HITSnDIFFs: From Truth Discovery to Ability Discovery by Recovering Matrices with the Consecutive Ones Property

Zixuan Chen
Northeastern University
chen.zixu@northeastern.edu
0000-0003-2872-1865

Subhodeep Mitra
Google
mitradeep1@gmail.com

R. Ravi
Carnegie Mellon University
ravi@andrew.cmu.edu
0000-0001-7603-1207

Wolfgang Gatterbauer
Northeastern University
w.gatterbauer@northeastern.edu
0000-0002-9614-0504

***Abstract*—We analyze a general problem in a crowd-sourced setting where one user asks a question (also called item) and other users return answers (also called labels) for this question. Different from existing crowd sourcing work which focuses on *finding the most appropriate label for the question (the "truth")*, our problem is to *determine a ranking of the users based on their ability to answer questions*. We call this problem "*ability discovery*" to emphasize the connection to and duality with the more well-studied problem of "truth discovery".**

To model items and their labels in a principled way, we draw upon Item Response Theory (IRT) which is the widely accepted theory behind standardized tests such as SAT and GRE. We start from an idealized setting where the relative performance of users is *consistent* across items and better users choose better fitting labels for *each* item. We posit that a principled algorithmic solution to our more general problem should solve this ideal setting correctly and observe that the response matrices in this setting obey the *Consecutive Ones Property* (C1P). While C1P is well understood algorithmically with various discrete algorithms, we devise a novel variant of the HITS algorithm which we call "HITSnDIFFs" (or HnD), and prove that it can recover the ideal C1P-permutation in case it exists. Unlike fast combinatorial algorithms for finding the consecutive ones permutation (if it exists), HnD also returns an ordering when such a permutation does not exist. Thus it provides a *principled heuristic for our problem that is guaranteed to return the correct answer in the ideal setting*. Our experiments show that HnD produces user rankings with robustly high accuracy compared to state-of-the-art truth discovery methods. We also show that our novel variant of HITS scales better in the number of users than ABH, the only prior spectral C1P reconstruction algorithm.

***Index Terms*—truth discovery, item response theory, consecutive ones property**

I. Introduction

Motivation. We first present a couple of examples from a class to a more general crowdsourcing context.

Example 1 (Student ranking). *Kiyana, an innovative instructor for an online class who suffered from the leakage of previous exam questions and difficulty of creating new ones, notices a lot of interactions in student forums like Piazza [2] and pilots a new learning approach. Since students are willing to ask and answer questions, Kiyana aims to utilize such communication for both practice and assessment of the class by requiring students to suggest and answer multiple-choice questions (MCQs) themselves. The task of students is to come up with meaningful MCQs (including question stems and choices) related to the topics of the class and answer the questions from others. In this way, the initiative of and interactions between students are encouraged, and their performances can be used as another important measure (e.g., a "participation" score) towards the grade assessment, in addition to traditional exam scores. To assess students, Kiyana first simply counts how many times a student answers questions, which is the traditional way for an instructor to give a participation score for students in a forum. However, in this way, the grades are biased towards students who answer a lot of questions randomly. The second attempt is to require all students to answer the same number of questions and rank them by how many questions they answer correctly, which still requires a lot of efforts for her to figure out all the correct answers and suffers from the problem that each question has quite different difficulties. Kiyana wonders whether there is a more principled way for ranking students by their abilities to answer questions correctly.*

Example 2 (Crowd workers ranking). *Daiyu wants to release a human intelligence task which consists of a set of questions at a crowdsourcing platform like Amazon Mechanical Turk [1]. Suffering from low-quality answers from the crowd workers, she wonders whether there is a better way to select top crowd workers instead of simply setting thresholds for the number of tasks they have finished or finished successfully.*

The above examples motivate our problem of "ability discovery" which ranks the users (students/workers) based on their abilities to answer questions correctly.

The ability discovery problem. We have m users and n items.[1] Each item has up to k labels,[2] and the labels usually vary between items (the items are thus "heterogeneous"). Each user chooses up to one label to each of the items, and two

[1] We use item/question and label/option/answer/choice interchangeably.

[2] In other words, the item(s) with the most unique labels has k different labels. Labels can be proposed either from questioners or answerers.

users may choose the same label to the same item. Our goal is to derive a principled way for *determining a ranking of the users in terms of their ability to pick correct labels for the items* based solely on the user responses.

Connection to truth discovery. Ability discovery can be considered a dual problem of the widely studied *truth discovery* problem [69]. The setup is similar; the difference is that the truth discovery problem measures success in finding the truth (thus the correct label for each item) whereas our problem focuses on finding the correct ranking of the users by their relative abilities. While the truth discovery problem occurs in a wide range of problems related to crowdsourcing and has been of intense focus for the data management community [69], the ability discovery problem gets little attention and is usually treated as a sub-problem: if one knows the truth, it is easier to rank the users based on the choices they make. In turn, if one knows the order of user abilities, it is easier to determine the truth. However, we show in Section IV that it is not straightforward to rank users correctly, even when given the correct answers to the questions beforehand, which means even the perfect truth discovery method is not guaranteed to perform well for the ability discovery problem. Furthermore, we argue that ability discovery is much more than a sub-problem of truth discovery and highlight its importance in two aspects: 1) It has different application scenarios as we discussed in the examples. 2) Different from correctness of answers, user abilities are *abstract and cannot to be measured directly*, which makes ability discovery results valuable but also hard to evaluate with *no acknowledged ground truth*.

Assumptions. Similar as in truth discovery, we assume an objective total order on the labels of each item based on their correctness, and a total order on the users based on their latent one-dimensional abilities for choosing the correct labels.

Our approach. We first define an idealized scenario in which the user responses are *consistent* with their abilities across the items and characterize it as the response matrix having the *Consecutive Ones Property* (C1P). We then suggest an efficient spectral method that we call HITSNDIFFS (HND) for reconstructing such ideal orderings if they exist. We prove that in the ideal error-free scenario (better users always make better choices) our method is guaranteed to find the correct ranking, which puts our approach on a stronger theoretical footing as a cross between a heuristic and an exact algorithm. Importantly, our method generalizes to the general non-ideal case and allows us to compare it with existing truth discovery methods for ranking users. One key innovation in our work is the use of Item Response Theory (IRT) [57] to model both label rankings as well as the propensity of users of different abilities to choose such labels, and the previously not made connection to C1P. We utilize 3 different generative models from the IRT literature to generate realistic synthetic data. Experiments on these data show that (i) our new method is *more accurate than existing truth discovery methods*, and (ii) it can also serve as scalable approach that *reconstructs the C1P order if it exists* and *generalizes much better on non-ideal inputs* than the only other C1P order reconstruction method that works in the non-ideal scenario.

Contributions. (1) We connect the notion of *consistent responses* in heterogeneous multiclass classification to the well-known *Consecutive Ones Property* (C1P) from seriation theory [4], [21], [27]. We argue that any principled solution to ranking users by their abilities should be able to recover the correct ranking when responses are consistent with abilities.

(2) We propose a novel yet simple adaptation of the HITS algorithm [29] that we call HITSNDIFFS (HND) for ranking users based on their abilities. We prove the surprising result that HND *recovers the consecutive ones ranking* of users when a unique such order exists. Unlike fast combinatorial algorithms for finding the C1P ordering if it exists, HND *can also deal with the general case when no such order exists*. This makes HND an ideal candidate for our problem (and even becomes an exact algorithm in special settings). We compare HND against ABH [4], which is the only other spectral approach that has these properties, and give intuitive and experimental evidence for why HND performs better.

(3) We show how *Item Response Theory (IRT)* [57], which is widely deployed in educational testing, provides a natural and mathematically principled theory (including generative models) for modeling heterogeneous item ranking that includes the C1P as a special case of consistent responses.

(4) We conduct extensive experiments on synthetic datasets generated by 3 polytomous IRT models and show that HND can outperform other existing truth discovery approaches in terms of accuracy of the user ranking. We also show (both in theory and with experiments) that HND has better scalability and accuracy than ABH (which is the only other C1P reconstruction approach known today that can be used for the general ranking problem).

Outline. Section II defines our problem, draws the connection to the C1P property, and introduces IRT as generalization of C1P. Section III introduces our approach, gives its formal properties and compares it to closely related work. Section IV presents experiments. Section V discusses additional related work on truth discovery before Section VI concludes. Code, proofs and more experiments are available online [8], [9].

II. Formal setup

A. *Ability discovery problem formulation*

Consider a setting with m users choosing among k options for each of n items. Items are *heterogeneous* in that they have *different options*, as is the case in MCQs used in standardized test settings (see Figure 1a). This setup is different from typical multiclass classification [12] where all n items have the same class of k labels. To emphasize the difference, we refer to our setup as *heterogeneous multiclass classification*.

User responses can be represented in one-hot encoding as a $(m \times kn)$ *binary response matrix* $\mathbf{C}$ (see Figure 1b) where each row represents the choices of a user and each column represents an option for some item. The number of non-zeros in $\mathbf{C}$ is mn and the number of non-zeros in each row n.

We assume that each user j has a latent one-dimensional *ability* θ_j that represents the user's ability to choose high-quality options for each item. Our goal is to rank the users by their abilities to choose high-quality options.

Definition 1 (Ability discovery)**.** *Given m users and their choices among k options for n* heterogeneous *items as binary response matrix* $\mathbf{C}$.[3] *Rank the users by their abilities to choose higher quality options for each item.*

Several approaches on homogeneous items assume the probabilities of users getting a correct answer to be identical across questions and encode user abilities as a $(k \times k)$ dimensional confusion matrix per user [12], [69]. In our setting, this is not the case: each option h for item i may have a different "*quality*" η_{ih}. The higher the quality is, the more likely it is picked by better students. The probability of answering a question correctly (i.e. choosing the highest-quality option) then depends on the interplay between the user ability and the option qualities. This setup perfectly fits *Item Response Theory (IRT)* [36], [57], summarized in detail in Section II-D.

