Exploring Outliers in Crowdsourced Ranking for QoE

Qianqian Xu[1], Ming Yan[2], Chendi Huang[3],
Jiechao Xiong[3,4], Qingming Huang[5,6,7], Yuan Yao[8*]
[1] State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, CAS, Beijing, 100093, China
[2] Department of Computational Mathematics, Science and Engineering and Department of Mathematics, Michigan State University, East Lansing, MI, 48824, USA
[3] BICMR-LMAM-LMEQF-LMP, School of Mathematical Sciences, Peking University, Beijing, 100871, China
[4] Tencent AI Lab, Shenzhen, 518057, China
[5] University of Chinese Academy of Sciences, Beijing, 100049, China
[6] Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, 100190, China
[7] Key Lab of Big Data Mining and Knowledge Management, CAS, Beijing, 100190, China
[8] Department of Mathematics, Hong Kong University of Science and Technology, Hong Kong, 100871
xuqianqian@iie.ac.cn,yanm@math.msu.edu,cdhuang@pku.edu.cn
jcxiong@tencent.com,qmhuang@ucas.ac.cn,yuany@ust.hk

ABSTRACT

Outlier detection is a crucial part of robust evaluation for crowdsourceable assessment of Quality of Experience (QoE) and has attracted much attention in recent years. In this paper, we propose some simple and fast algorithms for outlier detection and robust QoE evaluation based on the nonconvex optimization principle. Several iterative procedures are designed with or without knowing the number of outliers in samples. Theoretical analysis is given to show that such procedures can reach statistically good estimates under mild conditions. Finally, experimental results with simulated and real-world crowdsourcing datasets show that the proposed algorithms could produce similar performance to Huber-LASSO approach in robust ranking, yet with nearly 8 or 90 times speed-up, without or with *a prior* knowledge on the sparsity size of outliers, respectively. Therefore the proposed methodology provides us a set of helpful tools for robust QoE evaluation with crowdsourcing data.

CCS CONCEPTS

•Information systems →Data cleaning; Rank aggregation;

KEYWORDS

HodgeRank; Outlier Detection; l_0-regularization; Iterative Hard Thresholding; Iterative Least Trimmed Squares; Adaptive Algorithms

*Corresponding author.

MM'17, October 23–27, 2017, Mountain View, CA, USA.

DOI: https://doi.org/10.1145/3123266.3123267

1 INTRODUCTION

In recent years, the Quality of Experience (QoE) [20, 23] has become a major research theme within the multimedia community. QoE measures a user's subjective expectation, feeling, perception, and satisfaction with respect to multimedia content. Measuring and ensuring good QoE of multimedia content is highly subjective in nature.

A variety of approaches can be employed to conduct subjective tests, among which Mean Opinion Score (MOS) [1] and paired comparison are the two most popular ones. In the MOS test, individuals are asked to specify a rating from "Bad" to "Excellent" (e.g., Bad-1, Poor-2, Fair-3, Good-4, and Excellent-5) to grade the quality of a stimulus; while in paired comparison approach, raters are only asked to make intuitive comparative judgments instead of mapping their perception on a categorical or numerical scale. Among these there may be tradeoffs in the amount of information the preference label contains and the bias associated with obtaining the label. For example, while a graded relevance judgment on a five-point scale may contain more information than a binary judgment, raters may also make more errors due to the complexity of assigning finer-grained judgments. In [5], it shows that MOS may suffer from three fundamental problems: (i) it is unable to concretely define the concept of scale; (ii) the interpretations of the scales among raters are highly different; (iii) it is difficult to verify whether a rater gives false ratings either intentionally or carelessly. Therefore, the paired comparison method is currently gaining growing attention. It not only promises assessments that are easier and faster to obtain with less demanding task for raters, but also yields more reliable data with less personal scale bias in practice. However, a shortcoming of paired comparison is that it has more expensive sampling complexity than the MOS test, since the number of pairs grows quadratically with the number of items to be ranked.

To tackle the cost problem, with the growth of crowdsourcing platforms such as MTurk, InnoCentive, CrowdFlower, CrowdRank, and AllOurIdeas, researchers who wish to

seek help from the Internet crowd can post their task requests on websites for QoE evaluation [5, 8, 9, 13, 23, 24]. Methods for rating/ranking via pairwise comparisons in QoE evaluation in crowdsourcing scenarios must address a number of inherent difficulties including: (i) incomplete and imbalanced data; (ii) streaming and online data; (iii) outliers. To meet the first challenge, the work in [6, 24, 26] propose randomized paired comparison methods which accommodate incomplete and imbalanced data. A general framework named *HodgeRank on Random Graphs* (HRRG) not only deals with incomplete and imbalanced data collected from crowdsourcing studies but also derives the constraints on sampling complexity that the random selection must adhere to in crowdsourcing experiment. Furthermore, a recent extension of HRRG is introduced in [25, 28] to deal with streaming and online data in the crowdsourcing scenario, providing the possibility of making assessment procedure significantly faster without deteriorating the accuracy.

The third challenge of crowdsourcing QoE evaluations is because not every Internet rater is trustworthy. In other words, due to the lack of supervision when raters perform experiments in crowdsourcing, they may provide erroneous responses perfunctorily, carelessly, or dishonestly [5]. Such random decisions are useless and may deviate significantly from other raters' decisions. So outliers have to be identified and removed in order to achieve a robust QoE evaluation. Many methods have been developed for outlier detection, such as M-estimator [10], Least Median of Squares (LMS) [18], S-estimators [19], Least Trimmed Squares (LTS) [17], and Thresholding based Iterative Procedure for Outlier Detection (Θ-IPOD) [21] etc. Besides, there are also distribution-based [2], depth-based [12], distance-based [14, 15], density-based [3], and clustering-based [11] methods for outlier detection. The authors of [5] proposed Transitivity Satisfaction Rate (TSR), which checks all the intransitive triangles, e.g., $A \succ B \succ C \succ A$, to identify and discard noisy data provided by unreliable raters in QoE. However, TSR can only be applied to complete and balanced paired comparison data. When the paired data is incomplete and imbalanced, i.e., having missing edges, the question of how to detect the noisy pairs remains open. The work in [27] attacks this problem and formulates the outlier detection as a LASSO problem based on sparse approximations of the cyclic ranking projection of paired comparison data in Hodge decomposition. Regularization paths of the LASSO problem could provide an order on samples tending to be outliers. However, the solution of the LASSO problem is biased. Solving the LASSO path is too slow and the problem has to be solved for many times for model selection via cross-validation.

In this paper, we propose simple and fast algorithms based on nonconvex optimization for outlier detection and robust ranking in QoE evaluation. The contributions of this paper are as follows:

1. We propose 3 iterative procedures solving some nonconvex optimization problems arising from outlier detection with or without knowing the number of outliers in samples.

2. Theoretical analysis shows that such procedures can reach statistically good estimates under mild conditions.

3. Experiments with simulated and crowdsourcing real-world data show that our algorithms work effectively in practice.

2 METHODOLOGY

In this section, we propose some simple iterative algorithms for outlier detection by solving some nonconvex optimization problems. These algorithms are based on either a prior knowledge on the number of outliers or adaptive estimation of the outlier sparsity size. Specifically, we propose iHT and iLTS with known outlier sparsity size and aLTS for adaptive estimation of outliers without knowing its precise number. In spite of the NP-hardness for finding global optimizers in the worst case, we show that such simple algorithms are able to reach statistically good estimates under mild conditions. Before the algorithms are described, a brief introduction on robust ranking is provided which motivates our main development.

