Abstract

Truth discovery is a general name for a broad range of statistical methods aimed to extract the correct answers to questions, based on multiple answers coming from noisy sources. For example, workers in a crowdsourcing platform. In this paper, we consider an extremely simple heuristic for estimating workers’ competence using average proximity to other workers. We prove that this estimates well the actual competence level and enables separating high and low quality workers in a wide spectrum of domains and statistical models. If we further assume conditional independence of workers, there is a linear or approximately linear relation between workers’ quality and their average similarity. This applies in particular when labels are ordinal rankings.

Finally, weighing workers according to their average proximity in a crowdsourcing setting, results in substantial improvement over unweighted aggregation and other truth discovery algorithms in practice.

We provide a result that may be of independent interest, showing that the optimal way to aggregate rankings under Condorcet noise model with known heterogeneous competence is a weighted Kemeny voting rule with specific weights.

“All happy families are alike; each unhappy family is unhappy in its own way.”

— Leo Tolstoy, Anna Karenina

1 Introduction

Consider a standard crowdsourcing task such as identifying which images contain a person or a car [10], or identifying the location in which pictures were taken [26]. Such tasks are also used to construct large datasets that can later be used to train and test machine learning algorithms. Crowdsourcing workers are usually not experts, thus answers obtained this way often contain many mistakes [33, 34], and multiple answers are aggregated to improve accuracy.

From a theory/statistics perspective, “truth discovery” is a general name for a broad range of methods that aim to extract some underlying ground truth from noisy answers. While the mathematics of truth discovery dates back to the early days of statistics, at least to the Condorcet Jury Theorem [7], the rise of crowdsourcing platforms suggests an exciting modern application of aggregating complex labels from varied domains such as image processing and natural language, to healthcare. For example, the Etch-a-Cell project uses volunteers to trace the boundary of tumors on Electron Microscopy images [31].

Yet, the vast majority of the theoretical literature on truth discovery follows Condorcet by focusing on binary, multi-label or sometimes real-valued questions (see Related Work section), while specific applications with complex labels often rely on specialized algorithms.

Many of these algorithms aim to identify first the most competent workers. While some of them employ highly sophisticated analysis, others are much more direct: for example, Kobayashi [18] suggests a ‘frustratingly easy’ algorithm that ranks workers by their average cosine similarity to
others in a text summarization task; and Kurvers et al. [19] prove that the Hamming distance of a worker from others is correlated with her competence in answering yes/no questions. Of course, using average similarity or distance is not a new idea, and is extensively employed outside the context of aggregation, for example in Games with a Purpose [32, 15] to identify outliers, and in peer prediction to incentivize effort [35].

In this paper we argue that average similarity is a powerful tool, with nothing special about Cosine or Hamming similarity in particular. Our main observation can be written as follows:

**Theorem** (Anna Karenina principle, informal). The expected average similarity of each worker to all others, is roughly linearly increasing in her competence.

Essentially, the theorem says that as in Tolstoy’s novel, “good workers are all alike,” whereas “each bad worker is bad in her own way” and thus not similar to other workers.

**Contribution and paper structure**

After the preliminary definitions in Section 2, we prove a formal version of the Anna Karenina principle and show how it can be used to identify poor workers in Section 3 without assuming specific label structure. We show how additional assumptions lead to tighter corollaries of exactly or approximately linear relation between pairwise similarity and competence. To the best of our knowledge these are the first formal guarantees on general-domain truth discovery.

In Section 4 we explain how to leverage the Anna Karenina principle for aggregation using a simple algorithm (P-TD). In addition, we prove that if the competence of all workers is known, then the optimality of the Kemeny-Young voting rule under the Condorcet noise model can be extended to heterogeneous voters (with appropriate weights).

We demonstrate on real and synthetic data, that P-TD substantially improves aggregation accuracy, competing well with advanced and domain-specific algorithms.

Most proofs, as well as additional empirical results are available in the full version of the paper: https://tinyurl.com/2p923tzv.

The full version of the AAAI23 paper is available on arXiv:

**Related work**

The Condorcet Jury Theorem [7] was perhaps the first formal treatment of truth discovery, and extensions to experts with heterogeneous competence levels were surveyed by Grofman et al. [14]. The idea of estimating workers’ competence in order to improve aggregation is thus underlying many of the algorithms in the area (a recent survey is in [23]). We should note that self-reporting of accuracy often leads to poor results [12, 29].

**Average similarity** We have mentioned in the introduction the two applications of average similarity to truth discovery that we are aware of. Both of them assume a specific label structure and (somewhat surprisingly) both are quite recent: Kobayashi [18] proved that cosine similarity approximates a known kernel density estimator. Kurvers et al. [19] focused on binary questions with independent errors, showing both theoretically and empirically that the expected average Hamming proximity correlates with the true competence, albeit without comparing to any other algorithm.

