https://doi.org/10.26599/IJCS.2022.9100037 International Journal of Crowd Science

Unsupervised Classification by Iterative Voting

Evgeny Kagan[1] ✉ and Alexander Novoselsky[2]

ABSTRACT

In the paper we present a simple algorithm for unsupervised classification of given items by a group of agents. The purpose of the algorithm is to provide fast and computationally light solutions of classification tasks by the randomly chosen agents. The algorithm follows basic techniques of plurality voting and combinatorial stable matching and does not use additional assumptions or information about the levels of the agents' expertise. Performance of the suggested algorithm is illustrated by its application to simulated and real-world datasets, and it was demonstrated that the algorithm provides close to correct classifications. The obtained solutions can be used both separately and as initial classifications in more complicated algorithms.

KEYWORDS

decision making; unsupervised classification; plurality voting; uncertainty

Classification problem is one of the basic tasks that are solved by any living organism and origin the scientific and cultural evolution of humanity. Formally such problem is considered as a problem of relating certain items with the classes or dividing the variety items into the classes with respect to definite measure of similarity[1].

With respect to the considered setup, classification is based on different assumptions and conducted by different methods. For example, in the classification by a single agent, similarity measures between the items are assumed to be known, and the problem is solved by certain clustering methods. In the classification by several agents, in contrast, relations between the items and the classes are derived from the opinions of the agents and the main problem is an aggregation of the agents' classification into the group opinion.

Widely accepted method of aggregating opinions of several agents is voting, which in its different versions is applied in the most of political and social communities and used in the decision problems. Together with obvious similarity between voting in political and social communities and professional committees, there is also an obvious difference. In the first case, the group of agents includes both experts and non-experts in the considered field, and in the second case the group includes only competent agents, while opinions provided by non-experts are omitted.

With the development of social networks and Internet communication, the voting techniques were extended to the arbitrary groups that include both competent and non-competent agents. Such approach is widely known as the "wisdom of the crowd"[2,3]. Similar to the political and social votes, aggregation of the agents' opinions is conducted by a certain version of plurality voting, and an opinion chosen by the plurality of the agents is considered as a correct one.

However, since in the "wisdom of the crowd" approach, the resulting opinion is strongly influenced by the opinions of non-expert agents, direct application of such techniques often fails.

To avoid the influence of non-experts and bring the aggregated opinion closer to the correct one, there were several suggested methods (e.g., Ref. [4]).

The first type of such methods, the supervised classification, considers classification as a process of supervised learning with the known correct classification. Then it is required either to choose the agents such that each of them provides an opinion, which is close to the correct one, or to train all the agents using the known correct classification. As a result, the chosen or leant agents acting individually or in group will immediately provide correct or close to correct classification in further considerations of similar classification tasks.

The second type of the classification methods is unsupervised classification. In these methods, correct classification is unknown, and it is required to choose the agents such that certain combination of their opinions can be considered as correct classification. Verification of the criterions for choosing the agents and of the method of combination of their opinions is conducted using the tasks with known solutions. The chosen agents are considered as experts and, similar to above, these agents and verified methods of combination of their opinions result in correct or close to correct classification in the classification tasks in similar domains of knowledge.

One of the most effective classification techniques that apply the weighted agents' opinions is based on the Dawid and Skene (DS) algorithm[5], which implements the expectation-maximization approach. Following this approach, classification is conducted by iterating two steps[6]: expectation of the correct choice with respect to the agents' expertise, and maximization of the likelihood of the agents' expertise with respect to the expected correct choice. Such techniques form an effective addition to the plurality voting approach that provides close to correct classifications in the case of small groups of agents, where the competent agents can be clearly separated from the non-competent agents.

However, in most cases the experts cannot be distinguished, and resulting classification based on the opinions of all the agents, both experts and non-experts, is far from the correct one. The same problem appears when the agents' opinions are biased. In

1 Department of Industrial Engineering, Ariel University, Ariel 40700, Israel
2 Weizmann Institute of Science, Rehovot 76100, Israel
Address correspondence to Evgeny Kagan, evganyk@ariel.ac.il

other words, the existing algorithms provide correct results for honest competent agents and fail for non-competent or dishonest agents.