Example 3. *Figure 1a shows $m=4$ users answering $n=3$ MCQs. Each question has $k = 3$ choices labeled A to C in decreasing order of quality. Figure 1b shows a response matrix $\mathbf{C}'$ and its binary form $\mathbf{C}$. Assuming that users' choices are "consistent" with their abilities (i.e. correctness of labeling increases with ability), the only possible ranking of users for the observed $\mathbf{C}'$ is 1, 2, 3, 4, or its reverse. Figure 1c illustrates an IRT model for the probability of picking the correct answer A for each item as function of user abilities when the correct ranking is 1, 2, 3, 4. For example, user 2 has the ability to label items 1 and 2 correctly and thus chooses the correct answer A for both items. Our problem is to rank the users by their mastery of the subject based solely on the users' choices without knowing the correct labels.*

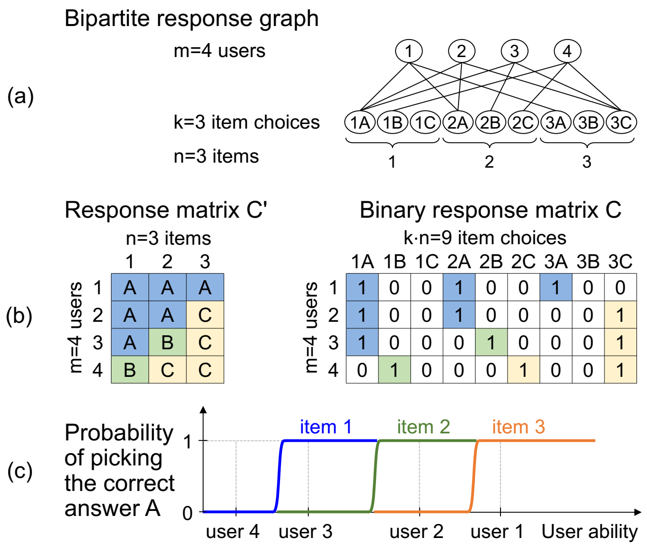
B. The ideal case with consistent responses

We call a response matrix "*consistent*" if there is a (total) order of the users (based on their abilities) that is consistently reflected in their responses across all items. In this ideal case, *if a user j_1 chooses a better option (i.e. one of higher quality) than user j_2 for an item, then user j_1 must also choose an equal or better option than user j_2 for any other item.* This implies that there are also, for each item, a total order among the options from best to worst, and the better users choose better or equal options for every item.

In other words, assume that the user abilities θ_j are all distinct, and also that for every item i, the qualities η_{ih} of its options are all distinct, then there is *a unique total order of the users, and of the options for each item.*

Definition 2 (Consistent Responses)**.** *A response matrix* $\mathbf{C}$ *is* consistent *if there exists an assignment of user abilities $\boldsymbol{\theta}$ and option qualities $\boldsymbol{\eta}$, s.t. for any pair of users j_1 and j_2 with $\theta_{j_1} > \theta_{j_2}$, and for any item i where user j_1 chooses option h_1 and user j_2 chooses option h_2, we have $\eta_{ih_1} \geq \eta_{ih_2}$.*

[3]To simplify the discussion and different from Section I, we assume here that each item has exactly k choices. For items with $k' < k$ choices, we can assume them to have $k - k'$ more choices and nobody choosing them.



	1	2	3
1	A	A	A
2	A	A	C
3	A	B	C
4	B	C	C

	1A	1B	1C	2A	2B	2C	3A	3B	3C
1	1	0	0	1	0	0	1	0	0
2	1	0	0	1	0	0	0	0	1
3	1	0	0	0	1	0	0	0	1
4	0	1	0	0	0	1	0	0	1

Fig. 1: (a) Ability discovery problem: $m=4$ users choose one from $k=3$ choices of labels A, B, C for each of $n = 3$ items. (b) Input: the $(m \times k)$ response matrix $\mathbf{C}'$, or equally its flattened $(m \times kn)$ binary response matrix $\mathbf{C}$. (c) Model: the probability of picking the correct answer in terms of the user ability for Items 1,2,3. The abilities of all 4 users are marked on the horizontal axis.

C. Relation to Consecutive ones Property (C1P)

We observe that consistent response matrices, when row-sorted according to user abilities, satisfy a widely studied ordering property in seriation, called the *consecutive ones property* (C1P) [4], [21], [27]. We follow the notation from seriation theory and call it a *P-matrix.*

Definition 3 (C1P, P-matrix & pre-P-matrix [4])**.** *A binary matrix satisfies* C1P *and is called a* P-matrix *if in each column, all the 1's are consecutive. If the rows of a matrix can be permuted so it becomes a P-matrix, we call it a* pre-P-matrix.

In other words, no 0's appear between any two 1's in a column in a P-matrix (see $\mathbf{C}$ in Figure 1b). To see that consistent responses with users sorted by abilities θ give a P-matrix, suppose for a contradiction that a column corresponding to an option for an item has two or more blocks of ones. Then the users corresponding to the zeros in between these blocks will have chosen another option for which the quality is strictly higher or lower than that of this option since the option qualities are assumed to be distinct. But this violates consistency since the rows are ordered by user ability.

Observation 1 (Consistent Responses imply C1P)**.** *A response matrix* $\mathbf{C}$ *is consistent iff it is a pre-P-matrix.*

Consequently, ranking the users in a scenario of consistent responses corresponds to the problem of determining a permutation of the rows of $\mathbf{C}$ so that the result obeys C1P.

State-of-the-art on C1P. Booth and Leuker [6] ("**BL**") proposed the fastest known algorithm for finding all possible permutations of the rows that reconstruct the C1P ordering in time linear in the number of nonzero entries in the matrix, taking time $O(mn)$ in our setting. However, their method fails to produce an ordering of the rows when the matrix is not a pre-P-matrix, and therefore cannot be used as a general heuristic for simulated or real-world datasets that are not ideal. In contrast, Atkins et al. [4] ("**ABH**") proposed an elegant spectral method to determine whether a matrix obeys C1P, thus giving a rare continuous method to solve a combinatorial ordering problem. Moreover, it can also be adapted as a heuristic when the input matrix is not a pre-P-matrix. To the best of our knowledge, this is the only currently known method that can be used for ability discovery that is also guaranteed to recover a C1P ranking if it exists.

Our goal. Our goal is to develop a fast and principled algorithm that just like ABH (i) returns a P-matrix in the special case of pre-P matrix inputs, and (ii) can solve the problem in the more general case when the response matrix is *not pre-P*. As we show, the performance of ABH quickly drops in the non-ideal setting (the general IRT setting in Section II-D) which makes it unusable for ability discovery. We show that our method is more robust and generalizes better, thus being the first method with a useful accuracy for ability discovery that is also guaranteed to solve the ideal case.

D. *Relation to Item Response Theory (IRT)*

Brief introduction of IRT models. A large body of work from psychological and educational researchers called *Item Response Theory (IRT)* [36], [57] studies the mathematical functions relating the probability of an examinee's response on a test item to an underlying ability. All major educational tests, such as the Scholastic Aptitude Test (SAT) [36] and Graduate Record Examination (GRE) [28], are based on IRT. IRT forms a mathematically principled, experimentally validated, and widely used theory on how users answer items. Figure 2 summarizes the connections between various IRT models, and our online appendix [9] contains the full details on IRT.

Dichotomous IRT models can be seen as variations of the standard logistic or sigmoid function $\sigma : \mathbb{R} \to [0, 1]$ defined by $\sigma(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$, which is widely used in machine learning models and a smooth relaxation of the Heaviside step function $H(z) = \mathbb{I}(z \geq 0)$ [40]. These models describe the probability $\mathbb{P}_i(\theta)$ for a student with ability θ to answer a question i correctly. We discuss here only the 3PL model (other binary models follow based on Figure 2). Parameter θ is the latent user ability, and a, b, c are latent item factors characterizing the questions and their options, representing discrimination, difficulty and random guessing, respectively:

$$\mathbb{P}_i^{\text{3PL}}(\theta) = c_i + (1-c_i)\sigma\big(a_i(\theta - b_i)\big) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta - b_i)}}$$

Multinomial IRT models measure the probability $\mathbb{P}_{ih}(\theta)$ for a student with ability θ to choose an option h for a question i. Thus different from binary models whose parameters belong to questions, multinomial IRT models have parameters *for each option*. For example, the Graded Response Model (GRM) model [50] assumes one discrimination a_i parameter for each question i, and one difficulty parameter b_{ih} per option:

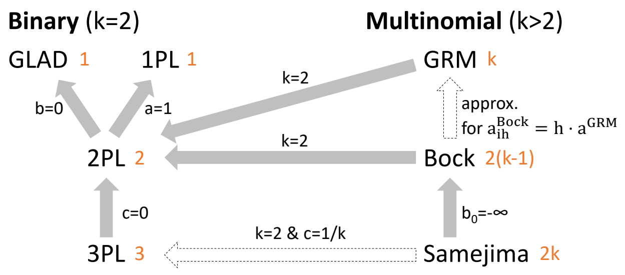


Fig. 2: Correspondences between the discussed IRT models. Orange numbers show number of free parameters per question. Arrows mean "specializes into." White arrows require special assumptions for the specialization: Bock to GRM holds only approximately after fixing $a_h^{\text{Bock}} = h \cdot a^{\text{GRM}}$, Samejima to 3PL for $k = 2$ when $c = 1/k$. It turns out that 2PL is isomorphic to standard logistic regression, and Bock is identical to multinomial logistic regression.

$$\mathbb{P}_{ih}^{\text{GRM}}(\theta) = \mathbb{P}_{ih}^{*\text{GRM}}(\theta) - \mathbb{P}_{i,h+1}^{*\text{GRM}}(\theta)$$
$$\mathbb{P}_{ih}^{*\text{GRM}}(\theta) = \sigma\big(a_i(\theta - b_{ih})\big) = \frac{1}{1 + e^{-a_i(\theta - b_{ih})}}$$
$$-\infty = b_{i0} < b_{i1} < \ldots < b_{i,k-1} < b_{ik} = \infty$$

The Bock model [5] further assigns a discrimination parameter a_{ih} for each option h, and the Samejima model [49] takes into account random guessing by adding a dummy option.

Option qualities. Notice that the option discrimination parameters in IRT determine the order of options in terms of quality (or correctness). Intuitively, the higher a_i for a question i is, the more "discriminating" it is (the probability of chosen the correct answers more quickly increases with student ability). The probability of answering a question correctly is then a function of the user ability and all the question and option parameters (discrimination scores and difficulties). This provides a flexible way to model the option quality.

Another way to model the option qualities is to define the option quality as a function of the user abilities of users who choose this option (e.g., HITS sums up the user abilities), as we discuss in Section III-A.

Connection between IRT and C1P. We introduced the definition of consistent responses in Section II-B, When a response matrix is consistent, it is a pre-P-matrix and can be permuted to become a P-matrix which has the consecutive ones property (C1P). If the response matrix is a pre-P-matrix, the corresponding response function of the probability for a user to choose a specific option $h \in \{0, \ldots, k-1\}$ can be expressed as the difference between two Heaviside step functions:

$$\mathbb{P}_{ih}(\theta) = H(\theta - b_{ih}) - H(\theta - b_{i,h+1})$$

for appropriately chosen b_{ih} such that:

$$-\infty = b_{i0} < b_{i1} < \ldots < b_{i,k-1} < b_{ik} = \infty$$

Notice that this response function is exactly the GRM model in the limit of $a \to \infty$: Whenever the user ability θ is between b_{ih} and $b_{i,h+1}$, then this student chooses option h.

