First published in: Service-oriented computing : ICSOC 2010, San Francisco, CA, Dec. 7-10, 2010. Ed.: P. Maglio.
Berlin : Springer, 2010. S. 243 - 257. ISBN 978-3-642-17357-8

Statistical Quality Control for Human-based Electronic Services

Robert Kern, Hans Thies, Gerhard Satzger
Karlsruhe Institute of Technology (KIT)

Abstract

Crowdsourcing in form of human-based electronic services (people services) provides a powerful way of outsourcing tasks to a large crowd of remote workers over the Internet. Research has shown that multiple redundant results delivered by different workers can be aggregated in order to achieve a reliable result. However, existing implementations of this approach are rather inefficient as they multiply the effort for task execution and are not able to guarantee a certain quality level. As a starting point towards an integrated approach for quality management of people services we have developed a quality management model that combines elements of statistical quality control (SQC) with group decision theory. The contributions of the workers are tracked and weighted individually in order to minimize the quality management effort while guaranteing a well-defined level of overall result quality. A quantitative analysis of the approach based on an optical character recognition (OCR) scenario confirms the efficiency and reach of the approach.

Keywords: Crowdsourcing, human computation, statistical quality control, weighted majority vote

1 Introduction

The idea of human-based electronic services is that they look like Web services but they are not performed by a computer, instead they use human workforce out of a crowd of Internet users. The success of Amazon's Mechanical Turk[1] (MTurk) platform and the growing number of companies that build their business model entirely on that platform demonstrate the potential of this approach. The MTurk platform acts as a broker between requesters who publish human intelligence tasks (HITs) and workers who work on those tasks in return for a typically small monetary compensation. Kern et al. proposed the term people services (pServices) for this type of human-based electronic services [10].

As there is limited control over the individual contributors, particular attention has to be paid to the quality of the work results.One quality assurance

[1] `www.mturk.com`

approach that is heavily used in practise and that can be applied to a broad set of pServices scenarios is the majority vote (MV) approach which *introduces redundancy by passing the same task to multiple workers and aggregating the results in order to compute the result with the highest probability for correctness* [9]. Existing applications of this approach typically apply a fixed level of redundancy to each individual task, i.e. each task is performed by multiple workers. From the perspective of quality management that means that the quality of each individual task is validated. However, the concepts of statistical quality control (SQC) teach us, that the quality management effort can usually be drastically reduced by taking only samples rather than by performing a full inspection of all individual items. [16]. Moreover, a fixed degree of redundancy is both inefficient and incapable of assuring a certain level of result quality because the level of agreement (and so the expected result quality) varies depending on the error rates of the involved workers. For some tasks, the agreement might be extremely high (e.g. all workers agree on exactly the same result), for others the worker results might be at odds (e.g. half of the workers return result A, while the other half returns B).

In this paper, a quality management (QM) approach for pServices is proposed which improves the traditional MV approach in three ways:

1. It reduces the QM effort in *horizontal* direction by validating only a sample of tasks rather than all tasks.
2. It reduces the QM effort in *vertical* direction by dynamically adjusting the level of redundancy rather than working with a fixed level of redundancy.
3. It allows to guarantee a certain quality level by taking individual worker error rates into account.

Within the multifaceted dimensions of quality, this paper concentrates on the correctness dimension as the ability to return a minimum percentage of results that are free of error [10]. According to Jurans definition of quality as *fitness for use* [8], the paper assumes that the service requester can clearly categorize a task result as correct or incorrect. The level of correctness is determined by a comparison with the ideal result (gold standard) provided by the service requester. After providing some fundamentals of SQC in section 2, the QM approach for pServices is presented in section 3. It has been implemented as a QM component on top of the MTurk platform and it has been evaluated using an optical character recognition (OCR) scenario. The results are provided in section 4. The paper closes with related work and a summary and outlook in sections 5 and 6.

2 Fundamentals

This chapter describes some fundamentals about SQC which are required for the considerations in section 3. Specifically, the paper leverages the concept of *sampling plans*.

2.1 Acceptance Sampling

Acceptance Sampling is the process to decide based on a sample whether a set of units meets certain quality requirements or not. Acceptance sampling determines the probability of a lot of units being within the specified quality levels, and accepts or rejects lots based on its quality characteristics. A sampling plan is a procedure where a sample of n units is drawn from a lot of size N. If the number of defects in the sample is higher than the *acceptance number* c, the lot is rejected. Otherwise it is accepted. If the units do not occur in batches, but in a continuous production, such as in line assembly or in a service scenario, the process has to be decomposed into artificial batches. However, before a whole batch has been handled, quality levels for this batch cannot be guaranteed and the results of this batch cannot be further processed. In order to overcome this restriction, *continuous sampling plans* have been developed.

