Artificial Intelligence Research and Development
I. Sanz et al. (Eds.)

doi:10.3233/FAIA230699

Mathematical and Computational Models for Crowdsourced Geolocation

Rocco BALLESTER [a] Yanis LABEYRIE [a,b] Mehmet Oguz MULAYIM [a]
Jose Luis FERNANDEZ-MARQUEZ [c] Jesus CERQUIDES [a]
[a] *IIIA-CSIC, Campus UAB, 08193 Cerdanyola, Spain*
[b] *Ecole Centrale de Marseille*
[c] *Centre Universitaire d'Informatique, University of Geneva*

Abstract. Social media generate large amounts of almost real-time data which can turn out extremely valuable in an emergency situation, specially for providing information within the first 72 hours after a disaster event. Despite there is abundant state-of-the-art machine learning techniques to automatically classify social media images and some work for geolocating them, the operational problem in the event of a new disaster remains unsolved. Currently the state-of-the-art approach for dealing with these first response mapping is first filtering and then submitting the images to be geolocated to a crowd of volunteers, assigning the images randomly to the volunteers. In this work we evaluate the potential of artificial intelligence (AI) to assist emergency responders and disaster relief organizations in geolocating social media images from a zone recently hit by a disaster by (i) building a simple model of the capabilities of each of the volunteers of the crowd; and (ii) intelligently assigning tasks to those volunteers which can perform them better. We present in this paper some new methods that outperform random allocation of tasks in a simplified, scalable scenario. Moreover, we show that for a given set of tasks and volunteers, we are able to process them with a significantly lower annotation budget, that is, we are able to make less volunteer solicitations without loosing any quality on the final consensus.

Keywords. social media, disaster response, machine learning, geolocation, crowdsourcing

1. Introduction

Photo geolocation is a hard task that has not been efficiently solved at the present time despite many efforts and plenty of different techniques and strategies. Currently, the state-of-the-art approach for geolocation is first filtering and then submitting the images to be geolocated to a crowd of volunteers (e.g., [4]), assigning the images randomly to the volunteers. In this work we aim to build a paradigm to learn important characteristics about the annotators and use this information to get a much more efficient consensus.

Our primary motivation for precise outdoor image geolocation is its application to the Disaster Management field, where it can have a critical impact because a fast response is of paramount importance to help emergency aid. Our results show that we are able to learn volunteers' error profiles accurately enough so as to later use this information to increase the quality on our geolocations.

In the following section we provide a formalization of the main problems we will be dealing with, namely the *crowdsourced geolocation problem* and the *active crowdsourced geolocation problem*. Later, we define two statistical models, and several estimators and compare them on their quality when solving problems. We end up with a brief overview of the related work and some conclusions. To the best of our knowledge, this is the first time when the crowdsourced image geolocation problem has been mathematically formalized.

All experiments and analyses shown in this work can be found and reproduced at https://github.com/IIIA-ML/crowdsourced_geolocation.

2. Problem formalization

Definition 1 (Crowdsourced Geolocation Problem (CGP))**.** In the CGP we are interested in determining as accurately as possible the locations (in $\mathbb{R}^{\mathtt{d}}$) of $\mathtt{t}$ points given as a set of tasks $T = [1..\mathtt{t}]$. The information at our disposal to locate the points comes from a set of $\mathtt{a}$ *annotations* (i.e., reported locations) $A = [1..\mathtt{a}]$ by a set of $\mathtt{w}$ individuals $W = [1..\mathtt{w}]$ which we refer to as *workers* or *annotators*. For each reported location $a \in A$, we know (i) the point $t_a \in T$ for which the location is reported, (ii) the worker $w_a \in W$ that performed the annotation, and (iii) the value of the position reported $r_a \in \mathbb{R}^d$. So, the input to the CGP is a tuple $\langle t, w, r\rangle$ where $t \in T^{\mathtt{a}}, w \in W^{\mathtt{a}}$, and $r \in \mathbb{R}^{\mathtt{a}\times\mathtt{d}}$, and its output is a matrix $x \in \mathbb{R}^{\mathtt{t}\times\mathtt{d}}$ containing the estimated positions for the points in T.