To summarize, IRT models can be seen as a relaxed version of the response function in the ideal case when the response matrix can be permuted to obey C1P. As widely accepted models for MCQs, they strongly support our principle of satisfying the more strict C1P in the ideal case and can be used to generate data in non-ideal cases (see Section IV).

III. A FAMILY OF HITS ALGORITHMS

We review the HITS algorithm and variants that have been proposed for truth discovery. We then describe a natural averaging version of HITS that we call "AVGHITS." Our key observation is that the *eigenvector corresponding to the 2nd largest eigenvalue* of its update matrix reconstructs the user (or row) ordering with C1P if one exists and is unique.[4] We then describe a variant that we call "HITSNDIFFS" (or HND in short) on a tripartite graph to find this eigenvector efficiently: it keeps an additional vector of differences between adjacent scores and updates it in the loop of the AVGHITS algorithm to compute the ordering of users that we require. We prove that HND returns the correct ordering when the user responses are consistent and compare its time complexity and expected accuracy for non-consistent responses with other methods.

Required background from spectral graph theory. We say that $\mathbf{v}$ is an eigenvector of $(n \times n)$ quadratic matrix $\mathbf{A}$ with eigenvalue λ if $\mathbf{v} \neq 0$ and $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$. If $\mathbf{A}$ is symmetric, then all eigenvalues are real [31]. We use indices to refer to eigenvalues sorted algebraically: $\lambda_1 \geq \lambda_2 \geq \dots \lambda_{n-1} \geq \lambda_n$. We write $\mathbf{v}_i$ to refer to the eigenvector corresponding to eigenvalue λ_i and will be cavalier in referring to it as "the i-th largest eigenvector" when we really mean "the eigenvector corresponding to the i-th largest eigenvalue."

If the matrix is non-negative and irreducible, then according to the Perron-Frobenius theorem [20], [39], [45] the first eigenvector is also the largest by amplitude ($\lambda_1 = \max_i |\lambda_i|$) and the corresponding eigenvector $\mathbf{v}_1$ is positive.

A. "HITS" and its variants for truth discovery

Hubs and Authorities (HITS) [29][5] is a classic spectral approach that several truth discovery approaches have built upon. The original aim of the algorithm is to rate web pages by two scores: authority and hub score. These scores are recursively defined such that the hub scores are proportional to the sum of the authority scores of the nodes they point to, and the authority scores are proportional to the sum of the scores of the hubs pointing to them, thus reflecting a mutually consistent set of scores [41].

In the context of truth discovery, the authority and hub scores can be interpreted as the user abilities and option correctness scores, respectively. Let $\mathbf{C}$ represent the $m \times n$ binary response matrix where $C_{j,i} = 1$ iff user j chooses item i, and $\mathbf{s}$ be the m-dimensional user score vector, and $\mathbf{w}$ be the n-dimensional item weight vector. In matrix notation, the scores are recursively defined as:

$$\mathbf{s} \leftarrow \beta \mathbf{C}\mathbf{w} \qquad \mathbf{w} \leftarrow \alpha \mathbf{C}^\top \mathbf{s}$$

where α and β are normalization constants and $\mathbf{C}^\top$ is the transpose of $\mathbf{C}$. The algorithm starts from an initial assignment, such as $\mathbf{s} = \mathbf{e}$ and iteratively updates then normalizes $\mathbf{w}$ and $\mathbf{s}$. The user scores $\mathbf{s}$ will converge to the 1st eigenvector of the matrix $\mathbf{C}\mathbf{C}^\top$.

TruthFinder [64] modifies HITS by first taking *the average instead of the sum* of the chosen item scores as user scores and interpreting them as probabilities of the users being correct on any question. It then defines an item's score as the probability of it being true given the independent probabilities of all the users choosing the item. When appropriately initialized, the approach does not require normalization. In the following matrix formulation, let $\mathbf{C}^{\text{row}}$ represent the row-normalized response matrix $\mathbf{C}$:

$$\mathbf{s} \leftarrow \mathbf{C}^{\text{row}}\mathbf{w} \qquad \mathbf{w} \leftarrow \mathbf{1} - \exp\left(\mathbf{C}^\top \log(\mathbf{1} - \mathbf{s})\right)$$

Investment [44] calculates the ability of users as the sum of the scores of their chosen options, weighted by the user ability they invested in the previous iteration. **PooledInvestment** [44] extends Investment by using a different formula for the item scores. Both variants use non-linear scaling of the item scores with different user-specified hyperparameters.

Our method. We build upon this key idea of updating scores in a bipartite graph of users and items by iteratively summing scores from one side to update the other. However, in contrast to other methods, we focus on the 2nd largest instead of the dominant eigenvector of a new variant and show that this approach is *guaranteed to recover the correct ranking in case of consistent responses*. As we will show in Section IV, no other existing truth discovery method can do that.

B. "AVGHITS"

In our setup, there are nk different choices (k choices for each of n items). Consider a bipartite graph $G = (L \cup R, E)$ that corresponds to the $(m \times nk)$ response matrix $\mathbf{C}$: Partition L contains a vertex for each of m users: $L = \{u_1, ..., u_m\}$. Partition R is a collection of n vertex sets $R = \{I_1, ..., I_n\}$, one for each item. Each set I_i contains $k_i \leq k$ vertices $I_i = \{c_{i1}, ..., c_{ik_i}\}$ where c_{ih} represents option h of item i. We add an edge to between a user u_j and an option c_{ih} E if user j chooses option h for item i.

To make our derivations easier to follow, we will conveniently assume that each item i has the same number $k_i = k$ of options. Notice however, that this is not required for our approach. We further assume $\mathbf{C}$ to be connected. This requirement applies to all spectral truth ranking methods including HITS as otherwise the relative ranking of users (or items) from different components can't be established.[6] Finally, define $\mathbf{s}$ as a $(m \times 1)$ user score vector and $\mathbf{w}$ as a

[4] We consider an ordering and its reverse ordering to be the same.

[5] HITS originally stood for "Hyperlink-Induced Topic Search."

[6] PageRank achieves the connectivity with the teleport operation.

$(kn \times 1)$ option weight vector denoting weights for each of the kn options according to their order in $\mathbf{C}$.

We call AVGHITS the modification of the HITS update rule that uses *averages* instead of sums to iteratively update the user scores and option weights *in both directions*: the user score s_j is updated to be the average of the weights of all the options that user j picked, and an option weight c_{ih} is updated to be the average of the scores of all users who picked it. In the following matrix formulation, let $\mathbf{C}^{\text{row}}$ represent the row-normalized and $\mathbf{C}^{\text{col}}$ the column-normalized response matrix $\mathbf{C}$. At each iteration (until convergence), we update the user score vector $\mathbf{s}$ and the option weight vector $\mathbf{w}$ as follows:

$$\mathbf{s} \leftarrow \mathbf{C}^{\text{row}}\mathbf{w} \qquad\qquad \mathbf{w} \leftarrow (\mathbf{C}^{\text{col}})^{\top}\mathbf{s}$$

By combining the above two update equations, we can update user scores between iterations directly by replacing the two normalized response matrices with one *update matrix* $\mathbf{U}$:

$$\mathbf{s} \leftarrow \underbrace{\mathbf{C}^{\text{row}}(\mathbf{C}^{\text{col}})^{\top}}_{\mathbf{U}}\mathbf{s} \tag{1}$$

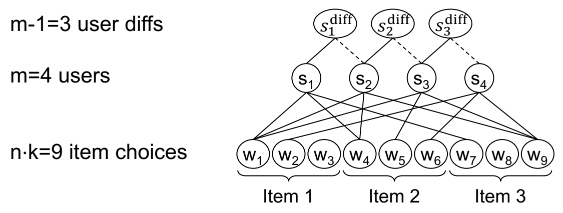
These iterations are not yet very helpful. Indeed, we observe that the largest eigenvector of $\mathbf{U}$ is the all-ones vector $\mathbf{e}$, and this is the vector of user scores that AVGHITS converges to. It turns out that it is the eigenvector corresponding to the *2nd largest eigenvalue* of $\mathbf{U}$ that we seek.

C. *Our algorithm "HITSNDIFFS" (HND)*

In the following, we show a simple algorithm to find the 2nd largest eigenvector ordering of $\mathbf{U}$ and prove that it can be used to find the unique consecutive ones ordering of the response matrix $\mathbf{C}$. By "the eigenvector ordering", we mean the ranking of entries in this eigenvector in terms of their values. For example, $\mathbf{v}_1 = \{0.36, 0.8, 0.48\}$ and $\mathbf{v}_2 = \{0.48, 0.64, 0.6\}$ have the identical ordering $\{3, 1, 2\}$ or its reverse $\{1, 3, 2\}$. Our algorithm does not return the 2nd largest eigenvector of $\mathbf{U}$ but instead returns a vector with the *identical ordering*.

The 2nd largest eigenvector of a matrix can be found using a variant of the deflation method [38], [40], which we will discuss in detail in Section III-F. Here we present a novel, conceptually simple, and faster algorithm that we term "HITS and DIFFS" (HITSNDIFFS or HND) that extends AVGHITS from a bipartite to a tripartite graph and whose iterative updates converge to a user ranking that is *guaranteed to be* C1P *in the ideal case*, and that *performs well also in more general settings*. This approach leverages particular properties of our problem that don't apply to the 2nd largest eigenvector orderings of any matrix from more general settings.

First, we propose a new intermediate step that calculates differences between user scores in the iterative updates of AVGHITS. Rather than updating the user scores iteratively, HITSNDIFFS updates the *differences* between adjacent user scores by using a suitably modified update matrix, and results in the scores converging to the ordering according to the second largest eigenvector of $\mathbf{U}$. Furthermore, this modification only adds a linear overhead of computing the user score difference vectors and normalizing it in every iteration. As



$$\mathbf{S} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \qquad \mathbf{T} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

Fig. 3: HITSNDIFFS uses a *3-partite graph* of option weights, user scores, and user diffs. Contrast this graph with Figure 1. The update equations (see Algorithm 1) use two re-shaping matrices $\mathbf{S}$ and $\mathbf{T}$.

we will show in Theorem 2, when the response matrix obeys C1P, then HND reconstructs the correct ordering of the rows.

As shown in Figure 3, we define a new vector $\mathbf{s}^{\text{diff}}$ of differences in user scores with entries $s_j^{\text{diff}} = s_{j+1} - s_j$, $j \in [m{-}1]$. This is equal to $\mathbf{s}_j^{\text{diff}} = \mathbf{S}\mathbf{s}$ where $\mathbf{S} \in \mathbb{R}^{(m-1)\times m}$ is shown in Figure 3. In the reverse direction, there are infinitely many vectors $\mathbf{s}$ that can be generated from a given $\mathbf{s}^{\text{diff}}$, all shifted by different constants. Since we only want a final ordering of users, we can WLOG set the first element of the vector $\mathbf{s}$ to be 0. The transformation then is $\mathbf{s} = \mathbf{T}\mathbf{s}^{\text{diff}}$ where $\mathbf{T} \in \mathbb{R}^{m\times(m-1)}$ is the lower unit triangular matrix[7].

