A Classification Model for Diverse and Noisy Labelers

Hao-En Sung[1], Cheng-Kuan Chen[1(✉)], Han Xiao[2], and Shou-De Lin[1]

[1] National Taiwan University, Taipei 10617, Taiwan
{b00902064,b98901048}@ntu.edu.tw, sdlin@csie.ntu.edu.tw
[2] Zalando, 10178 Berlin, Germany
han.xiao@zalando.de

Abstract. With the popularity of the Internet and crowdsourcing, it becomes easier to obtain labeled data for specific problems. Therefore, learning from data labeled by multiple annotators has become a common scenario these days. Since annotators have different expertise, labels acquired from them might not be perfectly accurate. This paper derives an optimization framework to solve this task through estimating the expertise of each annotator and the labeling difficulty for each instance. In addition, we introduce similarity metric to enable the propagation of annotations between instances.

Keywords: Noisy labeler · Crowdsourcing

1 Introduction

With the emerging of social networks and web services, it becomes popular to exploit crowdsourcing to obtain annotations of instances through online services such as *Amazon Mechanical Turk (AMT)*[1] for training a classification model. Although it is easy to obtain labels this way, those labels often come from imperfect labelers whose expertise toward the assigned task may vary significantly. Such noisy annotations can affect the performance of a traditional supervised machine learning model, which assumes all the training labels are reliable.

To address this issue, previous works [1–3] proposed a probabilistic framework to estimate the annotation quality from each annotator. One main disadvantage of such framework is that it fails to consider the feature-based similarities between instances. Moreover, they rely on certain predefined distribution to model annotator's expertise, which is often challenged in real scenario.

Instead of probabilistic framework, this paper proposes a novel optimization framework, which relaxes the predefined assumption of annotator's expertise toward instances. Our method further captures the similarity information shared among instances in feature space to yield a more effective solution. Our task can

H.-E. Sung and C.-K. Chen—denotes equal contribution.

[1] https://www.mturk.com/mturk/welcome.

J. Kim et al. (Eds.): PAKDD 2017, Part I, LNAI 10234, pp. 58–69, 2017.
DOI: 10.1007/978-3-319-57454-7_5

be represented using Fig. 1. On the left side of this figure, we are obtaining a set of training instances i_1 to i_3, and each instances is labelled by every annotator from u_1 to u_3. The spirit of our proposed optimization framework is shown as the right figure of Fig. 1. It not only learns the annotator-instance confidence (dashed line) but also acquires the instance-instance relationship in feature space (solid line) to improve the performance.

The main contribution of this paper can be summarized as follows.

1. We introduce a novel framework that enables the propagation of annotations between instances. It relaxes the probabilistic distribution presumption on annotators' expertise as well as the independence assumption between annotations, which are usually required by probabilistic models.
2. Our model learns the latent variables to capture the expertise of each annotator and the labeling difficulty of each instance, which are essential information for most active learning frameworks.
3. We have conducted experiments on several datasets to verify our model.

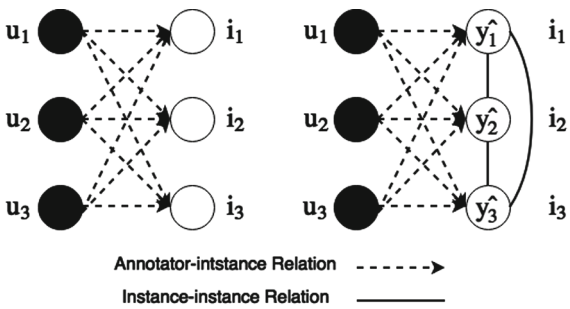


Fig. 1. Black nodes and white nodes represent annotators and instances, respectively. Dashed arrow that links one black node and one white node indicates the annotator-instance relationship; whereas, solid line that links two white nodes represents the instance-instance relationship.

The organization of this work is listed as follows. We first introduce the related works in Sect. 2. The formal problem definition and the derivation of our learning model are introduced in Sect. 3. In Sect. 4, we show the performance of our model on both simulated and real annotation datasets. We finally summarize this paper and propose future works in Sect. 5.

2 Related Work

There are mainly two kinds of scenarios for modeling multiple-annotator problems, and various algorithms solve either of them with different motivations. One of the prevalent frameworks tries to detect malicious annotators in order to

remove or flip their responses; while the other models rank the expertise of each annotator and re-weight the annotation results based on the ranking.