2.1 Robust Ranking

Assume that there are m raters and n items to be ranked by the m raters. Let N be the the total number of paired comparisons (samples). Let vector $\mathbf{Y} = (Y_{ij}^{\alpha})_{i<j;\alpha} \in \mathbb{R}^N$ denote the degree that rater α prefers item i to item j. Without loss of generality, we assume that $Y_{ij}^{\alpha} > 0$ if rater α prefers item i to item j and $Y_{ij}^{\alpha} < 0$ otherwise. In addition, we assume that the paired comparison data is *skew-symmetric* for each α, i.e., $Y_{ij}^{\alpha} = -Y_{ji}^{\alpha}$. In practice, Y_{ij}^{α} can be continuous, dichotomous or of a k-point Likert scale with $k \geq 3$ according to the strategy used in QoE evaluation.

It is natural to assume that

$$Y_{ij}^{\alpha} = s_i^* - s_j^* + Z_{ij}^{\alpha *}, \tag{1}$$

$\mathbf{s}^* = (s_1^*, \ldots, s_n^*)^T \in \mathbb{R}^n$ is the true ranking score on n items and $Z_{ij}^{\alpha *}$ is the noise satisfying $Z_{ij}^{\alpha *} = -Z_{ji}^{\alpha *}$. When the noise $Z_{ij}^{\alpha *}$ is independent and identically distributed with zero mean, least squares (LS) problem has been used in [24–26] to derive ranking scores in subjective multimedia assessments.

However, not all comparisons are trustworthy and there may be sparse outliers due to different test conditions, human errors, or abnormal variations in content. Putting in a mathematical way, here we consider

$$Y_{ij}^{\alpha} = s_i^* - s_j^* + E_{ij}^{\alpha *} + N_{ij}^{\alpha *}, \tag{2}$$

or equivalently

$$\mathbf{Y} = \mathbf{X}\mathbf{s}^* + \mathbf{E}^* + \mathbf{N}^*. \tag{3}$$

where $\mathbf{E}^* = (E_{ij}^{\alpha *}) \in \mathbb{R}^N$, which models the *outliers*, is sparse and has a much larger magnitude than $\mathbf{N}^* = (N_{ij}^{\alpha *})$, which models the Gaussian *noise*, and $\mathbf{X} \in \mathbb{R}^{N \times n}$ satisfies that: if Y_{ij}^{α} $(i < j)$ is the kth entry of $\mathbf{Y}$, then the kth row of $\mathbf{X}$ equals to $\mathrm{e}_i - \mathrm{e}_j$, here $\mathrm{e}_i \in \mathbb{R}^n$ satisfies that only the ith entry is 1 and others are 0. Such $\mathbf{X}$ is often called the (generalized) "gradient operator" on graph $G = (\{1, \ldots, n\}, \{(i,j) : Y_{ij}^{\alpha} \text{ exists}\})$, with $\mathbf{L} = \mathbf{X}^T\mathbf{X}$ being the (unnormalized) graph Laplacian.

When sparse outliers exist ($E_{ij}^{\alpha *} \neq 0$ for a small number of (i,j,α)), the solution to the least squares problem on all the comparisons becomes unstable and may give an inaccurate estimation. If the outliers can be detected and removed, the solution to the least squares problem on the remaining pairwise comparisons is more accurate and gives a better estimation.

In [27], a robust regression approach called Huber-LASSO is used to detect outliers:

$$\underset{\mathbf{s}\in\mathbb{R}^n,\mathbf{E}}{\text{minimize}} \frac{1}{2}\|\mathbf{Y}-\mathbf{X}\mathbf{s}-\mathbf{E}\|_2^2+\lambda\|\mathbf{E}\|_1. \tag{4}$$

This is a convex optimization problem and the LASSO path $\lambda \mapsto \mathbb{E}_\lambda$ could provide information on the order of samples tending to be outliers.

However, there are two issues with this approach: 1) the Huber-LASSO estimator $\hat{\mathbf{s}}$ is always biased, even under the identifiable condition $\mathbf{s} \perp 1_n$ where $1_n = (1,\ldots,1)^T \in \mathbb{R}^n$; 2) computing the Huber-LASSO path to get top outliers is computationally expensive.

In order to remove the bias in the solution, we replace the l_1-norm of $\mathbf{E}$ in (4) with the l_0-"norm" of $\mathbf{E}$ and obtain

$$\underset{\mathbf{s}\in\mathbb{R}^n,\mathbf{E}}{\text{minimize}} \frac{1}{2}\|\mathbf{Y}-\mathbf{X}\mathbf{s}-\mathbf{E}\|_2^2+\lambda\|\mathbf{E}\|_0. \tag{5}$$

where $\|\mathbf{E}\|_0$, the l_0-"norm" of $\mathbf{E}$, is the number of nonzero components in $\mathbf{E}$. Although this is a nonconvex optimization problem which is NP-hard in the worst case, in the sequel we shall see that under mild conditions even simple iterative algorithms may detect where the outliers are and lead to statistically good estimators.

2.2 iHT and iLTS with Known K

Algorithm 1 iterative Hard Thresholding (iHT)

Input: $\mathbf{Y} = (Y_{ij}^{\alpha})$, $K \geq 0$, $\epsilon > 0$.
Initialization: $\mathbf{E}^0 = (E_{ij}^{\alpha})^0 = 0$.
for $k = 0, 1, \ldots$ **do**
  Update $\mathbf{E}$ by

$$\mathbf{E}^{k+1} = \text{Proj}_K((\mathbf{I}_N - \mathbf{H})\mathbf{Y} + \mathbf{H}\mathbf{E}^k),$$

  If $\|\mathbf{E}^{k+1} - \mathbf{E}^k\| \leq \epsilon$, break.
end for
return $\hat{\mathbf{E}} = \mathbf{E}^k$, $\hat{\mathbf{s}} = (\mathbf{X}^T\mathbf{X})^\dagger\mathbf{X}^T(\mathbf{Y}-\mathbf{E}^k)$.

First of all, Proposition 1, whose proof is provided in the supplementary material, shows that problem (5) is, in a sense, equivalent to

$$\begin{cases} \underset{\mathbf{s}\in\mathbb{R}^n,\mathbf{E}}{\text{minimize}} & \frac{1}{2}\|\mathbf{Y}-\mathbf{X}\mathbf{s}-\mathbf{E}\|_2^2, \\ \text{subject to} & \|\mathbf{E}\|_0 \leq K \end{cases} \tag{6}$$

and

$$\begin{cases} \underset{\mathbf{s}\in\mathbb{R}^n,\Lambda}{\text{minimize}} & \frac{1}{2}\|\Lambda \circ (\mathbf{Y}-\mathbf{X}\mathbf{s})\|_2^2, \\ \text{subject to} & \Lambda = (\Lambda_{ij}^{\alpha}) \in \{0,1\}^N,\ \|\Lambda\|_0 \geq N-K \end{cases} \tag{7}$$

where $\circ$ is elementwise Hadamard product operator. The index of zero entries of Λ indicate outliers. Problem (7) is actually the Least Trimmed Squares (LTS) in robust regression [17]. A benefit of (7) lies in that the global ranking score $\mathbf{s}$ does not depend on the outlier magnitude estimate, by dropping off the outliers.

