K-Nearest Neighbor Based Crowdsourcing Data Classification Algorithm (Postprint)

Authors: Li Jiaye, Yu Hao

Date: 2019-01-28T00:00:00+00:00

Abstract

To address the quality control problem in crowdsourced data processing, we propose a weighted K-nearest neighbor voting classification method. This method does not solely consider the label of a single instance to return an answer, but rather comprehensively integrates the neighbors of the instance to obtain a more accurate answer. Simultaneously, it assigns appropriate weights to the neighbors of the instance to further improve the algorithm' s performance while preserving the simplicity of traditional majority voting classification. The K-nearest neighbor voting classification algorithm can effectively address scenarios with insufficient labels, and by weighting the neighbors, it can mitigate the effects of imbalanced labels, thereby enhancing the algorithm' s generalization capability. Experimental results across various scenarios demonstrate that the weighted K-nearest neighbor voting classification method achieved excellent results.

Full Text

Preamble

Crowdsourcing Data Classification Algorithm via K-Nearest Neighbor

Li Jiaye[1], Yu Hao[2]†
(1. Guangxi Key Laboratory of Multi-source Information Mining & Security, Guangxi Normal University, Guilin Guangxi 541004, China;
2. School of Information Science & Engineering, Central South University, Changsha 410083, China)

Abstract: Addressing the quality control challenges in crowdsourcing data processing, this paper proposes a weighted K-nearest neighbor voting classification method. Rather than relying solely on the label of a single sample to produce an answer, this method obtains more accurate results by comprehensively considering the sample' s neighbors. By applying appropriate weights to neighboring

samples, the algorithm' s performance is further improved while maintaining the simplicity of traditional majority voting. The K-nearest neighbor voting classification algorithm effectively resolves situations with insufficient labels, and weighting the neighbors mitigates the impact of imbalanced labeling, thereby enhancing the algorithm' s generalization capability. Experiments across various scenarios demonstrate that the proposed weighted K-nearest neighbor voting method achieves excellent results.

Keywords: crowdsourcing data; quality control; K-nearest neighbor voting; majority voting

0 Introduction

In the era of artificial intelligence, the importance of data is self-evident [**?**], influencing all aspects of the world. Alibaba' s City Brain project applies AI technology to urban big data, enabling suspect identification within 20 minutes; Didi Chuxing analyzes urban data to plan optimal routes for vehicles, alleviating traffic congestion; Walmart analyzes customer purchase records to enable precise advertising for merchants. While most data collection tasks can be automated, machines often struggle to accurately process certain tasks such as image category annotation [**?**] or product quality assessment. Recent research has shown that data effectiveness and volume can impact experimental results even more than algorithmic optimization itself [**?**]. Consequently, obtaining high-quality, large-scale datasets has become an urgent challenge for researchers. In 2009, Princeton University' s Fei-Fei Li team introduced the ImageNet dataset [**?**], now the world' s largest image recognition database. Over nearly a decade, ImageNet has profoundly impacted computer vision and the entire machine learning field, with object classification accuracy improving to 97.3% between 2010–2017, surpassing human-level performance [**?**]. This number continues to approach 100%, demonstrating that large-scale real datasets are crucial for scientific research. Initially, annotating 16 million images seemed impossible, but Li' s team discovered Amazon Mechanical Turk, a crowdsourcing platform that distributed annotation tasks to interested individuals worldwide, completing the task in just over two years. ImageNet's success proved crowdsourcing to be both necessary and efficient. However, crowdsourcing introduces challenges: annotators are often non-experts with varying skill levels, producing incorrect labels or leaving samples unlabeled due to uncertainty. These issues result in noisy and incomplete crowdsourced data [**?**], making targeted processing essential for effective utilization.

Previous work by Zhang et al. [**?**] proposed an Efficient kNN Algorithm Based on Graph Sparse Reconstruction, which uses ℓ_1-norm to dynamically generate different K values for different samples, achieving good performance. However, this method only applies to standard datasets, not crowdsourced data, and is sensitive to missing labels. Additionally, Hao et al. [**?**] proposed a semi-

supervised classification algorithm based on fuzzy nearest neighbor label propagation, which can classify unlabeled data but is limited to semi-supervised learning and single-label classification, making it unsuitable for multi-label crowdsourced data. This paper proposes a weighted K-nearest neighbor voting classification algorithm specifically for crowdsourced data problems, applying K-nearest neighbor voting to handle missing labels while using neighbor weights for further improvement.