Our Anna Karenina theorem entails the Kurvers et al. result as a special case, and provides explicit performance guarantees for the heuristic suggested by Kobayashi.
**Domain-specific algorithms** Many truth-discovery algorithms have been proposed for specific label structures, mostly for categorical (multiple-choice) and real-valued labels. Often these algorithms entwine accuracy and ground truth estimation, by iteratively aggregating labels to obtain an estimate of the ground truth, and using that in turn to estimate workers competence. This approach was pioneered by the EM-style Dawid-Skene estimator [9], with many follow-ups [16, 13, 2, 37, 41, 22].

Another class of algorithms uses spectral methods to infer the competence and/or other latent variables from the covariance matrix of the workers [28, 40], or from their pairwise Hamming similarity [20]. Note that covariance can also be thought of as a measure worker similarity in the context of binary labels. In rank aggregation, every voting rule can be considered as a truth-discovery algorithm [25, 5].

Some of these works also provide formal convergence guarantees and/or bounds on the error that are subject to assumptions on the distribution of answers.

**General labels** When there are complex labels that are not numbers or categories, but for example contain text, graphics and/or hierarchical structure, there may not be a natural way to aggregate them but we would still want to evaluate workers’ competence.

Two recent papers suggest to use the pairwise distance (or similarity) matrix as a general domain-independent abstraction, then applying sophisticated algorithms on this matrix: The *multidimensional annotation scaling* (MAS) model [4] extends the Dawid-Skene model by calculating the labels and competence levels that would maximize the likelihood of the observed distance matrix, using the Stan probabilistic programming language; Another approach is to find a ‘core’ of good workers [17], by looking for a dense subgraph of the similarity matrix.

While we adopt the approach that pairwise similarity is the right domain-independent abstraction for general labels, we argue that usually there is no need for such complex algorithms: a ‘frustratingly easy’ average is sufficient.

**Rank aggregation** The idea that voting rules can be used as a statistical tool to extract the most likely truth goes back at least to Condorcet: He suggested the model where every voter/expert swaps each pair of alternatives with a fixed independent probability [7]. The optimal solution to this particular problem is given by the Kemeny voting rule, as shown by Young [39]. Followup work considered other noise models [11, 36].

More recently, Procaccia et al. considered voting rules as a possible solution to adversarial noise [30].

In all these works, voters are assumed to be equally competent a-priori. However almost every voting rule has a natural weighted variation (e.g. by duplicating voters with high competence), so a good method to estimate competence is likely to boost the performance of any such voting rule.

### 2 Preliminaries

We consider a set $N$ of $n$ workers, each providing a report in some space $Z$. We denote elements of $Z$ (typically $m$-length vectors, see below) in bold. Thus, an instance of a truth discovery is a pair $\langle S = (s_i)_{i\in N}, z \rangle$, where $s_i \in Z$ is the report of worker $i$, and $z \in Z$ is the ground truth. $S$ is also called a dataset.\footnote{It is ok if $s_i$ is a partial vector, as long as there is enough intersection between pairs of workers.}

**Distance measures** We assume there is a distance measure (not necessarily a metric) $d : Z \times Z \to \mathbb{R}_+$ over pairs of labels.
Distance measures can often be derived from an inner product. Formally, consider an arbitrary symmetric inner product space \((Z, \langle \cdot, \cdot \rangle)\). This induces a norm \(\|x\|^2 := \langle x, x \rangle\) and a distance measure \(d(x, y) := \|x - y\|^2\) (not necessarily a metric). A special case of interest is the normalized Euclidean product on \(Z = \mathbb{R}^m\), defined as \(\langle x, y \rangle_E := \frac{1}{m} \sum_{j \leq m} x_j y_j\); and the corresponding normalized squared Euclidean distance (NSED), a natural way to capture the dissimilarity of two items [6].

**Rankings** Let \(\mathcal{L}(C)\) be the set of all permutations over a set of candidates \(C\). Any ranking (permutation) \(L \in \mathcal{L}(C)\) corresponds to a binary vector \(s \in \{0, 1\}^m\) where \(m = |C|\) (i.e., each dimension is a pair of candidates), which we denote by \(s_L \in \{0, 1\}^m\). If there is \(L \in \mathcal{L}(C)\) s.t. \(s_L = s\) then we denote it \(L_s\). In particular we denote by \(L_z\) the ground truth order over \(C\).

A natural metric over rankings is the [normalized] Kendall-tau distance (a.k.a swap distance), that coincides with the [normalized] Hamming distance on the binary representation. That is, \(d_{KT}(L, L') = d_{Hamming}(s_L, s_{L'})\). Hamming distance, in turn, is a special case of NSED.