In order to overcome this problem, we suggest a simple algorithm for unsupervised classification of given items by an arbitrary group of agents that, on one hand, follows basic techniques of plurality voting, and, on the other hand, implements the well-known approach of stable matching[7].

In the considered version of the algorithm, the iterations required by original stable matching algorithm are substituted by double plurality voting such that the second voting is conducted only by the agents whose opinion is close to the opinion, which was chosen by the plurality at the first voting.

Despite the simplicity, the algorithm provides clear choice of the authoritative agents such that their choice may strongly differ from the choice of the plurality obtained at the first step and is close or even equal to correct one.

The algorithm was verified and tested on the simulated and real-world datasets[8,9]. It was shown that in all considered cases it provides efficient solution, and the resulting classification is close to the correct classification.

Surprisingly, the obtained results track the suggested method back to origins of the voting methods and the wisdom of the crowd techniques. In Refs. [10, 11], Galton stressed that the estimated value obtained by the voting differs from theoretical mean value, so either the votes should be weighted, or the estimated value different from the mean should be considered.

In the suggested algorithm, the estimated solution is provided by both experts and non-experts and used as a reference for choosing the experts, while final classification, which usually differs from the estimated solution, combines the opinions of the authoritative agents.

1 Problem Formulation and Setup Example

The problem addresses the situation, in which a set of items should be distributed over several classes by the group of agents. For example, if the items are the paintings and the classes are the painters, then the classification problem requires associating each painting with the painter. The main problem is to aggregate opinions of the agents into a collective classification, which is as close as possible to the unknown correct classification.

1.1 Formal setup

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n items that should be distributed over m classes $C_1, C_2, \ldots, C_m$, $1 < m < n$, such that each item x_i is included only in a single class C_j and all the items are included into appropriate classes. The set $\gamma = \{C_1, C_2, \ldots, C_m\}$ of classes is a partition of the set X, that is, $C_j \subset X$, $j = 1, 2, \ldots, m$, $C_{j'} \cap C_{j''} = \varnothing$ for $j' \neq j''$, and $\bigcup_{j=1}^{m} C_j = X$.

Assume that the classification is conducted by a group $A = \{a_1, a_2, \ldots, a_l\}$ of l agents and each agent $a_k \in A$, $k = 1, 2, \ldots, l$, provides classification, which is represented by the agent's partition $\gamma_k = \{C_{k,1}, C_{k,2}, \ldots, C_{k,m}\}$. Then the problem is to aggregate the agents' partitions γ_k into a single partition $\gamma = \{C_1, C_2, \ldots, C_m\}$ such that it as best as possible represents the unknown correct partition $\gamma^* = \{C_1^*, C_2^*, \ldots, C_m^*\}$.

As indicated above, the most popular and the simplest method of aggregation of the agents' partitions is the plurality voting in which the item is included into a class that was chosen by most agents (the ties are broken randomly). More sophisticated methods consider the levels of the agent's expertise and choose the classes following these levels. Below, consider an example of the setup and classifications obtained by the plurality voting.

1.2 Setup example

To clarify the formulated problem, let us consider a simple example. For consistency and further comparisons, the example follows Ref. [12], which suggested the DS-based approach called "the wisdom in the crowd" that, in contrast to the known "the wisdom of the crowd" techniques, considers the levels of the agents' expertise. The example of the dataset that includes $n = 12$ items classified by $l = 6$ agents to $m = 4$ classes is presented in Table 1.

The agents' partitions γ_1, γ_2, ..., γ_6 are generated from the correct classification

$$\gamma^* = \{\{x_2, x_6, x_8\}, \{x_1, x_3, x_{10}\}, \{x_4, x_5, x_{11}\}, \{x_7, x_9, x_{12}\}\},$$

by random distortion, and partition

$$\gamma_0 = \{\{x_6, x_8, x_{10}\}, \{x_1, x_3, x_9\}, \{x_2, x_4, x_{11}\}, \{x_5, x_7, x_{12}\}\},$$

is the partition obtained by the plurality voting. Notice that since there are several broken ties, this partition is not a unique solution.