For the first scenario, two-coin model for annotators was proposed in [2,4] to detect potential malicious annotators, which is also known as MAP-ML algorithm. A classification model with weighting matrix w is obtained during model learning and the label of each instance with feature $\boldsymbol{x}_i$ equips the probability $\sigma(w^T \boldsymbol{x}_i)$ of being true, where $\sigma(\cdot)$ is the sigmoid function. Each annotator flips a coin with bias α^i as sensitivity if the label is predicted true; whereas a coin with bias β^i as specificity is flipped if the label is predicted false. Under this framework, nasty annotations will be flipped automatically during model learning. In his later work [5], it further defines a criterion to evaluate spammer during learning process. These two works implicitly assume that the sensitivity and specificity of each annotator are independent from instances and they neglect the possibility that one annotator might equip varied levels of expertise toward different instances, which is often challenged in real world applications.

For the second scenario, [3] uses Gaussian Mixture Model combined with MAP-ML (known as GMM-MAPML) to evaluate annotator performances. Later works in [6,7] further define a specific threshold to eliminate low-quality annotations during model learning. Another model proposed in [1] and his later extended learning and active learning works [8–12] use a probabilistic model $p(y_i^{(u)}|\boldsymbol{x}_i, z_i)$ to learn annotations provided by different annotators, where z_i is the ground truth, $\boldsymbol{x}_i$ is the feature vectors, and $y_i^{(u)}$ is the label of instance i given by annotator u. Apart from the first scenario, it assumes that each annotator has varied levels of expertise toward different kinds of problems, which implies $p(y_i^{(u)}|\boldsymbol{x}_i, z_i) \neq p(y_i^{(u)}|z_i)$. The labeling expertise from u to i can be calculated through Logistic Regression with Bernoulli or Gaussian model.

The above-mentioned methods rely on two strong assumptions: annotator expertise follows predefined distribution and annotation processes are independent with one another. In our work, we relax these two assumptions and further integrate the similarity relationship between instances into our model. We also notice that some recent works [13,14] address the similar problem as ours. However, one main difference is that their work focus on active learning while ours is to design a new learning framework.

3 Methodology

To convey our idea, we formally define the problem in Sect. 3.1. In Sect. 3.2, we propose our learning model with detailed derivations.

3.1 Problem Definition

This paper mainly focus on the binary classification task, and leave multi-class one as our future work. We consider a dataset with n instances $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, where $\boldsymbol{x}_i$ is a d dimensional feature vector for instance i, i.e. $\boldsymbol{x}_i \in \mathbb{R}^d$. Each instance i is assumed to be annotated by arbitrary number of annotators u with

label $y_i^{(u)} \in \{0, 1\}$, while we mainly follow the settings in [1] to consider full annotations in our following experiments.

The goal is to learn a model to predict each instance label by aggregating labels provided by all annotators toward all instances. Since the annotations are noisy, here we want to exploit the item similarity for a more robust model. For instance, if we want to predict the label $\hat{y}_1$ in Fig. 1, not only annotations toward instance 1 but also annotations toward instance 2 and 3 are taken into account, weighted by corresponding similarity between those items. The propagation of annotations weighted by similarity relationships is motivated by neighbor-based algorithm that similar instances are more likely to share similar annotations.

3.2 Learning Model

To model the annotator-instance relationship and the similarity relationship, we introduce two latent variables that will be jointly updated during optimization. One is the difficulty denoted as $dft \in \mathbb{R}$, which is used to model the labeling difficulty: the more difficult in labeling an instance, the higher it is. The other one is the expertise vector denoted as $\boldsymbol{expt} \in \mathbb{R}^d$, which is designed to convey the annotator expertise toward one instance. We model the annotator's expertise as a vector instead of a scalar because the annotator might have varied level of expertise toward different instances. We then define the task as minimizing an objective function f (Eq. 1), which consists of 4 components corresponding to 4 hypotheses as will be described later.

$$f = \alpha \cdot h_1 + \beta \cdot h_2 + \gamma \cdot h_3 + \delta \cdot h_4, \tag{1}$$

where α, β, γ, δ are hyperparameters chosen based on cross validation on each dataset. To simplify the notation, we denote sigmoid function as $S(\cdot)$. Intuitively, we also use $1 - S(dft)$ instead of $S(dft)$ to represent how easy it is to label such instance in following hypotheses.

