This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

From Social to Individuals: a Parsimonious Path of Multi-level Models for Crowdsourced Preference Aggregation

Qianqian Xu, Jiechao Xiong, Xiaochun Cao, Qingming Huang, and Yuan Yao

Abstract—In crowdsourced preference aggregation, it is often assumed that all the annotators are subject to a common preference or social utility function which generates their comparison behaviors in experiments. However, in reality annotators are subject to variations due to multi-criteria, abnormal, or a mixture of such behaviors. In this paper, we propose a parsimonious mixed-effects model, which takes into account both the fixed effect that the majority of annotators follows a common linear utility model, and the random effect that some annotators might deviate from the common significantly and exhibit strongly personalized preferences. The key algorithm in this paper establishes a dynamic path from the social utility to individual variations, with different levels of sparsity on personalization. The algorithm is based on the Linearized Bregman Iterations, which leads to easy parallel implementations to meet the need of large-scale data analysis. In this unified framework, three kinds of random utility models are presented, including the basic linear model with L_2 loss, Bradley-Terry model, and Thurstone-Mosteller model. The validity of these multi-level models are supported by experiments with both simulated and real-world datasets, which shows that the parsimonious multi-level models exhibit improvements in both interpretability and predictive precision compared with traditional HodgeRank.

Index Terms—Preference Aggregation; HodgeRank; Mixed-Effects Models; Linearized Bregman Iterations; Personalized Ranking; Position Bias.

1 Introduction

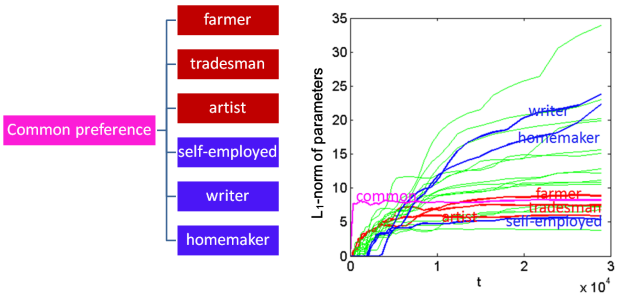
With the Internet and its associated explosive growth of information, individuals today in the world are facing with the rapid expansion of multiple choices (e.g., which book to buy, which hotel to book, etc.). All of these examples yield comparisons without explicitly revealing an underlying preference or utility function. That is, only partial ordered choices subject to the preference are observed instead of the whole utility function, especially the paired comparisons that all partial ordered choices can be converted into. Therefore the aggregation of incomplete comparison data to reveal the global preference function has been one important topic in the last decades.

In recent years, researchers in this odyssey usually take three approaches: (i) common consensus, that assumes all users' choices are stochastic revelation of a common global preference or utility function on candidates; (ii) collaborative filtering for personalized ranking, which often assumes different users have correlated preference functions represented by some low rank rating matrices; (iii) mixture of random utility models [13], [29], [31], that assumes the personal choice comes from one of a small set of underlying random utility models which are yet unknown. Regarding common consensus pursuit, there has been a large volume of studies from the social choice theory to modern "rank aggregation" in computer science [3], [7], [8], [11], [19], [28], [29], [35], [54], on how to consistently aggregate the pairwise comparisons into a global consensus ranking that summarizes the preference of all users. On the other hand, low rank models of collaborative filtering [27], [40], [41], [53] assume that there exist a small number of underlying intrinsic utility functions such that every individual's personalized preference is a linear combination of these intrinsic utility functions, using nuclear norm as a penalty, while mixture models can be consistently recovered using tensor moment matching methods [31]. However, few work takes the wide spectrum by considering both the social preference and individual variations simultaneously. In other words, they do not take into account the multi-level hierarchies from

- *Q. Xu and X. Cao are with State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, (email: xuqianqian@iie.ac.cn; caoxiaochun@iie.ac.cn).*
- *J. Xiong is with BICMR-LMAM-LMEQF-LMP, School of Mathematical Sciences, Peking University, Beijing 100871 & Tencent AI Lab, Shenzhen 518057, China, (email: jcxiong@tencent.com).*
- *Q. Huang is with the University of Chinese Academy of Sciences & Institute of Computing Technology of Chinese Academy of Sciences, Beijing 100190, China, (email: qmhuang@ucas.ac.cn).*
- *Y. Yao is with Department of Mathematics, Hong Kong University of Science and Technology, Hong Kong, (email: yuany@ust.hk).*

A publicly available package (i.e. datasets and code) to reproduce our experiments can be downloaded from https://github.com/qianqianxu010/PAMI2017

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Common preference with six representative occupation group preferences (b) Regularization paths of all 21 group preferences

Fig. 1: A two-level preference learning in MovieLens: (a) the common preference with six representative occupation group preference; (b) the purple is the common preference, the remaining 21 paths represent the occupation group preferences, the red are the three groups with most distinct preferences from the common, the blue are the three groups with most similar preferences to the common, and the green ones are the others.

social choice to individuals.

To see the nature of such hierarchies in preference learning, let's consider the movie rating from MovieLens dataset for example. Fig.1 shows a two-level movie preference functions learned from this dataset: the common preference and 21 group preferences. Fig.1 (a) illustrates this two-level hierarchical model with six representative groups, among which farmer, tradesman, artist are the top 3 groups exhibiting a large deviation from the common preferences, while self-employed, writer, homemaker are those showing similar preference with the common. Such results suggest that the main consuming groups of this website include homemaker, writer, and un-employed, who have more freedom to spend their time compared with other occupations. A coarse-grained model may just consider the common preference for all the users, or a refined model may incorporate these group variations to reflect diversity, while a further refined case may consider diversity in individual level. Fig.1 (b) shows the group preference diversity using the methodology proposed in this paper. The purple curve represents the common preference, while the remaining 21 curves there represent the 21 occupation group preferences in regularization paths, of which the earlier popping up to be nonzero, the more salient distinction is the group preference from the common. At different locations of t-axis, models of different diversity levels can be chosen.

In addition to the intrinsic preference diversity among users, there are abnormal behaviours of participants in crowdsourcing experiments due to diverse environment. Even they might share the same preference or utility function in making choices, they might suffer various disturbances during the experiments. For example, i) one typically clicks one side more often than another. As some pairs are highly confusing or annotators get too tired, in these cases, some annotators tend to click one side hoping to simply raise their record to receive more payment; while for pairs with substantial differences, they click as usual. ii) some extremely careless annotators, or robots pretending to be human annotators, actually do not look at the instances and click one side all the time to quickly receive payment for work. Such a kind of behavior is called the annotator's position bias which has been studied in [10].

These examples above suggest us that we have to take into account of user or annotator specific variations in a crowdsourced preference aggregation task. In this paper, we propose *a simple dynamic scheme that can learn multi-level utility models from the social common preference to individual diversity in a unified spectrum*, adapted to different statistical models (e.g. linear, Bradley-Terry, and Thurstone-Mosteller etc).

As the classical social choice theory [1] points out, preference aggregation toward a global consensus is doomed to meet the conflicts of interests. *What is a suitable way to quantitatively analyze the conflicts of interests*?

In this paper, we are inspired by the Hodge-theoretic approach proposed in [19] which decomposes the pairwise comparison data into three orthogonal components: the global consensus ranking, the local inconsistency as triangular cycles, and the global inconsistency as harmonic cycles. Instead of merely extracting from the data the global ranking component, often called *HodgeRank*, the latter two are both cycles, collectively decoding all the conflicts of interests in the data. To decipher the sources of the conflicts of interests, we further decompose the cycles by considering two types of annotator-specific variations here: annotator's personalized preference deviations from the common ranking which characterize multi-criteria in comparisons, and annotator's position bias which deteriorates the quality of data. This results in a linear mixed-effects extension of HodgeRank, called Mixed-Effects HodgeRank here. Such a principle can be applied to various generalized linear models that will be studied in this paper.

To initiate a task of crowdsourced preference aggregation, we usually assume the majority of participants share a common preference interest and behave rationally, while deviations from that exist but are sparse. So a parsimonious model is desired in this paper, with sparsity structure on personalized preference deviations and position biases. Due to the unknown amount of such sparse random effects in reality, it is natural to pursue a family of parsimonious models at a variety levels of sparsity. Algorithmically we developed the Linearized Bregman Iterations (LBI) as discretized Inverse Scale Space method in our setting, which is a simple iterative procedure generating a sequence of parsimonious models, evolving from the common global ranking in HodgeRank, to annotator's

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

personalized ranking with a fully parametric model that might overfit the data. As the algorithm iterates, typically it appears early the large deviations in a personalized preference or abnormal behaviour, and the annotators who follow the common show at a later stage. In practice when the number of participants is large and sample size is relatively small, early stopping regularization is needed to prevent the overfitting in full model. Due to the algorithmic simplicity, it allows an easy (synchronized) parallel implementation to meet the need of large-scale data analysis.

As a summary, our main contributions in this new framework are highlighted as follows:

(A) A linear mixed-effects extension of HodgeRank including both the fixed effect of common ranking, and the random effects such as annotator's preference deviations and position bias, which can be easily extended to generalized linear models including Bradley-Terry (BT) and Thurstone-Mosteller (TM) models that improve the efficiency for binary comparison data than the basic linear model (associated with the L_2 loss).