Proposition 1. *For a given $\lambda > 0$, pick any global optimal $(\tilde{\mathbf{s}}, \tilde{\mathbf{E}})$ for problem (5), and let $K = \|\tilde{\mathbf{E}}\|_0$. Let*

$$\begin{aligned} S_1 = \{\, \mathbf{s} : \|\mathbf{E}\|_0 = K \text{ and} \\ (\mathbf{s},\mathbf{E}) \text{ is optimal for problem (5)} \,\} \\ S_2 = \{\, \mathbf{s} : (\mathbf{s},\mathbf{E}) \text{ is optimal for problem (6)} \,\} \\ S_3 = \{\, \mathbf{s} : (\mathbf{s},\Lambda) \text{ is optimal for problem (7)} \,\}. \end{aligned}$$

Then $S_1 = S_2 = S_3$.

Hence now we turn to problem (6) and (7), both have a parameter K, which is considered as an upper bound of the number of outliers. Because of the two l_0-"norm", finding the global optimal solution is NP-hard. We attempt to find approximate (but sufficient) solutions via the alternating minimization method.

Note that once we fix $\mathbf{E} = \mathbf{E}^k$ for problem (6), then we just need to solve an ordinary least squares problem and get a corresponding $\mathbf{s}^k$ simply by

$$\mathbf{s}^k = (\mathbf{X}^T\mathbf{X})^\dagger\mathbf{X}^T(\mathbf{Y}-\mathbf{E}^k). \tag{8}$$

Here $A^\dagger$ is the Moore–Penrose pseudoinverse of a matrix A. And if we fix $\mathbf{s} = \mathbf{s}^k$, we just need to take a Hard Thresholding, i.e.

$$\mathbf{E}^{k+1} = \text{Proj}_K\left(\mathbf{Y}-\mathbf{X}\mathbf{s}^k\right), \tag{9}$$

where Proj_K is an operator which sets all entries to 0 except K entries with largest squares. For example,

$$\text{Proj}_3(-1,5,2,-4,-6) = (0,5,0,-4,-6).$$

Plugging (8) into (9), such a procedure implies

$$\begin{aligned} \mathbf{E}^{k+1} &= \text{Proj}_K(\mathbf{Y}-\mathbf{X}(\mathbf{X}^T\mathbf{X})^\dagger\mathbf{X}^T(\mathbf{Y}-\mathbf{E}^k)) \\ &= \text{Proj}_K((\mathbf{I}_N-\mathbf{H})\mathbf{Y}+\mathbf{H}\mathbf{E}^k), \end{aligned}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^\dagger\mathbf{X}^T$ is the "hat matrix". Such a procedure is described precisely in Algorithm 1 and called iterative Hard Thresholding (iHT).

For problem (7), when Λ^k is fixed, update $\mathbf{s}$ by solving a least squares problem using only the comparisons indicated by Λ^k, i.e.

$$\mathbf{s}^k = (\mathbf{X}^T\text{diag}(\Lambda^k)\mathbf{X})^\dagger(\mathbf{X}^T\text{diag}(\Lambda^k)\mathbf{Y}). \tag{10}$$

When fixing $\mathbf{s} = \mathbf{s}^k$, updating Λ is to choose $N-K$ entries of $\mathbf{Y}-\mathbf{X}\mathbf{s}^k$ with smallest squares, then set the $N-K$ corresponding entries of Λ^{k+1} to be 1, and others to be 0. The procedure is described precisely in Algorithm 2.

[1] If the Kth and $(K+1)$th largest squares have the same value, there are multiple choices of Λ^{k+1}. In this case, randomly choose one of them different from all Λ's appeared before. If all the choices have appeared, break.

Algorithm 2 An Iterative Procedure for LTS (iLTS)

Input: $\mathbf{Y} = (Y_{ij}^{\alpha})$, $K \geq 0$.
Initialization: $\Lambda^0 = (\Lambda_{ij}^{\alpha})^0 = 1_N$.
for $k = 0, 1, \ldots$ **do**
Update $\mathbf{s}$ to get $\mathbf{s}^k$ by (10).
Update Λ by choosing $N-K$ entries of $\mathbf{Y} - \mathbf{X}\mathbf{s}^k$ with smallest squares[1], then setting the $N-K$ corresponding entries of Λ^{k+1} to be 1, and others to be 0.
Check if the new Λ^{k+1} is different from all Λ^l $(l \leq k)$ appeared before. **If** not, break.
end for
return $\hat{\Lambda} = \Lambda^k,\ \hat{\mathbf{s}} = \mathbf{s}^k$.

2.3 Consistency of iHT and iLTS

A natural question is, under what conditions can these two algorithms detect the true outlier set. The following theorems, whose proofs are given in the supplementary material, present some RIP-like sufficient conditions which can be met in outlier detection.

THEOREM 2.1 (SPARSISTENCY OF IHT). *Assume that* $\mathbf{Y} = (Y_{ij}^{\alpha})$ *satisfies the model* (3) *with* $\|\mathbf{E}^*\|_0 = K^*$ *and* $\mathbf{E}^*_{\min} = \min_{E_{ij}^{\alpha *} \neq 0} |E_{ij}^{\alpha *}|$. *Now, for arbitrary* $K \geq K^*$ *satisfying*

$$\theta := \sup_{J \subseteq \{1,2,\ldots,N\},\ |J| \leq 3K} \left\| \mathbf{X}_J (\mathbf{X}^T \mathbf{X})^{\dagger} \mathbf{X}_J^T \right\|_2 < \frac{1}{2} \tag{11}$$

(here $\mathbf{X}_J$ *is the submatrix consist of some columns of* $\mathbf{X}$ *indexed by* J*),* $\mathbf{E}^k$ *in Algorithm 1 converges to the true outlier vector* $\mathbf{E}^*$ *in the following sense*

$$\|\mathbf{E}^k - \mathbf{E}^*\|_2 \leq (2\theta)^k \cdot \|\mathbf{E}^0 - \mathbf{E}^*\|_2 + \frac{2\|\mathbf{N}^*\|_2}{1-2\theta}. \tag{12}$$

Moreover, if

$$\theta < \frac{1}{2} - \frac{\|\mathbf{N}^*\|_2}{\mathbf{E}^*_{\min}}, \tag{13}$$

then for sufficiently large k, $\mathrm{supp}(\mathbf{E}^k) \supseteq \mathrm{supp}(\mathbf{E}^*)$ *holds. If* (13) *holds and* $K = K^*$ *additionally, then for sufficiently large* k, $\mathrm{supp}(\mathbf{E}^k) = \mathrm{supp}(\mathbf{E}^*)$ *holds.*

REMARK 1. *Condition* (11) *resembles the condition in [7], with the measurement matrix* A *replaced by* $\mathbf{I}_N - \mathbf{H}$*, and the number of nonzero entries* s *replaced by* K*.*

REMARK 2. *According to the statement of the theorem above, we should choose* K *to be at least* K^**. But it is unnecessary to exactly let* K *be the unknown number* K^**, since we allow* K *to be larger than* K^**. However, usually* K *can not be too large, due to the condition* $\theta < 1/2$ *which must be satisfied.*

In the definition of θ*, note that* $\mathbf{X}_J(\mathbf{X}^T\mathbf{X})^{\dagger}\mathbf{X}_J^T := \mathbf{H}_{J,J}$ *is a* $|J| \times |J|$ *submatrix of* $\mathbf{H}$*, and* $\|\mathbf{H}\|_2 \leq 1$ *always holds since* $\mathbf{I}_N - \mathbf{H}$ *is always positive semi-definite. If* $3K$ *(upper bound of* $|J|$*) is small enough, then* θ *can be smaller than* $1/2$*, satisfying the proposed condition. For example, if* $n = 10$*,* $K = 1$*, and each pair has exactly one comparison, then* $\theta = 0.4 < 1/2$*.*