Formally, the data setup is represented by the $n \times l$ relation matrix $R = \|r_{ik}\|_{n \times l}$, where r_{ik} is an index $r_{ik} = j$ of the class C_j, with which the i-th item was associated by the k-th agent.

The presented setup is a usual setup of the classification problem. The agents' partitions are obtained by using certain questionnaires, each of which includes m options with respect to the number of classes, and it is required to aggregate these partitions γ_1, γ_2, ..., γ_l into a partition γ, which will as correct as possible represent an unknown correct classification γ^*. The case of questionaries with different numbers of options is effectively processed by the algorithm suggested by Ratner et al.[13]

2 Suggested Algorithm

The suggested algorithm continues a direction of previously developed methods, in which final classification is created by the agents, who were recognized as experts in certain fields of knowledge.

However, in contrast to the DS algorithm[5] and its successors[6,12], where the experts are defined as agents with similar opinions, in the suggested algorithm the competent agents are defined as the agents whose opinions are in a higher percent of similarity with the opinions defined by plurality voting. Then, the second voting is conducted over the opinions of this group of agents. The

Table 1 Example of the dataset with $n = 12$ items, $l = 6$ agents, and $m = 4$ classes.

Item	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ^*	γ_0
x_1	3	2	2	4	3	1	2	2
x_2	1	3	1	4	3	2	1	3
x_3	3	2	2	1	3	1	2	2
x_4	2	4	4	3	1	3	3	3
x_5	4	4	4	3	1	3	3	4
x_6	1	3	1	1	2	2	1	1
x_7	4	3	3	1	4	4	4	4
x_8	1	3	1	2	2	1	1	1
x_9	2	1	3	2	4	4	4	2
x_{10}	3	2	2	1	3	1	2	1
x_{11}	4	1	4	3	2	3	3	3
x_{12}	3	1	3	1	4	4	4	4

algorithm is outlined as shown in Algorithm 1.

As indicated above, the suggested algorithm acts in three stages. At first, it applies plurality voting and defines classification $\gamma_0 = \{C_{0,1}, C_{0,2}, \ldots, C_{0,m}\}$. Then it recognizes the group $E \subset A$ of experts as a group of the agents whose classifications $\gamma_k = \{C_{k,1}, C_{k,2}, \ldots, C_{k,m}\}$ are close to the classification γ_0. Finally, the algorithm applies plurality voting among the agents from the group E of experts.

Consequently, complexity of the algorithm is defined by the complexity of its stages and is $\mathcal{O}(l(m+n))$, where n is the number of items, m is the number of classes, and l is the number of agents.

To clarify the actions of the suggested algorithm, let us consider a running example. For convenience, the example continues consideration of the dataset presented in Table 1.

Since the classification γ_0 resulted by the plurality voting over the group A of all $l = 6$ agents is already calculated (Lines 1–4 in Algorithm 1), we will continue recognizing the group E of experts (Lines 5–8).

Similarity s_k between classifications γ_k and γ_0 is determined as a ratio of the equivalent indices in the columns $(r_{1,k},\ r_{2,k},\ \ldots,\ r_{i,k})^{\mathrm{T}}$ and $(r_{1,0},\ r_{2,0},\ \ldots,\ r_{i,0})^{\mathrm{T}}$ in the dataset table, that is

$$s(\gamma_k, \gamma_0) = (\text{number of equivalent indices in the columns } (r_{1,k},\ r_{2,k},\ \ldots,\ r_{i,k})^{\mathrm{T}} \text{ and } (r_{1,0},\ r_{2,0},\ \ldots,\ r_{i,0})^{\mathrm{T}}) \cdot (\text{number of items } n)^{-1}.$$

For example, the columns $r_{i,1}$ and $r_{i,0}$ include equivalent values for the items x_5, x_6, x_7, x_8, and x_9, and different values $r_{i,1}$ and $r_{i,0}$ for the other items. Then, similarity between γ_1 and γ_0 is

$$s(\gamma_1, \gamma_0) = 5/17 = 0.42.$$

By the same manner, similarity between γ_2 and γ_0 is

$$s(\gamma_2, \gamma_0) = 4/17 = 0.33,$$

and so far. The resulting similarities are presented in Table 2.