In general, the real locations of the points are unknown. We will refer to the real location of point $k \in T$ as $p_k \in \mathbb{R}^{\mathtt{d}}$. We will use $w_j^{-1} = \{a \in A | w_a = j\}$ to denote the set of annotations performed by worker j and $t_k^{-1} = \{a \in A | t_a = k\}$ for the set of annotations of task k. In the following, we will assume $\mathtt{d} = 1$; that is, we will assume that the points have to be located in a line.

As in the CGP in the Active Crowdsourced Geolocation Problem (ACGP) we are interested in determining as accurately as possible the locations of $\mathtt{t}$ points. However, in this case the annotations are not provided to us, but instead we are entitled to select the annotators for each task.

Definition 2 (Active Crowdsourced Geolocation Problem (ACGP))**.** We are given a set of $\mathtt{w}$ individuals $W = [1..\mathtt{w}]$ which we refer to as *workers* and a set of $\mathtt{t}$ objects $T = [1..\mathtt{t}]$ which we need to geolocate and an annotation budget, which indicates how many annotations we can make. We are requested to define an annotation policy that at each point in time and based on the result so far, decides which is the next task to annotate and to which worker should the annotation be requested. The resulting annotation is incorporated as an input to the problem. Furthermore, after obtaining the set of $\mathtt{a}$ annotations $A = [1..\mathtt{a}]$, the output of the problem is a matrix $x \in \mathbb{R}^{\mathtt{t}\times\mathtt{d}}$ containing the estimated positions for the $\mathtt{t}$ points.

3. The Constant Normal Model

The simplest model we will introduce for crowdsourced geolocation assumes that the reported locations follow a normal distribution centered in the real value of the point and with a standard deviation which is different for each annotator.

Definition 3 (Constant Normal Model (CNM))**.** We say that the set of annotations A follow a CNM with parameter $\sigma = \langle \sigma_1, \ldots, \sigma_{\mathtt{w}} \rangle$ if for each annotation $a \in A$ we have that $r_a \sim N(p_{t_a}, \sigma_{w_a})$.

Next, in Sections 3.1-3.3 we propose different estimators for σ, provided the input of a CGP problem (pseudo-code algorithms for these estimators can be found in the supplementary material). Then, in Section 3.4, we compare the different estimators proposed and we evaluate how much benefit we can take of the constant normal model for the CGP in different scenarios.

3.1. Direct estimation

An initial estimation of σ can be obtained by (i) using the common estimator of the mean of a normal distribution for the position of each point, and then (ii) estimating sigma of each annotator by using the common estimator for the standard deviation. Formally, the direct estimation method proposes to estimate σ as $\hat{\sigma_j} = \sqrt{\frac{\sum_{i \in w_j^{-1}} (r_i - \hat{p}_{t_i})^2}{|w_j^{-1}|}}$ (1), where $\hat{p}_k = \frac{\sum_{i \in t_k^{-1}} r_i}{|t_k^{-1}|}$ (2). Thus, the final consensus can be computed by a weighted average.

3.2. Iterative estimation

In the former section we have proposed an estimator for σ which in turn relies on an estimator for p_k. There, we used the mean of the locations reported for task k as the estimator. This is the Maximum Likelihood Estimate (MLE) of the mean as long as the annotators have equal variances. However, when the annotators have different variances, the expression for the MLE of the mean is different, as shown below (proof provided in the supplementary material).

Proposition 1. *Provided a set of $x_1, \ldots, x_n$ values such that each x_i is a sample from $N(\mu, \sigma_i)$, the MLE for μ is $\hat{\mu} = \frac{\sum_{i=1}^n \sigma_i^{-2} x_i}{\sum_{i=1}^n \sigma_i^{-2}}$* (3).

Note that Equation 3 can be understood as computing a weighted average of the observations. We can define the weight of observation x_i as $\omega_i = \frac{\sigma_i^{-2}}{\sum_{j=1}^n \sigma_j^{-2}}$ (4), with $0 < \omega_i < 1$ for each $i \in [1..n]$ and $\sum_{i=1}^n \omega_i = 1$, and then the MLE estimate can be expressed as $\hat{\mu} = \sum_{i=1}^n \omega_i x_i$ (5). The iterative estimation method takes it from where the direct estimation finished, and uses Proposition 1 to improve the estimate of the means, provided that we have an estimate for σ. Thus, provided we have an estimate $\hat{\sigma}$, it computes the means as $\hat{p}_k = \frac{\sum_{i \in t_k^{-1}} r_i \hat{\sigma}_{w_i}^{-2}}{\sum_{i \in t_k^{-1}} \hat{\sigma}_{w_i}^{-2}}$ (6). The resulting algorithm is described in Algorithm 1 in the supplementary material.