THEOREM 2.2 (CONVERGENCE OF ILTS). *Algorithm 2 converges in finite steps. Moreover, let*

$$F(\mathbf{s}, \Lambda) = \frac{1}{2}\|\Lambda \circ (\mathbf{Y} - \mathbf{X}\mathbf{s})\|_2^2 + \iota(\Lambda \in \{0,1\}^K,\ \|\Lambda\|_0 \geq N - K),$$

where $\iota(A)$ *is the indicator function, which equals* 0 *if the event* A *happens, and equals* $+\infty$ *otherwise. Then the output* $\mathbf{s}^k$ *with the corresponding* Λ^k *satisfies*

(1) $(\mathbf{s}^k, \Lambda^k)$ *is a coordinatewise minimum point of* $F(\mathbf{s}, \Lambda)$*, namely, for any* $\mathbf{s}, \Lambda$*,*

$$F(\mathbf{s}^k, \Lambda^k) \leq F(\mathbf{s}^k, \Lambda),$$
$$F(\mathbf{s}^k, \Lambda^k) \leq F(\mathbf{s}, \Lambda^k).$$

(2) $\mathbf{s}^k$ *is a local minimum point of* $E(\mathbf{s}) := \min_{\Lambda} F(\mathbf{s}, \Lambda)$.

REMARK 3. *There is no convergence analysis for iHT in general case. But this theorem tells that iLTS always converges, though they are two different iterative algorithms for two equivalent problems.*

THEOREM 2.3 (SPARSISTENCY OF ILTS). *Assume that* $\mathbf{Y} = (Y_{ij}^{\alpha})$ *satisfies the model* (3) *with* $\|\mathbf{E}^*\|_0 = K^*$*,* $\mathbf{E}^*_{\min} = \min_{E_{ij}^{\alpha *} \neq 0} |E_{ij}^{\alpha *}|$ *and* $\Lambda^* \in \{0,1\}^N$ *satisfying*

$$\Lambda_{ij}^{\alpha *} = \begin{cases} 1, & E_{ij}^{\alpha *} = 0, \\ 0, & E_{ij}^{\alpha *} \neq 0. \end{cases}$$

Now, for arbitrary $K \geq K^*$*, let*

$$\mu := \sup_{|J| \geq N-K} \|\mathbf{X}_{J^c}(\mathbf{X}_J^T\mathbf{X}_J)^{\dagger}\mathbf{X}_J^T\|_2 \tag{14a}$$

$$\eta := \sup_{|J| \geq N-K} \|\mathbf{X}_{J^c}(\mathbf{I}_n - (\mathbf{X}_J^T\mathbf{X}_J)^{\dagger}(\mathbf{X}_J^T\mathbf{X}_J))\|_2 \tag{14b}$$

$$\epsilon := \sqrt{2} \cdot \frac{(2+\mu)\|\mathbf{N}^*\|_2 + \eta\|\mathbf{s}^*\|_2}{\mathbf{E}^*_{\min}}. \tag{14c}$$

If

$$\varphi := \sup_{|J'| \leq 2K,\ |J| \geq N-K} \|\mathbf{X}_{J'}(\mathbf{X}_J^T\mathbf{X}_J)^{\dagger}\mathbf{X}_{J'}^T\|_2 < \sqrt{2} - 1 - \epsilon, \tag{15}$$

then for the $\hat{\Lambda}$ *corresponding with the output* $\hat{\mathbf{s}}$ *of Algorithm 2,* $\mathrm{supp}(\hat{\Lambda}) \subseteq \mathrm{supp}(\Lambda^*)$ *holds. If* (15) *holds and* $K = K^*$ *additionally, then* $\mathrm{supp}(\hat{\Lambda}) = \mathrm{supp}(\Lambda^*)$*.*

REMARK 4. *In the vast majority of cases,* $\eta = 0$*. In fact, as long as for each* $J \subseteq \{1, 2, \ldots, N\}$*,* $|J| \geq N - K$*, any row of* $\mathbf{X}_{J^c}$ *is a linear combination of* $\mathbf{X}_J$ *(which means that, removing the samples indicated by rows of* $\mathbf{X}_{J^c}$ *does not disturb the original structure of connected components of the graph), there is a matrix* $\mathbf{M}$ *such that*

$$\mathbf{X}_{J^c} = \mathbf{M}\mathbf{X}_J.$$

Thus

$$\begin{aligned} &\mathbf{X}_{J^c}(\mathbf{I}_n - (\mathbf{X}_J^T\mathbf{X}_J)^{\dagger}(\mathbf{X}_J^T\mathbf{X}_J)) \\ &= \mathbf{M}(\mathbf{X}_J - \mathbf{X}_J \cdot (\mathbf{X}_J^T\mathbf{X}_J)^{\dagger}\mathbf{X}_J^T \cdot \mathbf{X}_J) \\ &= \mathbf{M}(\mathbf{X}_J - \mathbf{X}_J\mathbf{X}_J^{\dagger}\mathbf{X}_J) = 0, \end{aligned}$$

which implies that $\eta = 0$*.*

Remark 5. *According to the statement of the theorem above, we should choose K to be at least K^*. But it is unnecessary to exactly let K be the unknown number K^*, since we allow K to be larger than K^*. However, usually K can not be too large, due to the condition $\varphi < \sqrt{2} - 1 - \epsilon$ which must be satisfied.*

Remark 6. *Conditions* (11) *and* (15) *play similar roles as Restricted Isometry Property (RIP) in compressed sensing [4].*

2.4 Adaptive LTS with Unknown K

If the exact number of outliers K is given or can be accurately estimated, Algorithm 1 or 2 can be used to detect the outliers and improve the performance of least squares solutions. However, in practice, the exact number of outliers K is generally unknown. If K is underestimated, we are able to remove some outliers and the remaining outliers will still damage the performance of the least squares solutions. On the other hand, if K is overestimated, too many comparisons are removed. The resulting data is not enough for robust QoE evaluation and provides unstable solutions. Therefore, a method to estimate the number of outliers accurately is strongly desired.

We propose a method to estimate the number of outliers automatically for dichotomous choice $Y_{ij}^{\alpha} \in \{\pm 1\}$. In this case, a natural way is to consider those outliers as the paired comparisons which disagree with the sign (or preference order) of global ranking score differences.

As the number of outliers is unknown, firstly we use the least squares problem to find an estimation of $\mathbf{s}$, then the total number of comparisons with wrong directions (Y_{ij}^{α} has different sign with $s_i - s_j$), which is denoted as $\widetilde{K}$, is an overestimation of K. Then we obtain an underestimation of the number of outliers via multiplying by $\beta_1 \in (0,1)$, i.e., $\underset{\sim}{K} = \beta_1 \widetilde{K}$. We remove $\underset{\sim}{K}$ comparisons that have largest violations to the current score because they are most likely to be outliers. The remaining comparisons are used to find the new estimation of $\mathbf{s}$ via the least squares problem. In this case, we are able to remove some outliers and improve the estimation for $\mathbf{s}$. With these improved estimation for $\mathbf{s}$, we are able to remove more outliers. So we increase the underestimation $\underset{\sim}{K}$ by β_2 ($\beta_2 \in (1, \infty)$). However, this number can not be larger than $\widetilde{K}$, the smallest overestimation of the number of outliers, because we do not want to remove too many comparisons. Therefore the update of $\underset{\sim}{K}$ is just $\underset{\sim}{K} = \min(\lceil \beta_2 \underset{\sim}{K} \rceil, \widetilde{K})$ where $\lceil x \rceil$ is the smallest integer no smaller than positive real number x. Iterations go on until $\underset{\sim}{K} = \widetilde{K}$, and it gives an accurate estimation of the number of outliers. This algorithm is named aLTS for adaptive Least Trimmed Squares, and Algorithm 3 describes such a procedure precisely.