It is seen that two agents, a_2 and a_5, provided classifications that are far from the classification γ_0, while classifications provided by the other agents are essentially closer to the classification γ_0.

Then, if the $s = 40\%$, then the group of experts E will include the agents a_1, a_3, a_4, and a_6. The dataset after exclusion of the non-competent agents a_2 and a_5 is shown in Table 3.

Finally, the second plurality voting is conducted among the agents a_1, a_3, a_4, and a_6 (Lines 9–12 in the outline of Algorithm 1). The resulting classification (see the column γ in Table 3) is

$$\gamma = \{\{x_2, x_3, x_6, x_8, x_{10}\}, \{x_1, x_9\}, \{x_4, x_5, x_{11}, x_{12}\}, \{x_7\}\}.$$

Note again that since there are several broken ties, this classification is not unique.

It is seen that the current example similarity between this classification γ and correct classification γ^* is

$$s(\gamma^*, \gamma) = 8/17 = 0.67,$$

that is equal to the similarity between classification γ_0 and the correct classification γ^*,

$$s(\gamma^*, \gamma_0) = 8/17 = 0.67.$$

However, for each tie the probability of correct solution is higher. Let us consider this property, which holds in most cases when the similarities are different and even when $s(\gamma^*, \gamma) < s(\gamma^*, \gamma_0)$.

Denote by $p_{ij}(\gamma)$ the probability that in the classification γ, item x_i will be included into the class C_j and consider the probabilities of choosing the classes in the classification γ_0 (the first plurality voting) and in the classification γ (the second plurality voting). These probabilities are shown in Table 4.

It is seen that after the first plurality voting (classification γ_0) by all the agents from the group A, the uncertainty in choosing the

Algorithm 1 Classification by iterative voting

Input: Set $X = \{x_1, x_2, \ldots, x_n\}$ of items,
set $A = \{a_1, a_2, \ldots, a_l\}$ of agents,
number m of classes, $m < n$,
matrix $R = \|r_{ik}\|_{n \times l}$ of labels,
percent s of similarity.

Output: Classification $\gamma = \{C_1, C_2, \ldots, C_m\}$.

First plurality voting: find classification $\gamma_0 = \{C_{0,1}, C_{0,2}, \ldots, C_{0,m}\}$.

1. For each item $x_i \in X$, $i = 1, 2, \ldots, n$, do
2. Choose index j, which appears in the row $(r_{i,1},\ r_{i,2},\ \ldots,\ r_{i,k})$ maximum times (ties are broken randomly).
3. Insert x_i into the class $C_{0,j} \in \gamma_0$.
4. End

Recognition of experts

5. For each agent $a_k \in A$, $k = 1, 2, \ldots, l$, do
6. Compute similarity between classifications γ_k and γ_0.
7. End
8. Choose the group E of agents with percent s of similarity.

Second plurality voting: find classification $\gamma = \{C_1, C_2, \ldots, C_m\}$

9. For each item $x_i \in X$, $i = 1, 2, \ldots, n$, do
10. Choose index j, which appears in the row $(r_{i,1},\ r_{i,2},\ \ldots,\ r_{i,k})$ and corresponds to the agents from the experts' group E maximum times (ties are broken randomly).
11. Insert x_i into the class $C_j \in \gamma$.
12. End

Table 2 Similarities between agents' classifications and the classification obtained by plurality voting.

Item	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6
$s(\gamma_k, \gamma_0)$	0.42	0.33	0.42	0.42	0.33	0.50

Table 3 Dataset after exclusion of non-competent agents.

