A General-Purpose Crowdsourcing
Computational Quality Control Toolkit for Python

Dmitry Ustalov,[1] **Nikita Pavlichenko,**[2] **Vladimir Losev,**[2] **Iulian Giliazev,**[3*] **Evgeny Tulin**[2]

[1] Yandex, Piskariovski Prospekt, building 2, block 2, Benois Business Centre, Saint Petersburg, Russia 195027
[2] Yandex, Ulitsa Lva Tolstogo 16, Moscow, Russia 119021
[3] Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow Region, Russia 141701
{dustalov,pavlichenko,losev,tulinev}@yandex-team.ru, giliazev.iua@phystech.edu

Abstract

Quality control is a crux of crowdsourcing. While most means for quality control are organizational and imply worker selection, golden tasks, and post-acceptance, computational quality control techniques allow parameterizing the whole crowdsourcing process of workers, tasks, and labels, inferring and revealing relationships between them. In this paper, we demonstrate Crowd-Kit, a general-purpose crowdsourcing computational quality control toolkit. It provides efficient implementations in Python of computational quality control algorithms for crowdsourcing, including uncertainty measures and crowd consensus methods. We focus on aggregation methods for all the major annotation tasks, from the categorical annotation in which latent label assumption is met to more complex tasks like image and sequence aggregation. We perform an extensive evaluation of our toolkit on several datasets of different nature, enabling benchmarking computational quality control methods in a uniform, systematic, and reproducible way using the same codebase. We release our code and data under an open-source license at https://github.com/Toloka/crowd-kit.

Introduction

Means for quality control in crowdsourcing include organizational approaches, such as task design, decomposition, golden tasks preparation, yet reliably automated, and computational approaches that employ relationships and statistical properties of workers, tasks, and labels. Many crowdsourcing studies of complex crowdsourcing pipelines aim to reduce their tasks to multi-classification or combine multi-classification with post-acceptance, e.g., in a seminal paper by Bernstein et al. (2010). At the same time, researchers from such fields as natural language processing, computer vision, and others develop discipline-specific methods. To be conveniently employed, these methods need to be integrated with popular data science libraries and frameworks. However, such toolkits as SQUARE (Sheshadri and Lease 2013), CEKA (Zhang et al. 2015), Truth Inference (Zheng et al. 2017), spark-crowd (Rodrigo, Aledo, and Gámez 2019), require additional effort to be embedded in applications. We believe in addressing this issue by developing **Crowd-Kit**, an open-source production-ready Python toolkit for computational quality control in crowdsourcing. It implements popular quality control methods, providing a common ground for reliable experimentation and application. We perform an extensive evaluation of the Crowd-Kit library to provide the common ground for comparisons. *In all the experiments in this paper, we used our implementations of the corresponding methods.*

Crowd-Kit Design and Maintenance

Our fundamental design principle of Crowd-Kit is to bridge the gap between crowd science and vivid data science ecosystem of NumPy, SciPy, Pandas, and scikit-learn (Pedregosa et al. 2011). We implemented Crowd-Kit in Python and employ the highly optimized data structures and algorithms available in these libraries, ensuring compatibility with the application programming interface (API) of scikit-learn and data frames of Pandas.

We implemented all the methods in Crowd-Kit from scratch in Python. Although unlike spark-crowd (Rodrigo, Aledo, and Gámez 2019) our library does not provide means for running on a distributed computational cluster, it leverages efficient implementations of numerical algorithms in underlying libraries widely used in the research community. Besides aggregation methods, Crowd-Kit offers annotation quality characteristics, such as uncertainty (Malinin 2019) and agreement with aggregate (Appen Limited 2021).

Crowd-Kit is platform-agnostic, allowing analyzing data from any crowdsourcing marketplace (as soon as one can download the data). Crowd-Kit is an open-source library available under Apache license both on GitHub and PyPI: https://github.com/Toloka/crowd-kit and https://pypi.org/project/crowd-kit/, correspondingly.

Categorical Aggregation

Crowd-Kit includes aggregation methods for categorical data, in which latent label assumption is met. We implement most traditional methods for categorical answer aggregation, including such models as *Dawid-Skene* (DS, 1979), GLAD (Whitehill et al. 2009), and M-MSR (Ma and Olshevsky 2020). We also offer an implementation of *Majority Vote* (MV) as well as its such weighted variations as *Worker Agreement with Aggregate* (Wawa) as described in Appen Limited (2021).

Method	D_Product	D_PosSent	S_Rel	S_Adult	binary1	binary2
MV	0.897	0.932	0.536	0.763	0.931	0.936
Wawa	0.897	0.951	0.557	0.766	0.981	0.983
DS	0.940	0.960	0.615	0.748	0.994	0.994
GLAD	0.928	0.948	0.511	0.760	0.994	0.994
M-MSR	—	0.937	0.425	0.751	0.994	0.994

Table 1: Comparison of the implemented categorical aggregation methods (accuracy is used).

