have m human experts that estimate the true class of instances coming from some unknown probability distribution. We collect these instances together with the experts' class estimates and label them with their true classes. The resulting instances' collection form our training data. In this context our classification problem is to estimate the true class of a new test instance, given the training data and the class estimates given by m human experts for that instance.

To solve the problem we defined above we propose to apply a well-known ensemble approach, namely the stacked-generalization approach [6]. The key idea is to view each human expert as a base classifier and to learn a meta classifier that predicts the class for a new instance given the class estimates provided by m human experts for that instance. This implies that the meta classifier combines the votes of the experts into a final vote. It is proposed to be learned using the training data given with expert class estimates and true classes. We experimented with the stacked-generalization approach on a classification problem that involved 12 human experts. The experiments showed that the approach can outperform significantly the best expert and the majority vote of the experts in terms of classification accuracy.

Our work can be compared with other work on classification considered in the context of human computation and crowdsourcing [5, 7]. In these two fields the main emphasis is on classification problems where the training data is labeled by experts only; i.e., the true instance classes are not provided. We note that our classification problem is conceptually simpler but somehow it has not been considered so far. There are many applications in medicine, finance, meteorology etc. where our classification problem is central. Consider for example a set of meteorologists that predict whether it will rain next day. The true class arrives in 24 hours. We can record the meteorologist predictions and the true class over a time period to form our data. Then stacked generalization is applied and thus we hopefully will be able to predict better than the best meteorologist or the majority vote of the meteorologists.

The remainder of the paper is organized as follows. Section 2 formalizes our classification task and describes the stacked generalization as an approach to that task. The experiments are given in Section 3. Finally, Section 4 concludes the paper.

2 Classification Problem and Stacked Generalization

Let X be an instance space, Y a class set, and $p(x, y)$ be an unknown probability distribution $p(x, y)$ over the labeled space $X \times Y$. We assume existence of m number of human experts capable of estimating the true class of any instance $(x, y) \in X \times Y$ according to $p(x, y)$. We draw n labeled instances $(x, y) \in X \times Y$ from $p(x, y)$. Any expert $i \in 1..m$ provides an estimate $y^{(i)} \in Y$ of the true class y of each instance x without observing y. This implies that the description x of any instance is effectively extended by the class estimates $y^{(1)}, ..., y^{(m)} \in Y$ given by the m experts. Thus, we consider any instance as a $m + 2$-tuple $(x, y^{(1)}, ..., y^{(m)}, y)$. The set of the n instances formed in this way results in training data D. In this context we define our classification problem. Given the training data D, a test instance $x \in X$, the class estimates $y^{(1)}, ..., y^{(m)} \in Y$ provided by the m experts for x, the classification problem is to estimate the true class for the instance x according to the unknown probability distribution $p(x, y)$.

Our solution to the classification problem defined above is to employ stacked generalization [6]. The key idea is to consider each human expert $i \in 1..m$ as a base classifier (providing class estimates) and then to learn a meta classifier that combines the class estimates of the experts into a final class estimate. The meta classifier is a function that can have two possible forms either $h : X, Y^m \rightarrow Y$ or $h : Y^m \rightarrow Y$. The difference is whether the instance descriptions in X are considered. Once the decision of the meta-classifier format is finalized we build the classifier using the training data D. We note that our use of the stacked generalization does not impose any restrictions on the type of the meta classifier (as opposed to [1]).

3 Experiments

For our experiments we chose a difficult language-style classification problem[4]. We had 317 sentences in English that were composed according to either a Chinese style or an American style. An example of two such sentences with the same meaning are given below:

- Chinese Style: "I recommend you to take a vacation."
- American Style: "I recommend that you take a vacation."

The sentences were labeled by 12 experts that did not know the true classes of those sentences. The language-style classification problem was to estimate the true class for any new sentence given the class estimates provided by the 12 experts for that sentence.

The language-style classification problem was indeed a difficult problem. The expert accuracy rates were in the interval [0.27, 0.54]. The mean accuracy rate was 0.39 and standard deviation was 0.08. The accuracy rate of the majority vote of the experts was 0.71.

The sentences with the labels of the 12 experts and their true classes formed our training data. We trained meta classifiers predicting the true class of the sentences. We considered two types of meta classifiers $h : X, Y^{12} \rightarrow Y$ and $h : Y^{12} \rightarrow Y$. The input of the first type of meta classifiers consisted of bag-of-word representation of the sentence to be classified and the classes provided by all the 12 experts. The input of the second type consisted of the classes provided by the 12 experts only. The output of both types of meta classifiers was the class estimate for the instance to be classified.

In addition we experimented with the meta classifiers with and without use of feature selection. The feature-selection procedure employed was the wrapper method based on greedy stepwise search [2].

The accuracy rates of the meta classifiers were estimated using 10-fold cross-validation. However, to decide whether these classifiers were good, we needed to determine whether their accuracy rates were statistically greater than those of the best expert and majority vote of the experts. We note that this is not a trivial problem, since the k-fold cross-validation is not applicable for the human experts employed. Nevertheless we performed paired t-test that we designed as follows. We split the training data randomly into k folds. For any fold we received: the class estimates provided by the meta classifiers, the class estimates of the best expert, and the class estimates of the majority vote of the experts for all the instances in the fold. Using this information we computed for any fold $j \in 1..k$ the accuracy rate a_j^m of the meta classifiers, the accuracy rate a_j^{be} of the best expert, and the accuracy rate a_j^{mv} of majority vote of the experts. Then we computed the paired difference $d_j = a_j^m - a_j^{be}$ ($d_j = a_j^m - a_j^{mv}$), and the point estimate $\bar{d} = (\sum_{j=1}^{k} d_j)/k$. Using this data the t-statistics that we used was $\frac{\bar{d}-\mu_d}{S_d/\sqrt{n}}$, where μ_d is the true mean and S_d is the sample standard deviation.