2.2 Continuous Sampling Plans

Continuous Sampling Plans (CSPs) control the inspection frequency and replacement of defects in such a way that a certain *average outgoing quality limit* (AOQL) is not exceeded. Dodge developed the first continuous sampling plan, the CSP-1. This plan has been further developed and adapted by Dodge et. al and Lieberman et al. amongst others [4, 12]. The most celebrated and most used continuous sampling plan still is the CSP-1. The reason is not only its relative simplicity, but also its efficiency, which in few cases is exceeded by other continuous sampling plans like the CSP-2 [5]. Dodge made the following assumptions developing the CSP-1:

1. The process of incoming units is under statistical control and follows a Bernoulli distribution.
2. Sample inspection is perfect.
3. Defective units are replaced by good ones.

The sampling plan is designed for attributes, thus quality parameters are categorized as either good or defective. This means that if the incoming process is under statistical control i.e. the incoming fraction defective p does not change over time, the process can be described by a Bernoulli process with defect probability p. As illustrated by figure 1, the sampling plan starts with 100% inspection. If i consecutive units are found free of defects, only a fraction f of the units are inspected. If a unit is found to be defective, the model returns to 100% inspection and the process starts from the beginning. Defective units are either reworked or replaced with good ones [16]. Important characteristics of the CSP-1 are the *average fraction inspected* (AFI), the *average outgoing quality* (AOQ) and the *average outgoing quality limit* (AOQL) [16].

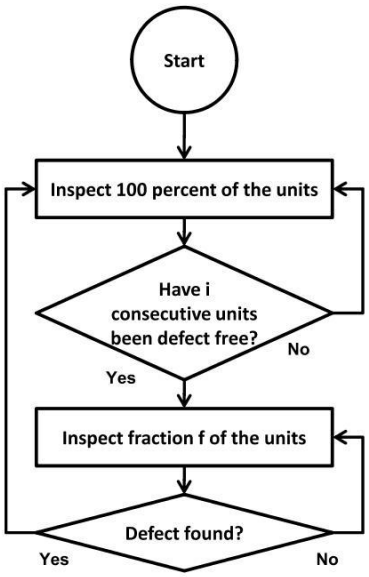


Figure 1: Procedure of the continuous sampling plan CSP-1

The average fraction inspected (AFI) depends on the parameters i and f and on the incoming fraction defective p:

$$AFI(p|i;f) = \frac{1}{1 + (\frac{1}{f} - 1)(1-p)^i} \tag{1}$$

The average outgoing quality is equal to the average amount of defective units passing through without being inspected.

$$AOQ(p|i;f) = \frac{(\frac{1}{f} - 1)p(1-p)^i}{1 + (\frac{1}{f} - 1)(1-p)^i} \tag{2}$$

The AOQ depends on the incoming fraction defective p. It is monotonically increasing with p until reaching its maximum AOQL at p_M. For values higher than p_M, AOQ is monotonically decreasing because CSP-1 is moving more and more to full inspection and is thus detecting and replacing more of the defective items. AOQL is called the *average outgoing quality limit*, it is the worst (highest) value of AQL that can be reached depending on the incoming fraction defective p. AOQL can be determined as:

$$AOQL = \frac{(i+1)p_M - 1}{i} \tag{3}$$

There are multiple combinations of i and f which result in the same value of AOQL. In order to guarantee the average outgoing quality limit AOQL with minimum inspection effort, i and f must be determined in such a way that AFI

is minimized. The optimal selection of i and f depends on the scenario, e.g. on the overall number of units (run length). Several increments of the CSP-1 have been provided in order to adapt it to different scenarios. Two of them are outlined in the following:

2.3 Imperfect inspection

In case of imperfect inspection, two major inspection errors can be made:

- E_1: a good item can be classified as defective, also referred to as a *type 1 inspection error*.
- E_2: a defective item can be classified as good, also referred to as a *type 2 inspection error*.

In the following, A refers to the event that an item is defective. The probability of the event that an item is classified as defective (B) can be calculated as:

$$P(B) = P(A) * P(\neg E_2) + P(\neg A) * P(E_1) \tag{4}$$

Wang and Chen have presented a model to calculate a minimal AFI under the assumption of imperfect inspection [24]. According to them, under the assumption that the optimal value for $i = i^*$ is already known, an optimal value for f^* can be calculated by

$$f^* = \frac{(1 - P(B))^{i^*}(1 - \frac{AOQL}{\hat{p}})}{((1 - P(B))^{i*} - 1)(1 - \frac{AOQL}{\hat{p}}) + (1 - P(E_2))} \tag{5}$$

where AOQL is the specified value for the average outgoing quality limit and $\hat{p}$ is the incoming fraction defective.

