Journal of Machine Learning Research 16 (2015) 2853-2858 Submitted 9/14; Revised 2/15; Published 12/15

CEKA: A Tool for Mining the Wisdom of Crowds

Jing Zhang JZHANG@NJUST.EDU.CN
School of Computer Science and Information Engineering
Hefei University of Technology (HFUT), Hefei 230009, China
Department of Software Engineering, School of Computer Science and Engineering
Nanjing University of Science and Technology (NJUST), Nanjing 210094, China

Victor S. Sheng SSHENG@UCA.EDU
Bryce A. Nicholson BNICHOLSON1210@HOTMAIL.COM
Department of Computer Science, University of Central Arkansas, Conway, AR 72035, USA

Xindong Wu XWU@UVM.EDU
School of Computer Science and Information Engineering, HFUT, Hefei 230009, China
Department of Computer Science, University of Vermont, Burlington, VT 05405, USA

Editor: Mark Reid

Abstract

CEKA is a software package for developers and researchers to mine the wisdom of crowds. It makes the entire knowledge discovery procedure much easier, including analyzing qualities of workers, simulating labeling behaviors, inferring true class labels of instances, filtering and correcting mislabeled instances (noise), building learning models and evaluating them. It integrates a set of state-of-the-art inference algorithms, a set of general noise handling algorithms, and abundant functions for model training and evaluation. CEKA is written in Java with core classes being compatible with the well-known machine learning tool WEKA, which makes the utilization of the functions in WEKA much easier.

Keywords: crowdsourcing, learning from crowds, multiple noisy labeling, inference, noise handling, repeated labeling simulation

1. Introduction

The emergence of crowdsourcing (Howe, 2006) has changed the way of knowledge acquisition. It has already attracted vast attentions of the machine learning and data mining research community in the past several years. Researchers show great interests in utilizing crowdsourcing as a new approach to acquire class labels of objects from common users, which costs much less than the traditional way—annotating by domain experts. In order to improve the labeling quality, an object usually obtains multiple labels from different non-expert annotators. Then, inference algorithms will be introduced to estimate the ground truths of these objects. Many inference algorithms have been proposed in recent years. Besides, building learning models from the inferred crowdsourced data is another research issue with great challenges, which aims at lifting the quality of a learned model to the level that can be achieved by training with the data labeled by domain experts.

To facilitate the research on mining the wisdom of crowds, we develop a novel software package named Crowd Environment and its Knowledge Analysis (CEKA). The main contri-

bution of CEKA lies on three aspects. (1) It provides comprehensive functions, which not only includes a great number of ground truth inference algorithms with a uniform easy-to-use programming interface but also includes a lot of well designed functions for the management of crowdsourced data. (2) It is seamlessly compatible with the famous machine learning tool WEKA (Hall et al., 2009), which facilitates the combination of the previous inference and the subsequent model learning procedures. (3) It is written in Java and completely open source. Therefore, many new ideas and methods, such as noise correction for crowdsourcing, are easily integrated. The project CEKA is available at: http://ceka.sourceforge.net/.

2. Design Principles and System Architecture

The design of CEKA follows three basic guidelines. (1) *Preferring integration of existing algorithms rather than implementing them.* Unless the original implementations of algorithms are not released, we always try to integrate the original versions rather than re-implementing them. The work that we have done is to unify the input/output file formats and wrap the different algorithms into some newly designed java classes with a uniform easy-to-use member functions. (2) *Seamlessly compatible with WEKA.* When input files that contain crowdsourced data are loaded into the memory and form a `Dataset` object, this object `Dataset` and all `Examples` inside can directly cooperate (e.g. training a model and conducting a cross-validation) with the related classes in WEKA. (3) *Extendibility.* Because machine learning in crowdsourcing is an emerging research domain, many topics such as multi-label tasks in crowdsourcing have not been touched yet. In order to integrate future research easily, when designing the core components of CEKA, we attempt to make the class structures as extendable as possible.