[4] The data can be freely downloaded from: https://dke.maastrichtuniversity.nl/smirnov/hua.zip.

Table 1. Accuracy rates of meta classifiers for the language-style classification task. s ($\bar{s}$) indicates (non-) presence of sentence representation in the input. w ($\bar{w}$) indicates (non-) use of wrapper. The rates in bold are significantly greater than the accuracy rate 0.71 of the majority vote of the experts on 0.05 significance level.

Classifier	$\bar{s}$-$\bar{w}$	s-$\bar{w}$	$\bar{s}$-w	s-w
AdaBoostM1	**0.78**	**0.76**	**0.76**	0.71
k-Nearest Neighbor	**0.75**	**0.76**	**0.77**	**0.78**
Logistic Regression	**0.76**	**0.72**	**0.76**	0.71
Naive Bayes	**0.76**	**0.72**	**0.78**	**0.76**
RandomForest	**0.73**	**0.75**	**0.76**	**0.74**

The accuracy rates of the meta classifiers are provided in Table 1. Since the majority vote of the human experts outperformed the best expert, the table shows the results of the statistical paired t-test of comparison of the accuracy rates of the meta classifiers and the majority vote of the human experts on 0.05 significance level[5].

Two main observation can be derived from Table 1:

(O1) 18 out of 20 meta classifiers have accuracy rate significantly greater than the accuracy rate 0.71 of majority vote of the experts.
(O2) the stacked generalization achieves the best classification accuracy rates when:
(O2a) the instances to be classified are represented by the expert estimates only, and
(O2b) feature selection is employed. In this case we achieved an average rate of 0.766.

During the experiments we recorded the running time of training the meta classifiers. The results are provided in Table 2. They show that:

(O3) wrapper-based meta classifiers require more time. Among them the most efficient are the meta classifiers that do not employ the sentence representation.
(O4) meta classifiers that do not use wrappers require less time. Among them the most efficient are the meta classifiers that do not employ the sentence representation.

4 Conclusion

This section analyzes observations (O1) - (O4) from section 3. Based on the analysis it provides final conclusions.

We start with observation (O1). This observation allows us to conclude that the stacked generalization can outperform significantly the best expert and the majority vote of the experts in terms of generalization performance . This implies that the classification problem we defined and the approach we proposed are indeed useful.

Observation (O2a) is a well-known fact in stacked generalization [1]. However in the context of this paper it has additional meaning. More precisely we can state that for

[5] For the sake of completeness we trained classifiers $h : X \to Y$ as well. Their accuracy rates were in interval [0.47, 053]; i.e., they were statistically worse than the experts' majority vote.

Table 2. Time (ms) for building meta classifiers. s ($\bar{s}$) indicates (non-) presence of sentence representation in the input. w ($\bar{w}$) indicates (non-) use of wrapper.

Classifier	$\bar{s}$-$\bar{w}$	s-$\bar{w}$	$\bar{s}$-w	s-w
AdaBoostM1	0.03	2.21	14.85	257.63
k-Nearest Neighbor	0	0	26.49	296.84
Logistic Regression	0.02	0.05	4.95	133.47
Naive Bayes	0	0.1	0.31	53.11
RandomForest	0.03	1.33	23.51	219.64

our classification problem we do have to know the class estimates of the experts only in order to receive the best accuracy rates. The input from the application domain (in our case English text) is less important. In addition we note that according to observations (O3) and (O4) the use of the expert class estimates only implies less computational cost.

Observation (O2b) is an expected result in context of feature selection. However it also has a practical implication for our classification problem, namely it allows to choose combination of the most adequate experts. In our experiments for example only half of the experts was chosen to maximize the accuracy. This means that we can reduce the number of human experts and thus the overall financial cost. Of course this has a price: increase of computational complexity according to observation (O3).

Future research will focus on the problem of human-experts' evolution. Indeed in real life the experts change due to many factors (e.g.; training, ageing etc.). Solving this problem will have a high practical impact. For that purpose we plan to apply techniques from concept drift [8] and transfer learning [3].

References

1. S. Dzeroski and B. Zenko. Is combining classifiers better than selecting the best one. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 123–130, 2002.
2. I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh. *Feature Extraction, Foundations and Applications*. Physica-Verlag, Springer, 2006.
3. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.
4. A. Quinn and B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011*, pages 1403–1412. ACM, 2011.
5. V. Raykar, S. Yu, L. Zhao, G. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.
6. D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.
7. Y. Yan, R. Rosales, G. Fung, M. Schmidt, G. Hermosillo Valadez, L. Bogoni, L. Moy, and J. Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. *Journal of Machine Learning Research*, 9:932–939, 2010.
8. I. Zliobaite. *Learning under Concept Drift: an Overview*. Technical Report. 2009, Faculty of Mathematics and Informatics, Vilnius University: Vilnius, Lithuania, 2009.