2.1 MV Method

As shown in Table 1 , consider a crowdsourcing dataset where annotators A, B, and C label samples $X_1, X_2, \ldots, X_n$. The entries represent labels assigned by annotators. In practice, crowdsourced data inevitably suffers from missing labels and errors due to annotator negligence or limited expertise.

The classic Majority Voting (MV) method [**?**] determines the correct label by majority rule, as shown in Equation (1):

$$\hat{c} = \arg\max_{c \in \Omega} \sum_{l \in S_x} \mathbb{1}(l = c)$$

where S_x is the set of all labels for a sample (with cardinality equal to the number of annotators), $\mathbb{1}(\cdot)$ returns 1 if the condition is true and 0 otherwise, l is a label, c is the true class, and $\Omega = \{1, 2, \ldots, C\}$ is the set of possible classes. If $v(c|x) > 0.5$, the MV-derived label is correct.

However, MV assumes equal competence among annotators [**?**]. For instance, in Table 1, if annotator A (college-educated, highly skilled) labels correctly while B and C (high school and middle school-educated, respectively) err, MV produces an incorrect result [**?**, **?**]. Furthermore, MV only considers labels for the current sample, ignoring neighbor information. In reality, some samples may be too difficult for any annotator, resulting in no labels or complete label absence, making MV unable to return a definitive answer.

1.1 KNN Algorithm

The K-nearest neighbor (KNN) algorithm [**?**] is a classic data mining technique. Its core principle is that a data point' s class can be determined by its k nearest correctly classified neighbors, assuming nearby points share the same class. Specifically, given a k value, the algorithm calculates distances between the query point and all other points, retains the k nearest neighbors, and assigns the most frequent class among them. Distance metrics include Manhattan and Euclidean distances. Since KNN relies only on local neighbors rather than global class distributions, it excels at multi-class problems with overlapping sample

regions. Many improvements exist: Zhang et al. [**?**] proposed CM-Knn with dynamic k values for different test data to reduce sensitivity to k; Zhang [**?**] developed KNN-CF incorporating certainty factors to address class imbalance. KNN' s simplicity, effectiveness, and suitability for multi-class problems make it widely applicable.

1.2 Random Forest Algorithm

Random forest is a machine learning algorithm based on decision trees [**?**]. Decision trees [**?**] are classic supervised learning models that, unlike linear logistic regression, form a tree structure (non-linear model). While logistic regression linearly separates samples by weighting all features against a threshold, decision trees process features separately through a hierarchical structure for more precise non-linear partitioning. The root node represents the most important feature, child nodes represent distinguishing features, and leaf nodes represent final classes. Random forest applies ensemble learning to decision trees, combining votes from multiple trees into a strong classifier for improved accuracy. Key characteristics include: random sampling of training sets (with replacement) to increase generalization, and bootstrap sampling that introduces randomness while maintaining correlation, enabling unbiased estimation of internal errors. Random forest' s high accuracy, suitability for high-dimensional data, and robustness to missing values make it popular for recommendation systems and predictive modeling.

2.2 W-Knv Method

To address MV' s limitations, this paper proposes the Weighted K-nearest neighbor voting method (W-Knv), defined in Equation (2):

$$\hat{c} = \arg\max_{c\in\Omega} \left[\sum_{l\in S_x} \mathbb{1}(l=c) + \alpha \sum_{i=1}^{k} \alpha_i v(c|x_i) \right]$$

where $x_i \in N_K(x)$ denotes the i-th neighbor of sample x, α controls the weight vector, and $v(c|x_i)$ represents the vote for class c from neighbor x_i. The weight vector $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_k]$ assigns weights to neighbors based on proximity, with closer neighbors receiving higher weights. The mean of α' s elements is denoted $\bar{\alpha}$. As α increases, the neighbor term' s influence grows; when all $\alpha_i = 0$, W-Knv reduces to MV. The weight vector allows scientific inference even with missing or imbalanced labels.