(B) A path of parsimonious estimates of the preference deviation and position bias at different sparsity levels, based on Linearized Bregman Iterations as a discretization of Inverse Scale Space method, which allows a simple synchronized parallelization for an almost linear speed-up.

This paper is an extension of our conference paper [52], where we proposed a basic linear mixed-effect model which not only can derive the common preference on population-level, but also can estimate an annotator's large preference/utility deviation in an individual-level, as well as an abnormal annotator's position bias. However, there are some limitations in this work. First, it does not aim to predict the preferences (social or individual) based on features of new users and/or new alternatives. Such featured data are ubiquitous in E-commerce etc., such as recommendations of books, movies, and restaurants, based their styles and categories of users. To learn such preference or ranking functions with predictive power on unseen products, a feature representation of the candidates in comparison must be used as model input in addition to the local ranking orders. Second, other types of models are not studied, such as generalized linear models which are particularly efficient for discrete choice data. In current new version, we propose *a unified framework that includes various generalized linear models which learn both the social preference functions based on features of alternatives to-be-compared and personalized utility functions conditioning on user categories*. Such a model with at least two levels of diversity, enables us to simultaneously learn a coarse-grained social social function together with fine-grained personalized rankings, equipped with prediction power for the choices of new users on new alternatives. In this paper, we shall see that the Linearized Bregman Algorithm can be adapted to all these generalized linear models with fast and parallel path algorithms, and particularly enjoys the improved statistical precision of generalized linear models for binary comparisons in real world datasets, without losing the algorithmic simplicity in basic linear model.

The remainder of this paper is organized as follows. Sec.2 contains a review of related works. Then we systematically introduce the methodology for parsimonious mixed-effects HodgeRank estimation in Sec.3. Extensive experimental validation based on one simulated and three real-world crowdsourced datasets are demonstrated in Sec.4. Finally, Sec.5 presents the conclusive remarks.

2 Related Work

Statistical preference aggregation, in particular ranking or rating from pairwise comparisons, is a classical problem which can be traced back to the 18^{th} century. Various methods have been studied for this problem, including the Borda count [11], maximum likelihood method such as the Bradley-Terry model [8], rank centrality (PageRank/MC3) [7], [30], and most recently, HodgeRank [19].

HodgeRank, as an application of combinatorial Hodge theory to the preference or rank aggregation problem from pairwise comparison data, was first introduced in [19], inspiring a series of studies in computer science [34], [36], [50] and game theory [5], in addition to traditional applications in fluid mechanics [6] and computer vision [55], etc. It is a general framework to decompose paired comparison data on graphs, possibly imbalanced (where different candidate pairs may receive different number of comparisons) and incomplete (where every voter may only give partial comparisons), into three orthogonal components (gradients, local cycles, and harmonic cycles). In these components HodgeRank not only provides us a mean to determine a global ranking from paired comparison data under various statistical models (e.g., Uniform, Thurstone-Mosteller, Bradley-Terry, and Angular Transform), but also measures the inconsistency of the global ranking obtained. The inconsistency shows the validity of the ranking obtained and can be further studied in terms of its geometric scale, namely whether the inconsistency in the ranking data arises locally or globally. Local inconsistency can be fully characterized by triangular cycles, while global inconsistency involves cycles consisting nodes more than three (harmonic cycles), which may arise due to data incompleteness and once presented with a large component indicates some serious conflicts in ranking data.

In [38], [45], it shows that under a natural statistical model, where pairwise comparisons are drawn

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

randomly and independently from some underlying probability distribution, the rank centrality (PageRank) and HodgeRank algorithms both converge to an optimal ranking under a "time-reversibility" condition. However, PageRank is only able to aggregate the pairwise comparisons into a global ranking over the items. HodgeRank not only provides us a mean to determine a global ranking under various statistical models, but also measures the inconsistency of the global ranking obtained. Exploiting the random graphs, we can efficiently control the global inconsistency via topology of random clique complexes [49] as well as the sampling efficiency [37].

However, all of these methods have a major drawback: they aim to find *one* global ranking thus cannot analyze the conflicts of interests or discrepancies across users. In HodgeRank [19], such conflicts are encoded in the components of cyclic rankings, which are not user-specific. On the other hand, in crowdsourcing scenarios, users may vote following multi-criteria or under different environments that contribute to the preferential diversity. Deciphering such behaviors becomes necessary for a better exploit of crowdsourcing data.

Recently, some personalized ranking methods arose from the standard collaborative filtering (CF) approach that is based on matrix factorization [40], [41], [53]. The key idea behind them is to find a low rank user rating matrix via nuclear norm regularization such that every user's utility is a linear combination of such low-rank ratings. However such models are not a natural fit in crowdsourcing scenarios where the majority of voters share some common preference while some annotators might deviate from that significantly.

Beyond the CF approach, there are various techniques to model annotators' abnormal behaviors in general crowdsourcing [18], [21], [23]–[25], [43], [46], [56]–[59], etc. The basic idea of these work is to characterize user quality using some probabilistic behavior models. The models roughly lie in two categories [43]: either a single parameter is associated with each user's quality indicating the probability that the annotator correctly answers a task [17], [26], or a general confusion matrix is used for each user as extensions from the classic work of Dawid and Skene (DS) [9], [39], [48]. In particular, [21] considers task dependent user quality parameters or confusion matrices such that the majority follows the common parameter while some may deviate from that with personalized parameters; on the other hand, [46] directly exploits the correlations between user confusion matrices to discover hidden groups of users.

While these methods can model the quality of the workers in general crowdsourcing experiments for label aggregation, they lack the consideration for peculiarity in crowdsourced preference aggregation where every user may vote following some utilities. For example, in pairwise comparisons, the confusion matrix approach will lead to an adversarial mixture ranking model [44], where every voter follows a mixture of rational behavior by voting according to the common ranking model and abnormal behavior by voting according to its adversarial ranking. However, voters are not necessarily adversarial; for example, robot clickers on one side can be captured by position bias in our model and random clickers can be captured by his/her deviations in personalized ranking, both of which are clearly not adversarial voters. Therefore the models with quality parameter or confusion matrix above are coarse-grained models in crowdsourced ranking, insufficient to capture the preferential diversity. In this paper, we are inspired by the HodgeRank approach, and propose a parsimonious multilevel model for personalized rankings that decipher conflicts of interests but are not necessary adversarial, so may capture a wider or more refined preferential diversity in crowdsourced rank aggregation than previous models.

3 METHODOLOGY

In this section, we systematically introduce the methodology for parsimonious mixed-effects HodgeRank estimation. Specifically, we first start from introducing the proposed mixed-effects model based on HodgeRank, in which three kinds of random utility models are presented including the basic linear model with L_2 loss, Bradley-Terry model, and Thurstone-Mosteller model, etc. Then we present a simple iterative algorithm called Linearized Bregman Iterations to generate paths of parsimonious models at different sparsity levels, followed by Synchronized Parallel LBI to meet the need of large-scale data analysis. Finally, early stopping regularization is discussed in the end of this section.

3.1 Mixed-Effects HodgeRank on Graphs

Suppose there are n alternatives or items to be ranked, represented by n data points with a feature matrix $\Phi = [\phi_i^T]_{i=1}^n \in \mathbb{R}^{n\times d}$, where ϕ_i is a d-dimensional feature vector representing item i. The pairwise comparison labels collected from users can be naturally represented as a directed comparison graph $G = (V; E)$. Let $V = \{1, 2, \ldots, n\}$ be the vertex set of n items and $E = \{(u, i, j) : i, j \in V, u \in U\}$ be the set of edges, where U is the set of all users who compared items. User u provides his/her preference between choice i and j, such that $y_{ij}^u > 0$ means u prefers i to j and $y_{ij}^u \leq 0$ otherwise. Hence we may assume $y : E \to R$ with skew-symmetry (orientation) $y_{ij}^u = -y_{ji}^u$. The magnitude of y_{ij}^u can represent the degree of preference and it varies in applications. The simplest setting is the binary choice, where $y_{ij}^u = 1$ if u prefers i to j and $y_{ij}^u = -1$ otherwise. In applications, users are often categorized by their classifications, such as occupations and ages, hence y_{ij}^u may be a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

summary statistics of all the pairwise comparisons between i and j among the same category of users.

The general purpose of preference aggregation is to look for a global score $\theta\colon V \to \mathbb{R}$ such that

$$\min_{\theta\in\mathbb{R}^{|V|}} L(\theta) := \sum_{i,j,u} \omega_{ij}^u l(\theta_i - \theta_j, y_{ij}^u), \tag{1}$$

where $l(a,b)\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}$ is a loss function, ω_{ij}^u denotes the confidence weights on $\{i,j\}$ made by rater u (for simplicity, assumed to be $\omega_{ij}^u = 1$ for the provided voting data), and θ_i (θ_j) represents the global ranking score of item i (j, respectively). In HodgeRank, one benefits from the use of square loss $l(a,b) = (a-b)^2$ which leads to fast algorithms to find optimal global ranking θ, which becomes one component of a general orthogonal decomposition of paired comparison data [19], i.e.

$$y = global\ ranking \oplus cycles,$$

where the component *cycles* can be further decomposed into

$$cycles = local\ cycles \oplus global\ cycles.$$

Local cycles are triangular cycles, e.g. $i \succ j \succ k \succ i$; while global cycles, also called harmonic cycles, are loops involving nodes more than three (e.g. $i \succ j \succ k \succ \ldots \succ i$) and typically traversing all nodes in the graph. These cycles may arise due to conflicts of interests in ranking data. Therefore to analyze the statistical models of cycles is crucial to understand the conflicts of interests.