We can now get a user difference score update rule:

$$\mathbf{s}^{\text{diff}} \leftarrow \mathbf{S}\mathbf{s} = \mathbf{S}\mathbf{C}^{\text{row}}(\mathbf{C}^{\text{col}})^{\top}\mathbf{s} = \underbrace{\mathbf{S}\mathbf{C}^{\text{row}}(\mathbf{C}^{\text{col}})^{\top}\mathbf{T}}_{\mathbf{U}^{\text{diff}}}\mathbf{s}^{\text{diff}} \tag{2}$$

In other words, $\mathbf{U}^{\text{diff}} = \mathbf{SUT}$ is a "difference update" matrix that is used to update $\mathbf{s}^{\text{diff}}$ from one iteration to the next. With these update equations, $\mathbf{s}^{\text{diff}}$ converges to the largest eigenvector of $\mathbf{U}^{\text{diff}}$. Our algorithm HND that implements this is described in Algorithm 1.

We now prove the connection between the 1st eigenvector of $\mathbf{U}^{\text{diff}}$ and the 2nd largest eigenvector of $\mathbf{U}$.

Lemma 1 (Eigenvector correspondence)**.** *$\mathbf{x}$ is the 2nd largest eigenvector of $\mathbf{U}$ iff $\mathbf{y} = \mathbf{S}\mathbf{x}$ is the largest eigenvector of $\mathbf{U}^{\text{diff}}$.*

Proof sketch. Due to the limit of space, we only provide the high-level ideas of our proofs in the paper. First, we can find out that each row of $\mathbf{U}$ has sum 1. Using this, we can prove that the largest eigenvector of $\mathbf{U}$ is in the direction of the all ones vector $\mathbf{e} = \mathbf{1}_m$ if the largest eigenvalue of $\mathbf{U}$ has multiplicity 1 (i.e. the graph is a single connected component). Let $\mathbf{x}$ be an eigenvector of $\mathbf{U}$ that is not in the direction of the all ones vector, i.e. $\mathbf{x} \neq \alpha\mathbf{e}$. Note that $\mathbf{TS} = (\mathbf{I}_m - \mathbf{e}\mathbf{e}_1^{\top})$

[7]It is this fixing that intuitively keeps the ordering, but changes the actual amplitudes.

Algorithm 1: HITSNDIFFS (HND-power): A fast implementation of equation (2) to calculate the 2nd eigenvector ordering of $\mathbf{U} = \mathbf{C}^{\text{row}}(\mathbf{C}^{\text{col}})^\top$

```
Input: Response matrix $\mathbf{C}$, randomly initialized user scores $\mathbf{s}_0$
Output: User scores $\mathbf{s}$
1: $\mathbf{s}^{\text{diff}} \leftarrow \mathbf{s}_0^{\text{diff}}$          // initialize user score differences
2: repeat
3:     $\mathbf{s} \leftarrow \mathbf{T}\mathbf{s}^{\text{diff}}$          // update user scores
4:     $\mathbf{w} \leftarrow (\mathbf{C}^{\text{col}})^\top \mathbf{s}$          // update option weights
5:     $\mathbf{s} \leftarrow \mathbf{C}^{\text{row}}\mathbf{w}$          // update user scores
6:     $\mathbf{s}^{\text{diff}} \leftarrow \mathbf{S}\mathbf{s}$          // update user score differences
7:     Normalize $\mathbf{s}^{\text{diff}}$ to be a unit vector
8: until convergence or iteration limit
9: $\mathbf{s} \leftarrow \mathbf{T}\mathbf{s}^{\text{diff}}$
```

and each row of $\mathbf{SU}$ sums to 0. Then,

$$\begin{aligned}
\mathbf{Ux} &= \lambda\mathbf{x} \\
\mathbf{SUx} &= \lambda\mathbf{Sx} \\
\mathbf{SU}(\mathbf{I}_m - \mathbf{e}\mathbf{e}_1^\top)\mathbf{x} &= \lambda\mathbf{Sx} \\
\mathbf{SUTSx} &= \lambda\mathbf{Sx} \\
\mathbf{U}^{\text{diff}}\mathbf{y} &= \lambda\mathbf{y}, \quad \text{where} \quad \mathbf{y} = \mathbf{Sx} \qquad (3)
\end{aligned}$$

Therefore, $\mathbf{U}^{\text{diff}}$ has exactly the same eigenvalues as $\mathbf{U}$ except the largest eigenvalue 1, and the eigenvectors of $\mathbf{U}^{\text{diff}}$ are the differences between the entries of the corresponding eigenvector of $\mathbf{U}$. Thus we prove the lemma. □

Theorem 1 (2nd eigenvector of AVGHITS recovers C1P)**.** *If* $\mathbf{C}$ *is a pre-P-matrix with a unique consecutive ones ordering of its rows and each row has the same row sum, then this ordering of the rows of* $\mathbf{C}$ *is given by the ranking of the rows sorted by values in the 2nd largest eigenvector of* $\mathbf{U}$.

Proof sketch. We can first prove that if $\mathbf{C}$ is a pre-P-matrix with a unique consecutive ones ordering of its rows and each row has the same row sum, $\mathbf{U}$ is an R-matrix (defined in [4]) where in each row and column, the entries closer to the diagonal are larger than or equal to the further entries. Using this, we can prove every entry in $\mathbf{U}^{\text{diff}}$ is non-negative by computing each entry in $\mathbf{U}^{\text{diff}}$ step by step according to its definition, which means $\mathbf{U}^{\text{diff}}$ is a non-negative matrix. We can now apply the Perron-Frobenius Theorem [20], [45]: there exists a non-negative eigenvector of $\mathbf{U}^{\text{diff}}$ corresponding to the largest eigenvalue of $\mathbf{U}^{\text{diff}}$. We know $\mathbf{U}^{\text{diff}}$ has exactly the same eigenvalues as $\mathbf{U}$, except the largest eigenvalue 1, and the eigenvectors of $\mathbf{U}^{\text{diff}}$ are the differences between the elements of the corresponding eigenvector of $\mathbf{U}$. Since the differences between the elements of the eigenvector corresponding to the 2nd largest eigenvalue of $\mathbf{U}$ (largest eigenvalue of $\mathbf{U}^{\text{diff}}$) is non-negative, that eigenvector of $\mathbf{U}$ is monotonic. Therefore, sorting the rows according to the second largest eigenvector ordering of the corresponding update matrix $\mathbf{U}$ gives a P-matrix, proving the theorem. □

Theorem 2. *If* $\mathbf{C}$ *is a pre-P-matrix with a unique C1P ordering of its rows and each row has the same row sum, then* HND *reconstructs the consistent ordering of the users taking linear time in the number of nonzeros in* $\mathbf{U}$ *per iteration.*

Proof sketch. From Lemma 1, we know by converting the converged largest eigenvector of $\mathbf{U}^{\text{diff}}$ back into a user score, we regain the ordering of the rows according to values in the second largest eigenvector of $\mathbf{U}$. This, along with Theorem 1, proves this theorem that HND detailed in Algorithm 1 reconstructs the ideal consistent ordering. □

D. Decile entropy-based symmetry breaking

Notice that reversing the order of a P-matrix still leaves it as a P-matrix. Thus all methods for solving C1P suffer from a natural *symmetry breaking problem*: they have to decide between the order returned by an algorithm *or its exact inverse*.

Our solution to this symmetry-breaking problem is motivated by the following observation: *users with higher ability tend to converge on the correct option as a majority answer*, while users with lower ability at the other end of the ordering tend to answer randomly. This idea is similar to the main argument in [33] that experts tend to answer similar correct answers. Thus the lower end of the user ordering has a *higher entropy* in the choices picked than the higher quality end. Notice that this idea is also implicit in IRT models with random guessing where users with low ability choose uniformly random among the options (hence have high entropy), whereas users of high ability pick the single correct choice.

We operationalize this idea in a new heuristic that is very effective in practice: Given a ranking of the users, we compute, for the top and the bottom user decile, the average entropy of the chosen item options across all items. We pick the side with lower entropy as the users with higher quality. We use this "*decile entropy method*" for both HND and ABH in our experiments.

E. Why HND *works better than ABH*

HND and ABH rely on strikingly similar intuitions about spectral properties of matrices: HND ranks users by the 2nd largest eigenvector of $\mathbf{U}$ (whose difference is the largest eigenvector of $\mathbf{U}^{\text{diff}} = \mathbf{S}\mathbf{C}^{\text{row}}(\mathbf{C}^{\text{col}})^\top\mathbf{T}$), whereas it can be shown that ABH ranks users by the 2nd smallest eigenvector of the Laplacian matrix $\mathbf{L}$ of $\mathbf{C}\mathbf{C}^\top$ (whose difference is the smallest eigenvector of $\mathbf{M} = \mathbf{SLT}$). In an ideal scenario with consistent responses (thus in an IRT scenario with very large discriminations), both methods are guaranteed to return the correct C1P ordering.[8]

We can also expect the accuracy to be identical in the other extreme scenario where all questions have 0 discrimination. But how can we expect their accuracy to compare in the more general scenario?

We interpret the general IRT scenario as random perturbations [54] from the ideal C1P case. Now notice that the smallest eigenvector of $\mathbf{M}$ is identical to the largest eigenvector of $\beta\mathbf{I}_{m-1} - \mathbf{M}$ where β is larger than all the entries and

[8]Recall that they are guaranteed to return the same ordering, but not the same eigenvector.

all the eigenvalues of $\mathbf{M}$.[9] Thus the comparison of HND and ABH corresponds to the largest eigenvector of $\mathbf{U}^{\text{diff}}$ against $\beta\mathbf{I}-\mathbf{M}$. Since both matrices have all non-negative entries in the ideal scenario, we know from the Perron-Frobenius theorem [20], [45] that all values in their largest eigenvector are non-negative.

The user score of the kth user equals to the cumulative sum of the first $k-1$ entries in the eigenvector. In the ideal case when C is a C1P matrix, every entry of $\mathbf{s}^{\text{diff}}$ is non-negative so $\mathbf{s}$ give us a perfect ranking of the students. In the non-ideal scenario when C is not a C1P matrix and the users are permuted by their abilities, the sign of the entries in $\mathbf{s}^{\text{diff}}$ change. When the eigenvector is even, a simple sign change in one of the entries does not influence the entire ranking of $\mathbf{s}$ but when the eigenvector has a large variance, a simple sign change in one large entry can break the entire ranking. For example, if the kth entry of $\mathbf{s}^{\text{diff}}$ is quite large but the sign is negative, the error of the ranking of the $(k+1)$th students can be very large.