A reasonable criterion for convergence is that the norm of the difference between $\hat{\sigma}$ this iteration and the previous one is below a specific threshold. When analyzing the results of estimation, we will realize that the iterative estimator estimates σ_j to be 0 for the best annotator. Thus, its usage leads to erroneous estimates.

3.3. Conservative estimation

In this section, we try to tackle the problem of iterative estimation, namely that it overestimates the accuracy of one of the annotators. This objective is accomplished in the conservative estimation method by ensuring that the estimates for σ are *conservative*. By conservative we imply that, when used to compute the MLE of the means by Equation 3, the weights we assign to each observation (the ω_i's defined in Equation 4) are never too large or too small. In this sense, the most conservative estimate for σ will be the one which assigns equal value to every annotator. This will result in the weights ω_i being $\frac{1}{n}$, and the corresponding estimate (Equation 5) will be the average mean. Hence a conservative estimator for σ will result in variances larger than its true value for the more accurate annotators and variances smaller than the true value for the less accurate annotators.

The main idea to build a conservative estimator is to use different estimators for the mean for each of the annotators. In this way, instead of computing a single estimate $\hat{p}_k$ for each task $k \in T$, we compute an estimate $\hat{p}_{k,j}$ for each task $k \in T$, and for each annotator $j \in W$. $\hat{p}_{k,j}$ differs from $\hat{p}_k$ in that it excludes the annotations of annotator j. Hence given $\hat{\sigma}$, we compute $\hat{p}$ using $\hat{p}_{k,j} = \frac{\sum_{i \in t_k^{-1} \setminus w_j^{-1}} r_i \hat{\sigma}_{w_i}^{-2}}{\sum_{i \in t_k^{-1} \setminus w_j^{-1}} \hat{\sigma}_{w_i}^{-2}}$ (7), and given $\hat{p}$, we compute $\hat{\sigma}$ as $\hat{\sigma_j} = \sqrt{\frac{\sum_{i \in w_j^{-1}} (r_i - \hat{p}_{t_i,j})^2}{|w_j^{-1}|}}$ (8). The conservative estimator is described in Algorithm 2 in the supplementary material.

To improve our estimations, we modified our estimator by adding a check to ensure balanced weights and replacing the weight of the best annotator with that of the second-best. This upgraded approach will be called *conservative-2* estimation and aims to prevent overconfidence scenarios. We will show in section 3.4 that this modification results in better estimations.

3.4. Experimental setting and results

In this section, we explain the experimental setup[1] used in our study and compare different estimators based on results obtained from synthetic data. The experimental parameters, including number of points, number of annotators, number of annotators a point is given to annotate (the so-called redundancy), sigma distribution, and points distribution are varied to assess their impact on the methods' performance. Each experiment is repeated 50 times for statistical significance. Performance of each method is measured using mean location norm error and

[1]All experiments and analyses shown in this work can be found and reproduced at https://github.com/IIIA-ML/crowdsourced_geolocation

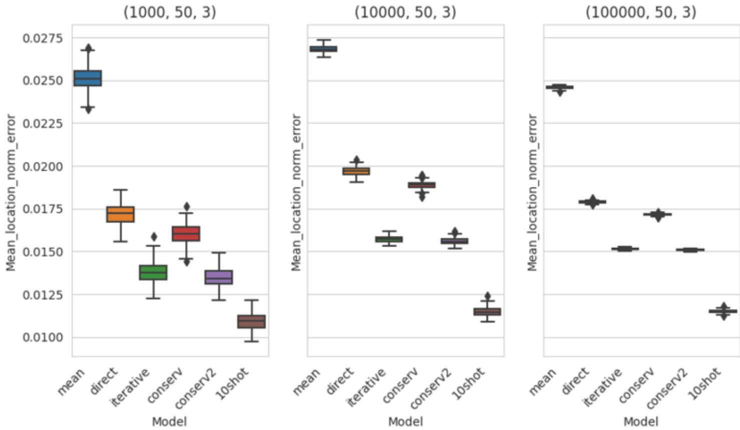

Figure 1. Location error depending on number of points annotated.

mean sigma norm error, which calculate the difference between predicted and true locations or sigmas.