In crowdsourcing scenarios, the conflicts of interests are mainly due to two kinds of sources: the multi-criteria adopted by different annotators when they compare items in V; the abnormal behavior of annotators in the experiments, e.g. simply clicking one side of the pair when they got bored, tired, or distracted. From this viewpoint, the source of such cycles in HodgeRank are usually caused by the personalized ranking, position bias, and stochastic noise.

To be specific, together with the global ranking component in HodgeRank, we consider the following linear mixed-effects model for annotator's pairwise ranking:

$$y_{ij}^u \sim F((\phi_i^T\eta + \phi_i^T\xi^u) - (\phi_j^T\eta + \phi_j^T\xi^u) + \gamma^u), \tag{2}$$

- η is the common preference parameter such that the inner product with the i^{th} feature ϕ_i, $\theta_i := \phi_i^T\eta$ gives the common preference score on item i, as a fixed effect;
- ξ^u is the user's preference deviation parameter from the common consensus such that $\theta_i^u := \phi_i^T(\eta+\xi^u)$ becomes user u's personalized preference score, as a random effect;
- γ^u is an annotator's position bias, which captures the careless behavior by clicking one side during the comparisons;
- The distribution F can be arbitrary cumulative distribution function.

Here η is population-level parameter which indicates some common coefficient weight vector of the feature. In reality, as the preference vary greatly across different types of users, we allow each type of user to have their personalized parameters. These personalized parameters can be obtained by adding some random effects ξ^u to the population parameter η, representing personalized deviations from the population behavior. Moreover, γ^u measures an annotator's position bias, i.e. the tendency of u always clicking one side in paired comparison experiments. Under the random design of pairwise comparison experiments, a candidate should be placed on the left or the right randomly, so the position should not affect the choice of a careful annotator. However, some annotator might get confused, tired or distracted in experiments, such that he/she always clicks one side during some periods in experiments, which can be detected by such $\gamma^u \neq 0$ [51].

Considering the variety of applications, this model may incorporate several types of feature matrices $\Phi = [\phi_i^T]_{i\in V}$, motivated but are not limited to the following examples.

- Φ **is an identity matrix.** For example, in world-college ranking, V consists of colleges to be ranked, and $y_{ij}^u = 1$ indicates user u prefers college i to j. In this scenario, we do not have the features of each college but only the pairwise comparisons obtained from users.
- Φ **is low-level (or deep) visual features.** For example, in music ratings, ϕ_i can be the low-level audio features extracted from each audio frame (spectrum power, Zero Crossing Rate, intensity, bandwidth, pitch and MFCC, etc).
- Φ **is categorical type.** For example, in movie ratings, ϕ_i can be the genres of movie i, (e.g., Action, Adventure, Animation, Comedy, Drama, etc). Or in dining restaurant ratings, ϕ_i can be the cuisine types (e.g., Bar, Cafe-Coffee-Shop, Cafeteria, Fast-Food, etc) of the restaurant i.

To make the notation clear, let $d^u \in \mathbb{R}^{|E|\times|V|}$ satisfies $d^u\theta(v,i,j) = 1_{(u=v)}(\theta_i - \theta_j)$ and $d = \sum_u d^u$. Let $A \in \mathbb{R}^{|E|\times|U|}$ satisfies $A\gamma(u,i,j) = \gamma^u$. Denote $\theta = \Phi\eta$, $\delta^u = \Phi\xi^u$, and $\beta = [\xi,\gamma]$. Let $X = [d^1\Phi, \ldots, d^{|U|}\Phi, A]$, so $X\beta = \sum_u d^u\Phi\xi^u + A\gamma$.

Different distribution functions F respond to different statistic models. For example, when F is normal function or sub-gaussian function, it indicates data follows the normal distribution or sub-gaussian distribution, which means

$$y_{ij}^u = (\phi_i^T\eta + \phi_i^T\xi^u) - (\phi_j^T\eta + \phi_j^T\xi^u) + \gamma^u + \varepsilon_{ij}^u, \tag{3}$$

or in matrix form

$$y = d\Phi\eta + X\beta + \varepsilon. \tag{4}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

where ε_{ij}^u measures the random noise in sampling which is of zero mean and bounded. For notational simplicity, we abuse the notation y to denote the vector $[y_{ij}^u]$. In this case, the loss function is often the L_2 loss, the negative log-likelihood of Gaussian:

$$L(\eta, \beta) = \frac{1}{2m}\|y - (d\Phi\eta + X\beta)\|_2^2. \tag{5}$$

For robust statistics, one can also adopt L_1 loss [35] or Huber's loss which is equivalent to the L_2 loss with sample-wise sparse γ^u.

For binary comparison data $y_{ij}^u \in \{\pm 1\}$, there is a family of generalized linear model (GLM) in statistics:

$$\begin{aligned} P(y_{ij}^u = 1) &= 1 - P(y_{ij}^u = -1) \\ &= \Psi((\phi_i^T\eta + \phi_i^T\xi^u) - (\phi_j^T\eta + \phi_j^T\xi^u) + \gamma^u) \end{aligned} \tag{6}$$

or

$$\Psi^{-1}(P(y = 1)) = d\Phi\eta + X\beta. \tag{7}$$

where $\Psi(t)$ is a symmetric cumulative distribution function (CDF) whose continuous inverse is well-defined. For example,

1. *Bradley-Terry* model:

$$\Psi(t) = \frac{1}{1 + e^{-t}}. \tag{8}$$

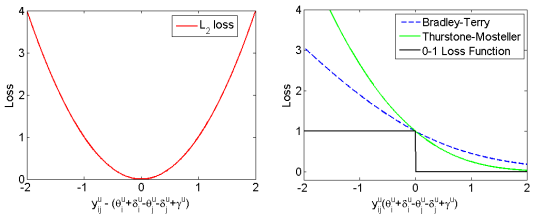
2. *Thurstone-Mosteller* model:

$$\Psi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-x^2/2} dx \tag{9}$$

More models can be found in [8], [19], [49], [50]. We note that in general Hodge theoretical framework for binary pairwise comparison data, one can map binary comparison data into skew-symmetric flows on graphs by $\hat{y} = \Psi^{-1}(P(y_{ij}^u = 1))$ and Hodge decomposition can be applied to such flows [19], [49]. In this paper, for the GLM model (6), the loss function is chosen as the following negative log-likelihood,

$$L(\eta, \beta) = -\frac{1}{m}\sum_{i,j,u} \log \Psi(y_{ij}^u(\phi_i^T(\eta_i + \xi_i^u) - \phi_j^T(\eta_j + \xi_j^u) + \gamma^u)). \tag{10}$$

Here we use the symmetry $\Psi(-t) = 1 - \Psi(t)$. Fig.2 illustrates the comparisons of three losses, including L_2, Bradley-Terry (BT), and Thurstone-Mosteller (T-M), respectively. One can see that as in classifications, Bradley-Terry and Thurstone-Mosteller provide convex surrogates [2] of binary comparison 0-1 loss with a better approximation than the L2 loss. Therefore one should expect that these two models may provide a better efficiency in reducing the pairwise mismatch (Kendall τ-distance) from the observed data, as we shall see later in this paper.



(a) Linear (b) BT & TM

Fig. 2: Comparison of three loss functions.

3.2 Parsimonious Paths of Multi-level Models with Linearized Bregman Iteration

In crowdsourced preference aggregation scenarios with good controls, it is natural to assume a parsimonious model. In such a model, the majority of annotators carefully follows the common behavior governed by the fixed effect parameter θ, while only a small set of annotators might have nonzero personalized deviations and abnormal behavior in position bias. This amounts to assume that parameter δ^u to be group sparse, i.e. δ_i^u vanishes for all i simultaneously, and γ^u to be sparse as well, i.e. zero for most of careful annotators.

Let's consider two representative scenarios:

• When Φ is an identity matrix, $\delta^u = \xi^u$, So such a sparsity pattern motivates us to consider the following penalty function with a mixture of LASSO (L_1) penalty on γ and group LASSO penalty on ξ^u:

$$P(\beta) = \|\gamma\|_1 + \sum_u \|\xi^u\|_2. \tag{11}$$

Remark 1: Usually a normalization factor $\sqrt{n}$ is used before a group lasso penalty $\|\xi^u\|_2$, where n is the group size of ξ^u. But here all the ξ^u have the same group size, and $\|d^u\|_F = \sqrt{2}\|A^u\|_F$, so the column norm of d^u is on average $\frac{\sqrt{2}}{\sqrt{n}}$ times of $\|A^u\|_F$, this basically cancels out the factor $\sqrt{n}$. So here we just use this simple formula.