**Noise model** We do not make any assumptions regarding the ground truth \(z\). The type \(t_i\) of a worker determines her distribution of answers. A dataset is constructed in two steps:

1. Sample a finite population of workers i.i.d from a distribution \(T\) (called a proto-population) over a set of types \(T\). For our running example, suppose that \(Z\) is the set of permutations over 5 alternatives \(L(\{a,b,c,d,e\})\), \(T\) is uniform over \([0.3,0.8]\), \(n = 5\) and sampled types are \(t = (0.35,0.4,0.4,0.5,0.75)\), where lower is more accurate.

2. Workers each report their answers \(S\), which depend on the ground truth \(z\) (the identity permutation in our example), on their types, and on a random factor. We next describe an example where labels are rankings. See a different example with real-valued answers and Gaussian noise in full version.

Formally, a noise model is a function \(\mathcal{Y} : Z \times T \to \Delta(Z)\). That is, the report of worker \(i\) is a random variable \(s_i\) sampled from \(\mathcal{Y}(z, t_i)\). We note that \(T, \mathcal{Y}\) and \(z\) together induce a distribution \(\mathcal{Y}(z, T)\) over answers (and thus over datasets), where \(s \sim \mathcal{Y}(z, T)\) means we first sample a type \(t \sim T\) and then a report \(s \sim \mathcal{Y}(z, t)\).

The data in our example (Table 1) was sampled from the noise model \(\mathcal{Y}\) that is a Mallows distribution centered on a ranking \(L_z\) with individual parameter \(\phi_i := t_i\). This means worker \(i\) reports every order \(L_s = L_{s_i} \in \mathcal{L}\) with probability proportional to \(\phi_i^{-d(L_z, L_s)}\), where \(d(\cdot, \cdot)\) is the Kendall-tau distance. Thus for low \(\phi_i\) the worker is very likely to return the true permutation with few errors, and for \(\phi_i\) close to 1, the distribution is almost uniform.

**Workers’ competence** Competent workers are close to the truth, in expectation. More formally, given some ground truth \(z\) and a distance measure \(d\), we define the fault (or incompetence) of a worker \(i\) as her expected distance from the ground truth, denoted \(f_i(z) := E_{s_i \sim \mathcal{Y}(z, t_i)}[d(s_i, z)]\).

In the case of Mallows distribution with parameter \(\phi_i\), it is known that it is equivalent to Condorcet Noise model, which swaps every pair of items with probability \(p_t\), where \(\phi_i = \frac{e^{p_t}}{1 + e^{p_t}}\), see [24].

We denote the mean fault by \(\mu_T(z) := E_{s \sim \mathcal{Y}(z, T)}[d(s, z)]\), omitting \(T\) and/or \(z\) when clear from the context.

**Aggregation** Given an instance \((S, z)\), an aggregation function returns predicted labels \(\hat{z}\). We define the error as \(d(z, \hat{z})\). Unweighted aggregation is usually trivial in binary domain (Majority) and in real valued domains (mean or median). However in the ranking domain there are countless
as the mean bias of the proto-population. E.g. in Euclidean space that the expectation of distance measures) on Fig. 1.

Theorem 1 (Anna Karenina Principle)

and weigh them accordingly, so that the result under any voting rule would improve. Most common rules also have natural weighted extensions.

We do not aim to find voting rules that are better than others, but to estimate workers’ competence and weigh them accordingly, so that the result under any voting rule would improve.

3 Fault Estimation

Our key approach is relying on estimating \( f_i \) using the average distance of worker \( i \) from all other workers. Formally, we define \( d_{i,i'} := d(s_i, s_{i'}) \), and the average pairwise distance is

\[
\pi_i := \frac{1}{n-1} \sum_{i' \in N \setminus \{i\}} d_{i,i'}.
\]

Next, we analyze the relation between \( \pi_i = \pi_i(S) \) (which is a random variable) and \( f_i \), which is an inherent property that is deterministically induced by the worker’s type. For an element \( s \in Z \) we consider the induced noise variable \( \epsilon_s := s - z \).\(^3\) We denote by \( \tilde{Y}(z, t) \) the distribution of \( \epsilon_s \) (where \( s \sim \tilde{Y}(z, t) \)). Thus under NSED we have that \( d(s, z) = ||\epsilon_s||^2 \).

We define \( b_i(z) := E_{s \sim \tilde{Y}(z, t)}[\epsilon_i] \) as the bias of a type \( i \) worker, and \( b_T(z) := E_{s \sim \tilde{Y}(z, T)}[\epsilon] \) as the mean bias of the proto-population. E.g. in Euclidean space \( b_i(z) \) is a vector where \( b_{ij}(z) > 0 \) if \( i \) tends to overestimate the answer of question \( j \), and negative values mean underestimation.

Our main conceptual result is an approximately linear connection between the expectations of \( \pi_i \) and \( d(s_i, z) \).

