Vote Aggregation Techniques in the Geo-Wiki Crowdsourcing Game: A Case Study

Artem Baklanov[1,2,3], Steffen Fritz[1], Michael Khachay[2,3], Oleg Nurmukhametov[2], Carl Salk[1], Linda See[1], and Dmitry Shchepashchenko[1]

[1] International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria, {baklanov, fritz, salk, see, schepd}@iiasa.ac.at,
[2] Krasovsky Institute of Mathematics and Mechanics, Ekaterinburg, Russia mkhachay@imm.uran.ru, oleg.nurmuhametov@gmail.com
[3] Ural Federal University, Ekaterinburg, Russia

Abstract. The Cropland Capture game (CCG) aims to map cultivated lands using around 170000 satellite images. The contribution of the paper is threefold: (a) we improve the quality of the CCG's dataset, (b) we benchmark state-of-the-art algorithms designed for an aggregation of votes in a crowdsourcing-like setting and compare the results with machine learning algorithms, (c) we propose an explanation for surprisingly similar accuracy of all examined algorithms. To accomplish (a), we detect image duplicates using the perceptual hash function pHash. In addition, using a blur detection algorithm, we filter out unidentifiable images. In part (c), we suggest that if all workers are accurate, the task assignment in the dataset is highly irregular, then state-of-the-art algorithms perform on a par with Majority Voting. We increase the estimated consistency with expert opinions from 77% to 91% and up to 96% if we restrict our attention to images with more than 9 votes.

Keywords: crowdsourcing, image processing, votes aggregation

1 Introduction

Crowdsourcing is a new approach for solving data processing problems for which conventional methods appear to be inaccurate, expensive, or time-consuming. Nowadays, the development of new crowdsourcing techniques is mostly motivated by so called Big Data problems, including problems of assessment and clustering of large datasets obtained in aerospace imaging, remote sensing, and even in social network analysis. For example, by involving volunteers from all over the world, the Geo-Wiki project tackles the problems of environmental monitoring with applications to flood resilience, biomass data analysis and forecasting, etc. The Cropland Capture game, which is a recently developed Geo-Wiki game, aims to map cultivated lands using around 170000 satellite images from the Earth's surface. Despite recent progress in image analysis, the solution to these problems is hard to automate since human-experts still outperform the majority of learnable machines and other artificial systems in this field. Replacement

of rare and expensive experts by a team of distributed volunteers seems to be promising, but this approach leads to challenging questions: how can we aggregate individual opinions optimally, obtain confidence bounds, and deal with the unreliability of volunteers?

The main goals of the Geo-Wiki project are collecting land cover data and creating hybrid maps [15]. For example, users answer 'Yes' or 'No' to the question: 'Is there any cropland in the red box?' in order to validate the presence or absence of cropland [14]. In the paper [2], which is related to use of Geo-Wiki data, researchers studied the problem of using crowdsourcing instead of experts. The research showed that it is possible to use crowdsourcing as a tool for collecting data, but it is necessary to investigate issues such as how to estimate reliability and confidence.

This paper presents a case study that aims to compare the performance of several state-of-the-art vote aggregation techniques specifically developed for the analysis of crowdsourcing campaigns using the image dataset obtained from the Cropland Capture game. As a baseline, some classic machine learning algorithms such as Random Forest, AdaBoost, etc., augmented with preliminary feature selection and a preprocessing stage, are used.

The rest of the paper is structured as follows. In Section 2, we give a brief overview of the vote aggregation algorithms involved in our case study. In Section 3, we describe the general structure of the dataset under consideration. In Section 4, we propose quality improvements for the initial image dataset and introduce our vote aggregation heuristic. Finally, in Section 5, we present our benchmarking results.

2 Related work

In the theoretical justification of crowdsourcing image-assessment campaigns, there are two main problems of interest. The first one is the problem of ground truth estimation from crowd opinion. The second one, which is equally important, deals with the individual performance assessment of the volunteers who participated in the campaign. The solution to this problem is in the clustering of voters with respect to their behavioural strategies into groups of *honest workers*, *biased annotators*, *spammers*, *malicious users*, etc. Note that a different approach is proposed in paper [1] that uses the biclustering to group the annotators based on their attempted questions.

Reflection of this posterior knowledge by reweighing of individual opinions of the voters can substantially improve the overall performance of the aggregated decision rule.