• When Φ is low-level (or deep) visual features, such a sparsity pattern only needs to assume traditional LASSO (L_1) penalty on both γ and ξ^u.

$$P(\beta) = \|\gamma\|_1 + \sum_u \|\xi^u\|_1. \tag{12}$$

Given the Loss function and Penalty function, the following Linearized Bregman Iterations (LBI) give rise to a sequence of parsimonious (sparse) models:

$$\eta^{k+1} = \eta^k - \alpha\kappa\nabla_\eta L(\eta^k, \beta^k) \tag{13a}$$
$$z^{k+1} = z^k - \alpha\nabla_\beta L(\eta^k, \beta^k), \tag{13b}$$
$$\beta^{k+1} = \kappa \cdot \text{prox}_P(z^{k+1}), \tag{13c}$$

where $\beta^0 = 0$, $\eta^0 = 0$, $z^0 = 0$, k is the iteration index, and the proximal map associated with the penalty

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

function P is given by

$$\text{prox}_P(z) = \arg\min_{v \in R^{(|V|+1)|U|}} \left(\frac{1}{2}\|v - z\|^2 + P(z) \right).$$

Here variable z is an auxiliary parameter used for gradient descent, where by Moreau decomposition $z = \rho + \beta/\kappa, \rho \in \partial P(\beta)$.

The Linearized Bregman Iteration (13) generates a path of global ranking score estimators $\theta^k = \Phi\eta^k$ and sparse estimators for preference deviation and position bias, $\beta^k = (\xi^k, \gamma^k)$. It starts from the null model, and evolves into parsimonious mixed effect models with different levels of sparsity until the full model, often overfitted. To avoid the overfitting, early stopping regularization is required to find an optimal tradeoff between the model complexity and in-sample error. In this paper, we find that cross validation works to find the early stopping time that will be discussed in Sec.3.4.

The Linearized Bregman algorithm was firstly introduced in [32] as a scalable algorithm for large scale image restoration with TV-regularization. It has several advantages than the widely used LASSO-type convex regularizations. First of all, it is simpler than LASSO in generating the sparse regularization paths: instead of a parallel run of several optimization problem over a grid of regularization parameters, a single run of LBI generates the whole regularization path. LBI is thus desired in dealing with big problems.

The main advantage of such a three line algorithm, not only lies in its algorithmic simplicity, but also gives us more statistical precision. In fact, it has been shown [33] that LBI can be less biased than LASSO as if nonconvex regularizations [12]. Precisely as $\kappa \to \infty$ and $\alpha_t \to 0$, the limit dynamics of Linearized Bregman Iterations in sparse linear regression may achieve the model selection consistency under nearly the same condition as LASSO yet return the unbiased Oracle estimator, while the LASSO estimator is well-known biased. In our case, LBI (13) is a discretization of the following limit differential inclusion:

$$\frac{d\eta}{dt} = -\nabla_\eta L(\eta, \beta) \tag{14a}$$

$$\frac{d\rho}{dt} = -\nabla_\beta L(\eta, \beta), \tag{14b}$$

$$\rho(t) \in \partial P(\beta(t)). \tag{14c}$$

It evolves as gradient descent flows on a subspace restricted by $\rho(t) \in \partial P(\beta(t))$. For example, for a LASSO penalty $P(\beta) = \|\beta\|_1$, the support set $S_t = \{i : \beta_i(t) \neq 0\}$ must lead to $\rho_{S_t}(t) = \pm 1$ as a constant function, which leads to $\nabla_\beta L(\eta, \beta_{S_t}(t)) = -\frac{d\rho_{S_t}}{dt} = 0$, whence $\beta(t)$ is a minimizer (maximum likelihood estimator) restricted on the support set S_t. Such a minimizer is unbiased when sign consistency is reached, hence is statistically more accurate than any convex regularized estimator such as LASSO. For more details, we refer the readers to see [33] and references therein. Dynamics (14) is often called *Inverse Scale Space* as it evolves with coarse-to-fine models, where at different t one obtains models at different levels.

Here we give some remarks on the implementation details of the Linearized Bregman Iterations (13).

- The parameter κ determines the bias of the sparse estimators, a bigger κ leading to the less biased ones. The parameter α is the step size which determines the precise of the path, with a large α rapidly traversing a coarse-grained path. However one has to keep $\alpha\kappa$ small to avoid possible oscillations of the paths, e.g. $\alpha\kappa\|d\Phi\Phi^T d^T + XX^T\|_2/m < 2$. The default choice in this paper is $\alpha = \frac{m}{\kappa\|d\Phi\Phi^T d^T + XX^T\|_2}$ as a tradeoff between performance and computation cost.
- The step (13a) can also be replaced by
$$\eta^{k+1} = \arg\min_\theta L(\eta, \beta^k)$$
if it is easy to solve.
- Now we turn to simplify the third step (13c) with an explicit formula for the proximal map with the particular penalty function defined in Eq. (12). Recovering β^{k+1} from z^{k+1} is equivalent to the following group shrinkage on each group component of β, i.e. γ^u and ξ^u:
$$\begin{aligned} \beta^{k+1} &= \kappa\textbf{Shrinkage}(z^{k+1}) \\ &\triangleq \begin{cases} \xi^{u,k+1} = \kappa \max(0, 1 - 1/\|z_{\xi^u}\|_2) z_{\xi^u} \\ \gamma^{u,k+1} = \kappa \max(0, 1 - 1/|z_{\gamma^u}|) z_{\gamma^u} \end{cases} \end{aligned} \tag{15}$$

Now we are ready to give the following Linearized Bregman Algorithm for our Mixed-Effects HodgeRank as Alg. 1.

Algorithm 1 LBI for ME-Model

Input: Data (d, X, y), damping factor κ, step size α.
Initialize: $\beta^0 = 0, \eta^0 = 0, z^0 = 0, t^0 = 0$.
for $k = 0, \ldots, K$ **do**
1) $pred^k = d\Phi\eta^k + X\beta^k$
2) $g^{k+1} = \textbf{Gradient}(y, pred^k)$
3) $\eta^{k+1} = \eta^k - \frac{\alpha\kappa}{m}\Phi^T d^T g^{k+1}$
4) $z^{k+1} = z^k - \frac{\alpha}{m} X^T g^{k+1}$
5) $\beta^{k+1} = \kappa\textbf{Shrinkage}(z^{k+1})$
6) $t^{k+1} = (k+1)\alpha$.

end for
Output: Solution path $\{t^k, \eta^k, \beta^k\}_{k=0,1,\ldots,K}$.

The **Gradient** function is different for different models. For linear model

$$\textbf{Gradient}(y, pred) = pred - y.$$

While for GLM, it can be written as follows:

$$\textbf{Gradient}(y, pred) = -\psi(y. * pred)./\Psi(y. * pred)$$

Here $.*$ and $./$ means entry-wise multiplication/division, respectively, and $\psi(t) = \Psi'(t)$ is the probability density function corresponding to $\Psi(t)$. Here $\psi(t) = \frac{e^t}{(1+e^t)^2}$ corresponds to Bradley-Terry

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

model and $\psi(t) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ for Thurstone-Mosteller model.

3.3 Synchronized Parallel LBI

To meet the needs of large-scale data analysis, we would like to introduce a vanilla version of synchronized parallel LBI. The algorithm 1 only needs matrix-vector multiplication, which is easy to be parallelized. Algorithm 2 is the synchronized parallel version of Algorithm 1.

Algorithm 2 SynPar-LBI of Algorithm 1

Initialization: Given parameter κ, $\triangle t$ and thread number P, $k = 0, z^0 = 0, w^0 = 0$.
Split data and variables:
$U = \bigcup_{i=1}^{P} U_i$, $\{1, \ldots, p\} = \bigcup_{i=1}^{P} J_i$.
Iteration: For each thread i
$update_i = 0$. For all u in U_i,

$$pred_u^k = d^u \Phi \eta^k + d^u \Phi (\xi^u)^k + A^u \gamma^u. \tag{16a}$$
$$g_u^{k+1} = \mathbf{Gradient}(y^u, pred_u^k) \tag{16b}$$
$$z_{\xi^u}^{k+1} = z_{\xi^u}^k + \frac{\alpha}{m} \Phi^T (d^u)^T g^{k+1}. \tag{16c}$$
$$z_{\gamma^u}^{k+1} = z_{\gamma^u}^k + \frac{\alpha}{m} (A^u)^T g^{k+1}. \tag{16d}$$
$$(\xi^u)^{k+1} = \kappa \, \mathrm{shrink}(z_{\xi^u}^{k+1}). \tag{16e}$$
$$(\gamma^u)^{k+1} = \kappa \, \mathrm{shrink}(z_{\gamma^u}^{k+1}). \tag{16f}$$
$$update_i = update_i + \Phi^T (d^u)^T g^{k+1}. \tag{16g}$$

Synchronize.

$$\eta_{J_i}^{k+1} = \eta_{J_i}^k + \frac{\alpha\kappa}{m} \sum_{j=1}^{P} update_j.$$

Synchronize.
Stopping: exit when stopping rules are met.

3.4 Early Stopping Regularization

The Alg.1 or 2 actually returns a solution path with many estimators of different sparsity. So we need to find an optimal stopping time among $t^k = \alpha k$ to choose some best estimators and avoid overfitting. Here we sketch the procedure of cross-validation to choose the optimal stopping time:

- Given the training data, fix κ and α, then split the data into K folds. Then choose a list of parameter t.
- **for** $k = 1, \ldots, K$ **do**
  1) Run Alg.1 or 2 on the training data except k-th fold to get the solution path.
  2) For pre-decided parameter list of t, use a linear interpolation to get $(\eta(t), \beta(t))$.
  3) On the k-th fold of training data, use the estimator $(\eta(t), \beta(t))$ to predict, and then compute prediction error.

  end for
- Return the optimal t_{cv} with minimal average prediction error.