Our baseline is the current state of the art practice, which consists in obtaining the locations by just computing the mean of the annotations at each point. The following results show the evaluation of this practice plus the regarding to the already explained methods. Furthermore, the active strategy is also shown (named *10Shot* in the plots) which will be explained later on.

The influence of the total number of points on the methods' precision is shown in Fig.1. We start with a redundancy value set to 3, and vary the number of points between 1000, 10000, and 100000. The points are requested at random from a pool of 50 annotators with variances sampled from a uniform distribution between 0 and 0.1.

More points do not significantly improve annotation quality, but they do reduce error variance. Working with more points leads to better mean and variance estimations. We then investigate how much less annotation budget is needed for a given error level when using a more advanced method. Fig. 2 shows the mean location error as a function of redundancy for different methods and error levels.

We can see that if we for instance set the desired error value to 0.01, the current practice would require around 20 annotators per image while the *direct* method would only need 5 and the *conservative-2* or *iterative* method would only need 3. Thus, looking at Fig.2 we could state that the best overall method for the CGP is the *conservative-2 estimation*, solving it with a significantly lower annotation budget having fixed the error.

3.5. Active strategy for the ACGP: the K-Shot method

In this section, we tackle the Active Crowdsourced Geolocation Problem (ACGP) where we are given an annotation budget and we actively decide which annotator is responsible for annotating each of the points.

We solve the ACGP implementing the K-Shot method, where we divide the points into K batches. We randomly select annotators to geolocate the first batch

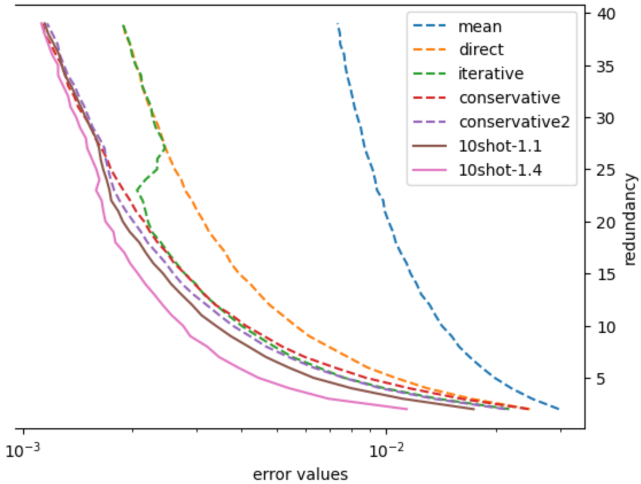

Figure 2. Mean location error depending on the redundancy value. CGP methods are shown in dotted lines while the ACGP method, 10shot, is shown in solid lines and using a greediness value of 1.1 and 1.4.

and estimate their variance profiles using *conservative-2* estimator. Then, we choose annotators with the best profiles for the following batches using the greediness parameter, properly explained later on in this section. We repeat this process for all batches until all points are annotated. Algorithm 3 in the supplementary material summarises this approach.

The greediness parameter helps control how many times certain points are requested to be annotated by annotators. A higher greediness value means that the best annotators are more likely to receive requests to annotate points. A greediness value of $\delta = 1$ distributes requests uniformly, while a value of $\delta = 2$ doubles the probability of the best annotator receiving requests compared to the worst annotator. In other words, this parameter balances precision and cost.

After several analyses, a greediness value of $\delta = 1.1$ ensures that the best annotator does not annotate more than twice the points it would have annotated if the points would have been evenly requested among all workers.

3.6. Comparison between CGP and ACGP

In this section, we aim to show that we can indeed benefit from actively assigning the points to the workers intelligently based on learning their variance profiles. We decided to arbitrary set $K = 10$ and $\delta = 1.1$ for the active method described in the previous section. As explained before, this greediness value guarantees that we do not overcharge some annotators.

K-Shot method outperforms other estimators in all experiment setups (Fig.1 and Fig.2), regardless of the number and distribution of points, and even with a small greediness. Moreover, and as it is expected, the precision of this method can be increased by raising the greediness parameter.