Remark 7. *There are only two parameters to choose, and these two parameters are easy to set. They are chosen according to inequalities $\beta_1 < 1 < \beta_2$ ($\beta_1 = 0.75$ and $\beta_2 = 1.03$ are fixed in our numerical experiments). β_1 has to be small to make sure that the first estimation of the number of outliers is underestimated. Then the underestimation $\underset{\sim}{K}$ increases geometrically with rate β_2, and β_2 can not be too large, because the remain comparisons are not enough for robust QoE evaluation after too many comparisons are removed.*

Algorithm 3 adaptive LTS (aLTS)

Input: $\mathbf{Y} = (Y_{ij}^{\alpha})$, $\beta_1 < 1$, $\beta_2 > 1$.
Initialization: $\Lambda^0 = 1_N$, $\underset{\sim}{K}^{-1} = 0$, $\widetilde{K}^{-1} = +\infty$.
for $k = 0, 1, \ldots$ **do**
Update $\mathbf{s}$ to get $\mathbf{s}^k$ by (10).
Let $\widetilde{K}^k$ be the total number of comparisons with wrong directions, i.e., Y_{ij}^{α} has different sign with $s_i^k - s_j^k$.

$$\widetilde{K}^k = \min\{\widetilde{K}^k, \widetilde{K}^{k-1}\}.$$

$$\underset{\sim}{K}^k = \begin{cases} \lceil \beta_1 \widetilde{K}^k \rceil, & \text{if } k = 0; \\ \min(\lceil \beta_2 \underset{\sim}{K}^{k-1} \rceil, \widetilde{K}^k), & \text{otherwise.} \end{cases}$$

If $\underset{\sim}{K}^k = \widetilde{K}^k$, break.
Update Λ to get Λ^{k+1} in the same way as in Algorithm 2, with K replaced by $\underset{\sim}{K}^k$.
end for
return $\hat{\Lambda} = \Lambda^k$, $\hat{\mathbf{s}} = \mathbf{s}^k$, $\hat{K} = \widetilde{K}^k$.

Remark 8. *The algorithm is able to detect most of the outliers in our experiments. However, there may be mistakes in the detection, and these mistakes happen mostly between two successive items in the order. Therefore, we can add one step to just compare every pair of two successive items and make the correction on the detection, i.e., if $s_i^k > s_j^k$, but $|\{Y_{ij}^{\alpha} : Y_{ij}^{\alpha} > 0\}| < |\{Y_{ij}^{\alpha} : Y_{ij}^{\alpha} < 0\}|$, then remove $\{Y_{ij}^{\alpha} : Y_{ij}^{\alpha} < 0\}$ from outliers and add in $\{Y_{ij}^{\alpha} : Y_{ij}^{\alpha} > 0\}$.*

Algorithm 3 always stops in finite steps, as shown in the following lemma.

Lemma 2.4. *Algorithm 3 stops in no more than k^* steps, where*

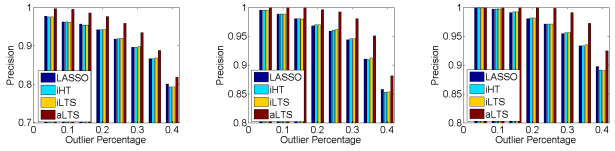
$$k^* = \left\lfloor \frac{-\log \beta_1}{\log \beta_2} \right\rfloor + 2.$$

Proof. It follows from the fact that the sequence $\{\widetilde{K}^k\}$ is non-increasing, and $\{\underset{\sim}{K}^k\}$ is a geometrically increasing sequence which is bounded by the smallest component of $\{\widetilde{K}^k\}$. Specifically, assume that k^* steps have been taken in Algorithm 3, then k has approached $k^* - 1$, and $\underset{\sim}{K}^k \geq \beta_2 \underset{\sim}{K}^{k-1}$ for $0 < k < k^* - 1$, so

$$\widetilde{K}^0 \geq \widetilde{K}^{k^*-2} \geq \underset{\sim}{K}^{k^*-2} \geq \beta_2^{k^*-2} \underset{\sim}{K}^0 \geq \beta_2^{k^*-2} \beta_1 \widetilde{K}^0,$$

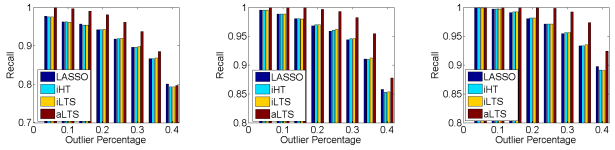
which leads to the result. □

Such a result only ensures that the algorithm stops with a possible overestimation of the number of outliers because $\widetilde{K}^k$ is always an overestimation for the number of outliers. The following theorem presents a stability condition when Algorithm 3 returns the correct number of outliers.



(a) SN=1000 (b) SN=2000 (c) SN=3000

Figure 1: *Precision*s for simulated data via LASSO, iHT, iLTS, and aLTS, 100 times repeat.



(a) SN=1000 (b) SN=2000 (c) SN=3000

Figure 2: *Recall*s for simulated data via LASSO, iHT, iLTS, and aLTS, 100 times repeat.

THEOREM 2.5. *Consider binary choice data with outliers*

$$Y_{ij}^{\alpha} \text{ is an outlier, if } Y_{ij}^{\alpha} \neq \text{sign}(s_i^* - s_j^*). \quad (16)$$

Assume that there exists an integer k_0 such that for all $k \geq k_0$, least squares estimator $\mathbf{s}^k$ is order-consistent to the true score $\mathbf{s}^*$, i.e., $\mathbf{s}^k$ induces the same ranking order as the true score $\mathbf{s}^*$, then Algorithm 3 returns the correct number of outliers.

PROOF. As $\mathbf{s}^k$ is an order-consistent solution of the ground-truth, by definition, $\widetilde{K}^k$ gives the correct number of outliers, say K^*. It actually holds for all $k \geq k_0$, that $\widetilde{K}^k = K^*$. From Lemma 1, the claim follows. □

REMARK 9. *One scenario is the generalized linear model where $p(i \succeq j) = f(s_i^* - s_j^*)$ for some cumulate distribution function f symmetric w.r.t. $f(0) = 1/2$. With a large enough sample, all the pairwise preferences in the minority direction can be regarded as "outliers" and dropping such outliers will not change the order consistency of least square estimators.*

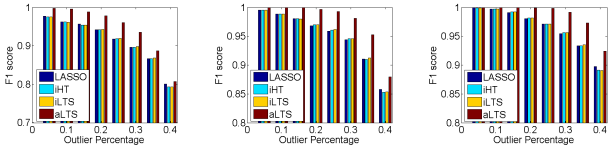
Note that Theorem 2.5 does not require Λ^k to correctly identify the outliers, but just stable estimator $\mathbf{s}^k$ to be order-consistent to $\mathbf{s}^*$. In practice, this might not be satisfied easily. But, as we shall see in the next section, Algorithm 3 typically returns stable estimators that only deviate locally.

3 EXPERIMENTS

A key question in the outlier detection community is how to evaluate the effectiveness of outlier detection algorithms when the ground-truth outliers are not available. In this section, we will first show the effectiveness of the proposed method on simulated data with known ground-truth outliers, followed by real-world crowdsourcing datasets without ground-truth outliers.