Remark: Because the Alg.1 or 2 only return the estimator at discrete $\{t^k\}$ and may not contain the pre-decided parameter t, so we use a linear interpolation of the nearest two estimator (η^k, z^k) and (η^{k+1}, z^{k+1}) to approximate $(\eta(t), z(t))$. $\beta(t)$ is further obtained by using $\mathbf{Shrinkage}(z(t))$.

4 Experiments

In this section, four examples are exhibited with both simulated and real-world data to illustrate the validity of the analysis above and applications of the methodology proposed. The first example is with simulated data while the latter three exploit real-world data collected by crowdsourcing.

4.1 Simulated Study

Settings We validate the proposed algorithm on simulated data with $n = |V| = 20$ labeled by 100 users. Specifically, we first generate the feature matrix for each nodes: $\Phi = [\phi_i^T]_{i=1}^n \in \mathbb{R}^{n \times d}$, where ϕ_i is a d-dimensional ($d = 10$ in this experiment) column feature vector drawn randomly from $\mathcal{N}(0, 1)$ representing node i. Then each entry of the common coefficient β has a probability $p_1 = 0.4$ with nonzero value and they are drawn randomly from $\mathcal{N}(0, 1)$. Besides, for each user u, each entry of his personalized deviation coefficient δ_u has a probability $p_2 = 0.4$ to be nonzero and is drawn randomly from $\mathcal{N}(0, 1)$. Moreover, each user has a probability $p_1 = 0.4$ having a nonzero γ^u, and those nonzero γ^u is drawn randomly from $\mathbb{N}(0, 2^2)$. At last, we draw N^u samples for each user randomly with binary response y_{ij}^u following the model $P(y_{ij}^u = 1) = \Psi((\phi_i^T \eta + \phi_i^T \xi^u) - (\phi_j^T \eta + \phi_j^T \xi^u) + \gamma^u)$, where $\Psi(t) = 1/(1 + e^{-t})$. The sample number N^u uniformly spans in $[N_1, N_2] = [50, 200]$. Finally, we obtain a multi-edge graph labeled by 100 users.

Comparative Results To see whether our proposed method could provide more precise preference function for users by introducing individual-specific parameters, we randomly split the whole data sample into training set and testing set. In particular, we first split the items into training item (75% of the total items) and testing item (the remaining 25%). Then pairwise comparisons which contain one/two of the testing item will be pushed into the testing set, while others will be treated as training set. In other words, via this partition, for each comparisons in the testing set, at least one item is a new comer which has never appear in the training set. To ensure the statistical stability, we repeat this procedure 20 times. We compare our fine-grained model with 7 competitors, i.e., RankSVM [20], RankBoost [14], RankNet [4], gdbt [15], dart [47], Unified Robust Learning to Rank (URLR) [16], and HodgeRank [19]. Tab.1 shows the experimental results of the proposed mixed-effects model compared with other coarse-grained models, which indicates that all of our models exhibit smaller

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 1: Coarse-grained vs. fine-grained model (i.e., Ours) on test error (i.e. mismatch ratio) in simulated data.

	min	mean	max	std
RankSVM [20]	0.1846	0.3165	0.4831	0.0750
RankBoost [14]	0.2216	0.3415	0.4818	0.0690
RankNet [4]	0.1980	0.3213	0.4728	0.0740
gdbt [15]	0.2043	0.3241	0.4809	0.0767
dart [47]	0.2241	0.3235	0.4761	0.0732
URLR [16]	0.2061	0.3198	0.4378	0.0726
HodgeRank [19]	0.1946	0.3097	0.4788	0.0760
Linear	0.1449	0.1892	0.2368	0.0264
Bradley-Terry	0.1282	0.1799	0.2355	0.0303
Thurstone-Mosteller	0.1300	0.1822	0.2368	0.0297

TABLE 2: Coarse-grained vs. fine-grained model (i.e., Ours) on test error (i.e. mismatch ratio) in movie dataset.

	min	mean	max	std
RankSVM [20]	0.3580	0.4538	0.5642	0.0515
RankBoost [14]	0.4249	0.4663	0.5140	0.0267
RankNet [4]	0.419	0.4585	0.5168	0.0221
gdbt [15]	0.3217	0.4215	0.5070	0.0453
dart [47]	0.2988	0.4335	0.5189	0.0487
URLR [16]	0.3998	0.4409	0.4876	0.0230
HodgeRank [19]	0.3918	0.4361	0.4666	0.0196
Linear	0.3103	0.3316	0.3479	0.0105
Bradley-Terry	0.3016	0.3278	0.3440	0.0117
Thurstone-Mosteller	0.3072	0.3291	0.3457	0.0113

M	T(M)(s)	M	T(M)(s)
1	232.09	9	29.38
2	120.67	10	26.80
3	83.40	11	25.27
4	62.70	12	24.05
5	51.15	13	22.55
6	42.04	14	20.84
7	37.27	15	20.23
8	34.60	16	20.24

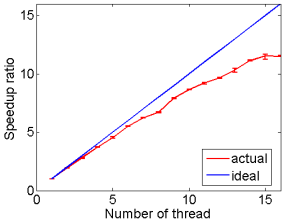


Fig. 3: Left: Mean running time (20 times repeat) of SynPar-LBI with thread number changing from 1 to 16 in simulated data. Right: The linear speedup of parallel LBI in simulated data.

M	T(M)(s)	M	T(M)(s)
1	629.66	9	72.61
2	326.75	10	64.93
3	220.81	11	60.74
4	164.83	12	54.85
5	134.05	13	53.22
6	107.34	14	48.50
7	94.67	15	47.91
8	83.79	16	43.75

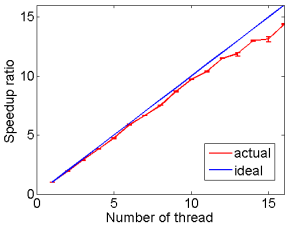


Fig. 4: Left: Mean running time (20 times repeat) of SynPar-LBI with thread number changing from 1 to 16 in movie dataset. Right: The linear speedup of parallel LBI in movie dataset.

test error (i.e. mismatch ratio) due to their parsimonious multi-levels. Besides, it is worth mentioning that GLM-based models (i.e. Bradley-Terry and Thurstone-Mosteller) could exhibit better performance than linear model which suggests that these two are more suitable for binary data.

Speedup of SynPar-LBI We then demonstrate the linear speedup of the synchronized parallel LBI. In evaluating a parallel system, the typical performance measure is *speedup*, which is defined as the ratio of the elapsed time when executing a program on a single thread (the single thread execution time) to the execution time when M threads are available. Let $T(M)$ be the time required to complete the task on M threads. The speedup $S(M)$ is the ratio: S(M)=T(1)/T(M).

In our setting, $M = 1, 2, 3, ..., 16$. Fig.3 (Left) shows the mean running time for 20 times repeat of SynPar-LBI with thread number changing from 1 to 16 in a 16-core server with Intel(R) Xeon(R) E5-2670 2.60GHz CPU and 384GB of RAM. The server runs Linux 4.2.0 64bit. Furthermore, Fig.3 (Right) shows the error bar of speedup with confidence interval [0.25 0.75]. It is easy to find that the parallel LBI could speed up the running time almost in a linear manner.

4.2 Movie Preference Prediction

Dataset The MovieLens 1M DataSet [1] is comprised of 3952 movies rated by 6040 users. Each movie is rated on a scale from 1 to 5, with 5 indicating the best movie and 1 indicating the worst movie. There are a total of one million ratings in this dataset. Moreover, demographic information is provided voluntarily by the users, including gender, age range, occupation. Each movie titles are identical to titles provided by the IMDb [2] and each can be represented as a 18-dimensional genre feature vector, including Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western.

Settings We then select a subset of this dataset containing 100 movies rated by 420 users, ensuring that each user has at least 20 ratings while each movie has been rated by at least 10 users. Since the proposed algorithm is designed for pairwise comparisons, we convert the rating information into a set of pairwise comparisons. More specifically, we create a pairwise comparison (i, j) if item i is rated higher by user u than item j. Note that no pairwise comparison data is generated if two items are given the same rating.

Individual Preference Follow the experiment design in simulated study, we also split the dataset into training set and testing set. All the experiments were repeated 20 times to reduce variance. Similar to the simulated dataset, the proposed fine-grained method could produce better performance than coarse-grained models with smaller mean test error, shown in Tab.2. Moreover, Fig.4 shows the running time of SynPar-LBI on this movie dataset and we can easily find the nearly linear speedup.

Occupation and Age Preference Movie preference behavior, may be influenced by the occupation and age factors. Tab.3 (a) shows the occupation categories in this dataset while Tab.3 (b) illustrates the age range. To exhibit the occupation influence of movie preference behavior, users from the same occupation

1. https://grouplens.org/datasets/movielens/

2. http://www.imdb.com/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 3: Occupations and age ranges in movie dataset.