2.4 Short production runs

Blackwell developed a Markov-chain model for the CSP-1 under short production runs [1]. McShane and Turnbull extended his model to compute probability limits on outgoing quality [15]. Although computationally expensive, their model can be used to determine a CSP-1 with minimal inspection by iteratively increasing i, determining the smallest value of f that meets the AOQL, and finally calculating the AFI. The details model go beyond the scope of this paper and can be found in [15].

3 Statistical Quality Control for People Services

3.1 Assumptions

Because of the nature of pServices as *Web based software services that deliver human intelligence, perception, or action to customers as massively scalable*

resources [10], it is obvious that pServices require some kind of Web platform. Figure 2 gives an overview of the basic pService scenario which comprises three roles: the pService *requester*, the *pService platform* and the *workers* who belong to a worker pool. The pService platform acts as a mediator between the pService requester who publishes pService tasks and pService workers who select tasks and work on it in return for a typically small compensation. The paper makes some additional assumptions about the underlying pService platform:

1. It allows for tracking individual workers based on an individual *worker ID*

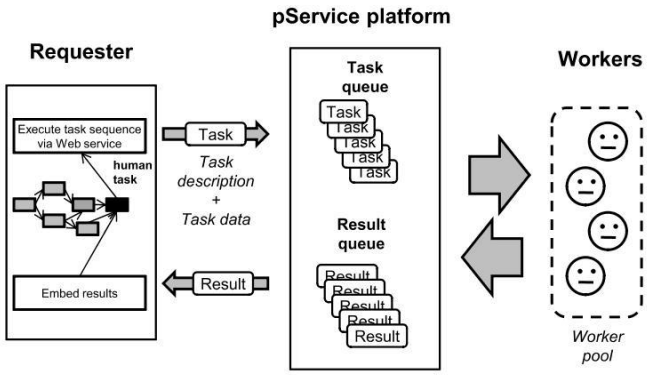


Figure 2: Scenario of basic pService platform

It is further assumed that there is a large number of equivalent tasks which consist of the same *task description* but different *task data*. The task description primarily contains the instructions for the workers how to perform the task as well as information about the expected result quality. The task data is the variable part which might represent different pictures to be annotated, different addresses to be validated or different products to be classified. A *task instance* represents a task for an individual item of the task data, e.g. for an individual picture to be annotated.

3.2 Acceptance Sampling for pServices

The objective of the model described in this paper is to leverage acceptance sampling in order to ensure that pService results are delivered within a certain average outgoing quality limit AOQL, while the inspection costs in terms of labor work are minimized. The model can be seen as a quality management (QM) component on top of the basic pService platform described in the previous section. The overall scenario is given by figure 3. The model assumes that for a given task type there is an individual error rate p_x for each worker x (A). This

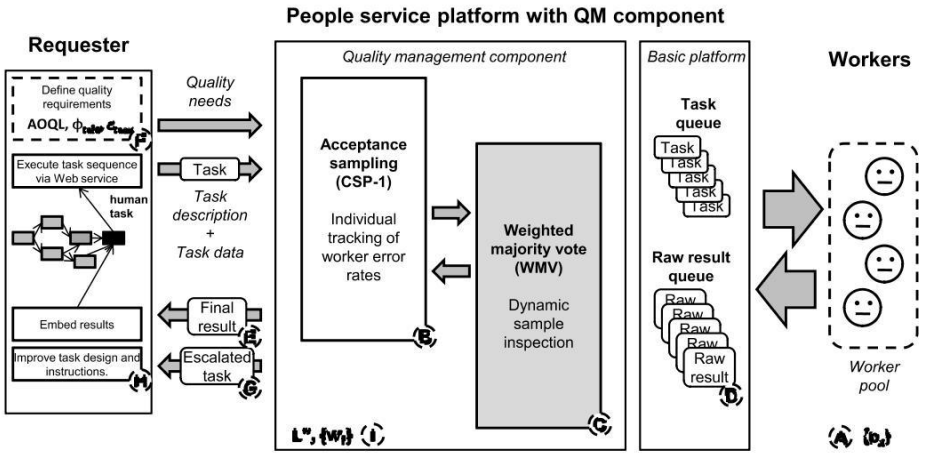


Figure 3: Schematic overview of pService platform with QM component

error rates of individual workers are independent from each other, the sampling has to be done at worker-level. The results are aggregated, and the same AOQL is applied to all workers that work on this task type i.e. the same quality of work results is requested from all participating workers. The QM component uses continuous acceptance sampling in order to guarantee a certain long-run average outgoing quality limit AOQL defined by the service requester.

The QM component consists of two functional parts: an acceptance sampling component (B) and a sample inspection component (C). The acceptance sampling component leverages the basic continuous sampling plan (CSP-1) with the increment of imperfect inspection and replacement and the increment of limited runtimes as presented in section 2. The CSP-1 leverages continuous sampling of *raw results* (D) delivered by the workers and turns them into *final results* (E) in order to guarantee an average outgoing quality limit AOQL that is defined by the requester along with other quality requirements (F).