3.1 Simulated Data

The simulated data is constructed as follows. A random total order on n items is created as the ground-truth order.



(a) SN=1000 (b) SN=2000 (c) SN=3000

Figure 3: *F1* scores for simulated data via LASSO, iHT, iLTS, and aLTS, 100 times repeat.

Then we add paired comparison edges (i, j) randomly with preference directions following the ground-truth order. We simulate the outliers by randomly choosing a portion of the comparison edges and reversing them in preference directions. A paired comparison graph with outliers, possibly incomplete and imbalanced, is constructed.

Here we choose $n = 16$, which is consistent with the real-world datasets, and make the following definitions for the experimental parameters. The total number of paired comparisons occurred on the graph is **SN** (Sample Number), and the number of outliers is **ON** (Outlier Number). Then the outlier percentage **OP** can be obtained as **ON**/**SN**.

Most outlier detection algorithms adopt a tuning parameter (say t) in order to select different number of data samples as outliers [27], and the number of outliers detected changes as t changes. If t is picked too restrictively, then the algorithm will miss true outlier (false negatives). On the other hand, if the algorithm declares too many data samples as outliers, then it will lead to too many false positives. This tradeoff can be measured in terms of ***precision*** and ***recall***, which are commonly used for measuring the effectiveness of outlier detection methods. Specifically, the ***precision*** is defined as the percentage of reported outliers that truly turn out to be outliers; and the ***recall*** is correspondingly defined as the percentage of ground-truth outliers that have been reported as outliers.

We then compare LASSO, iHT, iLTS, and aLTS for outlier detection on the simulated data. For ease of comparison, here we should tell LASSO, iHT, and iLTS in advance the exact number of outliers existed in the dataset. Because, different from aLTS, these three methods can not estimate the number of outliers in the dataset automatically.

The mean ***precision***s, ***recall***s, and ***F1-score***s over 100 runs for these four methods on different choices of **SN** and **OP** are shown in Figures 1, 2, and 3. ***F1-score*** is a combined measure that assesses the ***precision***/***recall*** tradeoff, which reaches its best value at 1 and worst score at 0.

It is easy to see that the performances of LASSO, iHT, and iLTS are very similar, while aLTS could produce better performance (indicated by higher ***precision***s, ***recall***s, and ***F1-score***s in almost all cases). In addition, we compare the computing time required for these four methods to finish all the 100 runs in Tables 1. All computation is done using MATLAB R2014a, on a Mac Pro desktop PC, with 2.8 GHz Intel Core i7-4558u, and 16 GB memory. It is easy to see that on the simulated dataset, iHT, iLTS, and aLTS algorithms

Table 1: Computing time for 100 runs in total on simulated data via LASSO, iHT, iLTS, and aLTS.

(a) LASSO

time (s)	OP=5%	OP=10%	OP=15%	OP=20%	OP=25%	OP=30%	OP=35%	OP=40%
SN=1000	18.62	19.72	22.51	23.80	23.48	22.56	21.05	18.56
SN=2000	20.58	29.21	33.17	34.81	34.57	31.54	29.82	25.78
SN=3000	28.59	37.50	40.62	40.88	41.60	38.91	34.94	29.38

(b) iHT

time (s)	OP=5%	OP=10%	OP=15%	OP=20%	OP=25%	OP=30%	OP=35%	OP=40%
SN=1000	0.23	0.20	0.24	0.27	0.31	0.35	0.40	0.43
SN=2000	0.29	0.33	0.40	0.48	0.50	0.57	0.65	0.72
SN=3000	0.41	0.48	0.55	0.60	0.69	0.79	0.83	0.97

(c) iLTS

time (s)	OP=5%	OP=10%	OP=15%	OP=20%	OP=25%	OP=30%	OP=35%	OP=40%
SN=1000	0.27	0.24	0.30	0.33	0.37	0.40	0.43	0.45
SN=2000	0.37	0.41	0.53	0.60	0.63	0.73	0.79	0.83
SN=3000	0.53	0.63	0.70	0.78	0.90	1.03	0.99	1.12

(d) aLTS

time (s)	OP=5%	OP=10%	OP=15%	OP=20%	OP=25%	OP=30%	OP=35%	OP=40%
SN=1000	4.86	3.50	3.13	2.79	3.00	2.93	2.85	2.81
SN=2000	6.36	5.35	4.97	4.91	4.75	4.42	4.29	4.27
SN=3000	7.96	7.61	6.81	6.70	6.34	6.09	5.51	5.92

are much faster than LASSO, which implies their advantages in dealing with large-scale data. Specifically, iHT and iLTS can achieve up to about 30–90 times faster than LASSO, and aLTS is almost 3–8 times faster than the time for LASSO. As aLTS does not have any information about the number of outliers existed in the dataset and should estimate the number of outliers automatically, its computation cost is reasonably more expensive compared with iHT and iLTS.

3.2 Real-world Data

Two crowdsourcing real-world datasets are adopted in this subsection. Since there is no ground-truth for outliers in real-world datasets, we can not compute ***precision*** and ***recall*** as in the simulated data to evaluate the performance of the methods. Therefore, we inspect the outliers returned by four methods and compare them with the whole data to see whether they are reasonably good outliers or not.

The first dataset PC-VQA, which is collected by [26], contains 38,400 pairwise comparisons of the LIVE dataset [22] from 209 random raters. The paired comparison data in this dataset is complete and balanced. Take reference (a) in the PC-VQA dataset as an illustrative example (other reference videos exhibit similar results). The number of outliers estimated by aLTS is used for LASSO/iHT/iLTS to choose the regularization parameter and select the outliers.

Outliers detected by these methods are shown in the paired comparison matrix in Table 2. The paired comparison matrix is constructed as follows (Table 3 is constructed in the same way). For each video pair $\{i, j\}$, let n_{ij} be the number of comparisons for items i and j, among which a_{ij} raters agree that the quality of item i is better than item j (a_{ji} carries the opposite meaning). So $a_{ij} + a_{ji} = n_{ij}$ if no tie occurs. In the PC-VQA dataset, $n_{ij} = 32$ for any video pair $\{i, j\}$. The order of the video IDs in the matrix is arranged such that the global ranking score calculated by the least squares problem with all the comparisons is decreasing (from high to low). The number of outliers estimated by aLTS from this reference video is 716. So we choose the parameter for LASSO/iHT/iLTS to detect 716 outliers, and the exact number of outliers returned by LASSO/iHT/iLTS is 718, which is slightly larger than 716.

Table 2: Paired comparison matrices of reference (a) in PC-VQA dataset. Red numbers are overlapping outliers obtained by LASSO, iHT, iLTS, and aLTS. Open blue circles are those obtained by LASSO/iHT/iLTS but not aLTS, while filled blue circles are those obtained by aLTS but not LASSO/iHT/iLTS.