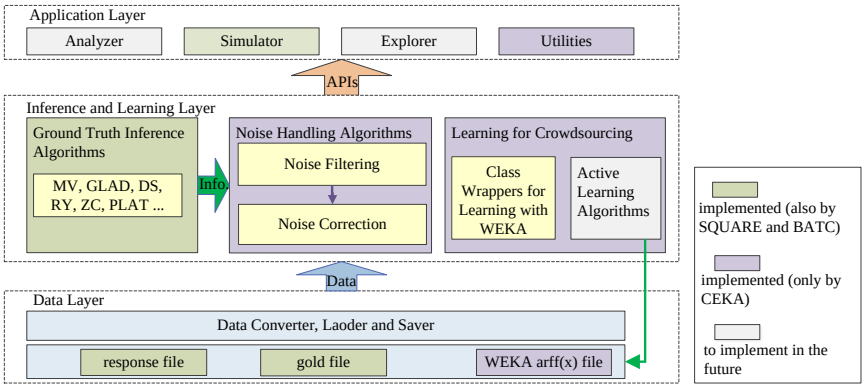

Figure 1: The architecture of CEKA

Figure 1 illustrates the hierarchical architecture of CEKA, in which we also compare it with two other tools for crowdsourcing SQUARE (Sheshadri and Lease, 2013) and BATC (Nguyen et al., 2013). Generally, SQUARE and BATC only provide some inference algorithms and several simple analysis functions. By contrast, CEKA conceives a more ambitious blueprint. It attempts to support the entire knowledge discovery procedure including analysis, inference and model learning. In the data layer, CEKA is able to read an arff(x) file defined by WEKA, which contains features of instances for subsequent model building. In the inference and learning layer, it provides a large number of inference algorithms. Our

on-going studies find that mislabeled instances after inference can be effectively detected and corrected, if a noise (mislabeled instance) handling algorithm can take advantage of the information generated in the previous inference procedure. Thus, CEKA provides a batch of noise handling algorithms. The core classes in this layer are derived from related classes in WEKA. In the application layer, CEKA provides a lot of utilities such as calculating performance evaluation metrics (i.e., accuracy, recall, precision, F source, AUC, M-AUC), manipulating data (i.e., shuffling, splitting and combining data), etc.

Algo.	CEKA	SQUARE	BATC	Comments	Algo.	CEKA	SQUARE	BATC	Comments
MV	●	●	●		CF	●			
DS	●	●	●		IPF	●			
GLAD	●	●	●	transplanted to Windows	MPF	●			
KOS	●		●		VF	●			
RY	●	●	●	by SQUARE	PLC	●			
ZenCrowd	●	●		by SQUARE	STC	●			
PLAT	●			for biased binary labeling	CC	●			unpublished
AWMV	●			unpublished					
GTIC	●			unpublished					

Table 1: Algorithms in CEKA compared with SQUARE and BATC

3. Algorithms

For the anonymous nature of crowdsourcing, CEKA currently only focuses on agnostic inference algorithms, which are independent of any other prior knowledge besides annotations assigned by non-experts. CEKA includes several novel inference algorithms proposed by the authors such as ground truth inference using clustering (GTIC) for multi-class labeling, adaptive weighted majority Voting (AWMV) for biased binary labeling as well as the well-known algorithms majority voting (MV), Dawid & Skene's algorithm (DS) (Dawid and Skene, 1979), GLAD (Whitehill et al., 2009), KOS (Karger et al., 2011), RY (Raykar et al., 2010), ZenCrowd (Demartini et al., 2012), and PLAT (Zhang et al., 2015). To embody our thought of introducing noise handling to improve the data quality of crowdsourcing, we have proposed a novel framework and an algorithm adaptive voting noise correction (AVNC) for crowdsourcing. In this framework, CEKA also includes a batch of noise filtering and correction algorithms, such as classification filtering (CF) (Gamberger et al., 1999), iterative partition filtering (IPF) (Khoshgoftaar and Rebours, 2007), multiple partition filtering (MPF) (Khoshgoftaar and Rebours, 2007), voting filtering (VF) (Brodley and Friedl, 1999), polishing label correction (PLC) (Teng, 1999), self-training correction (STC) (Triguero et al., 2014) and clustering correction (CC). Table 1 lists all algorithms in its current version (v1.0), comparing with SQUARE (Sheshadri and Lease, 2013) and BATC (Nguyen et al., 2013). Although our proposed algorithms GTIC, AWMV, and CC are under review, all of them still can be accessed in the source code.