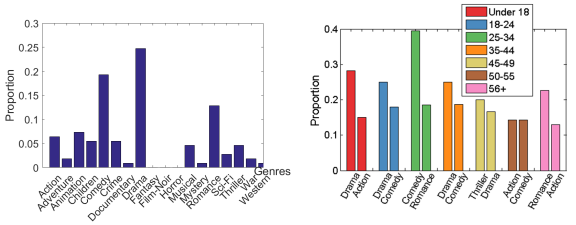
(a) Occupation categories

Index	Occupation	Index	Occupation
1	academic or educator	12	programmer
2	artist	13	retired
3	clerical or admin	14	sales or marketing
4	college or grad student	15	scientist
5	customer service	16	self-employed
6	doctor or health care	17	technician or engineer
7	executive or managerial	18	tradesman or craftsman
8	farmer	19	unemployed
9	homemaker	20	writer
10	K-12 student	21	other or not specified
11	lawyer		

(b) Age ranges

Index	Age Range
1	Under 18
2	18-24
3	25-34
4	35-44
5	45-49
6	50-55
7	56+

are treated as a group. To further investigate the characteristics of groups with personalized preference, we plot the LBI regularization paths of the preference deviations, as has been shown in Fig.1 (b) in introduction. The purple curve indicates the path of the common preference parameter, being the first popping up. The red curves represent the top 3 groups (i.e., farmer, artist, and tradesman) who jumped out early. Groups who jumped out earlier are those with a large deviation from the common ranking. Besides, the blue curves indicate the bottom 3 groups (i.e., homemaker, writer, and self-employed) jumped out later, and those often show similar preference with the common. In particular, the common preference is illustrated in Fig.5(a) where the bars are the proportions of movie genres among top 50% movies ranked by common consensus preference. One can see that the top four genres in the common (social) preference are Drama, Comedy, Romance, and Animation, respectively.



(a) (b)

Fig. 5: (a) The common preference in MovieLens dataset; (b) Preference of 7 groups with different age range in MovieLens dataset.

Despite those trends with occupation, movie preference also undergoes changes with age, and Fig.5(b) illustrates the evolution of preference over age groups. One can see that users under the year of 18 prefer Drama and Action movies best, while ones between 18-24 are willing to watch Drama and Comedy instead. When users slowly waltz into their 25-34, they begin to enjoy the love story. However, when they get to their 40s, it happened that they grew to like the thriller movie best. Not surprisingly, as they continue into old age such as beyond 56, their retrospect on whole life cherishes love in a deep way and Romance movie returns to be their favourite again.

TABLE 4: Coarse-grained vs. fine-grained model (i.e., Ours) on test error (i.e. mismatch ratio) in IQA dataset (reference image 1).

	min	mean	max	std
RankSVM [20]	0.1334	0.1627	0.1808	0.0145
RankBoost [14]	0.1552	0.1727	0.1898	0.0107
RankNet [4]	0.1945	0.2260	0.2525	0.0136
gdbt [15]	0.1337	0.1491	0.1654	0.0102
dart [47]	0.1337	0.1532	0.1667	0.0113
URLR [16]	0.1894	0.2140	0.2553	0.0177
HodgeRank [19]	0.1678	0.1874	0.2019	0.0081
Linear	0.1002	0.1094	0.1239	0.0065
Bradley-Terry	0.0638	0.0708	0.0817	0.0054
Thurstone-Mosteller	0.0625	0.0731	0.0825	0.0057

M	T(M)(s)	M	T(M)(s)
1	57.38	9	7.54
2	29.98	10	7.00
3	20.99	11	6.40
4	16.11	12	6.11
5	13.06	13	5.97
6	11.16	14	5.76
7	9.69	15	5.53
8	8.82	16	5.29

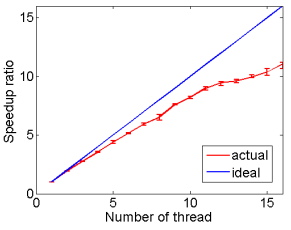
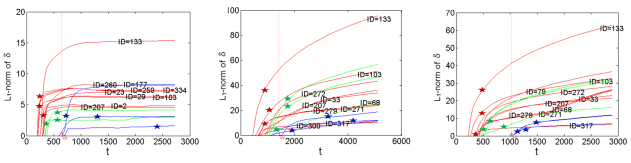


Fig. 6: Left: Mean running time (20 times repeat) of SynPar-LBI with thread number changing from 1 to 16 in IQA dataset. Right: The linear speedup of parallel LBI in IQA dataset.

4.3 Image Quality Assessment (IQA)

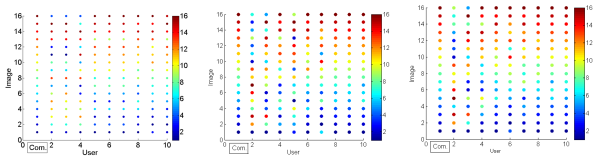
Settings Two publicly available datasets, LIVE [42] and IVC [22], are used in this work. It includes $52,043$ paired comparisons collected from 342 observers of different cultural background. The number of responses each reference image receives is different. To validate whether the annotators' preference function we estimated is good enough, we randomly take reference image 1 as an illustrative example while other reference images exhibit similar results.

Results Tab.4 shows the mean test error (70% da-



(a) Linear (b) BT (c) TM

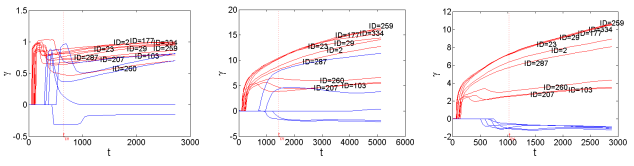
Fig. 7: LBI regularization path of δ exhibiting personalized ranking in IQA dataset (reference image 1). (Red: top 10 personalized ranking annotators; Green: middle 3; Blue: bottom 3)



(a) Linear (b) BT (c) TM

Fig. 8: Ranking order comparison of common vs. personalized rankings of 9 representative annotators in IQA dataset (reference image 1).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Linear (b) BT (c) TM

Fig. 9: LBI regularization path of γ exhibiting position bias in IQA dataset (reference image 1). (Red: top 10 position-biased annotators; Blue: bottom 5 position-biased annotators).

ta for training, 30% for testing) results of 20 times achieved by this scheme. It is shown that consistent with the simulated data, in this dataset, the mixed-effects model with three losses could also provide better approximate results of the annotators' preference than the HodgeRank estimator. Moreover, Fig.6 shows the running time of SynPar-LBI on this IQA dataset and we can easily find the nearly linear speedup.

To further investigate the characteristics of annotators with personalized ranking, Fig.7 illustrates annotator's LBI regularization paths of preference deviations with optimal t (i.e., t_{cv}) returned by cross-validation in three losses. The red curves in Fig.7 represent the top 10 annotators who jumped out early. Moreover, Fig.8 shows the order comparisons of common ranking (i.e., com.) and personalized ranking of 9 representative annotators at t_{cv}. The X-axis represents user index: user = 2, 3, 4 jumped out early corresponding to paths labeled with red stars in Fig.7; user = 5, 6, 7 jumped out in the middle time corresponding to green stars; user = 8, 9, 10 jumped out late corresponding to blue stars. The order of faces in Y-axis is arranged from lower to higher (i.e., from color blue to red) according to the common ranking score calculated by our method. The color represents the ranking position returned by the corresponding user. It is easy to see users jumped out late exhibit almost consistent ranking order with the common ranking, while the earlier ones are almost the adversarial against the common.

Remark It is easy to see that among the top 10 annotators returned by linear model, 9 of them (except annotator with ID = 133) click one side almost all the time (i.e., position-biased annotators), while results returned by other two are not. The reason of such a phenomenon lies in the difference of linear model and probability model. Such kind of user always has $y_{ij}^u \equiv 1$ or -1. In simple linear model, to fit such user, only a position bias term γ^u is not enough. Since the common score η always exists and is nonzero, only $\gamma^u = 1$ or -1 and $\xi^u = -\eta$ can fit the data well, so under the linear model, these users' ξ is nonzero. While in the other two GLM, the probability explain makes a single γ^u enough to fit the data. Since a γ^u with much larger magnitude the $\eta_i - \eta_j$ can already dominant the probability $\Psi((\eta_i + \xi_i^u) - (\eta_j + \xi_j^u) + \gamma^u)$

TABLE 5: Top 10 position-biased annotators in IQA dataset (reference image 1).

Order	ID	Left	Right	Order	ID	Left	Right
1	259	96	0	6	2	55	0
2	334	90	0	7	260	49	2
3	177	77	0	8	23	42	0
4	103	74	4	9	207	46	2
5	29	58	0	10	287	34	0

even $\xi^u = 0$. Also in the other data, there also exist such kind of users, but their samples are not as many as those in this data, so such a phenomenon is not observed in the other data. This gives an example that GLM can be qualitatively better than the linear model for binary comparison data.

Moreover, Fig.9 illustrates the LBI regularization paths of annotator's position bias with red lines represent the top 10 annotators. It is easy to see that the corresponding results returned from these three loss functions are exactly the same. Tab.5 further shows the click counts of each side (i.e., Left and Right) for these top 10 position-biased annotators. It is easy to see that these annotators can be divided into two types: (1) click one side all the time (with ID in blue); (2) click one side with high probability (others). Although it might be relatively easy to identify the annotators of type (1) above by inspecting their inputs, it is impossible for eye inspection to pick up those annotators of type (2) with mixed rational and abnormal behaviors. Therefore it is essential to design such a statistical methodology to quantitatively detect these kind of position-biased annotators for crowdsourcing platforms in market. It is interesting to see that annotators highlighted with blue color in Tab.5 click the left side all the time. We then go back to the crowdsourcing platform and find out that the reason behind this is a *default choice* on the left button, which induces some lazy annotators to cheat for the task.