Theorem 1 (Anna Karenina Principle).

\[
E_{S \sim \tilde{Y}(z, T)}[\pi_i | t_i, z] = f_i(z) + \mu_T(z) - 2 \langle b_i(z), b_T(z) \rangle.
\]

We can see this linear relation in both synthetic and real datasets (with different labels and distance measures) on Fig. 1.

The proof is rather straight-forward, and is deferred to . In particular it shows by direct computation that the expectation of \( d_{i,i'} \) for every pair of workers is

\[
E[d_{i,i'} | t_i, t_{i'}, z] = f_i(z) + f_{i'}(z) - 2 \langle b_i(z), b_{i'}(z) \rangle.
\]

\(^3\)When labels are rankings, \( s, z \) etc. are their representations as binary vectors.
Figure 1: Each scatterplot presents all workers in a single instance, with their disparity $\pi_i$ vs. their real error $d(s_i, z)$.

Despite (or perhaps because of) its simplicity, the principle above is highly useful for estimating workers’ competence. If $\pi_i$ is roughly linearly increasing in $f_i$, a naïve approach to estimate $f_i$ from the data is by setting $\hat{f}_i$ to be some increasing function of $\pi_i$.

However there are several obstacles we need to overcome in order to get theoretical guarantees. In particular: concentration bounds; estimation of $\mu$; and the biases that appear in the last term; all of which will be tackled in the next sections.

Concentration bounds  How far is the empirical average $\pi_i$ from its expectation? We show that when the noise on all questions is independent and bounded, the probability of a large estimation error decreases linearly with the sample size $\min\{n, m\}$. 

Noise-independent bounds  What can be said without any assumptions on the distribution of labels? We argue that we can at least tell particularly poor workers from good workers.

Here we show a bound for binary labels (that also applies to rankings). In the AAAI23 version we provide a similar result for real-valued labels.

**Corollary 2** (Anna Karenina principle for binary labels and rankings). Assume labels are binary vectors with Hamming distance, or rankings with Kendall-tau distance. For any worker $i$, $E[\pi_i|t_i] \in [f_i - \mu, f_i + \mu]$.

**Proof.** By Theorem 1, $E[\pi_i|t_i] = f_i + \mu_T - 2 \langle b_i, b_T \rangle$. When labels are binary vectors, $\langle b_i, b_T \rangle$ simply counts in how many entries $j \leq m$ both $b_{ij} = s_{ij} - z_j$ and $b_{i'j} = s_{i'j} - z_j$ are nonzero, and thus $\langle b_i, b_{i'} \rangle = \min\{f_i, f_{i'}\}$, and

$$\langle b_i, b_T \rangle = E_T[\langle b_i, b_{i'} \rangle] = E_T[\min\{f_i, f_{i'}\}] \leq \min\{f_i, E[f_{i'}]\} = \min\{f_i, \mu_T\}.$$

The upper bound is since $\langle b_i, b_T \rangle \geq 0$, and the lower bound is since

$$E[\pi_i|t_i] = f_i + \mu_T - 2 \langle b_i, b_T \rangle \geq f_i + \mu_T - 2\mu = f_i - \mu,$$

as required. □
In particular, this means that given enough samples to accurately estimate \( E[\pi] \), we can always distinguish good workers with \( f_\star \approx \mu \) from bad workers with \( f_{\star\star} \approx 3\mu \), as \( E[\pi_\star] < 2\mu < E[\pi_{\star\star}] \).

Without further assumptions, this condition is tight (i.e. it may not be possible to separate between workers whose faults are \( \mu \) and \( 3\mu \)). Consider the following types, for \(|C| = 3|\):

<table>
<thead>
<tr>
<th>Meaning</th>
<th>notation</th>
<th>ranking</th>
<th>as binary vector</th>
<th>error ( f_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truth</td>
<td>( z )</td>
<td>abc</td>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>Poor worker</td>
<td>( s_\star )</td>
<td>bac</td>
<td>100</td>
<td>1/3</td>
</tr>
<tr>
<td>Good worker</td>
<td>( s_{\star\star} )</td>
<td>eba</td>
<td>111</td>
<td>3/3</td>
</tr>
<tr>
<td>Common type</td>
<td>( s' )</td>
<td>acb</td>
<td>001</td>
<td>1/3</td>
</tr>
</tbody>
</table>

If the mass of the common type is \( \approx 1 \), then \( \mu \approx f' = 1 \). This means \( f_\star \approx \mu \) and \( f_{\star\star} \approx 3\mu \). Yet \( \pi_\star \approx d(s_\star, s') = 2/3 = d(s_{\star\star}, s') \approx \pi_{\star\star} \) (in fact \( \pi_\star = \pi_{\star\star} \) exactly), so the good and poor workers cannot be distinguished. This example shows that in order to identify the poor workers, we need a reasonable amount of good workers.