There are two basic settings of the latter problem. In the first setup, a crowdsourcing campaign admits some quantity of images previously labeled by experts (these labels are called *golden standard*). In this case, the problem can be considered as a supervised learning problem, and for its solution, conventional algorithms of ensemble learning (for example, boosting [11, 20, 7]) can be used. On the other hand, in most cases, researchers deal with the full (or almost full)

absence of labeled images; ground truth should be retrieved simultaneously with estimation of voters' reliability, and some kind of unsupervised learning techniques should be developed to solve the problem.

Prior works in this field can be broadly classified in two categories: EM-algorithm inspired and graph-theory based. The works of the first kind extend results of the seminal paper [3], applying a variant of the well known EM-algorithm [4] to a crowdsourcing-like setting of the computer-aided diagnosis problem. For instance, in [13], the EM-based framework is provided for several types of unsupervised crowdsourcing settings (for categorical, ordinal and even real answers) taking into account different competency level of voters and different levels of difficulty in the assessment tasks. In [12], by proposing a special type of prior, this approach is extended to the case when most voters are *spammers*. Papers [8, 17, 10] develop the fully unsupervised framework based on Independent Bayesian Combination of Classifiers (IBCC), Chinese Restaurant Process (CRP) prior, and Gibbs sampling. Although EM-based techniques perform well in many cases, usually, they are criticized for their heuristic nature since in general there are no guarantees that the algorithm finds a global optimum.

Another approach applied to reliability of the voters is based on recent results obtained for random regular bipartite graphs. Karger et al. [6] obtained both an asymptotically optimal graph construction and an asymptotically optimal iterative inference algorithm on this graph. These results are extended in [9] by applying approximate variational methods including belief propagation and mean field.

Furthermore, in [5], an efficient reputation algorithm for identifying adversarial workers in crowdsourcing campaigns is elaborated. For some conditions, the reputation scores proposed are proportional to the reliabilities of the voters given that their number tends to infinity. Unlike the majority of EM-based techniques, the listed results have solid theoretical support, but conditions for which their optimality is proven (especially the graph-regularity condition) are too restrictive to apply them straightforward in our setup.

The aforementioned arguments have motivated us to carry out a case study on the applicability of several state-of-the-art vote aggregation techniques to an actual dataset obtained from the Cropland Capture game. Precisely, we compare the classic EM algorithm, methods proposed in [5], [6], and a heuristic based on the computed reliability of voters. As a baseline, we use the simple Majority Voting (MV) heuristic and several of the most popular universal machine learning techniques.

3 Dataset

We carry out a benchmark of state-of-the-art vote aggregation techniques using the actual dataset obtained from the Cropland Capture game. The results of the game were captured as shown in two tables. The first table contains details of the images: *imgID* is an image identifier; *link* is the URL of an image; *latitude* and *longitude* are geo-coordinates which refer to the centroid of the image; *zoom*

is the resolution of an image (values: 300, 500, 1000 m). The following table shows some sample of image data.

imgID	*link*	*latitude*	*longitude*	*zoom*
3009	http://cg.tuwien.ac.at/~sturn/crop/img_-112.313_42.8792_1000.jpg	42.8792	-112.313	1000
3010	http://cg.tuwien.ac.at/~sturn/crop/img_-112.313_42.8792_500.jpg	42.8792	-112.313	500
3011	http://cg.tuwien.ac.at/~sturn/crop/img_-112.313_42.8792_300.jpg	42.8792	-112.313	300

All votes, i.e. 'a single decision by a single volunteer about a single image' [14], were collected in the second table: *ratingID* is a rating identifier; *imgID* is an image identifier; *volunteerID* is a volunteer's identifier; *timestamp* is the time when a vote was given; *rating* is a volunteer's answer. The possible values for *rating* are as follows: 0 ('Maybe'), 1 ('Yes'), -1 ('No'). The following table shows some sample of vote data.

ratingID	*imgID*	*volunteerID*	*timestamp*	*rating*
75811	3009	178	2013-11-18 12:50:31	1
566299	3009	689	2013-12-03 08:10:38	0
641369	3009	1398	2013-12-03 17:10:39	-1
3980868	3009	1365	2014-04-10 16:52:07	1

4 Methodology

4.1 Detection of duplicates and blurry images

Since the dataset collected via the game was formed by combining different sources, it is possible that almost the same images can be referenced by different records. In order to check this, we download all 170041 .jpeg images (512*512 size). The total size of all images is around 9 Gb. Then we employ perceptive hash functions to reveal such cases. Examples of such functions are aHash (Average Hash or Mean Hash), dHash, and pHash [19]. Perceptual hashing aims to detect images such that a human cannot see the difference. We find that pHash performs much better than computationally less expensive dHash and aHash methods. Note that for a fixed image, the set of all images that is similar according to pHash will contain all images with the corresponding MD5 or SHA1 hash. To summarize, we detect duplicates for 8300 original images; votes for duplicates were merged.