Item	γ_1	γ_3	γ_4	γ_6	γ^*	γ
x_1	3	2	4	1	2	2
x_2	1	1	4	2	1	1
x_3	3	2	1	1	2	1
x_4	2	4	3	3	3	3
x_5	4	4	3	3	3	3
x_6	1	1	1	2	1	1
x_7	4	3	1	4	4	4
x_8	1	1	2	1	1	1
x_9	2	3	2	4	4	2
x_{10}	3	2	1	1	2	1
x_{11}	4	4	3	3	3	3
x_{12}	3	3	1	4	4	3

Table 4 Probabilities of choosing the classes for classification γ_0 (the first plurality voting) and classification γ (the second plurality voting).

Item	γ_0 (first plurality voting)				γ (second plurality voting)			
	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
x_1	0	1/2	1/2	0	1/4	1/4	1/4	1/4
x_2	1/2	0	1/2	0	1	0	0	0
x_3	1/3	1/3	1/3	0	1	0	0	0
x_4	0	0	1/2	1/2	0	0	1	0
x_5	0	0	0	1	0	0	1/2	1/2
x_6	1	0	0	0	1	0	0	0
x_7	0	0	0	1	0	0	0	1
x_8	1	0	0	0	1	0	0	0
x_9	0	1/2	0	1/2	0	1	0	0
x_{10}	1/3	1/3	1/3	0	1	0	0	0
x_{11}	0	0	1/2	1/2	0	0	1/2	1/2

class remains for 67% of the items; for 8 items from 12 items the first voting does not result in an exact class. However, after the second plurality voting by the agents from the recognized group $E \subset A$ of experts (classification γ), the uncertainty in choosing the class decreases down to 25% of the items; the exact classes were not provided only for 3 items from 12 items. As demonstrated in the next section, for big datasets such decrease in the uncertainty leads to essentially better resulting classifications.

3 Verification of Algorithm

The suggested algorithm was verified using the simulated and real-world datasets. Below, we present the results of these verifications.

3.1 Simulated dataset

The dataset was generated by random perturbation of randomly generated classification. The algorithm of generating the dataset is outlined in Algorithm 2.

In the simulations, the label j for correct classification γ^* (Line 2) and the label j_ε for the agent's classification γ_k (Line 15) were drawn with respect to the uniform distribution, and the value ε for the expert and non-expert agents (Line 8) was drawn with respect to the normal distribution.

An example of the dependence of the percent of the correct classifications on chosen similarities is shown in Fig. 1. The number of items is $n = 50$, the number of agents is $l = 10$ among which the number of experts is $l_e = 2$, and the number of classes is $m = 5$.

In Fig. 1, the x axis represents the thresholding similarity $s(\gamma, \gamma_0)$ between the agents' classifications γ and the classification γ_0 by plurality voting, and y axis shows the percent of correct classifications by the experts chosen using the thresholding similarity $s(\gamma, \gamma_0)$. For example, $s(\gamma, \gamma_0) = 0.9$ means that the group of experts used for the second voting includes the agents whose classifications γ are at least 90% similar to the plurality vote classification γ_0.

It is seen that starting from the threshold similarity $s(\gamma, \gamma_0) = 0.57$, the second plurality voting either improves the resulting classification or results in the same classification as the first plurality voting. The percent of correct classifications is up to 4% higher than that in single plurality voting.

The similar dependencies are observed for any reasonable simulated datasets that approve the applicability of the algorithm and substantiate its verification on the real-world data.

Algorithm 2 Generation of the dataset

Input: set $X = \{x_1, x_2, \ldots, x_n\}$ of items,
set $A = \{a_1, a_2, \ldots, a_l\}$ of agents,
set of experts $E \subset A$,
expert threshold ε_e,
non-expert threshold $\varepsilon_{ne} < \varepsilon_e$,
number m of classes, $m < n$.

Output: correct classification
$\gamma^* = \{C_1^*, C_2^*, \ldots, C_m^*\}$,
agents' classifications
$\gamma_k = \{C_{k,1}, C_{k,2}, \ldots, C_{k,m}\}$, $k = 1, 2, \ldots, l$.