The CSP-1 requires a mechanism for sample inspection. For this purpose, a *weighted majority vote approach* (WMV) was designed which will be described in detail in section 3.4. The WMV dynamically increases the redundancy by including additional workers in the MV decision until a predefined significance φ_{min} is reached. Because the inspection process performed by the WMV is not perfect but only meets a quality level of φ_{min}, Case et al.'s model for CSP-1 with imperfect inspection is utilized in combination with Wang and Chen's increment. As some tasks may not conform to the specifications of that task type, e.g. they are harder to solve than the others or the task description does not apply to all individual tasks, they are escalated back to the requester (G) if a predefined *escalation limit* ε_{max} (F) is reached. That way, he can use this information to improve task design and provide the correct results himself (H). As we assume a

fixed payment per task, the QM costs can be minimized by minimizing the total number of tasks. Because the WMV (as well as the traditional MV) appraoch assumes that the raw results delivered by multiple workers can be compared to each other or aggregated into a consolidated result, the mechanism works only for *deterministic tasks* i.e. for tasks that have a certain well-defined optimal result [9].

Additional parameters are administrated by the platform itself (I): The Markov chain CSP-1 model developed by McShane and Turnbull (see section 2) is used to take into account that some workers may contribute only few results. It determines a starting value of i, considering the expected run length L^*. L^* specifies the expected *run length* of a process per worker, that is the average amount of tasks of the same task type each worker will work on.

The CSP-1 is implemented using an inspection status w_x for each worker. The initial value will be $w_x = i$ which will be reduced by 1 for each consecutive result that the worker has been submitted and that has been classified as correct. If $w_x = 0$, only fractional inspection will take place. Once, a result submitted by the worker is classified as incorrect, his inspection status will be reset to $w_x = i$.

The worker error rate p_x describes the expected error rate of worker x, anticipated from historical values. Due to the nature of human work, p_x should never completely reach 0.

3.3 Worker Pool Management.

A worker who constantly stays in full inspection mode leads to high costs, so depending on the availability of workers and the costs for inspection, a decision has to be made as to which workers are not profitable and should be removed from the worker pool. Therefore, the maximum error rate e has been introduced. If a worker's error rate exceeds the maximum error rate $p_x > e$, he may not participate.

3.4 Sample Inspection Process - The weighted majority vote (WMV) approach

The weighted majority vote (WMV) is used for sample inspection. All raw results that have to be inspected according to the CSP-1 for the respective worker, are validated by passing redundant task assignments to other workers in order to be able to come to a group decision which meets a minimum inspection quality level φ_{min}. The process of the WMV is explained based on figure 4.

The basic idea is to publish one additional (redundant) task assignment (2), retrieve the result (3) and calculate based on his individual error rate whether the required minimum inspection quality φ_{min} has already been met (4). If this is the case, the final result is returned (4). If the required quality has not yet been met, it is checked in step (5), whether a quality improvement can be expected by adding more workers. If that is not the case, the task is escalated back to the requester. Otherwise, the process continues with step (2) where

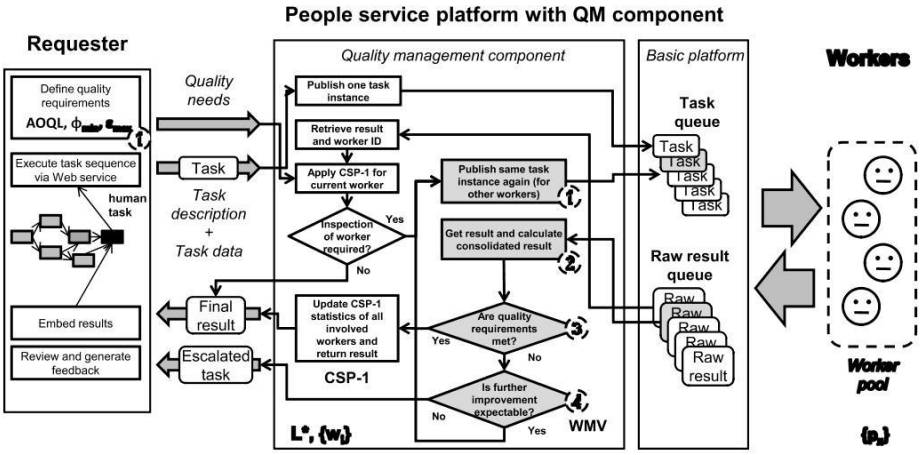


Figure 4: Detailed overview of pService platform with QM component

another redundant task assignment is published. The process is continued until either the