Based on our observation above, we expect HND to work better than ABH as the variance of largest eigenvector of $\beta\mathbf{I}-\mathbf{M}$ should be much larger compared to $\mathbf{U}^{\text{diff}}$. The result is verified with dedicated experiments in Section IV-D: Figure 6a shows our observations on the variances of the $\mathbf{s}^{\text{diff}}$ for $\beta\mathbf{I}-\mathbf{M}$ and $\mathbf{U}^{\text{diff}}$. Figure 6b and Figure 6c verify that ABH is less accurate and less stable than HND.

F. Complexity Comparison

We analyze the asymptotic time complexity of the various methods. We compare HND against (i) existing C1P reconstruction algorithms BL and ABH, and (ii) the deflation method [38], [40] as an alternative method to compute the 2nd largest eigenvector for AVGHITS.

The time complexity depends on the number of users m, the number of questions n, and the number of iterations t which may be different for different methods. We assume $t \ll n$ and $t \ll m$ and thus only focus on m and n. Notice that although the response matrix $\mathbf{C}$ is a $(m \times kn)$-matrix, it has only $\mathcal{O}(mn)$ non-zero entries since every user can pick only one label per question.

HITSNDIFFS. A naive way to calculate the ranking is to first compute $\mathbf{U}^{\text{diff}}$ and then use the power method on it. However, computing $\mathbf{U}^{\text{diff}}$ requires a matrix-matrix multiplications before the iterations with time complexity $\mathcal{O}(m^2 n)$. Since $\mathbf{U}^{\text{diff}}$ is a $(m-1)\times(m-1)$ matrix, the time complexity to run the power method on $\mathbf{U}^{\text{diff}}$ is $\mathcal{O}(m^2)$ per iteration. This gives a total time complexity for the naive implementation as $\mathcal{O}(m^2 n) + \mathcal{O}(m^2 t) = \mathcal{O}(m^2 n)$. By instead running the mutual updates of $\mathbf{w}$, $\mathbf{s}$ and $\mathbf{s}^{\text{diff}}$ as described in Algorithm 1, we can replace matrix-matrix multiplications with several matrix-vector multiplications and thereby get a more efficient implementation of HITSNDIFFS in $\boxed{\mathcal{O}(mnt)}$. In other words, the speed-up results from applying the associativity law and replacing the calculation $\mathbf{s} \leftarrow (\mathbf{S}\mathbf{C}^{\text{row}}(\mathbf{C}^{\text{col}})^\top\mathbf{T})\mathbf{s}^{\text{diff}}$ with $\mathbf{s} \leftarrow \mathbf{S}(\mathbf{C}^{\text{row}}((\mathbf{C}^{\text{col}})^\top(\mathbf{T}\mathbf{s}^{\text{diff}})))$. One detail is that we need however to implement Line 3 differently. Materializing the matrix $\mathbf{T}$ would take $\mathcal{O}(m^2)$. Instead, we calculate the entries for $\mathbf{s}$ from $\mathbf{s}^{\text{diff}}$ via cumulative summation (e.g. via `numpy.cumsum` in Python).

The deflation method. Theorem 1 showed that our problem can be formulated as finding the 2nd largest eigenvector $\mathbf{v}_2$ of $\mathbf{U}$. The problem of finding the eigenvector corresponding to the second largest eigenvalue of a given matrix $\mathbf{A}$ can be solved with the *deflation method* [38], [40]. The idea is to first calculate the dominant eigenvector $\mathbf{v}_1$, and then eliminate the influence of the $\mathbf{v}_1$ from $\mathbf{A}$ to get a new matrix $\mathbf{B}$ whose 1st eigenvector is the 2nd eigenvector of $\mathbf{A}$. Then $\mathbf{v}_2$ of $\mathbf{A}$ can be obtained by using the power method on $\mathbf{B}$. We next argue (and later show experimentally in Section IV-C) that using the deflation method is slightly less efficient than HND (in addition to being not as simple to formulate as HND-power).

The most widely known deflation method [38], [40] only works for symmetric matrices and does not apply to the asymmetric $\mathbf{U}$. [60] presents several more variants of the deflation method including some that work for non-symmetric matrices. Most of those methods either require matrix-matrix multiplication or both the left dominant eigenvector and the right dominant eigenvector. The only exception is Wilkinson's vector annihilation [62] which only needs the right dominant eigenvector (which we know is a unit vector in the direction of the all ones vector in our case). However, [60] claimed that this method is difficult to apply in practice because of the need to conduct annihilation between the power iterations and we found no open-source implementation.

For our experiments in Section IV we implement *Hotelling's matrix deflation* [61] which uses both the left and right largest eigenvectors and thus requires one more round of the power iteration.[10] The experimental result in Section IV-C verifies that HND is not just conceptually simpler but also slightly more efficient than the deflation method.

ABH [4]. To reconstruct the C1P ordering, ABH requires the computation of the Fiedler vector [17], which is the eigenvector corresponding to the 2nd smallest eigenvalue of the related Laplacian matrix $\mathbf{L}$.

The original ABH paper [4] does not propose an explicit solution and instead refers to the Lanczos algorithm [30], [43], whose time complexity is $\mathcal{O}(dmt)$ with d being the average number of non-zero entries in a row of a given $(m\times m)$ matrix $\mathbf{A}$. When the Laplacian matrix is dense as in our scenarios the time complexity of the Lanczos algorithm is $\mathcal{O}(m^2 t)$. It is efficient for eigenvector computations on large symmetric matrices [10], and the state-of-the-art Fiedler vector solver [25] uses Lanczos. However, implementation by libraries such as Scipy [58] and Tenpy [23], require the full matrix as input, which would require us to compute the Laplacian matrix first.

[9]To see that, assume $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$. Then $(\mathbf{A}+\beta\mathbf{I})\mathbf{v} = \mathbf{A}\mathbf{v} + \beta\mathbf{I}\mathbf{v} = \lambda\mathbf{v} + \beta\mathbf{v} = (\lambda+\beta)\mathbf{v}$. Thus if $\mathbf{v}$ is an eigenvector of $\mathbf{A}$ with eigenvalue λ, then $\mathbf{v}$ is also an eigenvector of the spectrally shifted matrix $\mathbf{A}+\beta\mathbf{I}$, but with eigenvalue $\lambda+\beta$.

[10]We first calculate the dominant left eigenvector via power iteration, then deflate the matrix, and then calculate the dominant right eigenvector on the deflated matrix.

This calculation involves matrix-matrix multiplications and, as we show in Section IV-C, results in $\boxed{\mathcal{O}(m^2n)}$.

We provide another solution for ABH which is not in the original paper [4]. Similar to how we implement HND as Algorithm 1, we can also implement ABH by using the power method on the matrix $\beta\mathbf{I}_{m-1} - \mathbf{M}$ to get its largest eigenvector without having matrix-matrix multiplications. As we discussed in Section III-E, this largest eigenvector of $\beta\mathbf{I}_{m-1} - \mathbf{M}$ is identical to the smallest eigenvector of $\mathbf{M}$ which can be used to compute the order of the 2nd smallest eigenvector of $\mathbf{L}$ in the same way as Algorithm 1. The total time complexity for this algorithm is $\boxed{\mathcal{O}(mnt + m^2t)}$. When m and n are close or $m < n$, the time complexity becomes $\mathcal{O}(mnt)$ which is the same as HND. However, when $m \gg n$, the time complexity becomes $\mathcal{O}(m^2t)$, which is larger than $\mathcal{O}(mnt)$ of HND.

BL [6]. The original paper by Booth and Leuker (BL) [6] for reconstructing the C1P property can work directly on the initial response matrix and runs in $\mathcal{O}(mk + n + f)$, where f is the non-zero entries in the matrix. In our setup where $f = \mathcal{O}(mn)$, the time complexity is $\mathcal{O}(mn)$. Therefore, BL is the fastest method when it works. Since it cannot be used for solving ability discovery in general, we are not using it in our experiments.

HITS [29], Truthfinder [64], Investment [44], PooledInvestment [44]. All these methods are iteration-based variants of HITS that take at least $\mathcal{O}(mnt)$ and differ in how they iterate between the user and the item scores. In practice, only HITS can be defined as an eigenvector problem with a closed-form solution and efficient linear algebra implementation. Truthfinder converges in practice, while Investment and PooledInvestment can not converge and return different results depending on initialization. Our approach in contrast comes with the same computational properties as HITS: guarantee of convergence, unique solution, an intuitive formulation as a spectral problem, and an efficient matrix implementation.

IV. Experiments

Our experiments compare the *accuracy* of ability discovery and the *scalability* of the various methods. The main take-aways from the experiments are: 1) HND robustly returns rankings for users with accuracy on average *better than or equal to* other truth discovery methods; 2) HND is competitive with two "*cheating competitors*" (that are provided the ground truth information about the correct options for each question that is usually not available); 3) HND has better scalability as a C1P reconstruction algorithm than ABH.

A. Experimental setup

Environment. All scalability experiments are run on Intel Xeon E5-2680 CPUs with an exclusive environment and 128G allocated memory. We implemented HND in Python 3.8.1.

Benchmarks. [52] provides an extensive benchmark for the *truth discovery problem*, and [69] provides a broad survey of existing truth discovery methods. Two points stand out: 1) All open-source datasets used in the two papers *lack ground truth for user abilities*. 2) All 20 datasets fall under the setting with *homogeneous items*. As we discussed in Section I, it is easy to understand the lack of ground truth for the ability discovery problem because the *user abilities are abstract and not able to be obtained from external knowledge*. To make up for it, we first create synthetic data based on the polytomous models from *Item Response Theory (IRT)* (recall Section II-D). Moreover, we use a real-world MCQ dataset with approximate (but not accurate enough) ground truth as a supplementary evidence to verify the usefulness of HND in Section IV-E.

Polytomous synthetic data generator. We use the three polytomous IRT models from Section II-D (GRM [50], Bock [5] and Samejima [49]) to generate synthetic data sets with known ground truth. Samejima model takes random guessing into account so it models the educational test scenario where students try to maximize their scores. Bock and GRM models with no random guessing models the crowdsourcing scenario where workers usually do not guess.