Accepting the idea of the wisdom of the crowd, in order to make a better decision for an image, we need to collect more votes for each image. The detection of all similar images increases statistically significant effects and decreases the dimensionality of the data. In addition, if the detection is performed before the start of the campaign, there is a reduction in the workload of the volunteers.

A visual inspection of images shows the presence of illegible and blurry (unfocused) images. As expected, these images bewildered the volunteers. Thus, we apply automatic methods for blur detection. Namely, by using the Blur Detection algorithm [18], we detect 2300 poor quality images such that it is not possible to give the right answers even for experts. Note that for those images, voting inconsistency is high; volunteers and experts change their opinions frequently. After consultation with the experts, we remove all images of poor quality. Note that the image processing steps turn out to be crucial for decreasing the noise

level and uncertainty in the dataset. Unfortunately, since the testing dataset is obtained after image processing, it is impossible to estimate direct impact of these steps on the accuracy of aggregated votes.

4.2 Majority voting based on reliability

In this subsection we present a conjunction of majority voting and the widely used notion of reliability (see, for example, [5]). It is a standard to define reliability w_i of worker i as

$$w_i = 2p_i - 1$$

where p_i is the probability that worker i gives a correct answer (it is assumed that it does not depend on the particular task); obviously, $w_i \in [-1, 1]$. We use traditional weighted MV with weights obtained by the above rule. The heuristic admits a refinement; one may iteratively remove a volunteer with the highest penalty, then recalculate penalties, and obtain new results for the weighted MV.

The proposed heuristic is presented in Algorithm 1. Note that mapping $\mathbf{I} : \{False; True\} \to \{0; 1\}$ is defined by the rule: $\mathbf{I}(True) = 1, \mathbf{I}(False) = 0$.

Algorithm 1 Weighted MV

Input: V is the set of all volunteers;
I is the set of all images with at least 1 vote;
$R = \left(r_{v,i}\right)_{v=1,i=1}^{|V|,|I|}$ is the rating matrix (see (2));
E is the set of images with ground truth labels;
$(e_i)_{i \in E} \in \{-1; 1\}^{|E|}$ are ground truth labels for images from E.
Output: the predicted labels $\{y_1, y_2, ..., y_{|I|}\}$
Initialization:

for $v \in V$: **do**
  if $\sum_{i \in I \cap E} \mathbf{I}(r_{v,i} \neq 0) \neq 0$ **then**
    $w_v \leftarrow 2 \times \frac{\sum_{i \in I \cap E} \mathbf{I}(r_{v,i} = e_i)}{\sum_{i \in I \cap E} \mathbf{I}(r_{v,i} \neq 0)} - 1$
  else
    $w_v \leftarrow 0$

Repeat
Calculate penalties for volunteers according to Algorithm 2 [5]. The algorithm takes I, V, R as inputs and gives a vector $(p_v)_{v \in V} \in [0, 1]^{|V|}$ as output. For volunteer $\hat{v}$ with the highest penalty, we set

$$w_{\hat{v}} \leftarrow 0,$$
$$r_{\hat{v},i} \leftarrow 0 \quad \forall i \in I.$$

Until reaching a pre-specified number of iterations
Output: the predictions $(y_i)_{i \in I}$

$$y_i = argmax_{k \in \{-1;1\}} \sum_{v \in V} w_v \mathbf{I}(r_{v,i} = k). \tag{1}$$

---

5 Experiments

During the crowdsourcing campaign, around 4.6 million votes were collected. The voting protocol was converted to a rating matrix. The matrix consists of ratings given to images (matrix columns) by the volunteers (matrix rows)

$$R = \left(r_{v,i}\right)_{v=1,i=1}^{|V|,|I|}, \tag{2}$$

V is the set of all volunteers ($|V|$=2783);
I is the set of all images with at least 1 vote ($|I|$=161752);
$r_{v,i}$ is a vote given by a volunteer to an image.