Hypothesis 1 (h_1): similar instances should share the same annotation, unless they are difficult to be classified

To model the similarity relationship between instances, we compute the similarity score $R_{i,j}$ by euclidean distance in feature space and map them into $[0, 1]$ by $e^{-|\mathbf{x}_i - \mathbf{x}_j|_2}$. The larger $R_{i,j}$ indicates i and j are more similar to each other. The prediction $\hat{y}$ is a real value, which is mapped into $[0, 1]$ through $S(\cdot)$. Naively, we set 0.5 as the threshold for label 0 and 1. With the introduction of $R_{i,j}$, $\hat{y}$ and dft, we can write down our first hypothesis as follows.

$$h_1(dft_i, \hat{y_i}) := \sum_{i,j} R_{i,j} \cdot (S(\hat{y_i}) - S(\hat{y_j}))^2 \cdot ((1 - S(dft_i)) + (1 - S(dft_j))) \tag{2}$$

The equation shows that for any given pairs of prediction outcomes, if they are similar (i.e. large $R_{i,j}$), then their prediction shall less likely to be different, unless they are considered as instances that are not easy to be classified (representing by small latent variables $1 - S(dft_i)$). In other words, the predicted label

of an instance j can be more easily propagated to another instance i if they are similar and assumed to be classified easier. There are actually multiple ways to represent the joint easiness measurement $1 - S(dft)$, while we find that simple summation is effective through the experiments.

Hypothesis 2 (h_2): the model shall trust labelers whose expertise matches the instance better

This hypothesis assumes the quality of annotation depends on how the expertise of annotators matches the instances to be announced. We assume a latent vector $\boldsymbol{expt}_u$ is used to represent the annotator's expertise, and its inner product with an instance shows the confidence of this annotator toward this specific instance. With these factors, we model the annotations as Eq. (3):

$$h_2(\boldsymbol{expt}, dft_i, \hat{y}_i) := \sum_{u,i} \left(S(\hat{y}_i) - y_i^{(u)} \right)^2 \cdot \left(S(\boldsymbol{expt}_u^\intercal \cdot \boldsymbol{x}_i) + (1 - S(dft_i)) \right). \quad (3)$$

For any given pair of annotated label $y_i^{(u)}$ and predicted label $\hat{y}_i$, the model will favor one with higher annotator's confidence and lower instance difficulty by minimizing Eq. (3). Our model leverages the annotator's confidence toward each instance, and downplays the ones without sufficient confidence during learning.

Hypothesis 3 (h_3): instances are generally not difficult to be classified

Model with only terms h_1 and h_2 have the tendency to maximizes the instance difficulty, which will inevitably reduce the amount of information that can be used to make the final prediction $\hat{y}_i$. Thus, we add summation of reciprocal of $1 - S(dft)$ as regularization term to our model that encourages our model to reduce its belief to the difficulty of each instance.

$$h_3(dft_i) := \sum_i (1 - S(dft_i))^{-1} \quad (4)$$

Hypothesis 4 (h_4): each annotator's expertise vector should be smooth

To avoid overfitting, we need to constraint the annotator's expertise vector $\boldsymbol{expt}_u$ as a regularization term.

$$h_4(\boldsymbol{expt}) := \sum_u \|\boldsymbol{expt}_u\|_2^2 \quad (5)$$

Put everything together. The objective function to be minimized looks like:

$$f(\mathbf{expt}_u, dft_i, \hat{y}_i) = \alpha \cdot \left(\sum_{i,j} R_{i,j} \cdot (S(\hat{y}_i) - S(\hat{y}_j))^2 \cdot ((1 - S(dft_i)) + (1 - S(dft_j))) \right)$$

$$+\beta \cdot \left(\sum_{u,i} \left(S(\hat{y}_i) - y_i^{(u)} \right)^2 \cdot (S(\boldsymbol{expt}_u^\intercal \cdot \boldsymbol{x}_i) + (1 - S(dft_i))) \right)$$
$$+\gamma \cdot \left(\sum_i (1 - S(dft_i))^{-1} \right) + \delta \cdot \left(\sum_u \|\boldsymbol{expt}_u\|_2^2 \right) \tag{6}$$

We can infer annotations for each instance by jointly update the latent parameters to minimize the object. We apply gradient descent to get the local optima of $\boldsymbol{expt}_u$, dft_i, and $\hat{y}_i$. The update formulas are listed as follow:

– Update formula for annotator expertise

$$\frac{\partial f}{\partial \boldsymbol{expt}_u} = S(\boldsymbol{expt}_u^\intercal \cdot \boldsymbol{x}_i) \cdot (1 - S(\boldsymbol{expt}_u^\intercal \cdot \boldsymbol{x}_i))$$
$$\cdot \beta \cdot \left(\sum_i \left(S(\hat{y}_i) - y_i^{(u)} \right)^2 \cdot \boldsymbol{x}_i \right) + 2 \cdot \delta \cdot \boldsymbol{expt}_u. \tag{7}$$

– Update formula for instance difficulty

$$\frac{\partial f}{\partial dft_i} = -S(dft_i) \cdot (1 - S(dft_i)) \cdot \left[\alpha \cdot \left(\sum_{i,j} R_{i,j} \cdot (S(\hat{y}_i) - S(\hat{y}_j))^2 \right) \right.$$
$$\left. + \beta \cdot \left(\sum_{u,i} \left(S(\hat{y}_i) - y_i^{(u)} \right)^2 \right) - \gamma \cdot (1 - S(dft_i))^{-2} \right]. \tag{8}$$

– Update formula for predicted label

$$\frac{\partial f}{\partial \hat{y}_i} = S(\hat{y}_i) \cdot (1 - S(\hat{y}_i))$$
$$\cdot \left[2 \cdot \alpha \cdot \left(\sum_{i,j} R_{i,j} \cdot (S(\hat{y}_i) - S(\hat{y}_j)) \cdot ((1 - S(dft_i)) + (1 - S(dft_j))) \right) \right.$$
$$\left. + 2 \cdot \beta \cdot \left(\sum_{u,i} \left(S(\hat{y}_i) - y_i^{(u)} \right) \cdot (S(\boldsymbol{expt}_u^\intercal \cdot \boldsymbol{x}_i) + (1 - S(dft_i))) \right) \right] \tag{9}$$

4 Experiment

We compare the derived algorithm to the state-of-the-art learning model proposed in [1]. There are two learning models in Yan's work: **M.L-Bernoulli** and **M.L-Gaussian**. Since the former has better performance than the latter according to the original paper, we compare our results to **M.L-Bernoulli** only.

Similar to [1], we also compare our model with multiple baseline algorithms. Beside individual annotator models, where each annotator learns a logistic

regression model disjointly from others, we also consider two majority-voting models learned with logistic regression — **L.R.-Majority** and **L.R.-Ensemble**. The former baseline, as in [1], takes the majority vote from all annotators as target labels while the later learns each annotator model separately in the first stage, and then combine learned models with a weighting matrix in the second stage.

We mainly perform our experiments on two kinds of dataset. One is simulated dataset that uses UCI datasets provided by [15], including UCI::Ionosphere, UCI::Cleveland, and UCI::Statlog. The experiment results are recorded in Sect. 4.1. The other one is a real dataset that uses Medical Text dataset cited in [16] with three different targets: Medical::Evidence, Medical::Focus and Medical::Polarity. The model performance is shown in Sect. 4.2. Finally, we demonstrate the contribution of each component of our model using UCI::Ionosphere in Sect. 4.3 and a summary of our experiment results in Sect. 4.4.

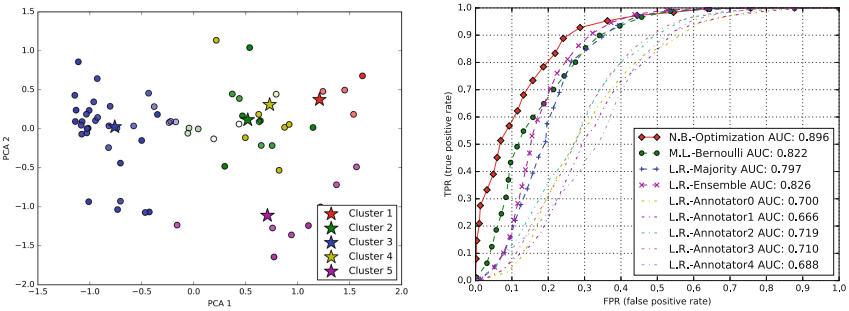
4.1 Simulated Datasets

For all UCI datsets, we follow similar setup in [1] with minor modifications to fit the scenario in real world better. The main procedure is summarized below.