4. The Geographic Normal Model

In the previous section, we presented the constant normal model, where each annotator had a single parameter –namely, their standard deviation σ_j– and we assumed that the reported location for any point located by the annotator followed a normal distribution centered on the true position of the point with σ_j as standard deviation. In this section, we allow the standard deviation of an annotator to vary depending on the true position of the point. That is, instead of a single parameter $\sigma_j \in \mathbb{R}$, in the Geographic Normal Model (GNM) we assume that $\sigma_j : \mathbb{R}^{\mathtt{d}} \to (0, \infty)$ is a continuous function of location.

Definition 4 (Geographic Normal Model (GNM)). We say that annotations follow a GNM with standard deviation functions $\sigma = \langle \sigma_1, \ldots, \sigma_{\mathtt{w}} \rangle$ if for each annotation $a \in A$, we have that $r_a \sim N(p_{t_a}, \sigma_{w_a}(p_{t_a}))$.

4.1. Parametric estimation

In this section, we propose to use a parametric expression to approximate the σ_j functions. Our approximation parameterizes each function by means of a number $\kappa \in (0, \infty)$, a set of $\mathtt{c}$ *reference points* $\mathbf{x} = \{x_1, \ldots, x_{\mathtt{c}}\}$, with $x_i \in \mathbb{R}^{\mathtt{d}}$, and a set of weights $\mathbf{v} = \{v_1, \ldots, v_{\mathtt{c}}\}$, with $v_i \in (0, \infty)$. Provided $\mathbf{x}$, $\mathbf{v}$, and a point p, the standard deviation at p is approximated as $s(p; \kappa, \mathbf{x}, \mathbf{v}) = \frac{\sum_{i=1}^{\mathtt{c}} v_i \exp(-\kappa d(p,x_i))}{\sum_{i=1}^{\mathtt{c}} \exp(-\kappa d(p,x_i))}$, where $d(p, x_i) = \|p - x_i\|$, that is, the Euclidean distance between p and x_i. We keep κ and $\mathbf{x}$ the same for all workers, but we have a different set of weights $\mathbf{v}^j$ for each worker $j \in W$. In the following we note $\hat{\sigma}_j(p) = s(p; \kappa, \mathbf{x}, \mathbf{v}^j)$ and so we have that for each $a \in A$, $r_a \sim N(\hat{p}_{t_a,w_a}, \hat{\sigma}_{w_a}(\hat{p}_{t_a}))$ where $\hat{p}_{k,j} = \frac{\sum_{i \in t_k^{-1} \setminus w_j^{-1}} r_i \hat{\sigma}_{w_i}(\hat{p}_k)^{-2}}{\sum_{i \in t_k^{-1} \setminus w_j^{-1}} \hat{\sigma}_{w_i}(\hat{p}_k)^{-2}}$, and $\hat{p}_k = \frac{\sum_{i \in t_k^{-1}} r_i}{|t_k^{-1}|}$ is the usual mean.

Now that the model is defined, our objective is to find the values for $\{\mathbf{v}^1, \ldots, \mathbf{v}^{\mathtt{w}}\}$ with maximum likelihood on that model. The problem can be effectively solved through quasi-Newton methods such as LBFGS [9] as long as the number of annotators and the number of reference points is not very large. Using MCMC simulation to obtain the mean of a set of posterior distribution samples with the help of the Stan software and by fixing the number of reference points to 15, we can get a proper estimation, as we will see in the following section.

When using this method to solve the CGP it will be referred as the *Spatial OneShot* method, as opposed to the name given to the active strategy.

4.2. Experimental setting and results

New experiments use continuous functions instead of single values to model annotators' profiles. Cubic polynomials are generated with a maximum height of 0.2 and a minimum of 0.001 using Scipy's CubicSpline library. Moreover, 10000 points, 10 annotators, a redundancy value of 5 and the same error metric are used. Again, each experiment is repeated 50 times for statistical significance.

Spatial OneShot is the best estimator according to Fig.3 (left), except for the *Spatial K-Shot* method. This is expected since the previous methods try to estimate constant variances.