Due to an unclear definition, the *'Maybe'* answer is hard to interpret. As a result, we treat *'Maybe'* as a situation when the user has not seen the image; both situations are coded as 0. If a volunteer has multiple votes for the same image, then *only the last vote is used*.

To evaluate the volunteers' performance, a part of the dataset (854 images) was annotated by an expert after the campaign took place. For these images 1813 volunteers gave 16,940 votes in total. Then we sampled two subsets for training and testing (70/30 ratio).

The baseline. We treat columns of the rating matrix as feature vectors of images. To use some conventional machine learning algorithms, *we first apply SVD to the whole dataset* to reduce dimensionality. A study of the explained variance helps us to make an appropriate choice for the number of features: 5, 14, 35. Then we transform the feature space of the testing and training subsets accordingly. On the basis of 10-fold cross-validation of the training subset, we fit parameters for the AdaBoost and Random Forest algorithms. For Linear Discriminant Analysis (LDA), we use default parameters. The accuracy of the algorithms with fitted parameters was estimated using the testing subset; see *Table 1.*

Table 1: Baseline algorithms

Number of features	*Random Forest*	*LDA*	*AdaBoost*
5	**89.92**	87.60	89.15
14	89.14	**90.70**	89.92
35	88.37	89.53	**91.08**

Table 2: Accuracy for 'crowdsourcing' algorithms without image-vote thresholding

iteration	*MV*	*EM*	*KOS*	*KOS+*	*weighted MV*
Base	89.81	89.81	88.99	89.81	90.63
1	90.05	90.16	88.88	90.16	**91.45**
2	90.05	90.05	88.64	90.16	**91.45**
3	89.67	89.58	88.17	89.70	91.22
4	89.34	89.46	88.17	89.22	90.98
5	89.93	89.81	88.41	89.58	91.10
6	89.81	89.93	88.52	89.58	90.98
7	90.16	90.05	88.64	89.46	90.98
8	90.16	89.93	88.88	89.58	90.87
9	90.16	89.81	89.11	89.70	90.75

Table 3: Accuracy for 'crowdsourcing' algorithms with image-vote thresholding. Only images with at least 4 votes are left in the expert dataset. In this case we have 729 images annotated by 1812 volunteers.

iteration	*MV*	*EM*	*KOS*	*KOS+*	*weighted MV*
Base	90.95	91.08	90.12	91.08	91.63
1	91.08	91.36	90.26	91.36	92.18
2	91.08	91.36	90.12	91.36	92.18
3	91.63	91.36	90.26	91.36	**92.32**
4	91.22	91.08	89.71	91.08	91.77
5	91.22	91.22	89.71	91.22	92.04
6	91.08	91.36	90.26	91.36	91.91
7	91.08	91.36	90.40	91.36	91.91
8	91.08	91.08	90.53	90.81	91.91
9	90.81	91.08	90.40	90.81	91.91

Table 4: Accuracy for 'crowdsourcing' algorithms with image-vote thresholding. Only images with at least 10 votes are left in the expert dataset. In this case we have 404 images annotated by 1777 volunteers.

iteration	*MV*	*EM*	*KOS*	*KOS+*	*weighted MV*
Base	94.55	94.55	94.06	94.55	95.05
1	94.55	94.55	93.81	94.55	95.05
2	94.55	94.55	93.81	94.55	95.05
3	94.55	94.55	94.06	94.55	95.05
4	94.55	94.55	94.06	94.55	95.05
5	94.55	94.55	94.06	94.55	95.05
6	94.55	94.80	94.06	94.55	95.30
7	94.55	94.80	94.06	94.80	95.30
8	94.55	94.80	94.06	94.80	95.30
9	94.80	94.80	94.06	95.05	**95.54**

Benchmarking of algorithms for an aggregation of crowd votes is performed as follows. We feed the expert dataset to the algorithms and check their accuracy on the same test subset as above. Note that the transformation of a feature space is not required in this case. In this section, we experimentally test the heuristic based on reliability and compare it with the state-of-art algorithms designed for crowdsourcing. We use publicly available code[4] that was developed for experiments in [5]. The code implements the iterative algorithm in [6] referred to as the KOS and EM algorithms [3]; both are implemented in conjunction with reputation algorithm 2 in [5] (also called Hard penalty). Note that KOS+ is a normalized version (see [5]) of KOS. This version may be more suitable for arbitrary graphs (KOS is developed for regular graphs). During each iteration, the reputation algorithm helps to exclude the volunteer with the highest penalty and recalculates the penalties for the remaining volunteers. The accuracy of the compared algorithms on the test sample is presented in Table 2. Note that the first row (*Base*) corresponds to results before the exclusion of volunteers.Surprisingly, all crowdsourcing algorithms perform on par with Majority voting. A possible explanation is the irregular task assignment leading, in particular, to a high percentage of images with only a few votes. To deal with this issue, we continue our analysis using *image thresholding by the number of votes received* (or simply image-vote thresholding). Namely, we perform the same benchmarking for two subsets of the expert dataset. The subsets were obtained by filtering images with the number of votes less than the threshold; see Table 3 and 4. Note that the training and the testing sets are different in the experiments reflected in Tables 2, 3, and 4.