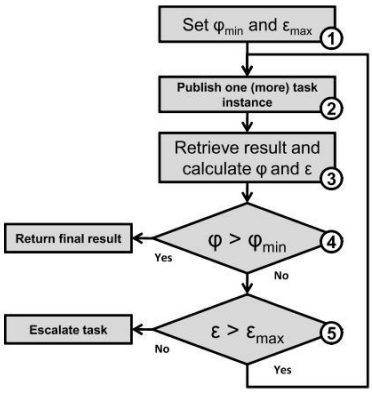


Figure 5: The weighted majority vote (WMV) approach

WMV. Assuming each worker x has an individual failure rate p_x when working on task y and returns raw result r_{xy}, the process is the following:

1. Specify desired level of inspection quality φ_{min} and escalation limit ε_{max}.
2. Make one (more) redundant assignment for task y available to the workers.
3. Retrieve the worker result r_{xy} and identify the result with the highest probability of correctness ε as well as the actual escalation limit ε.

4. If the φ exceeds the desired level inspection quality φ_{min}, return the result r_c with the highest probability of correctness and update the qualification values q_x of all participating workers, where $q_x = 1 - p_x$.

5. Escalate the task back to the requester if the overall probability ε for getting a result set R_y is lower or equal than the escalation limit ε_{max}, with $R_y = \{r(x_1), r(x_2), ..r(x_k)\}$, $x_1, x_2..x_k$ being the IDs of the workers who have worked on the task and k being the number of assignments for the task.

Steps 2 to 5 are repeated until the final result is returned in step 4 or the task is escalated in step 5. In step 3, the values φ and ε are calculated using equations 6, 7 and 8. Equation 6 determines the Bayes-conditional likelihood for result r_c being correct under the condition that the result set R_y was received.

$$\varphi_c = P(r_c \, is \, correct | R = R_y) = \frac{P(r_c \, is \, correct \cap R = R_y)}{P(R = R_y)} \tag{6}$$

$$= \frac{\prod_{\forall r_i = r_c} r_c q_i \prod_{\forall r_i \neq r_c} p_i}{\sum_{j=1}^{k} \prod_{\forall r_i = r_j} q_i \prod_{\forall r_i \neq r_j} p_i + \prod_{j=1}^{N} p_j} \tag{7}$$

$$\varepsilon_y = P(R = R_y) = (\sum_{j=1}^{k} \prod_{\forall r_i = r_j} q_i \prod_{\forall r_i \neq r_j} p_i) + \prod_{j=1}^{N} p_j \tag{8}$$

4 Evaluation

4.1 Experimental design

The QM approach has been implemented as a QM component on top of MTurk, accessing the platform through the SOAP interface available to service requesters. An *optical character recognition* (OCR) scenario was used for evaluation, which consists of a dataset of 1176 handwritten words. In each of the tasks, a worker was asked to type in a single handwritten word which was displayed as an image file (JPEG). The expected optimal result (gold standard) was specified by the author of the handwriting himself. On February 1st, 10 instances (assignments) of each task were uploaded to the MTurk platform. It was prohibited that a worker handles the exact same task more than once. The task payment was \$0.01 per task, with Amazon receiving a service charge of \$0.005 for each task. Consequently a total amount of $1,176 \times 10 = 11,760$ data sets has been collected during the evaluation leading to total expenses of $11,760(\$0.01 + \$0.005) = \$176.40$. The QM mechanism was simulated on the raw results in order to be able to run multiple simulations at different parameters and in order to have a baseline for comparing with the performance of the traditional MV mechanism.

4.2 Qualification testing

The MTurk platform provides means for limiting the access to tasks to those workers who have successfully completed a so called *qualification test*. Such a test can be designed individually for each type of task. The QM approach described in this paper implicitly determines the error rates of the workers, therefore there is typically no need to restrict the participation to those who have passed a qualification test. However, as the actual test was only simulated on a fixed number of instances (assignments) of each task, a qualification test was used to reduce the overall cost of the experiment as it excludes spammers and workers who submit bad quality right from the start. The test consisted of a series of 10 simple OCR tasks (10 words). All of them had to be typed in correctly in order to pass the test.