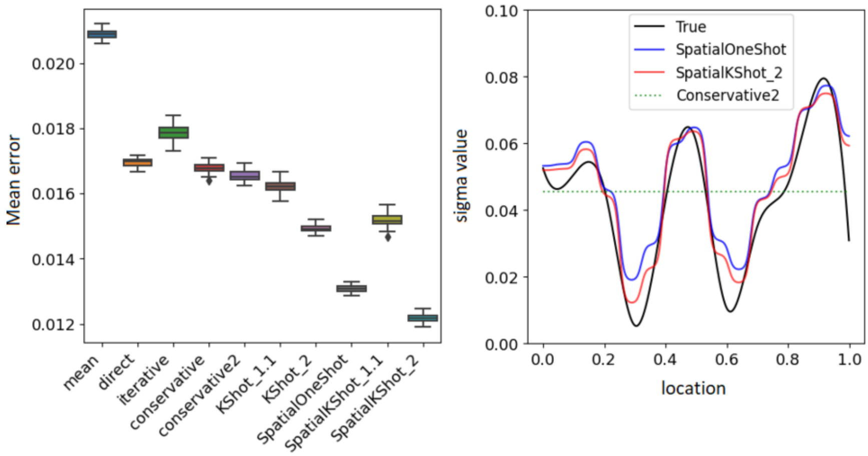

Figure 3. (Left) Mean norm error for each of the methods using $\delta = 1.1$ and $\delta = 2$ for the active methods. (Right) Annotator's variance profile (in black) and the variance profile learned by different methods using $\delta = 2$ for the active method.

4.3. Active strategy for the ACGP: the Spatial K-Shot method

The Spatial K-Shot method is similar to the K-Shot method. Still, it differs: it divides points into two groups, using a small percentage of the points to learn the variance profiles and iterating through the other points to recompute the variance functions. Initial annotations are randomly requested, while subsequent annotations are more likely to come from the best annotators, depending on the greediness parameter, as mentioned before. The algorithm described can be found in Algorithm 4 in the supplementary material.

4.3.1. Comparison between CGP and ACGP

In this section, we compare the methods regarding the CGP described previously and the active method exposed before. To do so, we again divided the K-Shot method in $K = 10$ equal batches and we used 10% of the points as the first batch in the Spatial K-Shot method. Moreover, we used a greediness value of $\delta = 1.1$ and $\delta = 2$ for both active methods. In Fig.3 (left) we can see the performance of the methods by using the mean error metric. We can see that the Spatial K-Shot method outperforms the rest only when the greediness parameter is high enough. Just as in our first model, the higher the greediness, the better the active method performs. However, a really high greediness value could have a detrimental effect since the best annotators would be overdemanded.

5. Related work

The difficulty in the automatic geolocation of photos without geotags has led to the use of crowdsourcing to compensate for the shortcomings of machine learning methods (e.g., [1, 6, 10, 13]). Geolocation precision is critical in emergencies to guide first responders. Studies show that leveraging the wisdom of the crowd improves data quality, but task assignment is challenging and requires worker expertise and quality (e.g., [8]). Research on worker-task matching spans a broad spectrum of approaches, such as modelling worker skills (e.g., [11]), spatial crowdsourcing (e.g., [15]), and incorporating a trust model between workers to attain data quality (e.g., [14]). We contribute to creating computational models of worker geolocation performance, building on a probabilistic framework for annotation aggregation [3]. Familiarity with regions of interest, independent of worker location, may also benefit geolocation tasks. This paper presents a geolocation-specific extension to the mentioned probabilistic framework and will be integrated into the Crowdnalysis software library [2]. Refer to [5, 7, 12] for further perspectives on crowdsourcing and different assignment approaches.

6. Conclusions and future work

This paper presents two different models, the Constant Normal Model and the Geographic Normal Model, to tackle the Crowdsourced Geolocation Problem and the Active Crowdsourced Geolocation Problem, respectively. One key aspect of this work is the focus on learning annotators' variance profiles w.r.t. location of tasks, an innovative approach that surpasses current state-of-the-art methods. Several dedicated methods have been developed to achieve this, such as the *conservative-2* and *K-Shot* algorithms, which enhance consensus quality and reduce annotation costs. We believe these contributions advance the field of crowdsourced geolocation and hold potential for various domains of application, such as crowdsourced data classification or crowdsourced data evaluation.

There are several future research avenues for further enhancement of the proposed methods. Firstly, analysing a broader range of scenarios and determining the optimal method under different conditions would be beneficial. This could involve fixing variables at different values, exploring various combinations of the number of points and annotators, and even utilizing different data distributions. Optimizing specific hyperparameters that were arbitrarily chosen, such as the number of batches in active methods or the number of reference points in the parametric estimator, could lead to improved results.