TABLE 6: Coarse-grained vs. fine-grained model (i.e., Ours) on test error (i.e. mismatch ratio) in WorldCollege ranking dataset.

	min	mean	max	std
RankSVM [20]	0.3009	0.3165	0.3289	0.0068
RankBoost [14]	0.3199	0.3321	0.3448	0.0073
RankNet [4]	0.3393	0.3533	0.3710	0.3533
gdbt [15]	0.3436	0.3579	0.3730	0.0086
dart [47]	0.3470	0.3716	0.3986	0.0134
URLR [16]	0.2893	0.3136	0.3269	0.0075
HodgeRank [19]	0.2979	0.3108	0.3230	0.0097
Linear	0.2530	0.2670	0.2757	0.0071
Bradley-Terry	0.2456	0.2555	0.2678	0.0099
Thurstone-Mosteller	0.2553	0.2616	0.2696	0.0067

4.4 WorldCollege Ranking

Settings We now apply the proposed method to the WorldCollege dataset, which is composed of 261 colleges. Using the Allourideas crowdsourcing platform, a total of 340 distinct annotators from various countries (e.g., USA, Canada, Spain, France, Japan) are shown randomly with pairs of these colleges, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2817205, IEEE Transactions on Pattern Analysis and Machine Intelligence

M	T(M)(s)	M	T(M)(s)
1	221.07	9	25.65
2	114.99	10	22.90
3	78.40	11	21.42
4	58.09	12	19.69
5	47.80	13	18.92
6	38.21	14	17.55
7	33.54	15	16.83
8	29.60	16	15.51

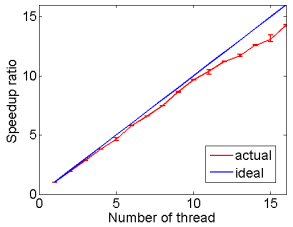
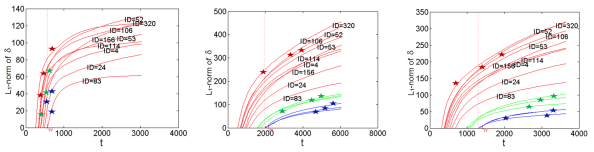


Fig. 10: Left: Mean running time (20 times repeat) of SynPar-LBI with thread number changing from 1 to 16 in WorldCollege ranking dataset. Right: The linear speedup of parallel LBI in WorldCollege ranking dataset.

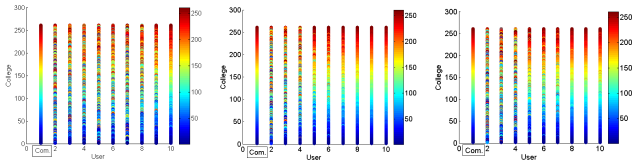


(a) Linear (b) BT (c) TM

Fig. 11: LBI regularization path of δ exhibiting personalized ranking in WorldCollege ranking dataset. (Red: top 9 personalized ranking annotators; Green: middle 3; Blue: bottom 3)

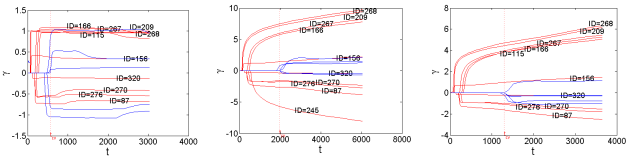
asked to decide which of the two universities is more attractive to attend. Finally, we obtain a total of 8,823 pairwise comparisons.

Results We apply the proposed method to the resulting dataset and find out that, similar to the simulation and other two real-world datasets, the mixed-effects model could produce better performance than Hodgerank with smaller mean test error, shown in Tab.6. Moreover, Fig.10 shows the linear speedup of SynPar-LBI on this dataset. Besides, noting in this dataset, only 9 annotators are treated as annotators with distinct personalized rankings at optimal t (i.e., t_{cv}) selected via cross-validation in linear model case,



(a) Linear (b) BT (c) TM

Fig. 12: Ranking order comparison of common vs. personalized rankings of 9 annotators in WorldCollege ranking dataset.



(a) Linear (b) BT (c) TM

Fig. 13: LBI regularization path of γ exhibiting position bias in WorldCollege ranking dataset. (Red: top 10 position-biased annotators; Blue: bottom 5 position-biased annotators).

TABLE 7: Top 10 position-biased annotators in WorldCollege ranking dataset.

Order	ID	Left	Right	Order	ID	Left	Right
1	268	148	0	6	270	20	70
2	209	127	0	7	267	45	0
3	156	189	67	8	276	16	54
4	320	253	324	9	166	35	0
5	87	11	62	10	115	34	0
				10	245	0	34

as is shown in Fig.11(a). However, other two losses with smaller mean test error detect more than 9 personalized-ranking annotators via cross-validation. To better illustrate comparison result of three losses, for the other two, we also only show the top 9 annotators, as is shown in Fig.11(b) and 11(c). It is pleasing to see that the top 9 annotators returned by three losses are exactly the same in this dataset. The common ranking vs. personalized ranking of 9 representative users is shown in Fig.12 with a similar observation to the other two datasets. Besides, the regularization paths of position bias and click counts of top 10 annotators in this dataset are shown in Fig.13 and Tab.7. It is easy to see that similar to the human age dataset, these annotators are either clicking one side all the time, or clicking one side with high probability in mixed behaviors. Clearly, when showing top 10 position-biased annotators, there is only one difference among these three cases, where linear model and Thurstone-Mosteller both pick out annotator with ID=115, while Bradley-Terry treats annotator with ID=245 as position-biased one. A further inspection of the dataset confirms that such a detection result is reasonable, as the ratio of left/right clicks of these two annotators are 34:0 and 0:34 respectively, as is shown in Tab.7.

5 Conclusions

In this paper, we propose a parsimonious mixed-effects model based on HodgeRank to learn user's preference or utility function in crowdsourced ranking, which takes into account both the personalized preference deviations from the common and position biases of the annotators. To be specific, common preference scores indicate the consistent ranking on population-level which approximates the behavior of all users, while a small set of annotators might have nonzero personalized deviations and abnormal behavior in position bias. Equipped with the newly developed Linearized Bregman Iteration, which is a simple iterative procedure generating a sequence of parsimonious models, we establish a dynamic path from the common utility to individual variations, with different levels of parsimony or sparsity on personalization. In this dynamic scheme, three kinds of models are systematically discussed, including the linear model with L2 loss, the Bradley-Terry model, and the Thurstone-Mosteller model. Experimental

studies conducted on simulated examples and real-world datasets show that our proposed method could exhibit better performance (i.e. smaller test error) compared with the traditional HodgeRank. In addition, generalized linear models may be more efficient to fit binary comparison data in terms of both the reduction of pairwise mismatch (Kendall τ-distance) from observations and the discrimination of position bias from personalized preference deviations. Our results suggest that the proposed methodology is an effective tool to investigate the diversity in annotator's behavior in modern crowdsourced preference data.

REFERENCES

[1] K. Arrow. *Social Choice and Individual Values, 2nd Ed.* Yale University Press, New Haven, CT, 1963.

[2] P. L. Bartlett, M. I. Jordan, and J. D. Mcauliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *International Conference on World Wide Web*, pages 107–117, 1998.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning*, pages 89–96, 2005.

[5] O. Candogan, I. Menache, A. Ozdaglar, and P. A. Parrilo. Flows and decompositions of games: Harmonic and potential games. *Mathematics of Operations Research*, 36(3):474–503, 2011.

[6] A. Chorin and J. Marsden. *A Mathematical Introduction to Fluid Mechanics*. Texts in Applied Mathematics. Springer, 1993.

[7] D. Cynthia, K. Ravi, N. Moni, and S. Dandapani. Rank aggregation methods for the web. In *International Conference on World Wide Web*, pages 613–622, 2001.

[8] H. David. *The Methods of Paired Comparisons*. Oxford University Press, 1988.

[9] A. P. Dawid and A. M. Skene. 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, 1979.

[10] R. L. Day. Position bias in paired product tests. *Journal of Marketing Research*, 6(1):98–100, 1969.

[11] J. de Borda. *Mémoire sur les Elections au Scrutin*. Histoire de l'Académie Royale des Sciences, 1781.

[12] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of American Statistical Association*, pages 1348–1360, 2001.

[13] V. Farias, S. Jagabathula, and D. Shah. A data-driven approach to modeling choice. In *Advances in Neural Information Processing Systems*, pages 504–512, 2009.

[14] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4(Nov):933–969, 2003.

[15] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[16] Y. Fu, T. M. Hospedales, T. Xiang, J. Xiong, S. Gong, Y. Wang, and Y. Yao. Robust subjective visual property prediction from crowdsourced pairwise labels. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):563–577, 2016.

[17] S. Guo, A. G. Parameswaran, and H. Garcia-Molina. So who won?: dynamic max discovery with the crowd. pages 385–396, 2012.