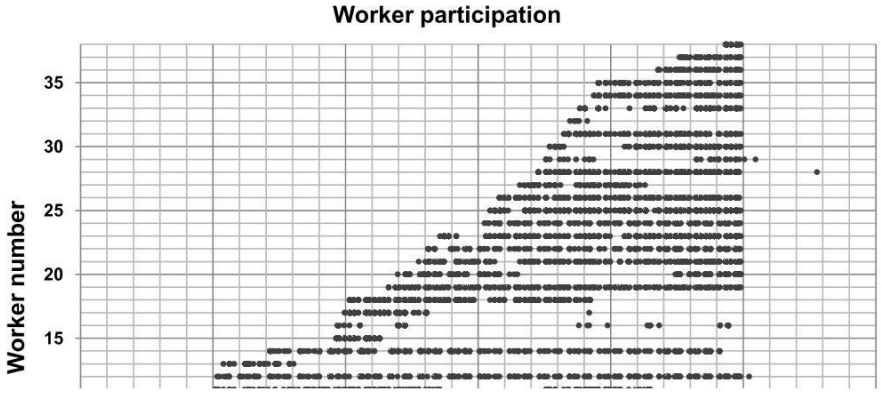
4.3 Execution performance

Probably the most astonishing result of the experiment was the speed with which the results were submitted. In the first pre-tests, a batch of 3,528 tasks was completed by 112 workers in less than 15 minutes at an execution rate of 14,088 tasks per hour. During other experiments we even observed total execution speeds up to 3 times as fast, because of more workers participating. We assume that the execution speed besides the payment also depends on the time of day, since most workers are U.S.- or Indian citizens [19]. Figure 6 illustrates the execution of the actual experiment in which 11,760 tasks have been processed by 36 workers in about 2:40 hours. One can observe how workers successively join the process. A similar chart is used by the crowdsourcing provider crowdflower.com.

4.4 Full inspection

The first simulation was a full inspection by running the WMV for all tasks. The CSP-1 was not used in this experiment. Running only the WMV leads to remarkably good quality. The inspection quality goal of 0.99 was almost perfectly met. Figure 7 shows the results of WMV compared to the traditional MV approach. The traditional MV was simulated based on the same data as the WMV by averaging all possible combinations of 2 to 9 answers within each set of 10 available answers per task for the two-fold up to the 9-fold MV. For each combination, the most occurring answer was chosen. If several answers occur the same amount of times (tie), a random choice between the answers occurring most was made, as suggested by Snow et al. [21].

We see that our WMV (98.36%) even outperforms the accuracy of a ninefold traditional MV (97.76%). That is a remarkable result given that the WMV is 4 times more efficient as it requires only 2.25 workers per task compared to 9 workers per task for the basic ninefold MV approach. In other words: the WMV approach has reduced the quality management effort by some 75 percent compared to the traditional MV approach. Figure 8 illustrates this relation.



Approach	MV 2	WMV	MV 3	MV 4	MV 5	MV 6	MV 7	MV 8	MV 9	MV 10
Average redundancy	2	2.25	3	4	5	6	7	8	9	10
Accuracy	0.927	0.984	0.954	0.958	0.962	0.974	0.975	0.977	0.978	0.977

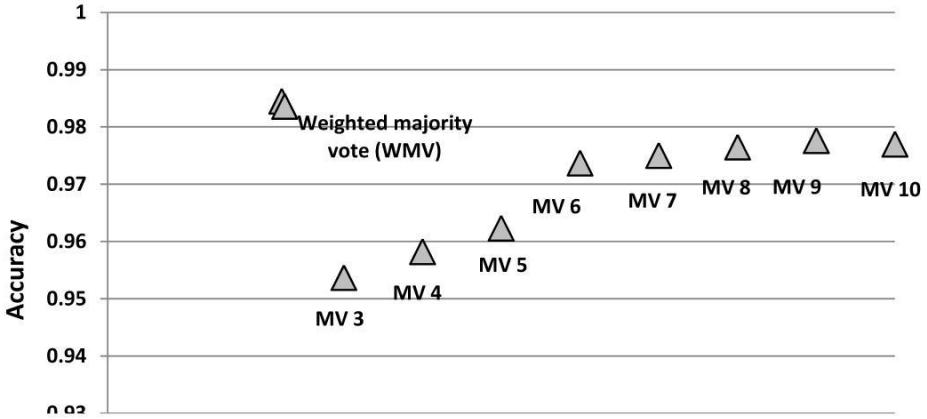
Figure 7: Comparison of the accuracy of different majority vote approaches

4.5 Acceptance Sampling

In a series of tests, the QM approach was used with CSP-1 for 3 different quality goals i.e. three different values of AOQL. Figure 9 shows the results of 10 simulation runs with an AOQL of 0.05:

- $AOQL = 0.05$; $i = 6$; $f = 0.249$; $\varphi_{min} = 0.99$; $\varepsilon_{max} = 0.01$

A total of 1.52 assignments per HIT was observed on average, which is a significant improvement even compared to the 100%-inspection with 2.25 assignments per HIT. A number of 1.91 percent of the HITs are escalated. Some 39 percent of all tasks are inspected. Figure 10 illustrates the decrease of the inspection rate over time, therefore a smaller inspection rate can be expected in the long run. The AOQL value was achieved in 6 out of the 10 cases. It is not surprising that in some runs the quality is slightly worse than the specified AOQL because of the short run time of only 1176 tasks. When averaging over several runs, we obtain a reliable outgoing fraction of 0.0491, which can be considered optimal as the goal is to minimize the QM effort rather than to overachieve the quality objective.