By default, we set user ability θ to be within $[0, 1]$, item difficulty b to be within $[-0.5, 0.5]$, and the item discrimination a to be within $[0, 10]$, all uniformly random. Besides varying the number of users, items and options, Section IV-B also has experiments with shifted b's (chosen to achieve a certain percentage of users giving correct answers).[11]

Methods and their implementations. For a thorough evaluation, we created three alternative implementations of HND and two alternative implementations of ABH. **HND-power** follows Algorithm 1 which only involves matrix-vector multiplications. **ABH-power** is our novel kimplementation of ABH that avoids matrix-matrix multiplications by using the power method on the matrix $\beta\mathbf{I}_{m-1} - \mathbf{M}$. **ABH-direct** is the implementation of ABH using the Lanczos algorithm as suggested by [4]. Section III-F discussed the drawback of requiring matrix-matrix multiplications. We use an efficient sparse Linear Algebra Python library called Scipy [58]. **HND-direct** implements HND similarly by directly computing the 2nd largest eigenvector of $\mathbf{U}$ by using the Arnoldi algorithm [3], which can be considered the general version of the Lanczos algorithm on asymmetric matrices, also using Scipy. **HND-deflation** implements HND with the deflation method discussed also in Section III-F. For HND-power, ABH-power and HND-deflation, the criterion for convergence is a maximal L2-norm of 10^{-5} over the change. For our experiments (other than Section IV-C) we used "HND-power" for HND and "ABH-direct" for ABH since they turned out to be the fastest implementations.

We also implemented **HITS** [29], **TruthFinder** [64], **Investment** and **PooledInvestment** [44]. Since none of those iteration-based approaches (except HITS) allows an efficient matrix formulation, our implementation in Python uses loops and is not efficient. We thus do not report scalability experiments on those methods as native implementation in C++ would bring those close to HND as discussed in Section III-F.

[11]For experiments with GRM data, we use the data generator from the GIRTH package which requires at least $k = 3$ options. We implemented Bock and Samejima generators ourselves and thus options can start from $k = 2$.

For Investment and PooledInvestment (which do not converge) we use 10 iterations instead of tuning the number of iterations.

Two cheating baselines. To show the effectiveness of HND, we also compare HND with two "cheating competitors" that are given additional ground truth information about questions that is usually not available: **True-answer** has information about which choices are correct for each question (which is usually not known in our scenario) and then ranks users by the number of correctly answered questions. **GRM-estimator** uses a Python package called GIRTH [51] that estimates the parameters of a GRM model including user abilities. However, it *requires knowing the order of options* for each question by correctness. Our comparison with this approach is notable because it is the theoretically "best" model to fit data generated by the same synthetic GRM process.

B. Accuracy on synthetic data

Accuracy. To determine the accuracy of a method, we calculate *Spearman's rank correlation coefficient* [53] between the returned user ranking and the ground truth ranking by their actual abilities. Spearman's correlation is defined as the *Pearson correlation between the rankings* of two scoring functions and ranges between -1 and 1. It is similar to Kendall's correlation, yet strictly preferred if there are ties in the data [46]. There can be negative accuracy at times (not shown in Figure 4), which means the returned ranking is negatively correlated or random whose coefficient is near 0.

Setup. We conduct experiments to determine the accuracy as function of the 1) *number of items* n, 2) *number of users* m, 3) *number of options* k, 4) *option difficulties* b_{ih}, and 5) *probability of questions to be answered* p. In the first experiment, we use data generated according to the three polytomous models (GRM, Bock, Samejima) from Section II-D to also show the robust performance of HND for all three models. In other experiments, we only use data generated according to the Samejima model since it is the most general one to avoid redundancy. To verify the ability of algorithms to recover a C1P ranking, we also generate data that 6) *follows the consistent response property* (which as discussed in Section II-C is the case for IRT models when the discrimination $a_{ih} \to \infty$). Every question has the same number of options, and every user answers every question. By default, we set the users $m = 100$, items $n = 100$, and options $k = 3$.

1. Varying number of questions n (Figures 4a to 4c[12]). HND has better than or equal accuracy as the other methods even including the two cheating competitors over data generated by all three models. Notice that the GRM-estimator works poorly for Samejima as it does not model random guessing.

2. Varying number of users m (Figure 4d). HND works also better than or equal to other approaches (except for the data point with low m where the cheating competitors win).

3. Varying number of options k (Figure 4e). HND stays top and accurate, even slightly outperforming True-answer.

4. Varying question difficulties b_{ih} (Figure 4f). Here, we change the difficulty range from the default $[-0.5, 0.5]$ to 7 different ranges, $[-1, 0]$, $[-0.75, 0.25]$, $[-0.5, 0.5]$, $[-0.25, 0.75]$, $[0, 1]$, $[0.25, 1.25]$, $[0.5, 1.5]$, while user abilities θ_j remain at $[0, 1]$. Thus even the least able user has a high probability to answer a difficult question in the easiest setting, while even the best user can be incorrect for some easy questions in the hardest setting. The x-axis here is the *average accuracy* on the questions across all the users. In all scenarios, we see HND outperforms other competitors.

5. Varying probability p of answering a question (Figure 4g). To show that HND works for more general scenarios where users *answer different number of questions*, we vary the probability of questions to be answered. For each pair of question and user, there is the probability of p for the user to answer the question. We see that HND performs well even when the dataset is not complete.

6. C1P (Figure 4h).[13] In addition to the three multinomial IRT models, we also generate response matrices that are consistent and can be reconstructed to be a P-matrix. These responses correspond to a random GRM instance with very strong discrimination a. We use these matrices to verify the effectiveness of HND in reconstructing a C1P permutation. To avoid ties in the rankings and provide a unique C1P ordering, we set both the user ability θ and the difficulty parameter b to be within $[0, 1]$, randomly chosen. We see that HND and ABH are indeed the only two methods that can reconstruct the C1P permutation if there exists one.

Summary. HITSNDIFFS is a *robust* method that outperforms the other approaches in most setups, especially those with high discrimination. We see this as vindication for designing an approach based on the principle that *consistent answers need to be solved correctly*. HND is also competitive even against the two cheating approaches which have the best item answer given (i.e. they have access to an oracle that can solve the entire problem of truth discovery). Moreover, we verified that HND and ABH are indeed the only ones that can reconstruct a C1P permutation if it exists.

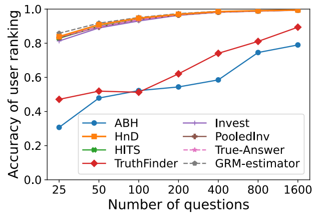
C. Scalability experiments

Figures 5a and 5b show the scalability in number of users (m) and questions (n) of our various implementations of ABH and HND, as well as the GRM-estimator. Each shown data point is the median over 5 runs, and we set a timeout of 1,000 seconds. **HND vs. ABH.** Figure 5a shows that ABH-direct and HND-direct scale with $O(m^2 k)$ in the number of users as predicted in Section III-F. The theoretic time complexity of ABH-power is $O(m^2 t)$ when m is much larger than n, and thus it also takes quadratic time. In contrast, HND-power can scale linearly and is about 20% faster than HND-deflation on average for $m > 1000$ users as it needs only one round of the power method. Figure 5b shows that although ABH-direct is slightly faster for fixed few users, all implementations are efficient even for a large number of questions.

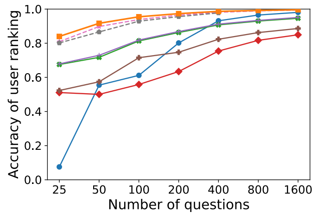
GRM-estimator. As a representative of max likelihood parameter estimation, the GRM-estimator is expected to perform

[12]The GRM estimator does not work when the question number is large.

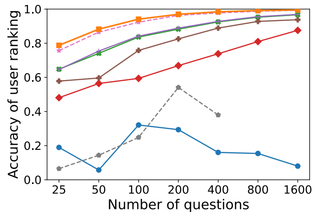
[13]In the experiments, PooledInv returned all negative coefficients but we consider its ranking to be the reverse one.



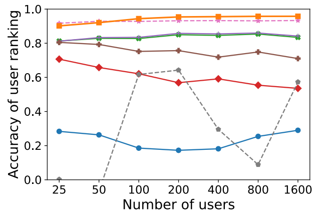
(a) Varying n (GRM)



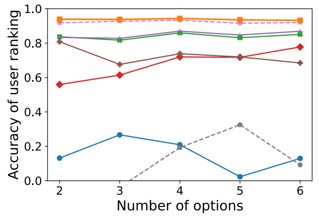
(b) Varying n (Bock)



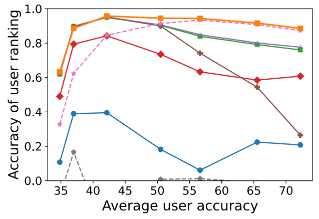
(c) Varying n (Samejima)



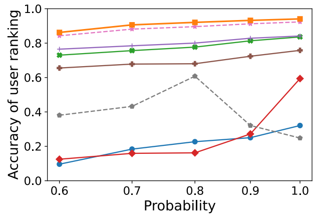
(d) Varying m (Samejima)



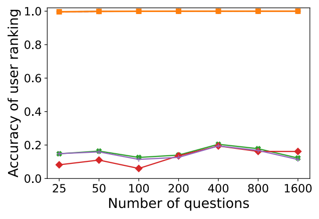
(e) Varying k (Samejima)



(f) Varying b_{ih} (Samejima)

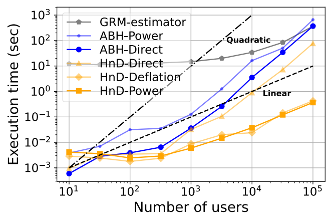


(g) Varying p (Samejima)

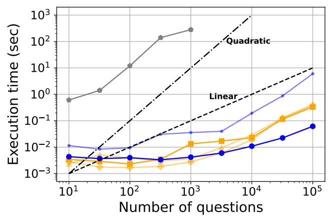


(h) Varying n (C1P)

Fig. 4: Section IV-B: Results of accuracy experiments (the legend is in the first figure).



(a) Scalability with users (m).



(b) Scalability with items (n).

Fig. 5: Section IV-C: Scalability experiments with $n = 100$ items and increasing numbers of users m in (a), or $m = 100$ users and increasing numbers of items n in (b). The experiments confirm that our method (HnD) scales linearly in the number of items and users, whereas ABH (even trying various alternative methods) has an unavoidable quadratic scalability in the number of users.

best on GRM data. However, Figure 5 shows that it is by orders of magnitude slower than HND.

Summary. HND scales asymptotically and practically better than the other existing C1P reconstruction algorithm ABH in the number of users. Moreover, our intuitive Algorithm 1 is slightly faster than an adaptation of the deflation method.

D. Stability experiments for ABH and HND

We next experimentally verify our prediction from Section III-E that HND generalizes better from the ideal case than ABH. In this setup, we fix $m = 100$ users, $n = 100$ items, $k = 3$ options, with user abilities and item difficulties equally spaced between $[0, 1]$ and $[-0.5, 0.5]$ respectively. For one item, all the option difficulties are the same. All items have identical discrimination a and all the options in one question have equally spaced a (as in the GRM model). We then vary the discriminations and compare the (i) variance of the respective eigenvectors used for ranking; (ii) the normalized average difference in rank between each user's ranking;[14] and (iii) the average accuracy of the predicted rankings for HND and ABH across repeatedly sampled response matrices.