Evaluation. To ensure the correctness of our implementations, we compared the observed aggregation quality with the already available implementations by Zheng et al. (2017) and Rodrigo, Aledo, and Gámez (2019) on the same datasets. Table 1 shows evaluation results, indicating a similar level of quality as them.

Pairwise Aggregation

Pairwise comparisons are essential for such tasks as information retrieval evaluation and subjective opinion gathering, where the latent label assumption is not met. We implemented the *Bradley-Terry* probabilistic transitivity model (BT, 1952) for pairwise comparisons.

Evaluation. Table 2 shows the comparison of the *Bradley-Terry* method implemented in Crowd-Kit to the random baseline on the graded readability dataset by Chen et al. (2013). Since it contains only 491 items, we additionally annotated on Toloka a sample of 2,497 images from the IMDB-WIKI dataset (Rothe, Timofte, and Van Gool 2018). This dataset contains images of people with reliable ground-truth age assigned to every image. The annotation allowed us to obtain 84,543 comparisons by 2,085 workers.

Method	Chen et al. (2013)	IMDB-WIKI
Bradley-Terry	0.543	0.885
Random	0.360	0.504

Table 2: Comparison of implemented pairwise aggregation methods (NDCG@10 is used for Chen et al. (2013) and NDCG@100 is used for IMDB-WIKI).

Sequence Aggregation

Crowd-Kit implements the *Recognizer Output Voting Error Reduction* (ROVER) dynamic programming algorithm by Fiscus (1997), known for its successful application in crowdsourced sequence aggregation (Marge, Banerjee, and Rudnicky 2010). We also offer implementations of *Reliability Aware Sequence Aggregation* (RASA and HRRASA) algorithms by Li and Fukumoto (2019) and Li (2020) that encode responses using Transformer-based representations and then iteratively estimate the aggregated response embedding.

Evaluation. We used two datasets, CrowdWSA (Li and Fukumoto 2019) and CrowdSpeech (Pavlichenko, Stelmakh, and Ustalov 2021). As the typical application for sequence aggregation in crowdsourcing is audio transcription, we used the word error rate as the quality criterion (Fiscus 1997) in Table 3.

Dataset	Version	ROVER	RASA	HRRASA
CrowdWSA	J1	0.612	0.659	0.676
	T1	0.514	0.483	0.500
	T2	0.524	0.498	0.520
CrowdSpeech	dev-clean	0.676	0.750	0.745
	dev-other	0.132	0.142	0.142
	test-clean	0.729	0.860	0.859
	test-other	0.134	0.157	0.157

Table 3: Comparison of implemented sequence aggregation methods (average word error rate is used).

Image Aggregation

Crowd-Kit offers three image segmentation aggregation methods. First, it provides a trivial pixel-wise MV. Second, it implements a method similar to the one described in Jung-Lin Lee, Das Sarma, and Parameswaran (2018), performing an EM algorithm for counting the probability of a correct answer as the proportion of correctly classified pixels to the number of all pixels that at least one worker chose. Third, we implement a variation of RASA that scores Jaccard distances between segments and weights them proportionally to these distances.

Evaluation. We annotated on Toloka a sample of 2,000 images from the MS COCO (Lin et al. 2014) dataset consisting of four object labels. For each image, nine workers submitted segmentations. In total, we received 18,000 responses. Table 4 shows the comparison of the methods on the above-described dataset using the *intersection over union* (IoU) criterion.

Dataset	MV	EM	RASA
MS COCO	0.839	0.861	0.849

Table 4: Comparison of implemented image aggregation algorithms (IoU is used).

Conclusion

Our experience in running Crowd-Kit for processing crowdsourced data shows that it successfully handles industry-scale datasets without the need for a large computational cluster. We currently focus on providing a consistent API for benchmarking existing methods and implementing additional domain-specific aggregation techniques like sequence labels aggregation (Nguyen et al. 2017) and continuous answer aggregation. We believe that the availability of computational quality control techniques in a standardized way would open new venues for reliable improvement of the crowdsourcing quality beyond the traditional well-known methods and pipelines.

References

Appen Limited. 2021. Calculating Worker Agreement with Aggregate (Wawa). URL https://success.appen.com/hc/en-us/articles/202703205-Calculating-Worker-Agreement-with-Aggregate-Wawa-.

Bernstein, M. S.; Little, G.; Miller, R. C.; Hartmann, B.; Ackerman, M. S.; Karger, D. R.; Crowell, D.; and Panovich, K. 2010. Soylent: A Word Processor with a Crowd Inside. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, 313–322. New York, NY, USA: ACM. ISBN 978-1-4503-0271-5. doi:10.1145/1866029.1866078.

Bradley, R. A.; and Terry, M. E. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika* 39(3/4): 324–345. ISSN 0006-3444. doi:10.2307/2334029.

Chen, X.; Bennett, P. N.; Collins-Thompson, K.; and Horvitz, E. 2013. Pairwise Ranking Aggregation in a Crowdsourced Setting. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, 193–202. Rome, Italy: Association for Computing Machinery. ISBN 9781450318693. doi:10.1145/2433396.2433420.