Simulation #	1	2	3	4	5	6	7	8	9	10	Average	Average per HIT
Run length	1176	1176	1176	1176	1176	1176	1176	1176	1176	1176	1176.0	1.0000
Assignments	1783	1744	1830	1823	1741	1797	1793	1836	1782	1762	1789.1	1.5213
Escalated	20	25	23	25	21	24	16	26	22	23	22.5	0.0191
Inspected	439	427	469	475	433	471	489	482	453	458	459.6	0.3908
Incoming failures	84	88	90	92	85	91	87	95	88	76	87.6	0.0745
Outgoing failures	59	62	48	55	67	58	64	58	56	50	57.7	0.0491

Figure 9: Results of the acceptance sampling test for 10 simulations with AOQL=0.05

We further tested the quality model with different AOQL levels (Figure 9):

- $AOQL = 0.025$; $i = 5$; $f = 0.582$; $\varphi_{min} = 0.99$; $\varepsilon_{max} = 0.01$
- $AOQL = 0.075$; $i = 1$; $f = 0.039$; $\varphi_{min} = 0.99$; $\varepsilon_{max} = 0.01$

For AOQL=0.075 the quality is again precisely met. However, when increasing the quality demands to AOQL=0.025, the model does not manage to achieve the desired level anymore. The reason for that lies in the gap between the gold standard and the majority decision of the workers: In several cases, the majority of the workers identified a certain word (e.g. "five") even if the writer (who represented the gold standard) had written a different word (e.g. "fine").

5 Related Work

The concept of majority vote is widely used in the context of pServices. Redundant task execution is a basic feature for quality improvement provided by

Average per HIT	AOQL=0.025	AOQL=0.05	AOQL=0.075
Run length	1.0000	1.0000	1.0000
Assignments	1.7954	1.5213	1.0930
Escalated	0.0404	0.0191	0.0032
Inspected	0.6331	0.3908	0.0601
Incoming failures	0.0723	0.0745	0.0745
Outgoing failures	0.0287	0.0491	0.0710

Figure 11: Results of the acceptance sampling tests for different values of AOQL

platforms like MTurk. Sorokin and Forsyth as well as Snow et al. have analyzed the effect of the approach based on annotation scenarios [22, 21]. Snow et al. have investigated how many non-experts out of the crowd are needed in order to achieve better results than one expert. Depending on the scenario, they report a required number of non-experts between two and more than ten. Whitehill et al. consider how to integrate labeler's expertise into a majority vote mechanism for image labeling [25]. They propose a probabilistic model and use it to simultaneously infer the label of each image, the expertise of each labeler, and the difficulty of each image. Complementary approaches for quality management of pServices include iterative work processes [13], review processes [9] and the injection of gold standard tasks [22]. A maximum likelihood estimation can be used to estimate worker error rates as well as the correct categories of the task results [7, 9]. The approach leverages the EM algorithm dating back to Dawid and Skene [3]. Raykar et al. propose a specific form of an EM algorithm which is capable of generating a gold standard [17].

The validity of the majority vote model has been first mathematically proven by Condorcet's Jury Theorem [2]. Under the assumption that one of two outcomes is correct and each decision maker has the independent probability $p > 0.5$ to make the right decision, the probability for a correct group decision is greater

than the individual one. Latif-Shabgahi et al. have examined and classified a large number of software voting algorithms used in safety-critical systems [11]. Surowiecki illustrated that the aggregation of group responses may lead to better results than the information of any single group member - if the opinions are diverse, independent, decentralized, and an appropriate aggregation mechanism exists [23]. This phenomenon has been described as the wisdom of the crowds. Typical applications that leverage crowd intelligence are prediction markets [6], Delphi methods [20] and extensions of the traditional opinion poll. In the field of machine learning, Littlestone and Warmuth developed a weighted majority algorithm, that acts as a "master algorithm" and aggregates the answers of several prediction algorithms in order to determine the best prediction possible [14]. The aggregation mechanism is a vital part of each majority vote model. Revow et al. compare five combination strategies (majority vote, Bayesian, logistic regression, fuzzy integral, and neural network) and arrive at the conclusion that majority vote is as effective as the other, more complicated schemes to improve the recognition rate for the data set used [18].

6 Conclusion and Future Work

We have presented a statistical model for managing the correctness of human-based electronic services (people services) which leverages continuous acceptance sampling and group decision theory. The mechanism consists of two parts: The continuous acceptance sampling plan CSP-1 is used to track the contributions of each worker individually based on samples taken from their work results. A *weighted majority vote* (WMV) approach was introduced for the inspection of the samples which leverages a group decision of multiple workers. The number of workers participating in that group decision is adjusted dynamically depending on their individual error rates. By validating only a fraction of the tasks and keeping the validation effort per task at a minimum, the model is capable of guaranteeing a certain predefined level of result quality at minimum costs. An evaluation on Amazon's Mechanical Turk platform has shown a reduction of the quality management effort of up to 75 percent compared to existing approaches.