Another possible explanation is that we mostly deal with reliable volunteers, and thus, crowdsourcing algorithms cannot profit from the detection of spammers or from flipping votes of malicious voters. To analyze this hypothesis, we classify volunteers according to their performance. In this regard, we use notation introduced in [12]. Namely, as it was suggested, in Fig. 1, we depict the

[4] `https://github.com/ashwin90/Penalty-based-clustering`

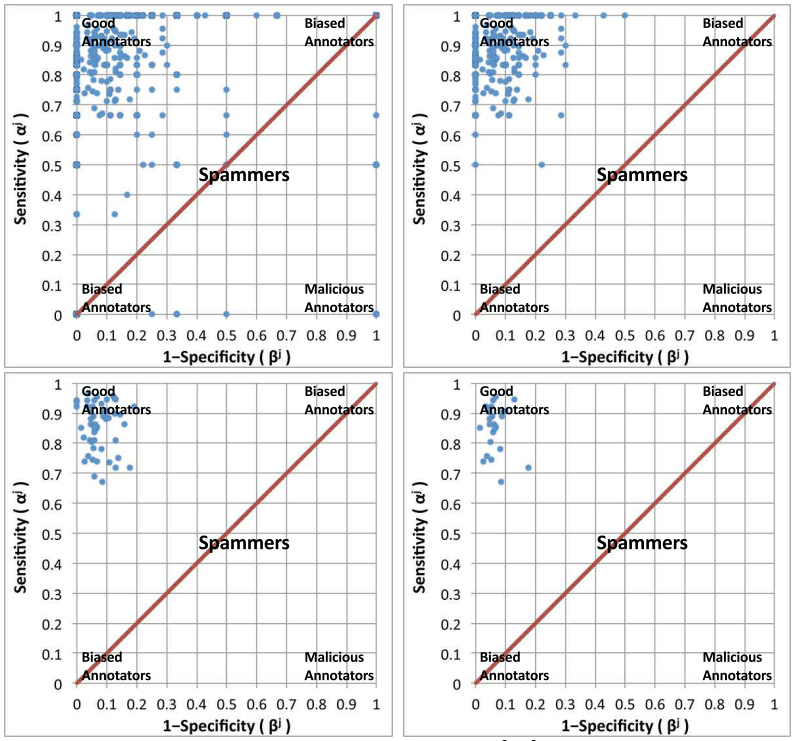

Fig. 1: In the figure we use notation introduced in [12]. Threshold = 0, 12, 44 , and 100 votes. These thresholds leave 1813, 262, 52, and 24 volunteers, respectively. ROCs of spammers lie on the red line.

Receiver Operating Characteristic (ROC) plot containing details of individual performance. Each plot in Fig. 1 depicts two values for each volunteer: *the sensitivity* and *the specificity*. If the true label is 1, then the sensitivity is defined as the probability that the volunteer votes 1 (this probability corresponds to the true positive rate). If the true label is -1, then the specificity is defined as the probability that the volunteer votes -1. Since the task assignment was highly irregular, it is important to study how voting activity of volunteers influences the ROC. Namely, Fig. 1 contains not one, but four ROCs, where each of them is obtained according to a different level of volunteer thresholding. This thresholding helps to remove volunteers that had a total number of votes less than that defined by the threshold. Note that the definition of spammer introduced in [12] may differ from an intuitive one. Namely, spammer is a volunteer voting randomly and independently of true classes of images. Fig. 1 provides plausible observations: there are no spammers among voters with more than 12 votes; good annotators prevail over all other types of annotators; there are frequently voting volunteers (more than 100 votes) showing better accuracy than any examined algorithm. These are the reasons why algorithms detecting spammers do not outperform the baseline noticeably.