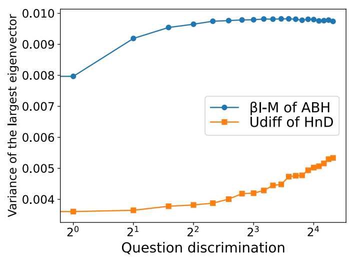
[14]Here difference means the average difference of each user's rankings from different runs, scaled down to $[0, 1]$ by the user number.

Figure 6a shows our observation from Section III-E that the variance of the largest eigenvector of $\mathbf{U}^{\text{diff}}$ of HND is much smaller than $\beta\mathbf{I}_{m-1} - \mathbf{M}$, which is expected to lead to the better stability and accuracy of the HND rankings. Figure 6b confirms that the ranking of a user is more stable for HND. Figure 6c shows the resulting increase in the accuracy of HND over ABH. This confirms our original goal to develop a spectral method that can achieve the same C1P ranking as ABH, yet generalizing better in the non-ideal case.

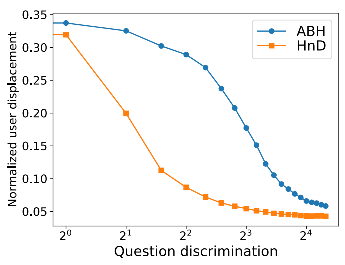
E. Accuracy experiments on real-world data

As mentioned in Section IV-A, we do not know *any existing benchmark with a known true ranking* of users by their abilities. In order to still verify the performance of HND on real-world datasets we use the ranking of the *"True-answer"* baseline as the ground truth. Notice that although this baseline performs well in our synthetic experiments, it is far from the perfect gold standard (sometimes even outperformed by HND) so the experimental result in this subsection should be seen only as a supplementary evidence. The six used real world MCQ datasets are from [33].

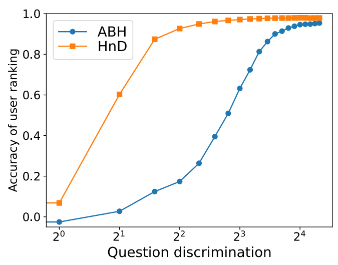
Figure 7 shows the average experimental result of six datasets where PooledInvestment and HITS perform slightly better than HND. However, we need to emphasize several points: (1) All datasets are very small in terms of question numbers (from 20 to 36) but have double user numbers on average, which indicates their limited discrimination. (2) There is no consistent winner on all six datasets (see detailed result in our online appendix [9]), an observation also made by [69] for the related truth discovery problem. (3) All other models except ABH with poor performance tend to have more similar accuracy than HND while HND tops them by far on 2 of the 6 datasets, which shows the novelty of HND and its usefulness on data of different distributions.



(a) Variance of eigenvector used by HND or ABH



(b) Normalized user displacement



(c) Accuracy of user ranking

Fig. 6: Section IV-D: Stability experiments: (a) The variance of the eigenvector used by HND is smaller than that used by ABH, which makes it more robust to perturbations from the ideal C1P case. This leads to HND having lower difference in the user ranking (b) and higher accuracy (c).

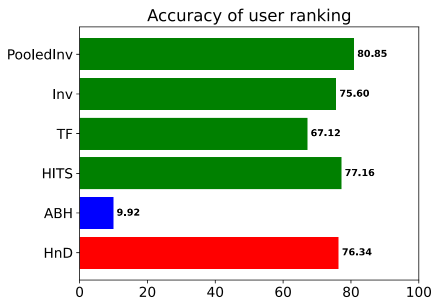


Fig. 7: Section IV-E: Correlation of user ranking on real-world datasets with the "True-answer" baseline that serves us as approximate gold standard user ranking. Notice that the true ranking of users by ability is not known.

V. Additional Related Work

In this section, we discuss additional approaches for the truth discovery problem. This is in addition to existing C1P reconstruction algorithms discussed in Section II-C and HITS-based truth discovery approaches discussed in Section III-A.

Spectral approaches. Dalvi et al. [11] proposed two methods that output the user abilities and item labels with the help of eigenvector computation. Ghosh et al. [22] proposed a method that only outputs the item labels, which involves calculating the first eigenvector of a symmetric matrix. Both approaches work only for binary problems and are not obvious to generalize for $k > 2$ options.

Other truth discovery approaches. [33] proposes the concept of experts and utilizes the observation that experts are more likely to reach consensus on a set of single questions (called hyper-questions in the paper) to conduct majority vote on hyper-questions instead of single questions but cannot quantify user abilities nor rank them. [37] relies on embeddings that cannot be easily converted into a ranking on users. [34] uses confidence and focus on long-tail data. [12], [26] are optimization-based methods that only consider homogeneous questions (recall Section II-A).

Other truth discovery problems. Many approaches have been proposed for truth discovery. Most have different setups and are not applicable to our problem. [55], [63] change the setup of the problem by assigning different tasks to two groups of workers, where the first group answers the questions and the second group evaluates the answers. [15], [47], [67], [70] work on the problem of how to assign questions to only a subset of the sources. [13], [14] pay attention to the sources of information, yet focus on the copying relationships between sources. In our setup, no information is copied between users. [48] discusses the problem of how much training data is needed to gain high-quality models. [42] studies truth discovery in quantitative applications, such as percentage annotation and object counting.

Crowdsourcing. The ability discovery problem is closely connected to the truth discovery problem which occur in a wide range of data management problems related to crowdsourcing [32], [69]. Various crowdsourcing systems have been proposed [16], [68], and the crowdsourcing approach has been refined for various tasks, such as query answering [19], entity resolution [7], annotating Twitter data [18], top-k algorithms [66] and various other labeling tasks [24], [56], [59].

Expert finding. The expert finding problem [35], [65] also aims to assess the trustworthiness of users. The difference is that it focuses on *finding experts with expertise (skills) specific to a given question* while our ability discovery problem aims to assess an overall user ability.

VI. Conclusions

We proposed HITSNDIFFS, a novel variant of HITS, with surprising theoretical and practical properties for ability discovery. On the theoretical side, we showed that 1) C1P of the response matrix models consistent solutions for the problem; 2) our method reconstructs the correct user rankings in the consistent case; 3) does so in linear time and 4) can handle more general cases (in contrast to other linear discrete algorithms). On the practical side, we showed that HND handles the problem of ability discovery with robust accuracy and greater scalability in terms of the number of users than the only existing C1P reconstruction algorithm that works for general cases.

Acknowledgment

Work supported in part by the National Science Foundation (NSF) under award numbers IIS-1762268 and IIS-1956096.

References

[1] "Amazon mechanical turk," https://www.mturk.com/, 2023.

[2] "Piazza," https://piazza.com/, 2023.

[3] W. E. Arnoldi, "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Quarterly of applied mathematics*, vol. 9, no. 1, pp. 17–29, 1951. [Online]. Available: http://www.jstor.org/stable/43633863

[4] J. E. Atkins, E. G. Boman, and B. Hendrickson, "A spectral algorithm for seriation and the consecutive ones problem," *SIAM Journal on Computing*, vol. 28, no. 1, pp. 297–310, 1998. [Online]. Available: https://doi.org/10.1137/S0097539795285771

[5] R. D. Bock, "Estimating item parameters and latent ability when responses are scored in two or more nominal categories," *Psychometrika*, vol. 37, no. 1, pp. 29–51, 1972. [Online]. Available: https://doi.org/10.1007/BF02291411

[6] K. S. Booth and G. S. Lueker, "Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms," *Journal of Computer and System Sciences*, vol. 13, no. 3, pp. 335–379, 1976. [Online]. Available: https://doi.org/10.1016/S0022-0000(76)80045-1

[7] C. Chai, G. Li, J. Li, D. Deng, and J. Feng, "Cost-effective crowdsourced entity resolution: A partial-order approach," in *SIGMOD*, 2016, pp. 969–984. [Online]. Available: https://doi.org/10.1145/2882903.2915252

[8] Z. Chen, S. Mitra, R. Ravi, and W. Gatterbauer, "HITSnDIFFS: From truth discovery to ability discovery by recovering matrices with the consecutive ones property: Code and experiments," 2023. [Online]. Available: https://github.com/northeastern-datalab/HITSnDIFFs/

[9] ——. (2023) HITSnDIFFS: From truth discovery to ability discovery by recovering matrices with the consecutive ones property (extended version). [Online]. Available: https://arxiv.org/abs/2401.00013

[10] J. K. Cullum and R. A. Willoughby, *Lanczos algorithms for large symmetric eigenvalue computations: Vol. I: Theory*. SIAM, 2002. [Online]. Available: https://doi.org/10.1137/1.9780898719192

[11] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi, "Aggregating crowdsourced binary ratings," in *WWW*, 2013, pp. 285–294. [Online]. Available: https://doi.org/10.1145/2488388.2488414

[12] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Applied statistics*, pp. 20–28, 1979. [Online]. Available: https://doi.org/10.2307/2346806

[13] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava, "Global detection of complex copying relationships between sources," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1358–1369, 2010. [Online]. Available: https://doi.org/10.14778/1920841.1921008

[14] X. L. Dong, L. Berti-Equille, and D. Srivastava, "Integrating conflicting data: the role of source dependence," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 550–561, 2009. [Online]. Available: https://doi.org/10.14778/1687627.1687690

[15] X. L. Dong, B. Saha, and D. Srivastava, "Less is more: Selecting sources wisely for integration," *Proceedings of the VLDB Endowment*, vol. 6, no. 2, pp. 37–48, 2012. [Online]. Available: https://doi.org/10.14778/2535568.2448938

[16] J. Fan, G. Li, B. C. Ooi, K. Tan, and J. Feng, "icrowd: An adaptive crowdsourcing framework," in *SIGMOD*, 2015, pp. 1015–1030. [Online]. Available: https://doi.org/10.1145/2723372.2750550

[17] M. Fiedler, "Laplacian of graphs and algebraic connectivity," *Banach Center Publications*, vol. 25, no. 1, pp. 57–70, 1989. [Online]. Available: https://doi.org/10.4064/-25-1-57-70

[18] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze, "Annotating named entities in twitter data with crowdsourcing," in *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010, pp. 80–88. [Online]. Available: https://aclanthology.org/W10-0713/

[19] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin, "CrowdDB: answering queries with crowdsourcing," in *SIGMOD*, 2011, pp. 61–72. [Online]. Available: https://doi.org/10.1145/1989323.1989331

[20] G. Frobenius, "Über matrizen aus nicht negativen elementen," *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften*, pp. 456–477, 1912. [Online]. Available: https://commons.wikimedia.org/wiki/File:Ueber_Matrizen_aus_nicht_negativen_Elementen.pdf

[21] D. R. Fulkerson and O. A. Gross, "Incidence matrices and interval graphs." *Pacific Journal of Mathematics*, vol. 15, no. 3, pp. 835 – 855, 1965. [Online]. Available: http://dx.doi.org/10.2140/pjm.1965.15.835

[22] A. Ghosh, S. Kale, and P. McAfee, "Who moderates the moderators?: crowdsourcing abuse detection in user-generated content," in *EC*, 2011, pp. 167–176. [Online]. Available: https://doi.org/10.1145/1993574.1993599

[23] J. Hauschild and F. Pollmann, "Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy)," *SciPost Phys. Lect. Notes*, p. 5, 2018, code available from https://github.com/tenpy/tenpy. [Online]. Available: https://scipost.org/10.21468/SciPostPhysLectNotes.5

[24] H. Hu, Y. Zheng, Z. Bao, G. Li, J. Feng, and R. Cheng, "Crowdsourced POI labelling: Location-aware result inference and task assignment," in *ICDE*, 2016, pp. 61–72. [Online]. Available: https://doi.org/10.1109/ICDE.2016.7498229

[25] Y. Hu, J. Scott, and H. MC73, "A fast multilevel fiedler and profile reduction code," in *RAL-TR-2003-36*, 2003.