**Arbitrary Labels and Symmetric Noise**

A trivial implication of Theorem 1 is when the average worker is unbiased:

**Corollary 3** (Anna Karenina principle for zero bias). If \( b_T = 0 \) then \( E[\pi_i | t_i] = f_i + \mu_T \) for all \( i \).

This means that given enough samples, we can retrieve workers’ exact fault level with high accuracy, by setting \( f_i := \pi_i(S) - \mu \). This will be important later on when we discuss aggregation.

What if we have arbitrary labels, and use other distance measures than NSED? Suppose that \( d \) is an arbitrary distance metric over space \( Z \), \( z \in Z \) is the ground truth, and \( s_i \in Z \) is the report of worker \( i \). \( f_i \) and \( \pi_i \) are defined as before. Intuitively, we say that the noise model \( \mathcal{Y} \) is symmetric if for every point \( x \) there is an equally-likely point that is on “the other side” of \( z \) (note that this in particular implies zero bias).

**Theorem 4** (Anna Karenina principle for symmetric noise and distance metrics). If \( d \) is any distance metric and \( \mathcal{Y} \) is symmetric, then \( \max \{ \mu, f_i \} \leq E[\pi_i | t_i] \leq \mu + f_i \).

An immediate corollary of Theorem 4 is that for poor workers with \( f_i \geq \mu \), the average distance \( \pi_i \) is a 2-approximation for \( f_i \) (up to noise). See details and proof in the full version.

**Domain-specific results**

**Real-valued labels and Gaussian noise** One of the most common noise models for real-valued labels is that each worker adds independent Gaussian noise to each label, with mean zero and variance that reflects her fault \( f_i \). This model is known as Additive White Gaussian noise. When workers’ variance is known, the optimal aggregation is a weighted average, where individual weights are inversely proportional to the variance [1].

An intriguing property of our suggested average similarity method, is that under a slight variation of AWG, it extracts the maximum likelihood estimator for \( f_i \), with a particular regularization factor. This is the main technical result of the AAAI’23 version, and is omitted from this version.

**Binary labels** Kurvers et al. [19] considered the average similarity of workers when answering a set of yes/no questions, and the type of a worker is her probability \( p_i \) to answer correctly independently over each question, a model known as the one-coin model or the Dawid-Skene model.

They showed that the (expected) average similarity is an increasing linear function of \( p_i \).
Interestingly, the result from [19] can also be obtained directly from Theorem 1, by plugging in the Hamming distance (which is just NSED on the binary cube \([-1, 1]^d\) instead of \(\mathbb{R}^d\)). Note that if we define the type \(t_i\) as the probability of providing a false answer (i.e. as \(1 - p_i\)), then the fault \(f_i = E[d(s_i, z)]\) is exactly \(t_i\).

**Corollary 5** (Anna Karenina principle for the binary One-Coin model). *For the binary domain with the One-Coin model,*

\[
E[\pi_i | t_i] = \mu_t + (1 - 2\mu_t)f_i.
\]

This result can also be easily extended to multiple-choice labels. For details see the full version.

**Cosine similarity** When label vectors are normalized, we have that \(d(x, y) = 2(1 - \cos(x, y))\), meaning that ranking workers by decreasing average cosine similarity (as suggested in [18]) is the same as ranking them by increasing average NSED. Our results above provide sufficient conditions for when this separates good workers from poor ones.

**Rankings** According to the Independent Condorcet Noise ICN model [39], a worker of type \(t_i = f_i^* \in [0, 1]\) observes a vector \(\vec{s}_i \in \{0, 1\}^m\) where for every pair of candidates \(j = (a, b)\), we have \(s_{ij} \neq z_j\) with probability \(f_i^*\).

By definition, the ICN model is a special case of the binary IER model, where \(m = \binom{|C|}{2}\), and the fault level of a type \(t_i\) voter is \(f_i = f_i^*\). We thus get the following result as an immediate corollary of Corollary 5.

**Corollary 6** (Anna Karenina principle for Condorcet Noise model). *For the ranking domain with the independent Condorcet noise model,* \(E[\pi_i] = \mu + (1 - 2\mu)f_i\).

### 4 Aggregation

Our Proximity-based Truth Discovery (P-TD) algorithm is a direct adaptation of the Anna Karenina principle. The idea is very simple:

1. Compute the average distance [or similarity] \(\pi_i\) of every worker;
2. Estimate fault [or competence] \(\bar{f}\) from \(\bar{\pi}\);
3. Aggregate answers, giving higher weight to workers with low fault [high competence].

Our default implementation (denoted P-TD\(^D\)) simply sets weights proportional to the estimated competence, which is in turn proportional to the average similarity, as in [18, 19].