Generate correct classification
$\gamma^* = \{C_1^*, C_2^*, \ldots, C_m^*\}$.
1. For each item $x_i \in X$, $i = 1, 2, \ldots, n$, do
2. Choose label $j \in \{1, 2, \ldots, m\}$ by random.
3. Insert x_i into the class $C_j^* \in \gamma^*$.
4. End

Generate agents' classifications
$\gamma_k = \{C_{k,1}, C_{k,2}, \ldots, C_{k,m}\}$, $k = 1, 2, \ldots, l$.
5. For each agent $a_k \in A$, $k = 1, 2, \ldots, l$, do
6. For each class $C_j^* \in \gamma^*$, $j = 1, 2, \ldots, m$, do
7. For each item $x \in C_j^*$ do
8. Draw $\varepsilon > 0$ by random.
9. If $a_k \in E$, then
10. Set $\tau = \varepsilon_e$,
11. Else
12. Set $\tau = \varepsilon_{ne}$.
13. End
14. If $\varepsilon > \tau$, then
15. Draw $j_\varepsilon \in \{1, 2, \ldots, m\}$by random,
16. Else
17. Set $j_\varepsilon = j$.
18. End
19. Insert x into the class $C_{k,j_\varepsilon} \in \gamma_k$.
20. End
21. End
22. End

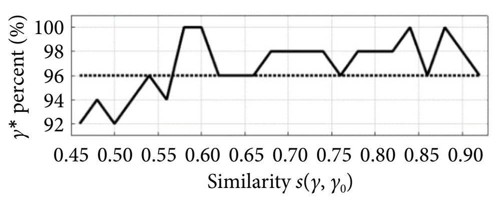

Fig. 1 Percent of correct classifications with respect to the similarity $s(\gamma, \gamma_0)$ between the agents' classifications γ and classification γ_0 by plurality voting in the simulated dataset. The dashed line shows the percent of correct classifications by plurality voting.

3.2 Real-world datasets

For verification we used two real-world datasets. The first dataset is the Toloka Aggregation Features[8] collected by Yandex.Toloka. In the dataset, the random agents had to classify the websites into

$m=5$ classes by the presence of adult content. The original dataset contains 60 572 opinions of 836 agents about 1052 websites. Also, it includes correct classification γ^* created by the authoritative experts in culturology.

For verification, from this dataset were chosen the records that form a complete table with $n=51$ items and $l=61$ agents, and the number of classes is $m=5$.

In order to check the correctness of the algorithm, it was applied with different percent of similarities. Dependence of the percent of the correct classifications on the chosen similarities is shown in Fig. 2. The meaning of the variables in Fig. 2 is the same as in Fig. 1.

It is seen that starting from the threshold similarity $s(\gamma,\gamma_0)=0.8$, the second plurality voting improves the resulting classification. The percent of correct classifications is up to 6% higher than that in single plurality voting.

The second dataset is the Bluebird dataset[9]. In the dataset, the agents had to recognize bluebird in the images. The dataset contains the opinions of $l=39$ agents about $n=108$ items and $m=2$ classes, and the verification was conducted using all records of this dataset.

The dependence of the percent of the correct classifications on the chosen similarities is shown in Fig. 3. The meaning of the variables is the same as in Figs. 1 and 2.

In this dataset, improvement of the resulting classification starts from the threshold similarity $s(\gamma,\gamma_0)=0.72$. The maximal percent of correct classifications is by 7% higher than that in single plurality voting.

The results obtained in both datasets demonstrate that if in the second voting participate the agents whose classifications γ are more than 80% closer to the classification γ_0, then the suggested algorithm provides higher percent (up to 6%–7%) of correct classifications than the single plurality voting. Such result is compatible and even better than the results demonstrated by the algorithm of collaborative classification[12] which is one of the best heuristic algorithms for such tasks.

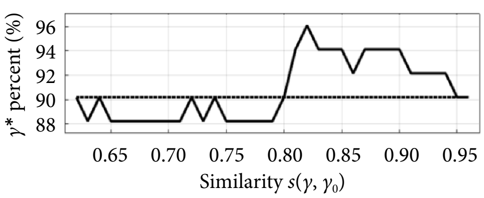

Fig. 2 Percent of correct classifications with respect to the similarity $s(\gamma,\gamma_0)$ between the agents' classifications γ and classification γ_0 by plurality voting in the Toloka Aggregation Features dataset. The dashed line shows the percent of correct classifications by plurality voting.