1. Data preprocessing: including filling missing values, feature normalization and one-hot encoding.
2. Distribute instances into $K = 5$ clusters using K-Means algorithm. It tries to simmulate the instances into 5 different categories.
3. Assign $|U| = 5$ annotators to $K = 5$ clusters correspondingly. Each annotator is considered as an expert in its own cluster with higher labelling accuracy. Our simulation assumes the labelling accuracy of an expert to an instance is guilded by $0.6 + 0.4 \times e^{-\|x_i - c_k\|_2}$, where c_k is the center of cluster k. In other words, for one annotator, the labeling accuracy is closed to one in its own cluster, and can go down to 0.6 in the other clusters.

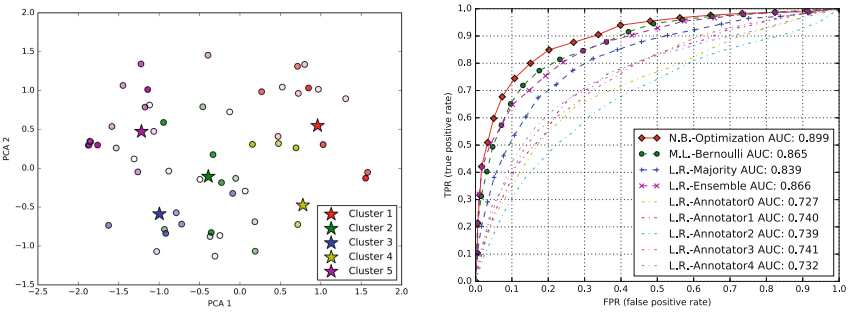
We consider fully-assigned annotations from each annotator to each instance for model learning and conduct experiments under 5-fold cross-validation. For evaluation, we use area-under-ROC curve (AUC) as the evaluation metric and report the average performance. We then repeat 5-fold experiments for $T = 30$ times to conduct *Wilcoxon Signed Rank Test* to examine whether the comparison is statistically significant under.

For all simulated experiments, we provide the ROC curve and calculate AUC as the performance metric, as shown in Figs. 2(b), 3(b) and 4(b). In addition, we provide an auxiliary cluster graph to show that our model can effectively locate difficult instances. To plot the cluster graph, we apply PCA to reduce high-dimension feature space to two-dimensional space then color each instance with its *dft* value: the lighter the color, the more difficult it is. Since each instance is annotated by five annotators during the experiment, they are represented as five centroids in K-Means. We would expect our model give higher difficulty value (i.e. lighter in color) to instances which are closer to boundary. The results are presented in Figs. 2(a), 3(a) and 4(a).



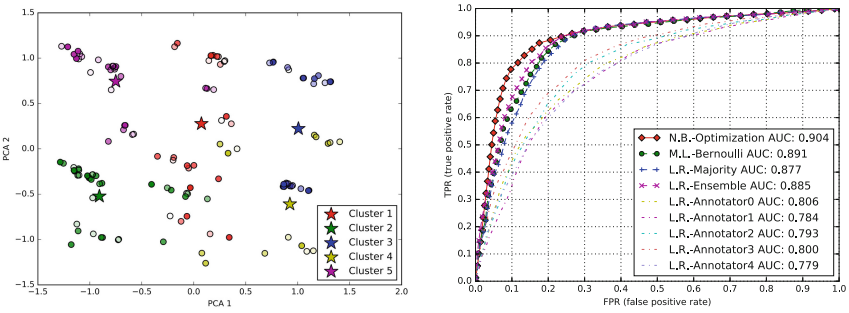
(a) Cluster Distribution (b) ROC Curve

Fig. 2. UCI:: Ionosphere dataset



(a) Cluster Distribution (b) ROC Curve

Fig. 3. UCI:: Cleveland dataset



(a) Cluster Distribution (b) ROC Curve

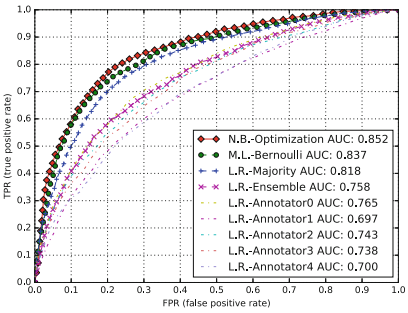
Fig. 4. UCI:: Statlog dataset

The results in Figs. 2(b), 3(b) and 4(b) show that our model (denoted as **N.B. optimization**) outperforms the state-of-the-art and other baselines. For the cluster distribution figures, we do find that the points near the cluster

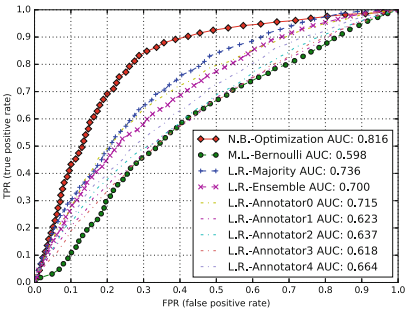
boundary are of lighter color, which means that the instance difficulty are correctly captured by our model. We would like to point out that being able to identify instances with higher difficulty is very important for tasks such as active learning, which implies our model as a suitable basis for active learning given multiple noisy annotators.