[26] H. Kajino, Y. Tsuboi, and H. Kashima, "A convex formulation for learning from crowds," in *AAAI*, 2012, pp. 73–79. [Online]. Available: https://doi.org/10.1609/aaai.v26i1.8105

[27] D. Kendall, "Incidence matrices, interval graphs and seriation in archeology," *Pacific Journal of mathematics*, vol. 28, no. 3, pp. 565–570, 1969. [Online]. Available: https://msp.org/pjm/1969/28-3/p08.xhtml

[28] N. M. Kingston and N. J. Dorans, "The feasibility of using item response theory as a psychometric model for the GRE aptitude test," *ETS Research Report Series*, vol. 1982, no. 1, 1982. [Online]. Available: http://dx.doi.org/10.1002/j.2333-8504.1982.tb01298.x

[29] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999. [Online]. Available: https://doi.org/10.1145/324133.324140

[30] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," 1950. [Online]. Available: https://doi.org/DOI:10.6028/JRES.045.026

[31] D. Lay, S. Lay, and J. McDonald, *Linear Algebra and Its Applications*. Pearson, 2016. [Online]. Available: https://books.google.com/books?id=L8SUoAEACAAJ

[32] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," in *ICDE*, 2017, pp. 39–40. [Online]. Available: https://doi.org/10.1109/ICDE.2017.26

[33] J. Li, Y. Baba, and H. Kashima, "Hyper questions: Unsupervised targeting of a few experts in crowdsourcing," in *CIKM*, 2017, pp. 1069–1078. [Online]. Available: https://doi.org/10.1145/3132847.3132971

[34] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han, "A confidence-aware approach for truth discovery on long-tail data," *Proceedings of the VLDB Endowment*, vol. 8, no. 4, pp. 425–436, 2014. [Online]. Available: https://doi.org/10.14778/2735496.2735505

[35] S. Lin, W. Hong, D. Wang, and T. Li, "A survey on expert finding techniques," *J. Intell. Inf. Syst.*, vol. 49, no. 2, pp. 255–279, 2017. [Online]. Available: https://doi.org/10.1007/s10844-016-0440-5

[36] F. M. Lord, M. R. Novick, and A. Birnbaum, *Statistical Theories of Mental Test Scores*. Addison-Wesley, 1968. [Online]. Available: https://psycnet.apa.org/record/1968-35040-000

[37] S. Lyu, W. Ouyang, H. Shen, and X. Cheng, "Truth discovery by claim and source embedding," in *CIKM*. ACM, 2017, pp. 2183–2186. [Online]. Available: https://doi.org/10.1145/3132847.3133069

[38] L. W. Mackey, "Deflation methods for sparse PCA," in *NIPS*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., 2008, pp. 1017–1024. [Online]. Available: https://proceedings.neurips.cc/paper/2008/hash/85d8ce590ad8981ca2c8286f79f59954-Abstract.html

[39] C. D. Meyer, Ed., *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, 2000. [Online]. Available: https://doi.org/10.1137/1.9781611977448

[40] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022, p. 255. [Online]. Available: http://probml.github.io/book1

[41] M. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc., 2010. [Online]. Available: https://doi.org/10.1093/acprof:oso/9780199206650.001.0001

[42] R. W. Ouyang, L. M. Kaplan, A. Toniolo, M. Srivastava, and T. J. Norman, "Aggregating crowdsourced quantitative claims: Additive and multiplicative models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1621–1634, 2016. [Online]. Available: https://doi.org/10.1109/TKDE.2016.2535383

[43] B. N. Parlett and D. S. Scott, "The lanczos algorithm with selective orthogonalization," *Mathematics of Computation*, vol. 33, no. 145, pp. 217–238, 1979. [Online]. Available: https://doi.org/10.2307/2006037

[44] J. Pasternack and D. Roth, "Knowing what to believe (when you already know something)," in *COLING*, C. Huang and D. Jurafsky, Eds., 2010, pp. 877–885. [Online]. Available: https://aclanthology.org/C10-1099/

[45] O. Perron, "Zur Theorie der Matrices," *Mathematische Annalen*, vol. 64, no. 2, pp. 248–263, 1907. [Online]. Available: https://doi.org/10.1007/BF01449896

[46] M.-T. Puth, M. Neuhaeuser, and G. D. Ruxton, "Effective use of spearman's and kendall's correlation coefficients for association between two measured traits," *Anim. Behav.*, vol. 102, pp. 77–84, April 2015. [Online]. Available: https://doi.org/10.1016/j.anbehav.2015.01.010

[47] T. Rekatsinas, X. L. Dong, and D. Srivastava, "Characterizing and selecting fresh data sources," in *SIGMOD*, 2014, pp. 919–930. [Online]. Available: https://doi.org/10.1145/2588555.2610504

[48] T. Rekatsinas, M. Joglekar, H. Garcia-Molina, A. G. Parameswaran, and C. Ré, "Slimfast: Guaranteed results for data fusion and source reliability," in *SIGMOD*, 2017, pp. 1399–1414. [Online]. Available: https://doi.org/10.1145/3035918.3035951

[49] F. Samejima, "A new family of models for the multiple-choice item." Tennessee Univ Knoxville Dept of Psychology, Tech. Rep., 1979. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA080350

[50] ——, "Graded response model," in *Handbook of modern item response theory*. Springer, 1997, pp. 85–100. [Online]. Available: https://doi.org/10.1007/978-1-4757-2691-6_5

[51] R. Sanchez, "Girth: Item Response Theory in Python." [Online]. Available: https://github.com/eribean/girth

[52] A. Sheshadri and M. Lease, "Square: A benchmark for research on computing crowd consensus," in *First AAAI conference on human computation and crowdsourcing*, 2013. [Online]. Available: https://ojs.aaai.org/index.php/HCOMP/article/view/13088

[53] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904. [Online]. Available: https://doi.org/10.2307/1422689

[54] G. W. Stewart and J. guang Sun, *Matrix perturbation theory*. Academic Press, 1990. [Online]. Available: https://www.worldcat.org/title/matrix-perturbation-theory/oclc/908946968

[55] T. Sunahase, Y. Baba, and H. Kashima, "Pairwise HITS: Quality estimation from pairwise comparisons in creator-evaluator crowdsourcing process," in *AAAI*, vol. 31, no. 1, 2017, pp. 977–984. [Online]. Available: https://doi.org/10.1609/aaai.v31i1.10634

[56] A. Tarasov, S. J. Delany, and C. Cullen, "Using crowdsourcing for labelling emotional speech assets," *W3C EmotionML Workshop*, 2010. [Online]. Available: https://www.w3.org/2010/10/emotionml/papers/tarasov.pdf

[57] W. J. Van Der Linden and R. K. Hambleton, "Item response theory: Brief history, common models, and extensions," in *Handbook of modern item response theory*. Springer, 1997, pp. 1–28. [Online]. Available: https://doi.org/10.1007/978-1-4757-2691-6_1

[58] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. [Online]. Available: https://doi.org/10.1038/s41592-019-0686-2

[59] P. Welinder and P. Perona, "Online crowdsourcing: rating annotators and obtaining cost-effective labels," in *CVPRW*, 2010, pp. 25–32. [Online]. Available: https://doi.org/10.1109/CVPRW.2010.5543189

[60] P. A. White, "The computation of eigenvalues and eigenvectors of a matrix," *Journal of the Society for Industrial and Applied Mathematics*, vol. 6, no. 4, pp. 393–437, 1958. [Online]. Available: http://www.jstor.org/stable/2098714

[61] ——, "Hotelling's matrix deflation," *Journal of the Society for Industrial and Applied Mathematics*, vol. 6, no. 4, pp. 414–415, 1958. [Online]. Available: http://www.jstor.org/stable/2098714

[62] ——, "Wilkinson's vector annihilation," *Journal of the Society for Industrial and Applied Mathematics*, vol. 6, no. 4, p. 414, 1958. [Online]. Available: http://www.jstor.org/stable/2098714

[63] J. Yang, J. Fan, Z. Wei, G. Li, T. Liu, and X. Du, "A game-based framework for crowdsourced data labeling," *The VLDB Journal*, vol. 29, no. 6, pp. 1311–1336, 2020. [Online]. Available: https://doi.org/10.1007/s00778-020-00613-w

[64] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 6, pp. 796–808, 2008. [Online]. Available: https://doi.org/10.1109/TKDE.2007.190745

[65] S. Yuan, Y. Zhang, J. Tang, W. Hall, and J. B. Cabotà, "Expert finding in community question answering: a review," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 843–874, 2020. [Online]. Available: https://doi.org/10.1007/s10462-018-09680-6

[66] X. Zhang, G. Li, and J. Feng, "Crowdsourced top-k algorithms: An experimental evaluation," *PVLDB*, vol. 9, no. 8, pp. 612–623, 2016. [Online]. Available: http://www.vldb.org/pvldb/vol9/p612-zhang.pdf

[67] Y. Zheng, R. Cheng, S. Maniu, and L. Mo, "On optimality of jury selection in crowdsourcing," in *EDBT*. OpenProceedings.org, 2015, pp. 193–204. [Online]. Available: https://doi.org/10.5441/002/edbt.2015.18

[68] Y. Zheng, G. Li, and R. Cheng, "DOCS: domain-aware crowdsourcing system," *PVLDB*, vol. 10, no. 4, pp. 361–372, 2016. [Online]. Available: http://www.vldb.org/pvldb/vol10/p361-zheng.pdf

[69] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?" *PVLDB*, vol. 10, no. 5, pp. 541–552, 2017. [Online]. Available: https://doi.org/10.14778/3055540.3055547

[70] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng, "QASCA: A quality-aware task assignment system for crowdsourcing applications," in *SIGMOD*, 2015, pp. 1031–1046. [Online]. Available: https://doi.org/10.1145/2723372.2749430