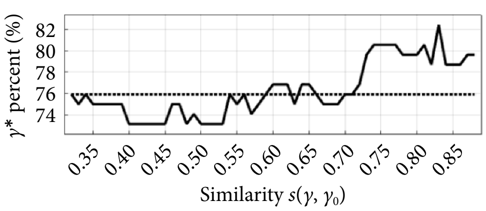

Fig. 3 Percent of correct classifications with respect to the similarity $s(\gamma,\gamma_0)$ between the agents' classifications γ and classification γ_0 by plurality voting in the Bluebird dataset. The dashed line shows the percent of correct classifications by plurality voting.

4 Conclusion

In the paper we presented the simple algorithm for unsupervised classification of given items by an arbitrary group of competent and non-competent agents.

The algorithm is inspired by the combinatorial stability matching method and implements the basic techniques of plurality voting. Following this approach, the items are associated with the classes by recurrent plurality voting. In the trials the algorithm included two stages of the classification by plurality voting.

The algorithm was verified on the simulated and the real-world datasets and resulted in the classifications, which are closer to correct classification than the classification provided by single plurality voting.

The algorithm can be used both in the classification tasks, which require fast and computationally light solutions, and as an initial stage of the Dawid-Skene algorithm and its successors.

Dates

Received: 15 October 2022; Revised: 12 December 2022; Accepted: 30 December 2022

References

[1] L. Duan, S. Oyama, H. Sato, and M. Kurihara, Separate or joint? Estimation of multiple labels from crowdsourced annotations, *Expert Systems with Applications*, vol. 41, no. 13, pp. 5723–5732, 2014.

[2] C. -M. Chiu, T. -P. Liang, and E. Turban, What can crowdsourcing do for decision support, *Decision Support Systems*, vol. 65, pp. 40–49, 2014.

[3] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, and L. Moy, Learning from crowds, *J. Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.

[4] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan, Whose vote should count more: Optimal integration of labels from labelers of unknown expertise, in *Proc. 22nd International Conf. Neural Information Processing Systems*, Vancouver, Canada, 2009, pp. 2035–2043.

[5] A. P. Dawid and A. M. Skene, Maximum likelihood estimation of observer error-rates using the EM algorithm, *J. Royal Stat. Soc. Series C*, vol. 28, no. 1, pp. 20–28, 1979.

[6] V. B. Sinha, S. Rao, and V. N. Balasubramanian, Fast Dawid-Skene: A fast vote aggregation scheme for sentiment classification, presented at 7th KDD Workshop on Issues of Sentiment Discovery and Opinion Mining (KDD WISDOM 2018), London, UK, 2018.

[7] D. Gale and L. S. Shapley, College admissions and the stability of marriage, *American Mathematical Monthly*, vol. 120, no. 5, pp. 386–391, 2013.

[8] Toloka aggregation relevance 5, https://research.yandex.com/datasets/toloka, 2022.

[9] P. Welinder, S. Branson, S. Belongie, and P. Perona, The multidimensional wisdom of crowds, in *Proc. 23rd International Conf. on Neural Information Processing Systems*, Vancouver, Canada, 2010, pp. 2424–2432.

[10] F. Galton, One vote, one value, *Nature*, vol. 75, p. 414, 1907.

[11] F. Galton, Vox populi, *Nature*, vol. 75, pp. 450–451, 1907.

[12] A. Ghanaiem, Unsupervised collaborative classification under uncertainty, MSc dissertation, Department of Industrial Engineering, Tel-Aviv University, Tel Aviv, Israel, 2020.

[13] N. Ratner, E. Kagan, P. Kumar, and I. Ben-Gal, Unsupervised classification for uncertain varying responces: The wisdom-in-the-crowd (WICRO) algorithm, *Knowledge-Based Systems*, https://doi.org/10.1016/j.knosys.2023.110551, 2023.