4.2 EvaluationMedical Text Dataset

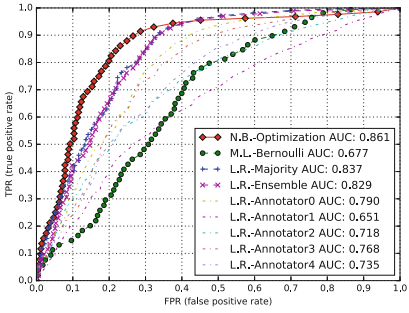
Medical text data is annotated by real annotators and was first used in [16]. In the collected corpus, there are total of 10000 sentences; whereas one sentence may be consisted of multiple text fragments. Annotation process runs in two rounds. In the first round of annotation, 3 annotators are randomly chosen from 8 annotator-pool to label 10000 sentences. Later in the second round of annotation, randomly selected 1000 sentences are labeled by other 5 annotators.



(a) Medical::Evidence ROC Curve



(b) Medical::Focus ROC Curve



(c) Medical::Polarity ROC Curve

Fig. 5. Medical text dataset

Since there may be multiple labels for one text segment, Medical Text labeling is actually a multilabel-multiclass problem. Based on [16], available labels for each text fragment include focus (G for generic, M for methodology, S for

Table 1. h_1, h_2, are two hypotheses on annotators and instances we mentioned in Sect. 3.2. h_3, h_4 are two regularization terms we used to prevent model from overfitting.

Hypothesis combination	Area Under Curve (AUC)
$h_1 + h_2 + h_3 + h_4$	0.896
$h_1 + h_2 + h_3$	0.895
$h_1 + h_2 + h_4$	0.824
$h_1 + h_2$	0.847

Table 2. Hypothesis tests between Yan's and our algorithm are examined on 3 simulated datasets and a real dataset with three targets through AUC evaluation metric. Experiments on simulated datasets are repeated 30 times with 5-fold cross-validation; while experiments on real dataset are repeated 5 times with 5-fold cross-validation. P-value with $*$ indicates siginificance.

	M.L.-Bernoulli	N.B.-Optimization	P-value
UCI:: Ionosphere	0.822 ± 0.033	$\mathbf{0.896 \pm 0.026}$	$< 0.00001^*$
UCI:: Cleveland	0.865 ± 0.027	$\mathbf{0.899 \pm 0.010}$	$< 0.00001^*$
UCI:: Statlog	0.891 ± 0.014	$\mathbf{0.901 \pm 0.009}$	0.000034^*
Medical:: Evidence	0.837 ± 0.005	$\mathbf{0.852 \pm 0.004}$	0.004065^*
Medical:: Focus	0.598 ± 0.029	$\mathbf{0.816 \pm 0.009}$	0.000034^*
Medical:: Polarity	0.677 ± 0.006	$\mathbf{0.861 \pm 0.008}$	$< 0.00001^*$

science), evidence (E0 means no evidence, E3 means direct evidence, while E1, E2 are in between), and polarity (P for positive, N for negative, ranging from N3 to N0, then from P0 to P3). In our experiment, we regard Medical::Evidence, Medical::Focus, and Medical::Polarity as different tasks, and transform each of them into a binary classification task. For Medical::Focus and Medical:Evidence, we follow the binarization process in [7]. For Medical:Polarity, the annotation contains N3, N2, N1, N0, P0, P1, P2, P3. We treat P0, P1, P2, P3 as 1 while others are 0. The results are presented in Fig. 5(a) through Fig. 5(c).

