ThePlantGame: Actively Training Human Annotators for Domain-specific Crowdsourcing

Maximilien Servajean, Alexis Joly, Dennis Shasha, Julien Champ, Esther Pacitti

ThePlantGame: Actively Training Human Annotators for Domain-specific Crowdsourcing

Maximilien Servajean
INRIA Zenith team
LIRMM
servajean@lirmm.fr

Alexis Joly
INRIA Zenith team
LIRMM
alexis.joly@inria.fr

Dennis Shasha
NYU
shasha@courant.nyu.edu

Julien Champ
INRIA Zenith team
LIRMM
champ@lirmm.fr

Esther Pacitti
INRIA Zenith team
LIRMM
pacitti@lirmm.fr

ABSTRACT

In a typical citizen science/crowdsourcing environment, the contributors label items. When there are few labels, it is straightforward to train contributors and judge the quality of their labels by giving a few examples with known answers. Neither is true when there are thousands of domain-specific labels and annotators with heterogeneous skills. This demo paper presents an *Active User Training* framework implemented as a serious game called ThePlantGame. It is based on a set of data-driven algorithms allowing to (i) actively train annotators, and (ii) evaluate the quality of contributors' answers on new test items to optimize predictions.

1. ACTIVE USER TRAINING

Classical crowdsourcing algorithms for multi-label classification tasks [1, 3, 5, 6] are typically based on the Bayesian inference of the most probable labels according to the confusion matrix of each worker. Applying such approaches in the context of classification tasks with very large number of classes and expert knowledge is however challenging in two principle ways. First, the very high number of classes, *e.g.* thousands of plant species, makes it impossible to train a complete confusion matrix for each participant as it would require them to answer a huge number of queries (typically quadratic in the number of classes). Furthermore, the brute-force approach consisting of a quiz across the full list of classes is not tractable for non-specialist contributors. To address these issues, we propose an *Active User Training* framework where the training hypothesis space is adaptively and dynamically chosen for each user and observation, through machine learning techniques. Figure 1 describes the framework architecture. It is composed of five modules:

1. Automatic annotation: when a new data item i is added to the system, a machine learning model (*i.e.* a convolutional neural network in our experiment) predicts the probability of its true label t_i. Once the probability $p(t_i = j)$ of one class overcomes a system defined threshold (*i.e.* 99%), the item is automatically tagged as validated. Otherwise, it is processed by steps 2, 3 and 4 to reduce its uncertainty.

2. Active Training: to actively train the annotators, the system automatically creates quizzes of size $k << n$ classes in which the probability of appearance of a class j is proportional to its likelihood $p(t_i = j)$ given an item i. Thus, the annotators actually learn how to disambiguate the classes that are the most confused in the unvalidated data. Using such Monte-Carlo sampling rather than a hard selection of the top-k classes allows to train more complementary the annotators and avoids sticking on the current bias of the system (*e.g.* the true class might not be in the top-k).

3. Skills-aware Assignment: to classify as many data as possible based on the current known confusion of each annotators, the system assigns unvalidated items to users – up to a system defined threshold – that are very likely – given a threshold – to be able to disambiguate them. This is done by solving the following optimization problem (using *e.g.* Cplex solver):

$$\begin{aligned} \underset{x_{ik}}{\text{maximize}} \quad & \sum_{i=0}^{N}\sum_{k=0}^{K} x_{ik}p_{ik} \\ \text{subject to} \quad & \sum_{i=0}^{N} x_{ik} < n_k \ \forall k, \sum_{k=0}^{K} x_{ik} < m_i \ \forall i \\ & x_{ik} \in \{0,1\} \ \forall i,k. \end{aligned}$$

where $x_{ik} \in \{0,1\}$ indicates if item i has been assigned to user k or not, p_{ik} is the likelihood of user k to give a correct classification proposition for item i (according to her/his confusion matrix as discussed in step 4). x_{ik} is constrained by n_k and m_i to limit the number of assignments per user and per object. Solving this optimization problem will result in maximizing the number of assignment that will receive a correct classification proposition.

4. Classification inference: Once classification propositions have been given by the annotators to a set of items, an inference model will update their probability distributions. This model initializes the confusion of each user based on their performance during the training and the probabilities of each class based on the validated data. The model then infer the actual confusion of each worker using a Bayesian network. It has the following joint probability distribution:

$$\begin{aligned} p(\kappa,\Pi,t,c,\theta|A,B,\nu) = & \prod_{i=1}^{N}\{\kappa_{t_i}\prod_{k=1}^{K}\pi^{(k)}_{t_i,c_i^{(k)}}\}p(\kappa|\nu) \\ & p(\Pi|\theta)p(\theta|A,B) \end{aligned} \quad (1)$$

MM '16 October 15-19, 2016, Amsterdam, Netherlands

ACM ISBN 978-1-4503-3603-1/16/10.