The W-Knv method offers several advantages:
a) It mitigates the impact of missing labels and noise by leveraging neighbor

information, providing definitive answers regardless of label completeness.
b) Unlike MV, which struggles with imbalanced labels from annotators of varying competence, W-Knv easily obtains sufficient labels from neighbors.
c) It captures relationships between samples, as similar samples tend to cluster, yielding more accurate results than MV' s label-only approach.
d) The weight vector α can be tuned to suppress erroneous labels, improving performance.

3.1 Experimental Datasets and Parameter Settings

Experiments validate the W-Knv algorithm using eight UCI datasets [**?**]: CCUDS, CNAE, Drift, Ecoli, Yale, Chess, Movements, and Soybean. Dataset details are shown in Table 2 .

Parameters are configured to simulate realistic crowdsourcing scenarios:
a) **Average label count** (λ): Set to 3 or 5 to model varying annotation counts per sample, including unlabeled or partially labeled cases.
b) **Beta distribution parameter** (con): The number of labels per sample follows $S \sim B(con, con)$, where S represents the label count. This ensures variable annotation counts, making the synthetic data more realistic.
c) **Reliability parameter** (rel): Represents the probability of correct annotation, set in the range $(0.5, 1)$ since annotator error rates are typically low.

Crowdsourced data is generated synthetically:
a) Random forest classifies the data, and its predictions versus true labels produce a confusion matrix $M \in \mathbb{R}^{c\times c}$, where c is the number of classes.
b) Matrix R is constructed from M by: (i) setting diagonal elements $R_{cc} = rel$, and (ii) setting off-diagonal elements $R_{c'c} = \frac{1-rel}{|M_c|-1}$ for $c' \neq c$, where M_c is the c-th row of M.
c) For a sample of class c, S elements are drawn from row c of R to form its label set. Repeating this for all samples generates the crowdsourced data.

This process produces synthetic crowdsourced data that closely approximates real-world characteristics.

3.2 Experimental Results and Analysis

All experiments run on MATLAB 2014a under Windows 7. Figure 1 [Figure 1: see original paper] shows W-Knv accuracy across different K values with $rel = 0.6$, $con = 1$, and $\lambda = 3$ or 5. Accuracy remains stable for $K \geq 5$, so $K = 5$ is used in subsequent experiments.

Figures 2-9 [FIGURE:2-9] compare MV and W-Knv across eight datasets while varying con (with $K = 5$, $rel = 0.6$, $\lambda = 3$ or 5). Since con values are identical for the Beta distribution, small con yields few labels per sample, while large

con approaches λ. Accuracy increases with *con* for both methods, but MV suffers more when *con* is small due to insufficient labels. W-Knv consistently outperforms MV across all label count variations.

Figures 10–17 [FIGURE:10-17] show 10 experimental runs on eight datasets with $K = 5$, $con = 1$, $\alpha = [5, 4, 3, 2, 1]$, testing *rel* values of 0.5, 0.6, and 0.7. While accuracy doesn' t always increase with *rel*, W-Knv generally surpasses MV. Adjusting *rel* simulates annotator skill levels to evaluate performance.

Both MV (Equation 1) and W-Knv (Equation 2) have linear time complexity $O(n)$ relative to sample count. W-Knv requires additional neighbor computations but remains linear overall. Setting all $\alpha_i = k$ (equal neighbor weights) reduces W-Knv to standard KNN applied to crowdsourced data, which performs worse than the weighted version. The proposed weighted approach thus represents an improved application of KNN principles to crowdsourced label aggregation.

W-Knv' s superior performance stems from three factors:
a) Robustness to missing or sparse labels, ensuring applicability across scenarios.
b) Leveraging both sample labels and neighbor relationships for more accurate inference.
c) Weighted neighbors suppress inaccurate labels, enhancing performance.

---

4 Conclusion

This paper proposes a weighted K-nearest neighbor voting classification algorithm for crowdsourced learning. By identifying neighbors and assigning distance-proportional weights, the method improves performance while retaining MV' s simplicity. Experiments confirm its effectiveness. Future work will explore different classification algorithms and probability estimation methods for further improvement.