1. Select 1000 sentences that have been labeled from all 8 annotators.
2. Remove 309 sentences that are segmented differently by various annotators. 691 sentences remain.
3. Partition 691 sentences into 874 text fragments.
4. Apply stopword removal and rare term removal to get 848 text segments and 279 words as column features.
5. Calculate TF-IDF scores for each word in 279 column features as instance features and transform origin multi-class ground truths into binary ones according to different task targets.
6. Repeat 5-fold experiments for $T = 5$ times, and then put *Paired T-test* on averaged 5-fold results to judge the significance of performance improvement.

4.3 Component Importance of N.B.-Optimization

To examine which component has the most influence on the prediction quality, we use UCI::Ionosphere as the experimental dataset to evaluate some combinations of parameters, i.e. α, β, γ, and δ, and set some of them to 0. From Table 1, it is clear that h_1 and h_2 are both the crucial components to the model and the model become more robust with h_3. h_4 provides only marginal boost on the performance.

4.4 Experiment Summary

From the above experiment results, we can tell that our model is good at identifying instance difficulty and has great performance in both simulated and real dataset with four robust hypotheses.

5 Conclusion

Unlike the existence of ground truths in traditional supervised learning problems, perfect labels in crowdsourcing scenario are not guaranteed, as the labelers may equip varied levels of expertise toward different scope of knowledge. Thus, a model such as ours that can utilize information from highly-diversified and noisy data sources is highly demanded.

Acknowledgement. This material is based upon work supported by the Air Force Office of Scientific Research, Asian Office of Aerospace Research and Development (AOARD) under award number FA2386-15-1-4013, and Taiwan Ministry of Science and Technology (MOST) under grant number 105-2221-E-002-064-MY3.

References

1. Yan, Y., Rosales, R., Fung, G., Schmidt, M.W., Valadez, G.H., Bogoni, L., Moy, L., Dy, J.G.: Modeling annotator expertise: learning when everybody knows a bit of something. In: AISTATS, pp. 932–939 (2010)
2. Raykar, V.C., Yu, S., Zhao, L.H., Jerebko, A., Florin, C., Valadez, G.H., Bogoni, L., Moy, L.: Supervised learning from multiple experts: whom to trust when everyone lies a bit. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 889–896. ACM (2009)
3. Zhang, P., Obradovic, Z.: Learning from inconsistent and unreliable annotators by a Gaussian mixture model and Bayesian information criterion. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011. LNCS (LNAI), vol. 6913, pp. 553–568. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23808-6_36
4. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. J. Mach. Learn. Res. **11**(Apr), 1297–1322 (2010)
5. Raykar, V.C., Yu, S.: Eliminating spammers and ranking annotators for crowdsourced labeling tasks. J. Mach. Learn. Res. **13**(Feb), 491–518 (2012)

6. Zhang, P., Obradovic, Z.: Integration of multiple annotators by aggregating experts and filtering novices. In: 2012 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1–6. IEEE (2012)
7. Zhang, P., Cao, W., Obradovic, Z.: Learning by aggregating experts and filtering novices: a solution to crowdsourcing problems in bioinformatics. BMC Bioinform. **14**(Suppl 12), S5 (2013)
8. Yan, Y., Fung, G.M., Rosales, R., Dy, J.G.: Active learning from crowds. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 1161–1168 (2011)
9. Yan, Y., Rosales, R., Fung, G., Dy, J.: Modeling multiple annotator expertise in the semi-supervised learning scenario. arXiv preprint arXiv:1203.3529 (2012)
10. Yan, Y., Rosales, R., Fung, G., Farooq, F., Rao, B., Dy, J.G., Malvern, P.: Active learning from multiple knowledge sources. In: AISTATS, vol. 2, p. 6 (2012)
11. Yan, Y., Rosales, R., Fung, G., Dy, J.: Active learning from uncertain crowd annotations. In: 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 385–392. IEEE (2014)
12. Yan, Y., Rosales, R., Fung, G., Subramanian, R., Dy, J.: Learning from multiple annotators with varying expertise. Mach. Learn. **95**(3), 291–327 (2014)
13. Long, C., Hua, G.: Multi-class multi-annotator active learning with robust gaussian process for visual recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2839–2847 (2015)
14. Rodrigues, F., Pereira, F., Ribeiro, B.: Learning from multiple annotators: distinguishing good from random labelers. Pattern Recogn. Lett. **34**(12), 1428–1436 (2013)
15. Lichman, M.: UCI machine learning repository (2013)
16. Rzhetsky, A., Shatkay, H., Wilbur, W.J.: How to get the most out of your curation effort. PLoS Comput. Biol. **5**(5), e1000391 (2009)