4. Usage Example

CEKA can be easily deployed in both Windows and Linux systems. We have transplanted some algorithms such as GLAD from Linux to Windows. Figure 2 demonstrates a simple ex-

periment including the ground truth inference, noise correction and performance evaluation. In this sample code, like DS, all inference algorithms provide a uniform interface function `doInference`, which assigns every instance an integrated label. The class `Dataset` is completely compatible with the class `Instances` in WEKA, which can be directly accepted by a WEKA classifier as its parameter to train a model. Simply as the code shows, the statistical information of the performance will be obtained when the class `PerformanceStatistic` is applied to a `Dataset` object with the ground truth provided.

```
String respPath=D:/adult.response.txt; // labels obtained from crowd
String arffPath=D:/adult.arffx;        // ground truth and features
Dataset data = loadFile(respPath, null, arffPath);
// infer the ground truth by Dawid & Skene's algorithm
DawidSkene dsAlgo = new DawidSkene(50);
dsAlgo.doInference(data);
// noise filtering with the CF algorithm
Classifier [] classifiers = new Classifier[1];
Classifiers[0] = new SMO();            // SMO Classifier in WEKA
ClassificationFilter noiseFilter = new ClassificationFilter(10);
Dataset[] subData = null;              // cleansed and noise data sets
cf.FilterNoise(data, classifiers[0]);  // conduct noise filtering
subData[0] = noiseFilter.getCleansedDataset();
subData[1] = noiseFilter.getNoiseDataset();
// noise correction with STC algorithm
SelfTrainCorrection stc = new SelfTrainCorrection(subData[0], subData[1], 1.0);
stc.correction(classifiers[0]);        // correct mislabeled data
// combining two data sets and then evaluate performance
DatasetManipulator.addAllExamples(subData[0], subData[1]);
PerformanceStatistic perfStat = new PerformanceStatistic();
perfStat.stat(subData[0]);
```

Figure 2: A sample code for a basic usage

5. Conclusion and Future Work

CEKA is an easy-to-use open-source package for inference and machine learning tasks in crowdsourcing. The current version of CEKA includes a large number of ground truth inference algorithms, noise handling algorithms and useful functions supporting different learning tasks. That CEKA is designed to cooperate with WEKA definitely facilitates and accelerates the research progress in this field. CEKA is still growing. The future work includes introducing crowdsourcing-specific active learning strategies, developing several GUI tools (analyzer, simulator, and explorer) as well as integrating more inference, noise handling and learning algorithms proposed either by the authors or other researchers.

Acknowledgments

This research has been supported by the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China, under grant IRT13059, the National 973 Program of China under grant 2013CB329604, the National

Natural Science Foundation of China under grant 61229301, and the US National Science Foundation under grant IIS-1115417.

References

Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–161, 1999.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, pages 20–28, 1979.

Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *World Wide Web*, pages 469–478. ACM, 2012.

Dragan Gamberger, Nada Lavrac, and Ciril Groselj. Experiments with noise filtering in a medical domain. In *ICML*, pages 143–151, 1999.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6):1–4, 2006.

David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *NIPS*, pages 1953–1961, 2011.

Taghi M Khoshgoftaar and Pierre Rebours. Improving software quality prediction by noise filtering techniques. *Journal of Computer Science and Technology*, 22(3):387–396, 2007.

Quoc Viet Hung Nguyen, Thanh Tam Nguyen, Ngoc Tran Lam, and Karl Aberer. Batc: a benchmark for aggregation techniques in crowdsourcing. In *ACM SIGIR*, pages 1079–1080. ACM, 2013.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

Aashish Sheshadri and Matthew Lease. Square:. In *The First AAAI Conference on Human Computation and Crowdsourcing*, pages 156–164, 2013.

Choh-Man Teng. Correcting noisy data. In *ICML*, pages 239–248, 1999.

Isaac Triguero, José A Sáez, Julián Luengo, Salvador García, and Francisco Herrera. On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing*, 132:30–41, 2014.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043, 2009.

Jing Zhang, Xindong Wu, and Victor S Sheng. Imbalanced multiple noisy labeling. *IEEE Transaction on Kownledge and Data Engineering*, 27(2):489–503, 2015.