Video ID	1	9	10	13	7	8	11	14	15	3	12	4	16	5	6	2
1	0	22	29	30	30	29	29	29	30	28	29	32	32	31	32	31
9	10	0	22	20	14	23	23	25	29	29	32	30	29	30	29	31
10	3	10	0	22	11	21	29	23	31	27	31	30	32	30	32	31
13	2	12	10	0	18	22	23	27	31	28	29	29	29	25	27	28
7	2	18	21	14	0	21	14	16	28	23	31	25	19	27	26	28
8	3	9	11	10	11	0	25	14	28	25	29	27	24	25	28	32
11	3	9	3	9	18	7	0	22	27	26	26	30	30	27	27	31
14	3	7	9	5	16	18	10	0	28	27	18	29	29	26	28	29
15	2	3	1	1	4	4	5	4	0	25	20	22	26	25	29	24
3	4	3	5	4	9	7	6	5	7	0	11	⓯	26	24	29	28
12	3	0	1	3	1	3	6	14	12	21	0	16	20	24	26	26
4	0	2	2	3	7	5	2	3	10	⓱	16	0	15	26	27	30
16	0	3	0	3	13	8	2	3	6	6	12	17	0	22	24	28
5	1	2	2	7	5	7	5	6	7	8	8	6	10	0	26	27
6	0	3	0	5	6	4	5	4	3	3	6	5	8	6	0	21
2	1	1	1	4	4	0	1	3	8	4	6	2	4	5	11	0

Table 3: Paired comparison matrices of reference (c) in PC-IQA dataset. Red numbers, open blue circles, and filled blue circles carry the same meanings as in Table 2.

Image ID	1	8	16	2	3	11	6	12	9	14	5	13	7	10	15	4
1	0	13	9	16	19	12	15	13	14	14	14	17	16	17	16	16
8	6	0	8	7	8	5	13	7	7	8	19	8	15	9	12	15
16	4	0	0	9	11	9	8	15	3	18	16	17	12	7	21	18
2	5	5	6	0	8	9	10	11	7	14	13	14	14	13	14	15
3	3	4	6	7	0	6	11	9	10	16	12	15	14	14	18	13
11	4	6	3	5	6	0	❺	3	5	6	21	5	11	7	12	18
6	0	2	7	4	2	⑦	0	12	12	7	22	15	17	13	13	17
12	3	4	1	4	4	3	1	0	8	15	18	12	9	8	13	17
9	1	3	3	5	1	3	1	0	0	5	18	10	14	9	7	16
14	0	0	1	0	0	3	7	2	1	0	14	15	10	8	17	19
5	0	0	0	0	0	0	0	0	0	1	0	14	19	19	15	17
13	0	0	0	0	0	0	0	0	0	0	6	0	5	7	17	16
7	0	0	0	0	0	0	0	0	0	0	0	5	0	8	9	18
10	0	0	0	0	0	0	0	0	0	0	0	2	2	0	❸	11
15	0	0	0	0	0	0	0	0	0	0	0	0	0	⑤	0	11
4	0	0	0	0	0	0	0	0	0	0	0	1	0	6	6	0

The outliers detected by these methods are mainly distributed in the lower left corner of this matrix, which implies that the outliers are those comparisons with large deviations from the global ranking scores by LS. It is easy to see that outliers returned by LASSO, iHT, iLTS, and aLTS are almost the same except on one pair (ID = 3 and ID = 4). In this dataset, 15 raters agree that the quality of ID = 3 is better than that of ID = 4, while 17 raters have the opposite opinion. LASSO, iHT, and iLTS return the same results which tend to choose comparisons with large deviations from the global ranking scores as outliers. So these three treat the 17 comparisons preferring ID = 4 as outliers because ID = 3 ranks above ID = 4. However, aLTS prefers to choose the minority as outliers and treats the 15 comparisons preferring ID = 3 as outliers. Such a small difference only leads to a local order change of ID = 3 and ID = 4. Therefore the ranking algorithms are stable.

The global ranking scores of the four algorithms, namely LASSO, iHT, iLTS, and aLTS are shown in Table 4(a). For the ease of seeing the differences on global rating scores after outlier detection, we also report the results obtained by LS which has been used in [24, 26] to derive ranking scores in subjective multimedia assessments. After the detected

Table 4: Comparison of different rankings. Five ranking methods are compared with the integer representing the ranking position and the number in parentheses representing the global ranking score returned by the corresponding algorithm.

(a) Ref (a) in the PC-VQA

ID	LS	LASSO/iHT/iLTS	aLTS
1	1 (0.7930)	1 (0.9123)	1 (0.9129)
9	2 (0.5312)	2 (0.7537)	2 (0.7539)
10	3 (0.4805)	3 (0.6317)	3 (0.6322)
13	4 (0.3906)	4 (0.5522)	4 (0.5524)
7	5 (0.2852)	5 (0.4533)	5 (0.4537)
8	6 (0.2383)	6 (0.3159)	6 (0.3163)
11	7 (0.2148)	7 (0.2113)	7 (0.2120)
14	8 (0.1641)	8 (0.1099)	8 (0.1103)
15	9 (-0.1758)	9 (-0.1024)	9 (-0.1029)
3	10 (-0.2227)	11 (-0.3195)	12 (-0.3999)
12	11 (-0.2500)	10 (-0.2149)	10 (-0.2158)
4	12 (-0.2930)	12 (-0.4054)	11 (-0.3252)
16	13 (-0.3633)	13 (-0.5311)	13 (-0.5332)
5	14 (-0.4414)	14 (-0.6573)	14 (-0.6568)
6	15 (-0.6289)	15 (-0.8054)	15 (-0.8057)
2	16 (-0.7227)	16 (-0.9046)	16 (-0.9042)

(b) Ref (c) in the PC-IQA

ID	LS	LASSO/iHT/iLTS	aLTS
1	1 (0.7575)	1 (0.9015)	1 (0.9022)
8	2 (0.5670)	2 (0.7088)	2 (0.7129)
16	3 (0.5124)	3 (0.6472)	3 (0.6504)
2	4 (0.4642)	4 (0.5242)	4 (0.5248)
3	5 (0.4423)	5 (0.4119)	5 (0.4148)
11	6 (0.3277)	6 (0.2592)	7 (0.1763)
6	7 (0.3128)	7 (0.2515)	6 (0.3124)
12	8 (0.2423)	8 (0.1209)	8 (0.1261)
9	9 (0.1453)	9 (0.0043)	9 (0.0069)
14	10 (-0.0455)	10 (-0.1274)	10 (-0.1243)
5	11 (-0.3376)	11 (-0.3205)	11 (-0.3214)
13	12 (-0.4785)	12 (-0.4621)	12 (-0.4560)
7	13 (-0.5396)	13 (-0.5515)	13 (-0.5494)
10	14 (-0.7486)	14 (-0.7005)	15 (-0.7485)
15	15 (-0.7658)	15 (-0.7511)	14 (-0.7106)
4	16 (-0.8559)	16 (-0.9163)	16 (-0.9166)

outliers are removed, the orders of some competitive videos are changed. LASSO, iHT, iLTS, and aLTS all think that ID = 12 has better performance than ID = 3 and ID = 4. However, the orders of ID = 3 and ID = 4 are exchanged in aLTS and LASSO/iHT/iLTS, because they choose different preference directions as outliers.

The second dataset PC-IQA [25] is incomplete and imbalanced. This dataset contains 15 reference images and 15 distorted versions of each reference image. So the total number of images is 240. These images come from two publicly available datasets: LIVE [22] and IVC [16]. Totally, 186 raters, each of whom performs a varied number of comparisons via Internet, provide 23,097 pairwise comparisons.

Tables 3 and 4(b) show the comparable experimental results of LASSO, iHT, iLTS, and aLTS on reference image (c) (other reference images exhibit similar results). The number of outliers estimated by aLTS is 173, so we choose the parameter of LASSO/iHT/iLTS to detect 173 outliers. The exact number of outliers returned by LASSO/iHT/iLTS is 177, which is slightly larger than 173. We can see that the difference of the detection between LASSO/iHT/iLTS and aLTS happens on two pairs: 1) ID = 6 and ID = 11; 2) ID = 10 and ID = 15. Same as in the last experiment, these methods differ in outlier detection for highly comparable pairs. aLTS prefers to choose the minority in paired comparisons, i.e., the 5 comparisons preferring ID = 11 over ID = 6 and the 3 comparisons preferring ID = 10 over ID = 15, while LASSO/iHT/iLTS selects comparisons with largest deviations from global ranking scores even when the votings are in majority. Such a difference leads to a local order change of involved items only.