Dawid, A. P.; and Skene, A. M. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society, Series C (Applied Statistics)* 28(1): 20–28. ISSN 0035-9254. doi:10.2307/2346806.

Fiscus, J. G. 1997. A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, 347–354. Santa Barbara, CA, USA: IEEE. doi:10.1109/ASRU.1997.659110.

Jung-Lin Lee, D.; Das Sarma, A.; and Parameswaran, A. 2018. Quality Evaluation Methods for Crowdsourced Image Segmentation. Technical report, Stanford University. URL http://ilpubs.stanford.edu:8090/1161/.

Li, J. 2020. Crowdsourced Text Sequence Aggregation Based on Hybrid Reliability and Representation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 1761–1764. Virtual Event, China: Association for Computing Machinery. ISBN 9781450380164. doi:10.1145/3397271.3401239.

Li, J.; and Fukumoto, F. 2019. A Dataset of Crowdsourced Word Sequences: Collections and Answer Aggregation for Ground Truth Creation. In *Proceedings of the First Workshop on Aggregating and Analysing Crowdsourced Annotations for NLP*, AnnoNLP '19, 24–28. Hong Kong: Association for Computational Linguistics. doi:10.18653/v1/D19-5904.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, 740–755. Zurich, Switzerland: Springer International Publishing. ISBN 978-3-319-10602-1. doi:10.1007/978-3-319-10602-1_48.

Ma, Q.; and Olshevsky, A. 2020. Adversarial Crowdsourcing Through Robust Rank-One Matrix Completion. In *Advances in Neural Information Processing Systems 33*, 21841–21852. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2020/file/f86890095c957e9b949d11d15f0d0cd5-Paper.pdf.

Malinin, A. 2019. *Uncertainty Estimation in Deep Learning with application to Spoken Language Assessment*. Ph.D. thesis, University of Cambridge, Cambridge, England, UK. doi:10.17863/CAM.45912.

Marge, M.; Banerjee, S.; and Rudnicky, A. I. 2010. Using the Amazon Mechanical Turk for transcription of spoken language. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP 2010, 5270–5273. Dallas, TX, USA: IEEE. doi:10.1109/ICASSP.2010.5494979.

Nguyen, A. T.; Wallace, B.; Li, J. J.; Nenkova, A.; and Lease, M. 2017. Aggregating and Predicting Sequence Labels from Crowd Annotations. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 299–309. Vancouver, BC, Canada: Association for Computational Linguistics. doi:10.18653/v1/P17-1028.

Pavlichenko, N.; Stelmakh, I.; and Ustalov, D. 2021. Crowd-Speech and Vox DIY: Benchmark Dataset for Crowd-sourced Audio Transcription. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. URL https://openreview.net/forum?id=3_hgF1NAXU7. arXiv:2107.01091 [cs.SD].

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12(85): 2825–2830. ISSN 1532-4435. URL https://jmlr.org/papers/v12/pedregosa11a.html.

Rodrigo, E. G.; Aledo, J. A.; and Gámez, J. A. 2019. spark-crowd: A Spark Package for Learning from Crowdsourced Big Data. *Journal of Machine Learning Research* 20: 1–5. ISSN 1532-4435. URL https://jmlr.org/papers/v20/17-743.html.

Rothe, R.; Timofte, R.; and Van Gool, L. 2018. Deep Expectation of Real and Apparent Age from a Single Image Without Facial Landmarks. *International Journal of Computer Vision* 126(2): 144–157. ISSN 1573-1405. doi:10.1007/s11263-016-0940-3.

Sheshadri, A.; and Lease, M. 2013. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *First AAAI Conference on Human Computation and Crowdsourcing*, HCOMP 2013, 156–164. Association for the Advancement of Artificial Intelligence. URL https://ojs.aaai.org/index.php/HCOMP/article/view/13088.

Whitehill, J.; Wu, T.-f.; Bergsma, J.; Movellan, J. R.; and Ruvolo, P. L. 2009. Whose Vote Should Count More:

Optimal Integration of Labels from Labelers of Unknown Expertise. In *Advances in Neural Information Processing Systems 22*, NIPS 2009, 2035–2043. Vancouver, BC, Canada: Curran Associates, Inc. ISBN 978-1-61567-911-9. URL https://papers.nips.cc/paper/3644-whose-vote-should-count-more-optimal-integration-of-labels-from-labelers-of-unknown-expertise.pdf.

Zhang, J.; Sheng, V. S.; Nicholson, B. A.; and Wu, X. 2015. CEKA: A Tool for Mining the Wisdom of Crowds. *Journal of Machine Learning Research* 16(88): 2853–2858. ISSN 1532-4435. URL https://jmlr.org/papers/v16/zhang15a.html.

Zheng, Y.; Li, G.; Li, Y.; Shan, C.; and Cheng, R. 2017. Truth Inference in Crowdsourcing: Is the Problem Solved? *Proceedings of the VLDB Endowment* 10(5): 541–552. ISSN 2150-8097. doi:10.14778/3055540.3055547.