As we make more assumptions on the structure of labels and the statistical model, we can use an appropriate Anna-Karenina theorem to improve Step 2, resulting in a domain-specific implementation P-TD\(^{AK}\). See full version for details on various domain-specific implementations.

While weights can be set heuristically based on (estimated) fault, in some domains this can be guided by theory. In particular, when faults \(\bar{f}\) are known then Aitken [1] showed the optimal weights under Gaussian noise are \(w_i := 1/f_i\); and Grofman et al. [14] showed that for binary labels setting \(w_i^* := \log \frac{1}{1 - f_i}\) (henceforth ‘Grofman weights’) is optimal.\(^4\)

Lastly, we can iteratively repeat the process by computing the weighted average distance to other workers. This iterative P-TD algorithm is denoted by iP-TD.

\(^4\)In fact, this goes back to Neyman and Pearson [27]. We thank Aryeh Kontorovich for this reference.
ALGORITHM 1: (P-TD) FOR RANKING DATA

Input: Dataset \( S \in \mathcal{L}(k)^n \), voting rule \( r \).

Output: Est. fault levels \( \hat{f} \in \mathbb{R}^n \); answers \( \hat{z} \in \mathbb{R}^m \).

Compute \( d_{ii}′ \leftarrow d_{ii}(s_i, s_i′) \) for every pair of workers;

for each worker \( i \in N \) do

set \( \pi_i \leftarrow \frac{1}{n-1} \sum_{i′ \neq i} d_{ii′} \); \hspace{1cm} // Step 1

end

Set \( \bar{\mu} \leftarrow \frac{1}{2n} \sum_{i \in N} \pi_i \);

for each worker \( i \in N \) do

Set \( \hat{f}_i \leftarrow \pi_i - \bar{\mu} \); \hspace{1cm} // Step 2

Set \( w_i \leftarrow \log \frac{1-\hat{f}_i}{\hat{f}_i} \);

end

Set \( \hat{z} \leftarrow r(S, \vec{w}) \); \hspace{1cm} // Step 3

return \( (\hat{f}, \hat{z}) \);

---

Rank aggregation

Note that while our results on fault estimation from the binary domain directly apply (at least to the ICN model), aggregation is more tricky: an issue-by-issue aggregation may result in a non-transitive (thus invalid) solution, and a voting rule is guaranteed to output a valid ranking. While there is much literature on which voting rule comes closest to the ground truth (see, e.g., [3, 11]), we assume here that the voting rule has already been fixed, and are concerned mainly about the effect of weighing voters according to their estimated fault.

The Kemeny rule and optimal aggregation

It is well known that for both Condorcet noise model and Mallows model (a restriction of Condorcet model to transitive orders), when all voters have the same fault level, the maximum likelihood estimator of \( L_z \) is obtained by applying the Kemeny-Young (henceforth, KY) voting rule on \( S \), see Young [39].

The KY rule \( r_{KY} \) computes the binary vector \( y^0 \) that corresponds to the majority applied separately on every pair of candidates (that is, \( y^0 := \text{sign} \sum_{i \leq n} s_i \)); then \( r_{KY}(S) := \arg\min_{L \in \mathcal{L}(C)} d_H(x_L, y^0) \).

A natural conjecture is that applying a weighted version of KY with Grofman weights \( \vec{w}^* \) that is an MLE when fault levels \( f_i \) are known. We did not find any explicit reference to this question or to the case of distinct fault levels in general.\(^5\) As it turns out, the conjecture is correct. This result may be of independent interest. The full proof is in full version.

**Theorem 7.** Suppose that the ground truth \( L_z \) is sampled from a uniform prior on \( \mathcal{L}(C) \). Suppose that instance \( I = (S, \vec{z}) \) is sampled from population \( (f_i)_{i \in N} \) via the ICN model where \( f_i \) are known. Then \( \hat{L} := r_{KY}(S, w^*(f_1), \ldots, w^*(f_n)) \) is the MLE of \( L_z \).

**Proof sketch.** Recall that the original proof for weighted majority [14] divided the profile space into two equal and symmetric parts, according to whether they are classified as positive or negative. We proceed in a similar fashion, carefully combining ingredients from the proofs in [14] and in [39]: the profile space is now partitioned into \( |C|! \) parts instead of 2, one for each potential ground truth ranking \( L' \). We explicitly compare the log-likelihood ratio of \( \hat{L} \) and \( L' \) to show that the likelihood of the selected ranking is (weakly) higher than the likelihood of any other order.

\(^5\)There are some other extensions: [11] deal with a different variation of the noise model where it is less likely to swap pairs that are further apart; and in [36] the KY rule is extended to deal with more general noise models and partial orders.
5 Empirical Evaluation

In the AAAI23 version, we compared our P-TD and iP-TD algorithms to many other truth discovery algorithms on various datasets from multiple domains. Here we focus on the results for aggregating rankings. Recall that the underlying voting rule is assumed to be part of the input and thus we compare under different voting rules.