References

[1] Deng Zhenyun, Zhu Xiaofeng, Cheng Debo, et al. Efficient KNN classification algorithm for big data [J]. Neurocomputing, 2016, 195 (C): 143-148.

[2] Zhang Shichao, Li Xuelong, Zong Ming, et al. Efficient KNN classification with different numbers of nearest neighbors [J]. IEEE Trans on Neural Networks & Learning Systems, 2018, 29 (5): 1774-1785.

[3] Zhu Xiaofeng, Zhang Shichao, Jin Zhi, et al. Missing value estimation for mixed-attribute datasets [J]. IEEE Trans on Knowledge and Data Engineering, 2011, 23 (1): 110-121.

[4] Deng Jia, Dong Wei, Socher R, et al. ImageNet: A large-scale hierarchical image database [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. [S.l.]: IEEE Press, 2009: 248-255.

[5] Zhu Xiaofeng, Li Xuelong, Zhang Shichao. Block-row sparse multi-view multi-label learning for image classification [J]. IEEE Trans on Cybernetics, 2016, 46 (2): 450-461.

[6] Qin Yongsong, Zhang Shichao, Zhu Xiaofeng, et al. Semi-parametric optimization for missing data imputation [J]. Applied Intelligence, 2007, 27 (1): 79-88.

[7] Zhang Shichao, Zong Ming, Sun Ke, et al. Efficient KNN algorithm based on graph sparse reconstruction [M]// Advanced Data Mining and Applications. Germany: Springer Press, 2014: 356-369.

[8] Hao Jianbai, Chen Xianfu, Huang Shuangfu, et al. A semi-supervised classification algorithm based on fuzzy near-neighbor label transfer [J]. Microelectronics & Computer, 2010, 27 (2): 30-33.

[9] Zhang Minling, Zhou Zhihua. ML-KNN: a lazy learning approach to multi-label learning [J]. Pattern Recognition, 2007, 40 (7): 2038-2048.

[10] Zhang Shichao, Li Xuelong, Zong Ming, et al. Learning k for KNN classification [J]. ACM Trans on Intelligent Systems & Technology, 2017, 8 (3): 43.

[11] Zhang Shichao. KNN-CF approach: incorporating certainty factor to kNN classification [J]. IEEE Intelligent Informatics Bulletin, 2010, 11 (1): 25–34.

[12] Deng Shengxiong, Yan Jiangtao, Liu Yong, et al. Classification model of integrated random forests [J]. Application Research of Computers, 2015, 32 (6): 1621-1624.

[13] Han Hui, Mao Feng, Wang Wenyuan. Recent advances in decision tree algorithms in data mining [J]. Application Research of Computers, 2004, 21 (12): 5-8.

[14] Snow R, O' Connor B, Jurafsky D, et al. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks [C]// Proc of Conference on Empirical Methods in Natural Language Processing. [S.l.]: Association for Computational Linguistics, 2008: 254-263.

[15] Hu Huiqi, Zheng Yudian, Bao Zhifeng, et al. Crowdsourced POI labelling: location-aware result inference and task assignment [C]// Proc of IEEE International Conference on Data Engineering. [S.l.]: IEEE Press, 2016: 61-72.

[16] Ouyang R W, Kaplan L, Martin P, et al. Debiasing crowdsourced quantitative characteristics in local businesses and services [C]// Proc of the 14th International Conference on Information Processing in Sensor Networks. [S.l.]: ACM Press, 2015: 190-201.

[17] Li Guoliang, Wang Jiannan, Zheng Yudian, et al. Crowdsourced data management: a survey [J]. IEEE Trans on Knowledge & Data Engineering, 2016, 28 (9): 2296-2319.

[18] UCI repository of machine learning datasets [EB/OL]. [2016-05-27] http://archive.ics.uci.edu/ml/.

[19] Cao C C, She Jieying, Tong Yongxin, et al. Whom to ask?: Jury selection for decision making tasks on micro-blog services [J]. Proceedings of the Vldb Endowment, 2012, 5 (11): 1495-1506.

Note: Figure translations are in progress. See original paper for figures.