Moreover, both models can be enhanced by increasing the dimensions used. For instance, expanding from one dimension to two dimensions would enable an effective representation of latitude and longitude. This enhancement would facilitate more accurate and context-aware predictions, empowering the models to better understand and interpret spatially related data.

Additionally, active learning approaches could be used to cease requesting annotations for instances that have already achieved a clear enough consensus. This effective strategy would reduce costs by eliminating the need for further annotations in such cases. Last but not least, it remains a future task to implement and validate the methods described in a real-world scenario, gathering and requesting data from actual citizens.

Acknowledgements. This work has received funding from the European Union's Horizon 2020 research and innovation programme HumanE-AI-Net and Crowd4SDG projects (N. 952026 and N. 872944, respectively).

.

References

[1] Carlo Bono, Mehmet Oğuz Mülâyim, Cinzia Cappiello, Mark James Carman, Jesus Cerquides, Jose Luis Fernandez-Marquez, Maria Rosa Mondardini, Edoardo Ramalli, and Barbara Pernici. A citizen science approach for analyzing social media with crowdsourcing. *IEEE Access*, 11:15329–15347, 2023.

[2] Jesus Cerquides and Mehmet Oğuz Mülâyim. crowdnalysis: A software library to help analyze crowdsourcing results. https://doi.org/10.5281/zenodo.5898579, January 2022.

[3] Jesus Cerquides, Mehmet Oğuz Mülâyim, Jerónimo Hernández-González, Amudha Ravi Shankar, and Jose Luis Fernandez-Marquez. A conceptual probabilistic framework for annotation aggregation of citizen science data. *Mathematics*, 9(8), 2021.

[4] Ramian Fathi, Dennis Thom, Steffen Koch, Thomas Ertl, and Frank Fiedrich. VOST: A case study in voluntary digital participation for collaborative emergency management. *Information Processing & Management*, 57(4):102174, July 2020.

[5] David Geiger, Stefan Seedorf, Thimo Schulze, Robert C. Nickerson, and Martin Schader. Managing the Crowd: Towards a Taxonomy of Crowdsourcing Processes. *AMCIS 2011 Proceedings - All Submissions*, August 2011.

[6] Clemens Havas, Bernd Resch, Chiara Francalanci, Barbara Pernici, Gabriele Scalia, Jose Luis Fernandez-Marquez, Tim Van Achte, Gunter Zeug, Maria Rosa Mondardini, and Domenico Grandoni. E2mc: Improving emergency management service practice through social media and crowdsourcing analysis in near real time. *Sensors*, 17(12):2766, 2017. Publisher: MDPI.

[7] Danula Hettiachchi, Vassilis Kostakos, and Jorge Goncalves. A survey on task assignment in crowdsourcing. *ACM Comput. Surv.*, 55(3), feb 2022.

[8] Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, page 1301–1318, New York, NY, USA, 2013. Association for Computing Machinery.

[9] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989. Publisher: Springer.

[10] Amudha Ravi Shankar, Jose Luis Fernandez-Marquez, Barbara Pernici, Gabriele Scalia, Maria Rosa Mondardini, and Giovanna Di Marzo Serugendo. Crowd4Ems: A crowdsourcing platform for gathering and geolocating social media content in disaster response. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:331–340, 2019.

[11] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 889–896, New York, NY, USA, 2009. Association for Computing Machinery.

[12] Eric Schenk and Claude Guittard. Crowdsourcing: What can be Outsourced to the Crowd, and Why ?, December 2009.

[13] Sukrit Venkatagiri, Jacob Thebault-Spieker, Rachel Kohler, John Purviance, Rifat Sabbir Mansur, and Kurt Luther. Groundtruth: Augmenting expert image geolocation with crowdsourcing and shared representations. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW), nov 2019.

[14] Han Yu, Zhiqi Shen, Chunyan Miao, and Bo An. Challenges and opportunities for trust management in crowdsourcing. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 486–493, 2012.

[15] Yan Zhao, Yang Li, Yu Wang, Han Su, and Kai Zheng. Destination-aware task assignment in spatial crowdsourcing. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 297–306, New York, NY, USA, 2017. Association for Computing Machinery.