While many truth-discovery methods (some estimate workers’ competence) have been suggested in the literature, almost all of them are dedicated either to binary labels or to real-valued labels. We therefore consider a simple baseline of unweighted aggregation (UA, that simply applies the selected voting rules without weights), and three other algorithms that are domain-independent: MAS [4], TOP2 and EXP [17]. Another possible baseline is to first use the given voting rule for aggregation, then evaluate workers’ by their distance from the result and aggregate again (D-TD).

Finally, Oracle Aggregation (OA) is a benchmark using the actual fault $f_i$ of each worker.

Datasets We used the following datasets from five different domains. We write the used distance measure in each domain in brackets.

DOTS subjective rankings of four images of dots, according to the number of dots they contain.

PUZZ contain subjective rankings of four images 8-puzzle boards, according to the number of steps from solution.

BUILD a dataset we collected by asking subjects to evaluate the height of buildings in pictures, and extracting the rankings.

DOTS and PUZZ each contain four datasets of tasks with different difficulty. Both datasets are from Mao et al. [25].

For aggregation, we used nine different ordinal voting rules, see full version for details.

In addition we generated synthetic datasets from Mallows model: in SYS.N workers parameter (accuracy) was sampled from a truncated Normal distribution, and in SYS.HS there were 20% accurate workers (‘Hammers’) and 80% random workers (‘Spammers’).

To obtain robust results we sampled $n$ workers and $m$ questions without repetition from each dataset (real or synthetic), and repeated the process at least 1000 times for every combination.

Figure 2: Error on PUZZ5 (Left) and DOTS3 (Right) under nine different voting rules.
Table 2: Results (RI) for rankings datasets, under three different voting rules \((n = 10,\) four ranked alternatives), and on the other complex annotation datasets.

<table>
<thead>
<tr>
<th>Ranking</th>
<th>v. rule</th>
<th>EXP</th>
<th>TOP2</th>
<th>MAS</th>
<th>P-TD**</th>
<th>iP-TD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYN.HS</td>
<td>Borda</td>
<td>-5%</td>
<td>-24%</td>
<td>-14%</td>
<td>-4%</td>
<td>-10%</td>
</tr>
<tr>
<td>SYN.N</td>
<td>Borda</td>
<td>-3%</td>
<td>+6%</td>
<td>+1%</td>
<td>-4%</td>
<td>-6%</td>
</tr>
<tr>
<td>BUILD</td>
<td>Borda</td>
<td>+0%</td>
<td>+0%</td>
<td>+0%</td>
<td>+0%</td>
<td>+0%</td>
</tr>
<tr>
<td>DOTS3</td>
<td>Borda</td>
<td>+2%</td>
<td>+7%</td>
<td>+2%</td>
<td>-1%</td>
<td>+1%</td>
</tr>
<tr>
<td>DOTS5</td>
<td>Borda</td>
<td>+1%</td>
<td>+9%</td>
<td>+4%</td>
<td>-1%</td>
<td>-1%</td>
</tr>
<tr>
<td>DOTS7</td>
<td>Borda</td>
<td>+1%</td>
<td>+13%</td>
<td>+5%</td>
<td>-3%</td>
<td>-2%</td>
</tr>
<tr>
<td>DOTS9</td>
<td>Borda</td>
<td>-3%</td>
<td>+7%</td>
<td>-1%</td>
<td>-6%</td>
<td>-10%</td>
</tr>
<tr>
<td>PUZZ5</td>
<td>Borda</td>
<td>+1%</td>
<td>+1%</td>
<td>+8%</td>
<td>-2%</td>
<td>-2%</td>
</tr>
<tr>
<td>PUZZ7</td>
<td>Borda</td>
<td>-10%</td>
<td>-7%</td>
<td>-18%</td>
<td>-15%</td>
<td>-20%</td>
</tr>
<tr>
<td>PUZZ9</td>
<td>Borda</td>
<td>+6%</td>
<td>+48%</td>
<td>+14%</td>
<td>-5%</td>
<td>-1%</td>
</tr>
<tr>
<td>PUZZ11</td>
<td>Borda</td>
<td>-0%</td>
<td>+6%</td>
<td>+3%</td>
<td>-3%</td>
<td>-3%</td>
</tr>
<tr>
<td>SYN.N</td>
<td>Plurality</td>
<td>-4%</td>
<td>-2%</td>
<td>-5%</td>
<td>-4%</td>
<td>-5%</td>
</tr>
<tr>
<td>BUILD</td>
<td>Plurality</td>
<td>-1%</td>
<td>+20%</td>
<td>-2%</td>
<td>-6%</td>
<td>-6%</td>
</tr>
<tr>
<td>DOTS3</td>
<td>Plurality</td>
<td>+2%</td>
<td>+14%</td>
<td>+2%</td>
<td>-2%</td>
<td>-1%</td>
</tr>
<tr>
<td>PUZZ5</td>
<td>Plurality</td>
<td>+1%</td>
<td>+12%</td>
<td>+1%</td>
<td>-3%</td>
<td>-2%</td>
</tr>
<tr>
<td>SYN.N</td>
<td>Copeland</td>
<td>-12%</td>
<td>-37%</td>
<td>-25%</td>
<td>-27%</td>
<td>-29%</td>
</tr>
<tr>
<td>BUILD</td>
<td>Copeland</td>
<td>-12%</td>
<td>-7%</td>
<td>-17%</td>
<td>-27%</td>
<td>-27%</td>
</tr>
<tr>
<td>DOTS3</td>
<td>Copeland</td>
<td>+0%</td>
<td>+0%</td>
<td>+1%</td>
<td>-0%</td>
<td>-0%</td>
</tr>
<tr>
<td>PUZZ5</td>
<td>Copeland</td>
<td>-7%</td>
<td>-12%</td>
<td>-16%</td>
<td>-19%</td>
<td>-18%</td>
</tr>
</tbody>
</table>