In our ongoing research we are expanding the scope of our QM mechanism to other aspects of quality like performance and availability. Furthermore, we are investigating the effect of worker feedback on the result quality.

References

[1] Blackwell, M.: The effect of short production runs on CSP-1. Technometrics 19(3), 259–263 (1977)

[2] le marquis de Condorcet, M., Caritat, A.N.: Essai sur l'application de l'analyse la probabilit des dcisions rendues la pluralit des voix (1785)

[3] Dawid, A., Skene, A.: Maximum likelihood estimation of observer Error-Rates using the EM algorithm. Journal of the Royal Statistical Society 28(1), 20–28 (1979)

[4] Dodge, H., Torrey, M.: Additional continuous sampling inspection plans. Industrial Quality Control (7), 7–12 (1951)

[5] Gosh, D.T.: An optimum continuous sampling plan CSP-2 with k i to minimise the amount of inspection when incoming quality p follows a distribution. The Indian Journal of Statistics 58(1), 105–117 (1996)

[6] Gruca, T.S., Berg, J.E., Cipriano, M.: Consensus and differences of opinion in electronic prediction markets. Electronic Markets 15(1), 13–22 (2005)

[7] Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on amazon mechanical turk (2010)

[8] Juran, J., Godfrey, A.: Juran's Quality Handbook. McGraw-Hill, New York, NY, USA, 5th edition edn. (2000)

[9] Kern, R., Bauer, C., Thies, H., Satzger, G.: Validating results of human-based electronic services leveraging multiple reviewers. In: Proceedings of the 16th Americas Conference on Information Systems (AMCIS). Lima, Peru (2010), (forthcoming)

[10] Kern, R., Zirpins, C., Agarwal, S.: Managing quality of Human-Based eServices. In: Feuerlicht, G., Lamersdorf, W. (eds.) Service-Oriented Computing - ICSOC 2008 Workshops, ICSOC 2008 International Workshops, Sydney, Australia, December 1st, 2008, Revised Selected Papers. Lecture Notes in Computer Science, vol. LNCS 5472, pp. 304–309. Springer (2009)

[11] Latif-Shabgahi, G., Bass, J.M., Bennett, S.: A taxonomy for software voting algorithms used in safety-critical systems. IEEE Transactions on Reliability 53(3), 319 (2004)

[12] Lieberman, G.J., Solomon, H.: Multi-Level continuous sampling plans. The Annals of Mathematical Statistics 26(4), 686–704 (1955)

[13] Little, G., Chilton, L.B., Goldman, M., Miller, R.C.: Turkit: Tools for iterative tasks on mechanical turk. In: Proceedings of the ACM SIGKDD Workshop on Human Computation. pp. 29–30 (2009)

[14] Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. Information and Comutation 108, 212–261 (1994)

[15] McShane, L.M., Turnbull, B.W.: Probability limits on outgoing quality for continuous sampling plans. Technometrics 33(4), 393–404 (1991)

[16] Montgomery, D.: Introduction to statistical quality control. Wiley & Sons, New York, NY, USA, 6th edition edn. (2008)

[17] Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. Journal of Machine Learning Research 11, 1297–1322 (2010)

[18] Revow, M., Williams, C.K.I., Hinton, G.E.: Using generative models for handwritten digit recognition. IEEE Trans. Pattern Anal. Mach. Intell. 18(6), 592–606 (1996)

[19] Ross, J., Irani, L., Silberman, M., Zaldivar, A., Tomlinson, B.: Who are the crowdworkers?: shifting demographics in mechanical turk. In: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems. pp. 2863–2872 (2010)

[20] Rowe, G., Wright, G.: The delphi technique as a forecasting tool: issues and analysis. International Journal of Forecasting 15(4), 353–375 (Oct 1999)

[21] Snow, R., OConnor, B., Jurafsky, D., Ng, A.Y.: Cheap and fastbut is it good? evaluating non-expert annotations for natural language tasks. In: EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 254–263. ACL, Stroudsburg, USA (2008)

[22] Sorokin, A., Forsyth, D.: Utility data annotation with amazon mechanical turk. In: CVPRW '08: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops. pp. 1–8. IEEE Computer Society, Washington, WA, USA (Jun 2008)

[23] Surowiecki, J.: The Wisdom of Crowds. Doubleday, New York, NY, USA, 1st edition edn. (2004)

[24] Wang, R., Chen, C.: Minimum average fraction inspected for continuous sampling plan CSP-1 under inspection error. Journal of Applied Statistics 24(5), 539–548 (Oct 1997)

[25] Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Advances in Neural Information Processing Systems 22. pp. 2035–2043 (2009)