3.3 Discussion

As we have seen in the numerical experiments, LASSO, iHT, iLTS, and aLTS mostly find the same outliers, and when they disagree, aLTS tends to choose the minority and LASSO/iHT/iLTS prefer to choose comparisons with large deviations from the global ranking scores even when the votings are in majority. When outliers consist of minority voting as in simulated experiments, aLTS performs better than LASSO, iHT, and iLTS. This can also be explained from the algorithm. We choose a small underestimation for the number of outliers, and increase this estimation until there is no outliers in the remaining comparisons. The parameter $\beta_2 > 1$ is chosen to be small so we will not overestimate the number of outliers too much.

Finally, we would like to point out that subject-based outlier detection can be a straightforward extension from our proposed algorithms. From the detection results, one may evaluate the reliability of one rater based on all the comparisons from the rater and remove all the comparison from unreliable raters.

4 CONCLUSIONS

In this paper, we proposed fast algorithms for outlier detection with nonconvex optimization and robust ranking in QoE evaluation. Specifically, for known K, the proposed iHT and iLTS could provide us almost the same performance compared with LASSO, and the computational speed can achieve up to 90 times faster than LASSO. For unknown K, we proposed an adaptive method called aLTS which could estimate the number of outliers and detect them without any prior information about the number of outliers in the dataset. This method is nearly 3–8 times faster than LASSO. The effectiveness and efficiency of the proposed methods is demonstrated on both simulated examples and real-world applications. The small distinctions between these four methods indicate that aLTS prefers to choosing minority voting data as outliers, while the LASSO, iHT, and iLTS select the comparisons with largest deviations from the global ranking score as outliers even when they are in majority. In both cases, the global rankings obtained are stable. In summary, we expect that the proposed outlier detection methods for QoE will be helpful tools for people in the multimedia community exploiting crowdsourceable paired comparison data for robust ranking.

5 ACKNOWLEDGMENTS

The research of Qianqian Xu was supported in part by National Key Research and Development Plan (No. 2016YFB0800403), National Natural Science Foundation of China (No. 61672514, 61422213, U1636214, 61390514, 61572042), Beijing Natural Science Foundation (4172068), Key Program of the Chinese Academy of Sciences (No. QYZDB-SSW-JSC003). The research of Ming Yan was supported in part by NSF grant under DMS-1621798. The research of Qingming Huang was supported in part by National Natural Science Foundation of China: U1636214, 61650202, 61332016 and 61620106009, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013. The research of Yuan Yao was supported in part by National Basic Research Program of China under grant 2015CB85600, 2012CB825501, and NSFC grant 61370004, 11421110001 (A3 project), as well as grants from Baidu, Microsoft Research-Asia, HKUST, and Tencent AI Lab.

REFERENCES

[1] *Methods for Subjective Determination of Transmission Quality*. ITU-T Rec. P.800, 1996.

[2] V. Barnett and T. Lewis. *Outliers in statistical data*, volume 3. Wiley New York, 1994.

[3] M. Breunig, H. Kriegel, R. Ng, and J. Sander. Lof: identifying density-based local outliers. In *ACM Conference on Management of Data*, volume 29, pages 93–104, 2000.

[4] E. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[5] K.-T. Chen, C.-C. Wu, Y.-C. Chang, and C.-L. Lei. A crowdsourceable QoE evaluation framework for multimedia content. pages 491–500. ACM Multimedia, 2009.

[6] A. Eichhorn, P. Ni, and R. Eg. Randomised pair comparison: an economic and robust method for audiovisual quality assessment. pages 63–68. International Workshop on Network and Operating Systems Support for Digital Audio and Video, 2010.

[7] S. Foucart. Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. In *Approximation Theory XIII: San Antonio 2010*, pages 65–77. Springer, 2012.

[8] B. Gardlo, M. Ries, and T. Hossfeld. Impact of screening technique on crowdsourcing QoE assessments. In *Radioelektronika, 2012 22nd International Conference*, pages 1–4, 2012.

[9] T. Hossfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold, and P. Tran-Gia. Best practices for QoE crowdtesting: QoE assessment with crowdsourcing. *IEEE Transactions on Multimedia*, 16(2):541–558, 2014.

[10] P. Huber. *Robust Statistics*. New York: Wiley, 1981.

[11] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[12] T. Johnson, I. Kwok, and R. Ng. Fast computation of 2-dimensional depth contours. In *ACM Knowledge Discovery and Data Mining*, pages 224–228, 1998.

[13] C. Keimel, J. Habigt, and K. Diepold. Challenges in crowd-based video quality assessment. In *International Workshop on Quality of Multimedia Experience*, pages 13–18, 2012.

[14] E. Knorr and R. Ng. Finding intensional knowledge of distance-based outliers. In *International Conference on Very Large Data Bases*, volume 99, pages 211–222, 1999.

[15] E. Knorr, R. Ng, and V. Tucakov. Distance-based outliers: algorithms and applications. *International Journal on Very Large Data Bases*, 8(3-4):237–253, 2000.

[16] P. Le Callet and F. Autrusseau. Subjective quality assessment irccyn/ivc database, 2005. http://www.irccyn.ec-nantes.fr/ivcdb/.

[17] A. Leroy and P. Rousseeuw. Robust regression and outlier detection. *John Wiley & Sons*, 1987.

[18] P. Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

[19] P. Rousseeuw and V. Yohai. Robust regression by means of S-estimators. In *Robust and nonlinear time series analysis*, pages 256–272. 1984.

[20] R. Schatz, T. Hoßfeld, L. Janowski, and S. Egger. From packets to people: Quality of experience as new measurement challenge. In *Data Traffic Monitoring and Analysis: From Measurement, Classification and Anomaly Detection to Quality of Experience*. Springer's Computer Communications and Networks series, 2012.

[21] Y. She and A. Owen. Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, 106(494):626–639, 2011.

[22] H. Sheikh, Z.Wang, L. Cormack, and A. Bovik. LIVE image & video quality assessment database, 2008.

[23] C.-C. Wu, K.-T. Chen, Y.-C. Chang, and C.-L. Lei. Crowdsourcing multimedia QoE evaluation: A trusted framework. *IEEE Transactions on Multimedia*, 15(5):1121–1137, 2013.

[24] Q. Xu, Q. Huang, T. Jiang, B. Yan, W. Lin, and Y. Yao. HodgeRank on random graphs for subjective video quality assessment. *IEEE Transactions on Multimedia*, 14(3):844–857, 2012.

[25] Q. Xu, Q. Huang, and Y. Yao. Online crowdsourcing subjective image quality assessment. pages 359–368. ACM Multimedia, 2012.

[26] Q. Xu, T. Jiang, Y. Yao, Q. Huang, B. Yan, and W. Lin. Random partial paired comparison for subjective video quality assessment via HodgeRank. pages 393–402. ACM Multimedia, 2011.

[27] Q. Xu, J. Xiong, Q. Huang, and Y. Yao. Robust evaluation for quality of experience in crowdsourcing. In *ACM Multimedia*, pages 43–52, 2013.

[28] Q. Xu, J. Xiong, Q. Huang, and Y. Yao. Online HodgeRank on random graphs for crowdsourceable QoE evaluation. *IEEE Transactions on Multimedia*, 16(2):373–386, 2014.