Evaluation  

The error of every algorithm is the distance (as specified above) to the ground truth, averaged over all samples of certain size of a particular dataset.

In the tables, we compute for each algorithm its Relative Improvement \(RI(Alg) := \frac{Err(Alg) - Err(UA)}{Err(UA)}\), where UA serves as a baseline. Thus RI is in the range \([-1, 1]\) where negative numbers mean improvement over UA.

In some cases we see that one algorithm has slightly higher average error (on the graphs) but lower RI, or that the gap in RI is more substantial. This is since the graphs average over instances of varying difficulty, so instances with high baseline error have more effect.

Results

We generated populations from Mallows distribution, where the proto population (distribution of \(\phi_i\)) was either a clipped Normal distribution (we used \(N(0.65, 0.15)\) and \(N(0.85, 0.15)\)), or a “hammer-spammer” (HS) distribution (20% hammers with \(\phi_i = 0.3\), and 80% spammers with \(\phi_i = 0.99\) which is just slightly better than random).

We can see in Table 2 that P-TD yields substantial improvement over unweighted aggregation, and beats the other, more sophisticated algorithms on most datasets. Note that while these algorithms are sometimes best (e.g. TOP2 on some synthetic datasets) they are highly unstable and often perform worse than UA.

Fig. 2 shows how P-TD, iP-TD and D-TD improve as the dataset grows, under all nine voting rules. Again we can see that P-TD and iP-TD are better, especially when using simple voting rules: when using KY this is less substantial.

Finally, Fig.3 compares P-TD to D-TD (on synthetic data), showing improvement for all voting rules and scales.
6 Conclusion

Average proximity can be used as a general scheme to estimate workers’ competence in a broad range of truth discovery and crowdsourcing scenarios. Due to the “Anna Karenina principle,” we expect the answers of competent workers to be much closer to others, than those of incompetent workers, even under very weak assumptions on the domain and the noise model. Under more explicit assumptions, the average distance accurately estimates the true competence.

The above results suggest an extremely simple, general and practical algorithm for truth-discovery (the P-TD algorithm), that weighs workers by their average proximity to others, and can be combined with most aggregation methods. This is particularly useful in the context of existing crowdsourcing systems where the aggregation rule may be subject to constraints due to legacy, simplicity, explainability, legal, or other considerations (e.g. a voting rule with certain axiomatic properties). In addition, average proximity is simple and flexible enough so we can modify it to deal with challenges outside the scope of the current paper, such as partial data [8, 16, 21]; semi-supervised learning [38]; or worker’s competence that varies across task types [4].

Despite its simplicity, the P-TD algorithm substantially improves the outcome compared to unweighted aggregation. It is also competitive with other, more sophisticated algorithms, especially in the common case of moderate input size. We thus conclude that the average similarity heuristic is indeed a frustratingly easy—and practical—tool for crowdsourcing.

An obvious shortcoming of P-TD is that a group of workers that submit similar labels (e.g. by acting strategically) can boost their own weights. Future work will consider how to identify and/or mitigate the effect of such groups.

Acknowledgements

This research was supported by the Israel Science Foundation (ISF; Grant No. 2539/20).

Previous versions of this paper have been rejected from 7 (seven!) AI conferences. The current version is much improved thanks to the comments, suggestions, and references provided by the (many) reviewers along the way.

References


[23] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. A survey on truth discovery. *ACM SIGKDD Explorations Newsletter*, 17(2):1–16, 2016.


[41] Bo Zhao and Jiawei Han. A probabilistic model for estimating real-valued truth from conflicting sources. *Proc. of QDB*, 1817, 2012.