6 Conclusions

Comparing the results in Table 1 and Table 2, it is remarkable that 'general purpose' learning algorithms slightly outperform 'special purpose' crowdsourcing algorithms. Surprisingly, the proposed simple heuristic (see Algorithm 1) based on reliability shows the best result. Also, numerical experiments show that Majority Voting performs on par with all other algorithms. The analysis of the ROCs of the volunteers suggests that surprisingly high accuracy of frequently voting volunteers coupled with the absence of spammers is a possible explanation for this result. The highly irregular task assignment in the dataset with a high percentage of images with a low number of votes may also contribute to this fact. Note that image-vote thresholding helps to improve the results of the 'crowdsourcing' algorithms (see Tables 2, 3, 4) although the results are still on a par with Majority Voting. This parity differs from an observation obtained in comprehensive benchmark [16] where *'MV was often outperformed by some other method.'*

In the future we plan to benchmark the remaining state-of-the-art methods for the aggregation of votes and include 'Maybe' votes into consideration.

Acknowledgments. This research was supported by Russian Science Foundation, grant no. 14-11-00109, and the EU-FP7 funded ERC CrowdLand project, grant no. 617754.

References

1. Chatterjee, S., Bhattacharyya, M.: A biclustering approach for crowd judgment analysis. In: Proceedings of the Second ACM IKDD Conference on Data Sciences. pp. 118–119. ACM (2015)
2. Comber, A., Brunsdon, C., See, L., Fritz, S., McCallum, I.: Comparing expert and non-expert conceptualisations of the land: an analysis of crowdsourced land cover data. In: Spatial Information Theory, pp. 243–260. Springer (2013)
3. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the em algorithm. Applied statistics pp. 20–28 (1979)
4. Dempster, A.P., et al.: Maximum likelihood from incomplete data via the EM algorithm. JRSS Ser. B pp. 1–38 (1977)
5. Jagabathula, S., et al.: Reputation-based worker filtering in crowdsourcing. In: Advances in Neural Information Processing Systems. pp. 2492–2500 (2014)
6. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: Advances in neural information processing systems. pp. 1953–1961 (2011)
7. Khattak, F.K., Salleb-Aouissi, A.: Improving crowd labeling through expert evaluation. In: 2012 AAAI Spring Symposium Series (2012)
8. Kim, H.C., Ghahramani, Z.: Bayesian classifier combination. In: International conference on artificial intelligence and statistics. pp. 619–627 (2012)
9. Liu, Q., Peng, J., Ihler, A.T.: Variational inference for crowdsourcing. In: Advances in Neural Information Processing Systems. pp. 692–700 (2012)
10. Moreno, P.G., Teh, Y.W., Perez-Cruz, F., Artés-Rodríguez, A.: Bayesian nonparametric crowdsourcing. arXiv preprint arXiv:1407.5017 (2014)

11. Pareek, H., Ravikumar, P.: Human boosting. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13). pp. 338–346 (2013)
12. Raykar, V.C.: Eliminating Spammers and Ranking Annotators for Crowdsourced Labeling Tasks. JMLR 13, 491–518 (2012)
13. Raykar, V.C., et al.: Learning from crowds. The Journal of Machine Learning Research 11, 1297–1322 (2010)
14. Salk, C.F., Sturn, T., See, L., Fritz, S., Perger, C.: Assessing quality of volunteer crowdsourcing contributions: lessons from the cropland capture game. International Journal of Digital Earth pp. 1–17 (2015)
15. See, L., et al.: Building a hybrid land cover map with crowdsourcing and geographically weighted regression. ISPRS Journal of Photogrammetry and Remote Sensing 103, 48–56 (2015)
16. Sheshadri, A., Lease, M.: Square: A benchmark for research on computing crowd consensus. In: First AAAI Conference on Human Computation and Crowdsourcing (2013)
17. Simpson, E., et al.: Dynamic bayesian combination of multiple imperfect classifiers. In: Decision Making and Imperfection, pp. 1–35. Springer (2013)
18. Tong, H., Li, M., Zhang, H., Zhang, C.: Blur detection for digital images using wavelet transform. In: Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on. vol. 1, pp. 17–20. IEEE (2004)
19. Zauner, C.: Implementation and benchmarking of perceptual image hash functions. Ph.D. thesis (2010)
20. Zhu, X., et al.: Co-training as a human collaboration policy. In: AAAI (2011)