[18] H. Hu, Y. Zheng, Z. Bao, G. Li, J. Feng, and R. Cheng. Crowdsourced POI labelling: Location-aware result inference and task assignment. In *IEEE International Conference on Data Engineering*, pages 61–72, 2016.

[19] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming*, 127(6):203–244, 2011.

[20] T. Joachims. SVM-rank: Support vector machine for ranking, 2009.

[21] E. Kamar, A. Kapoor, and E. Horvitz. Identifying and accounting for task-dependent bias in crowdsourcing. In *AAAI Conference on Human Computation and Crowdsourcing*, pages 92–101, 2015.

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

[23] G. Li, C. Chai, J. Fan, X. Weng, J. Li, Y. Zheng, Y. Li, X. Yu, X. Zhang, and H. Yuan. CDB: optimizing queries with crowd-based selections and joins. In *ACM International Conference on Management of Data*, pages 1463–1478, 2017.

[24] G. Li, J. Wang, Y. Zheng, and M. J. Franklin. Crowdsourced data management: A survey. *IEEE Trans. Knowl. Data Eng.*, 28(9):2296–2319, 2016.

[25] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng. Crowdsourced data management: Overview and challenges. In *ACM International Conference on Management of Data*, pages 1711–1716, 2017.

[26] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. CDAS: A crowdsourcing data analytics system. *PVLDB*, 5(10):1040–1051, 2012.

[27] Y. Lu and S. N. Negahban. Individualized rank aggregation using nuclear norm regularization. In *Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1473–1479. IEEE, 2015.

[28] W. Ma, J. M. Morel, S. Osher, and A. Chien. An L_1-based variational model for retinex theory and its application to medical images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 153–160, 2011.

[29] S. Negahban, S. Oh, and D. Shah. Iterative ranking from pairwise comparisons. In *Advances in Neural Information Processing Systems*, pages 2483–2491, 2012.

[30] S. Negahban, S. Oh, and D. Shah. Iterative ranking from pairwise comparisons. In *Annual Conference on Neural Information Processing Systems*, pages 2483–2491, 2012.

[31] S. Oh and D. Shah. Learning mixed multinomial logit model from ordinal data. In *Advances in Neural Information Processing Systems*, pages 595–603, 2014.

[32] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *SIAM Journal on Multiscale Modeling and Simulation*, 4(2):460–489, 2005.

[33] S. Osher, F. Ruan, J. Xiong, Y. Yao, and W. Yin. Sparse recovery via differential inclusions. *Applied and Computational Harmonic Analysis*, 41(2):436–469, 2016.

[34] B. Osting, C. Brune, and S. Osher. Enhanced statistical rankings via targeted data collection. In *International Conference on Machine Learning*, pages 489–497, 2013.

[35] B. Osting, J. Darbon, and S. Osher. Statistical ranking using the l_1-norm on graphs. *Inverse Problems and Imaging*, 7(3):907–926, 2013.

[36] B. Osting, J. Darbon, and S. Osher. Statistical ranking using the l_1-norm on graphs. *Inverse Problems & Imaging*, 7(3), 2013.

[37] B. Osting, J. Xiong, Q. Xu, and Y. Yao. Analysis of crowdsourced sampling strategies for hodgerank with sparse random graphs. *Applied and Computational Harmonic Analysis*, 41(2):540–560, 2016.

[38] A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International Conference on Machine Learning*, pages 118–126, 2014.

[39] V. C. Raykar, S. Yu, L. H. Zhao, A. K. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 889–896, 2009.

[40] J. D. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *International conference on Machine learning*, pages 713–719, 2005.

[41] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *International conference on Machine learning*, pages 880–887, 2008.

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

[43] A. Sheshadri and M. Lease. SQUARE: A benchmark for research on computing crowd consensus. In *AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[44] C. Suh, V. Y. F. Tan, and R. Zhao. Adversarial top-k ranking. *IEEE Transactions on Information Theory*, 63(4):2201–2225, 2017.
[45] N. M. Tran. Hodgerank is the limit of perron rank. *Mathematics of Operations Research*, 41(2):643–647, 2016. arXiv:1201.4632.
[46] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *International World Wide Web Conference*, pages 155–164, 2014.
[47] R. K. Vinayak and R. Gilad-Bachrach. DART: dropouts meet multiple additive regression trees. In *International Conference on Artificial Intelligence and Statistics*, 2015.
[48] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, pages 2035–2043, 2009.
[49] 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.
[50] 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.
[51] Q. Xu, J. Xiong, X. Cao, and Y. Yao. False discovery rate control and statistical quality assessment of annotators in crowdsourced ranking. In *International Conference on Machine Learning*, pages 1282–1291, 2016.
[52] Q. Xu, J. Xiong, X. Cao, and Y. Yao. Parsimonious mixed-effects HodgeRank for crowdsourced preference aggregation. page preprint. ACM Multimedia, 2016.
[53] J. Yi, R. Jin, S. Jain, and A. Jain. Inferring users' preferences from crowdsourced pairwise comparisons: A matrix completion approach. In *AAAI Conference on Human Computation and Crowdsourcing*, 2013.
[54] S. Yu. Angular embedding: A robust quadratic criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):158–173, 2012.
[55] J. Yuan, G. Steidl, and C. Schnorr. Convex Hodge decomposition and regularization of image flows. *Journal of Mathematical Imaging and Vision*, 33(2):169–177, 2009.
[56] Y. Zheng, R. Cheng, S. Maniu, and L. Mo. On optimality of jury selection in crowdsourcing. In *International Conference on Extending Database Technology*, pages 193–204, 2015.
[57] Y. Zheng, G. Li, and R. Cheng. DOCS: domain-aware crowdsourcing system. *PVLDB*, 10(4):361–372, 2016.
[58] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. Truth inference in crowdsourcing: Is the problem solved? *PVLDB*, 10(5):541–552, 2017.
[59] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng. QASCA: A quality-aware task assignment system for crowdsourcing applications. In *ACMInternational Conference on Management of Data*, pages 1031–1046, 2015.

Qianqian Xu received the B.S. degree in computer science from China University of Mining and Technology in 2007 and the Ph.D. degree in computer science from University of Chinese Academy of Sciences in 2013. She is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. Her research interests include statistical machine learning, with applications in multimedia and computer vision. She has authored or coauthored 10+ academic papers in prestigious international journals and conferences, among which she has published 5 full papers with the first author's identity in ACM Multimedia. She served as member of professional committee of CAAI, and member of online program committee of VALSE, etc.

Jiechao Xiong received the B.S. degree and the Ph.D. degree in statistics from the School of Mathematical Science, Peking University, Beijing, China, in 2011 and 2016, respectively. He is currently with Tencent AI Laboratory, Shenzhen, China. His research interests include statistical machine learning, algorithms, and artificial intelligence with industrial applications.

Xiaochun Cao, Professor of the Institute of Information Engineering, Chinese Academy of Sciences. He received the B.E. and M.E. degrees both in computer science from Beihang University (BUAA), China, and the Ph.D. degree in computer science from the University of Central Florida, USA, with his dissertation nominated for the university level Outstanding Dissertation Award. After graduation, he spent about three years at ObjectVideo Inc. as a Research Scientist. From 2008 to 2012, he was a professor at Tianjin University. He has authored and coauthored over 100 journal and conference papers. In 2004 and 2010, he was the recipients of the Piero Zamperoni best student paper award at the International Conference on Pattern Recognition. He is a fellow of IET and a Senior Member of IEEE. He is an associate editor of IEEE Transactions on Image Processing.

Qingming Huang (SM'08) received the B.S. degree in computer science and Ph.D. degree in computer engineering from Harbin Institute of Technology, Harbin, China, in 1988 and 1994, respectively. He is currently a Professor with the Graduate University of the Chinese Academy of Sciences (CAS), Beijing, China, and an Adjunct Research Professor with the Institute of Computing Technology, CAS. He has authored or coauthored nearly 200 academic papers in prestigious international journals and conferences. His research areas include multimedia video analysis, video adaptation, image processing, computer vision, and pattern recognition. Dr. Huang is a reviewer for IEEE Trans. on Multimedia, IEEE Trans. on Circuits and Systems for Video Technology, and IEEE Trans. on Communications. He has served as program chair, track chair and TPC member for various conferences, including ACM Multimedia, CVPR, ICCV, ICME, PSIVT, etc.

Yuan Yao received the B.S.E and M.S.E in control engineering both from Harbin Institute of Technology, China, in 1996 and 1998, respectively, M.Phil in mathematics from City University of Hong Kong in 2002, and Ph.D. in mathematics from the University of California, Berkeley, in 2006. Since then he has been with Stanford University and in 2009, he joined the Department of Probability and Statistics in School of Mathematical Sciences, Peking University, Beijing, China. He is currently an Associate Professor of Mathematics, Chemical & Biological Engineering, and by courtesy, Computer Science & Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China. His current research interests include topological and geometric methods for high dimensional data analysis and statistical machine learning, with applications in computational biology, computer vision, and information retrieval. Dr. Yao is a member of American Mathematical Society (AMS), Association for Computing Machinery (ACM), Institute of Mathematical Statistics (IMS), and Society for Industrial and Applied Mathematics (SIAM). He served as area or session chair in NIPS and ICIAM, as well as a reviewer of Foundation of Computational Mathematics, IEEE Trans. Information Theory, J. Machine Learning Research, and Neural Computation, etc.