DOI: http://dx.doi.org/10.1145/2964284.2973820

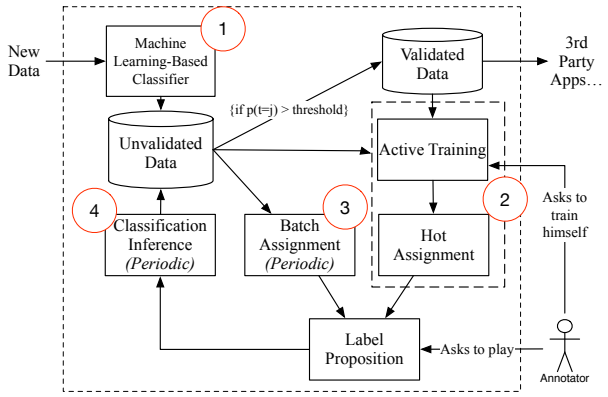

Figure 1: Framework Architecture.

Where κ is the global distribution of each class that follows a Dirichlet distribution of parameters ν, Π represents all confusion matrices where $\pi_{jl}^{(k)}$ is the probability that the user k would answer class l if the true class was j. Each row of a confusion matrix has a Dirichlet distribution of parameters $\theta_j^{(k)}$. t is the set of true labels of all items following a multinomial distribution of parameters κ. Finally, all θ_{jl} follow independent gamma distributions of parameter α_{jl} and β_{jl}. More details about the model are available in the following paper [4].

2. THE PLANT GAME

The proposed framework was implemented within a real playful platform (*The Plant Game*[1]) focused on the specific and complex problem of plants classification. We used the dataset released for the plant task of the LifeCLEF 2015 challenge [2]. Training images were used for the automatic creation of the quizzes (*i.e.* for the active training of the annotators) whereas test images were used as the unannotated data to be classified. To play, each user has to create an account and to self-evaluate his current expertise (*i.e.* beginner, intermediate, expert or expert+). The Plant Game then offers three game modes: (i) **Training mode**, *i.e.* the active training approach, (ii) **The Plant Game mode** where the player can annotate the unvalidated observations assigned to him, and (iii) the **Duel mode** in which two players can challenge each others based on randomly assigned items. After an item has become validated (thanks to one or several inference rounds), the player can see can see if she/he was wrong or not in The Plant Game mode view. Additionally, all users can check their ranking to see how well they perform.

3. EXPERIMENTS

Figure 2 shows a global overview of the benefits of our contributions. First, it is clear that complex classification tasks are impossible without training. Second, training only is not sufficient. The annotators can actually not learn all classes. Combining training and a correct assignment strategy is necessary to obtain the best results. Also, a correct aggregation method based on a Bayesian network enables even better Results.

Figure 3 shows that our contributions enable a clear gain in terms of data quality compared to a fully automated solution. For instance, we obtain a precision of 0.98 while the *CNN* only obtains 0.85.

[1]http://theplantgame.com

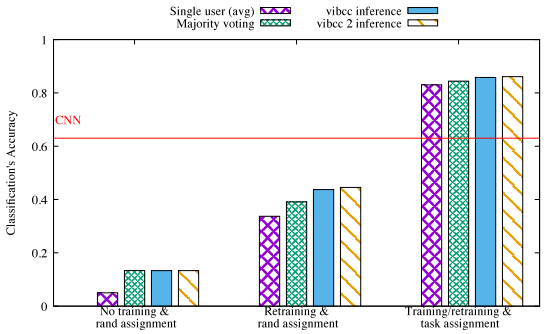

Figure 2: Classification quality with several assignment and training approaches combination.

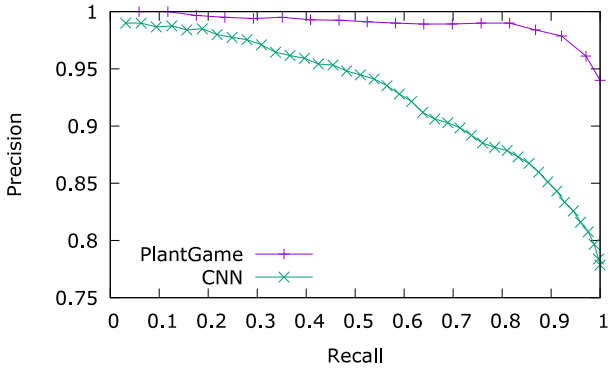

Figure 3: Recall precision curves (CNN alone vs. ThePlantGame)

4. REFERENCES

[1] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[2] H. Goëau, A. Joly, and P. Bonnet. Lifeclef plant identification task 2015. In *CLEF 2015*, 2015.

[3] H.-C. Kim and Z. Ghahramani. Bayesian classifier combination. In *International conference on artificial intelligence and statistics*, pages 619–627, 2012.

[4] M. Servajean, A. Joly, D. Shasha, J. Champ, and E. Pacitti. Thousands-of-labels crowdsourcing: an active bayesian approach. *IEEE Trans. on Multimedia (submitted)*, 2016.

[5] E. Simpson, S. Roberts, I. Psorakis, and A. Smith. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision Making and Imperfection*, pages 1–35. Springer, 2013.

[6] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proc. of WWW conference*, pages 155–164. ACM, 2014.